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

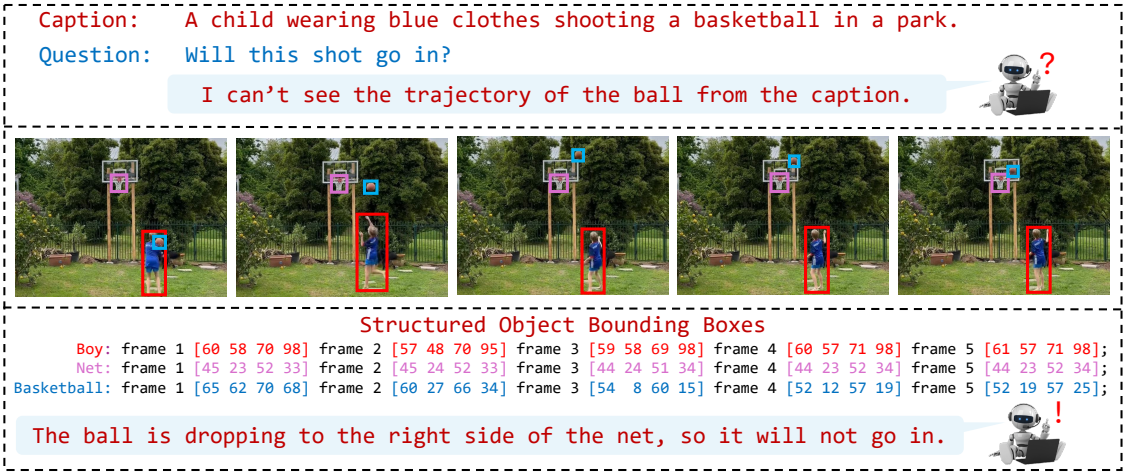


Figure 1. Socratic Models [36, 46, 48] perceive the world from the lens of natural language descriptions, which may miss important spatiotemporal information (*top*). Multimodal large language models (MLLMs), on the other hand, can integrate visual information via distributed embeddings, but typically 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 re-

[†]Paper approved as a Master's report for this author.

mains 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 success 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 knowl-

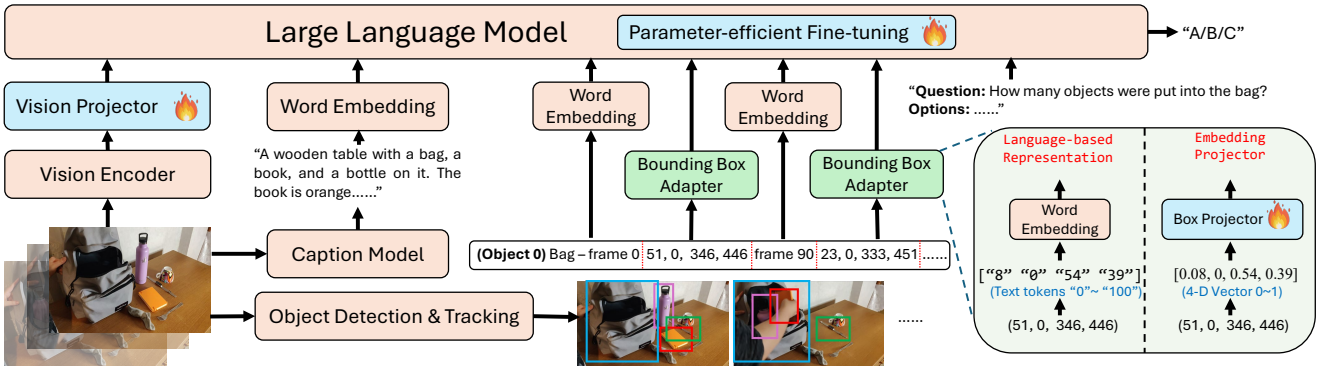


Figure 2. Pipeline of ObjectMLLM. The possible model inputs are visual embeddings, video frame captions, and object bounding boxes. ObjectMLLM encodes object coordinates with a language-based ("symbolic") representation, or with an embedding projector. The former represents the bounding boxes as plain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

edge 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 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,


```
There are in total 9 objects in the video.
(Object 0) bag - Frame 0 <box> Frame 90 <box> Frame 180 <box> Frame 270 <box>
Frame 360 <box>;
(Object 1) book - Frame 0 <box> Frame 90 <box> Frame 180 <box>;
(Object 2) watch - Frame 0 <box> Frame 90 <box> Frame 180 <box> Frame 270
<box>;
(Object 3) bottle - Frame 0 <box> Frame 90 <box> Frame 180 <box> Frame 270
<box> Frame 360 <box>;
(Object 4) cup ......
                <box> is a special token for one bounding box
```

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.

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 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.

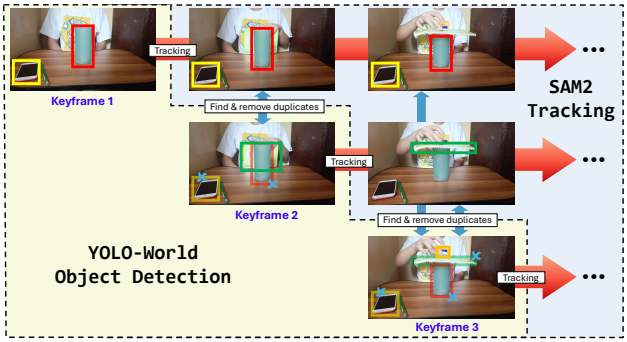


Figure 4. Workflow of object detection and tracking. We iteratively detect objects in uniformly sampled keyframes, and create new tracks for new objects not associated with any existing track.

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 vectorized 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.

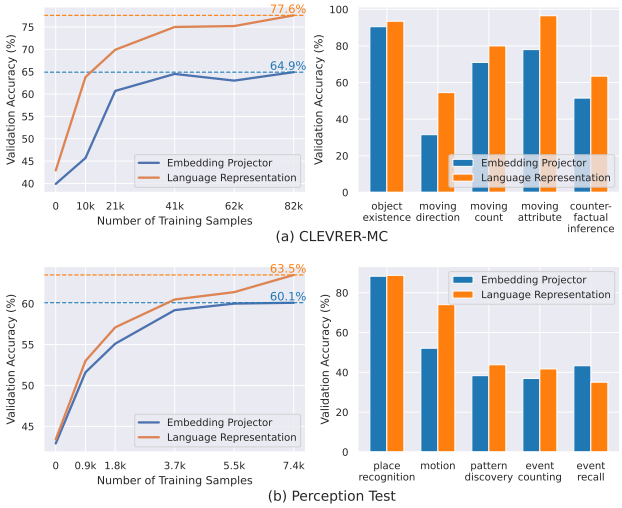


Figure 5. Performance of the box adapters under various training data amounts (left) and accuracy breakdown by question types (right). Only a subset of the question types in Perception Test are listed here. The language-based representation consistently outperforms the embedding projector with different 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.

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.

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

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.

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 downsampling rate varies from video to video.

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

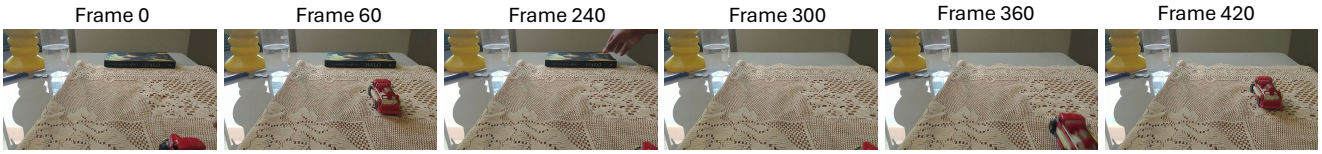
Figure 5 (right) shows the accuracy breakdown by question types. While the performances of the two adapters are comparable on some types of question, the language-based representation shows great superiority on motion-related questions. This phenomenon in motion questions happens to be consistent with 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 Section 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 capabil-



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.

ity 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 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 Section 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 Section 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 con-

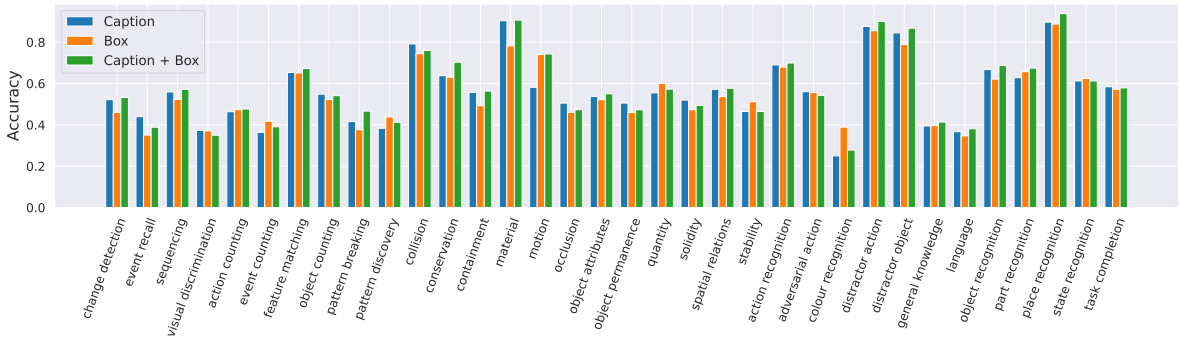


Figure 7. Accuracy of difference types of questions on Perception Test. Bounding boxes bring notable improvement on motion questions.

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.

sistently 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.

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 vi-

sion 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. 2

[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

How Can Objects Help Video-Language Understanding?

Supplementary Material

We elaborate on the implementation details of ObjectMLLM in Section A. In Section B, we explore the design choices for the bounding box projector, the effectiveness of object bounding boxes compared to object labels, the modality fusion strategy, and bounding box sampling rate. In Section C, we conduct experiments on NExT-QA and IntentQA to show that spatial cues play a minor role on these two benchmarks. Section D demonstrates the quality of the detected object bounding boxes and investigates how it affects the performance of ObjectMLLM. Qualitative results of our method are presented in Section E. Finally, Section F summarizes a few unsuccessful attempts.

A. Implementation Details

A.1. Object detection and tracking

In the object detection and tracking process, the video keyframes are sampled at 1 FPS. SAM 2 tracking is performed at the original frame rate of each video. We use the pre-trained YOLO-World checkpoint `YOLO-World-v2-L_CLIP-Large_800`. The employed SAM 2 checkpoint is `sam2.1_hiera_large`.

All the benchmarks (or their source datasets) in our experiments provide either manually annotated or algorithm-detected object bounding boxes. To adapt YOLO-World to the benchmarks, we fine-tune it on the training set of each benchmark individually. Fine-tuning is performed with a learning rate of $2e-4$, a weight decay of 0.05, and a batch size of 64. The number of training images, the number of training epochs, and the score thresholds used during inference are listed in Table A1.

A.2. Downsampling Rates of Bounding Boxes

As described in Section 4.2, we uniformly and temporally downsample the bounding box sequences to reduce the total number of input tokens. As the videos in CLEVRER-MC [18, 44] are 5 seconds in length, we sample one frame every one second, resulting in 6 sampled frames per video. For other benchmarks, the videos have varying lengths and numbers of objects. Therefore, we assign a separate sampling rate for each video. Specifically, we use binary search to find the maximal sampling rate for each video such that the number of bounding box tokens after downsampling is less than 1,000. We show the distributions of the sampling rates and the resulting numbers of frames in Figure A1.

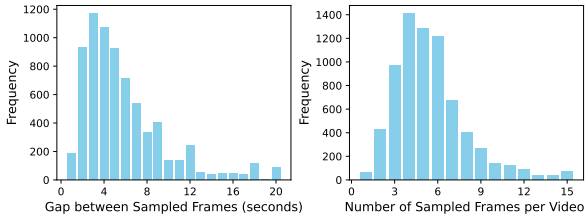
A.3. ObjectMLLM training

When fine-tuning LLaMA3-8B with LLaMA-Adapter [51], we use a batch size of 64. The learning rate is linearly warmed up to 0.0225 in the first 20% steps, after which cosine learning rate annealing is applied. The same learning rate is applied to the LLaMA-Adapter weights, bounding box embedding projector, and visual embedding projector.

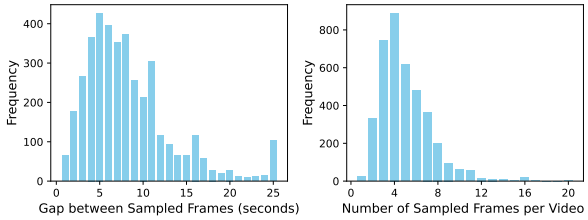
When fine-tuning VideoLLaMA2 with LoRA [8], we use a LoRA rank of 128 and a batch size of 128. The learning rate is linearly warmed up to $2e-5$ in the first 3% steps, after which cosine learning rate annealing is applied. The

Benchmark	#training images	#epochs	score threshold
CLEVRER-MC	60 k	7	$1e-3$
Perception Test	46 k	50	0.35
STAR	106 k	20	0.2
NExT-QA & IntentQA	151 k	10	0.4

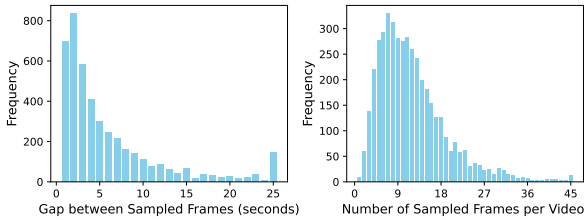
Table A1. Hyperparameters in YOLO-World fine-tuing and inference across different benchmarks.



(a) Perception Test



(b) STAR



(c) NExT-QA & IntentQA

Figure A1. Distributions of the bounding box sampling rates for Perception Test [29], STAR [41], NExT-QA [42], and IntentQA [15]. We show the gaps between the sampled frames and the numbers of the sampled frames. The last bin includes all elements greater than or equal to the corresponding x-coordinate value. IntentQA shares the same distribution as NExT-QA because it is sourced from NExT-QA.

Box adapter	Initialization	CLEVRER-MC	Perception Test
Embedding Projector	Random	42.8	57.6
	Zero	**64.9**	**60.1**

Table A2. Ablation on the initialization of the box embedding projector. The defulat random initialization in PyTorch is Kaiming uniform distribution. We find that zero-initialized linear layer as the box embedding projector always yields better performance than the default initialization.

same learning rate is applied to the LoRA weight and visual embedding projector. The pre-trained checkpoint used is `VideoLLaMA2-7B-16F`.

In both settings, the model is trained for 1 epoch on CLEVRER, 5 epochs on NExT-QA and STAR, 10 epochs on Perception Test and IntentQA, and 50 epochs on BABEL-QA. The vision encoder is kept frozen in all experiments.

B. Ablation Studies

B.1. Embedding projector

We show that the language-based box adapter is always more performant than the embedding projector in Section 4.3. However, it is possible that the poor performance of the embedding projector is due to its design. In this section, we explore a few design choices of the embedding projector, including initialization, number of layers, and number of resulting tokens. We find that the embedding projector is inferior to the language-based representation across all design choices.

Projector initialization. When training a projector between a novel modality and the LLM backbone, previous works (*e.g.* LLaVA [22] and Vamos [36]) use the default initialization for linear layers (the Kaiming uniform distribution in PyTorch). However, we find that the default random initialization significantly impedes the training of the bounding box embedding projector. As shown in Table A2, we find that initializing the linear layer weights to zero can facilitate the learning of embedding projector. We hypothesize that the default initialization would project the bounding boxes outside the LLM word embedding space, confusing the LLM backbone at the beginning of training. In contrast, a zero-initialized linear layer can map every bounding box to a zero vector, which is extremely close to the special tokens in the LLM vocabulary. This can prevent the bounding boxes from corrupting LLM behavior.

Number of projector layers. Instead of a single linear layer, we explore using a multilayer perceptron (MLP) as the bounding box projector. In this experiment, we set the number of hidden units in each layer to match the dimension of word embeddings of the LLM (4,096 for LLaMA3-8B). We use GeLU as the activation function. As Table A3 suggests, increasing the number of MLP layers yields only marginal performance gains for the embedding projector method. More importantly, it is still dominated by the language-based representation approach.

Box adapter	#Layers	CLEVRER-MC	Perception Test
Embedding Projector	1	64.9	60.1
	2	65.0	59.7
	3	65.0	60.2
Language-based Representation	-	**77.6**	**63.5**

Table A3. Ablation on the number of layers of the box embedding projector. Enlarging the number of MLP layers does not bring significant improvement. And they are outperformed by the language-based representation box adapter.

Box adapter	#Tokens per box	CLEVRER-MC
Embedding Projector	1	64.9
	9	63.4
Language-based Representation	9	**77.6**

Table A4. Ablation on the number of resulting tokens in the embedding projector method. Increasing the number of tokens to be the same as that in the language-based representation method degrades the performance.

Number of resulting tokens. While the embedding projector maps each bounding box into only one token, the language-based representation uses nine tokens to describe one bounding box (four numbers, three spaces, and two square brackets). One might argue that the expressiveness of the bounding box embedding projector is limited by the number of tokens. To address this concern, we experiment with bounding box projectors that map each bounding box into nine tokens rather than one token. The results are in Table A4. We find that increasing the number of resulting tokens per bounding box cannot improve the performance of the embedding projector method.

B.2. Are object labels sufficient on their own?

As shown in Figure 3, object labels (i.e., object names) are also provided when we format the bounding boxes. If the object labels are hidden, the bounding boxes themselves convey much less information because the object associated with each box is unknown. Object labels can provide important information to the model; for example, color recognition in Figure 7 is improved, which is definitely not inferable from unannotated bounding boxes alone. We raise the question: are object labels sufficient on their own, or does the model still derive benefits from bounding box information?

To answer this question, we train a model with object labels provided but bounding boxes hidden. In Table A6, we find that the model consistently performs better when the

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
	✓	✓	75.5(75.8)	65.7(64.1)	64.4(63.7)	76.6(77.2)	75.6(75.4)
✓	✓	✓	75.4(29.5)	63.9(33.8)	62.9(62.8)	76.2(75.8)	75.0(73.4)

Table A5. Ablation on modality fusion strategy. Blue ones are the results of jointly training on all the modalities at once, while the others are trained in a modality-by-modality manner. The modality-by-modality fusion strategy can outperform the jointly training in most cases. And the joint training is sometimes unstable when the video inputs are involved.

Input	CLEVRER-MC	Perception Test	STAR
Obj. label	59.8	60.0	58.8
Obj. label + box	**77.6**	**63.5**	**59.1**

Table A6. Ablation on the bounding boxes versus object labels. The model indeed utilizes the boxes other than the object labels.

FPS	0.25	0.5	1	2
Accuracy	74.0	73.9	77.6	78.0

(a) CLEVRER-MC

Max Seq. Len.	600	800	1000	1200
Avg. FPS	0.13	0.20	0.26	0.31
Accuracy	62.7	63.0	63.5	63.7

(b) Perception Test

Table A7. Ablation on bounding box temporal sampling rate. Higher sampling rate usually leads to higher accuracy.

object boxes are also provided, indicating that the model makes effective use of object bounding boxes. However, the differences are more notable on CLEVRER-MC and Perception Test than on STAR, indicating the improvement made by observing bounding boxes on STAR is mainly attributed to the revealed object labels. This is reasonable because the questions in STAR are annotated based off the scene graphs. It also highlights that emphasizing the question-related objects instead of describing the scene at a high level through video frame captions helps more on video question answering.

B.3. Modality fusion strategy

Section 3.4 mentions that we fuse the modalities (captions, bounding boxes, and videos) in a modality-by-modality approach instead of joint training at once. In Table A5, we compare the two fusion strategies. We find that the modality-by-modality method outperforms joint training in the majority of cases. In addition, the joint training approach is sometimes unstable. It leads to extremely low performance on CLEVRER-MC and Perception Test when using all three input modalities. Lastly, we find that both fusion methods are ineffective at leveraging the visual embeddings. We therefore urge the development of new multimodal fusion strategies that can make visual inputs valuable.

Figure A2. LLaVA-1.5 caption vs. GPT-4o spatial-free caption.

Caption	Box	NExT-QA	IntentQA
LLaVA-1.5		76.6	75.7
LLaVA-1.5	✓	76.6	75.6
GPT-4o Spatial-free		76.1	73.6
GPT-4o Spatial-free	✓	76.3	75.6

Table A8. Performance with spatial-free GPT-4o captions.

B.4. Temporal sampling rate of bounding boxes

We conduct an ablation study on the bounding box sampling rate on CLEVRER-MC and Perception Test in Table A7. On CLEVRER-MC, all the videos have the same duration. Therefore, we directly adjust the bounding box sampling rate from 0.25 FPS to 2 FPS. On Perception Test, we vary the maximum box token length mentioned in Section 4.2 and report the corresponding average frame rates. As the results show, a higher sampling rate generally yields higher accuracy, indicating a trade-off between performance and efficiency.

C. Spatial Information Ablation on NExT-QA and IntentQA

In our experiments in Table 1 and Table 3, incorporating object bounding boxes as additional information does not improve model performance on NExT-QA and IntentQA. While these benchmarks focus on causal reasoning about events, the spatial information may not play a crucial role to answer the questions. In this section, we further verify through experiments the limited usefulness of spatial information in these two benchmarks.

As Figure A2 shows, the video frame captions generated by LLaVA-1.5 include some spatial cues in the image. To eliminate the influence of spatial cues, we feed each video frame to GPT-4o and ask it to generate a caption that does

Box	CLEVRER-MC	Perception Test	STAR	NExT-QA	IntentQA
Model-tracked	**77.6**	63.5	59.1	63.7	**66.2**
Annotation	74.9	**66.8**	**78.9**	**65.5**	65.3

Table A9. Model performance with object bounding boxes tracked by computer vision models or with those annotated by the benchmarks. Bounding box annotations in CLEVRER-MC, NExT-QA, and IntentQA are also algorithm-detected. Boxes in Perception Test and STAR are manually annotated. The experiments are in the box-only setting.

not involve any spatial information. Specifically, we use the following prompt:

> <image> Faithfully describe the content of the above image, avoid mentioning specific objects and try your best to provide a scene-level, holistic summary. No mention of any spatial information.

Figure A2 shows an example of such spatial-free captions. With the spatial-free captions, we repeat our experiments on NExT-QA and IntentQA, using ObjectMLLM with LLaMA3 as the backbone. The results are in Table A8. We find that removing the spatial cues only results in a 0.5% drop in accuracy on NExT-QA and a 2.1% drop on IntentQA. This indicates that the questions in these two datasets are still highly answerable even without spatial information. More importantly, incorporating object bounding boxes in addition to spatial-free captions can recover the performance, especially on IntentQA. This again demonstrates our method's effectiveness in conveying spatial information to MLLMs.

D. Quality of the Tracked Bounding Boxes

In this section, we assess the quality of the extracted object bounding boxes and whether it hinders the performance of our model. Because the evaluation benchmarks themselves provide object bounding box annotations (either human-annotated or algorithm-detected), we can compare the boxes obtained by our workflow with them.

Figure A4 visualizes the tracked and annotated bounding boxes across different benchmarks. All the examples are from the validation/test set of the benchmarks. We find that the detection and tracking quality on synthetic videos (CLEVRER-MC) is highly accurate. On the realistic videos (Perception Test, STAR, NExT-QA, and IntentQA), our employed tracking method can capture the main objects although some noise is present.

In Table A9, we evaluate our model with the bounding box annotations as inputs. When the annotations serve as inputs, the model is trained again with the annotated boxes before evaluation to mitigate train-test domain shift. While the performance with model-tracked bounding boxes is 2%–3% worse than that with annotations on Perception Test and NExT-QA, it is even better than the annotations on CLEVRER-MC and IntentQA. This is reasonable because the bounding box annotations provided in the CLEVRER and IntentQA benchmarks are also algorithm-detected and potentially noisier than ours. In contrast, Perception Test and STAR both provide human-annotated object bounding boxes. However, the performance gap on STAR is significantly larger than on Perception Test. We notice that the questions in STAR are generated by functional programs based on annotated object relation graphs. Because only objects of interest are annotated in STAR, using object annotations as input provides a strong prior that may bias the answers. As the tracking quality on STAR (Figure A4(c)) is fairly accurate, we hypothesize that the large performance gap is caused by the choices of objects of interest rather than the tracking precision. How we can filter the objects of interest from a video remains an open and valuable research challenge.

E. Qualitative Results

We show a qualitative result on CLEVRER-MC in Figure A5. The model can determine whether an object is moving based on its bounding boxes, which is an essential capability in this benchmark.

In Figures A6 to A9, we show qualitative results from Perception Test. In these examples, model can determine the motion of cameras, stability of objection configurations, and the number of objects taken out from bags. These questions are not answerable for the caption-only model.

In Figures A10 and A11, we show failure cases of our caption-and-box model. From the captions and object bounding boxes, the model cannot reliably infer the object states and appearances. Therefore, it fails to provide correct answers. However, visual embeddings, in principle, should capture these visual characteristics. We highlight the importance of devising MLLMs that can efficiently and effectively utilize distributed visual representations.

We also examine the failure cases on NExT-QA and IntentQA, where we found that our model struggles with questions involving human actions. For example, in Figure A12, the model with bounding box inputs is aware that there are a person and a dog in the video. However, the person's action cannot be determined from the bounding boxes. In contrast, as captions provide explicit descriptions of actions, the model with caption input is better at answering these questions, contributing to the performance gap in Table 1 compared to the box-only model.

Box adapter	Perception Test
Language-based Representation	63.5
Embedding Projector	60.1
Visual Prompting	59.7

Table A10. Performance of integrating bounding boxes using visual prompting. It is significantly less performant than the language-based representation and embedding projector.

Box	Visual embedding	Perception Test Acc.
✓	✗	63.5
✓	Frame-level	62.7
✓	Object-level	63.3

Table A11. Ablation on different visual embedding levels. Object-level visual embedding works better than the frame-level embedding but still cannot bring additional improvement when symbolic boxes are used.

F. Unsuccessful Attempts

F.1. Integrating boxes via visual prompting

Beyond integrating object-centric information via structural bounding box coordinates, we explore incorporating box information through visual prompting. Inspired by prior work [33], we overlay bounding boxes directly onto video frames and extract visual embeddings from these annotated frames. Within the same video, objects are distinguished by unique colors, and the color assigned to each object remains consistent across frames to maintain temporal coherence. Figure A3 demonstrates an example of the annotated frames from Perception Test. As shown in Table A10, integrating bounding boxes using visual prompting performs worse than the embedding projector and language-based representation on Perception Test.

F.2. Integrating object-level visual embeddings

Intuitively, fine-grained object appearances like texture cannot be accurately described by video frame captions and bounding boxes. But, in principle, they can be captured by distributed visual representations like CLIP [30] embeddings. However, Table 1 illustrates that integrating frame-level visual embeddings upon captions and bounding boxes does not bring additional benefits to the performance.

Initially, we hypothesize that frame-level visual embeddings are too high-level to capture object details. To investigate this problem, we experiment with an object-level visual representation to replace the frame-level embedding. Specifically, we crop the objects from the video frames and extract their CLIP embeddings as object embeddings. Then, based on the language-based representation, we append each object embedding after its bounding box of the corresponding timestamp using the template below, where each ⟨|obj_emb|⟩ indicates an object embedding.

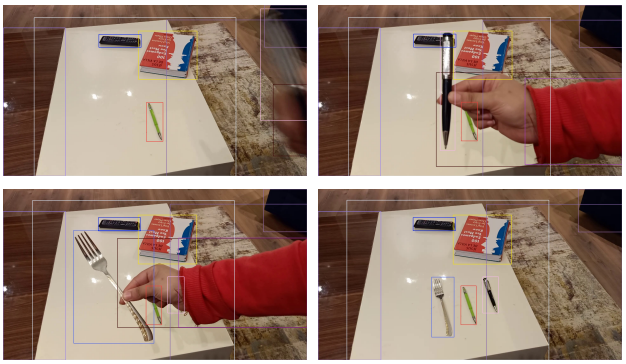
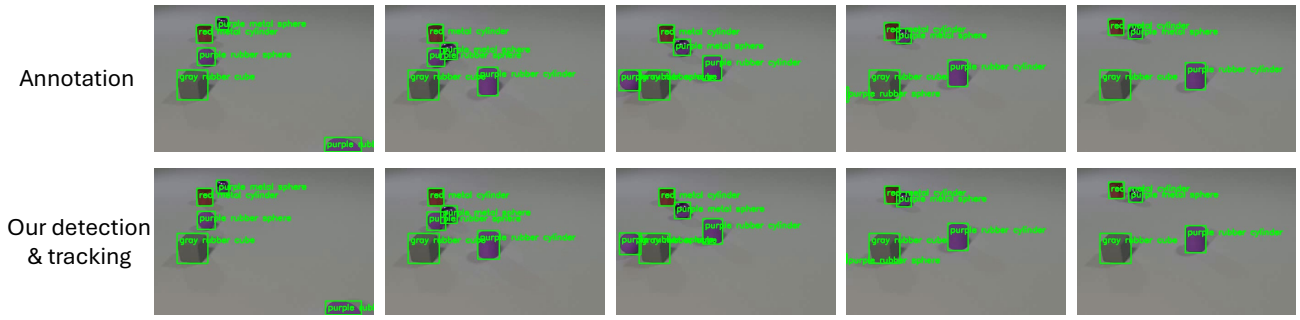


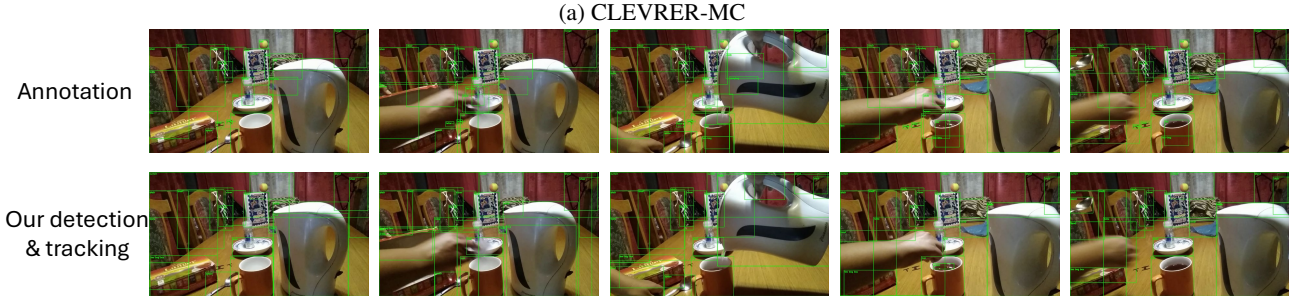
Figure A3. Examples of integrating bounding boxes via visual prompting from Perception Test.

(Object 0) bag – frame 0 [8 0 54 93] ⟨|obj_emb|⟩ frame 90 [4 0 52 94] ⟨|obj_emb|⟩ .....

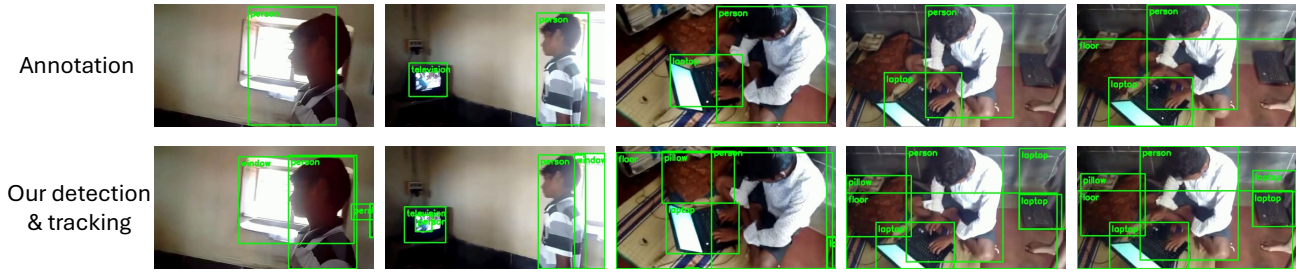
The results are in Table A11. The object-level visual embedding outperforms the frame-level embedding but still falls short of the box-only model. This experiment again highlights the difficulty of integrating distributed embedding into MLLMs in a data-efficient manner, which would be a challenging but valuable research topic.



(a) CLEVRER-MC



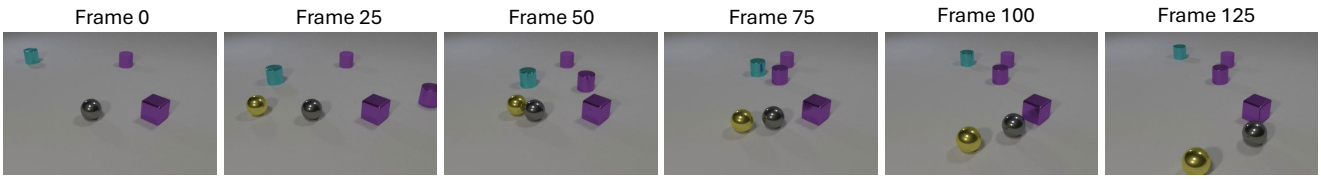
(b) Perception Test



(c) STAR

(d) NExT-QA & IntentQA

Figure A4. Visualization of the object bounding boxes across different benchmarks. IntentQA shares the same video source as NExT-QA. Our tracked bounding boxes are nearly perfect on CLEVRER-MC, while they are also fairly accurate on realistic videos.



Frame Captions

Frame 0: A white surface with three small objects placed on it…… One of the objects is a silver sphere, while the other two are cubes, one purple and the other blue…… with the silver sphere being closer to the left side of the image……
Frame 25: A white background with a variety of small, colorful objects placed on it. There are four distinct objects in the scene, each with a different color and shape…… The objects are positioned at various angles and distances from each other……
Frame 50: ……

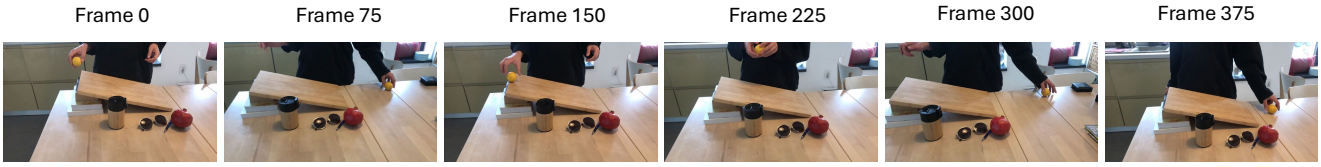
Object Bounding Boxes

(Object 0) purple metal cube - frame 0 [64 43 78 65] frame 25 [64 43 78 65] frame 50 [64 43 78 65] frame 75 [64 43 78 65]……
(Object 1) cyan metal cylinder - frame 0 [10 11 17 23] frame 25 [19 23 28 37] frame 50 [35 25 43 39] frame 75 [40 18 47 32]……
(Object 2) purple rubber cylinder - frame 0 [54 13 61 25] frame 25 [54 13 61 25] frame 50 [54 13 61 25] frame 75 [54 13 61 25]……
(Object 3) gray metal sphere - frame 0 [36 47 46 61] frame 25 [36 47 46 61] frame 50 [38 47 47 62] frame 75 [46 52 56 68]……
(Object 4) purple metal cylinder - frame 20 [99 44 100 53] frame 45 [67 28 76 44] frame 70 [50 23 58 38] frame 95 [50 23 58 37]……
(Object 5) yellow metal sphere - frame 12 [0 62 1 73] frame 37 [23 37 32 50] frame 62 [30 48 40 64] frame 87 [32 59 43 77]……

Question: How many moving metal objects are there?
Choices: (A) 2 (B) 1 (C) 3 (D) 4

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

Figure A5. Qualitative example on CLEVRER-MC. The model can determine whether an object is moving based on its bounding boxes.



Frame Captions

Frame 0: A wooden dining table with various items placed on it. There is a wooden cutting board, a cup, a book, a pair of sunglasses, and an apple. The person is standing near the table, holding a lemon…… The wooden cutting board is placed towards the center of the table, while the cup and the book are located closer to the left side. The sunglasses are positioned on the right side of the table……
……
Frame 300: A wooden table with various items placed on it. A wooden cutting board is the main focus, with a knife and a lemon on top of it. There is also a cup and a pair of sunglasses on the table. A person is standing near the table, possibly preparing to use the cutting board. In addition to the cutting board, there are two apples on the table, one near the center and the other towards the right side……

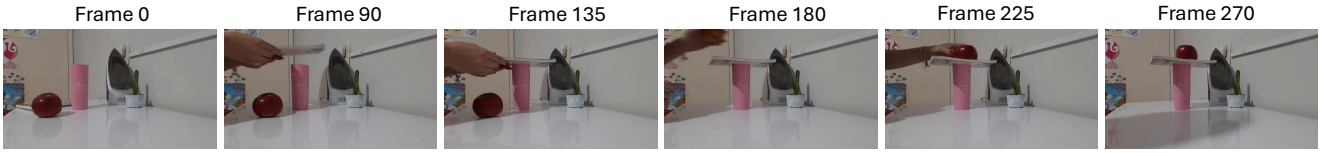
Object Bounding Boxes

(Object 0) table - frame 0 [13 36 100 100] frame 151 [5 42 94 100] frame 302 [0 42 67 100];
(Object 1) person - frame 0 [30 0 76 35] frame 151 [25 0 73 37] frame 302 [8 0 81 43];
(Object 2) wooden board - frame 0 [35 24 80 58] frame 151 [28 27 74 59] frame 302 [3 29 53 63];
(Object 3) jar - frame 0 [49 45 58 73] frame 151 [42 47 51 75] frame 302 [17 52 28 82];
……

Question: Is the camera moving or static?
Choices: (A) Moving (B) Static or shaking (C) I don't know

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

Figure A6. Qualitative example on Perception Test. The caption+box model can determine the motion of the camera from the changing object bounding boxes.



Frame Captions

Frame 0: A white table with a variety of objects on it. There is a cup, a potted plant, and a small ironing board. The cup is placed on the left side of the table, while the ironing board is situated towards the center. The potted plant is positioned on the right side……

……

Frame 270: A white table with a pink cup sitting on top of it. The cup is filled with an apple, and a small cactus is placed nearby. Above the table, there is an ironing board with an iron on it. The scene appears to be a simple, everyday arrangement of objects in a room.

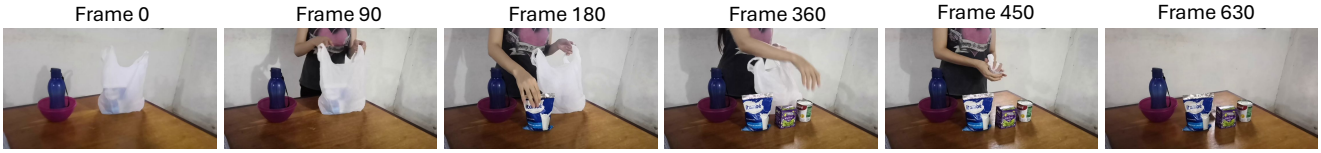
Object Bounding Boxes

(Object 0) tumbler - frame 0 [31 29 40 67] frame 90 [31 29 40 68] frame 180 [32 29 40 67] frame 270 [32 29 40 68];
(Object 5) book - frame 0 [6 59 32 68] frame 90 [24 12 47 22] frame 180 [21 23 52 32] frame 270 [21 22 52 33];
(Object 9) apple - frame 0 [13 53 28 73] frame 90 [13 53 28 74] frame 180 [16 0 28 11] frame 270 [33 13 44 49];
……

Question: Is the configuration of objects likely to be stable after placing the last object?
Choices: (A) One cannot judge the stability of this configuration.
(B) The configuration is likely to be stable.
(C) The configuration is likely to be unstable.

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

Figure A7. Qualitative example on Perception Test. The caption+box model can predict the stability of the object configuration because it is aware of the object locations.



Frame Captions

Frame 0: A wooden dining table with a pink bowl and a white plastic bag placed on it. The pink bowl is located on the left side of the table, while the white plastic bag is situated towards the right side. The bag appears to be a grocery bag……

……

Frame 630: A wooden dining table with various food items and a bottle placed on it. There are two cans of food, one located towards the right side of the table and the other towards the left side. A box of food is also present on the table, positioned near the center. In addition to the food items, there is a bowl situated on the left side of the table, and a spoon can be seen resting inside the bowl……

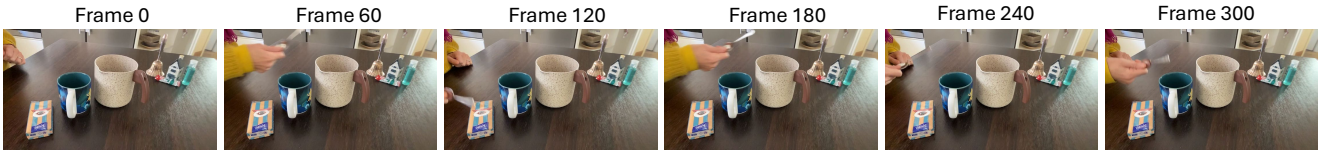
Object Bounding Boxes

(Object 3) bag - frame 0 [44 15 68 71] frame 90 [44 13 66 72] frame 180 [40 14 66 71] frame 270 [42 17 66 71] frame 360 [39 13 65 71] frame 450 [48 21 55 38] frame 540 [50 59 52 69];
(Object 5) tea bag box - frame 237 [50 27 58 33] frame 327 [52 63 62 83] frame 417 [52 64 62 83] frame 507 [52 63 62 83] frame 597 [52 63 62 83];
(Object 6) milk tetrapack - frame 124 [55 20 58 30] frame 214 [36 54 51 85] frame 304 [36 54 51 85] frame 394 [36 54 51 85] frame 484 [36 54 51 85] frame 574 [36 54 51 85];
(Object 7) box - frame 308 [53 21 57 31] frame 398 [62 59 70 78] frame 488 [62 59 69 78] frame 578 [62 59 69 78];
……

Question: How many objects did the person take out of the bag?
Choices: (A) 3 (B) 2 (C) 4

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

Figure A8. Qualitative example on Perception Test. The caption+box model can determine the number of objects taken out from the bag with the aid of object bounding boxes.



Frame Captions

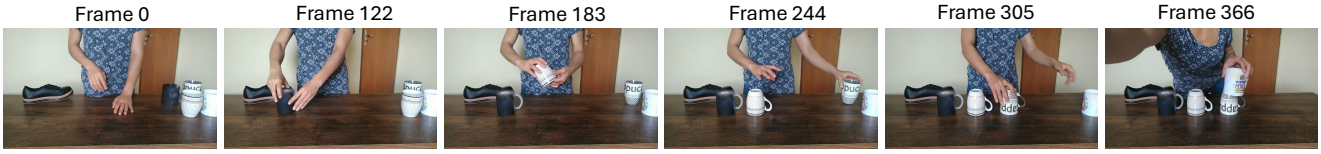
Frame 0: A wooden dining table with a cup and a mug placed on it. The cup is positioned towards the left side of the table, while the mug is situated closer to the center. The mug is larger than the cup and has a handle, making it a more functional and comfortable choice……
……
Frame 300: A wooden dining table with a variety of items placed on it. There are two cups, one of which is a coffee mug, and the other is a cream pitcher. The coffee mug is positioned towards the left side of the table…… A spoon can also be seen on the table……

Object Bounding Boxes

(Object 0) cup - frame 0 [43 21 68 65] frame 60 [43 21 68 65] frame 120 [43 21 68 65] frame 180 [43 21 68 65] frame 240 [43 21 68 65] frame 300 [43 21 68 65];
(Object 1) cup - frame 0 [25 35 41 74] frame 60 [25 35 41 74] frame 120 [25 35 41 74] frame 180 [25 35 41 74] frame 240 [25 35 41 74] frame 300 [25 35 41 74];
(Object 5) box - frame 0 [11 59 23 91] frame 60 [11 59 23 91] frame 120 [11 59 23 91] frame 180 [11 59 23 91] frame 240 [10 59 23 91] frame 300 [10 59 23 91];
(Object 9) spoon - frame 16 [0 21 2 24] frame 76 [14 28 31 39] frame 136 [1 9 12 35] frame 196 [17 21 32 32] frame 256 [30 4 47 18] frame 316 [0 23 3 29];
……

Question: What object does the person use to hit other objects? Caption Model: (A)
Choices: (A) pen (B) fork (C) spoon Caption + Box Model: (C)

Figure A9. Qualitative example on Perception Test. From the boudning box cooridnates, the caption+box model can observe that the spoon is moved to hit the other objects.



Frame Captions

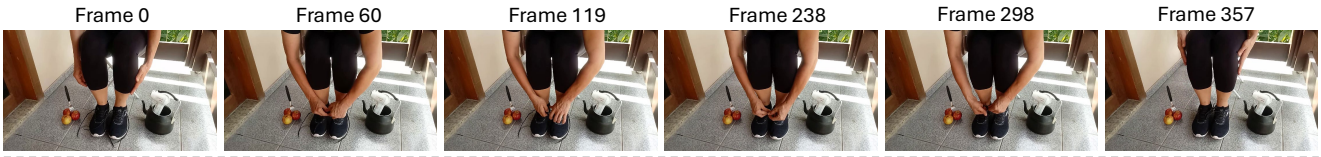
Frame 0: A person standing in front of a wooden dining table. The person is wearing a blue shirt and is positioned near the left side of the table. On the table, there is a cup placed towards the right side, and a pair of black shoes can be seen on the left side……
……
Frame 366: A wooden dining table with a variety of coffee mugs and cups placed on it. There are three coffee mugs, one of which is a tall mug, and two smaller cups. A person is standing near the table, holding a coffee mug, possibly preparing to pour coffee……

Object Bounding Boxes

(Object 0) cup - frame 0 [92 51 100 73] frame 61 [92 51 100 73] frame 122 [92 51 100 73] frame 183 [92 51 100 73] frame 244 [92 51 100 73] frame 305 [92 51 100 73] frame 366 [55 33 67 56];
(Object 2) glass - frame 0 [74 46 83 65] frame 61 [65 43 73 62] frame 122 [25 49 33 71] frame 183 [24 51 37 73] frame 244 [24 51 37 73] frame 305 [24 51 37 73] frame 366 [24 51 37 73];
(Object 6) cup - frame 0 [83 51 93 74] frame 61 [83 51 93 74] frame 122 [83 51 93 74] frame 183 [42 30 53 50] frame 244 [38 50 50 72] frame 305 [38 50 50 72] frame 366 [38 50 50 72];
(Object 7) cup - frame 0 [84 43 93 52] frame 61 [84 43 93 52] frame 122 [84 43 93 52] frame 183 [84 43 93 63] frame 244 [83 42 92 63] frame 305 [54 49 66 72] frame 366 [54 53 66 72];
……

Question: Did the person place all the containers facing upwards or downwards? Caption Model: (C)
Choices: (A) upwards (B) downwards (C) mixed Caption + Box Model: (C)

Figure A10. Failure case on Perception Test. The model cannot see the state of the mugs from either the captions or the bounding boxes. So it does not whether the mugs are upwards or downwards.



Frame Captions

Frame 0: A person sitting on a chair with their legs crossed. The person is wearing a pair of black shoes and appears to be in the process of putting on socks. A knife is placed nearby, possibly for cutting the socks. There are two apples in the scene……
……
Frame 238: A person sitting on a chair, wearing a pair of sneakers. They are in the process of tying their shoelaces, with a fork and a knife nearby. The person is surrounded by a few apples, with one placed close to the left side of the chair, another on the right side……
……
Frame 357: A woman sitting on a bench with her legs crossed. She is wearing a pair of sneakers and appears to be tying her shoelaces. There are a few apples placed on the ground near her, and a fork is also visible in the scene. A kettle can be seen in the background……

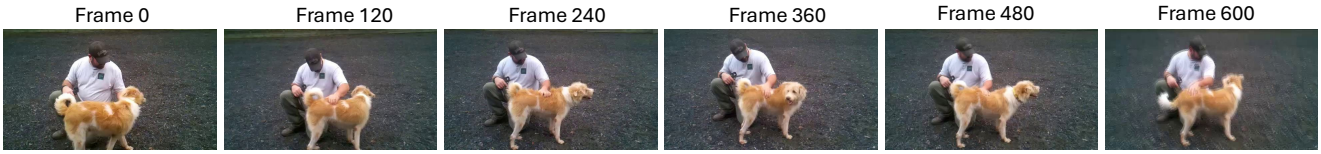
Object Bounding Boxes

(Object 2) shoe lace - frame 0 [35 65 51 89] frame 119 [35 66 43 88] frame 238 [50 65 53 72] frame 357 [47 61 51 74];
(Object 9) shoe lace - frame 0 [41 64 50 83] frame 119 [42 68 48 77] frame 238 [43 70 49 78] frame 357 [42 64 50 77];
(Object 10) shoe - frame 0 [49 63 58 90] frame 119 [49 70 58 90] frame 238 [49 70 58 91] frame 357 [49 62 58 90];
(Object 12) shoe - frame 0 [40 59 50 88] frame 119 [40 59 50 88] frame 238 [41 59 50 89] frame 357 [41 59 50 88];
……

Question: Is there something unusual about the way the person ties the shoe laces?
Choices: (A) The person ties correctly the left shoe lace, but not the right shoe lace.
(B) The person ties the shoe laces normally.
(C) The person ties the lace of the left shoe to the lace of the right shoe.

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

Figure A11. Failure case on Perception Test. Both the captions and the bounding boxes cannot tell if the shoe laces are tied normally. This suggests that our model has difficulty in recognizing the appearance of the objects.



Frame Captions

Frame 0: A man kneeling down on the ground next to a brown and white dog. The man appears to be petting the dog, showing affection and care for the animal. The dog is positioned to the left of the man, and they are both situated on a dirt surface……
……
Frame 600: A man kneeling down on the ground next to a brown dog. The man is petting the dog, showing affection and care. The dog is positioned to the left of the man, and both of them are on a dirt surface. The man appears to be wearing a hat……

Object Bounding Boxes

(Object 0) adult - frame 0 [22 10 57 94] frame 30 [20 1 56 86] frame 60 [26 2 60 84] frame 90 [27 7 61 89] frame 120 [26 16 59 88] frame 150 [25 11 54 83] frame 180 [21 6 52 78] frame 210 [18 7 48 80] frame 240 [18 9 50 80] frame 270 [20 10 52 80] frame 300 [21 9 52 79]……
(Object 1) dog - frame 0 [24 47 67 100] frame 30 [27 34 66 100] frame 60 [31 38 68 100] frame 90 [34 40 70 100] frame 120 [37 46 71 100] frame 150 [35 44 69 100] frame 180 [33 39 69 98] frame 210 [29 39 68 98] frame 240 [29 44 70 100] frame 270 [31 44 70 99] ……

Question: Why is the man kneeling down on the floor?
Choices: (A) feed the dog (B) crawling around (C) let kids walk through
(D) fell down (E) pet the dog

Caption Model: (E)
Box Model: (A)

Figure A12. Failure case on NExT-QA. Although the detection and tracking algorithm can tell that there are an adult and a dog in the video, their actions cannot be inferred from the object bounding boxes. The captioning model can capture the person's action so that the model with captions as inputs correctly answers this question.