DB-VIDEO: DETAIL-PRESERVING DUAL-BRANCH COMPRESSION FOR LONG VIDEO UNDERSTANDING

Anonymous authors
Paper under double-blind review

ABSTRACT

Video Large Language Models (VideoLLMs) face significant challenges in long video understanding due to the expensive computational cost of processing massive visual tokens. Token compression is hence desired to preserve the fundamental video semantics while simultaneously retaining the fine-grained details most relevant to the query. In this paper, we propose DB-Video, which introduces the novel detail-preserving dual-branch compression to achieve this objective. Our approach employs a dual-branch mechanism, i.e., a base semantic branch to maintain global context and a query-guided residual branch to capture query-specific high-resolution details. To preserve such fine-grained details, we further propose the reconstruction loss to enforce visual fidelity in the compressed latent space. Benefited by the proposed dual-branch compressor and reconstruction loss, DB-Video achieves a token compression ratio $17\times$ while preserving 98% of the original performance. Experimental results demonstrate that DB-Video outperforms other training-based compression methods and generalizes well across three model architectures, offering a more efficient strategy for long video understanding.

1 INTRODUCTION

Video Large Language Models (VideoLLMs) have achieved remarkable performance across a diverse range of video understanding tasks (Liu et al., 2023; Wang et al., 2022; Comanici et al., 2025; Bai et al., 2025). However, scaling VideoLLMs to effectively process long videos remains a significant challenge. Long videos inherently encapsulate richer semantic cues and extended temporal contexts, wherein a massive number of video tokens introduce substantial computational and memory overhead for the model.

To mitigate the overwhelming context explosion caused by extended video lengths, extensive research (Bolya et al., 2023; Zhang et al., 2025c;a) has explored compressing the visual tokens extracted from vision encoders. Many existing works design training-free methods, utilizing attention scores or token similarities to reduce the number of visual tokens (Chen et al., 2024a; Yang et al., 2025; Huang et al., 2025; Shen et al., 2025; Wen et al., 2025a; Fan et al., 2026). Despite their simplicity, these approaches are prone to information loss. In contrast, training-based compression paradigms introduce dedicated compression modules optimized via end-to-end training, yielding substantially better compression performance (Chen et al., 2024c; Zhang et al., 2025b; Wang et al., 2025b; Ren et al., 2025).

For long video understanding tasks, an optimal compression module should simultaneously preserve the foundational semantics of the video and retain the critical, fine-grained evidence most relevant to the query. Maintaining basic video semantics enables the model to comprehend key characters and overarching plots, thereby establishing essential spatiotemporal correlations. Concurrently, retaining query-aware fine-grained features empowers the model to attend to critical visual evidence during reasoning, thereby significantly enhancing the accuracy and confidence of its predictions.

To achieve these objectives, we propose the novel detail-preserving dual-branch compression, which comprises two distinct branches. The Base Semantic Branch processes a sparse set of tokens from low-resolution inputs, capturing the fundamental semantics of the video. The Query-Guided Residual Branch handles a dense stream of high-resolution visual tokens. Through cross-modal interaction with text tokens followed by downsampling, it generates residual features that are highly relevant to

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

the specific query. The outputs of these two branches are summed to produce the final compressed representation, effectively integrating global video semantics with query-specific local details.

The fine-grained details are hard to be learned through the commonly adopted optimization loss, e.g., vqa loss on the final output. Instead, we introduce a reconstruction objective to enhance the detail-preserving training of the compressor. Our core insight is that if the compressed representation can faithfully reconstruct the original high-resolution visual tokens, the compression process has successfully preserved key details necessary for answering the question. To this end, we jointly optimize the model using both a reconstruction loss and the vqa loss, thereby enhancing the compressor's capability on preserving fine-grained details.

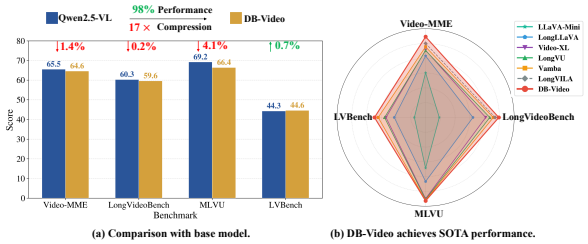


Figure 1: **Performance of DB-Video.** (a) Under a 17× compression ratio, DB-Video preserves the average 98% performance of the base model. (b) DB-Video achieves state-of-the-art performance across four widely adopted long video understanding benchmarks.

Building upon the proposed compressor architecture and optimization objectives, we develop **DB-Video**, a highly efficient model for long video understanding. As shown in Figure 1, across four challenging benchmarks (Wu et al., 2024; Fu et al., 2025; Zhou et al., 2025; Wang et al., 2025a), DB-Video outperforms existing training-based compression approaches (Chen et al., 2024c; Wang et al., 2025b; Shen et al., 2024; Zhang et al., 2025b; Shu et al., 2025; Ren et al., 2025). Compared to the base model, DB-Video retains an average 98% of the original performance while achieving a 17× reduction in visual token count. Moreover, our method can be applied to three different model architectures, consistently delivering favorable compression performance.

In summary, our primary contribution is the novel detail-preserving dual-branch compression, which efficiently compresses visual tokens without losing fundamental video semantics and visual details most relevant to the query. To optimize the model, we introduce the reconstruction training objective to ensure critical details are preserved within the compressed latent space. Extensive evaluations across four benchmarks demonstrate the robustness of this approach, i.e., DB-Video achieves a high compression ratio with minimal performance degradation, significantly outperforming existing training-based compression methods.

2 Related Work

2.1 Long Video Understanding

While multimodal large language models are effective in processing short video clips (Liu et al., 2023; Bai et al., 2025), they still face challenges in long video scenarios. Long-form videos contain richer semantics and more complex event distributions, inevitably leading to significantly longer context sequences. Recent methods enhance long video understanding capabilities by either expanding the context window or designing lightweight modules. LongVA (Zhang et al., 2024) and LongVILA (Chen et al., 2024c) extrapolate the context length of the LLM backbone, enabling the models to natively process a larger number of tokens. mPLUG-Owl3 (Ye et al., 2024) proposes novel hyper attention blocks to efficiently integrate vision and language into a common language-guided semantic space. LLaVA-OneVision (Li et al., 2024) performs task transfer from the image domain, demonstrating robust video reasoning capabilities. Video-XL (Shu et al., 2025) introduces visual summarization tokens, leveraging the inherent key-value sparsification capacity of LLMs to condense visual input. VideoChat-Flash (Li et al., 2025) proposes a hierarchical video token compression method to compress long video context from the clip level to the video level.

In contrast to prior works, our proposed DB-Video introduces a novel dual-branch compression architecture, preserving fundamental video semantics while simultaneously retaining the visual evidence most relevant to the query. This provides a novel compressed representation for long video understanding.

2.2 VIDEO TOKEN COMPRESSION

Video token compression approaches broadly fall into two categories: training-free and training-based methods. Training-free methods typically establish specific metrics to determine token importance. For example, FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025), and FlexSelect (Zhang et al., 2025e) utilize attention scores as the basis for token pruning, while DART (Wen et al., 2025a) employs token similarity as its criterion. While these methods offer a plug-and-play capability without the need for additional training, they are prone to information loss. Conversely, training-based methods generally achieve superior compression performance. LLaVA-Mini (Zhang et al., 2025b) and MQT-LLaVA (Hu et al., 2024) utilize query-based compression modules to achieve compression through a set of learnable tokens. Rather than explicitly reducing the number of tokens, Vamba (Ren et al., 2025) designs efficient modules to approximate the causal self-attention operation in transformer-based MLLMs, thereby accelerating video processing. Xcomp (Zhang et al., 2026) prunes visual tokens layer-by-layer during the training phase, realizing a progressive compression strategy. EPIC (Wen et al., 2025b) introduces token consistency distillation and layer consistency distillation, reducing the learning difficulty of compression.

Similarly, our method also relies on end-to-end optimization. However, we additionally incorporate a reconstruction loss in the feature space to explicitly preserve query-aware details throughout the compression process.

3 METHOD

3.1 FORMULATION AND OVERVIEW

VideoLLMs typically comprise a visual encoder, a modality projector, and a large language model. Given a question prompt $\mathcal{P}$ and a sampled video frame sequence $\mathcal{F} = \{f_1, f_2, \ldots, f_T\}$, the prompt is tokenized into text tokens $\mathbf{F}_t \in \mathbb{R}^{L\times D}$, where L is the text sequence length and D is the hidden dimension of LLM. Each frame is independently encoded, and the resulting features are concatenated and projected into the text embedding space:

$$\mathbf{F}_v = \text{Proj}\left([\text{VE}(f_t)]_{t=1}^{T}\right) \in \mathbb{R}^{N\times D}, \tag{1}$$

where N is the total number of visual tokens. For long videos, N often exceeds the LLM's context window. To address this, training-based compression methods insert a compression module C between the projector and the LLM:

$$\mathbf{F}_{\text{comp}} = C(\mathbf{F}_v) \in \mathbb{R}^{n\times D}, \tag{2}$$

where $n \ll N$ denotes the number of compressed tokens. This compression substantially reduces computational and memory overhead. The compressed visual tokens and text tokens are then concatenated and fed into the LLM, which autoregressively predicts each output token o_i conditioned on these inputs and the previously generated tokens $\mathbf{o}_{<i}$:

$$o_i = \text{LLM}([\mathbf{F}_{\text{comp}}; \mathbf{F}_t], \mathbf{o}_{<i}). \tag{3}$$

3.2 DUAL-BRANCH COMPRESSOR DESIGN

An effective compression module should preserve the fundamental semantics of the video while retaining fine-grained details most relevant to the textual query. To this end, we design a detail-preserving dual-branch compressor, which comprises a base semantic branch and a query-guided residual branch. In the following, we elaborate on the specific designs and functions of the two branches. An overview of the proposed method is shown in Figure 2.

3.2.1 BASE SEMANTIC BRANCH

For the raw video, we sample a sparse set of frames at low resolution (e.g., 128 frames, 224 $\times$ 224). Following the visual encoder and modality projector, we obtain visual tokens $\mathbf{F}_l \in \mathbb{R}^{N_l\times D}$, where N_l is the number of tokens. This branch efficiently captures the fundamental video semantics with minimal computational overhead, such as characters, environment, and actions. This branch contains no trainable parameters.

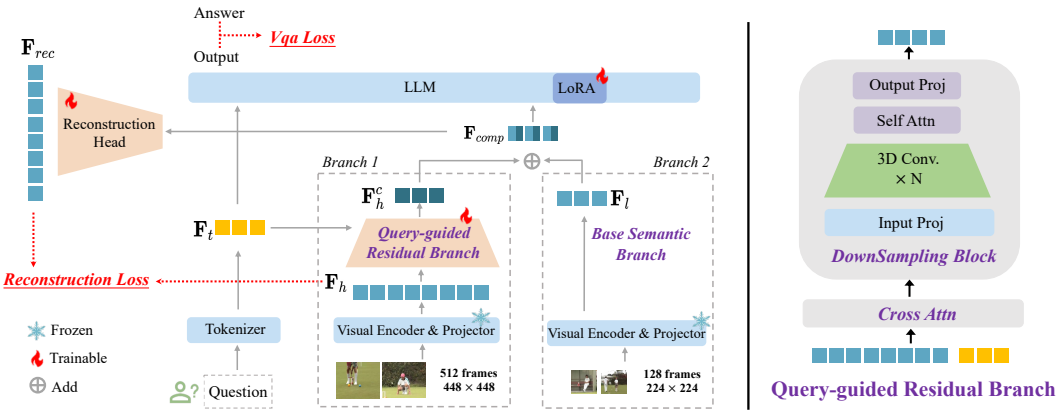


Figure 2: **Overview of the Detail-Preserving Dual-Branch Compressor of DB-Video. (Left) The Dual-Branch Architecture.** The base semantic branch extracts fundamental video semantics from low-resolution inputs, while the query-guided residual branch selectively filters and preserves query-relevant details from high-resolution inputs. The entire compressor is jointly optimized using both a reconstruction loss and a vqa loss. **(Right) The Query-Guided Residual Branch.** This branch consists of two core components: a cross-attention mechanism for multi-modal alignment and a downsampling block for efficient feature compression.

3.2.2 Query-Guided Residual Branch

We sample a dense set of frames at high resolution (e.g., 512 frames, 448 × 448) to obtain $\mathbf{F}_h \in \mathbb{R}^{N_h \times D}$, where $N_h \gg N_l$. $\mathbf{F}_h$ encapsulates rich fine-grained details of the video. The primary objective of this branch is to selectively filter details relevant to the textual query and preserve them during the compression process. To this end, the query-guided residual branch is architected around two primary modules: (1) a cross-attention mechanism that facilitates multi-modal alignment between visual and text tokens; and (2) a downsampling block that performs spatiotemporal compression through a cascade of dimensionality reduction, 3D convolutions, and self-attention refinement.

As illustrated in Figure 2, $\mathbf{F}_h$ first interacts with the text tokens $\mathbf{F}_t$ via cross-attention to generate $\mathbf{F}'_h$. Through this process, tokens relevant to the text are prioritized in the subsequent compression,

$$\mathbf{F}'_h = \mathbf{F}_h + \text{CrossAttn}(\mathbf{F}_h, \mathbf{F}_t). \tag{4}$$

Subsequently, $\mathbf{F}'_h$ is first normalized via layer normalization and then mapped to a low-dimensional space via an input projector to yield $\mathbf{H}_{in}$. This process establishes an information bottleneck for efficient compression:

$$\mathbf{H}_{in} = W_{in} \cdot \text{LN}(\mathbf{F}'_h) \in \mathbb{R}^{N_h \times d}, \tag{5}$$

where $d < D$. In this low-dimensional space, $\mathbf{H}_{\text{in}}$ is downsampled using cascaded temporal and spatial 3D convolutions to align the token counts of the two branches. Self-attention further refines the features into $\mathbf{H}_{\text{out}} \in \mathbb{R}^{N_l \times d}$. Finally, the features are mapped back to the original dimension through layer normalization and an output projector, yielding $\mathbf{F}^c_h \in \mathbb{R}^{N_l \times D}$:

$$\mathbf{H}_{\text{out}} = \text{SelfAttn}(\text{Conv3D}(\mathbf{H}_{\text{in}})), \qquad \mathbf{F}^c_h = W_{\text{out}} \cdot \text{LN}(\mathbf{H}_{\text{out}}). \tag{6}$$

3.2.3 Combination

The feature $\mathbf{F}^c_h$ from the query-guided residual branch encodes the fine-grained details most relevant to the question, while $\mathbf{F}_l$ from the base semantic branch encapsulates the foundational semantic of the video. They are fused via a learnable coefficient α to generate the final compressed representation $\mathbf{F}_{comp}$:

$$\mathbf{F}_{comp} = \mathbf{F}_l + \alpha \cdot \mathbf{F}^c_h. \tag{7}$$

3.3 DETAIL-PRESERVING TRAINING STRATEGY

Vqa Loss. During the training process, other methods employ the cross-entropy loss between the model's output and the ground truth to optimize the compression module and the LLM in an end-to-end manner, which can be denoted as the vqa loss $\mathcal{L}_{\text{vqa}}$:

$$\mathcal{L}_{\text{vqa}} = -\sum_{t=1}^{T} \log P(y_t \mid y_{<t}, \mathbf{F}_{comp}), \tag{8}$$

where y represents the ground truth text sequence.

However, relying solely on the vqa loss in our dual-branch compressor presents inherent limitations. The vqa loss predominantly focuses on high-level semantics, which can lead to the degradation of fine-grained visual details during compression. Consequently, the query-based residual branch may not be optimized effectively.

Reconstruction Loss. To address this issue, we introduce a reconstruction loss to enhance the model's capacity to perceive and preserve critical visual information throughout the compression process. Our core insight is that *if the compressed representation can faithfully reconstruct the original high-resolution features, then the compression has successfully preserved the visual information needed for answering the question.*

Distinct from pixel-level generation tasks that demand heavy models such as diffusion (Peebles & Xie, 2023), we focus on reconstructing visual tokens from the high-resolution input in the latent space of LLM. To this end, we design a lightweight reconstruction head that serves as a mirrored counterpart to the query-guided residual branch. Specifically, we exclude the cross-attention layer and replace the original 3D convolution layers with 3D transposed convolutions to perform upsampling.

The compressed representation $\mathbf{F}_{comp}$ is passed through the reconstruction head to yield the reconstructed features $\mathbf{F}_{rec} \in \mathbb{R}^{N_h \times D}$,

$$\mathbf{F}_{rec} = \text{RecHead}(\mathbf{F}_{comp}). \tag{9}$$

We then employ the MSE loss to minimize the discrepancy between $\mathbf{F}_{rec}$ and the original high-resolution visual tokens $\mathbf{F}_{\text{h}}$.

$$\mathcal{L}_{rec} = \frac{1}{N_h} \|\mathbf{F}_{\text{rec}} - \mathbf{F}_h\|_2^2. \tag{10}$$

Loss Combination with Gradient Isolation. During the training process, the visual encoder is frozen, while the compression module and the reconstruction head are trainable. Meanwhile, we fine-tune the LLM utilizing LoRA (Hu et al., 2022). The overall loss is formulated as:

$$\mathcal{L} = \mathcal{L}_{\text{vqa}} + \lambda \cdot \mathcal{L}_{\text{rec}}, \tag{11}$$

where λ is the coefficient that balances the two losses.

While these two losses are fundamentally complementary, we observe a potential optimization conflict at the cross attention layer within the query-guided residual branch during practical implementation. The primary objective of the cross attention layer is to filter and select the most text-relevant visual tokens; however, this intrinsically conflicts with $\mathcal{L}_{\text{rec}}$ which penalizes the loss of any visual details to achieve full reconstruction.

To resolve this tension, we apply a gradient isolation operation to prevent the gradients of $\mathcal{L}_{\text{rec}}$ from backpropagating into the cross attention layer. Consequently, the cross attention layer is strictly optimized by $\mathcal{L}_{\text{vqa}}$, allowing it to focus on semantic selection. The subsequent downsampling modules are jointly optimized by both losses, while the reconstruction head is updated exclusively by $\mathcal{L}_{\text{rec}}$. This simple operation ensures both training stability and effective compression. We provide the necessity of this operation in the subsequent ablation studies.

Table 1: **Performance comparison on four long video understanding benchmarks.** We follow the evaluation setup of Vamba (Ren et al., 2025) and cite their partial results. Underlines denote that Qwen2.5-VL is utilized as the base model. DB-Video outperforms other training-based compression methods across the four benchmarks.

Models	Size	Video-MME	LongVideoBench	MLVU	LVBench	Avg.
		w/o Sub.	Val	Dev	Test	
Proprietary Models						
GPT-4V (Achiam et al., 2023)	–	59.9	59.1	49.2	–	–
GPT-4o (Hurst et al., 2024)	–	71.9	66.7	64.6	34.7	59.5
Gemini-1.5-Pro (Team et al., 2024)	–	75.0	64.0	62.9	33.1	58.8
Open-source Models						
LLaVA-Video (Zhang et al., 2025d)	7B	64.3	58.2	70.8	44.4	59.4
mPLUG-Owl3 (Ye et al., 2024)	7B	59.3	52.1	63.7	43.5	54.7
LLaVA-OV (Li et al., 2024)	7B	58.3	56.4	62.6	39.9	54.3
Kangaroo (Liu et al., 2024)	8B	56.0	54.8	61.0	39.4	52.8
TimeMarker (Chen et al., 2024b)	8B	57.3	56.3	–	41.3	–
InternVL2.5 (Chen et al., 2024d)	8B	64.2	60.0	68.9	–	–
VideoChat-Flash (Li et al., 2025)	7B	65.3	64.7	74.7	48.2	63.2
Qwen2.5-VL (Bai et al., 2025)	7B	65.5	60.3	69.2	44.3	59.8
Training-based Compression Methods						
LLaVA-Mini (Zhang et al., 2025b)	7B	40.3	19.3	44.3	17.6	30.4
LongLLaVA (Wang et al., 2025b)	9B	51.6	42.1	53.3	31.2	44.6
LongVU (Shen et al., 2024)	7B	55.3	53.5	65.4	37.8	53.0
Video-XL (Shu et al., 2025)	7B	55.5	50.7	64.9	36.8	52.0
Vamba (Ren et al., 2025)	10B	57.8	55.9	65.9	42.1	55.4
LongVILA (Chen et al., 2024c)	7B	60.1	57.1	–	–	–
DB-Video	7B	**64.6**	**59.6**	**66.4**	**44.6**	**58.8**

4 EXPERIMENTS

4.1 EXPERIMENT SETTINGS

Training Dataset. To construct our training set, we draw from the multiple-choice QA data within the LLaVA-Video (Zhang et al., 2025d). We first filter out short videos, specifically those with a duration of less than one minute. From the remaining videos, we randomly sample **25K** instances for fine-tuning, which constitutes approximately one-eighth of the entire LLaVA-Video dataset.

Benchmarks and Compared Methods. Following prior work, we evaluate the performance of our model on four widely adopted long video understanding benchmarks: Video-MME (Fu et al., 2025), LongVideoBench (Wu et al., 2024), LVBench (Wang et al., 2025a), MLVU (Zhou et al., 2025). Details of each benchmark are provided in Appendix A.4. We compare our approach with a series of training-based compression methods. Specifically, we select LLaVA-Mini (Zhang et al., 2025b), LongLLaVA (Wang et al., 2025b), LongVU (Shen et al., 2024), Video-XL (Shu et al., 2025), Vamba (Ren et al., 2025), and LongVILA (Chen et al., 2024c). In terms of backbone, LLaVA-Mini and LongLLaVA are built upon LLaVA (Liu et al., 2023); LongVU, Video-XL, and Vamba are Qwen-based methods (Bai et al., 2023); and LongVILA is based on the VILA model (Liu et al., 2025). Notably, we follow the evaluation setup from Vamba (Ren et al., 2025) to ensure a fair comparison, and the performance metrics for the majority of these baselines are directly cited from the results reported in Vamba. Furthermore, we provide the performance of several closed-source models and open-source powerful VideoLLMs for reference.

Compression Ratio under a Fixed Token Budget. As shown in Appendix A.1 and Figure 4, DB-Video consists of two branches: one processing 512 frames at 448 resolution, and the other processing 128 frames at 224 resolution. To ensure a fair comparison, we construct the baseline with an input of 544 frames at 448 resolution, so that the total number of input tokens is identical for DB-Video and the baseline (557K in both cases). Since our method performs compression on the

Table 2: Comparison with training-free methods. All evaluated methods are configured to retain 6% of the tokens.

Models	Video-MME	LongVideoBench	MLVU	LVBench
Qwen2.5-VL	65.5	60.3	69.2	44.3
FastV	61.8	51.9	59.3	40.8
VisionZip	63.0	58.0	65.2	42.4
FastVID	62.1	**60.2**	65.2	41.5
DB-Video	**64.6**	<u>59.6</u>	**66.4**	**44.6**

vision encoder side, in this paper we define the compression ratio as the proportion of tokens that are fed into the LLM. Our method achieves a 17× compression ratio.

Implementation Details. The training process is conducted on 8 NVIDIA H20 GPUs for a single epoch. We employ the AdamW optimizer with a learning rate of 1e-5. For the base semantic branch, we sample 128 frames at a resolution of 224 × 224. For the query-guided residual branch, we sample 512 frames at a resolution of 448 × 448. In our ablation studies, to accelerate training, we reduce the sampling rates to 32 frames and 128 frames for the two branches, respectively. The coefficient λ for the reconstruction loss is set to 0.05. More implementation details are in the Appendix A.3.

4.2 Main Results

Comparison with Baseline. As shown in Table 1, compared to the base model Qwen2.5-VL, DB-Video reduces visual tokens by 17×. Under this aggressive compression, it successfully preserves 98.6%, 98.8%, and 95.9% of the original performance on Video-MME, LongVideoBench, and MLVU, respectively. We also find the accuracy even improves by 0.3 after compression on LVBench. This validates that DB-Video minimizes information loss while maximizing computational efficiency.

Comparison with Training-Based Compression Methods. DB-Video outperforms both LLaVA-based and Qwen-based compression approaches on four long video understanding benchmarks. It improves upon the Video-MME benchmark by 4.5 points and the LVBench benchmark by 2.5 points. Notably, our 7B model outperforms the larger 10B Vamba under the same evaluation setting.

Comparison with Training-Free Compression Methods. To provide a comprehensive evaluation, we compare our approach against several training-free methods, specifically FastV (Chen et al., 2024a), VisionZip (Yang et al., 2025), and FastVID (Shen et al., 2025). All methods are configured to retain 6% of the tokens, corresponding to a 17× compression ratio. As shown in Table 2, compared to the best training-free methods, DB-Video achieves accuracy improvements of 1.6, 1.2, and 2.2 points on the Video-MME, MLVU, and LVBench benchmarks, respectively.

Evaluation of Fine-Grained Scenarios. To evaluate whether DB-Video preserves fine-grained perceptual capability during compression, we conduct experiments on several fine-grained tasks, including the Counting and Attribute Perception sub-tasks in Video-MME, as well as multi-hop Needle-In-A-Haystack (NIAH) benchmarks designed for fine-grained perception (Li et al., 2025). As shown in Table 3, the compressed model achieves consistent improvements. This indicates that our dual-branch compression architecture, optimized with the reconstruction objective, effectively preserves question-relevant fine-grained details.

Generalization across Model Architectures. We evaluate the generalization of DB-Video across different model architectures. We adapt it to LLaVA-Video (Zhang et al., 2025d) and LLaVA-OneVision (Li et al., 2024), retaining the same 17× compression ratio as shown in Table 4. For LLaVA-Video, the compressed model retains 95% and 97% of the baseline performance on Video-MME and LVBench, respectively. For LLaVA-OneVision, it likewise preserves 97% and 93% of the original performance on the two benchmarks. These results demonstrate that our method can be applied to different model architectures while maintaining favorable performance under a high compression ratio.

Visualization and More Results. We provide qualitative case studies in Appendix A.6, results on short-video benchmarks in Appendix A.2.1.

Table 3: Evaluation results on fine-grained benchmarks.

Models	NIAH	Attribute	Counting
No Compression	46.0	75.6	44.8
DB-Video	**47.5**	**76.1**	**47.0**

Table 4: Generalization across different model architectures.

Backbone	Video-MME	LVBench
LLaVA-Video	64.3 / 61.3 (**95%**)	44.4 / 43.0 (**97%**)
LLaVA-OneVision	58.3 / 56.6 (**97%**)	39.9 / 37.1 (**93%**)

Table 5: Ablation studies of branch structure.

Branch	Video-MME				LVBench
	Short	Medium	Long	Overall	Test
Base Semantic Branch	**72.1**	57.2	49.7	**59.7**	36.5
Query Residual Branch	44.3	45.3	43.6	44.4	29.4
Two Branches	68.9	**57.9**	**51.0**	59.2	**38.8**

4.3 Ablation Studies

Branch Structure. To thoroughly investigate the dual-branch compression module, we decouple its base semantic branch and query-guided residual branch for independent evaluation. As shown in Table 5, employing only the base semantic branch yields strong performance on short videos, even surpassing the dual-branch architecture. This is primarily because short videos contain straightforward semantics that can be comprehended using only a few frames. However, performance declines significantly on longer videos, such as the medium and long subsets, as well as LVBench. Conversely, using the query-guided residual branch alone generally results in lower performance, as the residual compression design focuses mainly on detail extraction, inevitably leading to some information loss. In contrast, our proposed dual-branch structure integrates foundational semantics with query-relevant details, making it more suitable for long video understanding. We also provide a comparison with single-branch compression methods (*e.g.*Q-Former) in Appendix A.2.2.

Reconstruction Loss and Gradient Isolation. Table 6 shows the ablation study on our loss design. A comparison between rows 1 and 3 shows that incorporating a reconstruction loss improves performance across several benchmarks. This demonstrates that the reconstruction loss effectively captures and retains query-relevant details, thereby enhancing the model's overall compression capability. However, a comparison between rows 1 and 2 reveals that without the gradient isolation operation, incorporating the reconstruction loss actually leads to a performance drop. This validates our earlier analysis of a potential gradient conflict between the two objectives within the cross attention layer. The contrast between rows 2 and 3 demonstrates that applying the gradient isolation mechanism allows the reconstruction loss to function as intended. Overall, these experiments indicate that with appropriate gradient management, the reconstruction loss and vqa loss do not conflict; instead, they act synergistically to jointly enhance the model's compression capability.

Effect of Frame Number. As shown in Table 7, DB-Video achieves steady performance improvements across various long video understanding benchmarks as the number of sampled frames increases. Benefiting from its high compression ratio, DB-Video can accommodate significantly more video frames given the same memory budget. This highlights the potential of our method for processing longer video context.

Ablation of Hyperparameter λ. Table 7 illustrates the impact of the reconstruction loss coefficient λ on model performance. Given the distinct gradient propagation paths of the two loss functions, we assign a relatively small value to λ. We observe that the model performance remains stable when λ varies between 0 and 0.1, which demonstrates the robustness of our approach. Conversely, setting λ to a larger value leads to a slight performance degradation.

4.4 Efficiency Analysis

We compare the efficiency of DB-Video against the base model Qwen2.5-VL. This analysis contains a training stage and an inference stage across varying frame counts.

Table 6: Ablation studies of loss design.

$\mathcal{L}_{\text{vqa}}$	$\mathcal{L}_{\text{rec}}$	Grad. Iso.	Video-MME				LVBench
			Short	Medium	Long	Overall	Test
✓			67.9	57.1	50.6	58.5	38.2
✓	✓		67.2	55.8	50.6	57.8	36.8
✓	✓	✓	**68.9**	**57.9**	**51.0**	**59.2**	**38.8**

Table 7: Ablation studies of frame counts and the reconstruction loss hyperparameter λ. Branch 1 denotes the base semantic branch, and Branch 2 represents the query-guided residual branch.

Setting	Value	Video-MME				LVBench
		Short	Medium	Long	Overall	Test
Branch 1 / 2	32 / 128	68.9	57.9	51.0	59.2	38.8
	64 / 256	71.2	60.4	51.7	61.1	38.9
	128 / 512	**74.9**	**65.9**	**53.0**	**64.6**	**44.6**
λ	0.01	69.2	59.7	50.8	59.4	38.7
	0.05	68.9	57.9	51.0	59.2	38.8
	0.5	68.3	57.1	50.6	58.7	38.4

Training Stage. To ensure a fair comparison, all experiments are conducted on 6 NVIDIA H20 GPUs with a batch size of 1 per GPU. We report the average results over 10 training steps with two metrics: total GPU memory and average per-step runtime. As shown in Figure 3(a), DB-Video is more efficient during training. For instance, under a 128-frame setting, DB-Video achieves a 2.5× reduction in single-step runtime compared to the base model. Moreover, DB-Video supports significantly longer video sequences during training, whereas the base model encounters out-of-memory errors when the number of sampled frames exceeds 256. **Inference Stage.** Inference efficiency is evaluated in terms of FLOPs and latency. We conduct all inference experiments on a single NVIDIA H20 GPU and estimate FLOPs using the `calflops` library. As illustrated in Figure 3(b), DB-Video consistently requires fewer FLOPs and achieves lower latency than the base model across different numbers of input frames. For example, under the 128-frame setting, DB-Video reduces FLOPs by 65% and latency by 48%. Notably, the efficiency gain becomes more pronounced as the number of input frames increases.

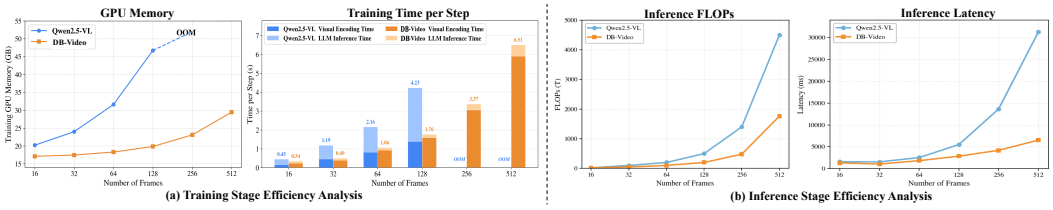


Figure 3: Efficiency analysis. (a) Training stage, measured by GPU memory usage and per-step runtime. (b) Inference stage, measured by FLOPs and latency.

5 Conclusion

We present DB-Video, a highly efficient compression method designed for long video understanding. We construct a novel dual-branch architecture that successfully decouples and preserves both the foundational video semantics and the query-aware, fine-grained visual evidence. To further mitigate information loss and enhance the model's sensitivity to fine-grained details during training, we introduce a reconstruction loss to fulfill the detail-preserving training objective. Extensive experiments on four benchmarks validate that DB-Video outperforms other training-based compression methods, maintaining fine-grained perceptual capabilities and high accuracy even at a high compression ratio. We further discuss the limitation and future work in Appendix A.5.

AI USE STATEMENT

In this work, we used generative AI tools for language editing and manuscript polishing. We did not use these tools to generate research ideas, develop the methodology, design experiments, or execute experimental code. All AI-assisted content was reviewed by the authors, and every language revision was manually checked. We take full responsibility for the final content of this work, including any text, claims, or artifacts produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies efficient long-video understanding using existing datasets and pretrained models. Potential ethical concerns include inherited biases and misuse for privacy-invasive surveillance. We encourage responsible use that respects privacy, consent, and applicable data licenses, together with careful evaluation of model biases and errors before deployment.

REPRODUCIBILITY STATEMENT

We provide detailed implementation information in Appendix A.3 and evaluation benchmark details in Appendix A.4. All datasets used in this work are publicly available, and no private data are used. We will publicly release our code and model checkpoints after they have passed our internal review.

REFERENCES

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. *arXiv preprint arXiv:2303.08774*, 2023.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL https://arxiv.org/abs/2502.13923.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster, 2023. URL https://arxiv.org/abs/2210.09461.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pp. 19–35. Springer, 2024a.

Shimin Chen, Xiaohan Lan, Yitian Yuan, Zequn Jie, and Lin Ma. Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability. *arXiv preprint arXiv:2411.18211*, 2024b.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, Ethan He, Hongxu Yin, Pavlo Molchanov, Jan Kautz, Linxi Fan, Yuke Zhu, Yao Lu, and Song Han. Longvila: Scaling long-context visual language models for long videos, 2024c. URL https://arxiv.org/abs/2408.10188.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024d.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Ziyang Fan, Keyu Chen, Ruilong Xing, Yulin Li, Li Jiang, and Zhuotao Tian. Flashvid: Efficient video large language models via training-free tree-based spatiotemporal token merging, 2026. URL https://arxiv.org/abs/2602.08024.

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 IEEE/CVF conference on computer vision and pattern recognition*, pp. 24108–24118, 2025.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. *Iclr*, 1(2):3, 2022.

Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models, 2024. URL https://arxiv.org/abs/2405.19315.

Xiaohu Huang, Hao Zhou, and Kai Han. Prunevid: Visual token pruning for efficient video large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 19959–19973, 2025.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

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, 2024. URL https://arxiv.org/abs/2408.03326.

Xinhao Li, Yi Wang, Jiashuo Yu, Xiangyu Zeng, Yuhan Zhu, Haian Huang, Jianfei Gao, Kunchang Li, Yinan He, Chenting Wang, Yu Qiao, Yali Wang, and Limin Wang. Videochat-flash: Hierarchical compression for long-context video modeling, 2025. URL https://arxiv.org/abs/2501.00574.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Jiajun Liu, Yibing Wang, Hanghang Ma, Xiaoping Wu, Xiaoqi Ma, Xiaoming Wei, Jianbin Jiao, Enhua Wu, and Jie Hu. Kangaroo: A powerful video-language model supporting long-context video input, 2024. URL https://arxiv.org/abs/2408.15542.

Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvila: Efficient frontier visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4122–4134, 2025.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Weiming Ren, Wentao Ma, Huan Yang, Cong Wei, Ge Zhang, and Wenhu Chen. Vamba: Understanding hour-long videos with hybrid mamba-transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21197–21208, 2025.

Leqi Shen, Guoqiang Gong, Tao He, Yifeng Zhang, Pengzhang Liu, Sicheng Zhao, and Guiguang Ding. Fastvid: Dynamic density pruning for fast video large language models, 2025. URL https://arxiv.org/abs/2503.11187.

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, 2024. URL https://arxiv.org/abs/2410.17434.

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*, pp. 26160–26169, 2025.

Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024.

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*, pp. 22958–22967, 2025a.

Xidong Wang, Dingjie Song, Shunian Chen, Junyin Chen, Zhenyang Cai, Chen Zhang, Lichao Sun, and Benyou Wang. Longllava: Scaling multi-modal llms to 1000 images efficiently via a hybrid architecture, 2025b. URL https://arxiv.org/abs/2409.02889.

Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, et al. Internvideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9972–9991, 2025a.

Zichen Wen, Shaobo Wang, Yufa Zhou, Junyuan Zhang, Qintong Zhang, Yifeng Gao, Zhaorun Chen, Bin Wang, Weijia Li, Conghui He, and Linfeng Zhang. Efficient multi-modal large language models via progressive consistency distillation, 2025b. URL https://arxiv.org/abs/2510.00515.

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. Longvideobench: A benchmark for long-context interleaved video-language understanding, 2024. URL https://arxiv.org/abs/2407.15754.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19792–19802, 2025.

Jiabo Ye, Haiyang Xu, Haowei Liu, Anwen Hu, Ming Yan, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. mplug-owl3: Towards long image-sequence understanding in multi-modal large language models, 2024. URL https://arxiv.org/abs/2408.04840.

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, 2024. URL https://arxiv.org/abs/2406.16852.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20857–20867, 2025a.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token, 2025b. URL https://arxiv.org/abs/2501.03895.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. Sparsevlm: Visual token sparsification for efficient vision-language model inference, 2025c. URL https://arxiv.org/abs/2410.04417.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Llava-video: Video instruction tuning with synthetic data, 2025d. URL https://arxiv.org/abs/2410.02713.

Yunzhu Zhang, Yu Lu, Tianyi Wang, Fengyun Rao, Yi Yang, and Linchao Zhu. Flexselect: Flexible token selection for efficient long video understanding, 2025e. URL https://arxiv.org/abs/2506.00993.

Zheyu Zhang, Ziqi Pang, Shixing Chen, Xiang Hao, Vimal Bhat, and Yu-Xiong Wang. One token per highly selective frame: Towards extreme compression for long video understanding, 2026. URL https://arxiv.org/abs/2604.14149.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, et al. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13691–13701, 2025.

A Appendix

Appendix Overview

The appendix is organized as follows.

A.1 Token Budget Matching and Compression Ratio Details 14
A.2 More Experimental Results 14
A.2.1 Results on Short Video Understanding Benchmarks 14
A.2.2 Comparison with Single-Branch Compression 14
A.2.3 Ablation Studies of Query-Guided Residual Branch 15
A.3 Implementation Details of DB-Video 15
A.4 Benchmark Details 15
A.5 Limitations and Future Work 16
A.6 Case Studies 17

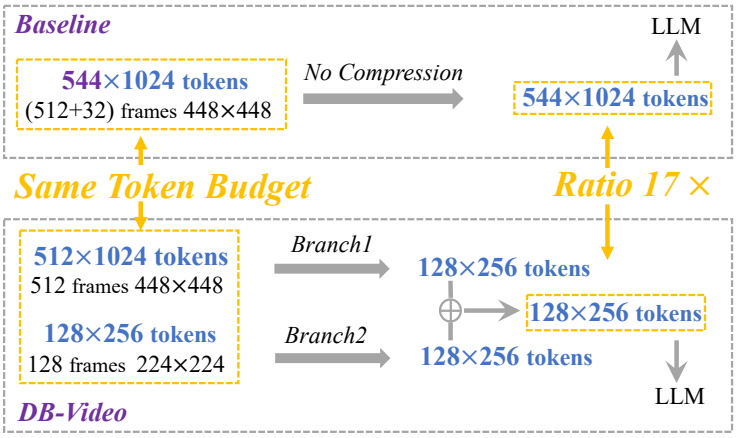


Figure 4: Clarification of the compression ratio under a fixed token budget. We strictly control the number of input tokens to be identical between DB-Video and the baseline, and then define the compression ratio as the proportion of tokens fed into the LLM.

A.1 Token Budget Matching and Compression Ratio Details

As shown in Figure 4, DB-Video processes 512 frames at 448×448 resolution through the query-guided residual branch and 128 frames at 224×224 resolution through the base semantic branch. These resolutions correspond to 1024 and 256 visual tokens per frame, respectively. To match the total visual input token budget, we configure the baseline to process 544 frames at 448×448 resolution:

$$\begin{aligned} N_{\text{in}}^{\text{DB}} &= 512 \times 1024 + 128 \times 256 = 557{,}056, \\ N_{\text{in}}^{\text{base}} &= 544 \times 1024 = 557{,}056. \end{aligned} \tag{12}$$

Thus, both methods receive approximately 557K visual tokens before compression, ensuring a comparison under the same input token budget.

Since compression is applied on the vision-encoder side, we measure its effect using the number of visual tokens actually passed to the LLM. Let $N_{\text{LLM}}^{\text{base}}$ and $N_{\text{LLM}}^{\text{DB}}$ denote these counts for the baseline and DB-Video, respectively. We define the compression ratio as

$$r = \frac{N_{\text{LLM}}^{\text{base}}}{N_{\text{LLM}}^{\text{DB}}} = 17. \tag{13}$$

Equivalently, DB-Video retains approximately 5.88% of the baseline's visual tokens at the LLM input.

A.2 More Experimental Results

A.2.1 Results on Short Video Understanding Benchmarks

While we have demonstrated the superior performance of DB-Video on long video understanding benchmarks in the main text, we further evaluate our model on short video benchmarks to ensure a comprehensive comparison. Specifically, we select two widely adopted benchmarks: MVBench and NExT-QA. We compare DB-Video against five representative training-based compression methods: LLaVA-Mini, LongLLaVA, LongVU, Video-XL, and Vamba. Following the evaluation protocol and the reported results from Vamba, we summarize the comparative performance in Table 8. As the results indicate, DB-Video achieves the highest accuracy on NExT-QA and ranks second on MVBench. This demonstrates that DB-Video maintains highly effective compression across varying video durations, while exhibiting even greater potential and robustness for longer video sequences.

A.2.2 Comparison with Single-Branch Compression

To further validate the effectiveness of our dual-branch design, we compare it with a strong single-branch baseline in this section. Specifically, we adopt a resampler-based single-branch compression

Table 8: Performance comparison on short-video understanding benchmarks. **Bold** and underlined values indicate the best and second-best performance, respectively.

Models	MVBench	NExT-QA
LLaVA-Mini	44.5	47.6
LongLLaVA	54.6	72.2
LongVU	**66.9**	78.0
Video-XL	55.3	77.5
Vamba	60.4	<u>78.1</u>
DB-Video	<u>63.7</u>	**79.7**

Table 9: Performance comparison between the dual-branch architecture and the single-branch resampler-based structure.

Architecture	Video-MME				LVBench
	Short	Medium	Long	Overall	Test
Resampler-based	43.9	46.3	43.3	44.5	33.3
Dual-branch	**68.9**	**57.9**	**51.0**	**59.2**	**38.8**

architecture, a classical design widely used in previous work. Typically, a resampler-based module employs a set of learnable tokens that interact with the input visual tokens to produce a compressed representation. As reported in Table 9, replacing our dual-branch architecture with the single-branch resampler leads to a significant performance degradation, demonstrating the superiority of our dual-branch design over its single-branch counterpart. We attribute this advantage to the decoupled nature of our architecture. The base semantic branch inherently preserves the fundamental semantics of the video, allowing the query-guided residual branch to focus exclusively on capturing query-relevant fine-grained details. Consequently, this functional decoupling enables more effective compression of long-form videos.

A.2.3 Ablation Studies of Query-Guided Residual Branch

The query-guided residual branch comprises two primary components: a cross attention layer and a downsampling block. We conduct an ablation study to evaluate this structural design, as illustrated in Table 10. Removing the cross attention layer leads to an accuracy decline in model performance. This indicates that the cross attention mechanism effectively filters and preserves query-relevant visual details, thereby enhancing the overall compression efficacy.

A.3 Implementation Details of DB-Video

We fine-tune the LLM using LoRA, with hyperparameters set to $r = 64$, $\alpha = 128$ and dropout rate of 0.05. Training is conducted in BF16 mixed precision with DeepSpeed ZeRO-2 optimization to enhance memory efficiency.

For the Downsampling Block within the Query-Guided Residual Branch, the input dimension of the LLM is $D = 3584$, while the compressed latent dimension is set to $d = 512$. We stack four 3D convolutional layers, each followed by SiLU activation. Specifically, the first layer performs temporal downsampling, while the second layer operates on both temporal and spatial dimensions. The third and fourth layers refine the features without altering the spatiotemporal resolution. Finally, we append two layers of self-attention for further feature enhancement. All newly introduced linear layers are initialized with Xavier uniform initialization, and convolutional layers are initialized using Kaiming normal initialization. The residual gate parameter α is initialized to 0.1.

A.4 Benchmark Details

In this section, we provide a detailed description of four widely adopted evaluation benchmarks for long video understanding.

Table 10: Ablation studies of the query-guided residual branch.

Model	Video-MME				LVBench
	Short	Medium	Long	Overall	Test
w/o Cross-Attention	67.9	57.4	50.8	58.7	38.5
w/ Cross-Attention	**68.9**	**57.9**	**51.0**	**59.2**	**38.8**

Video-MME (Fu et al., 2025) is a comprehensive multi-modal benchmark designed for video understanding across varying durations. It comprises 900 videos spanning 6 distinct domains and 30 subcategories, totaling 2700 question-answer pairs. Based on video duration, the dataset is categorized into Short, Medium, and Long subsets, covering content ranging from seconds to hours. The benchmark supports two evaluation protocols: w/o subtitle, which provides only the raw video input, and w/ subtitle, which includes subtitles as supplementary text input. We conduct our evaluation under the w/o subtitle setting.

LongVideoBench (Wu et al., 2024) is designed to evaluate a model's ability to reason over relevant video details within the referred temporal context. It explicitly emphasizes the integration of video retrieval and reasoning capabilities. LongVideoBench comprises 3763 videos and 6678 human-annotated multiple-choice questions spanning 17 fine-grained categories. The video durations range extensively from 8 seconds to 1 hour.

MLVU (Zhou et al., 2025) is a comprehensive benchmark focused on multi-task long video understanding. It features diverse video durations and various video genres, encompassing 3102 multiple-choice questions. The video lengths vary from 3 minutes to 2 hours. This benchmark supports both multiple-choice and open-ended formats. Furthermore, it categorizes tasks into three types: Holistic, Single-Detail, and Multi-Detail, totaling 9 sub-tasks. Following standard evaluation protocols, we select the multiple-choice setting and report the average accuracy across all tasks as evaluation metric.

LVBench (Wang et al., 2025a) is a challenging benchmark specifically tailored for assessing VideoLLMs on ultra-long videos. It consists of 1549 questions, with an average video duration of 4101 seconds. The benchmark covers six diverse tasks: temporal grounding, summarization, reasoning, entity recognition, event understanding, and key information retrieval. Designed to simulate real-world applications, LVBench demands a comprehensive understanding of extended temporal contexts.

A.5 Limitations and Future Work

In this paper, we propose DB-Video, a detail-preserving dual-branch compression framework for efficient long video understanding. Extensive experiments demonstrate that DB-Video largely preserves performance even at high compression ratios, outperforming existing training-based compression methods.

Limitations. Despite these promising results, our current work has certain limitations. First, the dual-branch architecture introduces additional overhead during the sampling phase, as the base semantic branch requires a separate sampling process. Although this cost is negligible compared to the computational burden of processing massive tokens through the deep layers of LLM, it nonetheless adds a slight latency to the visual encoding stage. Second, the design of the downsampling block within the query-guided residual branch is relatively straightforward, relying primarily on 3D convolutions without extensive exploration of more complex architectures.

Future Work. To address these limitations, future work could proceed in two directions. First, we aim to optimize the sampling efficiency of the base semantic branch, potentially by deriving the low-resolution features directly from the high-resolution input of the query-guided residual branch, thereby eliminating the need for a second sampling pass. Second, we plan to investigate more powerful structures for preserving query-relevant information, such as stacked transformer blocks or other advanced attention mechanisms. We hope these discussions provide the community with valuable insights into training-based compression methods, ultimately facilitating the development of more efficient VideoLLMs for long video understanding.

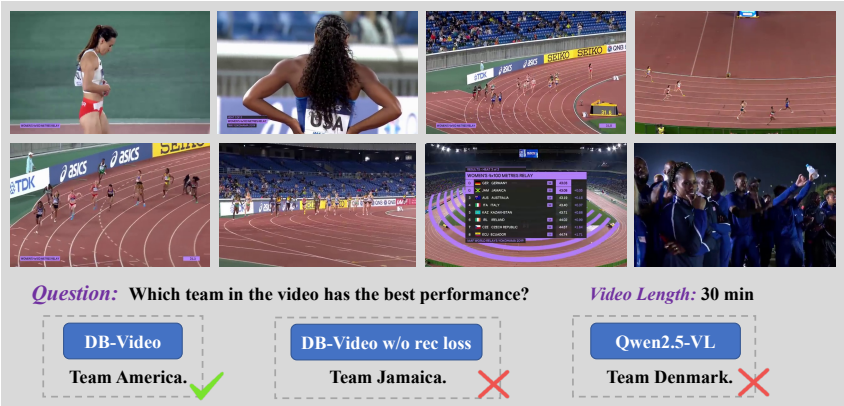


Figure 5: Qualitative comparison on a 30-minute long video sample. DB-Video correctly answers the question, whereas both the model trained without the reconstruction loss and the baseline Qwen2.5-VL produce incorrect responses.

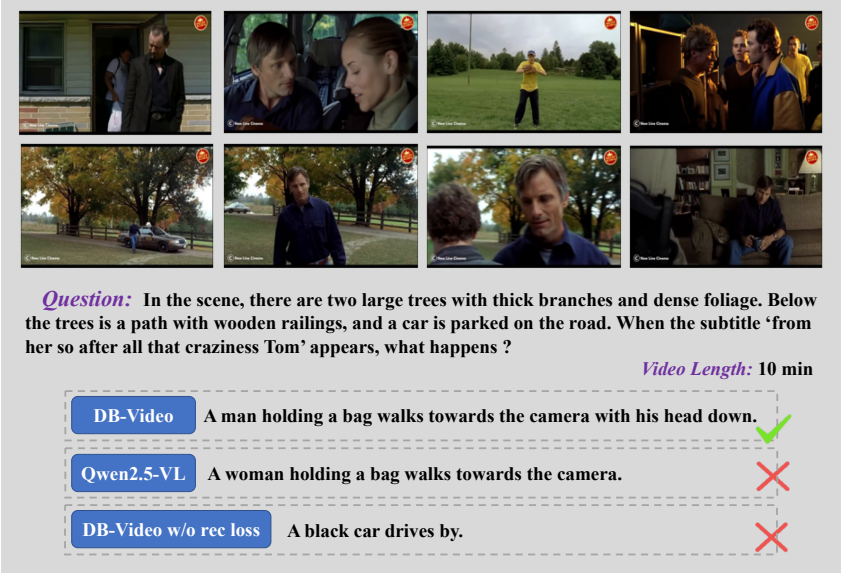


Figure 6: Case study on a long video: third-person movie scene.

A.6 Case Studies

To visually demonstrate the effectiveness of our method, we present a case study in Figure 5. The video input is a 30-minute long video of a women's running race. The question requires the model to not only understand the general event but also specifically locate the final results or the award ceremony under a long temporal context. Both Qwen2.5-VL and the model trained without the reconstruction loss fail to provide the correct answer. In contrast, DB-Video successfully identifies the correct answer. This case demonstrates that our proposed DB-Video effectively integrates fundamental video semantics with query-relevant details, thereby achieving superior performance in long video understanding tasks.

We further provide additional case studies across diverse perspectives and scenarios from LongVideoBench. Figure 6 illustrates a 10-minute movie scene from a third-person perspective. In this instance, DB-Video successfully generates accurate responses. In contrast, both the base model Qwen2.5-VL and the variant trained without reconstruction loss fail to answer correctly.