



LEGO: Language Enhanced Multi-modal Grounding Model

Zhaowei Li^{1,2}, Qi Xu¹, Dong Zhang², Hang Song¹, Yiqing Cai¹,
Qi Qi¹, Ran Zhou¹, Junting Pan¹, Zefeng Li¹, Van Tu Vu¹,
Zhida Huang¹, Tao Wang¹

¹ByteDance Inc, ²Fudan University
lizhaowei126@gmail.com

<https://lzw-lzw.github.io/LEGO.github.io/>

Image Grounding



Please describe the image and include the object positions.



Three dogs are sitting on the floor in front of three plates of food. They seem to be begging for food, possibly from their owners.

Video Grounding



Please describe this video.

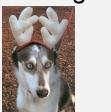


In the video, a young girl is sitting on a bed and reading a book. She is wearing glasses and seems to be enjoying the book. The girl is surrounded by toys. The video shows the girl's reading skills and her engagement with the book.

When did the baby push up the glasses and stretch her body in the video?

The baby push up the glasses and stretch her body in (0.16, 0.36).

Multi-modal Understanding



What is funny about this image?



The image features a dog wearing a red Santa hat and a reindeer antler headband. The dog is standing on a dirt surface in a field. The hat and antler headband are placed on the dog's head, creating a festive and playful appearance. The dog's attire adds a touch of holiday spirit to the scene.

Sound Localization



(An audio of dog barking.)

Where is the sound's point of emission in this image? Include the object positions in [x0, y0, x1, y1] format.



(An audio of a saw cutting through wood.)

In the image, a dog is running through a grassy field. The dog is running with its mouth open, creating a sound of running and barking. The scene is set in a park or a field, with trees in the background.

Multi-modal Understanding



Write a concise summary of the video clip.

In this captivating video, a drone takes us on a mesmerizing journey over the city of Florence, Italy. The aerial footage showcases the breathtaking beauty of the city, with its iconic landmarks and stunning architecture. As the drone soars through the sky, we are treated to a bird's-eye view of the city's famous bridges, including the Ponte Vecchio and the Ponte Santa Trinita.

Figure 1: LEGO is an end-to-end unified multi-modal grounding model. We showcase the performance of LEGO across a range of multi-modal tasks, including: 1)Image Grounding, 2)Video Grounding, 3)Sound Localization, 4)Multi-modal Understanding.

Abstract

Multi-modal large language models have demonstrated impressive performance across various tasks in different modalities. However, existing multi-modal models primarily emphasize capturing global information within each modality while neglecting the importance of perceiving local information across modalities. Consequently, these models lack the ability to effectively understand the fine-grained details of input data, limiting their performance in tasks that require a more nuanced understanding. To address this limitation, there is a compelling need to develop models that enable fine-grained understanding across multiple modalities, thereby enhancing their ap-

plicability to a wide range of tasks. In this paper, we propose **LEGO**, a language enhanced multi-modal grounding model. Beyond capturing global information like other multi-modal models, our proposed model excels at tasks demanding a detailed understanding of local information within the input. It demonstrates precise identification and localization of specific regions in images or moments in videos. To achieve this objective, we design a diversified dataset construction pipeline, resulting in a multi-modal, multi-granularity dataset for model training. The code, dataset, and demo of our model can be found at <https://github.com/lzw-lzw/LEGO>.

1 Introduction

Recently, significant advancements have been made in large language models (LLMs), which have demonstrated superior performance in a variety of natural language processing tasks (Touvron et al., 2023; Zeng et al., 2022). These models offer promise for building general-purpose artificial intelligence due to their comparable performance and strong generalizability. Building on the capabilities of LLMs, research on multi-modal large language models (MLLMs) has also advanced, enabling understanding across a broader range of modalities. By leveraging LLMs as a universal interface and training on multi-modal instruction data, MLLMs integrate the capabilities of existing multi-modal models into the LLM framework. Representative works include vision language models like LLaVA (Liu et al., 2023) and MiniGPT-4 (Zhu et al., 2023), these models align visual features obtained from image encoders with LLM embedding space through visual instruction tuning, facilitating tasks like image captioning and visual question answering.

However, the existing MLLMs primarily focus on global information and do not consider fine-grained local information in the multi-modal data. This limitation restricts their application in tasks requiring a more detailed understanding. To address these limitations, recent works (Peng et al., 2023; Chen et al., 2023b; You et al., 2023) in the field of MLLMs have explored techniques that enable finer alignment and understanding of inputs. Some approaches directly represent coordinates in textual form and train the model to understand the locations, while others introduce additional position-aware modules to comprehend local information. By considering local-level information, these models exhibit enhanced performance in tasks that demand precise multi-modal understanding at the region or object level.

The above-mentioned approach provides insights into fine-grained understanding, but it is limited to grounding tasks within the image modality. There is still much to explore in terms of fine-grained understanding of other modalities such as video, audio, and more. To address this gap, in this paper, we propose **LEGO**, a language enhanced multi-modal grounding model. LEGO is an end-to-end unified large language model that facilitates multi-modal and multi-granularity information understanding. Specifically, our model

employs modality-specific adapters to map feature representations from individual modality encoders to the embedding space of LLMs, enabling effective multi-modal understanding. To incorporate spatial and temporal information, we represent spatial coordinates and timestamps directly as textual numbers, avoiding the need for vocabulary expansion.

We design a three-stage training strategy for LEGO. In the first stage, we align each pre-trained multi-modal encoder with the LLM embedding space using multiple adapters. In the second stage, we aim to enable the model to grasp fine-grained information, including spatial coordinates and temporal segments. In the third stage, we perform cross-modal instruction tuning to refine the model’s responses. However, obtaining fine-grained multi-modal instruction data is challenging. Therefore, we construct a multi-modal training dataset by employing various construction methods tailored to different data sources, covering a wide range of scenarios involving multi-modal interactions. By employing our diversified dataset, we enhance LEGO’s ability to understand and ground multi-modal information at various levels of granularity.

To summarize, our contributions are as follows:

- We introduce LEGO, an end-to-end multi-modal grounding model that accurately comprehends inputs and possesses robust grounding capabilities across multi modalities, including images, audios, and videos.
- To address the issue of limited data, we construct a diverse and high-quality multi-modal training dataset. This dataset encompasses a rich collection of multi-modal data enriched with spatial and temporal information, thereby serving as a valuable resource to foster further advancements in this field.
- Extensive experimental evaluations validate the effectiveness of the LEGO model in understanding and grounding tasks across various modalities.

2 Related Work

Multi-modal Large Language Models(MLLMs). Recently, large language models (LLMs) represented by GPTs(Brown et al., 2020; OpenAI, 2023) and LLaMA (Touvron et al., 2023) have received extensive attention from researchers. These models

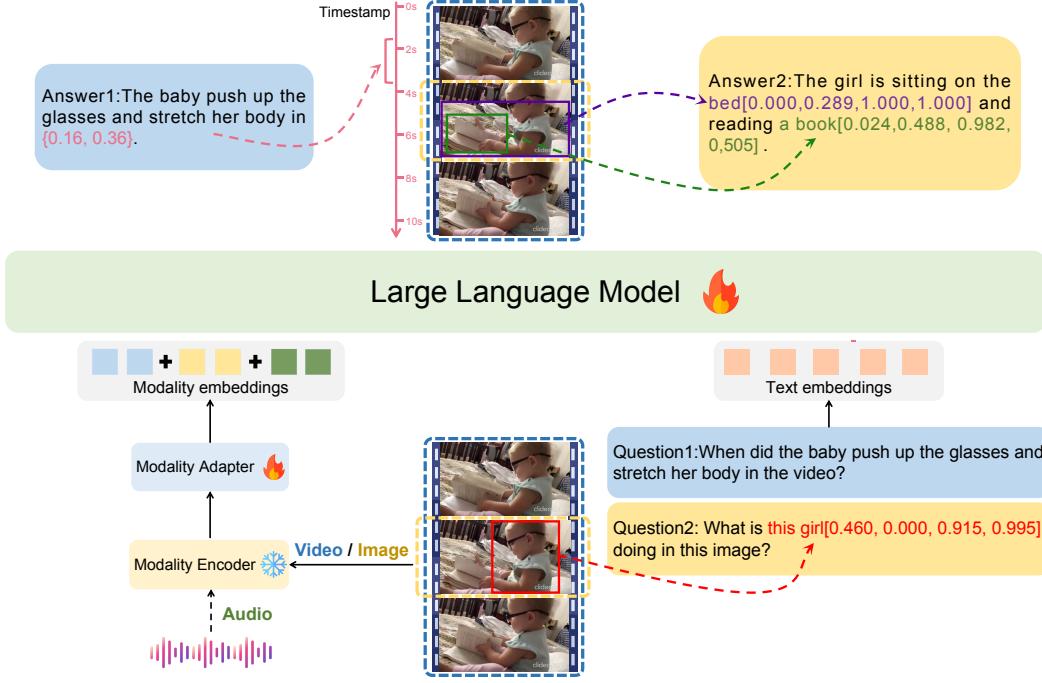


Figure 2: The overall structure of LEGO involves separate encoders and adapters for each modality (video, image, audio, etc.). The input from each modality is processed through its independent encoder and adapter, resulting in modality embeddings. The figure demonstrates two examples using video and image modalities. Blue boxes represent video as input, while yellow boxes represent image as input.

have achieved remarkable performance in various natural language processing tasks by leveraging pre-training on extensive web text corpora, thereby acquiring substantial language knowledge and understanding abilities.

Moreover, substantial progress has been made in the field of multi-modal LLMs, which extend the support for multi-modal input and output beyond language. State-of-the-art MLLMs typically fine-tune pre-trained LLMs with multi-modal instructions, training on a large number of modal-text pairs to enable understanding across multiple modalities. For example, in the image modality, models such as LLaVA (Liu et al., 2023), MiniGPT-4 (Zhu et al., 2023), and mPLUG-Owl (Ye et al., 2023) map image embeddings obtained from image encoders into the LLM space. Similarly, video LLMs like Video-LLaMA (Zhang et al., 2023c) and Valley (Luo et al., 2023), as well as speech LLMs like SpeechGPT (Zhang et al., 2023b) and LLaSM (Shu et al., 2023), acquire multi-modal understanding capabilities through similar approaches. In X-LLM (Chen et al., 2023a), each modality is processed independently through dedicated branches for multi-modal input processing. Pandagpt (Su et al., 2023) employs a unified embedding space trained by ImageBind (Girdhar et al.,

2023) to facilitate joint understanding of various modal inputs, including images, videos, audios, and more. On the other hand, Next-GPT (Wu et al., 2023) achieves both multi-modal input and output by connecting different modality-specific diffusion models at the output end. However, despite their ability to leverage global multi-modal information during training, these models often fail to adequately capture details. Consequently, their performance may be suboptimal when tackling tasks that require a more detailed understanding.

MLLMs For Grounding Task. In recent research, there has been a focus on training visual MLLMs to achieve fine-grained image understanding and visual grounding by leveraging fine-grained information. Approaches such as KOSMOS-2 (Peng et al., 2023) and Shikra (Chen et al., 2023b) achieve this by incorporating position coordinates into the training data, enabling MLLMs to understand the location information within images. On the other hand, approaches like NExT-Chat (Zhang et al., 2023a) and Ferret (You et al., 2023) enhance perception of fine-grained information by introducing additional image local feature extraction modules. Both categories of approaches have shown promising results in fine-

grained image perception. These advancements demonstrate the effort made to incorporate fine-grained information into MLLMs, enabling them to achieve more detailed understanding and grounding across different modalities. However, the aforementioned models are limited to the image modality, there is still a need for further exploration of fine-grained understanding in other modalities such as video and audio. BuboGPT(Zhao et al., 2023) enables cross-modal interaction between image, audio, and language, facilitating fine-grained understanding of different modalities. In contrast, our proposed model supports the understanding of multi-modal information at different granularities, with a unified end-to-end structure. It can be applied to complex multi-modal interactive tasks such as image grounding, video temporal grounding. To the best of our knowledge, this is the first large language model that achieves multi-modal and fine-grained perception across modalities.

3 Methods

In this section, we will present the structure of our model, including the branches for different modalities and the spatial-temporal representation methods. We will also discuss the pipeline for constructing our multi-modal dataset. Finally, we will introduce the three-stage training process of the LEGO model.

3.1 Overall Architecture

Figure 2 illustrates the overall architecture of the LEGO model. Each modality’s inputs are processed through specific encoders to extract features. These features are then mapped to the LLMs’ embedding space using several adapters. The modular design and adapter-based architecture allow seamless integration of new encoders to handle additional modalities, such as point clouds and speech, making our model easily extendable.

3.1.1 Image Branch

For the input image,we employ the pre-trained CLIP visual encoder ViT-L/14 (Radford et al., 2021) to extract image features. Similar to (Liu et al., 2023), we select the features before the last Transformer layer as the embedding of the image. The encoded image is represented as a fixed-length embedding vector $I \in R^{K_I \times d_I}$. To align the image representation with the embedding space of LLMs, we use a simple linear layer to map the obtained features to the dimensions of LLMs. The mapped

embeddings are then concatenated with text embeddings and used as input to LLMs, similar mapping methods are adopted for other modalities.

3.1.2 Video Branch

Considering the inherent imformation redundancy in videos and memory limitations, we adopt a uniform sampling approach for feature extraction. The video adapter incorporates a temporal position encoding and a video Q-Former with the same structure as the Q-Former in BLIP-2 (Li et al., 2023a) to aggregate video information. By combining these components, the video branch effectively captures both content and temporal information, enabling comprehension of visual content while preserving temporal cues for multi-modal grounding tasks. Specially, we uniformly sample M frames form the video. Each frame is processed by the image encoder, resulting in the representation $V_f = [v_1, v_2, \dots, v_M]$ where $v_i \in R^{K_f \times d_f}$ represents the K_f -length d_f -dimensional image embedding of the i -th frame. To preserve temporal information, we introduce temporal position encoding to the representation, and then the enhanced representation is fed into the Video Q-former, which generates k_V video embedding vectors of dimensions d_V . These vectors form the representation $V \in R^{k_V \times d_V}$ for the entire video.

3.1.3 Audio Branch

The audio branch follows a structure similar to the video branch. Since audio contains less information, we utilize ImageBind (Girdhar et al., 2023) as the audio encoder. ImageBind processes 2-second audio clips with a 16kHz sampling rate and converts them into spectrograms using 128 mel-spectrogram bins. We sample N 2-second audio segments from the original audio and transform each segment into a vector using ImageBind, resulting in the initial representation $A_s = [a_1, a_2, \dots, a_N]$, where $a_i \in R^{K_s \times d_s}$ represents the embedding of the i -th aduio segment. To incorporate temporal information, we incorporate temporal position encoding into A_s . Lastly, we obtain a fixed-length audio representation sequence, denoted as $A \in R^{k_A \times d_A}$, using the audio Q-former.

3.1.4 Spatial-temporal Representation

In our approach, the bounding box in an image is represented by four relative coordinate values: $[x_1, y_1, x_2, y_2]$. These values correspond to the upper left corner point $[x_1, y_1]$ and the

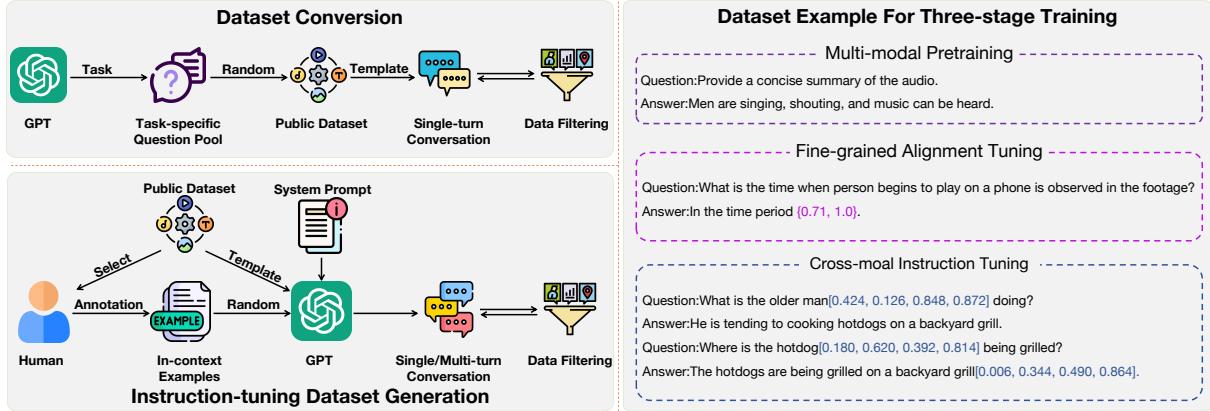


Figure 3: Our two-stage dataset construction pipeline and the dataset examples for three-stage training. The data of the first two training stages are obtained by dataset conversion, and the data of the third stage are obtained by instruction-tuning dataset generation.

lower right corner point $[x_2, y_2]$ of the bounding box. Each coordinate value is preserved with three decimal places. For example, a representation could be $[0.128, 0.565, 0.204, 0.720]$. We concatenate this textual representation after the description related to the bounding box. For instance, a sentence could be "Please describe this region $[0.128, 0.565, 0.204, 0.720]$ in detail." Similarly, for representing timestamps, we use two two-digit decimals to indicate the relative values of the start and end times of a time segment with respect to the total duration. To differentiate it from spatial positional information, we use the format $\{t_1, t_2\}$ to represent timestamps. For instance, a sentence could be "Describe this video clip $\{0.14, 0.32\}$ please." This representation allows us to train the model without requiring additional vocabulary expansion or separate training. This approach enabled the model to develop a comprehensive understanding of both spatial and temporal information, enhancing its perception of fine-grained information. Specific examples of instruction-tuning dataset are shown in Figure 3.

3.2 Dataset Construction Pipeline

To address the scarcity of fine-grained multi-modal datasets, we develop a large-scale, multi-modal, and multi-granularity dataset by leveraging publicly available datasets and tools. The construction pipeline of our dataset involves several key processes. Firstly, we gather a diverse range of multi-modal data from various sources, including images, videos, and audios. Subsequently, we apply multi-granularity transformations on the dataset to capture fine-grained information, such as objects

within images and video segments. To ensure the dataset's quality, we meticulously filter the generated data to ensure it adheres to the expected format and structure. Specifically, our dataset construction pipeline consists of the following two parts.

Dataset Conversion In this stage, we focus on constructing a multi-modal dataset for modality alignment and fine-grained alignment. The dataset quality is relatively lower as it is primarily obtained through converting publicly available datasets. As depicted in the upper left part of Figure 3, we provide task descriptions to GPT-3.5 to generate a task-specific question pool. For each data sample, a question is randomly selected from the pool and templates are used to convert the sample's format, resulting in question-answer pairs in a single-turn dialogue format. Finally, we filtered the generated dataset to ensure its quality. For image modality, we utilize the LLaVA-pretrain-595K (Liu et al., 2023) dataset for modality alignment. For fine-grained alignment, we selected specific datasets, including RefCOCO (Kazemzadeh et al., 2014), RefCOCO+ (Kazemzadeh et al., 2014), RefCOCOg (Mao et al., 2016) and Visual Genome (Krishna et al., 2017b). For the video modality, the Valley-Pretrain-703K(Luo et al., 2023) dataset is used for modality alignment, while the Charades-STA (Gao et al., 2017) dataset is employed for fine-grained alignment. For audio modality, the WaveCaps (Mei et al., 2023) dataset is utilized for training.

Instruction-tuning Dataset Generation This stage aims to generate a high-quality instruction-tuning dataset for LEGO models to better under-

stand and follow human instructions. As illustrated in the lower left part of Figure 3, we select a subset of publicly available datasets for human annotation to create in-context examples. It assists in guiding GPT-3.5 to follow similar patterns when generating instruction-tuning dataset. Subsequently, task-specific system prompts and randomly selected examples are input to GPT-3.5 to generate single-turn or multi-turn conversations. Finally, we perform data filtering to ensure dataset quality. For the image modality, we construct multi-turn dialogue datasets using the Flickr30K Entities (Plummer et al., 2015) dataset, including detailed descriptions and conversations. To enhance the model’s fine-grained reasoning capability, we utilize the VCR (Zellers et al., 2019) dataset to construct a reasoning dataset with location information. Regarding the video modality, we constructed datasets with temporal information by incorporating datasets from various video tasks such as DiDeMo (Anne Hendricks et al., 2017) and Activitynet Captions (Krishna et al., 2017a), along with other relevant sources. The training data for the audio modality is constructed based on the Clotho (Drossos et al., 2020) dataset to create an instruction fine-tuning dataset. Additionally, the model is trained using VGGSS (Chen et al., 2021) dataset to enhance cross-modal interaction capabilities. For more detailed information about the datasets, please refer to appendix A.

3.3 Training

The LEGO model utilizes the Vicuna1.5-7B as the language foundation model. Our training approach consists of three stages: multi-modal pre-training, fine-grained alignment tuning, and cross-modal instruction tuning.

Stage1 Multi-modal Pretraining. In this stage, we focus on enabling the model to comprehend multi-modal inputs through pretraining. The training data used in this stage primarily consists of public datasets or converted datasets mentioned in section 3.2. During the training process, the LLM model and the encoders for each modality remain frozen, only the adapter parameters for each modality are learned, while the LLM model and modality encoders remain frozen. Training is conducted with a batch size of 64, a learning rate of 2e-3, and is completed in approximately 10 hours using 8 A100 GPUs for LEGO-7B.

Stage2 Fine-grained Alignment Tuning. In the second stage, we conduct fine-grained alignment tuning, where the objective is to enhance the model’s understanding of spatial coordinates and timestamps. The training data used in this stage is the dataset we constructed which contains the spatial-temporal representation mentioned in Section 3.1.4. During the training process, the encoders for each modality are frozen, while the LLM and adapters are trained. Training is performed with a batch size of 32, a learning rate of 2e-5, and takes around 40 hours using 8 A100 GPUs for LEGO-7B.

Stage3 Cross-modal Instruction Tuning. In the third stage, we conduct cross-modal instruction tuning to further refine the model using generated data. This stage aims to enable the model to generate responses that better align with human preferences and improve multi-modal interactions. The instruction-tuning dataset generated as described in Section 3.2 is used for training. During training, similar to the second stage, the encoders for each modality are frozen, while the LLM and adapters are trained. The model is trained for one epoch with a batch size of 32 and a learning rate of 1e-5. Training on 8 A100 GPUs for LEGO-7B is completed in approximately 8 hours.

During the training process, in order to prevent catastrophic forgetting in subsequent training stages, we adopt a sampling strategy that incorporates training data from previous stages. The three-stage training process employs a consistent training objective as follows:

$$L(\theta) = -\mathbb{E}_{(x,y) \sim D_{\text{current}}} [\log p(y|x)] \\ - \alpha \cdot \mathbb{E}_{(x,y) \sim D_{\text{previous}}} [\log p(y|x)]$$

where D_{current} denotes the dataset in current training stage, D_{previous} denotes the dataset in previous training stage and α denotes the sampling rate of the dataset in previous stage. In the first training stage, α is set to 0.

4 Experiments

4.1 Image Grounding

To assess the image grounding capability of the LEGO model, we conduct experiments on the widely used Reference Expression Understanding (REC) task. The REC task requires the model to accurately locate the bounding box

Models	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
UNITER	81.41	87.04	74.17	75.90	81.45	66.70	74.02	68.67
MDETR	86.75	89.58	81.41	79.52	84.09	70.62	81.64	80.89
UniTAB	86.32	88.84	80.61	78.70	83.22	69.48	79.96	79.97
KOSMOS-2	52.32	57.42	47.26	45.48	50.73	42.24	60.57	61.65
Shikra	87.01	90.61	80.24	81.60	87.36	72.12	82.27	82.19
NExT-Chat*	85.5	90.0	77.9	77.2	84.5	68.0	80.1	79.8
Ferret*	87.49	91.35	82.45	80.78	87.38	73.14	83.93	84.76
LEGO(224)	83.85	89.07	81.90	79.55	86.63	71.73	80.96	81.52
LEGO(336)	86.88	90.85	82.22	80.74	86.79	72.08	81.29	83.05

Table 1: Performance comparison on the referring expression comprehension(REC) task. The numbers in bracket following the model names represent the resolution of the input images. "*" indicates that the model employs additional image region perception module.

corresponding to a given text reference expression within an image. Our experiments involve three datasets: RefCOCO(Kazemzadeh et al., 2014), RefCOCO+(Kazemzadeh et al., 2014), RefCOCOg(Mao et al., 2016). The baselines used for comparing include two types of models: the previous end-to-end multi-modal models UNITER(Chen et al., 2020), MDETR(Kamath et al., 2021), UniTAB(Yang et al., 2022), and the LLM-based multi-modal grounding models KOSMOS-2(Peng et al., 2023), Shikra(Chen et al., 2023b), NExT-Chat(Zhang et al., 2023a) and Ferret(You et al., 2023). For LEGO model, we use a unified prompt like "Output the coordinate of <exp>", where "<exp>" represents the reference expression to be localized. A predictions bounding box is considered correct if the intersection-over-union (IoU) between the predicted bounding box and the GT box is greater than 0.5.

The performance of the LEGO model and other comparative models on the REC task is presented in Table 1. LEGO demonstrates competitive performance across all datasets and performs comparably to approaches that incorporate an additional image region perception module.

4.2 Video Grounding

To evaluate the video grounding capability of LEGO, we conduct experiments on the temporal video grounding task. For the task, we employed datasets from Charades-STA (Gao et al., 2017). The predicted time segments are compared with the corresponding ground truth time segments to calculate the IoU. The evaluation met-

Models	Charades-STA	
	R@1(IoU=0.5)	R@1(IoU=0.7)
Video-LLaMA	2.1	0.6
VideoChat	3.3	1.3
Valley	4.7	1.6
LEGO	29.6	11.9

Table 2: Performance comparison on the temporal grounding task.

ric used is "R@1, IoU = m", which measures the percentage of correctly retrieved moments with an IoU greater than m. We set the values of m as 0.5, 0.7 to assess different levels of accuracy. The baseline models we compare are the following multi-modal large language models: Video-LLaMA(Zhang et al., 2023c), VideoChat(Li et al., 2023b) and Valley(Luo et al., 2023). As shown in Table 2, LEGO exhibits excellent performance in the task of temporal video grounding compared to the previously video MLLMs, which primarily focuses on entire video understanding.

4.3 Ablation Study

4.3.1 Effect of Image Resolution

To investigate the effect of different input image resolutions on the model's grounding ability, we conducted ablation experiments, as illustrated in Table 1. As the input image resolution increased from 224×224 to 336×336 , the model exhibited improvements across all datasets in the REC task. This result is consistent with our expectations, as

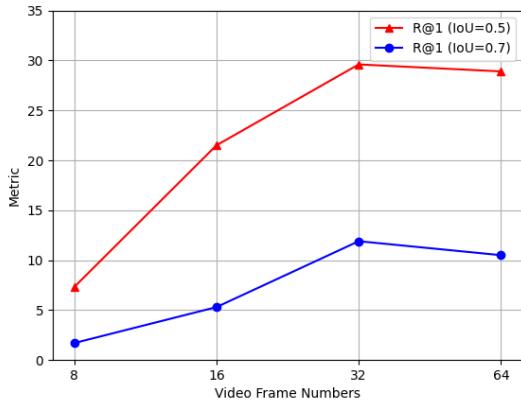


Figure 4: Results of different video frame numbers on the Charades-STA dataset.

higher image resolutions augment object detection capabilities, particularly for smaller objects and object boundaries

4.3.2 Effect of Video Frame Numbers

Considering the redundancy inherent of video information, we employ a frame sampling approach for video modeling. This raises the question: how many frames should be extracted from a video? To address this, we vary the number of sampled video frames and perform separate training, while keeping other settings consistent. Figure 4 presents the results obtained on the Charades-STA dataset. As the number of video frames increases from 8 to 32, there is a consistent improvement on the metric for the Charades-STA dataset, but when the number of frames further increases, the effect begins to decrease. This phenomenon is specific to the Charades-STA dataset, which consists of relatively short videos with an average length of 30.59 seconds. When only 8 frames are sampled, a large amount of valuable information in the video is lost, resulting in low temporal grounding accuracy. With an increasing number of frames, the grounding accuracy improves, but further increasing the number of frames will not provide additional benefits and only leads to increased training time.

5 Discussion

Language Hallucination. Similar to previous studies, our model is built upon the pretrained large language model, which may have certain limitations and occasionally exhibit hallucination phenomena. It is possible for the model to generate content that does not exist in the input or provide

incorrect knowledge.

Sampling Strategy. Due to computational memory constraints, LEGO adopts a sampling approach when processing videos and audios. However, this method inevitably results in some loss of crucial information, especially when dealing with longer videos. One future research direction is to explore better modeling approaches for longer videos and minimize information loss.

Grounding Ability. Despite achieving promising results in multi-modal grounding tasks, LEGO currently lacks the capability to output more fine-grained grounding results such as segmentation masks. In future work, we plan to expand the grounding tasks to support a broader range of grounding requirements.

6 Conclusion

In this paper, we propose a unified end-to-end multi-modal grounding model called LEGO. Through training on a diverse multi-modal and multi-granularity dataset, LEGO achieves better perception of multi-modal inputs and demonstrates improved performance on tasks requiring fine-grained understanding. To address the scarcity of relevant data, we create a multi-modal grounding dataset encompassing various modalities, tasks, and granularities. To encourage further advancements in this field, we will make our model, code, and dataset openly accessible. In future work, we aim to extend LEGO to accommodate additional input and output modalities while exploring more sophisticated grounding methods.

References

- Lisa Anne Hendricks, Oliver Wang, Eli Shechtman, Josef Sivic, Trevor Darrell, and Bryan Russell. 2017. Localizing moments in video with natural language. In *Proceedings of the IEEE international conference on computer vision*, pages 5803–5812.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.
- Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu. 2023a. X-ilm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages. *arXiv preprint arXiv:2305.04160*.

- Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. 2021. Localizing visual sounds the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16867–16876.
- Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. 2023b. Shikra: Unleashing multimodal llm’s referential dialogue magic. *arXiv preprint arXiv:2306.15195*.
- Yen-Chun Chen, Linjie Li, Licheng Yu, Ahmed El Kholy, Faisal Ahmed, Zhe Gan, Yu Cheng, and Jingjing Liu. 2020. Uniter: Universal image-text representation learning. In *European conference on computer vision*, pages 104–120. Springer.
- Konstantinos Drossos, Samuel Lipping, and Tuomas Virtanen. 2020. Clotho: An audio captioning dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 736–740. IEEE.
- Jiyang Gao, Chen Sun, Zhenheng Yang, and Ram Nevatia. 2017. Tall: Temporal activity localization via language query. In *Proceedings of the IEEE international conference on computer vision*, pages 5267–5275.
- Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. 2023. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15180–15190.
- Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. 2021. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790.
- Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara Berg. 2014. Referitgame: Referring to objects in photographs of natural scenes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 787–798.
- Ranjay Krishna, Kenji Hata, Frederic Ren, Li Fei-Fei, and Juan Carlos Niebles. 2017a. Dense-captioning events in videos. In *Proceedings of the IEEE international conference on computer vision*, pages 706–715.
- Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. 2017b. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *arXiv preprint arXiv:2301.12597*.
- Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. 2023b. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*.
- Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.
- Ruipu Luo, Ziwing Zhao, Min Yang, Junwei Dong, Minghui Qiu, Pengcheng Lu, Tao Wang, and Zhongyu Wei. 2023. Valley: Video assistant with large language model enhanced ability. *arXiv preprint arXiv:2306.07207*.
- Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. 2016. Generation and comprehension of unambiguous object descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 11–20.
- Xinhao Mei, Chutong Meng, Haohe Liu, Qiuqiang Kong, Tom Ko, Chengqi Zhao, Mark D Plumbley, Yuexian Zou, and Wenwu Wang. 2023. Wavcaps: A chatgpt-assisted weakly-labelled audio captioning dataset for audio-language multimodal research. *arXiv preprint arXiv:2303.17395*.
- OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. 2023. Kosmos-2: Grounding multimodal large language models to the world. *arXiv preprint arXiv:2306.14824*.
- Bryan A Plummer, Liwei Wang, Chris M Cervantes, Juan C Caicedo, Julia Hockenmaier, and Svetlana Lazebnik. 2015. Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models. In *Proceedings of the IEEE international conference on computer vision*, pages 2641–2649.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR.
- Yu Shu, Siwei Dong, Guangyao Chen, Wenhao Huang, Ruihua Zhang, Daochen Shi, Qiqi Xiang, and Yemin Shi. 2023. Llasm: Large language and speech model. *arXiv preprint arXiv:2308.15930*.
- Yixuan Su, Tian Lan, Huayang Li, Jialu Xu, Yan Wang, and Deng Cai. 2023. Pandagpt: One model to instruction-follow them all. *arXiv preprint arXiv:2305.16355*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro,

- Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. 2023. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*.
- Zhengyuan Yang, Zhe Gan, Jianfeng Wang, Xiaowei Hu, Faisal Ahmed, Zicheng Liu, Yumao Lu, and Lijuan Wang. 2022. Unitab: Unifying text and box outputs for grounded vision-language modeling. In *European Conference on Computer Vision*, pages 521–539. Springer.
- Qinghao Ye, Haiyang Xu, Guohai Xu, Jiabo Ye, Ming Yan, Yiyang Zhou, Junyang Wang, Anwen Hu, Pengcheng Shi, Yaya Shi, Chaoya Jiang, Chenliang Li, Yuanhong Xu, Hehong Chen, Junfeng Tian, Qian Qi, Ji Zhang, and Fei Huang. 2023. *mplug-owl: Modularization empowers large language models with multimodality*.
- Haoxuan You, Haotian Zhang, Zhe Gan, Xianzhi Du, Bowen Zhang, Zirui Wang, Liangliang Cao, Shih-Fu Chang, and Yinfei Yang. 2023. Ferret: Refer and ground anything anywhere at any granularity. *arXiv preprint arXiv:2310.07704*.
- Rowan Zellers, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. From recognition to cognition: Visual commonsense reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6720–6731.
- Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b: An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.
- Ao Zhang, Liming Zhao, Chen-Wei Xie, Yun Zheng, Wei Ji, and Tat-Seng Chua. 2023a. Next-chat: An lmm for chat, detection and segmentation. *arXiv preprint arXiv:2311.04498*.
- Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023b. Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities. *arXiv preprint arXiv:2305.11000*.
- Hang Zhang, Xin Li, and Lidong Bing. 2023c. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*.
- Yang Zhao, Zhijie Lin, Daquan Zhou, Zilong Huang, Jiashi Feng, and Bingyi Kang. 2023. Bubogpt: Enabling visual grounding in multi-modal llms. *arXiv preprint arXiv:2307.08581*.
- Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

A Dataset Appendix

A.1 Data Source

In Table 3, we provide a comprehensive list of the datasets used in constructing our training dataset. This includes the data utilized in the alignment stages as well as the data employed for instruction-tuning.

A.2 Dataset Conversion Template

Table 4 presents the templates utilized for various tasks during the training process. For the sake of demonstration, we provide three examples of instructions for each task.

A.3 Prompt for Instruction-tuning Dataset Generation

We use GPT-3.5 to generate the instruction-tuning dataset. In Figure 5, we provie the prompt we used to generate the detailed description dataset. In Figure 6, we provie the prompt we used to generate the conversation dataset. In Figure 7, we provie the prompt we used to generate the video grounding instruction-tuning dataset.

B More Visualization

To evaluate the performance of LEGO in multi-modal grounding tasks, we present visualizations in Figure 8, Figure 9, and Figure 10, showcasing the capability of the LEGO model in image, video, and audio grounding tasks, respectively. For more examples, please refer to our project page: <https://lzw-lzw.github.io/LEGO.github.io/>, where we provide additional illustrations and demonstrations.

Task	Dataset
Image Captioning	LLaVA-Pretrain-585k
REC/REG	Refcoco, RefCOCOg, Refcoco+, Visual Genome
Object Attribute	Visual Genome
Object Relation	Visual Genome
Image Instruction Tuning	LLaVA-Instrtuct-150k, VCR
Video Captioning	Valley-Pretrain-703k
Temporal Grounding	Didemo, Charades-STA, ActivityNet Captions
Audio Captioning	Wavecaps
Video Instruction Tuning	Valley-Instruct-73k, Videochat-11k
Audio Instruction Tuning	Clotho
Sound Localization	VGGSS

Table 3: The publicly available dataset sources used for constructing our training data.

Task	Template examples
Image Captioning	<p>Provide a brief description of the given image.</p> <p>Write a terse but informative summary of the picture.</p> <p>Share a concise interpretation of the image provided.</p>
REG	<p>What object is present within the specified region<region>?</p> <p>Can you identify the item within the region<region>?</p> <p>Describe the object located within the region<region>.</p>
REC	<p>In this image, where is <exp> located?</p> <p>Can you identify the position of <exp> within this image?</p> <p>Please describe the location of <exp> in this image.</p>
Object Attribute	<p>What color is this <exp>?</p> <p>How many <exp> are visible within this image?</p> <p>How mang <exp> are there in the image?</p>
Video Captioning	<p>Relay a brief, clear account of the video shown.</p> <p>Offer a succinct explanation of the footage presented.</p> <p>Present a compact description of the clip's key features.</p>
Temporal Grounding	<p>Describe the content shown in the video clip<time> of this video.</p> <p>What can you tell me about the video segment<time> in this video?</p> <p>Can you provide a description of the video snippet<time>?</p>
Event Detection	<p>When did <event> occur in the video?</p> <p>Tell me the timestamp when <event> happened.</p> <p>At what time does <event> take place in the video?</p>
Audio Captioning	<p>Analyze the audio and provide a description of its content.</p> <p>Examine the audio and describe the different sounds present.</p> <p>Provide a detailed summary of the auditory elements in the audio clip.</p>
Sound Localization	<p>What is the cause of the sound in this given image?</p> <p>Can you pinpoint the source of the sound in this image?</p> <p>Describe the location of the sound's origin in this image.</p>

Table 4: Instruction templates for different tasks we used during the dataset conversion phases. The templates include several placeholders: '<region>' represents the coordinates of a region in an image, '<exp>' represents the expression correspond to an image region, '<time>' represents a time segment in a video, and '<event>' represents an event to be located in a video. During the dataset conversion process, these placeholders are replaced with corresponding information.

System Message

You are an AI visual assistant that can analyze a single image. You receive several sentences, each describing the same image you are observing. In addition, specific object locations within the image are given, along with detailed coordinates. These coordinates are in the form of bounding boxes, represented as [x1, y1, x2, y2], with floating numbers ranging from 0 to 1. These values correspond to the top left x, top left y, bottom right x, and bottom right y.

The task is to create an accurate description related to the image based on the information. The description should involve mentioning the position of objects in the image. The position can be represented in the format [x1, y1, x2, y2]. Description should be accurate and concise, limited to 100 words.

Here are some additional requirements about generated descriptions:

1. In description, you need to mention bounding box coordinates to refer to some objects or regions, instead of directly say the object name or describing the regions in text.
2. Avoid introducing objects that do not exist in the original descriptions and avoid including excessive subjective perceptions to prevent creating illusions.
3. Only describe what you are certain about, and avoid providing descriptions that may be ambiguous or inaccurate.
4. The boxes provided in different sentences may have some coordinates that are the same or very close, which could be because different expressions refer to the same object. You should analyze and avoid describing a single object within a bounding box as multiple distinct entities.
5. The number of words in the output description should not exceed 100 words. Keep it as concise as possible. You do not need to include everything in several descriptions. Choose the output you think is the most important.

In-context Example

User:

```
{'sentence': 'Two young guys with shaggy hair look at their hands while hanging out in the yard .',
'boxes': {'Two young guys': [[0.474, 0.248, 0.655, 0.668], [0.589, 0.218, 0.781, 0.744]], 'shaggy hair': [[0.538, 0.248, 0.616, 0.31], [0.592, 0.226, 0.718, 0.29]], 'their hands': [[0.517, 0.366, 0.592, 0.404], [0.471, 0.394, 0.571, 0.448]]}}
```

```
{'sentence': 'Two young , White males are outside near many bushes .',
'boxes': {'Two young , White males': [[0.474, 0.248, 0.655, 0.668], [0.589, 0.218, 0.781, 0.744]], 'many bushes': [[0.0, 0.438, 0.631, 0.944], [0.826, 0.428, 0.994, 0.672]]}}
```

```
{'sentence': 'Two men in green shirts are standing in a yard .',
'boxes': {'Two men': [[0.474, 0.248, 0.655, 0.668], [0.589, 0.218, 0.781, 0.744]], 'green shirts': [[0.619, 0.286, 0.769, 0.486], [0.517, 0.31, 0.649, 0.47]]}}
```

```
{'sentence': 'A man in a blue shirt standing in a garden .',
'boxes': {'A man': [[0.589, 0.218, 0.781, 0.744]], 'a blue shirt': [[0.619, 0.286, 0.769, 0.486]]}}
```

```
{'sentence': 'Two friends enjoy time spent together .',
'boxes': {'Two friends': [[0.474, 0.248, 0.655, 0.668], [0.589, 0.218, 0.781, 0.744]]}}
```

Assistant:

In the image, two young men[0.474, 0.248, 0.655, 0.668][0.589, 0.218, 0.781, 0.744] with shaggy hair[0.538, 0.248, 0.616, 0.31][0.592, 0.226, 0.718, 0.29] are seen in a yard surrounded by bushes[0.000, 0.438, 0.631, 0.944][0.826, 0.428, 0.994, 0.672]. They are focused on their hands[0.517, 0.366, 0.592, 0.404][0.471, 0.394, 0.571, 0.448]. It seems that they enjoy time spent together.

Figure 5: The system message and in-context example we used for generating detailed description dataset.

System Message

You are an AI visual assistant that can analyze a single image. You receive several sentences, each describing the same image you are observing. In addition, specific object locations within the image are given, along with detailed coordinates. These coordinates are in the form of bounding boxes, represented as [x1, y1, x2, y2], with floating numbers ranging from 0 to 1. These values correspond to the top left x, top left y, bottom right x, and bottom right y.

The task is to create question-answer pairs related to the image based on the information. The conversation should involve mentioning the position of objects in the image. The position can be represented in the format [x1, y1, x2, y2]. The questions and answers should be limited to 50 words.

Here are some additional requirements about generated conversations:

1. In conversation, you need to mention bounding box coordinates to refer to some objects or regions, instead of directly say the object name or describing the regions in text.
2. Avoid introducing objects that do not exist in the original descriptions and refrain from including excessive subjective perceptions to prevent creating illusions.
3. Only describe what you are certain about, and avoid providing descriptions that may be ambiguous or inaccurate.
4. The boxes provided in different sentences may have some coordinates that are the same or very close, which could be because different expressions refer to the same object. You should analyze and avoid describing a single object within a bounding box as multiple distinct entities.
5. The questions and answers should be limited to 50 words. Keep it as concise as possible.

In-context Example

User:

```
{'sentence': 'A child in a pink dress is climbing up a set of stairs in an entry way .',
'boxes': {'A child': [[0.019, 0.302, 0.285, 0.678]], 'a pink dress': [[0.019, 0.38, 0.227, 0.596]], 'a set of stairs': [[0.0, 0.636, 0.304, 0.958]], 'an entry way': [[0.099, 0.052, 0.288, 0.614]]}}
```

```
{'sentence': 'A little girl in a pink dress going into a wooden cabin .',
'boxes': {'A little girl': [[0.019, 0.302, 0.285, 0.678]], 'a pink dress': [[0.019, 0.38, 0.227, 0.596]]}}
```

```
{'sentence': 'A little girl climbing the stairs to her playhouse .',
'boxes': {'A little girl': [[0.019, 0.302, 0.285, 0.678]], 'the stairs': [[0.0, 0.636, 0.304, 0.958]], 'her playhouse': [[0.069, 0.014, 0.997, 0.774]]}}
```

```
{'sentence': 'A little girl climbing into a wooden playhouse .',
'boxes': {'A little girl': [[0.019, 0.302, 0.285, 0.678]], 'a wooden playhouse': [[0.069, 0.014, 0.997, 0.774]]}}
```

```
{'sentence': 'A girl going into a wooden building .',
'boxes': {'A girl': [[0.019, 0.302, 0.285, 0.678]]}}
```

Assistant:

```
{'question': 'What clothes is the girl[0.019, 0.302, 0.285, 0.678] wearing?',
'answer': 'The girl is wearing a pink dress[0.019, 0.380, 0.227, 0.596].'}
```

```
{'question': 'Where is the little girl[0.019, 0.302, 0.285, 0.678] going?',
'answer': 'She is climbing into a wooden playhouse[0.069, 0.014, 0.997, 0.774].'}
```

```
{'question': 'What is the child[0.019, 0.302, 0.285, 0.678] doing?',
'answer': 'She is climbing up a set of stairs[0.000, 0.636, 0.304, 0.958].'}
```

```
{'question': 'Why is this girl climbing the stairs?',
'answer': 'She is going to enter her playhouse[0.069, 0.014, 0.997, 0.774].'}
```

Figure 6: The system message and in-context example we used for generating conversation dataset.

System Message

You are an AI visual assistant that can analyze a single video. You will receive descriptions of several segments of this video along with corresponding timestamps for each description, indicating the time range of the corresponding video segment. These timestamps are in the form of bounding boxes, represented as {t1,t2} with floating numbers ranging from 0 to 1. These values correspond to the start time of the video segment is represented by t1, and the end time is represented by t2. Design a conversation between you and a person asking about this video. Ask three to ten questions and give corresponding answers. The answers should be in a tone that a visual AI assistant is seeing the video and answering the question.

Here are some additional requirements about generated questions and answers:

1. Only include questions that have definite answers.
2. Always answer as if you are directly looking at the video.
3. Make the question as diverse as possible. Include questions asking about the visual content of the video, including the events within the video segments, the order in which different events occur, and other related details.
4. Avoid introducing objects that do not exist in the original descriptions and refrain from including excessive subjective perceptions to prevent creating illusions.
5. The number of words in the question and answer should not exceed 50.

In-context Example

User:

```
{'A young woman is seen standing in a room and leads into her dancing.': '{0.01, 0.24}'}
```

```
{'The girl dances around the room while the camera captures her movements.': '{0.21, 0.74}'}
```

```
{'She continues dancing around the room and ends by laying on the floor.': '{0.68, 0.96}'}
```

Assistant:

```
{"question": "What happens during the time period{0.01,0.24}?",  
 "answer": "A young woman is seen standing in a room and leads into her dancing."}
```

```
{"question": "When does the women start dancing around the room?",  
 "answer": "The girl starts dancing around the room in {0.21,0.74}"}
```

```
{"question": "How does the girl end her dance routine?",  
 "answer": "The girl ends her dance routine by laying on the floor, which occurs in {0.68,0.96}.")}
```

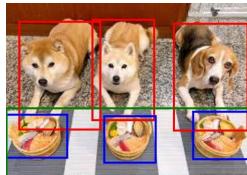
Figure 7: The system message and in-context example we used for generating video grounding dataset.



Please describe the image and include the object positions.







Three dogs are sitting on the floor in front of three plates of food. They seem to be begging for food, possibly from their owners.





What breed is the dog, please tell me something about it.

The dog in the middle is a Shiba Inu. Shibas are a Japanese breed of dog, characterized by their distinctive, thick double coat, which comes in various colors such as red, white, or black. They are known for their independent and sometimes aloof nature, but they are also loyal and affectionate with their owners. In the image, the Shiba Inu is sitting next to the other two dogs, which are of different breeds, and it is eating food from its plate.

Figure 8: Examples of LEGO's grounding capability in image modality.



Write a summary of this video.



In this video, a dog is seen attempting to ride a skateboard. The dog is initially hesitant and unsure of how to balance on the skateboard. However, with determination, the dog manages to find its balance and starts riding the skateboard. The dog's movements are a bit wobbly at first, but it gradually gains confidence and starts to ride smoothly. The dog's excitement is evident as it enjoys the experience of riding the skateboard. Despite the initial hesitation, the dog's perseverance pays off, and it successfully rides the skateboard for a short distance. This video showcases the dog's determination and ability to overcome challenges, as well as its enjoyment of the skateboarding experience.

Please give me the timestamp when the dog turns around to look at the camera?



The dog turns around to look at the camera in {0.09,0.15}.

When does the dog use its paws to push off and ride the skateboard?



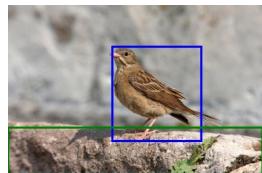
The dog uses its paws to push off and ride the skateboard in {0.42,0.58}.

Figure 9: Examples of LEGO's grounding capability in video modality.



(An audio of a bird singing.)

Where is the sound's point of emission in this image? Include the object positions in [x0, y0, x1, y1] format.

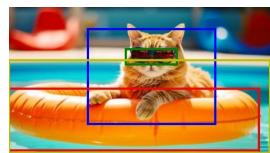


The sound's position of emission is from [a small bird](#) perched on [a rock](#).



(An audio of a cat meowing.)

Where in the image is the sound in the audio coming from? Output the coordinates of the sound source.



The sound in the audio is coming from the [orange cat](#) wearing [sunglasses](#). The cat is laying on [a float](#) in [a pool](#).



Figure 10: Examples of LEGO's grounding capability in audio modality.