How Can Objects Help Video-Language Understanding?

Zitian Tang[1] Shijie Wang[1] Junho Cho[2] Jaewook Yoo[2] Chen Sun[1]
[1]Brown University [2]Samsung Electronics

https://brown-palm.github.io/ObjectMLLM

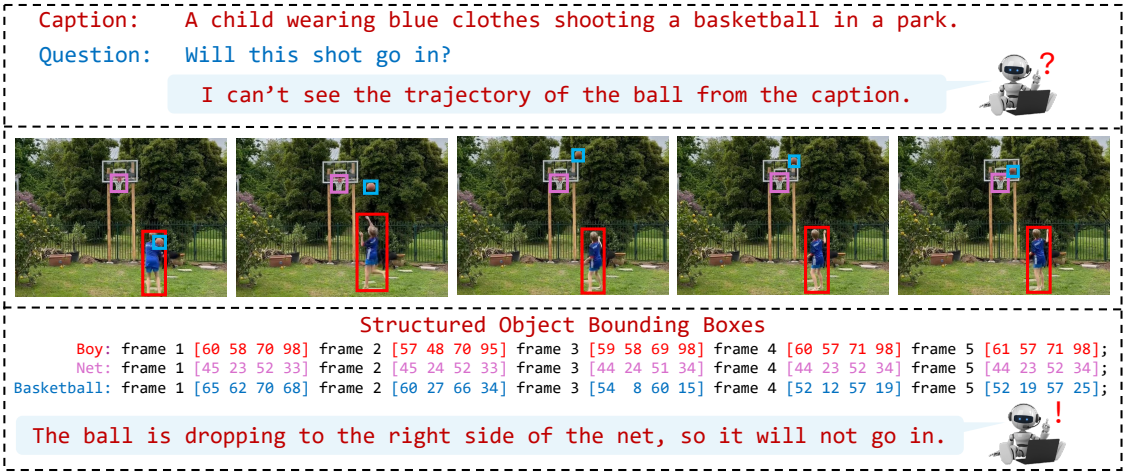


Figure 1. Socratic Models [36, 46, 48] perceive the world from the lens of natural language descriptions, which may miss important spatiotemporal information (*top*). Multimodal large language models (MLLMs), on the other hand, can integrate visual information via distributed embeddings, but typically require large-scale instruction tuning datasets for *adaptation*. We investigate how explicit, continuous object representations (*e.g.* box coordinates from object detectors) can help video-language understanding (*bottom*).

Abstract

Do we still need to represent objects explicitly in multimodal large language models (MLLMs)? To one extreme, pre-trained encoders convert images into visual tokens, with which objects and spatiotemporal relationships may be implicitly modeled. To the other extreme, image captions by themselves provide strong empirical performances for understanding tasks, despite missing fine-grained spatiotemporal information. To answer this question, we introduce ObjectMLLM, a framework capable of leveraging arbitrary computer vision algorithm to extract and integrate structured visual representation. Through extensive evaluations on six video question answering benchmarks, we confirm that explicit integration of object-centric representation remains necessary. Surprisingly, we observe that the simple approach of quantizing the continuous, structured object information and representing them as plain text performs the best, offering a data-efficient approach to integrate other visual perception modules into MLLM design. Our code and models are released at https://github.com/brown-palm/ObjectMLLM.

1. Introduction

What makes a good representation for video-language understanding? In the era of multimodal large language models (MLLMs), anything that can be *tokenized* has the potential to serve as a valid representation. Along the spectrum are two extremes: those that project arbitrary distributed representations to the input space of a pre-trained large language model via instruction tuning [7, 22], and those that model the visual world as interpretable concepts [40] and captions [36, 48], which can be directly consumed by LLMs via Socratic Methods [46]. It is open to debate whether either approach can effectively capture and convey the complexity of the visual world to an LLM "reasoner". As illustrated in Figure 1, video captions tend to ignore detailed information that captures the spatiotemporal object configurations. Meanwhile, despite inductive biases to guide MLLM encoders to be spatial aware [34], integrating visual information such as objects and their locations into LLMs remains a challenging endeavor [35].

We hypothesize that *explicit* object-centric recognition and modeling, supported by the rich literature from the computer vision community, remains essential to the suc-

cess of MLLMs. We then seek to answer the question, *how can objects help video-language understanding in MLLMs*, from two perspectives: representation and adaptation. Motivated by the effectiveness of caption-based representation for video understanding [26, 36, 39, 48], we hypothesize that there is a natural trade-off between the expressiveness of a visual representation, and the easiness for it to be adapted into a pre-trained LLM: A distributed representation is the most expressive, but needs larger amount of instruction tuning data for it to be integrated into LLMs [14, 22]. "Symbolic" representations that are language-based (*e.g.*, rendering quantized object coordinates as plain text), though less expressive, may be easier to use as they can be represented with the existing vocabulary of an LLM. We further hypothesize that symbolic object representations are more expressive in videos when they depict the trajectories of a moving object or its key points, as Johansson's biological motion perception experiment [11] showed that humans can successfully associate a collection of dots with human motions as soon as the dots start moving (*e.g.*, the trajectory of the basketball in Figure 1).

To validate these hypotheses, we introduce **ObjectMLLM**, a framework capable of leveraging arbitrary computer vision algorithm (*e.g.*, an object detector or human pose estimator) to extract and integrate structured visual representation into multimodal LLMs. With ObjectMLLM, we investigate the trade-off of designing object-centric representations, either by learning an embedding projector, or with the symbolic object representation. The former approach generates a distributed representation projected into the input space of an LLM, from a vectorized representation of object bounding boxes. The latter approach directly renders bounding boxes as *strings*, which are then tokenized accordingly. For both approaches, we apply parameter efficient fine-tuning to adapt the weights of the pre-trained LLMs together towards the target tasks. We observe that as hypothesized, while embedding projector leads to more compact object representations, it is less data-efficient compared to symbolic object representation, consistently yielding lower performance when fine-tuned for the same number of iterations. We then conduct thorough evaluations on six video QA benchmarks, where we observe that symbolic object representation consistently improves the model performance, especially on tasks that require spatiotemporal understanding (*e.g.*, PerceptionTest [29]).

In summary, our contributions are three-fold:

- We propose ObjectMLLM, a multimodal video understanding framework that seamlessly incorporates object spatial information from computer vision algorithms in multimodal LLMs.
- We study two bounding box adapters and show that a language-based representation is more performant and data-efficient than latent embedding projectors, indicating pre-trained LLMs may already be *spatially aware*.
- Our evaluation on video question answering benchmarks demonstrates the significance of ObjectMLLM when applied to both pre-trained LLMs and multimodal LLMs, and that the benefits generalize to other structured visual representation, such as human joint coordinates [5].

2. Related Works

2.1. Video Large Language Models

Large language models (LLMs) have recently shown remarkable progress in understanding and generating text across various domains. Their success has inspired the development of Video Large Language Models (Video-LLMs) [4, 10, 12, 13, 17, 36, 43, 45, 49, 53], which integrate videos into the language modeling framework and are widely applied in tasks such as video captioning and question answering. Most Video-LLMs consist of three components: a pre-trained visual encoder, an adaptation model, and an LLM backbone. One of the primary challenges for Video-LLMs, compared to Image-LLMs, is how to effectively and efficiently representing the rich contextual information in videos. Many Video-LLMs [4, 12, 13, 43, 49] employ pre-trained image encoders [27, 30, 47] to extract features from sampled frames individually, concatenating them to form video representations. Other approaches [20, 24, 38] utilize a dedicated video encoder to capture spatial-temporal features across the entire video. Additionally, Chat-UniVi [10] combines image and video encoders and implements spatial merging to reduce the number of video tokens for greater efficiency. Beyond video features, some models, such as Vamos [36], VideoChat [17], and LifelongMemory [39], flexibly incorporate action labels and video captions as inputs to represent videos from multiple perspectives. In this work, we investigate the influence of object-centric information in Video-LLMs and explore methods to incorporate structured representations, such as objects represented by sequences of bounding boxes and class labels, into Video-LLMs.

2.2. Modality adaptation in MLLMs

Modality adaptation in multimodal large language models (MLLMs) is critical when extending large language models to handle diverse inputs, including images, audio, and video. One intuitive approach for non-text modalities is to convert them into textual representations, such as captions [1, 36, 46, 48] or action labels [53]. Such textual representations provide good interpretability and data efficiency by leveraging the extensive language prior knowledge embedded in LLMs. Through this method, domain-specific expert models, such as video captioning and action recognition models, act as adaptation modules within the multimodal LLM framework. Another common approach

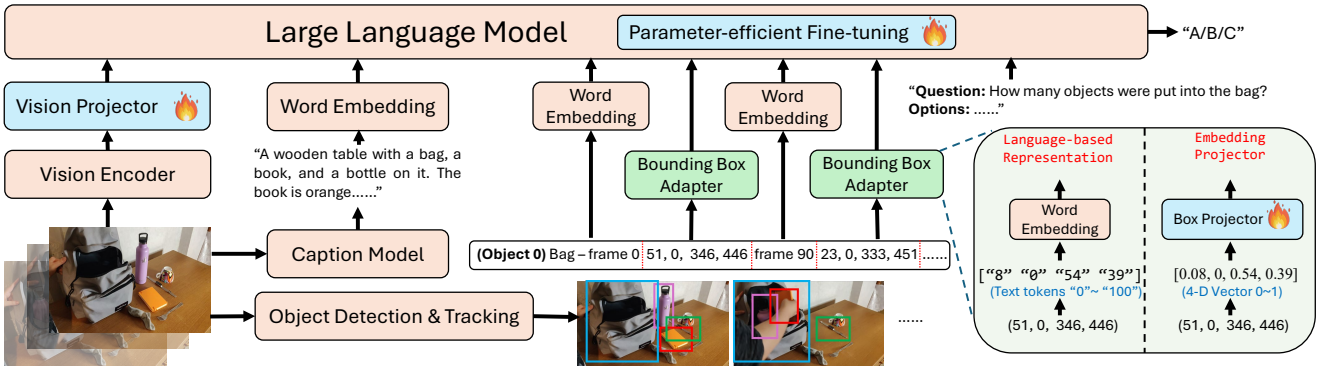


Figure 2. Pipeline of ObjectMLLM. The possible model inputs are visual embeddings, video frame captions, and object bounding boxes. ObjectMLLM encodes object coordinates with a language-based ("symbolic") representation, or with an embedding projector. The former represents the bounding boxes as plain text, while the latter maps the vectorized coordinates into the input space of the LLM.

for aligning non-text modalities to the text space is multimodal fine-tuning, which directly uses continuous embeddings and trains a projection module for adaptation. Two types of projection modules are frequently employed: MLP projectors and attention-based projectors [14]. For instance, LLaVA [22] utilizes a lightweight linear layer to project vision embeddings to input token for the LLM through multi-stage training on large-scale datasets, while LLaVA1.5 [23] further improves by adopting a two-layer MLP projector. Recent studies [21, 25] suggest that the specific structure of the projector exerts marginal influence on MLLM performance. Compared with textual representations, multimodal fine-tuning directly utilize continuous embeddings from encoders but generally requires substantial multi-stage training on large-scale multimodal datasets. In this work, we systematically compare various approaches for adapting structured object representations within Video-LLMs and evaluate the impact of different modality representations on video question answering tasks.

2.3. Objects in MLLMs

A prominent approach to integrate objects into MLLMs involves leveraging object detectors to extract region-based features for downstream tasks. OSCAR [19] introduces an object-aware pre-training paradigm that aligns object tags with textual data, enhancing contextual understanding. VinVL [50] builds upon OSCAR by employing a stronger object detector to extract more accurate region features. CoVLM [16] advances this direction by explicitly composing visual entities and relationships within text through the use of communication tokens. These tokens facilitate dynamic interaction between the visual detection system and the language system. When communication tokens are generated by the LLM, detection models respond by generating regions-of-interest (ROIs), which are then fed back into the LLM to improve language generation. Another line of work focuses on grounding VLMs, which are capable of localizing objects and predict bounding boxes or masks based on language references. Models such as Shikra [2], Kosmos-2 [28], and GLaMM [31] are trained on large scale grounding and localization dataset, and are designed specifically for these tasks. In these models, structured localization information such as bounding boxes is usually encoded and projected to align with LLMs and a decoding head is trained to make prediction. ObjectMLLM aligns with the first approach by investigating how object-centric information can enhance video understanding in multimodal LLMs.

3. Method

We aim to complement Multimodal Large Language Models (MLLMs) with structured visual information extracted by off-the-shelf computer vision algorithms. We focus on object-centric representation, which captures object location and motion. As Figure 1 shows, the object-centric representation encodes the position and movements of individual objects via 2D bounding boxes, object labels, and timestamps. We expect that enabling MLLMs to explicitly utilize object-centric representation can enhance their spatiotemporal understanding capability. For this purpose, we investigate whether and how we can boost video understanding by leveraging object bounding boxes.

Our investigation focuses on two perspectives: The first is the final video question answering accuracy, measuring how useful an object-centric representation is; the second is the amount of fine-tuning data required for a pre-trained LLM or multimodal LLM to utilize the object information properly, which we refer to as *data efficiency*. Both have practical motivations: as it is desirable to explore the explicit integration of different object detectors and trackers, or even computer vision models for pose estimation, panoptic segmentation, into LLMs – without the need to always perform large-scale instruction tuning. We are also interested in a more philosophical discussion on to what degree

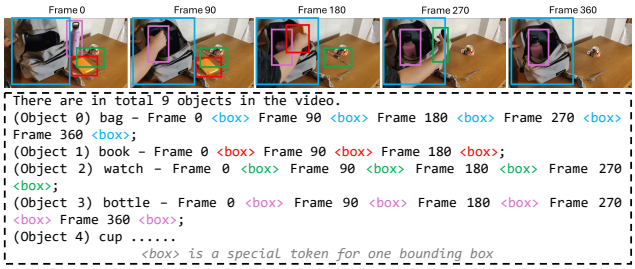


Figure 3. Template to format the object bounding boxes. The timestamps when an object is visible are listed, each of which is followed by tokens that describe the bounding box coordinates.

LLMs pre-trained on language data are *spatially aware*, and whether they can be tuned to perform spatial understanding in a data-efficient manner.

3.1. ObjectMLLM

We propose ObjectMLLM, a multimodal framework that integrates distributed visual embeddings, video frame captions, and object bounding boxes into one MLLM, as illustrated in Figure 2. The utilization of video frame embeddings and captions is in line with caption-enhanced MLLMs, *e.g.*, Vamos [36]. Specifically, we uniformly sample a fixed number of frames from a video, and employ an off-the-self image feature encoder and captioning model to extract visual embeddings and captions, respectively. The generated captions are directly fed to the LLM as inputs, while the visual embeddings are mapped into the word embedding space of the LLM by a vision projector, typically implemented as a lightweight neural network.

With off-the-shelf object detection and tracking models, we capture object bounding boxes from the video. Following the template in Figure 3, we list the timestamps when each object is visible, and append a special bounding box token after each timestamp. The textual part, including the object labels and timestamps, are directly tokenized and converted to word embeddings by the LLM. Each bounding box, which is represented by four continuous numbers, can either be rendered as plain text and tokenized as text tokens, or passed to a projector to produce an embedding in the LLM input space. The bounding box tokens are interleaved with the text tokens corresponding to the object labels and timestamps. In Section 3.3, we discuss the strengths and limitations of each bounding box representation.

3.2. Object detection and tracking

To obtain object-centric representations, we need the semantic labels and tracked bounding boxes of the objects in a video. The computer vision community has developed powerful, standalone models for object detection and tracking. We choose YOLO-World [3], an open-vocabulary object detector, to detect objects in a given video frame, and use SAM 2 [32] to track the detected objects across the video.

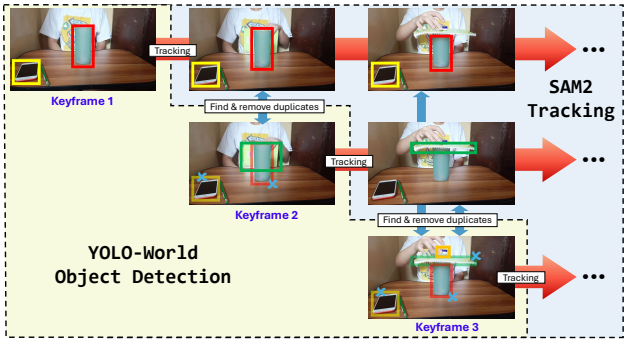


Figure 4. Workflow of object detection and tracking. We iteratively detect objects in uniformly sampled keyframes, and create new tracks for new objects not associated with any existing track.

The workflow is illustrated in Figure 4. For each video benchmark, we consider all the object categories in its training set as the vocabulary of YOLO-World, which detects objects from video frames that are uniformly sampled. For each subsequent frame after the initial one, we deploy both SAM 2 to track objects already detected in the previous frames, and also YOLO-World to detect all objects present in the current frame. We calculate the IoU between the detected objects and the objects from existing tracks. Detected objects with an IoU greater than 0.5 are removed as duplicates, and the remaining ones are used to create new tracks.

To mitigate the distribution shift compared to its pre-training data, YOLO-World is fine-tuned on the training set of each benchmark, respectively, before usage. The pre-trained SAM 2 is kept frozen for all benchmarks.

3.3. Object-centric representation

As illustrated in Figure 2, we consider two implementations of bounding box adapters. The language-based adapter turns continuous boxes into interpretable symbols that represent the spatial locations, and the embedding projector learns to project the vectorized bounding box coordinates of arbitrary object into the LLM input space. Intuitively, the language-based representation could be more data-efficient since it directly reuses the tokenizer and word embeddings from a pre-trained (multimodal) LLM.

Language-based representation: We perform normalization and quantization to map continuous bounding box coordinates into discrete integers. This conversion is lossy but uses fewer tokens than float numbers. We take values in the range of $[0, 100]$. Each value is tokenized and embedded with the existing LLM tokenizer. A drawback of this method is that it uses multiple tokens to represent one bounding box, requiring long context windows of the LLM, and is computationally more expensive.

Embedding projector: A commonly-adopted approach for a pre-trained MLLM to incorporate non-textual information is to train a embedding projector that maps the vector-

ized representation from a new data "modality" to the input space of the LLM. For example, LLaVA [22] trains a linear layer as the projector of image CLIP embeddings. ObjectMLLM takes a bounding box as a 4-dimensional embedding with continuous values, normalizes each dimension to floats in $[0, 1]$, and trains a linear layer as the embedding projector to produce vectorized representations with the same number of the dimensions as the LLM word embeddings.

3.4. Fine-tuning strategy

ObjectMLLM can be fine-tuned from either a pre-trained LLM or a multimodal LLM. We perform parameter-efficient fine-tuning on the LLM backbone, and jointly train the vision projector and the embedding projector of the bounding box adapter with the LLM backbone. All other parameters are kept frozen.

When starting from pre-trained LLMs, we adopt a modality-by-modality training strategy used by VideoLLaMA2 [4] to gradually incorporate incoming modalities. For example, to develop a model that incorporates both the caption and the bounding box modality, we first train the model in a caption-only setting. After the model learns to utilize video frame captions, we further fine-tune it with inputs containing both captions and boxes. The modality incorporation order we use is frame captions, bounding boxes, and visual embeddings across all the benchmarks.

4. Experiments

We first compare the two bounding box adapters in ObjectMLLM. We then choose the best-performing adapter, and investigate the effectiveness of integrating different input modalities. Finally, we study if object-centric representation can be used to improve the performance of MLLMs that are pre-trained to utilize visual embedding without explicit object-centric representations.

4.1. Benchmarks

To evaluate a model's understanding about object bounding boxes, we need benchmarks where spatial and temporal object information is essential to the questions. CLEVRER [44] is a synthetic video dataset focusing on object motion and collision. However, CLEVRER contains open-ended questions, making the performance measurement difficult. MVBench [18] converts some of the CLEVRER [44] questions into multi-choice questions. We use this part of data and name it CLEVRER-MC. To train our models on CLEVRER, we use the CLEVRER-sourced part of VideoChat2-IT [18]. It is also multi-choice questions but may have different question types from CLEVRER-MC.

Besides, we also evaluate the models on real-world video benchmarks – Perception Test [29], STAR [41], NExT-QA [42], and IntentQA [15]. While some questions in these benchmarks are related to spatiotemporal object motion, there are also questions focusing on causal reasoning. Evaluation on these benchmarks can reveal the scenarios where the object-centric representation can make a difference.

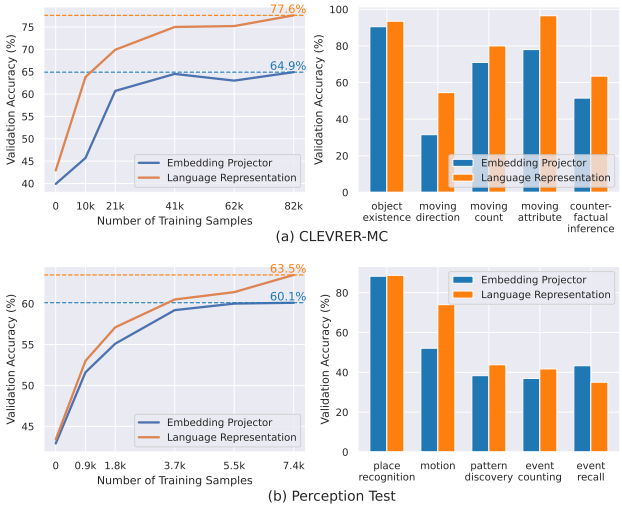


Figure 5. Performance of the box adapters under various training data amounts (left) and accuracy breakdown by question types (right). Only a subset of the question types in Perception Test are listed here. The language-based representation consistently outperforms the embedding projector with different numbers of training samples on both CLEVRER-MC and Perception Test, showing its effectiveness and data efficiency. In the breakdown, the language-based representation outperforms the embedding projector on motion-related questions by large margins.

4.2. Implementation

When starting from a pre-trained LLM to build ObjectMLLM, we follow Vamos [36] and use LLaMA3-8B [6] as the backbone and fine-tune it with LLaMA-Adapter [51]. The vision projector and box projector in the bounding box adapter are linear layers. The distributed visual embedding is extracted by CLIP ViT-L/14 [30] on 10 uniformly sampled frames per video. The embedding projector weights are all initialized to zero, which we find to lead to better performance than the default random initialization in PyTorch. Moreover, we use LLaVA-1.5-13B [23] to generate captions for 6 uniformly sampled frames from each video.

When starting from a pre-trained *multimodal* LLM, we use VideoLLaMA2-7B [4], which is pre-trained on 100M video-language data. It includes CLIP ViT-L/14 [30] as vision encoder, Spatial-Temporal Convolution as vision projector, and Mistral-7B-Instruct [9] as LLM backbone. We fine-tune it with LoRA [8] in our experiments.

To keep the context length under control, we uniformly sample the object bounding boxes at a lower frame rate such that the language-based representation of all the boxes in a video contains fewer than $1,000$ tokens. As different videos have different lengths and numbers of objects, the down-sampling rate varies from video to video.

Video	Caption	Box	CLEVRER-MC	Perception Test	STAR	NExT-QA	IntentQA
✓			40.3	59.6	59.7	70.7	68.2
	✓		47.8	62.4	60.1	**76.6**	**75.7**
		✓	**77.6**	63.5	59.1	63.7	66.2
✓		✓	77.1	62.7	59.3	71.8	71.7
	✓	✓	75.5	**65.7**	**64.4**	**76.6**	75.6
✓	✓	✓	75.4	63.9	62.9	76.2	75.0

Table 1. Accuracy under different combinations of modalities. Using object bounding boxes improves the performance on CLEVRER-MC, Perception Test, and STAR by large margins. Caption remains the most effective modality on NExT-QA and IntentQA.

Video	Caption	Box	OE	MD	MC	MA	CI	All
✓			51.0	21.0	44.5	37.0	48.0	40.3
	✓		62.5	26.5	50.5	50.0	49.5	47.8
		✓	**93.5**	**54.5**	80.0	96.5	**63.5**	**77.6**
	✓	✓	92.5	51.0	79.0	**97.0**	58.0	75.5
✓	✓	✓	92.0	47.5	**81.0**	95.5	61.0	75.4

Table 2. Accuracy of different question types on CLEVRER. While the bounding boxes boost the performance across all the question types, it is more significant on OE, MC, and MA than on others. OE: object existence; MD: moving direction; MC: moving count; MA: moving attribute; CI: conterfactual inference.

The hyperparameters for fine-tuning, bounding box downsampling rates, and implementation details of object detection and tracking are in Appendix A.

4.3. Comparison of adaptation methods

We first compare the two adapters on object-centric representations. For this purpose, we do not use the visual embeddings and captions, and train the model only with the object bounding boxes as input. We focus on the CLEVRER-MC and Perception Test benchmarks, as we empirically observe that they were designed to contain questions more closely related to spatiotemporal object configurations.

In Figure 5 (left), we evaluate the two adapters with various portions of the training data. With the full training data, the language-based representation outperforms the embedding projector across both benchmarks (77.6% *vs.* 64.9% on CLEVRER-MC and 63.5% *vs.* 60.1% on Perception Test). More importantly, the language-based representation can outperform the embedding projector with *any amount of data*. Notably, with only one-eighth (10k) of the training data on CLEVRER-MC, the fine-tuned model is able to utilize bounding boxes from language-based representation and achieves an accuracy of 63.8%, but the performance with the embedding projector remains low (44.5%). Although the embedding projector can keep the continuity of the bounding box coordinates, our experiments show that the LLM backbone struggles to understand the resulting box embeddings when limited fine-tuning data is used. Reusing the existing LLM vocabulary, which is done by the language-based representation, lead to effective and data-efficient understanding of the bounding boxes.

Figure 5 (right) shows the accuracy breakdown by question types. While the performances of the two adapters are comparable on some types of question, the language-based representation shows great superiority on motion-related questions. This phenomenon in motion questions happens to be consistent with Johansson's biological motion perception experiment [11] that humans can associate a collection of moving dots with human motions.

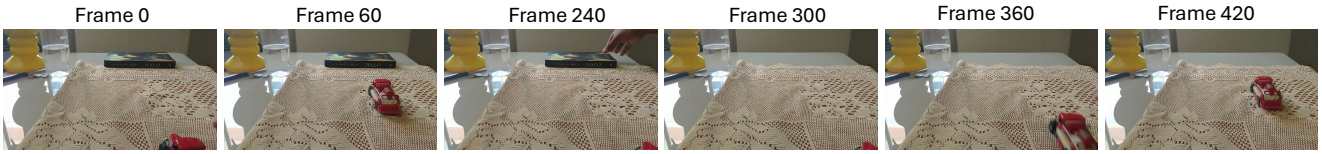
4.4. Influence of each modality

In Section 4.3, the language-based representation is proved to be a more effective bounding box adapter. In this section, we train ObjectMLLM with the language-based box adapter and incorporate visual embeddings, video frame captions, and object bounding boxes in one model. We also ablate the combinations of modalities to break down their contributions to performance. The results are shown in Table 1.

On CLEVRER-MC and Perception Test, the bounding-box-only model outperforms the video-only and caption-only models. And the caption-and-box model outperforms the caption-only model by a large margin on STAR. This indicates the importance of object-centric information on these benchmarks. Adding boxes to video-only baseline leads to consistent improvements, indicating that the visual embeddings alone are not sufficient to encode objects.

The most significant improvement achieved by bounding boxes is on CLEVRER-MC, whose questions focus on object motion and collision. Our qualitative results in Appendix E show that our model can easily identify moving objects from the bounding boxes, which is difficult to determine from the frame captions. We further break down the accuracy of different question types on CLEVRER-MC in Table 2. We find that the improvement on object existence, moving count, and moving attribute is large, but is less significant for moving direction and counterfactual inference. While counterfactual inference requires high-level reasoning, the moving direction of an object should be easily inferred from its bounding boxes. However, we find that the training data we use does not include questions about direction. This highlights that the learned understanding capability on symbolic representation cannot be perfectly generalized to all the tasks that are not involved during training.

The accuracy breakdown on Perception Test in Figure 7 shows substantial improvement on motion questions. The



Frame Captions

Frame 0: A dining table with a book and a glass of water placed on it. The book is positioned towards the left side of the table, while the glass of water is located on the left-most corner. The table's surface appears to be white……

……

Frame 420: A small red toy truck sitting on top of a knitted or crocheted blanket. The blanket is placed on a table, and the truck appears to be the main focus of the scene. The toy truck is positioned towards the right side of the blanket……

Object Bounding Boxes

(Object 0) book - frame 0 [47 19 81 33] frame 60 [47 19 81 32] frame 120 [47 19 81 32] frame 180 [47 19 81 32] frame 240 [47 19 81 32];
(Object 1) table cloth - frame 0 [6 26 100 100] frame 60 [6 26 100 100] frame 120 [6 26 100 100] frame 180 [6 26 100 100] ……
(Object 3) toy - frame 0 [73 86 95 100] frame 60 [68 41 83 70] frame 131 [83 89 100 100] frame 191 [68 40 85 68] frame 251 [89 93 100 100] frame 311 [89 93 100 100] frame 371 [68 39 84 66];
……

Question: What happened once the person removed an object from the tabletop?
Choices: (A) The launched object fell off the table.
(B) The launched object did not fall off the table.
(C) No object was removed from the tabletop.

Caption Model: (C)
Caption + Box Model: (B)

Figure 6. Qualitative example on Perception Test. Although the captions can capture the toy truck on the table, only the caption-and-box model can recognize the spatial relation between the toy truck and the table based on the object bounding boxes.

Setting	Models	Video	Box	CLEVRER-MC	Perception Test	STAR	NExT-QA	IntentQA
Zero-shot	VideoLLaMA2	✓		45.6	51.4	57.1	74.1	73.8
	ObjectMLLM	✓	✓	34.4	35.2	25.7	23.2	21.1
LoRA Fine-tuned	VideoLLaMA2	✓		67.9	66.0	66.5	**79.8**	**76.7**
	ObjectMLLM	✓	✓	**77.6**	**66.6**	**67.2**	78.5	75.5

Table 3. Performance of ObjectMLLM when a pre-trained VideoLLaMA2 is used. ObjectMLLM outperforms fine-tuned VideoLLaMA2 on benchmarks that focus more on spatial understanding. Notably, the performance gap on CLEVRER-MC is significant.

qualitative example in Figure 6 shows that the model can infer the spatial relation of the objects based on bounding boxes. Meanwhile, captions only miss the precise location of the truck, which is critical to answer the question. More qualitative examples are in Appendix E.

On NExT-QA and IntentQA, the box-only model cannot achieve better performance than the caption-only model and the video-only model. As discussed in Appendix E, these benchmarks focus on human actions and causal reasoning of events, which are difficult to be represented by object bounding boxes alone. This shows that spatiotemporal object information is not equally important on all benchmarks.

Finally, while our caption-and-box models can always beat or be on par with caption-only and box-only models, integrating visual embedding does not improve the performance over the caption-and-box models on any benchmark. This result is in line with Vamos [36], which highlights the difficulty in training effective embedding projectors for distributed representations with limited data.

4.5. Boosting pre-trained MLLMs with objects

We further study whether object representation can boost the performance of pre-trained MLLMs, which may already implicitly encode object information via their visual adapters. We develop ObjectMLLM from VideoLLaMA2 by including both the regular visual inputs and the language-represented object bounding boxes in the inputs. Table 3 shows that ObjectMLLM with pre-trianed VideoLLaMA2 backbone cannot understand the bounding boxes in a zero-shot manner. However, after LoRA fine-tuning the model with video and boxes as inputs on the target benchmarks, ObjectMLLM outperforms VideoLLaMA2 fine-tuned with only video inputs on CLEVRER-MC, Perception Test, and STAR. These results show that the object bounding boxes provide additional information over what VideoLLaMA2 can get from distributed visual embeddings. Perhaps not surprisingly, the relative gains are smaller compared to Table 1 as object information has already been partially integrated via visual adapters.

4.6. Comparison with existing MLLMs

In Table 4, we compare the performance of ObjectMLLM with existing MLLMs, including models with large-scale pre-trained visual adapter [4, 37, 45, 52] and models without it [36]. With object bounding boxes, ObjectMLLM consistently outperforms other MLLMs in both settings. The gaps are significant on CLEVRER-MC and Perception Test, which reveals the weakness of existing MLLMs in understanding spatiotemporal object configurations.

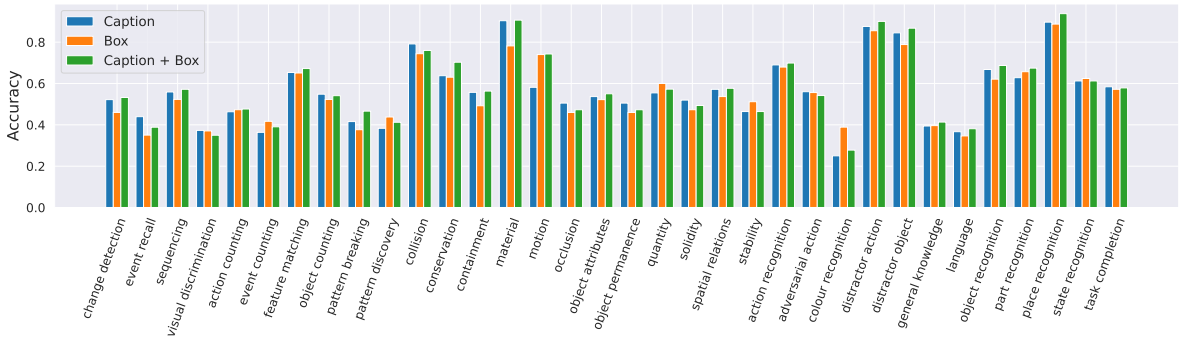


Figure 7. Accuracy of difference types of questions on Perception Test. Bounding boxes bring notable improvement on motion questions.

Models	Size	CLEVRER-MC	Perception Test	STAR	NExT-QA	IntentQA
w/ pre-trained visual adapter						
LLaVA-Next-Video-DPO [52]	7B	38.4*†	49.3*	-	-	-
VideoLLaMA2 [4]	7B	45.6†	51.4*	57.1*†	74.1†	73.8*†
SeViLA [45]	3B	-	62.0	64.9	73.8	-
ViLA [37]	3B	-	-	67.1	75.6	-
ObjectMLLM (VideoLLaMA2)	7B	**77.6**	**66.6**	**67.2**	**78.5**	**75.5**
w/o pre-trained visual adapter						
Vamos [36]	8B	-	62.3	63.7	**77.3**	74.2
ObjectMLLM (LLaMA3)	8B	**75.5**	**65.7**	**64.4**	76.6	**75.6**

Table 4. Comparison with existing MLLMs on five video QA benchmarks. Equipped with detected object bounding boxes, ObjectMLLM achieves consistent improvements over baseline methods without explicit object representations, when starting from both an MLLM with pre-trained visual adapters, or an LLM that takes video captions as inputs. *: Zero-shot generalization performance. †: Reproduced by us.

Modality	Video	Caption	Pose	V + P	C + P
Accuracy	66.6	60.2	63.5	68.6	**69.4**

Table 5. Evaluation results on BABEL-QA [5]. Human poses bring improvements over video embeddings and frame captions.

4.7. Generalize to richer symbolic representations

Language-based representation is also a natural way to incorporate structured visual representations other than object bounding boxes. For example, human pose can be represented by a list of keypoint names and coordinates in texts.

To demonstrate the effectiveness of ObjectMLLM with human pose, we use BABEL-QA [5], a human motion QA benchmark that focuses on human activity understanding. Each example in BABEL-QA has a sequence of 3D human keypoints over time and asks a question about their actions. In ObjectMLLM, we sample six frames from the sequence and represent the poses in text, with all coordinates normalized to integers within $[0, 1000]$, for example,

> Frame 0: pelvis [500 500 542] left hip [497 499 538] right hip [502 499 538] spine [499 498 548] ...

We render the poses as videos and employ CLIP for visual embeddings and GPT-4o for captions. Table 5 demonstrates the effectiveness of human poses, and that ObjectMLLM generalizes to richer structured visual representations.

5. Conclusion

We investigate how can objects help video-language understanding in the context of multimodal large language models. We demonstrate the effectiveness of object-centric representations extracted by off-the-shelf computer vision algorithms. Unlike distributed visual representations such as CLIP, object-centric representations can be integrated into MLLMs in a data-efficient manner, such as by rendering as plain text. We introduce ObjectMLLM that utilizes object-centric representations via bounding box adapters, and demonstrate that ObjectMLLM achieves competitive performance over approaches that utilize only visual embeddings or captions across six video QA benchmarks. We also observe that ObjectMLLM generalizes to richer structured visual representations, such as human pose on the BABEL-QA benchmark. We believe our observations highlight the importance of explicitly integrating computer vision models into MLLMs via language-based, or other data-efficient interfaces, making vision a first-class citizen for vision-language models again.

Acknowledgments: This work is supported by the Global Research Outreach program of Samsung. Our research was conducted using computational resources at the Center for Computation and Visualization at Brown University. We appreciate valuable feedback from Calvin Luo, Tian Yun, Yuan Zang, and Zilai Zeng.

References

[1] William Berrios, Gautam Mittal, Tristan Thrush, Douwe Kiela, and Amanpreet Singh. Towards language models that can see: Computer vision through the lens of natural language. *arXiv preprint arXiv:2306.16410*, 2023. 2

[2] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. *arXiv preprint arXiv:2306.15195*, 2023. 3

[3] Tianheng Cheng, Lin Song, Yixiao Ge, Wenyu Liu, Xinggang Wang, and Ying Shan. YOLO-World: Real-time open-vocabulary object detection. In *CVPR*, 2024. 4

[4] Zesen Cheng, Sicong Leng, Hang Zhang, Yifei Xin, Xin Li, Guanzheng Chen, Yongxin Zhu, Wenqi Zhang, Ziyang Luo, Deli Zhao, and Lidong Bing. VideoLLaMA 2: Advancing spatial-temporal modeling and audio understanding in video-llms. *arXiv preprint arXiv:2406.07476*, 2024. 2, 5, 7, 8

[5] Mark Endo, Joy Hsu, Jiaman Li, and Jiajun Wu. Motion question answering via modular motion programs. *ICML*, 2023. 2, 8

[6] Aaron Grattafiori et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 5

[7] Jiaming Han, Renrui Zhang, Wenqi Shao, Peng Gao, Peng Xu, Han Xiao, Kaipeng Zhang, Chris Liu, Song Wen, Ziyu Guo, Xudong Lu, Shuai Ren, Yafei Wen, Xiaoxin Chen, Xiangyu Yue, Hongsheng Li, and Yu Qiao. ImageBind-LLM: Multi-modality instruction tuning. *arXiv preprint arXiv:2309.03905*, 2023. 1

[8] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *ICLR*, 2022. 5, 1

[9] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023. 5

[10] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-UniVi: Unified visual representation empowers large language models with image and video understanding. In *CVPR*, 2024. 2

[11] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14:201–211, 1973. 2, 6

[12] Dohwan Ko, Ji Soo Lee, Wooyoung Kang, Byungseok Roh, and Hyunwoo J Kim. Large language models are temporal and causal reasoners for video question answering. In *EMNLP*, 2023. 2

[13] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-OneVision: Easy visual task transfer. *TMLR*, 2025. 2

[14] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 2, 3

[15] Jiapeng Li, Ping Wei, Wenjuan Han, and Lifeng Fan. IntentQA: Context-aware video intent reasoning. In *ICCV*, 2023. 5, 1

[16] Junyan Li, Delin Chen, Yining Hong, Zhenfang Chen, Peihao Chen, Yikang Shen, and Chuang Gan. CoVLM: Composing visual entities and relationships in large language models via communicative decoding. In *ICLR*, 2024. 3

[17] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. VideoChat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2

[18] Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. MVBench: A comprehensive multi-modal video understanding benchmark. In *CVPR*, 2024. 5, 1

[19] Xiujun Li, Xi Yin, Chunyuan Li, Pengchuan Zhang, Xiaowei Hu, Lei Zhang, Lijuan Wang, Houdong Hu, Li Dong, Furu Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *ECCV*, 2020. 3

[20] Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-LLaVA: Learning united visual representation by alignment before projection. In *EMNLP*, 2024. 2

[21] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. VILA: On pre-training for visual language models. In *CVPR*, 2024. 3

[22] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 5

[23] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024. 3, 5

[24] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-ChatGPT: Towards detailed video understanding via large vision and language models. In *ACL*, 2024. 2

[25] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Anton Belyi, et al. MM1: methods, analysis and insights from multimodal llm pre-training. In *ECCV*, 2024. 3

[26] Juhong Min, Shyamal Buch, Arsha Nagrani, Minsu Cho, and Cordelia Schmid. MoReVQA: Exploring modular reasoning models for video question answering. In *CVPR*, 2024. 2

[27] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *TMLR*, 2024. 2

[28] Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, Qixiang Ye, and Furu Wei. Grounding

multimodal large language models to the world. In *ICLR*, 2024. 3

[29] Viorica Pătrăucean, Lucas Smaira, Ankush Gupta, Adrià Recasens Continente, Larisa Markeeva, Dylan Banarse, Skanda Koppula, Joseph Heyward, Mateusz Malinowski, Yi Yang, Carl Doersch, Tatiana Matejovicova, Yury Sulsky, Antoine Miech, Alex Frechette, Hanna Klimczak, Raphael Koster, Junlin Zhang, Stephanie Winkler, Yusuf Aytar, Simon Osindero, Dima Damen, Andrew Zisserman, and João Carreira. Perception Test: A diagnostic benchmark for multimodal video models. In *NeurIPS*, 2023. 2, 5, 1

[30] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2, 5

[31] Hanoona Rasheed, Muhammad Maaz, Sahal Shaji, Abdelrahman Shaker, Salman Khan, Hisham Cholakkal, Rao M Anwer, Eric Xing, Ming-Hsuan Yang, and Fahad S Khan. GLaMM: Pixel grounding large multimodal model. In *CVPR*, 2024. 3

[32] Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollar, and Christoph Feichtenhofer. SAM 2: Segment anything in images and videos. In *ICLR*, 2025. 4

[33] Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. What does clip know about a red circle? visual prompt engineering for vlms. In *ICCV*, 2023. 5

[34] Shengbang Tong, Ellis L Brown II, Penghao Wu, Sanghyun Woo, ADITHYA JAIRAM IYER, Sai Charitha Akula, Shusheng Yang, Jihan Yang, Manoj Middepogu, Ziteng Wang, Xichen Pan, Rob Fergus, Yann LeCun, and Saining Xie. Cambrian-1: A fully open, vision-centric exploration of multimodal LLMs. In *NeurIPS*, 2024. 1

[35] Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. In *CVPR*, 2024. 1

[36] Shijie Wang, Qi Zhao, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. Vamos: Versatile action models for video understanding. In *ECCV*, 2024. 1, 2, 4, 5, 7, 8

[37] Xijun Wang, Junbang Liang, Chun-Kai Wang, Kenan Deng, Yu Lou, Ming Lin, and Shan Yang. ViLA: Efficient video-language alignment for video question answering. *arXiv preprint arXiv:2312.08367*, 2024. 7, 8

[38] Yi Wang, Kunchang Li, Yizhuo Li, Yinan He, Bingkun Huang, Zhiyu Zhao, Hongjie Zhang, Jilan Xu, Yi Liu, Zun Wang, Sen Xing, Guo Chen, Junting Pan, Jiashuo Yu, Yali Wang, Limin Wang, and Yu Qiao. InternVideo: General video foundation models via generative and discriminative learning. *arXiv preprint arXiv:2212.03191*, 2022. 2

[39] Ying Wang, Yanlai Yang, and Mengye Ren. LifelongMemory: Leveraging llms for answering queries in long-form egocentric videos. *arXiv preprint arXiv:2312.05269*, 2024. 2

[40] Zhenhailong Wang, Manling Li, Ruochen Xu, Luowei Zhou, Jie Lei, Xudong Lin, Shuohang Wang, Ziyi Yang, Chenguang Zhu, Derek Hoiem, et al. Language models with image descriptors are strong few-shot video-language learners. In *NeurIPS*, 2022. 1

[41] Bo Wu, Shoubin Yu, Zhenfang Chen, Joshua B Tenenbaum, and Chuang Gan. STAR: A benchmark for situated reasoning in real-world videos. In *NeurIPS*, 2021. 5, 1

[42] Junbin Xiao, Xindi Shang, Angela Yao, and Tat-Seng Chua. NExT-QA: Next phase of question-answering to explaining temporal actions. In *CVPR*, 2021. 5, 1

[43] Le Xue, Manli Shu, Anas Awadalla, Jun Wang, An Yan, Senthil Purushwalkam, Honglu Zhou, Viraj Prabhu, Yutong Dai, Michael S Ryoo, et al. xGen-MM (BLIP-3): A family of open large multimodal models. *arXiv preprint arXiv:2408.08872*, 2024. 2

[44] Kexin Yi, Chuang Gan, Yunzhu Li, Pushmeet Kohli, Jiajun Wu, Antonio Torralba, and Joshua B. Tenenbaum. CLEVRER: collision events for video representation and reasoning. In *ICLR*, 2020. 5, 1

[45] Shoubin Yu, Jaemin Cho, Prateek Yadav, and Mohit Bansal. Self-chained image-language model for video localization and question answering. In *NeurIPS*, 2023. 2, 7, 8

[46] Andy Zeng, Maria Attarian, brian ichter, Krzysztof Marcin Choromanski, Adrian Wong, Stefan Welker, Federico Tombari, Aveek Purohit, Michael S Ryoo, Vikas Sindhwani, Johnny Lee, Vincent Vanhoucke, and Pete Florence. Socratic models: Composing zero-shot multimodal reasoning with language. In *ICLR*, 2023. 1, 2

[47] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *ICCV*, 2023. 2

[48] Ce Zhang, Taixi Lu, Md Mohaiminul Islam, Ziyang Wang, Shoubin Yu, Mohit Bansal, and Gedas Bertasius. A simple llm framework for long-range video question-answering. In *EMNLP*, 2024. 1, 2

[49] Hang Zhang, Xin Li, and Lidong Bing. Video-LLaMA: An instruction-tuned audio-visual language model for video understanding. In *EMNLP*, 2023. 2

[50] Pengchuan Zhang, Xiujun Li, Xiaowei Hu, Jianwei Yang, Lei Zhang, Lijuan Wang, Yejin Choi, and Jianfeng Gao. VinVL: Revisiting visual representations in vision-language models. In *CVPR*, 2021. 3

[51] Renrui Zhang, Jiaming Han, Chris Liu, Peng Gao, Aojun Zhou, Xiangfei Hu, Shilin Yan, Pan Lu, Hongsheng Li, and Yu Qiao. LLaMA-Adapter: Efficient fine-tuning of language models with zero-init attention. In *ICLR*, 2024. 5, 1

[52] Yuanhan Zhang, Bo Li, haotian Liu, Yong jae Lee, Liangke Gui, Di Fu, Jiashi Feng, Ziwei Liu, and Chunyuan Li. LLaVA-NeXT: A strong zero-shot video understanding model. https://llava-vl.github.io/blog/2024-04-30-llava-next-video/, 2024. 7, 8

[53] Qi Zhao, Shijie Wang, Ce Zhang, Changcheng Fu, Minh Quan Do, Nakul Agarwal, Kwonjoon Lee, and Chen Sun. AntGPT: Can large language models help long-term action anticipation from videos? In *ICLR*, 2024. 2