Question-guided Visual Compression with Memory Feedback for Long-Term Video Understanding

Supplementary Material

This supplementary material provides: (i) detailed training setup and hyper-parameters of the Question-guided Visual Compression with Memory Feedback (QViC-MF) framework, as well as the inference configuration (Section A); (ii) a description of the masking strategies used in the visual compressor and their relation to Question-guided Multimodal Selective Attention (QMSA) (Section B); (iii) an analysis of the hyper-parameters for relevance score computation (Section C); (iv) full benchmark results on MLVU-dev, LVBench, VideoMME, and VNBench (Section D); (v) additional ablation studies on feedback frames, relevance computation, context tokens, and memory capacity (Section E); (vi) a computational complexity analysis of QViC-MF (Section F); (vii) a report on inference time and VRAM usage (Section G); (viii) an analysis of the reliability of relevance-based memory retrieval using needle hit-rate evaluation on VNBench Long (Section H); and (ix) a discussion of task-dependent behavior, limitations, and failure cases (Section I).

A. Training Details

Dataset. We use an 83K-sample subset obtained by randomly sampling 5% of the LLaVA-Video-178K dataset [19], which contains a wide range of instruction-following tasks such as captioning, open-ended VQA, and multiple-choice VQA. We adopt this subset to ensure a manageable computational footprint while maintaining the task diversity of the original dataset.

Protocols. The training hyper-parameters are summarized in Table 1. We train two modules: the visual compressor and the context seed embeddings, while keeping all other components frozen. LoRA [5] is applied to the base model of the visual compressor (LLaVA-Video-7B-Qwen2 [19]). The decoder LLM (Qwen2-7B [14]), which remains frozen during training, is also based on the LLaVA-Video-7B-Qwen2 architecture. During training, the model operates in a one-way compression setting without applying feedback from the context memory. Accordingly, we set the number of clip frames to $K = 64$ and the number of recalled frames to $K_{\mathrm{r}} = 0$, so that the visual compressor receives $K_{\mathrm{v}} = K + K_{\mathrm{r}} = 64$ frames, matching the input specification of the base model (LLaVA-Video-7B-Qwen2).

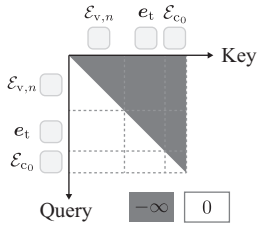
For each sample from LLaVA-Video-178K, we uniformly sample K frames from the video and feed their visual embeddings, together with the question embeddings, into the visual compressor to obtain K context embeddings.

Setting	
Batch size	1
Gradient accumulation steps	4
Learning rate	1e-4
Learning scheduler	Cosine decay
Warm-up ratio	0.03
Weight decay	0
Number of epochs	1
Optimizer	AdamW
DeepSpeed stage	3
LoRA rank	64
LoRA alpha	16
LoRA dropout	0.05
Context tokens C	16
Context memory capacity L	256
Clip frames K	64
Recalled frames K_{r}	0
Visual encoder	Frozen
Context seed embedding	Trainable
Visual compressor	Trainable
Decoder LLM	Frozen

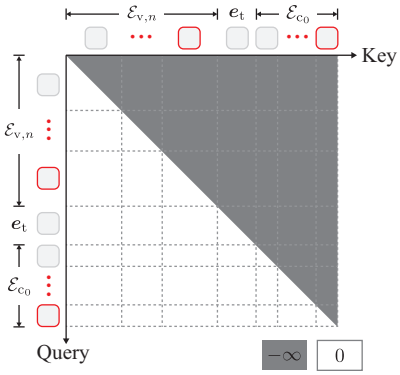
Table 1. Training settings of our QViC-MF framework. During training, the model operates in a one-way compression setting without applying feedback from the context memory, and thus the number of recalled frames is fixed to $K_{\mathrm{r}} = 0$.

To avoid overfitting the decoder LLM to a fixed and relatively small memory size, and to address the fact that the K sampled frames do not fill the entire memory when $L > K$ (with $L = 256$), we randomize the effective memory length during training. Specifically, we interpolate the K context embeddings using nearest-neighbor interpolation to a sequence length uniformly sampled between K and L, and use this sequence as the content of the context memory. Finally, the context memory and the question are fed to the decoder to generate an answer, and a standard cross-entropy loss with respect to the ground-truth answer is computed. Minimizing this loss updates the trainable context seed embeddings and the parameters of the visual compressor.

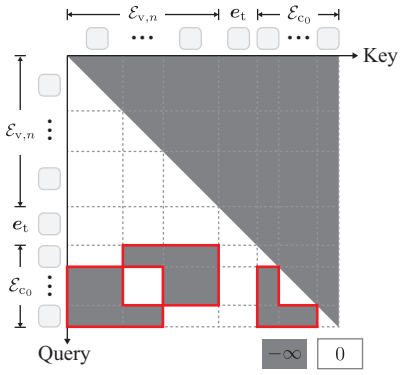
The inference-time configuration, including the hyper-parameters for relevance score computation, is summarized in Table 2, and the overall inference pipeline follows the procedure described in Section 3 of the main paper.



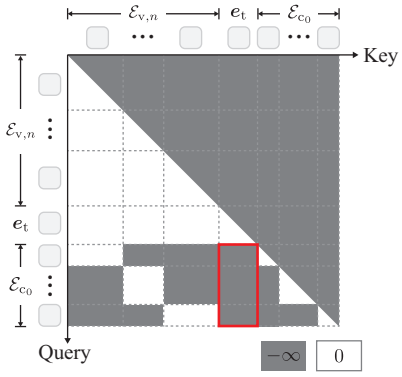
(a) Single-frame causal masking



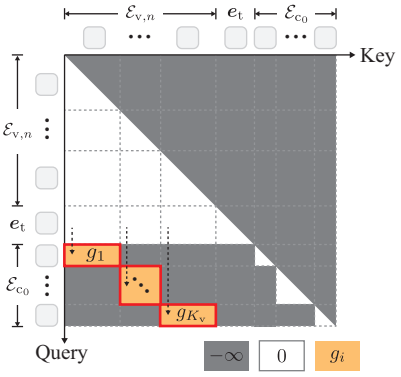
(b) Multi-frame causal masking



(c) Framewise Ctx2Vis masking



(d) Blocking Ctx2Txt



(e) Guiding Ctx2Vis (QMSA)

Figure 1. Variations of masking strategies used in the visual compressor. (a) A simple baseline that performs conventional frame-to-frame compression using single-frame causal masking. This corresponds to the "Single-frame" setting in Figure 4 of the main paper. (b) A naive extension of (a) to multi-frame compression, corresponding to the "Vanilla" setting in Table 3 of the main paper. (c) A variant of (b) that applies framewise context-to-visual masking, matching the masking matrix $\mathbf{M}$ in QMSA. (d) A variant of (c) that additionally blocks context-to-text attention, corresponding to applying both $\mathbf{M}$ and the blocking matrix $\mathbf{B}$ in QMSA. (e) The complete QMSA configuration, which further introduces guiding for context-to-visual attention. This corresponds to applying all three matrices: $\mathbf{M}$, $\mathbf{B}$, and $\mathbf{G}$. As in standard attention masking, $-\infty$ blocks token-pair attention (ignored after softmax), 0 passes the logits unchanged, and the guiding values g_i act as bias terms added to the attention logits.

Setting	
Context tokens C	16
Context memory capacity L	256
Clip frames K	32
Recalled frames K_{r}	32
Top heads K_{h} for computing $r_{n,i}$	5
Layer range $[L_1, L_2]$ for computing $r_{n,i}$	[17, 20]

Table 2. Inference settings of QViC-MF, including the hyperparameters for relevance score computation. Parameters shared with training settings are repeated for clarity.

B. Masking Strategies in the Visual Compressor

Figure 1 illustrates several masking strategies used in the proposed visual compressor. These masking schemes determine how each token attends to past or future visual, text, and context tokens across frames. The strategies play a crucial role not only in controlling the temporal receptive field but also in preventing information leakage that could lead to compression hallucination. Moreover, these strategies provide the structural foundations required for question-adaptive visual compression. In particular, our proposed Question-guided Multimodal Selective Attention (QMSA) builds on these masking components to selectively preserve

information relevant to the given question while avoiding undesirable cross-frame or cross-modal interactions.

Single-frame Causal Masking (Figure 1 (a)). The single-frame causal masking corresponds to conventional frame-to-frame attention. Each token can attend only to tokens within the same frame and only to those occurring earlier in temporal order. While this avoids any cross-frame information mixing, it also prevents the model from capturing temporal dependencies across frames. As a result, the visual compressor cannot exploit temporal continuity or motion cues, limiting its ability to perform temporally informed compression. In addition, because this configuration disallows any interaction across frames, it cannot support memory-feedback-based visual compression as used in QViC-MF. This configuration serves as a baseline and matches the "Single-frame" setting shown in Figure 4 of the main paper.

Multi-frame Causal Masking (Figure 1 (b)). A straightforward extension is the multi-frame causal masking, where each token may attend to all tokens from past frames while future frames remain masked. Although this enables temporal modeling and, in principle, allows the visual compressor to accept memory-feedback inputs across frames, it causes the context embeddings to become an entangled representation of the entire input clip rather than frame-local representations. As a result, a single context embedding no longer corresponds to the visual content of an individual frame, causing frame-wise addition or removal to produce a context memory whose frame-level semantics collapse. This configuration corresponds to the "Vanilla" setting reported in Table 3 of the main paper.

Framewise Context-to-Visual Masking (Figure 1 (c)). The framewise context-to-visual masking matrix $\mathbf{M}$ addresses the limitations of single-frame and multi-frame causal masking. This mask enables the visual compressor to capture temporal relationships during compression while keeping each context embedding as an independent frame-local representation. However, as shown in Figure 3 (a) of the main paper, text information may still leak into the context tokens, distorting the visual content of the context embeddings and causing compression hallucination.

Blocking Context-to-Text Masking (Figure 1 (d)). The blocking matrix $\mathbf{B}$ prevents context-to-text attention and mitigates compression hallucination. However, as shown in Figure 3 (b) of the main paper, removing text-context interaction hinders question-adaptive compression and reduces the model's ability to preserve information relevant to the given question.

Guiding Context-to-Visual Masking (QMSA; Figure 1 (e)). The full QMSA configuration is obtained by introducing a guiding matrix $\mathbf{G}$ in addition to $\mathbf{M}$ and $\mathbf{B}$. The matrix $\mathbf{G}$ enables controlled text-to-visual information flow into the context tokens, allowing the visual compressor to perform question-adaptive compression. With this design, QMSA resolves the limitations of the previous masking strategies and supports context-aware visual compression while maintaining temporal causality and preventing undesired cross-modal interference.

C. Effects of Hyper-parameters in Relevance Score Computation

Figure 2 illustrates the effects of two key hyper-parameters in our relevance score computation: the number of top attention heads K_{h} and the layer range $[L_1, L_2]$ from which the text-to-visual attention weights are extracted. We adopt the inference-time configuration (Table 2), namely $K_{\mathrm{h}} = 5$ and $[L_1, L_2] = [17, 20]$. We use a sample from the Needle QA (NQA) task in MLVU. The frames relevant to the question (around 140 seconds) are highlighted in red, while all other frames are unrelated to the query.

The middle row of Figure 2 shows the mean text-to-visual attention weights. For each condition (all heads vs. top-K_{h} heads and all layers vs. layers L_1–L_2), we plot the head-averaged attention weights for each layer, with different colors corresponding to different layers. Focusing on the heads that respond strongly to the question leads to more discriminative attention patterns across layers. In addition, we observe that text-visual interactions become particularly pronounced in the middle-to-late layers, suggesting that this layer range contributes more strongly to identifying question-relevant frames. The bottom row shows the relevance scores obtained by averaging these layer-wise mean attention weights across layers. Using all heads and all layers yields noisy and weakly discriminative signals, whereas selecting K_{h} and layers L_1–L_2 produces sharp peaks precisely at the question-relevant frames, resulting in much clearer relevance scores.

D. Detailed Results on Benchmarks

Tables 3 to 6 present the detailed benchmark results for MLVU-dev [21], LVBench [13], VideoMME [2], and VNBench [20]. Across all tables, we use two notations to indicate the input sampling strategy of each method: (i) Uniform Sampling ("N frm"), which evenly samples N frames per video, and (ii) Frame Rate Sampling ("N fps"), which samples videos at N frames per second.

MLVU-dev. MLVU-dev [21] evaluates diverse long-term video understanding tasks using videos ranging from 3

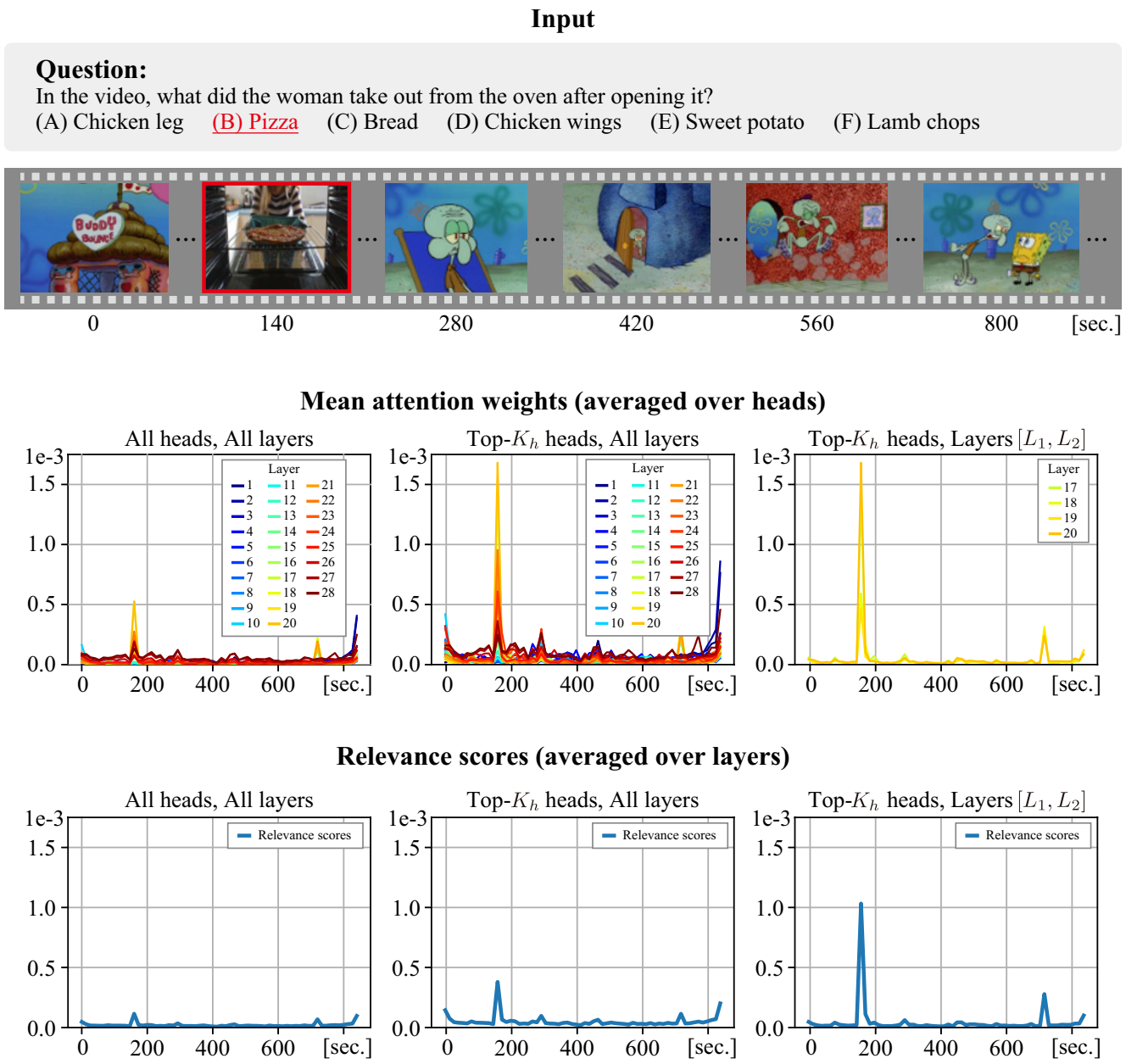


Figure 2. Visualization of attention patterns and relevance scores for a sample from the Needle QA (NQA) task in MLVU. The frames relevant to the question are highlighted with red boxes (around 140 seconds). The middle row shows the mean text-to-visual attention weights: for each condition (all heads vs. top-$K_{\mathrm{h}} = 5$ heads, and all layers vs. layers $L_1 = 17$ to $L_2 = 20$), we plot the head-averaged attention weights for each layer, with different colors indicating different layers. The bottom row shows the corresponding relevance scores obtained by averaging these layer-wise mean attention weights across layers. These results show that appropriately selecting heads and layers leads to more discriminative relevance scores aligned with the question. For visualization, we assign each visual token to the frame it originates from based on its fixed sequence position and aggregate the text-to-visual attention over the tokens of each frame. This provides an approximate yet practical frame-level attribution even in deep layers.

minutes to over 2 hours. Following the official protocol, we report the mean task accuracy (M-Avg). The benchmark consists of seven tasks: Topic Reasoning (TR), Anomaly Recognition (AR), Needle QA (NQA), Ego Reasoning (ER), Plot QA (PQA), Action Order (AO), and Action Count (AC). Compared with previous state-of-the-art (SoTA) methods, QViC-MF achieves a +1.7% improvement in M-Avg. The gain is particularly pronounced in the AO

Model	Size (LLM)	Input	M-Avg	TR	AR	NQA	ER	PQA	AO	AC
GPT-4o [1]	-	0.5 fps	64.6	-	-	-	-	-	-	-
MovieChat [11]	7B (Vicuna-v0)	2048 frm	25.8	29.5	25.0	24.2	24.7	25.8	28.6	22.8
LLaMA-VID [7]	7B (Vicuna-v1.5)	1 fps	33.2	50.8	34.5	30.1	32.7	32.5	23.9	27.8
MA-LMM [4]	7B (Vicuna-v1.1)	1000 frm	36.4	51.9	35.5	43.1	38.9	35.8	25.1	24.3
LLaVA-Mini [18]	7B (Vicuna-v1.5)	1 fps	42.8	76.0	50.0	44.5	37.5	49.0	24.3	18.4
DynFocus [3]	7B (Vicuna-v1.5)	32 frm	49.6	76.2	60.9	55.5	41.5	54.0	26.8	32.8
LongVA [17]	7B (Qwen2)	256 frm	56.3	-	-	-	-	-	-	-
Video-XL [10]	7B (Qwen2)	256 frm	64.9	-	-	-	-	-	-	-
LongVU [9]	7B (Qwen2)	1 fps	65.4	**87.5**	**76.0**	76.3	59.4	71.6	58.3	29.0
Frame-Voyager [15]	7B (Qwen2)	8 frm	65.6	-	-	-	-	-	-	-
Flash-VStream [16]	7B (Qwen2)	1 fps	66.3	-	-	-	-	-	-	-
LLaVA-Video [19]	7B (Qwen2)	64 frm	67.9	85.9	67.5	80.3	67.1	75.3	58.3	**40.8**
QViC-MF (Ours)	7B (Qwen2)	1 fps	69.0	84.0	66.5	84.2	71.3	76.4	68.3	32.0
QViC-MF (Ours)	7B (Qwen2)	2 fps	**69.6**	84.4	68.5	**84.8**	**72.7**	**76.8**	**69.5**	30.1

Table 3. Detailed comparison of the proposed QViC-MF framework with SoTA methods on the MLVU-dev. The best and second best results are highlighted in bold and underlined, respectively.

Model	Size (LLM)	Input	Overall	KIR	EU	ER	TG	Rea	Sum
GPT-4o [1]	-	348 frm	30.8	34.5	27.4	33.0	25.0	27.5	24.1
Gemini 1.5 Pro [8]	-	3600 frm	33.1	39.3	30.9	32.1	31.8	27.0	32.8
MovieChat [11]	7B (Vicuna-v0)	2048 frm	22.5	25.9	23.1	21.3	22.3	24.0	17.2
LLaMA-VID [7]	7B (Vicuna-v1.5)	1 fps	23.9	23.4	21.7	25.4	26.4	26.5	17.2
DynFocus [3]	7B (Vicuna-v1.5)	200 frm	31.5	31.3	32.6	30.1	25.5	33.3	30.5
LLaVA-Video [19]	7B (Qwen2)	64 frm	41.8	41.6	39.3	42.3	33.2	47.8	**32.8**
Flash-VStream [16]	7B (Qwen2)	1 fps	42.0	-	-	-	-	-	-
QViC-MF (Ours)	7B (Qwen2)	1 fps	**50.3**	**54.6**	**47.5**	52.3	**41.4**	50.8	31.0
QViC-MF (Ours)	7B (Qwen2)	2 fps	50.2	54.3	**47.5**	**52.4**	40.0	**51.2**	29.3

Table 4. Detailed comparison of the proposed QViC-MF framework with SoTA methods on the LVBench. The best and second best results are highlighted in bold and underlined, respectively.

Model	Size (LLM)	Input	Overall	Short	Medium	Long
GPT-4o [1]	-	384 frm	71.9	80.0	70.3	65.3
Gemini 1.5 Pro [8]	-	1 fps	70.3	78.8	68.8	61.1
DynFocus [3]	7B (Vicuna-v1.5)	16 frm	44.1	50.9	43.7	37.3
LongVA [17]	7B (Qwen2)	128 frm	52.6	61.1	50.4	46.2
Video-XL [10]	7B (Qwen2)	128 frm	55.5	64.0	53.2	49.2
Frame-Voyager [15]	7B (Qwen2)	8 frm	57.5	67.3	56.3	48.9
Flash-VStream [16]	7B (Qwen2)	1 fps	61.2	72.0	61.1	50.3
LLaVA-Video [19]	7B (Qwen2)	64 frm	63.3	-	-	-
QViC-MF (Ours)	7B (Qwen2)	1 fps	62.4	74.3	60.4	52.6
QViC-MF (Ours)	7B (Qwen2)	2 fps	**63.4**	**74.6**	**61.6**	**54.0**

Table 5. Detailed comparison of the proposed QViC-MF framework with SoTA methods on the VideoMME without subtitle setting. The best and second best results are highlighted in bold and underlined, respectively.

task, which requires completing long-range temporal event sequences; QViC-MF improves AO accuracy by +11.2%. This substantial gain indicates that the proposed memory-feedback mechanism effectively supports long-horizon temporal reasoning.

LVBench. LVBench [13] serves as a long-term video understanding benchmark with videos up to roughly two hours in duration. Following the official evaluation protocol, we report the overall accuracy across all questions. The benchmark covers six tasks: Key Information Retrieval (KIR),

Method	Size (LLM)	Input	Overall	Retrieval				Ordering				Counting			
				Edit	Insert-1	Insert-2	Avg	Edit	Insert-1	Insert-2	Avg	Edit-1	Edit-2	Insert	Avg
ALL (10-180 sec.)															
GPT-4o [1]	-	1 fps	64.4	100.0	98.0	87.3	95.3	88.4	86.6	45.2	73.4	36.8	0.0	36.1	24.5
Gemini 1.5 Pro [8]	-	1 fps	66.7	100.0	96.0	76.0	90.7	90.7	95.3	32.7	72.9	60.7	7.3	42.0	36.7
LLaMA-VID [7]	7B (Vicuna-v1.5)	1 fps	10.8	28.0	28.0	19.3	25.1	0.7	0.0	0.0	0.2	4.0	2.7	14.7	7.1
Qwen2-VL [12]	7B (Qwen2)	1 fps	33.9	98.0	76.0	33.3	69.1	16.0	12.7	8.7	12.4	26.0	9.3	24.7	20.0
LLaVA-OneVision [6]	7B (Qwen2)	64 frm	51.8	88.7	87.3	**55.3**	77.1	70.0	50.0	37.3	52.4	41.3	8.7	27.3	25.8
Video-XL [10]	7B (Qwen2)	1 fps	61.6	98.0	93.3	48.7	80.0	89.3	77.3	**75.3**	80.6	38.7	7.3	26.0	24.0
LLaVA-Video [19]	7B (Qwen2)	64 frm	62.5	90.0	88.7	52.0	76.9	78.7	77.3	67.3	74.4	**54.7**	11.3	**42.7**	**36.2**
QViC-MF (Ours)	7B (Qwen2)	1 fps	63.0	**99.3**	98.7	39.3	79.1	88.7	**84.7**	66.7	80.0	40.7	**16.0**	32.7	29.8
QViC-MF (Ours)	7B (Qwen2)	2 fps	**64.7**	**99.3**	**100.0**	42.0	**80.4**	**91.3**	**84.7**	69.3	**81.8**	45.3	14.0	36.7	32.0
Long (60-180 sec.)															
GPT-4o [1]	-	1 fps	56.3	100.0	98.0	84.0	94.0	73.5	80.0	26.5	60.0	22.3	2.0	20.4	14.9
Gemini 1.5 Pro [8]	-	1 fps	65.1	100.0	94.0	68.0	87.3	90.0	96.0	34.0	73.3	56.0	10.0	38.0	34.7
LLaMA-VID [7]	7B (Vicuna-v1.5)	1 fps	6.4	14.0	16.0	14.0	14.7	0.0	0.0	2.0	0.7	4.0	0.0	8.0	4.0
Qwen2-VL [12]	7B (Qwen2)	1 fps	33.6	96.0	80.0	26.0	67.3	24.0	12.0	8.0	14.7	34.0	6.0	16.0	18.7
LLaVA-OneVision [6]	7B (Qwen2)	64 frm	36.0	72.0	62.0	**40.0**	58.0	42.0	30.0	26.0	32.7	24.0	10.0	18.0	17.3
LLaVA-Video [19]	7B (Qwen2)	64 frm	40.4	70.0	66.0	38.0	58.0	42.0	42.0	32.0	38.7	36.0	**14.0**	24.0	24.7
QViC-MF (Ours)	7B (Qwen2)	1 fps	56.9	**100.0**	96.0	34.0	76.7	82.0	70.0	48.0	66.7	44.0	10.0	**28.0**	**27.3**
QViC-MF (Ours)	7B (Qwen2)	2 fps	**58.7**	**100.0**	**100.0**	38.0	**79.3**	**86.0**	**72.0**	**50.0**	**69.3**	**46.0**	8.0	**28.0**	**27.3**

Table 6. Detailed comparison of the proposed QViC-MF framework with SoTA methods on the VNBench. The best and second best results are highlighted in bold and underlined, respectively.

Event Understanding (EU), Entity Recognition (ER), Temporal Grounding (TG), Reasoning (Rea), and Summarization (Sum). Compared with previous SoTA methods, QViC-MF achieves a substantial improvement of +8.3% in overall accuracy. The gains are particularly notable in four tasks requiring spatiotemporal focus and long-range event memory: +13.0% in KIR, +8.2% in EU, +10.1% in ER, and +8.2% in TG. These improvements suggest that QViC-MF's memory-feedback mechanism effectively supports the retrieval and integration of key events over extended temporal spans.

VideoMME. VideoMME [2] covers multi-domain video comprehension with durations ranging from short clips to hour-long videos. We follow the video-only evaluation setting without subtitles and report the average accuracy across all samples. QViC-MF outperforms previous SoTA methods across all duration ranges. The improvement is most pronounced for the Long range, where QViC-MF achieves a +3.7% gain, demonstrating its advantage in understanding long-duration videos and maintaining consistent performance as the temporal horizon increases.

VNBench. VNBench [20] targets the challenging Needle-in-a-Haystack (NIAH) scenario and adopts a 4-try circular evaluation protocol, where a prediction is considered correct only if all four trials are answered accurately. We report results for both the full duration range ALL (10–180 sec) and the long-duration subset Long (60–180 sec), covering Retrieval, Ordering, and Counting tasks as well as their overall averages. QViC-MF achieves improvements of +2.2% in the ALL overall score and a substantial +18.3% in the Long overall score compared to prior SoTA methods. The gains in the long-duration setting are particularly large: +21.3% in Retrieval and +30.6% in Ordering. These results highlight the effectiveness of QViC-MF's spatiotemporally selective compression and memory-feedback mechanisms in the challenging long-horizon NIAH scenario, where identifying sparse, question-relevant events is crucial.

E. Further Ablation Studies

The ablations in this section focus on hyper-parameters that are intrinsic to the core mechanisms of QViC-MF, namely the memory-feedback process and the question-guided visual compression module. These factors directly affect how the model retrieves long-range evidence and preserves question-relevant information, and are therefore essential for understanding the behavior and effectiveness of QViC-MF. Tables 7 to 11 present the detailed ablation studies conducted in our work. All experiments vary one specific parameter in each ablation while keeping all other settings identical to the inference configuration described in Table 2. All results are reported using 2 fps input sampling.

Number of recall frames. Table 7 presents the ablation study on the balance between the number of current clip frames K and recall frames K_r. As noted in Section A, the visual compressor operates with a 64-frame input, and we divided this budget between K and K_r. We observe that $K_r = 0$ (no feedback) results in poor performance, es-

K, K_r	MLVU test	VNBench ALL	VNBench Long
64, 0	48.7	62.4	54.7
48, 16	58.3	63.8	**58.9**
32, 32	**59.4**	**64.7**	58.7
16, 48	58.0	63.9	57.1

Table 7. Ablation study on the balance between the number of current clip frames K and recall frames K_r. All other inference hyper-parameters follow the settings in Table 2. The magenta-highlighted configuration corresponds to the default setting used in Table 2. The best and second-best results are highlighted in bold and underlined, respectively.

K_h	MLVU test	VNBench ALL	VNBench Long
1	57.1	**64.7**	57.8
5	**59.4**	**64.7**	**58.7**
14	56.9	**64.7**	58.2
28 (all heads)	57.6	64.6	58.0

Table 8. Ablation study on the number of heads K_h used in relevance score computation. Setting $K_h = 28$ corresponds to using all attention heads. The magenta-highlighted configuration denotes the default setting used in Table 2. The best and second-best results are highlighted in bold and underlined, respectively.

pecially for long videos, whereas introducing recall frames consistently improves accuracy. However, increasing K_r also decreases throughput: as K becomes smaller, each 64-frame window covers a shorter portion of the video, increasing the number of windows required to process the entire sequence. Balancing these two factors, the configuration $(K, K_r) = (32, 32)$ provides the best trade-off between accuracy and computational efficiency.

Hyper-parameters for relevance score computation. Table 8 summarizes the ablation results for the number of attention heads K_h used in computing the relevance score $r_{n,i}$ in Eq. (4), and Table 9 presents the corresponding ablation for the layer range $[L_1, L_2]$. The results show that using $K_h = 5$ and layers $[L_1, L_2] = [17, 20]$ yields the best overall performance. The difference between MLVU and VNBench can be attributed to how frequently the context memory is updated during inference. MLVU contains many long videos whose total number of frames often exceeds the context memory capacity $L = 256$. In such cases, the model repeatedly prunes and updates memory entries making performance more sensitive to the quality of the relevance-score estimation. This explains why the choice of K_h and $[L_1, L_2]$ has a more noticeable impact on MLVU. In contrast, most videos in VNBench fall within the 256-frame limit, so entry pruning rarely occurs. As a result, the

$[L_1, L_2]$	MLVU test	VNBench ALL	VNBench Long
[1, 28] (all layers)	55.8	**64.9**	**58.9**
[17, 20]	**59.4**	64.7	58.7

Table 9. Ablation study on the layer range $[L_1, L_2]$ used in relevance score computation. Setting $[L_1, L_2] = [1, 28]$ corresponds to using all layers. The magenta-highlighted configuration denotes the default setting used in Table 2. The best results are highlighted in bold.

C	MLVU test	VNBench ALL	VNBench Long
1	54.5	56.7	51.6
4	54.3	**65.3**	**60.2**
16	**59.4**	64.7	58.7

Table 10. Ablation study on the number of context tokens per frame C. The magenta-highlighted configuration denotes the default setting used in Table 2. The best results are highlighted in bold.

L	M-Avg	NQA	ER	AO	TQA
32	56.1	68.3	73.6	48.6	48.8
64	58.1	**76.7**	**79.3**	54.3	44.2
128	57.7	73.3	**79.3**	54.3	46.5
256	**59.4**	71.7	71.7	**61.4**	**55.8**

Table 11. Ablation study on the context memory capacity L. We report results on the Needle QA (NQA), Ego Reasoning (ER), Action Order (AO), and Tutorial QA (TQA) tasks of MLVU-test, together with their mean average (M-Avg). The magenta-highlighted configuration denotes the default setting used in Table 2. The best and second-best results are highlighted in bold and underlined, respectively.

relevance score plays a smaller role during inference, and the performance differences across configurations become correspondingly smaller.

Number of context tokens. Table 10 shows the effect of varying the number of context tokens per frame C. For MLVU, performance drops sharply once C is reduced from 16 to 4 or 1. This trend suggests that the video content and tasks in MLVU, which often involve complex actions and temporal dependencies, require sufficiently detailed per-frame representations. With a larger value of C (e.g., $C = 16$), the model has enough capacity to capture the necessary visual information within each frame, whereas reducing C limits this capacity and leads to a loss of important cues. In contrast, VNBench remains relatively stable when C is reduced from 16 to 4, with only a substantial performance degradation occurring at $C = 1$. This robust-

ness reflects the nature of VNBench, where the relevant visual evidence is typically simpler (e.g., static scenes or short text-like cues), making the tasks less demanding in terms of per-frame representation quality. These results indicate that, for general long-video understanding scenarios as in MLVU, a moderate number of context tokens (e.g., $C = 16$) is preferable, providing sufficient capacity to encode frame-level information while maintaining stable downstream performance.

Context memory capacity. We analyze the effect of the context memory capacity L on the Needle QA (NQA), Ego Reasoning (ER), Action Order (AO), and Tutorial QA (TQA) tasks of MLVU-test, as shown in Table 11. Overall, the best mean performance (M-Avg) is achieved at $L = 256$, while $L = 64$ and $L = 128$ yield comparable averages. However, the optimal capacity depends on how the task distributes question-relevant information over time. For NQA and ER, where the crucial evidence is concentrated in a relatively small number of frames (e.g., a short needle action or a specific event/instance), smaller memory capacities ($L = 64$ or $L = 128$) tend to perform better, as they can retain the key frames while reducing the proportion of irrelevant distractor memories. In contrast, AO and TQA involve multiple actions or instances spread over longer temporal spans, and thus benefit from a larger memory capacity ($L = 256$), which reduces the risk of missing important frames even at the cost of storing more distractors. These results suggest that the trade-off between preserving all question-relevant evidence and suppressing distractor memories is crucial when choosing the memory capacity. As a direction for future work, mechanisms such as explicit forgetting or pruning of unused memories could help mitigate the impact of distractors when using large memory capacities.

F. Computational Complexity

Given a video of length T, QViC-MF processes it sequentially in fixed-size clips. At each step, the visual encoder and compressor operate on a bounded window of $K_{\mathrm{v}} = K + K_{\mathrm{r}}$ frames, resulting in a bounded encoder token length N_{enc} independent of T. Both the context memory and decoding operate on a fixed token budget independent of T, since the memory capacity is fixed and the decoder is invoked once. As a result, the total end-to-end computational complexity scales linearly with the video length, i.e., $O(T)$. Memory feedback may re-encode recalled frames, but the total number of encoded frames scales as $(1 + K_{\mathrm{r}}/K)T$, which incurs only a constant-factor overhead compared with one-way frame-wise methods. This linear-in-T scaling is also shared by recent memory-based approaches such as Flash-VStream [16], although architectural choices and constant factors differ.

G. Inference Time and VRAM Usage

On a system equipped with an H200 GPU (141 GB VRAM), Intel(R) Xeon(R) Platinum 8558 CPU @4.00GHz, and 3.0 TiB RAM, QViC-MF can construct the context memory at 28 frames per second.

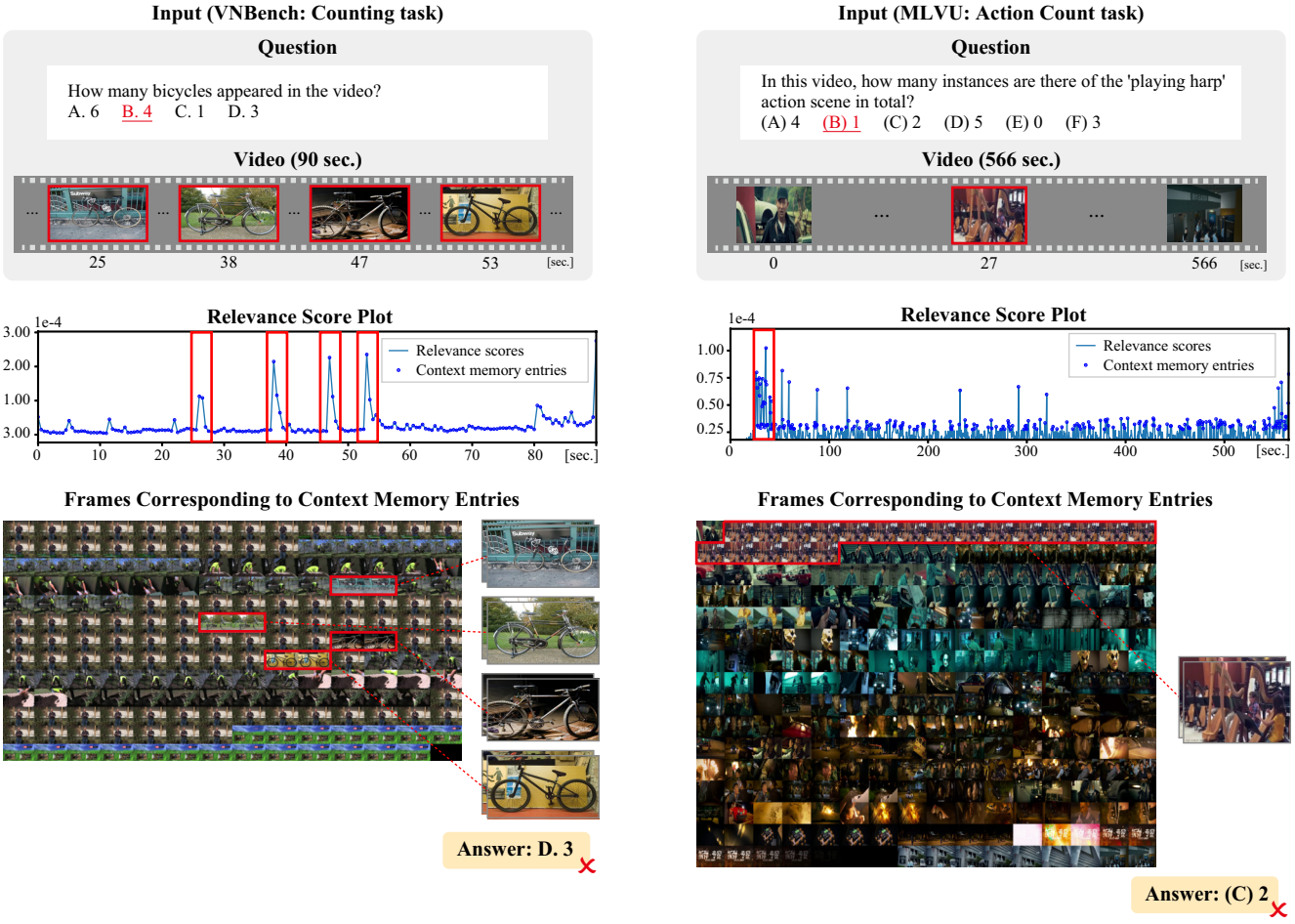
As mentioned in the main paper, QViC-MF uses the same base LLM (Qwen2-7B [14], 15 GiB) for both the visual compressor and the decoder, and applies lightweight tuning with LoRA [5] (0.7 GiB per LLM). By sharing LLM parameters, the total VRAM usage of QViC-MF can be kept around 16.4 GiB, comparable to typical LMMs such as LLaVA-Video-7B-Qwen2 [19]. In the current implementation, however, the visual compressor and decoder are instantiated as separate models, resulting in a total VRAM usage of 30.7 GiB. This can be reduced through the shared-parameter implementation described above. Each context embedding occupies less than 1 MiB, so storing several hundred embeddings in the context memory increases VRAM usage by less than 1 GiB. QViC-MF compresses input videos by sequentially adding context embeddings to a fixed-length context memory. This design ensures that VRAM usage scales with the memory length L, enabling stable resource usage even as video length increases.

Overall, QViC-MF achieves both efficient inference speed and low VRAM consumption for practical applications in long-term video understanding, thanks to its memory-efficient design.

H. Reliability of Relevance-based Memory Retrieval

The relevance scores used in QViC-MF are derived from internal attention statistics and serve as lightweight routing signals for memory retention and retrieval. While such signals may be imperfect, the design of QViC-MF prevents error amplification. The context memory has a fixed capacity with top-K_{r} retrieval, and recalled frames are re-encoded together with the current clip, updating only their corresponding memory slots. These mechanisms prevent self-reinforcing feedback loops.

To empirically evaluate the reliability of relevance-based retrieval, we measure the needle hit-rate on VNBench Long using annotated needle timestamps. To avoid trivial hits from storing all frames, we evaluate only videos longer than 60 seconds, sample input frames at 2 fps, and constrain the context memory to 64 frames. We report the sample-level needle hit-rate, where a sample is counted as correct only if all needles in the video are retained in the context memory. Under this constrained setting, relevance-based feedback achieves a 99.8% hit-rate (1796/1800 samples), whereas uniform sampling with the same 64-frame memory budget achieves only 38.7% (696/1800 samples), indicating that retrieval failures are rare in practice.



(a) Counting example in VNBench

(b) Counting example in MLVU

Figure 3. QViC-MF failure cases on counting tasks from VNBench and MLVU. For each example, the input question, relevance score plot, and the frames corresponding to context memory entries are shown. (a) VNBench example: the model correctly identifies and stores all four bicycle-containing segments in the context memory, yet it fails to produce the correct count in its final answer. (b) MLVU example: the model accurately detects and stores the single occurrence of the "playing harp" action, but it likewise fails to output the correct count.

I. Discussion and Limitations

Effectiveness of clip-level memory feedback. Clip-level memory feedback is not uniformly beneficial across all question types. Our evaluation includes tasks that require re-localizing sparse events and reasoning about their temporal order (e.g., VNBench and MLVU Action Order), as well as tasks that involve more global or descriptive understanding (e.g., MLVU Topic Reasoning and LVBench Summarization).

We observe that the proposed feedback mechanism yields the largest gains when questions require retrieving sparse, question-relevant events from long videos. In such cases, single-pass compression may fail to preserve the complete event, whereas memory feedback enables the model to revisit and refine relevant visual evidence across clips. In contrast, for global or summary-style questions that rely on distributed visual cues across the video, the benefits of feedback are smaller because answering these questions requires retaining broad contextual information rather than a small set of key events. Future work may explore hierarchical memory architectures that simultaneously maintain local event-level representations and global video-level summaries, enabling the model to better support both sparse event retrieval and holistic video understanding.

Counting tasks. Despite the overall performance improvements achieved by QViC-MF, its effectiveness on counting tasks remains limited. As shown in Tables 3 and 6, QViC-MF does not exhibit noticeable gains on tasks that require counting the number of specific visual events or object occurrences, such as the Action Count (AC) task in

MLVU and the Counting task in VNBench. We note that this limitation is not unique to QViC-MF; counting remains a challenging task even for strong proprietary models such as GPT-4o [1] and Gemini 1.5 Pro [8], as well as for existing open-source LMMs.

We present representative failure cases for the counting tasks in VNBench and MLVU in Figure 3. In these examples, QViC-MF successfully identifies and stores the frames corresponding to the events to be counted, yet the decoder LLM still fails to output the correct count. This suggests that the bottleneck lies not in the visual retrieval or memory mechanism but in the final reasoning stage performed by the language model. As a direction for future work, expanding training data that explicitly targets counting, or incorporating counting-specific mechanisms such as explicit video segmentation or instance tracking, may further enhance the counting ability of LMMs broadly.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report, 2024. 5, 6, 10

[2] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, 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*, pages 24108–24118, 2025. 3, 6

[3] Yudong Han, Qingpei Guo, Liyuan Pan, Liu Liu, Yu Guan, and Ming Yang. DynFocus: Dynamic cooperative network empowers llms with video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 8512–8522, 2025. 5

[4] Bo He, Hengduo Li, Young Kyun Jang, Menglin Jia, Xuefei Cao, Ashish Shah, Abhinav Shrivastava, and Ser-Nam Lim. MA-LMM: Memory-augmented large multimodal model for long-term video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13504–13514, 2024. 5

[5] Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. 1, 8

[6] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025. 6

[7] Yanwei Li, Chengyao Wang, and Jiaya Jia. LLaMA-VID: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024. 5, 6

[8] Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry Lepikhin, Timothy P. Lillicrap, Jean-Baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 5, 6, 10

[9] Xiaoqian Shen, Yunyang Xiong, Changsheng Zhao, Lemeng Wu, Jun Chen, Chenchen Zhu, Zechun Liu, Fanyi Xiao, Balakrishnan Varadarajan, Florian Bordes, Zhuang Liu, Hu Xu, Hyunwoo J. Kim, Bilge Soran, Raghuraman Krishnamoorthi, Mohamed Elhoseiny, and Vikas Chandra. LongVU: Spatiotemporal adaptive compression for long video-language understanding. In *Forty-second International Conference on Machine Learning*, 2025. 5

[10] Yan Shu, Zheng Liu, Peitian Zhang, Minghao Qin, Junjie Zhou, Zhengyang Liang, Tiejun Huang, and Bo Zhao. Video-XL: Extra-long vision language model for hour-scale video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26160–26169, 2025. 5, 6

[11] Enxin Song, Wenhao Chai, Guanhong Wang, Yucheng Zhang, Haoyang Zhou, Feiyang Wu, Haozhe Chi, Xun Guo, Tian Ye, Yanting Zhang, et al. MovieChat: From dense token to sparse memory for long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18221–18232, 2024. 5

[12] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024. 6

[13] Weihan Wang, Zehai He, Wenyi Hong, Yean Cheng, Xiaohan Zhang, Ji Qi, Ming Ding, Xiaotao Gu, Shiyu Huang, Bin Xu, et al. LVBench: An extreme long video understanding benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22958–22967, 2025. 3, 5

[14] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huanget, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3), 2024. 1, 8

[15] Sicheng Yu, Chengkai Jin, Huanyu Wang, Zhenghao Chen, Sheng Jin, Zhongrong Zuo, Xiaolei Xu, Zhenbang Sun, Bingni Zhang, Jiawei Wu, et al. Frame-Voyager: Learning to query frames for video large language models. In *The Thirteenth International Conference on Learning Representations*, pages 24–28, 2025. 5

[16] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, and Xiaojie Jin. Flash-VStream: Efficient real-time understanding for long video streams. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 21059–21069, 2025. 5, 8

[17] Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *Transactions on Machine Learning Research*, 2025. 5

[18] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. LLaVA-Mini: Efficient image and video large multimodal models with one vision token. In *The Thirteenth International Conference on Learning Representations*, 2025. 5

[19] Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun MA, Ziwei Liu, and Chunyuan Li. LLaVA-Video: Video instruction tuning with synthetic data. *Transactions on Machine Learning Research*, 2025. 1, 5, 6, 8

[20] Zijia Zhao, Haoyu Lu, Yuqi Huo, Yifan Du, Tongtian Yue, Longteng Guo, Bingning Wang, weipeng chen, and Jing Liu. Needle in a video haystack: A scalable synthetic evaluator for video MLLMs. In *The Thirteenth International Conference on Learning Representations*, 2025. 3, 6

[21] Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. MLVU: Benchmarking multi-task long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13691–13701, 2025. 3