VidCompress: Memory-Enhanced Temporal Compression for Video Understanding in Large Language Models

Xiaohan Lan, Yitian Yuan, Zequn Jie, Lin Ma

Meituan Inc.

Abstract

Video-based multimodal large language models (Video-LLMs) possess significant potential for video understanding tasks. However, most Video-LLMs treat videos as a sequential set of individual frames, which results in insufficient temporal-spatial interaction that hinders fine-grained comprehension and difficulty in processing longer videos due to limited visual token capacity. To address these challenges, we propose VidCompress, a novel Video-LLM featuring memory-enhanced temporal compression. VidCompress employs a dual-compressor approach: a memory-enhanced compressor captures both short-term and long-term temporal relationships in videos and compresses the visual tokens using a multiscale transformer with a memory-cache mechanism, while a text-perceived compressor generates condensed visual tokens by utilizing Q-Former and integrating temporal contexts into query embeddings with cross attention. Experiments on several VideoQA datasets and comprehensive benchmarks demonstrate that VidCompress efficiently models complex temporal-spatial relations and significantly outperforms existing Video-LLMs.

Introduction

Large Language Models (LLMs) have gained significant attention in the field of artificial intelligence (AI) due to their remarkable capabilities in understanding and generating human language. Models like GPT-3.5 (OpenAI 2023a), GPT-4 (Achiam et al. 2023) and LLaMA-3 (Dubey et al. 2024) demonstrate impressive performance across various natural language tasks like text generation, sentiment analysis, and machine translation (Devlin et al. 2019). Based on the powerful language and knowledge capabilities of LLMs, some studies (Li et al. 2023a; Zhu et al. 2023b) resort to extend text-only understanding by converting the visual input signals, such as images and videos, into tokens that LLMs can understand. Such multimodal LLMs can take more modalities as inputs, significantly broadening the applications of LLMs in AI communities to comprehend the diverse aspects of the physical world.

For multimodal large language models, to comprehend videos takes more challenges than images. The majority of recent Video-LLMs (Li, Wang, and Jia 2023; Lin et al. 2023) do not take the video as a whole but view it as a set of images. In other words, the video is transformed into sequential visual tokens using an image-level encoder followed by an adapter/projector. Temporal reasoning in Large Language Models (LLMs) is exclusively facilitated by the attention mechanisms within the transformer blocks. This way of indirect and late temporal modeling has two inevitable drawbacks: Firstly, changes between consecutive frames in videos convey motion or specific actions, representing short-term correlations. Meanwhile, frames farther apart depict logical relationships within events, indicating long-term associations. Relying solely on the LLM to manage the relationships between visual tokens in the video falls short of achieving such precise modeling, resulting in deficiencies in understanding fine-grained objects or actions, as well as in capturing enduring event connections. Second, the number of visual tokens input into the LLM increases with the length of the video, while the feasible visual token amount fed into the LLM is limited. Therefore, how to use limited visual token capacity to represent the informative video with complex temporal-spatial object relations for Video-LLMs becomes an essential issue to be addressed.

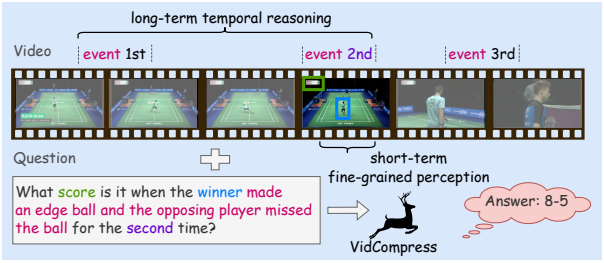


Figure 1: An example of a badminton match video, where temporal reasoning is required to detect the event of the winner scoring a point with mading a second edge ball, and single-frame fine-grained recognition is also needed to identify the specific score. Our proposed VidCompress, with a memory-aware dual-compressor architecture, is capable of performing both long-term and short-term temporal modeling to correctly answer the question.

To better model temporal short-term correlations and long-term associations across frames while keeping efficiency, as illustrated by the example in Figure 1, we propose a novel Video-LLM named VidCompress, which enables memory-enhanced temporal compression in videos. Vid-

Compress employs a dual-compressor architecture, consisting of a memory-enhanced compressor and a text-perceived compressor, to transform input videos into two types of visual tokens.

Specifically, we segment the entire video into fixed-size clips and sequentially feed them into the memory-enhanced compressor to generate memory visual tokens. The memory-enhanced compressor is composed of a multiscale transformer with memory-cache mechanism (Wu et al. 2022). Within a clip, the transformer performs inter-frame interactions by its spatial-temporal attention, aggregating temporal-adjacent information to build the short-term correlations. Also, benefited from the devised memory mechanism, the contextual information from previous clips can be preserved to model the long-term associations. In addition to long-/short-term temporal modeling, fine-grained perception within static frames is crucial for comprehensive video understanding as well. To this end, we introduce a text-perceived compressor to produce perceived visual tokens. First, we leverage Q-Former to compress the frame-wise visual feature, maintaining instruction-relevant visual contents. A cross-attention module is then employed to further integrate temporal contexts, using the memory-enhanced visual tokens from the other compressor as queries to create more condensed text-perceived visual tokens.

Afterwards, both the memory-enhanced tokens and text-perceived tokens of videos are adapted and input into the LLM alongside textual instructions to yield predicted textual tokens. To fully exploit the potential temporal reasoning power of VidCompress, we design a progressive training paradigm, encompassing both modality alignment and instruction tuning stages. We conduct experiments on multiple video question answering (VideoQA) datasets and benchmarks. Both experimental results and qualitative cases demonstrate that VidCompress excels in capturing temporal relationships as the video length increases. To conclude, our contributions can be summarized as follows:

- We introduce VidCompress, a novel Video-LLM that employs a dual-compressor architecture. This architecture integrates a memory-enhanced compressor and a text-perceived compressor to effectively transform input videos into two distinct types of visual tokens, thereby enhancing temporal understanding.
- Our memory-enhanced compressor employs a multiscale transformer with a memory-cache mechanism to capture both short-term and long-term temporal relationships, enhancing video comprehension.
- Experiments show VidCompress has achieved promising results on VideoQA tasks and multiple Video-LLM benchmarks, highlighting the potential of introducing early temporal modeling for Video-LLMs.

Related Work

In this section, we review recent research on LLMs and multimodal LLMs, as well as advancements in long-form video understanding.

Large Language Models By extending the scale of both data and model parameters, we ushered in a brand-new era of large language models. Based on the transformer architecture, a series of language foundation models with billionaire-level parameters such as LLaMA (Touvron et al. 2023), GPT (Achiam et al. 2023) and Claude (Anthropic 2024) have emerged with powerful language reasoning and conversational ability after training on large-scale data. More open-source models such as Alpaca (Taori et al. 2023) and Vicuna (Chiang et al. 2023) leverage the strategy of instruction tuning to further improve the foundation model. Given the powerful tokenized textual understanding and generalization ability, we can build a multimodal large language model via tokenizing the visual signals.

Multimodal Large Language Models Adapting LLMs to interpret visual tokens, multimodal large language models (MLLMs) can process both textual and visual inputs and produce coherent responses. Typically, MLLMs bridge the vision-language gap with lightweight adapters. For example, LLaVA (Liu et al. 2023b,a) and MiniGPT-4 (Zhu et al. 2023b) use a linear layer to project visual features into the language hidden space, while BLIP/BLIP-2 (Li et al. 2022, 2023a) introduces a query transformer (Q-Former) to efficiently extract visual features using learnable query embeddings. Additionally, high-quality image-text instruction pairs (Chen et al. 2023) are created for multimodal pretraining and fine-tuning to align the textual-visual space.

To support video understanding, researchers extend image-based MLLMs for video inputs. For instance, mplug-owl (Ye et al. 2023) processes video inputs similarly to images, which might overlook inter-frame dependencies. To address this, Video-ChatGPT (Maaz et al. 2023) incorporates pooling to extract spatial and temporal features, while VideoChat (Li et al. 2023b) adds a temporal modeling module between the visual encoder and adapter. Other methods use video-based visual encoders, such as VideoLLaVA (Lin et al. 2023) which initializes the multimodal encoder from LanguageBind (Zhu et al. 2023a), and VideoChat2, which employs UMT (Li et al. 2023d) for feature extraction. However, as the video length increases, the number of visual tokens also grows, leading to a longer context for LLMs to process with reduced efficiency. Some approaches mitigate this by downsampling frames, fixing video token lengths, or reducing tokens per frame. For instance, MovieChat (Song et al. 2023) employs a long-short memory mechanism for global or breakpoint understanding, while LLaMA-VID (Li, Wang, and Jia 2023) reduces visual tokens by representing each frame with two tokens.

Long-form Video Understanding Long-form video understanding is a traditional yet challenging problem in visual perception due to the high computational cost and complexity of modeling temporal relationships among lengthy video frames. Some approaches (Li et al. 2017; Miech, Laptev, and Sivic 2017) pre-compute visual features with a frozen backbone to reduce training overhead, while others use sparse sampling, which leads to information loss. Alternative methods (Wang et al. 2016; Christoph and Pinz 2016) use cache/memory mechanisms to handle long video

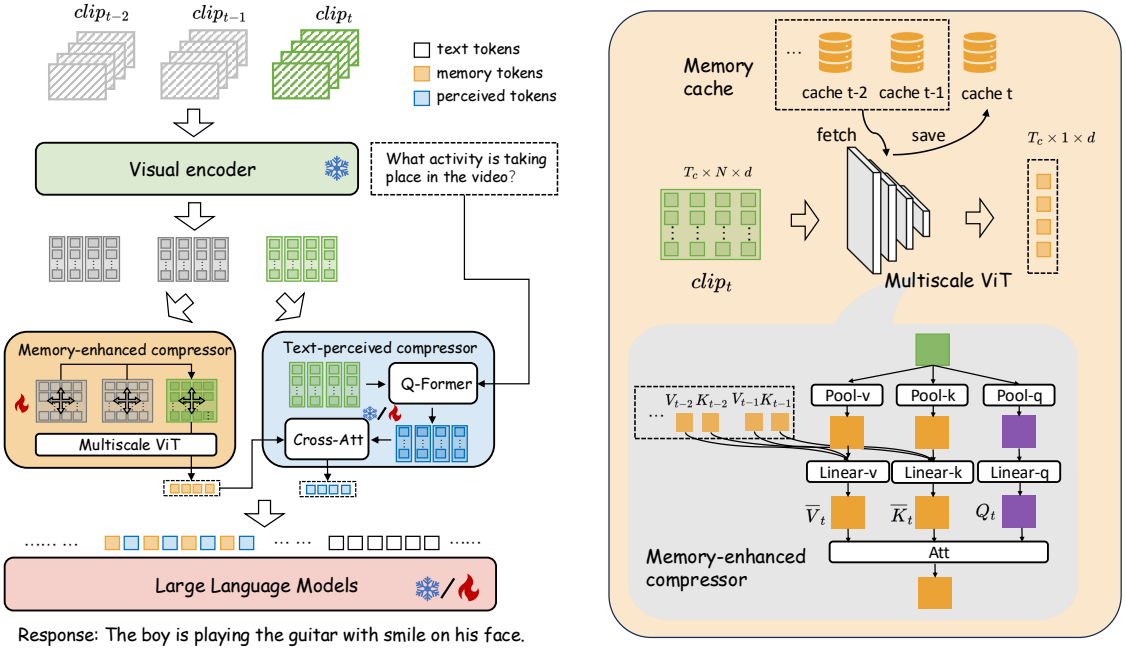


Figure 2: The overall framework of our proposed VidCompress, following a dual-compressor architecture. The visual encoder extracts frame-level features that are fed into the memory-enhanced compressor and text-perceived compressor to generate two types of visual tokens. The right part details the memory-enhanced compressor with devised memory-cache strategy.

sequences. Wu *et al*. 2019 introduce a long-term feature bank for detailed video understanding, and MeMViT (Wu et al. 2022) adds an augmented memory module to the transformer-based MViT (Fan et al. 2021) for more efficient video length scaling. Inspired by MeMViT, our approach also employs a memory-cached strategy within transformer blocks to integrate informative semantics from previous frames to current ones.

Method

This section offers a comprehensive description of our proposed VidCompress, detailing each module and the associated training strategy.

Overview

As illustrated in Figure 2, the framework consists of serveral key components, *i.e*., visual encoder, memory-enhanced compressor, text-perceived compressor and token adaption process for the LLM.

Visual Encoder

The visual encoder is responsible for extracting visual features from the input video frames. It processes a sequence of T frames, utilizing a vision transformer (ViT) to transform each frame into several visual tokens/patches. Therefore, the encoded video feature is formulated as $\mathbf{F}_v = \{\mathbf{v}_1, \mathbf{v}_2, ..., \mathbf{v}_T\}, \mathbf{v}_i \in \mathbb{R}^{N \times d}$, where N is the number of tokens/patches within each frame.

Memory-enhanced Compressor

Inspired by recent advances of video foundation models, we design the memory-enhanced compressor following the MeMViT (Wu et al. 2022) architecture. MeMViT, or Memory-Augmented Multiscale Vision Transformer, is a state-of-the-art model designed for efficient long-term video recognition. Building upon the Multiscale Vision Transformers (MViT) framework, MeMViT introduces memory augmentation to enhance its ability to capture and retain long-term dependencies in video sequences, which is essential for understanding activities that unfold over extended periods.

After obtaining $\mathbf{F}_v \in \mathbb{R}^{T \times N \times d}$ from the visual encoder, we further split it by clips with a fixed size. For the t-th video clip, its clip feature can be represented as $\mathbf{C}_t \in \mathbb{R}^{T_c \times N \times d}$, where T_c denotes the clip size. Then, we sequentially feed these clip features into the memory-enhanced compressor to generate memory-enhanced visual tokens. As shown in the right part of Figure 2, the t-th video clip feature $\mathbf{C}_t$ is fed into a four-layer multiscale transformer blocks, and each block performs spatial downsampling through a pooling attention operation with memory cache interaction. For one transformer block, we first take its input $\mathbf{X}_t$ (the first block's input is $\mathbf{C}_t$) to a pooling layer:

$$\mathbf{Q}_t = \mathcal{P}(\mathbf{W}_Q \mathbf{X}_t)\,, \tag{1}$$

$$\mathbf{K}_t = \mathcal{P}(\mathbf{W}_K \mathbf{X}_t)\,, \tag{2}$$

$$\mathbf{V}_t = \mathcal{P}(\mathbf{W}_V \mathbf{X}_t)\,. \tag{3}$$

Here, $\mathcal{P}$ performs a 3D convolution operation with 3D stride (s_T, s_H, s_W), kernel (k_T, k_H, k_W), and padding

(p_T, p_H, p_W). By setting appropriate stride, kernel and padding numbers, we could downsample the input's spatial dimension while keeping the temporal dimension. In this paper, each transformer block will downsample the input's spatial dimension by 4, and therefore after four blocks' computation, the $N = 256$ patch-level visual tokens of a single frame will be compressed into one unified token.

After the pooling layer, we further extend $\mathbf{K}_t$ and $\mathbf{V}_t$ with cached memory of the K and V values from previous clips and perform attention with these augmented keys and values:

$$\overline{\mathbf{K}_t} = \text{Concat}\left[\mathbf{K}^{\text{cache}}, \mathbf{K}_t\right] , \tag{4}$$

$$\overline{\mathbf{V}_t} = \text{Concat}\left[\mathbf{V}^{\text{cache}}, \mathbf{V}_t\right] , \tag{5}$$

$$\begin{aligned}\mathbf{Z}_t = \text{Attn}(&f_{\text{linear}_Q}(\mathbf{Q}_t) \\ &f_{\text{linear}_K}(\overline{\mathbf{K}_t}), f_{\text{linear}_V}(\overline{\mathbf{V}_t})) .\end{aligned} \tag{6}$$

Here $\mathbf{K}^{\text{cache}}$ and $\mathbf{V}^{\text{cache}}$ are the key and value vectors of previous M video clips, which is pre-stored in the memory cache. Thus, when we conduct attention operation, each video clip could interact with previous clips, and thus the long-range temporal context can be retained in this procedure. Notably, the spatial and temporal dimensions of all the Q/K/V values are flattened, thus such spatial-temporal attention also facilitates inter-frame fine-grained interactions. Finally, the output $\mathbf{Z}_t$ will be taken as the input to the next transformer block, and get further token compression.

To this end, both the long-term and short-term correlation could be established in this memory-enhanced attention mechanism, which is crucial for understanding activities spread over long video sequences. Additionally, since our video clips can be processed sequentially, each video clip only needs to access the previous M clips when computing attention. Therefore, the memory mechanism can be implemented in a first-in-first-out queue manner.

After the memory-enhanced compressor, the input video feature $\mathbf{F}_v \in \mathbb{R}^{T\times N\times d}$ are compressed in the spatial dimension, and thus yielding the memory-enhanced visual tokens $\mathbf{F}_m \in \mathbb{R}^{T\times 1\times d}$.

Text-perceived Compressor

In addition to long-term and short-term temporal modeling, the text-perceived compressor concentrates more on intra-frame interactions under the text/question guidance. It generates text-perceived visual tokens by utilizing Q-Former and a devised cross-attention module in a two-step compression manner.

At the first step, by employing the text-aware Q-Former similar to InstructBLIP (Dai et al. 2023), the visual tokens of each video frame (*e.g.*, $\mathbf{v}_i$) are compressed into N_q query tokens individually, thus maintaining text-relevant visual contents. For the i-th frame, we define the output of text-aware Q-Former as $\mathbf{q}_i \in \mathbb{R}^{N_q\times d}$. At the second step, to incorporate temporal context information into text-aware visual tokens $\mathbf{q}_i$, we employ a cross-attention module that aggregates both temporal and textual context and further compress $\mathbf{q}_i$ into one compact token. More specifically, for i-th frame, we take its corresponding memory-enhanced visual tokens $\mathbf{F}_m[i] \in \mathbb{R}^{1\times d}$ from the memory-enhanced compressor as the query to attend $\mathbf{q}_i$ and obtain its final compressed text-perceived visual tokens $\mathbf{F}_p[i]$:

$$\begin{aligned}\mathbf{F}_p[i] &= \text{CrossAttn}(\mathbf{F}_m[i], \mathbf{q}_i, \mathbf{q}_i) \\ &= \text{Softmax}(\mathbf{F}_m[i] \cdot \mathbf{q}_i^T/\sqrt{d}) \cdot \mathbf{q}_i .\end{aligned} \tag{7}$$

By aggregating all frames' text-perceived visual tokens, we thus could get $\mathbf{F}_p \in \mathbb{R}^{T\times 1\times d}$.

Settings	Training Phase	
	Modality Alignment	Instruction Tuning
Batch Size	256	128
Epoch	1	
Learning Rate	1e-3	2e-5
Learning Schedule	Cosine Decay	
Warmup Ratio	0.03	
Weight Decay	0	
Optimizer	AdamW	
Max Token	2048	
Visual Encoder	Freeze	
Mem-enhanced Comp.	Open	
Q-Former	Freeze	Open
Projectors	Open	
LLM	Freeze	Open
Video FPS	1	

Table 1: Training parameter settings of VidCompress, *Mem-enhanced Comp.* denotes the memory-enhanced compressor.

Token Adaption for LLM

As shown in Figure 2, the outputs of both memory-enhanced compressor $\mathbf{F}_m$, and the outputs of the text-perceived compressor $\mathbf{F}_p$ are then adapted to the language semantic space with two linear projectors. Thus, we get the final adapted memory and perceived visual tokens to represent the input video. Then, all the visual tokens along with the language tokens from the input text are fed into the pretrained Language Foundation Model (LLM) to return a reasonable video-based response.

In summary, our proposed VidCompress effectively incorporates multimodal context information. The use of dual-compressor architecture ensures that the model captures both temporal and language context to comprehend the video contents in an efficient way.

Training Strategy

As shown in Figure 2, we freeze/unfreeze some modules for different training stages. Generally, we divide the whole training procedure into two stages, namely modality alignment and instruction tuning, respectively.

For the modality alignment stage, we follow LLaMA-VID (Li, Wang, and Jia 2023) and use 790K high-quality image-text and video-text pairs to pretrain our VidCompress model. In this stage, the model mainly focuses on the

Model Name	LLM	Res.	MVBench	Video-MME	LVBench	MMBench-Video
GPT-4V (OpenAI 2023b)	-	224	43.70	-	-	1.53
GPT-4o (OpenAI 2024)	-	224	-	-	27.00	1.30
Gemini-1.5-Pro (Reid et al. 2024)	-	224	-	-	33.10	1.44
VideoChat2 (Li et al. 2023c)	Vicuna-7B	224	51.10	39.50	-	0.99
Video-LLaVA (Lin et al. 2023)	Vicuna-7B	224	-	39.90	-	-
TimeChat (Ren et al. 2023)	LLaMA2-7B	224	-	-	22.30	-
PLLaVA (Xu et al. 2024)	Vicuna-7B	224	46.60	-	-	1.03
ShareGPT4Video (Chen et al. 2024)	LLaMA3-8B	224	**51.20**	-	-	1.05
VidCompress (Ours)	Vicuna-7B	224	46.85	**43.00**	**28.68**	**1.14**

Table 2: The comparison of Video-LLMs on different video benchmarks. The metric for Video-MME represents the overall score training without subtitles. "-" denotes the value is not accessible. **Bold** indicates the best among open-source models.

alignment of vision and language semantic space. Therefore, we only unfreeze the memory-augmented compressor, the cross-attention module, and the two projectors involved in the token adaption. Other modules like visual encoder and Q-Former are frozen during this stage.

For the instruction tuning stage, the model should be fully trained for comprehensive video understanding and instruction following. To this end, we further unfreeze the Q-Former and language foundation model besides the unfreezed modules in the previous stage. We build our instruction tuning dataset from two sources. One part includes 763K pure-text/image/video QA pairs collected by LLaMA-VID. To prove the temporal reasoning ability, we also include 230K video QA pairs sampled from VideoChat2 (Li et al. 2023c).

Experiments

In this section, we present the experimental setup and benchmark VidCompress against other leading Video-LLMs. Also, we analyze key components and provide qualitative results.

Implementation Details

We adopt ViT-G/14 from EVA-G (Fang et al. 2023) as the visual encoder and Vicuna-7B (Chiang et al. 2023) as the LLM. Q-Former for the text-perceived compressor are initialized by its pretrained weights (Li et al. 2023a). Our memory-enhanced compressor refers to the structure of MeMViT but does not use the pretrained checkpoints. Instead, we design a customized, lightweight model with a 4-layer transformer trained from scratch. In our setup, the memory-enhanced compressor is connected to the back of the ViT to handle token compression. To keep the model lightweight and suitable for training, we set the video clip size to 8 and the cached memory size M to 3, where the values to choose would be discussed in ablation studies. Furthermore, we configure the stride, kernel and padding of the 3D convolution to (1, 2, 2), (3, 3, 3) and (2, 2, 2), respectively, enabling each transformer layer to perform 4x spatial downsamping while preserving the temporal dimension. More training settings are depicted in Table 1. The whole training procedure costs 72 hours on 8 A100 GPUs.

Results on Video Benchmarks

We compare our VidCompress with other state-of-the-art models on several video benchmarks, including MVBench (Li et al. 2023c), Video-MME (Fu et al. 2024), LVBench (Wang et al. 2024), and MMBench-Video (Fang et al. 2024) (*c.f.*, Table 2). Among these benchmarks, MVBench contains shorter videos, Video-MME includes short, medium, and long videos, while LVBench and MMBench-Video consist of longer videos. For relative fairness, we selected models that utilize 7B/8B LLMs for comparisons. Most of the open-source models also utilize Vicuna-7B as their LLMs and operate at an image resolution of 224, except for TimeChat and ShareGPT4Video, which employ different versions of the LLaMA model.

In summary, VidCompress demonstrates superior performance across multiple video benchmarks, outperforming comparable models such as VideoChat2, Video-LLaVA and PLLaVA. Meanwhile, our VidCompress shows particular strengths on the Video-MME, MMBench-Video and LVBench benchmarks that have longer videos, indicating its robust capability in comprehending complex video scenarios. This demonstrates the superiority of our video token compression mechanism and memory-aware temporal modeling design in analyzing and processing long videos. Specifically, on the Video-MME benchmark that includes minute-/hour-level testing videos, VidCompress achieves around 3% performance higher than VideoChat2 and Video-LLaVA. For another long-video benchmark LVBench, VidCompress outperforms TimeChat with 6.38% and the result is also competitive with closed-source methods like GPT-4o and Gemini-1.5-Pro. Besides, VidCompress achieves a score of 1.14 on MMBench-Video, surpassing VideoChat2, PLLaVA and ShareGPT4Video with significant gains.

On the benchmark with relatively shorter videos, our VidCompress also achieves comparable results with other models. The accuracy of VidCompress on MVBench is 46.85, placing it close to PLLaVA of 46.6 and slightly lower than ShareGPT4Video. The reason for not performing so well is due to that the short video is not able to benefit from our token compression strategy with the devised memory mechanism, so that the temporal reasoning capability is withheld.

Model Name	LLM	Res.	MSVD-QA		MSRVTT-QA		ActivityNet-QA	
			Acc	Score	Acc	Score	Acc	Score
VideoLLaMA (Zhang, Li, and Bing 2023)	DeBERTa-V2	224	51.6	2.5	29.6	1.8	12.4	1.1
LLaMA-Adapter (Zhang et al. 2023)	Vicuna-7B	224	54.9	3.1	43.8	2.7	34.2	2.7
VideoChat (Li et al. 2023b)	LLaMA-7B	224	56.3	2.8	45.0	2.5	26.5	2.2
VideoChat2 (Li et al. 2023c)	Vicuna-7B	224	70.0	3.9	54.1	3.3	**49.1**	3.3
Video-ChatGPT (Maaz et al. 2023)	Vicuna-7B	224	64.9	3.3	49.3	2.8	35.2	2.7
BT-Adapter (Liu et al. 2023c)	Vicuna-7B	-	67.5	3.7	57.0	3.2	45.7	3.2
Video-LLaVA (Lin et al. 2023)	Vicuna-7B	224	**70.7**	3.9	**59.2**	3.5	45.3	3.3
LLaMA-VID (Li, Wang, and Jia 2023)	Vicuna-7B	224	69.7	3.7	57.7	3.2	47.4	3.3
VidCompress (Ours)	Vicuna-7B	224	68.9	3.7	57.7	3.2	48.3	3.3

Table 3: The comparison of Video-LLMs on VideoQA datasets, with metrics of accuracy(%) and average GPT-evaluated scores.

Model	memory token	perceived token	Video-MME				MVBench
			short	medium	long	all	
VidCompress$_{\text{mem}}$	✓		42.5	37.8	33.2	38.3	41.3
VidCompress$_{\text{txt}}$		✓	**48.6**	**41.9**	37.2	**43.1**	45.6
VidCompress$_{\text{full}}$	✓	✓	46.4	**41.9**	**37.8**	43.0	**46.9**

Table 4: Ablation studies on VidCompress branches. We show the results on Video-MME and MVBench.

Results on Video QA datasets

We also compare VidCompress with other state-of-the-art models on several video question-answering (Video-QA) datasets, including MSVD-QA (Chen and Dolan 2011), MSRVTT-QA (Xu et al. 2016), and ActivityNet-QA (Caba Heilbron et al. 2015) (*c.f.*, Table 3). VidCompress consistently performs at a high level across all three datasets, demonstrating its effectiveness in completing video-based QA tasks. More specifically, VidCompress shows strong performance in the MSVD-QA benchmark with an accuracy of 68.9% and a score of 3.7, placing it among the top models, slightly behind Video-LLaVA and LLaMA-VID. In the MSRVTT-QA benchmark, VidCompress achieves an accuracy of 57.7% and a score of 3.2, which is competitive with other leading models like LLaMA-VID and BT-Adapter. On the ActivityNet-QA benchmark, VidCompress achieves an accuracy of 48.3% and a highest score of 3.3, which is comparable to LLaMA-VID and Video-LLaVA, both of which score 3.3 as well. Among these three datasets, ActivityNet-QA contains longest videos with diverse human activities. The superior results on ActivityNet-QA highlight VidCompress's capability to handle complex Video-QA tasks, making it a strong contender in the field.

Ablation Studies on VidCompress Branches

In this section, we conduct ablation studies by evaluating the effectiveness of the two different branches in VidCompress, with the results shown in Table 4. The settings of the ablation models are as follows:

- VidCompress$_{\text{mem}}$: This model retains only the memory-enhanced compressor branch and feeds only the memory-compressed tokens as visual tokens into the LLM.
- VidCompress$_{\text{txt}}$: This model retains only the text-perceived compressor branch, removes the cross-attention module, and directly feeds the 32 query tokens produced by the Q-Former into the LLM after average pooling.
- VidCompress$_{\text{full}}$: Our full model, VidCompress.

VidCompress$_{\text{mem}}$ achieves the lowest results among these three models. This outcome is expected because, unlike the Q-former in the text-perceived compressor, our memory-enhanced compressor is not pretrained and is merely trained from scratch. Consequently, it is more challenging for it to achieve better performance with a limited training dataset. VidCompress$_{\text{txt}}$ is a more general pipeline similar to previous Video-LLMs with Q-Former as the vision-text adapter, which achieves good results. However, the role of the memory single-branch in the overall model should not be underestimated. When we add the memory-enhanced compressor branch to VidCompress$_{\text{txt}}$ and feed both memory tokens and perceived tokens into the LLM, our full model, VidCompress$_{\text{full}}$ shows significant improvement on MVBench and noticeable improvement on long video results in Video-MME. This indicates that the introduction of long-term memory information enhances the performance of Video-LLMs in analyzing video sequences.

Ablation Studies on the Memory-Enhanced Compressor

The video clip size and cached memory size are two crucial factors that affect the memory-enhanced compressor, determining how far it can contact previous video sequences. Therefore, in this section, we conduct ablation studies on the video clip size and cached memory size to analyze their impacts on model performance (*c.f.*, Figure 4).

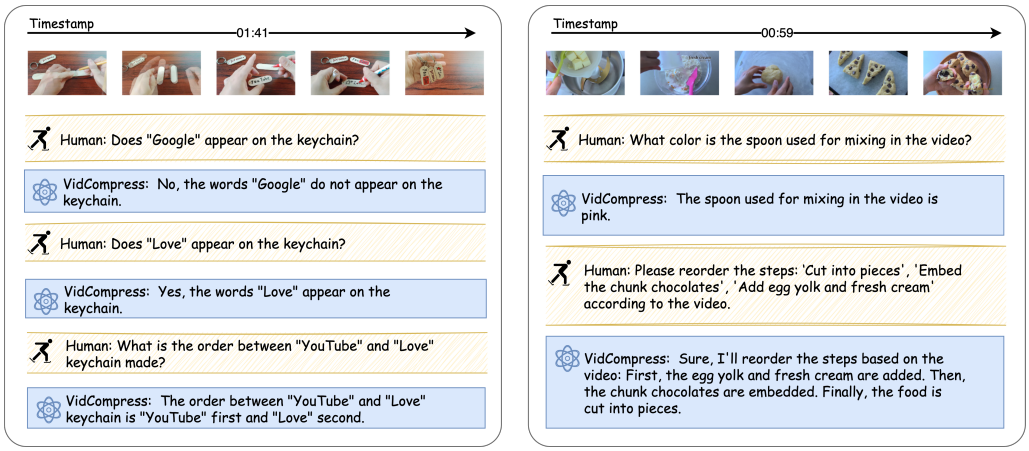
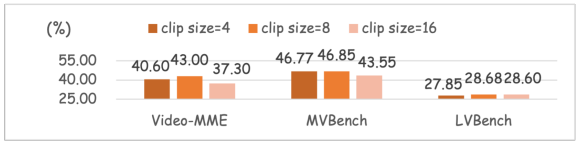


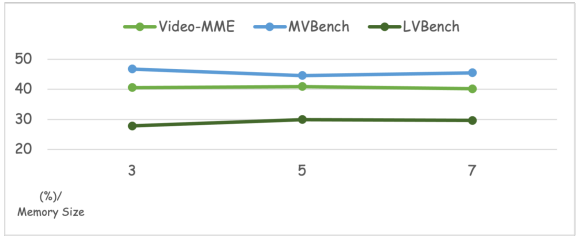
Figure 3: Chat examples of our VidCompress, with a DIY keychain video and a cooking video.

Figure 4a presents the results of choosing different clip sizes (4, 8, and 16) on various video benchmarks, including Video-MME, MVBench, and LVBench. The results indicate a consistent trend regarding the impact of the clip size, *i.e.*, across all three benchmarks, clip size 8 consistently delivers the best performance. While smaller clip size of 4 might lead to faster processing, the clip size of 8 significantly surpasses it on all three benchmarks. However, a larger clip size of 16 results in both performance drop on Video-MME and MVBench, surprisingly. Therefore, considering both efficiency and performance, clip size of 8 offers an optimal solution for VidCompress to capture both short-/long-term temporal relations for better video understanding.

We also investigate the impact of different cached memory sizes (3, 5, and 7) on the model performance, where the clip size is uniformly set to 4. As illustrated in Figure 4b, the optimal size differs among those three benchmarks, *e.g.*, as for MVBench the memory size should be 3 while the highest performance is observed with a memory size of 5 in terms



(a)



(b)

Figure 4: Ablation studies on (a) *clip size* and (b) *cached memory size*.

of Video-MME. Also, the trend looks inconsistent across all three benchmarks. Given that increasing the memory cache size does not lead to significant improvements, it is most efficient to select the smallest memory cache size of 3 for the final solution, which needs less computation resource while maintaining comparable performance.

Qualitative Results

The qualitative results depicted in the Figure 3 showcase the capabilities of our proposed VidCompress model in understanding video content with fine-grained details and temporal relations. VidCompress accurately identifies and distinguishes textual information within videos, as demonstrated by its correct responses to questions regarding the presence of specific words ("Google" and "Love") on a keychain. Additionally, the model effectively discerns visual details, such as identifying the color of a spoon used for mixing in a cooking video. Beyond these basic comprehension tasks, VidCompress also excels in understanding and reasoning about sequences of events, as evidenced by its ability to correctly reorder steps in a recipe according to the instructional video. These results highlight the model's strength in both fine-grained visual recognition and more complex temporal reasoning.

Conclusion

In this paper, we introduced VidCompress, a novel Video-LLM that addresses the challenges of temporal modeling in videos. VidCompress employs a dual-compressor architecture, combining a memory-enhanced compressor and a text-perceived compressor to generate two types of visual tokens. This design enhances both short-term and long-term temporal relationships and ensures fine-grained perception. Experiments on various video benchmarks and video question answering datasets demonstrate the superior performance of VidCompress, highlighting its ability to process and understand lengthy and complex video sequences. We believe VidCompress offers valuable insights for future research in enhancing temporal interactions in Video-LLMs.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic, A. 2024. The claude 3 model family: Opus, sonnet, haiku. *Claude-3 Model Card*, 1.

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the ieee conference on computer vision and pattern recognition*, 961–970.

Chen, D.; and Dolan, W. B. 2011. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 190–200.

Chen, L.; Li, J.; Dong, X.; Zhang, P.; He, C.; Wang, J.; Zhao, F.; and Lin, D. 2023. Sharegpt4v: Improving large multi-modal models with better captions. *arXiv preprint arXiv:2311.12793*.

Chen, L.; Wei, X.; Li, J.; Dong, X.; Zhang, P.; Zang, Y.; Chen, Z.; Duan, H.; Lin, B.; Tang, Z.; et al. 2024. Sharegpt4video: Improving video understanding and generation with better captions. *arXiv preprint arXiv:2406.04325*.

Chiang, W.-L.; Li, Z.; Lin, Z.; Sheng, Y.; Wu, Z.; Zhang, H.; Zheng, L.; Zhuang, S.; Zhuang, Y.; Gonzalez, J. E.; et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3): 6.

Christoph, R.; and Pinz, F. A. 2016. Spatiotemporal residual networks for video action recognition. *Advances in neural information processing systems*, 2: 3468–3476.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P.; and Hoi, S. 2023. InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning. arXiv:2305.06500.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Fan, H.; Xiong, B.; Mangalam, K.; Li, Y.; Yan, Z.; Malik, J.; and Feichtenhofer, C. 2021. Multiscale vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6824–6835.

Fang, X.; Mao, K.; Duan, H.; Zhao, X.; Li, Y.; Lin, D.; and Chen, K. 2024. MMBench-Video: A Long-Form Multi-Shot Benchmark for Holistic Video Understanding. *arXiv preprint arXiv:2406.14515*.

Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19358–19369.

Fu, C.; Dai, Y.; Luo, Y.; Li, L.; Ren, S.; Zhang, R.; Wang, Z.; Zhou, C.; Shen, Y.; Zhang, M.; et al. 2024. Video-MME: The First-Ever Comprehensive Evaluation Benchmark of Multi-modal LLMs in Video Analysis. *arXiv preprint arXiv:2405.21075*.

Li, F.; Gan, C.; Liu, X.; Bian, Y.; Long, X.; Li, Y.; Li, Z.; Zhou, J.; and Wen, S. 2017. Temporal modeling approaches for large-scale youtube-8m video understanding. *arXiv preprint arXiv:1707.04555*.

Li, J.; Li, D.; Savarese, S.; and Hoi, S. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, 19730–19742. PMLR.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Li, K.; He, Y.; Wang, Y.; Li, Y.; Wang, W.; Luo, P.; Wang, Y.; Wang, L.; and Qiao, Y. 2023b. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.

Li, K.; Wang, Y.; He, Y.; Li, Y.; Wang, Y.; Liu, Y.; Wang, Z.; Xu, J.; Chen, G.; Luo, P.; et al. 2023c. Mvbench: A comprehensive multi-modal video understanding benchmark. *arXiv preprint arXiv:2311.17005*.

Li, K.; Wang, Y.; Li, Y.; Wang, Y.; He, Y.; Wang, L.; and Qiao, Y. 2023d. Unmasked teacher: Towards training-efficient video foundation models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 19948–19960.

Li, Y.; Wang, C.; and Jia, J. 2023. LLaMA-VID: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*.

Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2023a. Improved Baselines with Visual Instruction Tuning.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023b. Visual Instruction Tuning.

Liu, R.; Li, C.; Ge, Y.; Shan, Y.; Li, T. H.; and Li, G. 2023c. One for all: Video conversation is feasible without video instruction tuning. *arXiv preprint arXiv:2309.15785*.

Maaz, M.; Rasheed, H.; Khan, S.; and Khan, F. S. 2023. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv preprint arXiv:2306.05424*.

Miech, A.; Laptev, I.; and Sivic, J. 2017. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*.

OpenAI. 2023a. ChatGPT. https://openai.com/blog/chatgpt.

OpenAI. 2023b. GPT-4V. Accessed: 2024-08-07.

OpenAI. 2024. GPT-4o. Accessed: 2024-08-07.

Reid, M.; Savinov, N.; Teplyashin, D.; Lepikhin, D.; Lillicrap, T.; Alayrac, J.-b.; Soricut, R.; Lazaridou, A.; Firat, O.; Schrittwieser, J.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Ren, S.; Yao, L.; Li, S.; Sun, X.; and Hou, L. 2023. TimeChat: A Time-sensitive Multimodal Large Language Model for Long Video Understanding. *arXiv preprint arXiv:2312.02051*.

Song, E.; Chai, W.; Wang, G.; Zhang, Y.; Zhou, H.; Wu, F.; Guo, X.; Ye, T.; Lu, Y.; Hwang, J.-N.; et al. 2023. Moviechat: From dense token to sparse memory for long video understanding. *arXiv preprint arXiv:2307.16449*.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford alpaca: An instruction-following llama model.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Springer.

Wang, W.; He, Z.; Hong, W.; Cheng, Y.; Zhang, X.; Qi, J.; Huang, S.; Xu, B.; Dong, Y.; Ding, M.; et al. 2024. LVBench: An Extreme Long Video Understanding Benchmark. *arXiv preprint arXiv:2406.08035*.

Wu, C.-Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; and Girshick, R. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 284–293.

Wu, C.-Y.; Li, Y.; Mangalam, K.; Fan, H.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2022. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13587–13597.

Xu, J.; Mei, T.; Yao, T.; and Rui, Y. 2016. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5288–5296.

Xu, L.; Zhao, Y.; Zhou, D.; Lin, Z.; Ng, S. K.; and Feng, J. 2024. Pllava: Parameter-free llava extension from images to videos for video dense captioning. *arXiv preprint arXiv:2404.16994*.

Ye, Q.; Xu, H.; Xu, G.; Ye, J.; Yan, M.; Zhou, Y.; Wang, J.; Hu, A.; Shi, P.; Shi, Y.; et al. 2023. mplug-owl: Modularization empowers large language models with multimodality. *arXiv preprint arXiv:2304.14178*.

Zhang, H.; Li, X.; and Bing, L. 2023. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.

Zhang, R.; Han, J.; Liu, C.; Gao, P.; Zhou, A.; Hu, X.; Yan, S.; Lu, P.; Li, H.; and Qiao, Y. 2023. Llama-adapter: Efficient fine-tuning of language models with zero-init attention. *arXiv preprint arXiv:2303.16199*.

Zhu, B.; Lin, B.; Ning, M.; Yan, Y.; Cui, J.; Wang, H.; Pang, Y.; Jiang, W.; Zhang, J.; Li, Z.; et al. 2023a. Languagebind: Extending video-language pretraining to n-modality by language-based semantic alignment. *arXiv preprint arXiv:2310.01852*.

Zhu, D.; Chen, J.; Shen, X.; Li, X.; and Elhoseiny, M. 2023b. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.