A Survey on Visual Understanding Multimodal Large Language Models

John Doe
School of Computer Science
University of Level Zero
Storage Room, Backrooms
entity1@backrooms.net

***Abstract*—Large Language Models (LLMs) have demonstrated remarkable performance in natural language processing tasks by increasing the number of model parameters and the volume of training data. To extend this capability to visual understanding tasks, multimodal models (MM-LLMs) have been developed by integrating LLMs with visual encoders. These models are capable of handling tasks such as image captioning, detailed image description, and image question answering, as well as more complex tasks like video understanding. This survey first outlines the characteristics and challenges of three visual understanding tasks: image understanding, short video understanding, and long video understanding. It then provides a detailed introduction to the model architectures used in these tasks, highlighting their similarities and differences, and discusses the evolving trends in model training methods. Additionally, the paper presents performance evaluations of several representative models and offers insights into the future directions of visual understanding MM-LLMs.**

***Index Terms*—Multimodal Models, Visual Understanding, Large Language Models, Image Understanding, Short-video Understanding, Long-video Understanding**

I. Introduction

Large Language Models (LLMs) have demonstrated the capability to generate and comprehend natural language text by increasing the number of model parameters and the volume of training data. They have shown robust performance across various downstream tasks, such as text expansion, summarization, and conversational dialogue. To extend the powerful capabilities of LLMs in natural language tasks to the domain of visual understanding, researchers have connected LLMs with visual encoders, thereby equipping LLMs with ”eyes” to interpret visual information. This integration has led to the development of multimodal models in the field of visual understanding (MM-LLMs). Furthermore, through continuous optimization and improvement of the interaction mechanisms between visual encoders and LLMs, a variety of MM-LLMs have been developed to adapt to multiple downstream tasks, including image captioning [1]–[5], detailed image description, visual question answering, and visual grounding [**?**], [6]–[8], extending even to video understanding [9]–[16]. Depending on whether the input visual information is an image or a video, and the duration of the video, visual understanding MM-LLMs are broadly categorized into three types: image understanding models, short video understanding models, and long video understanding models.

In visual understanding MM-LLMs for images and short videos, a single image or a sequence of consecutive frames is encoded into a series of visual tokens, which are then integrated with text tokens. These visual tokens either serve as a soft prompt [6] or directly interact with the internal mechanisms of the LLM [4], enabling the MM-LLM to transform visual input into linguistic output, thereby accomplishing visual information comprehension tasks. Such visual understanding models have already found extensive real-world applications, including the image-text and video-text chat functionalities in GPT-4 and other large models.

In the context of long-video visual understanding within MM-LLM, the model is required to process longer sequences of visual information inputs, which include more frames and more complex semantics. This inevitably involves the model’s capabilities in long-term memory and extended reasoning, necessitating more sophisticated designs in model architecture, training methods, and the establishment of a more robust framework for evaluating model effectiveness. Although the application of such visual understanding models has not yet become widespread, it is not difficult to predict that these models will have broader and more useful applications compared to image or short-video understanding models. Potential applications include commenting on live sports broadcasts or extracting highlights, narrating movies, assisting in the investigation of surveillance footage, and ultimately applying them in embodied intelligence, where they can assist robots in decision-making by analyzing real-time first-person perspective videos (egocentric video) from their cameras.

Although the three categories of visual understanding MM-LLMs differ in their application scenarios and detailed designs, their fundamental model architectures and training methodologies are largely similar. The basic framework of these models typically consists of three main components: a visual encoder module, a cross-modal connection module, and an LLM module. The training process generally can be divided into two main phases: the pre-training phase and the instruction fine-tuning phase. The pre-training phase is primarily aimed at aligning the visual and linguistic modalities, while the instruction fine-tuning phase focuses on enhancing the model’s ability to follow instructions or optimizing the consistency between the model’s outputs and the outputs required for downstream tasks.

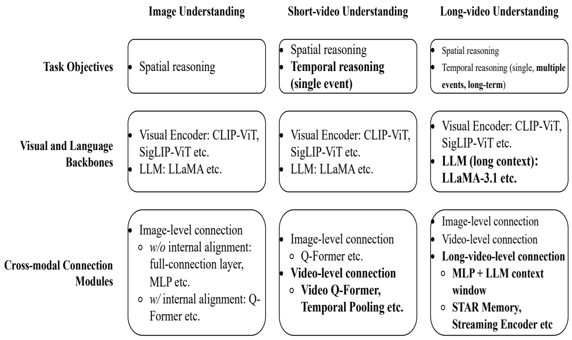


	Image Understanding	Short-video Understanding	Long-video Understanding
Task Objectives	• Spatial reasoning	• Spatial reasoning • **Temporal reasoning (single event)**	• Spatial reasoning • Temporal reasoning (single, **multiple events, long-term**)
Visual and Language Backbones	• Visual Encoder: CLIP-ViT, SigLIP-ViT etc. • LLM: LLaMA etc.	• Visual Encoder: CLIP-ViT, SigLIP-ViT etc. • LLM: LLaMA etc.	• Visual Encoder: CLIP-ViT, SigLIP-ViT etc. • **LLM (long context): LLaMA-3.1 etc.**
Cross-modal Connection Modules	• Image-level connection ○ *w/o* internal alignment: full-connection layer, MLP etc. ○ *w/* internal alignment: Q-Former etc.	• Image-level connection ○ Q-Former etc. • **Video-level connection** ○ **Video Q-Former, Temporal Pooling etc.**	• Image-level connection • Video-level connection • **Long-video-level connection** ○ **MLP + LLM context window** ○ **STAR Memory, Streaming Encoder etc**

Fig. 1. A brief comparison of three visual understanding tasks. The **bold** items are items that do not exist in previous visual understanding tasks.

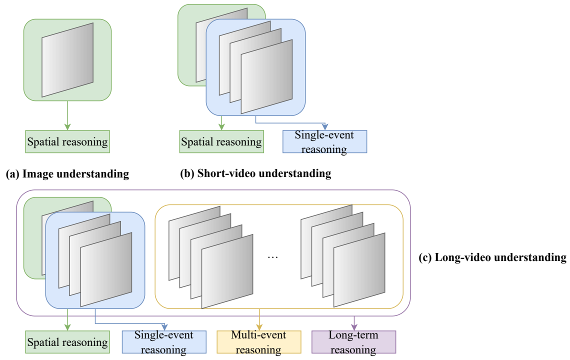


Fig. 2. A simple view of tasks of (a) image understanding (b) short-video understanding and (c) long-video understanding.

In the basic framework, the visual encoder module typically employs pre-trained image encoders such as CLIP-ViT and SigLIP-ViT. The cross-modal connection module ranges from simple fully connected layers or multilayer perceptrons (MLPs) to more complex designs like Q-Former [3], primarily used to map visual tokens from the visual latent space to text tokens in the linguistic latent space (alignment of visual and linguistic modalities). Notably, to accommodate long video inputs, the cross-modal connection module not only requires modality alignment but also needs to further compress a large number of high-dimensional image tokens, either by reducing their number or compressing their dimensions. This process must also consider the issue of information loss. Therefore, designing a cross-modal connection module that can effectively compress image tokens while preserving visual information remains a hot research topic. The LLM module can utilize currently mature large models. Examples of the actual architectures used for these three modules in different models are illustrated in Figure 1.

This paper primarily introduces the fundamental objectives and challenges of three visual understanding tasks: image understanding, short video understanding, and long video understanding (Section II). It also presents several model architectures related to these tasks, analyzing their similarities and differences, thereby providing a concise overview of the development trajectory of visual understanding MM-LLMs (Section III). Additionally, from the perspective of model training, such as advancements in training methods and dataset construction strategies, this paper further summarizes the development trends of visual understanding MM-LLMs (Section IV). Finally, the paper presents performance evaluation results of several typical models (Section V). Based on the aforementioned summaries and analyses, the paper briefly outlines several potential future directions for the development of visual understanding MM-LLMs (Section VI).

II. Visual Information Understanding

The fundamental reason for the differences in the MM-LLM architecture details for image, short video, and long video understanding lies in the varying input sizes, information dimensions, and semantic scales of these three types of tasks. The diverse challenges in these three tasks also stem from these differences. The characteristics of the three tasks are summarized in Figure 1, while a more intuitive illustration is provided in Figure 2.

A. Image Understanding

a) Task characteristics: In terms of input size, image understanding tasks, such as image captioning, detailed description, and visual question answering, typically involve only a single static image. The characteristic of such input is its small size, and given the current performance of computers, it often does not require compressing and storing visual tokens. Instead, the input image can be encoded at a finer granularity, such as by dividing the image into smaller patches and encoding them sequentially [8], [10]. Regarding the dimensionality of information, since an image contains only a single static content, it inherently includes only spatial information; thus, the model does not need to consider how to handle temporal information during the encoding process. In terms of semantic scale, images often contain no events (e.g., landscape photos) or only a single event (e.g., a snapshot of a dynamic scene), resulting in relatively limited semantic information. Consequently, the model does not need to perform cross-event reasoning or learn interactions among multiple semantics.

b) Task challenges: During the development of image understanding MM-LLM, initially, due to training solely on image-text pair data obtained from the internet, such datasets often contain a large amount of noisy data (for instance, an indoor home photo might come from a real estate website with its text description as ”new summer model”; this is clearly not appropriate training data), which negatively impacts the training process. Thus, (1) *how to acquire a large quantity of high-quality training data* is one of the challenges in image understanding tasks [1]. Additionally, in the early stages of the visual understanding MM-LLM field, individual models were often specially trained for specific downstream tasks, such as models for image classification or object detection only outputting class labels or bounding box coordinates. Such image understanding tasks can be termed as ”understanding subclasses within understanding tasks (understanding)"; whereas other tasks like image captioning and visual dialog fall into the category of "generation subclasses within understanding tasks (generation)". Therefore, (2) *how to unify understanding and generation subclasses within image understanding tasks* is also a challenge [2]. After preliminarily addressing these two types of "material-level" challenges, the challenges in image understanding tasks shift towards the "spiritual level", i.e., (3) *how to enhance the model's instruction-following capability and zero-shot learning ability*, (4) *how to improve the quality and correctness of the model's text output* [6], [7], and (5) *how to optimize model performance and achieve efficient local deployment of the model* [8].

B. Short-Video Understanding

It is important to note that the definition of "short video" discussed here differs from that used in entertainment platforms. In entertainment contexts, short videos typically refer to those with a duration of less than ten or five minutes. In contrast, the term "short video" in this context specifically refers to videos that are approximately one minute long or shorter. According to the definition used in the former context, the latter's notion of a short video could already be considered a "long video".

a) Task characteristics: Regarding input size, the short video understanding task clearly involves a larger input volume compared to image understanding tasks. A single input in short video understanding can be considered as multiple inputs in image understanding, where each frame of a video is treated as an individual image and fed into a visual encoder all at once for encoding. Performing fine-grained encoding on such data (encoding each frame separately) would inevitably lead to a significant increase in memory usage, making it impractical. Current research remains focused on effectively encoding video information without increasing the size of the visual encoding [17].

In terms of information dimensions, videos obviously contain not only spatial information within individual frames but also temporal information across multiple frames. This transforms what was a single image input into a continuous sequence input. When encoding, models must also consider how to effectively preserve temporal information. Regarding semantic scale, due to time constraints, a short video typically contains only one event (such as a person singing), which is slightly more semantically complex than a single image but still does not require the model to have strong memorization capabilities.

b) Task challenges: (1) Similar to image understanding tasks, the challenges in short video understanding tasks also include aspects such as dataset quality, the model's instruction-following capability, and model performance. (2) Additionally, since short videos contain more dimensions and a larger scale of semantic information, *how to efficiently encode this information into visual tags* is also one of the challenges [17].

C. Long-Video Understanding

a) Task characteristics: Regarding input size, similarly, a single input in long video understanding can be viewed as multiple inputs in short video understanding. Given that the duration of long videos often spans several minutes to several hours, the input volume for long video understanding grows exponentially compared to image understanding and short video understanding. Efficiently compressing and storing image tokens has thus become an urgent necessity. In terms of information dimensions, long videos are consistent with short videos, both containing spatial and temporal information.

On the semantic scale, long videos are highly likely to involve multiple events (for example, a person first singing, then dancing, followed by rapping, and finally playing basketball). Therefore, the semantic scale of long videos is further increased beyond that of short videos. Moreover, as time progresses linearly, the importance of events does not always decrease linearly with their distance from the present. This requires the model to have strong memory capabilities.

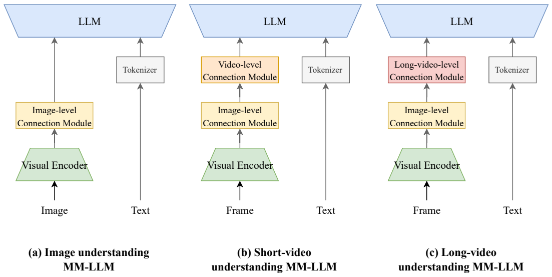
b) Task challenges: In addition to the challenges faced by image understanding and short video understanding tasks, long video understanding introduces new challenges due to its characteristics of multiple events and extended time spans. First, because long videos contain a larger number of events, (1) *how the model can efficiently store and remember a large amount of information* is one of the challenges. Moreover, there may be parallel, progressive, or contrasting relationships between multiple events, and the model needs to learn (2) *how to understand the connections between events and perform cross-event reasoning*.

Secondly, the time span of long videos is not only long but also highly variable, ranging from several minutes to several hours. For relatively shorter long videos, such as those spanning several minutes, models can adopt more efficient compression and storage methods similar to those used in short video understanding, processing the entire video input at once. However, for longer long videos, such as those spanning tens of minutes to several hours, continuing to use a single compression and storage method for the entire video inevitably leads to the loss of excessive temporal and spatial information. Therefore, the problem shifts to (3) *how to effectively perform dynamic storage of video inputs*, i.e., *how to efficiently conduct online long video understanding*.

For online long video understanding, the challenges can be further broken down into: (5) *how to balance old video memories with new video information (designing an efficient memory bank)*, (6) *how to efficiently retrieve specific memories from the memory bank (aligning memories with text)*, and (7) *how to quickly respond to user text inputs (decoupling memory videos from response inputs)* [9], [16].

III. Development of Model Architectures

This section primarily discusses the architectural design of models for image understanding, short video understanding, and long video understanding tasks. The illustration of the three common modules (visual encoder module, cross-modal connection module, and LLM module) is shown in Figure 2. This section will focus on the structure of the cross-modal connection module, which not only handles the fundamental


LLM
Tokenizer
Image-level Connection Module
Visual Encoder
Image
Text
LLM
Video-level Connection Module
Tokenizer
Image-level Connection Module
Visual Encoder
Frame
Text
LLM
Long-video-level Connection Module
Tokenizer
Image-level Connection Module
Visual Encoder
Frame
Text

(a) Image understanding MM-LLM

(b) Short-video understanding MM-LLM

(c) Long-video understanding MM-LLM

Fig. 3. A simple view of model architectures in (a) image understanding, (b) short-video understanding and (c) long-video understanding.

task of modal alignment in image understanding but also plays a crucial role in integrating temporal information across multiple frames and events in video understanding tasks, as well as efficiently compressing, memorizing, and retrieving visual information.

In conjunction with some of the challenges proposed in Section II, it can be seen that the selection of the LLM module can address challenges (3) and (4) in image understanding tasks (which also exist in the other two types of tasks). The design of the cross-modal connection module can resolve challenge (2) in image understanding tasks, challenge (2) in short video understanding tasks, and challenges (2) to (7) in long video understanding tasks.

A. Visual Encoder Module and LLM Module

TABLE I
VISUAL ENCODER MODULES AND LLM MODULES USED IN SOME VISUAL UNDERSTANDING MM-LLMS

Model	**Visual Encoder**	**LLM**
InstructBLIP	EVA-CLIP-ViT-G/14	FlanT5, Vicuna-7B/13B
VideoChat	EVA-CLIP-ViT-G/14	StableVicuna-13B
MovieChat	EVA-CLIP-ViT-G/14	LLama-7B
LLaVA-NeXT	CLIP-ViT-L/14	Vicuna1.5-7B/13B, Mistral-7B, Nous-Hermes-2-Yi-34B
MiniGPT-4-Video	EVA-CLIP-ViT-G/14	LLaMA2-7B, Mistral-7B
LLaVA-OneVision	SigLIP-SO400M	Qwen2-7B

Zoom in for a better view of the table.

Although visual understanding MM-LLMs can be categorized into those designed for images and those for long/short videos, the visual encoder modules and LLM modules used across these MM-LLM types are largely similar. There is not much difference in the selection of visual encoder modules. However, the development of LLM modules in the field of natural language processing has provided more choices over time: early large models for image understanding transitioned from using smaller-parameter LLMs to larger-parameter LLMs, while large models for long video understanding, which require long-term reasoning capabilities, naturally tend to choose LLMs with more parameters and larger context windows. For details on the selection of these two modules in some visual understanding MM-LLMs, see Table I.

a) Visual encoder module: The pre-trained visual encoder primarily handles extracting visual information from raw images/video frames and representing this information using visual tokens in a latent space. Widely used visual encoder modules for the three types of visual understanding MM-LLMs include CLIP-ViT-L/14, EVA-CLIP-ViT-G/14, OpenCLIP-ViT-bigG/14, and SigLIP-SI400M, among others.

Research has shown that the parameter size or architecture type of the visual encoder module has a relatively minor impact on the overall performance of visual understanding MM-LLMs [6], [7]. Apart from the performance of the cross-modal connection module, at the level of the visual encoder module, factors that significantly influence the overall model performance include the quality of the training data, the resolution size that the module can accept, and the dimensionality of the visual tokens generated by the module [10].

b) LLM module: The pre-trained LLM is primarily responsible for reasoning and decision-making based on multimodal information passed from the input, ultimately generating natural language output that meets user expectations. Since the design of the overall architecture of visual understanding MM-LLMs may involve modifications to the internal architecture of the LLM, more widely used are open-source large models such as Flan-T5, LLaMA, Vicuna, QWen, Mistral, OpenFlamingo, etc.

After the cross-modal connection module aligns the visual and language spaces, the performance of the LLM module has the most significant impact on the overall performance of the visual understanding MM-LLM [6], [7]. A less performant LLM module tends to overfit the training data during the training phase. For example, if the training data contains a large number of image-short caption pairs, the final model's ability to generate longer text outputs will be limited [5]. In contrast, high-performance LLM modules can often transfer their capabilities to the visual understanding MM-LLM, such as strong instruction-following abilities and robust zero-shot learning capabilities [7]. If coupled with a larger parameter size, the emergent phenomena observed in LLMs can also manifest in MM-LLMs, enabling the latter to exhibit new capabilities during testing that were not introduced during training [6], [7]. In recent large models for long video understanding, LLMs with larger context windows have been directly utilized, successfully transferring their long-context analysis capabilities into these long video understanding models [11].

B. Cross-modal Connection Module

The cross-modal connection module is primarily responsible for aligning the visual space with the language space, providing soft prompts related to visual information to the downstream LLM module or directly integrating visual information into the LLM model. If the performance of the LLM is considered as whether a visual understanding MM-LLM can "speak well," then the performance of the cross-modal connection module can be viewed as whether the model "can speak" at all: only when the model "can speak" can it learn to "speak well." Similar to the three categories of visual understanding tasks, cross-modal connection modules can also be divided into three categories: image-level, video-level, and long-video-level. Among these, the image-level connection module serves as the technical foundation for the other two types, with most of the modal alignment work being accomplished at this level. The other two types of modules build upon the image-level module by introducing specialized architectures designed to understand the corresponding visual information, thereby enhancing the model's ability to process short/long video inputs after modal alignment.

a) Image-level connection modules: The image-level connection module primarily handles the most fundamental modal alignment tasks. This modal alignment can be divided into two subcategories: internal modal alignment within the connection module and external modal alignment. Internal modal alignment exists only in cross-modal connection modules that can accept both image and text inputs, such as Q-Former [3]. This alignment method requires training only the visual encoder module and the cross-modal connection module, using objectives abstracted from various downstream tasks to train these two modules to handle generalized tasks. Ultimately, without connecting to an LLM, the visual encoding module and the connection module alone can achieve good performance on some simple downstream tasks.

External modal alignment does not impose any requirements on the types of inputs accepted by the connection module. This alignment method involves end-to-end training/fine-tuning of the entire model with the LLM connected (training parameters of the connection module and the LLM; training the visual encoder module's parameters is optional), ultimately ensuring that the output of the connection module can serve as soft prompts for the LLM or be integrated into the LLM, effectively assisting the LLM in understanding visual information.

Based on the above introduction, it is not difficult to analyze the interaction between the two alignment methods: internal modal alignment essentially prepares for external modal alignment by pre-aligning visual modality information towards the language modality, thus accelerating the speed and improving the effectiveness of external modal alignment.

Depending on whether the connection module performs internal modal alignment, image-level connection modules can also be divided into two subcategories.

Without internal alignment. Image-level connection modules are the earliest developed and most widely applied type. Their structure is very simple, typically consisting of one to multiple linear layers (i.e., fully connected layers or multilayer perceptrons) [6], [7], [10]. Although such connection modules can only perform external alignment, they achieve excellent results after joint instruction fine-tuning with the visual encoder module or LLM module. Additionally, due to their simple design and ease of training, these connection modules are widely used in numerous visual understanding MM-LLMs.

With internal alignment. Image connection modules are relatively less common but include representative examples such as the ALBEF architecture, the MED (Mixture of Encoder-Decoder) structure proposed in the BLIP architecture, and the Q-Former module introduced in the BLIP-2 architecture. The architecture of such connection modules is more complex, generally involving two unimodal encoders (one for visual modality and one for language modality), a vision-based language encoder, and a vision-based language decoder. The two unimodal encoders separately accept visual and language inputs, outputting visual tokens and text tokens, respectively. Subsequently, the text tokens are fed into the vision-based language encoder/decoder, while the visual tokens are input into a cross-attention module, producing multimodal encodings or decoding multimodal information to predict subsequent text outputs. During this process, there are three main types of outputs: unimodal visual tokens and text tokens, multimodal encodings, and predicted text outputs. Internal alignment is achieved by designing objective functions based on these three outputs, optimizing the final output of the connection module before integrating it with the LLM module, thereby completing the internal modal alignment of the connection module. Moreover, these three optimization objectives indicate that internally aligned connection modules can unify understanding and generation tasks within image understanding tasks. To further enhance the modal alignment performance, some model architectures adopt a strategy combining internally aligned and non-internally aligned connection modules [3], [7].

Although connection modules without internal alignment are simple and effective, they cannot mitigate the memory usage issues caused by images that are too numerous or have excessively high resolutions (the number of visual tokens output by such modules grows linearly with the increase in image resolution or quantity). Connection modules with internal alignment can employ Perceiver techniques to represent images of various resolutions using a fixed number of token vectors. Typical examples include the Q-Former module in BLIP-2 and the Perceiver Resampler module in Flamingo [4].

b) Video-level connection modules: The video-level connection module primarily handles extracting temporal information from sequential visual data and compressing the size of visual tokens. Some short video understanding models do not design complex video-level connection modules; instead, they simply concatenate the outputs (a series of frame tokens) of image-level connection modules and input them into the LLM module. Such a design inevitably makes the model sensitive to the length of the input video (i.e., the number of frames), with the memory usage of visual tokens increasing linearly with the number of input frames [5].

Notably, if a series of frame tokens are directly compressed and stored using a strategy that does not independently compress each frame's tokens, the model effectively extracts temporal information between frames during the compression process. Thus, analogous to the perception induction techniques used in image-level connection modules, video-level connection modules can use a smaller number of tokens to integrate information from multiple frames. This approach does not treat multi-frame inputs independently and achieves the extraction of temporal information. A typical example is the Video Perceiver [18].

Additionally, 3D convolution techniques, which add a new temporal dimension to traditional 2D convolutions (spatial dimensions), can also compress the size of frame tokens [19]. Since frames are not independently compressed, this method can also extract temporal information.

c) Long-video-level connection modules: Long-video-level connection modules, building on the tasks of video-level connection modules, also need to efficiently compress and store large volumes of video information and quickly retrieve memories based on user instructions. A common approach for storing and retrieving memories is to use a memory bank: dynamically append the latest frame encodings of videos to the memory bank and appropriately "forget" (compress/discard) previous information based on capacity. When retrieving memories, the encoding of user instructions is used to find highly similar tokens in the memory bank, which are then extracted and input into the LLM module [16]. Since this type of connection module requires using language modality information to retrieve visual modality information, it clearly benefits from internal alignment for better retrieval performance.

Additionally, some models further compress the size of the memory bank by not directly appending new frame encodings to the memory bank and discarding older frame encodings. Inspired by perception induction techniques, they use fewer encoding vectors and multiple encoding methods to store long video information. Each encoding method provides a different "perspective" for the downstream LLM module to understand the long video. Unlike memory banks that directly append/concatenate new memories, the encoding vectors in this approach collectively summarize the video's historical information, with individual vectors theoretically lacking independent meaning. Therefore, such memory banks do not require retrieval; instead, all vectors are input into the LLM module when needed to assist its reasoning [9].

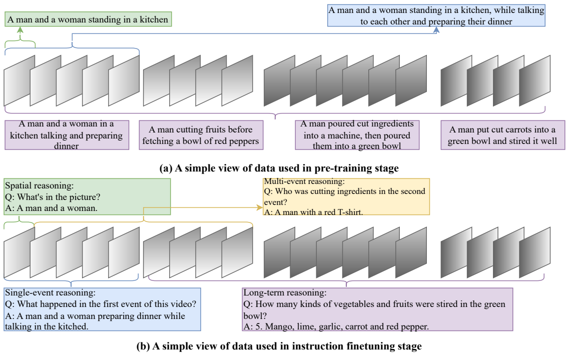
Besides the two types of memory bank-based connection module designs mentioned above, some models attempt to use simple MLP designs for long-video connection modules. In these models, the memory function is embodied in the context input to the LLM, leveraging the LLM's context window to remember historical information. With appropriate training strategies, these models can also achieve good performance [14], [15].

The three designs of long-video-level connection modules described above have all decoupled memory storage from user interaction. Specifically, to respond to user interactions, the model does not need to re-encode the entire video each time but can dynamically extract memories or directly use content from the memory bank/LLM context window. This approach enables faster response times and initially addresses the challenges of online long video understanding.

IV. Improvements on Training Settings

This section primarily discusses the advancements in training methods and dataset construction for visual understanding MM-LLMs. In early model designs, training consisted of only one phase. As the field evolved, two-phase training—pre-training and instruction fine-tuning—became the norm. The former is used for modal alignment, while the latter optimizes the model's instruction-following performance and zero-shot learning capabilities. From the perspective of two-phase training, early single-phase training can be seen as encompassing only the pre-training phase.

Combining some of the challenges proposed in Section II, it becomes evident that the instruction fine-tuning phase further addresses challenges (3) and (4) in image training; certain models' training set construction methods or training approaches also address challenges (1) in image training (these three challenges are also present in the other two types of tasks).


A man and a woman standing in a kitchen
A man and a woman standing in a kitchen, while talking to each other and preparing their dinner
A man and a woman in a kitchen talking and preparing dinner
A man cutting fruits before fetching a bowl of red peppers
A man poured cut ingredients into a machine, then poured them into a green bowl
A man put cut carrots into a green bowl and stired it well
Spatial reasoning:
Q: What's in the picture?
A: A man and a woman.
Multi-event reasoning:
Q: Who was cutting ingredients in the second event?
A: A man with a red T-shirt.
Single-event reasoning:
Q: What happened in the first event of this video?
A: A man and a woman preparing dinner while talking in the kitched.
Long-term reasoning:
Q: How many kinds of vegetables and fruits were stired in the green bowl?
A: 5. Mango, lime, garlic, carrot and red pepper.

(a) A simple view of data used in pre-training stage

(b) A simple view of data used in instruction finetuning stage

Fig. 4. A simple view of training data in training stage (a) pre-training and (b) instruction tuning.

A. Two-stage Training

a) Pre-training: During the pre-training phase, the cross-modal connection module is necessarily trained. Depending on the specific settings of different models, parameters of the visual encoder module and the LLM module may be frozen or unfrozen during sub-phases of pre-training. In image understanding tasks, the pre-training phase primarily uses a large number of simple image-short caption pairs to train the model [6], [7], [17]. During the early single-phase training process, since this was the only form of training, the quality of the dataset became particularly important. Given that manually filtering noise from large datasets is impractical, self-supervised or model distillation approaches were developed [1], [2] to optimize the quality of the pre-training dataset (see IV-B for details).

In short video understanding tasks, the pre-training phase still uses image-short caption pairs to train the model but also introduces short video-short text pairs for training. Long videos can be considered as concatenated segments of short videos; thus, the pre-training data for long videos takes the form of "a long video - multiple text segments with timestamps." Additionally, if the "multiple text segments" within the pre-training data for long videos contain references to other video segments, then pre-training on such data effectively achieves event-level visual-language modal alignment from an alignment perspective. This allows the LLM to actively seek connections between different events during inference. Figure 4 (a) provides a visual illustration of the training data format.

b) Instruction finetuning: During the instruction fine-tuning phase, the cross-modal connection module and the LLM module are necessarily trained; some models may unfreeze all parameters. In image understanding tasks, the data used during this phase includes high-quality image-short caption pairs and image-long description pairs. Additionally, there may be image-text pairs adapted for other downstream tasks, such as image-question-answer pairs, image-bounding box coordinates-object name tuples, etc. [6].

In short video understanding tasks, the data used during the instruction fine-tuning phase is similar to that used in image understanding tasks. However, long video understanding involves cross-event reasoning (long-term reasoning) and long-term memory. Therefore, the data used for instruction fine-tuning of long video understanding models also includes cross-event descriptions and overall video summaries. Figure 4 (b) provides a visual illustration of the training data format.

B. Some Other Effective Training Practice

In addition to the basic pre-training and instruction fine-tuning, some models also design additional training methods based on their specific needs. These training methods are primarily aimed at enhancing the robustness of the model, ensuring that it can achieve good performance even when trained on noisy data.

a) Momentum distillation: This method was introduced as a model training approach in the ALBEF architecture. Since this architecture does not undergo instruction fine-tuning, its performance can be affected by noisy data in the training dataset. Therefore, the ALBEF proposes this method to enhance the training effectiveness of the model. This method can be viewed as establishing a queue of historical snapshots of the model: after each epoch or batch is trained, a new snapshot of the model is saved and added to the queue. This queue can be referred to as the model's "momentum queue," reflecting the evolution of the model parameters over time. During training, the model does not solely use the ground truth as the training target but also uses the outputs from models in the queue as additional targets. The actual training target is thus a weighted average of the ground truth and the outputs from historical snapshots. The work refers to these targets derived from historical snapshots as "pseudo targets."

Since these pseudo targets are generated by models that have not fully fitted (noisy) training data, they may be more accurate than the ground truth for instances where the ground truth is incorrect. Specifically, if the ground truth is erroneous, the pseudo targets could represent predictions based on other correctly labeled training data, making them potentially more reliable. Ultimately, this approach demonstrates that using pseudo targets can lead to better model performance when training on noisy datasets.

The process of saving model snapshots to the momentum queue is akin to an online self-distillation mechanism, where the model continuously distills its knowledge into models with the same architecture during training. Hence, this method is referred to as momentum distillation [1].

b) CapFilt (captioning and filtering): This method was introduced as a model training approach in the BLIP architecture. Similar to ALBEF, this architecture does not undergo instruction fine-tuning. The method leverages the Mixture of Encoder-Decoder (MED) structure within BLIP by extracting the multimodal text encoder-decoder during training. These components are used separately: one as a classifier to determine whether captions in image-caption pairs accurately describe the corresponding images, and the other as a generative model to produce image captions.

In each training iteration, updated data is used to re-pretrain the MED structure of BLIP. After training, the MED structure extracts the aforementioned classifier and generative model, which are then used to filter noisy data from the original dataset, forming updated data for the next training iteration. This cycle repeats until the final model is obtained. The original work refers to the classifier as the "filter" and the generative model as the "captioner," collectively naming this approach CapFilt.

Additionally, the work notes that this process can also be viewed as a form of model distillation. In this context, the captioner and filter alternate roles as teacher and student models, distilling their knowledge into the captions or filtered data, allowing the counterpart to learn from it.

V. Performance Benckmarking

A. Performance Analysis

Table II shows the performance of several models as indicated in their respective works. To highlight more apparent performance comparisons, the table only includes evaluation frameworks for visual question answering that were commonly tested by these models, as well as data on the quality scores of their generated content rated by GPT. Regarding the scoring items by GPT, CI stands for the correctness of information (Correctness of Information), DO for the level of detail in the generated information (Detail Orientation), CU for the model's understanding of the question-answer context (Context Understanding), TU for the model's understanding of temporal information (Temporal Understanding), and CO for the coherence of the generated content (Consistency).

TABLE II
Performance of some visual understanding models

Model	MSVD-QA	ActivityNet-QA	CI	DO	CU	TU	CO	Average
InstructBLIP	41.8	-	-	-	-	-	-	-
MovieChat	75.2	45.7	2.76	2.93	3.01	2.24	2.42	2.67
LLaVA-NeXT	-	53.2	3.39	3.29	3.92	2.60	3.12	3.26
MiniGPT-4-Video	72.9	45.9	2.93	2.97	3.45	2.47	2.60	2.88
LLaVA-OneVision	-	56.6	-	-	-	-	-	3.49

Statistical data are from original work. Zoom in for a better view of the table.

By analyzing the data in the table, we can derive some challenges related to long video understanding:

- The more frames a video contains, the higher the complexity of the information introduced, and the greater the demand for the model's reasoning capabilities.

- Short video understanding models that perform well on videos ranging from tens of seconds to about one minute tend to perform poorly on longer videos that span several minutes. In contrast, long video understanding models often exhibit good performance on both long and short videos. This may be attributed to the ability of long video understanding models to capture spatiotemporal detail information present in short videos as well.

B. Performance Optimization for Local Deployment

Research on performance optimization for local deployment has a representative model known as MiniCPM-V [8]. This model optimizes its performance through five primary methods: (1) *Compression of parameter quantity.* In terms of architectural design, it adopts an Perceiver strategy to capture information from the visual modality using a fixed number of tokens; additionally, it employs parameter quantization techniques to reduce the bit size occupied by each value in memory. (2) *Optimization of memory allocation.* The model loads parameters of the visual encoder and LLM in batches and sequentially rather than attempting to load both at once. This approach can decrease the frequency of memory swapping. (3) *Configuration file optimization.* This method is tailored specifically for the deployment of `llama.cpp`. By utilizing an automated parameter-setting learning method, more reasonable configurations can be generated during deployment; for instance, CPUs of the device can be allocated more efficiently according to the computational needs of the model. (4) *Local compilation.* It was found that compiling the model on the target device before running leads to faster execution. It is speculated that this is because post-compilation assembly code usage aligns better with the architecture of the target device. (5) *Utilizing NPU acceleration on supported devices.*

Following the aforementioned performance optimization strategies, the MiniCPM-V model was successfully deployed on both the Xiaomi 14 Pro and vivo X100 Pro mobile devices. This deployment achieved an average encoding latency of 10 seconds and an average decoding throughput of 6 tokens per second (with the human reading speed being approximately 5 tokens per second), significantly enhancing the performance of visual understanding MM-LLM on local mobile devices post-deployment.

VI. Future Research Directions

Regarding the future research directions for visual understanding MM-LLM, the primary focus is on enhancing the performance of online long-video understanding. Nevertheless, the following potential research directions/strategies could serve all three types of visual understanding tasks: *(1) Creating more training resources; (2) Designing more challenging evaluation frameworks; (3) Designing more effective and robust model architectures; (4) Developing richer application scenarios to drive model advancements through practical use.*

VII. Conclusion

This survey provides an overview of the development trajectories of three types of visual understanding MM-LLM (image understanding, short video understanding, and long video understanding). Initially, it analyzes the objectives and challenges of these three types of visual understanding tasks, such as dataset quality issues, the unification of generation and understanding tasks, the quality of generated text, and the memory capabilities for visual content. Subsequently, it introduces the architectural design patterns of the three types of models and some typical architectures, analyzing how these architectures address the various task challenges. The paper then describes the basic training paradigms and two training methods aimed at enhancing model stability. Finally, it briefly showcases the performance of several typical models on question-answering tasks and provides a concise introduction to performance optimization strategies for local deployment on mobile devices. Based on the above analysis, this paper also proposes some future research directions for visual understanding MM-LLM.

Acknowledgment

Thanks CSUC for providing this chance to simulate a thorough progress of paper submission.

References

[1] J. Li, R. R. Selvaraju, A. D. Gotmare, S. Joty, C. Xiong, and S. Hoi, “Align before Fuse: Vision and Language Representation Learning with Momentum Distillation,” Oct. 2021, arXiv:2107.07651 [cs]. [Online]. Available: http://arxiv.org/abs/2107.07651

[2] J. Li, D. Li, C. Xiong, and S. Hoi, “BLIP: Bootstrapping Language-Image Pre-training for Unified Vision-Language Understanding and Generation,” Feb. 2022, arXiv:2201.12086 [cs]. [Online]. Available: http://arxiv.org/abs/2201.12086

[3] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models,” Jun. 2023, arXiv:2301.12597 [cs]. [Online]. Available: http://arxiv.org/abs/2301.12597

[4] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, “Flamingo: a Visual Language Model for Few-Shot Learning,” Nov. 2022, arXiv:2204.14198 [cs]. [Online]. Available: http://arxiv.org/abs/2204.14198

[5] W. Dai, J. Li, D. Li, A. M. H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, “InstructBLIP: Towards General-purpose Vision-Language Models with Instruction Tuning,” Jun. 2023, arXiv:2305.06500 [cs]. [Online]. Available: http://arxiv.org/abs/2305.06500

[6] H. Liu, C. Li, Q. Wu, and Y. J. Lee, “Visual Instruction Tuning,” Dec. 2023, arXiv:2304.08485 [cs]. [Online]. Available: http://arxiv.org/abs/2304.08485

[7] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, “MiniGPT-4: Enhancing Vision-Language Understanding with Advanced Large Language Models,” Oct. 2023, arXiv:2304.10592 [cs]. [Online]. Available: http://arxiv.org/abs/2304.10592

[8] Y. Yao, T. Yu, A. Zhang, C. Wang, J. Cui, H. Zhu, T. Cai, H. Li, W. Zhao, Z. He, Q. Chen, H. Zhou, Z. Zou, H. Zhang, S. Hu, Z. Zheng, J. Zhou, J. Cai, X. Han, G. Zeng, D. Li, Z. Liu, and M. Sun, “MiniCPM-V: A GPT-4V Level MLLM on Your Phone,” Aug. 2024, arXiv:2408.01800 [cs]. [Online]. Available: http://arxiv.org/abs/2408.01800

[9] H. Zhang, Y. Wang, Y. Tang, Y. Liu, J. Feng, J. Dai, and X. Jin, “Flash-VStream: Memory-Based Real-Time Understanding for Long Video Streams,” Jun. 2024, arXiv:2406.08085 [cs]. [Online]. Available: http://arxiv.org/abs/2406.08085

[10] Y. Zhang, “LLaVA-NeXT: A Strong Zero-shot Video Understanding Model,” Apr. 2024. [Online]. Available: https://llava-vl.github.io/blog/2024-04-30-llava-next-video/

[11] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu, and C. Li, “LLaVA-OneVision: Easy Visual Task Transfer,” Oct. 2024, arXiv:2408.03326 [cs]. [Online]. Available: http://arxiv.org/abs/2408.03326

[12] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang, Y. Lu, J.-N. Hwang, and G. Wang, “MovieChat: From Dense Token to Sparse Memory for Long Video Understanding,” Mar. 2024, arXiv:2307.16449 [cs]. [Online]. Available: http://arxiv.org/abs/2307.16449

[13] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, “VideoChat: Chat-Centric Video Understanding,” Jan. 2024, arXiv:2305.06355 [cs]. [Online]. Available: http://arxiv.org/abs/2305.06355

[14] S. Wu, J. Chen, K. Q. Lin, Q. Wang, Y. Gao, Q. Xu, T. Xu, Y. Hu, E. Chen, and M. Z. Shou, “VideoLLM-MoD: Efficient Video-Language Streaming with Mixture-of-Depths Vision Computation,” Aug. 2024, arXiv:2408.16730 [cs]. [Online]. Available: http://arxiv.org/abs/2408.16730

[15] J. Chen, Z. Lv, S. Wu, K. Q. Lin, C. Song, D. Gao, J.-W. Liu, Z. Gao, D. Mao, and M. Z. Shou, “VideoLLM-online: Online Video Large Language Model for Streaming Video,” Jun. 2024, arXiv:2406.11816 [cs]. [Online]. Available: http://arxiv.org/abs/2406.11816

[16] R. Qian, X. Dong, P. Zhang, Y. Zang, S. Ding, D. Lin, and J. Wang, “Streaming Long Video Understanding with Large Language Models,” May 2024, arXiv:2405.16009 [cs]. [Online]. Available: http://arxiv.org/abs/2405.16009

[17] H. Zou, T. Luo, G. Xie, Victor, Zhang, F. Lv, G. Wang, J. Chen, Z. Wang, H. Zhang, and H. Zhang, “From Seconds to Hours: Reviewing MultiModal Large Language Models on Comprehensive Long Video Understanding,” Sep. 2024, arXiv:2409.18938 [cs]. [Online]. Available: http://arxiv.org/abs/2409.18938

[18] Y. Wang, R. Zhang, H. Wang, U. Bhattacharya, Y. Fu, and G. Wu, “Vaquita: Enhancing alignment in llm-assisted video understanding,” 2023. [Online]. Available: https://arxiv.org/abs/2312.02310

[19] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, “Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution,” 2024. [Online]. Available: https://arxiv.org/abs/2409.12191