Goldfish: Vision-Language Understanding of Arbitrarily Long Videos

Kirolos Ataallah[1], Xiaoqian Shen*[1], Eslam Abdelrahman*[1], Essam Sleiman†[2], Mingchen Zhuge[1], Jian Ding[1], Deyao Zhu[1], Jürgen Schmidhuber[1,3], and Mohamed Elhoseiny[1]

[1] King Abdullah University of Science and Technology
[2] Harvard University
[3] The Swiss AI Lab IDSIA, USI, SUPSI

Abstract. Most current LLM-based models for video understanding can process videos within minutes. However, they struggle with lengthy videos due to challenges such as “noise and redundancy", as well as “memory and computation" constraints. In this paper, we present *Goldfish*, a methodology tailored for comprehending videos of arbitrary lengths. We also introduce the TVQA-long benchmark, specifically designed to evaluate models’ capabilities in understanding long videos with questions in both vision and text content. *Goldfish* approaches these challenges with an efficient retrieval mechanism that initially gathers the top-k video clips relevant to the instruction before proceeding to provide the desired response. This design of the retrieval mechanism enables the Goldfish to efficiently process arbitrarily long video sequences, facilitating its application in contexts such as movies or television series. To facilitate the retrieval process, we developed MiniGPT4-Video that generates detailed descriptions for the video clips. In addressing the scarcity of benchmarks for long video evaluation, we adapted the TVQA short video benchmark for extended content analysis by aggregating questions from entire episodes, thereby shifting the evaluation from partial to full episode comprehension. We attained a 41.78% accuracy rate on the TVQA-long benchmark, surpassing previous methods by 14.94%. Our MiniGPT4-Video also shows exceptional performance in short video comprehension, exceeding existing state-of-the-art methods by 3.23%, 2.03%, 16.5% and 23.59% on the MSVD, MSRVTT, TGIF,and TVQA short video benchmarks, respectively. These results indicate that our models have significant improvements in both long and short-video understanding.Our models and code have been made publicly available Goldfish.

Keywords: Multimodal Learning, LLMs, Long-range Video Understanding, Retrieval Augmented Generation, Applications

[*] Equal contribution

[†] work was done during internship in KAUST

1 Introduction

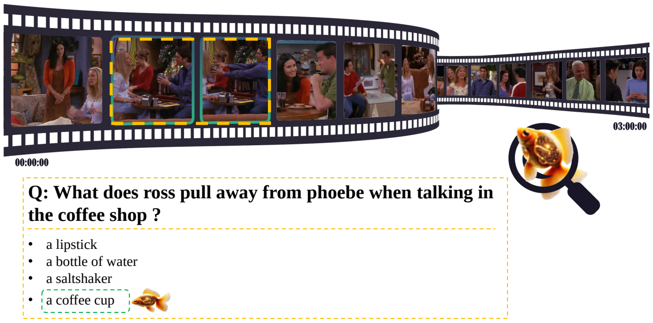

00:00:00
03:00:00
Q: What does ross pull away from phoebe when talking in the coffee shop ?
a lipstick
a bottle of water
a saltshaker
a coffee cup

Fig. 1: GoldFish Model: A long-video model capable of handling lengthy videos by filtering out noisy information and focusing on the most relevant content to accurately answer questions.

The complex and detailed nature of videos provides deep insight, making them crucial for understanding and interacting with the visual world. Recent advances in large vision language models (VLMs) have progressed from image to video-centric multimodal dialogue system [18, 20, 22, 26, 35, 46], enabling these models to process and respond to inputs comprising a video, a user query and, optionally, video subtitles. Despite the progress in adapting VLMs for video, most of the previous works [18, 22, 26, 46] focus on understanding short videos (in minutes) and struggle to deal with long videos. Recent approaches attempted to address this limitation. For example, MovieChat [35]use memory consolidation module and LLaMa-Vid [20] compress image representations into fewer tokens. These strategies improve the capacity to handle larger context windows, enabling these models to process significantly longer videos. However, this compression results in the loss of spatial and temporal visual details and leads to unsatisfactory performance in the understanding of long videos (see Tab. 3). We question: *what factors contribute to the increased difficulty in understanding long videos compared to short videos?* We approach this question by identifying several challenges:

- **Noise and Redundancy**: As demonstrated in the "needle in a haystack" test [6] in the NLP domain, LLMs tend to overlook valuable information within overly extensive contexts. Similarly, long videos often contain irrelevant or redundant information, Making it challenging for the current video-centric LLM to extract meaningful content, especially with a collapsed spatial as well as temporal resolution.
- **Computational and Memory Complexity**: The longer the video, the greater the computational and memory costs required for processing. Current video-centric Large Language Models (LLMs) [22, 26, 46] inherently always have a limitation on the maximum length of videos that they are capable of processing.

- **Lacking Effective Benchmarks for long video understanding**: Existing benchmarks for long videos, such as LLama-Vid [20], primarily generate questions by feeding movie summaries and scripts into a language model, omitting visual data. This approach leads to questions that are text-centric and may be answerable without needing access to the visual content.

To address the challenges of *noise and redudancy* and *computational and memory costs*, we argue that *accurate identification of video clips relevant to queries is a crucial aspect in understanding long videos.* We propose Goldfish, a framework for understanding videos of arbitrary lengths; see Fig. 1. Goldfish addresses these issues by incorporating a retrieval mechanism that selects the top-k relevant video clips before responding to queries. Specifically, Goldfish segments long videos into shorter clips, applies a *Video Descriptor* module to each clip to generate a detailed description of each video clip, and then executes *retrieval module* by comparing the similarities in the text domain between the query text embeddings and the detailed description text embeddings. Following this, the query and corresponding summaries are forwarded to an *answer module* to formulate responses. The Video Descriptor module is actually a short video model (MiniGPT4-Video), which extends MiniGPT-v2 [4]'s architecture to encode not just a single image, but multiple frames with their aligned subtitles. We map the frame tokens through a linear layer to language tokens. Following this, we tokenize the user query and the video subtitles, then introduce both lists of tokens to the LLM, this model not used by zero shot image level but trained on three stages by using video data to enhancing our model's ability to interpret and respond to video content, and this is one of our contribution as we achieved SOTA results for short video benchmarks. In addressing the challenge of *Lacking Effective Benchmarks for long video understanding*, we adapted the TVQA short video benchmark for extended content analysis by aggregating questions from entire episodes, thereby shifting the evaluation from partial to full episode comprehension. We extensively evaluated the proposed Goldfish on previous video benchmarks and our proposed long video benchmark and demonstrated superiority for long video understanding. For example, Goldfish surpasses the competitive concurrent work LLaMA-VID model [20] by about 15% in accuracy. The proposed MiniGPT4-Video also outperforms existing state-of-the-art methods by 3.23%, 2.03%, 16.5% and 6.43% on the MSVD, MSRVTT, TGIF,and TVQA short video benchmarks. Our contributions can be summarized as follows:

- We developed the Goldfish framework for long video understanding, which eased the challenges of long video understanding by introducing a retrieval design. Only top-k relevant video clips are used to answer the questions. *While most previous works can only perform couple of minutes videos, Goldfish can efficiently process arbitrarily long videos.*
- We proposed a new TVQA-long benchmark for long video understanding. Compared to the previous long video benchmarks, TVQA-ong benchmark requires the model to understand both the visual and text content.
- We developed MiniGPT4-Video, which extends VLM to process from single image to multiple frames. By converting frame tokens to language tokens and

incorporating the user's query, we improved the model content understanding by training it for 3 stages by using video data. MiniGPT4-Video can function both as a component for detailed video descriptor within Goldfish and as an independent model for short video tasks.

- Our proposed Goldfish is adept at processing long video understanding, which is verified by achieving SoTA experimental results on 4 long video benchmarks, including LLama-Vid, MovieChat, Movie QA and TVQA with only the vision content and achieved SOTA results with vision and subtitles with zeroshot evaluation on TVQA as TVQA is the only benchmark can be used for zeroshot evaluation because the other models trained on the movies datasets. Apart from the long video understanding, our MiniGPT4-Video also outperformed other methods on 5 short video benchmarks, including Video ChatGPT benchmark, MSVD, MSRVTT, TGIF, and TVQA.

2 Related Work

2.1 LLM-Based Short Video Understanding

Recently, vision-language models such as Video-LLaMA [46] and VideoChat [18] extend the BLIP-2 [17] architecture for video embedding extraction and both employ two streams for audio and visual signals. Video-LLaMA employs a Video Q-Former and an Audio Q-Former for the two streams, while VideoChat has a video embedder and a perception toolkit for captioning, tags, etc. On the other hand, Video-ChatGPT [26] leverages a single stream where the architecture first encodes each frame and then has a spatial and temporal pooling process that is finally mapped to an LLM with a linear layer. Video LLaVA [22] takes advantage of the LanguageBind module to map both image and video inputs to the same embedding space.

2.2 LLM-Based Long Video Understanding

Understanding long videos, such as movies or TV series that exceed two hours in duration, poses significant challenges (as we discussed in Sec. 1) for current video-centric multimodal dialogue systems [22, 26, 46]. Recent MovieChat [35] attempts to address this problem with a memory module containing both *long-term* and *short-term* memory. Short-term memory consists of dense frame-wise encodings that are managed in a FIFO (First In, First Out) queue. When short-term memory is full, the contents are sent to a memory consolidation module, which combines adjacent embeddings by merging similar ones, and then stores them in long-term memory. However, the memory mechanism of this work struggles to capture meaningful information relevant to specific tasks. A concurrent work LLaMA-VID [20] builds a more efficient method by representing each frame with only two tokens, namely context token and content token. These two methods compress the input frame embeddings, increasing the number of frames fitting into the model context window.

Both MovieChat [35] and LLaMA-VID [20] have addressed the *computation and memory* challenge to an extent by compressing visual features. However, their approach of using features from the entire video to predict answers has led them to face issues with *noise and redundancy challenge.* In Goldfish, we introduce a retrieval-based framework that utilizes only the top-k relevant video clips for question answering. This retrieval approach mitigates both challenges and enables efficient processing of long videos.

2.3 Retrieval Systems

LLMs have recently shown promising capabilities in a wide range of different tasks, however, face challenges such as hallucination, when a model outputs a nonsensical or incorrect output typically on queries that extend outside of its training data. Retrieval-Augmented Generation (RAG) is a technique where an LLM leverages an external knowledge base through a retrieval mechanism, mitigating hallucinations while storing long context. There are multiple RAG variations introduced for language retrieval [7,11,12,16,19,19,21,30,31,39,40,42,47,50] and recently have been translated for image retrieval as well [5,10,23]. Most recently there has also been some work in video retrieval [13,41], however, none of these methods can do robust, long-video retrieval. We draw inspiration from these works and develop a retrieval system in the domain of video-centric LLM for long-video retrieval.

3 Goldfish

3.1 Retrieval-based Long Video Understanding

To understand long videos that exceed the context of a normal video large language model, we introduce a three-part system: (1) *Video Descriptor* empowered by a MiniGPT4-Video model and a text encoder, (2) similarity-based *Retrieval Module*, and (3) *Answer Module.* An overview of our system is demonstrated in Fig. 2. Our system works as follows. Firstly, in our Video Descriptor,we break the long video down into smaller clips, with each clip limited by a maximum number of frames that can be supported by our MiniGPT4-Video context length (4K). Then, MiniGPT4-Video provides a concise detailed summary for each clip, which is further encoded to an embedding by a text encoder. Given a user query encoded to an embedding by the same text encoder, our *Retrieval Module* retrieves the most related k clips from the long video and sends them to the *Answer Module* to formulate an answer to the query.

Video Descriptor. Our Video Descriptor breaks down lengthy videos into multiple non-overlapped short clips, each accompanied by textual descriptions and corresponding embeddings for the Retrieval Module. The input for the Video Descriptor is a sequence of frames, denoted as $V = \{v_1, v_2, \dots, v_T\}$, where $v_i \in \mathbb{R}^{3\times H\times W}$ represents the i-th frame, and T is the sequence's length. These frames are then grouped into m chunks, with each chunk represented

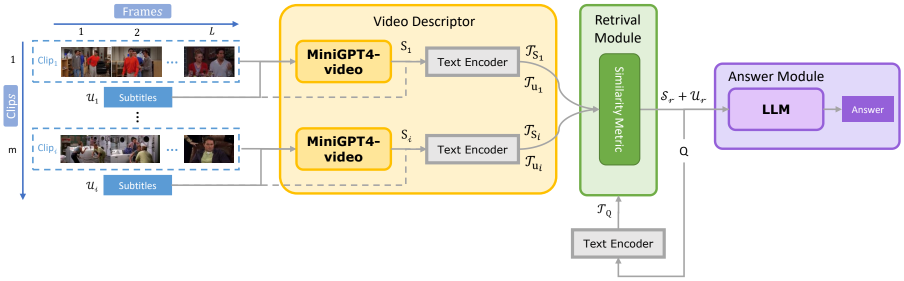


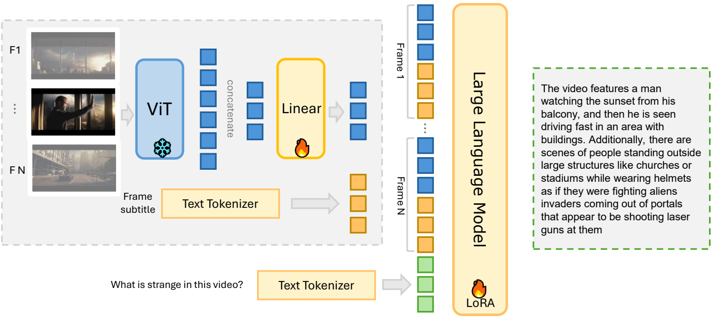
Fig. 2: Goldfish framework,First break down the long video into clips, then encode them in Video Descriptor according to their timing and corresponding subtitles, then encode the use query and retrieve the most related clips in the retrieval module, and finally send the top-K clips information to the answer module to get the final answer.

as $C_k, k \in [1, m]$, comprising at most L consecutive frames $v_{k,j}$ from the video V, where $v_{k,j}$ signifies the j-th frame within the k-th chunk. Here, L is determined by the maximum number of frames that can be accommodated within the context window of our MiniGPT4-Video introduced later. Consequently, the video can be represented as a sequence of clips: $V = \{C_1, C_2, \ldots, C_m\} = \{(v_{1,1}, ..., v_{1,L}), (v_{2,1}, ..., v_{2,L}), ..., (v_{m,1}, ..., v_{m,L})\}$.

We employ our short video model (MiniGPT4-Video) to handle the processing and generation of descriptions for each video clip. Drawing from existing LLM-based vision-language models [4,17,24], we adapt this framework to the video domain, resulting in our MiniGPT4-Video model. The architecture of this model is illustrated in Fig. 3. For the video encoding stage, we utilize EVA-CLIP [36], integrating a projection layer to map visual features from the vision space to the text space of the LLM. To optimize the contextual capabilities of the LLM, we condense every four adjacent visual tokens into a single token, effectively reducing the token count per image by 75%, from 256 to 64 similar as [4]. Through training, the LLM learns to process these video frame features, generating comprehensive clip descriptions S_1, S_2, S_m for each clip essential for conducting visual question-answering tasks in the vision-language domain.

After generating descriptions for the video clips, we proceed to encode them along with their respective subtitles using a text encoder. The set of encoded descriptions is defined as: $\{T_{s_1}, T_{s_2}, ..., T_{s_m}\}$, and the encoded corresponding subtitles $\{u_1, u_2, ...u_m\}$ are defined as $\{T_{u_1}, T_{u_2}, ..., T_{u_m}\}$, where $T_{u_i}, T_{s_i} \in \mathbb{R}^d, i \in [1, m]$, and d is the dimensionality of text encoder space. Specifically, we employ OpenAI's `text-embedding-3-small` [28] model as our chosen text encoder based on table 2 in section 4.4 .

Retrieval Module. The Retrieval Module plays a crucial role in identifying video clips most pertinent to a user query, leveraging the pre-processed clip embeddings from the Video Descriptor. Upon receiving a user query Q, we initially encode it using the text encoder, resulting in the embedding $T_Q \in \mathbb{R}^d$. Subsequently, we compute its cosine similarities with each candidate key K_i from the embeddings set of the clip descriptions and subtitles with $K_i \in \{T_{u_1}, T_{u_2}, ..., T_{u_m}, T_{s_1}, T_{s_2}, ..., T_{s_m}\}$ via $\frac{\mathbf{K_i} \cdot \mathbf{T_Q}}{|\mathbf{K_i}||\mathbf{T_Q}|}$. Next, we select the Top-k similarity scores and retrieve the corre-



The video features a man watching the sunset from his balcony, and then he is seen driving fast in an area with buildings. Additionally, there are scenes of people standing outside large structures like churches or stadiums while wearing helmets as if they were fighting aliens invaders coming out of portals that appear to be shooting laser guns at them

Fig. 3: MiniGPT4-video architecture: For each frame, we use EVA-CLIP to get the visual tokens and concatenate each adjacent visual token into a singular token then convert these tokens to the language model space using a linear layer and get the language token from LLM tokenizer. Concatenate both the visual and subtitle text tokens together and do this for all the sampled frames and appending the instruction tokens at the end of the input sequence.

sponding descriptions or subtitle indexes, effectively eliminating irrelevant clips from the long video.

Answer module. In the final stage, we provide the original user query along with our retrieved clip descriptions (and subtitles, if available) as a context to our answer module, which generates the ultimate query response. For this purpose, we utilize Llama2-chat [38] as our chosen Answer module instead of MiniGPT4-Video in the text tasks. see the supplementary for more details and ablations.

3.2 Training Pipeline

Large-scale image-text pair pretraining. In the first stage, we train a linear layer, similar as [49], which projects the visual feature encoded by the vision encoder (EVA-CLIP [36]) to the LLM's text space with captioning loss. We leverage a combined image captioning dataset that includes images from LAION [33], Conceptual Captions [34], and SBU [29] to align the visual feature with LLM's input space. To efficiently utilize the context length of LLM for video, we concatenate every four neighboring visual tokens into a single token, reducing the number of tokens per image by 75% from 256 to 64 same as in [4].

Large-scale video-text pair pretraining. In the second stage, we enable the model to understand short videos by taking multiple frames as input. Specifically, we sample a maximum of 45 frames from each short video. During this stage, we use the predefined prompts in the following template:

<s>[INST]<Img><FrameFeature_ 1><Sub><Subtitle text_ 1>... <Img> <FrameFeature_ N><Sub><Subtitle text_ N><Instruction></INST>

where $N \leq 45$. In this prompt, each *<FrameFeature>* is replaced by the sampled video frame encoded by the vision backbone. The *<Subtitle text>* represents the subtitle for the corresponding frame if applicable, and *<Instruction>* represents a randomly sampled instruction from our predefined instruction set containing variant forms of instruction, such as *"Briefly describe this video"*. We use combined video captioning data incorporating CMD [1] and WebVid [2] for large-scale video captioning training.

Video question answering instruction finetuning. In this phase, we adopt the same training strategy implemented in the second stage but focus on leveraging high-quality video-question-answering datasets for instruction fine-tuning. This fine-tuning stage helps to enhance the model's ability to interpret the input video and generate precise responses to the corresponding questions. The template is the same as the second stage with <*Instruction*> replaced by general questions as mentioned in the Video-ChatGPT [26] dataset.

4 Experiments

4.1 Datasets

Training Datasets The Condensed Movies Video Captions dataset (CMD) [1] includes around 15,938 videos, with lengths between one to two minutes. However, CMD's captions are of limited quality, featuring an average sentence length of 14 words so we used it in the pre-training stage.

The Webvid dataset [2] contains two million videos. For our purposes, we've filtered 42K from this dataset to match CMD's video duration range, focusing on videos lasting one to two minutes and also used this dataset in the pre-training dataset.

The Video Instruction Dataset [27] offers 100K question-answer pairs across 13,224 videos, distinguished by its high-quality annotations. Questions come with detailed answers, averaging 57 words per sentence. This data set spans various types of questions, including video summarization-based and description-based QAs that delve into spatial, temporal, relationships, and reasoning aspects, as well as creative or generative QAs.

Short Benchmarks Our MiniGPT4-Video is tested with Video ChatGPT benchmark five skills and with open-ended and MCQ video-question answering benchmarks. The Video ChatGPT benchmark [27], utilizing the ActivityNet-200 dataset [3], is designed to test video-based conversation models on text generation, focusing on five critical dimensions: 1) Correctness of Information: Verifies the generated text's accuracy with video content to avoid errors or misinformation. 2) Detail Orientation: Assesses the responses for thoroughness and detail, ensuring coverage of essential video elements and inclusion of specific, rather than broad, information. 3) Contextual Understanding: Gauges the model's grasp of video context, ensuring responses are contextually appropriate. 4) Temporal Understanding: Checks the model's perception of event sequences within the video. 5) Consistency: Tests output reliability through similar question comparisons. For open-ended questions, model performance is measured using established datasets like MSRVTT-QA [43], MSVDQA [43], TGIF-QA FrameQA [9], and

ActivityNet-QA [45].
For multi-choice question assessments utilize the TVQA dataset [14], based on popular six TV shows, with a validation set of 15,253 QA pairs for evaluation.

Long Benchmarks We have conducted comprehensive evaluations on three extensive and demanding long video benchmarks: Movie-QA [37], LLama-vid [20], and Movie Chat [35]. Additionally, we adapted the short video benchmark TVQA for long video analysis.

For Movie-QA [37], we assessed the overlapped movies between Movie-QA and MovieNet [8] because Movie-QA videos is not avaialble and it is only short clips,we ended up with 30 overlapped movies from the validation set, each lasting between 1 and 2 hours. The new validation subset encompasses 1,081 questions, primarily based on movie plot.

The LLama-vid [20] dataset features QA pairs focusing on three domains: video summary (1k), movie plot (4k), and detailed reasoning (4k). The absence of category labels prompted us to employ GPT-4 for the classification of the questions, dividing them into two types : general questions (covering plot and reasoning) and summary questions. Due to the original dataset's training-only designation and lack of a validation set, we created a balanced validation set of 10 % of the full data comprising 800 general questions and 100 summary questions, focused solely on textual content.

Movie Chat [35] includes 1,000 meticulously selected video clips from a variety of movies and TV shows, accompanied by 14,000 manual annotations. These videos span 15 major genres and feature a comprehensive dense caption, three global mode QA pairs, and ten breakpoint-mode QA pairs with precise timestamps. The collection predominantly consists of videos lasting between 10K to 12K frames, with 14.6% exceeding this range and only 8.6% falling short, categorized exclusively as visual content.
we evaluated on only the available released training data which is about 10 % of the data because the test data not released while implementing this project.

Furthermore, we introduce an enhanced benchmark based on TVQA, comprising a validation set with 15,253 QA pairs derived from 842 episodes, addressing both textual and visual queries. Originally focused on short videos with 1-minute clips, we have expanded the scope to incorporate entire episodes into the assessment, regardless of the specific video segment to which the question pertains. This adjustment, termed TVQA-Long, significantly increases the difficulty by requiring the analysis of the complete video content to locate the answers. This adjustment facilitates the measurement of retrieval accuracy, as the ground truth clip for each question is known.

Evaluation Metrics . For open ended questions it is hard to evaluate the output of the llm with the ground truth, so following videochatGPT evaluation [27] we employed GPT-3.5 turbo to compare between the generated results and the ground truth. We used the same prompt as videochatgpt [27] to have a fair comparison with their results.

Table 1: Ablation study of the retrieval inputs. The reported numbers are the retrieval accuracy on the TVQA-Long, TVR-Text, and TVR-Vision. & means "and" while | indicates "or".

Retrieval I/P	TVQA	TVR	TVR
		Text	Vision
Subtitles	39.7	66.4	48.4
Summary	12.1	41.2	51.2
Subtitles & Summary	36.9	64.3	46.7
Subtitles | Summary	39.5	67.2	50.8

Table 2: Ablation study on the text encoder models

Text Encoder	Retrieval	Overall
	Acc.	Acc.
bert-base-nli-mean-tokens [32]	19.0	28.4
paraphrase-MiniLM-L6-v2 [32]	31.9	38.03
all-mpnet-base-v2 [32]	32.5	38.33
OpenAI-text-embedding-3-small [28]	46.6	41.78

4.2 Ablation Studies

Retrieval Importance. The retrieval system is one of our core contributions, thus, before ablating its inner design we conduct a simple experiment to demonstrate its importance. To this end, given a long video as an input, we directly fed a sampled version of it. More specifically, we downsample the input video by sampling 45 frames to fit the context length of our MiniGPT4-Video model, then fed it directly to our architecture as one clip. This could be seen as a vanilla approach to process a long video with our MiniGPT4-Video. To avoid the huge information loss that will be caused by this vanilla approach we propose our retrieval module, that given N clips it will automatically retrieve the Top-K clips that are related to the fed question Q. The performance of our model without the retrieval module is close to random, with an accuracy of approximately 25.07%. However, when the retrieval module is incorporated, the accuracy significantly improves, rising to 41.78% on the TVQA-Long benchmark. Notably, the TVQA-Long benchmark consists of 5 options per question, resulting in a random accuracy baseline of 20%.

Retrieval Inner Design. After demonstrating the importance of retrieval design, we ablate each design choice to implement an efficient retrieval system. For each clip i, given a question Q, the subtitles and the summaries embedding, termed E^i_{sub} and E^i_{sum}, respectively, we need to determine what is the best way to retrieve the corresponding clip to the input question. To this end, we explored four possible approaches: 1) Using only E^i_{sub}. 2) Using only E^i_{sum}. 3) Concatenate both embedding E^i_{sub} and E^i_{sum}, namely "and" approach. 4) Treat each type separately, namely "or" approach. For instance, if we have 20 clips, then we will feed 40 embeddings, each 20 representing the E^i_{sub} and E^i_{sum} separately. As shown in Table 1, on the TVQA dataset the summary do not add any value, which could be interpreted as our generated summary is unrepresentative. However, other interpretation is that, the questions provided in the TVQA dataset mainly rely on the text clues not the vision ones. To support this claim and to truly assess our generated summaries, we exploit the TVR dataset [15] which is another dataset of the same videos but different annotations that used in moment retrieval tasks and this dataset has a good prosperity that the descriptions in it is labeled as text descriptions, vision descriptions and text plus vision descriptions, so based on the description type, whether it is based on the visual clues or text. As shown in Table 1, on the TVR-Vision, the summary achieves the best performance, which show the high quality of our generated summaries via our short video model (MiniGPT4-Video)

Text Encoder. As shown in Figure 2, the input subtitles and the generated summaries are encoded using a text-encoder to generate E^i_{sub} and E^i_{sum}, respectively. Table 2 shows the impact of the text encoder on the retrieval accuracy and the overall accuracy, where the better retrieval is linearly correlated with the overall accuracy of the long-video model.

Answer Module. After getting the Top-K retrieved clips, the answer module is responsible to fuse the retrieved clips grounded by the question to produce the final answer. To this end, several ways are studied, as shown in Figure 6: A) Feed directly the retrieved summaries Sum and subtitles Sub with the question to the LLM model to directly answer the question or say I don't know If the provided information not enough. B) Feed the selected video clips V and the question Q to MiniGPT4-Video to generate a new information $info_Q$, which is grounded to the question. Then, feed the new information $info_Q$ with the general input summary Sum and the question Q to the LLM to produce the final answer. C) Following the previous option,with also adding the original subtitles to the context. The table in Figure 6 demonstrates that, option A is the best approach; feed the summaries and the subtitles directly to the LLM. In contrast, when we feed the video clips V, the accuracy drops significantly, options B and C. The reason behind this drop is the model hallucination, especially when the question is not related to the question, which leads to generate confusing information to the context $info_Q$. Please refer to the supplementary materials for detailed examples of the model hallucination in options B and C.

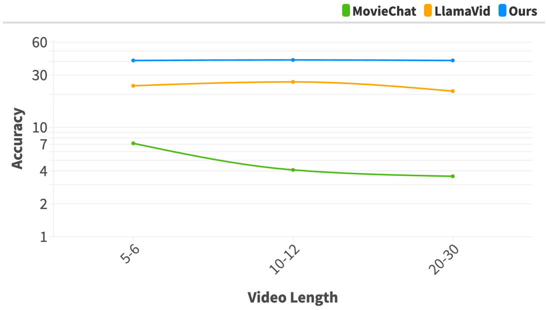
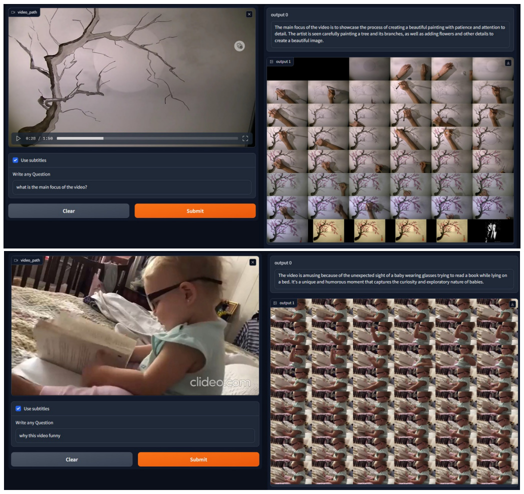


Fig. 4: Ablation study about the video length impact on 5% of TVQA validation set. , video length in minutes

To evaluate our framework's robustness with extended video lengths, we created three versions of the TVQA dataset by altering the aggregation window.

This window compiles long videos from ground-truth short clips that include the answer to a question. Specifically, we combined 5, 10, and 20 clips to produce videos averaging 6, 12, and 24 minutes, respectively. Figure 4 illustrates that our framework maintains its robustness regardless of video length, with both retrieval performance and overall accuracy remaining consistent even as video duration increases. These outcomes, detailed in Figure 4, are based on an analysis of 5% of the TVQA validation set.



☑ Use subtitles

Write any Question: what is the main focus of the video?

output 0: The main focus of the video is to showcase the process of creating a beautiful painting with patience and attention to detail. The artist is seen carefully painting a tree and its branches, as well as adding flowers and other details to create a beautiful image.

☑ Use subtitles

Write any Question: why this video funny

output 0: The video is amusing because of the unexpected sight of a baby wearing glasses trying to read a book while lying on a bed. It's a unique and humorous moment that captures the curiosity and exploratory nature of babies.

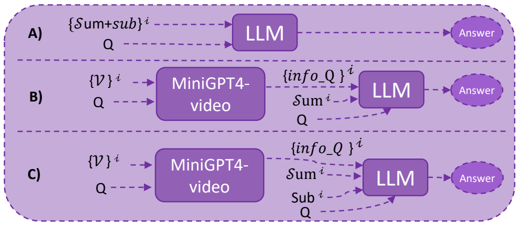
Fig. 5: MiniGPT4-video Qualitative results, demonstrating video understanding abilities; more qualitative results are provided in the supplementary.

4.3 Comparison to State-Of-The-Art

Long Video Benchmarking We evaluate the efficacy of our proposed framework, Goldfish:, across several well-established benchmarks, specifically the LLama-Vid [20], MovieChat [35], Movie QA [37], and TVQA-Long [14] datasets. To thoroughly examine our framework's capabilities, we analyze input modalities in two configurations: vision-only (V) and vision combined with input subtitles (V+T).

Our findings, detailed in Table 3, indicate that our framework surpasses all existing long video baselines in the vision modality.We establish state-of-the-art (SOTA) performance on these challenging benchmarks. This achievement holds true even under an unfair comparison against LLama-Vid [20], which benefits from using the MovieNet dataset while training and these movies are in both LLama-vid [20] benchmark and Movie QA [37]. Despite this advantage, our results significantly outperform the competition.

Incorporating both video frames and aligned subtitles into our model leads to an average performance boost of 8% across the benchmarks. As highlighted in



Answer module	**Acc.**
A)	41.78
B)	27.6
C)	27.72

Fig. 6: Answer module ablation study on TVQA dataset, validation set.

Table 3: Long video benchmarking results on four benchmarks: LLama-Vid, MovieChat, Movie QA, and our proposed TVQA-Long. The "V" modality indicates the use of video frames only, while "V+T" indicates the use of both video frames and subtitles. The dagger (†) symbol denotes methods that used the benchmark during training, implying an unfair comparison.

Method	Modalities	Open Ednded Questions				MCQ			
		LLama-Vid [20]		MovieChat [35]		Movie QA [37]		TVQA-Long	
		Acc.↑	Score↑	Acc.↑	Score↑	Acc.↑	Score↑	Acc.↑	Score↑
LLaMA-VID [20]	V	20.68	**2.41**	53.2	3.81	24.42	2.19	24.63	2.16
MovieChat [35]	V	11.71	1.45	NA	NA	16.18	1.68	5.0	0.86
Ours	V	**23.09**	2.19	**67.6**	**4.23**	**28.49**	**2.8**	**28.61**	**2.78**
LLaMA-VID [20]	V+T	41.4†	3.07†	NA	NA	37.65†	3.03†	26.81	2.21
Ours	V+T	31.49	2.43	NA	NA	35.24	**3.1**	**41.78**	**3.21**

Table 3, this enhanced approach enables us to outperform LLama-Vid [20] on the TVQA benchmark, providing a fair comparison since LLama-Vid [20] utilizes the other benchmarks during its training phase.

Short Video Benchmarking On short-video understanding, we continue to secure state-of-the-art (SOTA) results, outperforming contemporaneous works, including LLama-Vid [20]. To validate our framework's proficiency in short-video analysis, we conducted evaluations against current SOTA methodologies across an extensive suite of five benchmarks: Video ChatGPT, MSVD, MSRVTT, TGIF, and TVQA. These benchmarks collectively offer a comprehensive platform for assessing short-video comprehension capabilities, with five focusing on open-ended questions and TVQA featuring multiple-choice questions.

Our results, presented in Tables 4 and 5, demonstrate our framework's superiority over competing methods by a significant margin, affirming our considerable advancements across a varied and demanding collection of benchmarks. To thoroughly evaluate our approach, we devised two variations of our framework: one analyzing purely visual elements and another incorporating subtitles. The performance enhancements achieved with these models are noteworthy, registering gains of 3.23%, 2.03%, 16.5% and 23.59% on the MSVD, MSRVTT, TGIF, and TVQA benchmarks respectively. This underscores our framework's ability to achieve SOTA results across the board, markedly elevating performance in the domain of short-video understanding. The visualization results of our method are shown in Fig. 5. We will show more visualization results in the appendix.

Table 4: Qualitative results on Video-ChatGPT benchmark.

Method	**Using Subtitles**	**Video ChatGPT**				
		Information Correctness	Detailed Orientation	Contextual Understanding	Temporal Understanding	Consistency
LLaMA Adapter [48]	✗	2.03	2.32	2.30	1.98	2.15
Video LLaMA [46]	✗	1.96	2.18	2.16	1.82	1.79
Video-ChatGPT [26]	✗	2.40	2.52	2.62	1.98	2.37
BT-Adapter-7B [25]	✗	2.68	2.69	3.27	2.34	2.46
LLaMA-VID-7B [20]	✗	2.96	3.00	3.53	2.46	2.51
Ours-7B	✗	2.93	2.97	3.45	**2.47**	**2.60**
Video Chat [18]	✓	2.23	2.50	2.53	1.94	2.24
Ours-7B	✓	**3.08**	**3.02**	**3.57**	**2.65**	**2.67**

Table 5: Short video benchmarking results on MSVD, MSRVTT, TGIF, ActivityNet and TVQA.

Method	**Using Subtitles**	**Open Ended Questions**								**MCQ**
		MSVD		MSRVTT		TGIF		ActivityNet		TVQA
		Acc.↑	Score↑	Acc.↑	Score↑	Acc.↑	Score↑	Acc.↑	Score↑	Acc.↑
FrozenBiLM [44]	✗	32.2	–	16.8	–	41	–	24.7	–	29.7
LLaMA Adapter [48]	✗	54.9	3.1	43.8	2.7	–	–	34.2	2.7	–
Video LLaMA [46]	✗	51.6	2.5	29	1.8	–	–	12.4	1.1	–
Video Chat [18]	✗	56.3	2.8	45	2.5	34.4	2.3	26.5	2.2	–
Video-ChatGPT [26]	✗	64.9	3.3	49.3	2.8	51.4	3.0	35.2	2.7	23.35
BT-Adapter-7B [25]	✗	67.7	3.7	57	3.2	–	–	45.7	3.2	–
LLaMA-VID-7B [20]	✗	69.7	3.7	57.7	3.2	–	–	**47.4**	3.3	–
Ours-7B	✗	72.93	3.84	**58.83**	**3.29**	67.9	3.71	45.6	3.2	36.45
Ours-7B	✓	N/A	N/A	**59.73**	**3.3**	N/A	N/A	46.3	**3.4**	**46.94**

5 Conclusion

In this paper, we identified the main challenges of the current video-centric LLMs to process long videos. Based on the analyses, we introduced the Goldfish method, which eases the *noise and redundancy* challenge and *computational and memory* challenge. Goldfish introduces a retrieval approach that focuses on top-k relevant clips, allowing efficient processing of videos of any length. In contrast, most of the previous models can only process minutes-long videos. We developed MiniGPT4-Video, which enhances video content interpretation from single to multiple frames, significantly improving video understanding. This model serves both as a part of Goldfish for long video summarization and as a standalone model for short video tasks. Our Goldfish achieves state-of-the-art results in long video understanding across four benchmarks with only the vision content and achieved SOTA in with vision and subtitles in zeroshot evaluation on TVQA. Notably, in the proposed TVQA-long benchmark, we outperformed the previous method by 14.94%. Our MiniGPT4-Video also exceeds performance standards in short video benchmarks. We hope our proposed Goldfishmethod and the TVQA-long benchmark can benefit future research in the long video understanding.

References

1. Bain, M., Nagrani, A., Brown, A., Zisserman, A.: Condensed movies: Story based retrieval with contextual embeddings. In: Proceedings of the Asian Conference on Computer Vision (2020)
2. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1728–1738 (2021)
3. Caba Heilbron, F., Escorcia, V., Ghanem, B., Carlos Niebles, J.: Activitynet: A large-scale video benchmark for human activity understanding. In: Proceedings of the ieee conference on computer vision and pattern recognition. pp. 961–970 (2015)
4. Chen, J., Zhu, D., Shen, X., Li, X., Liu, Z., Zhang, P., Krishnamoorthi, R., Chandra, V., Xiong, Y., Elhoseiny, M.: Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. arXiv preprint arXiv:2310.09478 (2023)
5. Chen, W., Hu, H., Chen, X., Verga, P., Cohen, W.W.: Murag: Multimodal retrieval-augmented generator for open question answering over images and text (2022)
6. Doe, J.: The needle in a haystack test. `https://towardsdatascience.com/the-needle-in-a-haystack-test-a94974c1ad38` (January 2021), accessed: date-of-access
7. Gu, J., Wang, Y., Cho, K., Li, V.O.K.: Search engine guided non-parametric neural machine translation (2018)
8. Huang, Q., Xiong, Y., Rao, A., Wang, J., Lin, D.: Movienet: A holistic dataset for movie understanding (2020)
9. Jang, Y., Song, Y., Yu, Y., Kim, Y., Kim, G.: Tgif-qa: Toward spatio-temporal reasoning in visual question answering (2017)
10. Karpukhin, V., Oğuz, B., Min, S., Lewis, P., Wu, L., Edunov, S., Chen, D., tau Yih, W.: Dense passage retrieval for open-domain question answering (2020)
11. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: Nearest neighbor language models (2020)
12. Khattab, O., Santhanam, K., Li, X.L., Hall, D., Liang, P., Potts, C., Zaharia, M.: Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp (2023)
13. Le, H., Chen, N.F., Hoi, S.C.H.: Vgnmn: Video-grounded neural module network to video-grounded language tasks (2022)
14. Lei, J., Yu, L., Bansal, M., Berg, T.L.: Tvqa: Localized, compositional video question answering (2019)
15. Lei, J., Yu, L., Berg, T.L., Bansal, M.: Tvr: A large-scale dataset for video-subtitle moment retrieval (2020), `https://arxiv.org/abs/2001.09099`
16. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., tau Yih, W., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks (2021)
17. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
18. Li, K., He, Y., Wang, Y., Li, Y., Wang, W., Luo, P., Wang, Y., Wang, L., Qiao, Y.: Videochat: Chat-centric video understanding (2024)
19. Li, X., Zhao, R., Chia, Y.K., Ding, B., Joty, S., Poria, S., Bing, L.: Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources (2024)

20. Li, Y., Wang, C., Jia, J.: Llama-vid: An image is worth 2 tokens in large language models (2023)
21. Liang, H., Zhang, W., Li, W., Yu, J., Xu, L.: Intergen: Diffusion-based multi-human motion generation under complex interactions (2023)
22. Lin, B., Zhu, B., Ye, Y., Ning, M., Jin, P., Yuan, L.: Video-llava: Learning united visual representation by alignment before projection. arXiv preprint arXiv:2311.10122 (2023)
23. Lin, W., Byrne, B.: Retrieval augmented visual question answering with outside knowledge (2022)
24. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
25. Liu, R., Li, C., Ge, Y., Shan, Y., Li, T.H., Li, G.: One for all: Video conversation is feasible without video instruction tuning (2023)
26. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models (2023)
27. Maaz, M., Rasheed, H., Khan, S., Khan, F.S.: Video-chatgpt: Towards detailed video understanding via large vision and language models. arXiv preprint arXiv:2306.05424 (2023)
28. OpenAI: New embedding models and api updates (2024), `https://openai.com/blog/new-embedding-models-and-api-updates`
29. Ordonez, V., Kulkarni, G., Berg, T.: Im2text: Describing images using 1 million captioned photographs. Advances in neural information processing systems **24** (2011)
30. Peng, H., Parikh, A.P., Faruqui, M., Dhingra, B., Das, D.: Text generation with exemplar-based adaptive decoding (2019)
31. Ram, O., Levine, Y., Dalmedigos, I., Muhlgay, D., Shashua, A., Leyton-Brown, K., Shoham, Y.: In-context retrieval-augmented language models (2023)
32. Reimers, N.: Pretrained models (2024), `https://www.sbert.net/docs/pretrained_models.html`
33. Schuhmann, C., Vencu, R., Beaumont, R., Kaczmarczyk, R., Mullis, C., Katta, A., Coombes, T., Jitsev, J., Komatsuzaki, A.: Laion-400m: Open dataset of clip-filtered 400 million image-text pairs (2021)
34. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
35. Song, E., Chai, W., Wang, G., Zhang, Y., Zhou, H., Wu, F., Chi, H., Guo, X., Ye, T., Zhang, Y., Lu, Y., Hwang, J.N., Wang, G.: Moviechat: From dense token to sparse memory for long video understanding (2023)
36. Sun, Q., Fang, Y., Wu, L., Wang, X., Cao, Y.: Eva-clip: Improved training techniques for clip at scale. arXiv preprint arXiv:2303.15389 (2023)
37. Tapaswi, M., Zhu, Y., Stiefelhagen, R., Torralba, A., Urtasun, R., Fidler, S.: Movieqa: Understanding stories in movies through question-answering. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4631–4640 (2016)
38. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
39. Wang, Y., Li, P., Sun, M., Liu, Y.: Self-knowledge guided retrieval augmentation for large language models (2023)
40. Weston, J., Dinan, E., Miller, A.H.: Retrieve and refine: Improved sequence generation models for dialogue (2018)

41. Whitehead, S., Ji, H., Bansal, M., Chang, S.F., Voss, C.: Incorporating background knowledge into video description generation. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 3992–4001 (2018)
42. Wu, Y., Wei, F., Huang, S., Wang, Y., Li, Z., Zhou, M.: Response generation by context-aware prototype editing (2018)
43. Xu, D., Zhao, Z., Xiao, J., Wu, F., Zhang, H., He, X., Zhuang, Y.: Video question answering via gradually refined attention over appearance and motion. In: Proceedings of the 25th ACM international conference on Multimedia. pp. 1645–1653 (2017)
44. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models (2022)
45. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering (2019)
46. Zhang, H., Li, X., Bing, L.: Video-llama: An instruction-tuned audio-visual language model for video understanding. arXiv preprint arXiv:2306.02858 (2023)
47. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Guiding neural machine translation with retrieved translation pieces (2018)
48. Zhang, R., Han, J., Zhou, A., Hu, X., Yan, S., Lu, P., Li, H., Gao, P., Qiao, Y.: Llama-adapter: Efficient fine-tuning of language models with zero-init attention. arXiv preprint arXiv:2303.16199 (2023)
49. Zhu, D., Chen, J., Shen, X., Li, X., Elhoseiny, M.: Minigpt-4: Enhancing vision-language understanding with advanced large language models. arXiv preprint arXiv:2304.10592 (2023)
50. Zhuge, M., Gao, D., Fan, D.P., Jin, L., Chen, B., Zhou, H., Qiu, M., Shao, L.: Kaleido-bert: Vision-language pre-training on fashion domain. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12647–12657 (2021)