A Appendix

A.1 Dataset Details

In the section, we document further details on the dataset construction process. We also provide some statistics about our dataset.

A.1.1 Dataset Size

We document the number of images sampled from each source dataset and the number of conversations generated in Table A1. Additionally, we visualize the distribution of the number of rounds for each dataset in Fig. A1.

Datasets	Training Set			Validation Set		
	# of Convs	# of Images	Max Rounds	# of Convs	# of Images	Max Rounds
RefCOCO(+/g)	55188	27674	18	4263	2701	17
Visual Genome	367674	94221	2	40980	10524	2
PACO-LVIS	40827	40827	19	2178	2178	16
LVIS	71388	71255	17	13898	13898	18
Pascal Panoptic Part	4577	4577	17	4690	4690	18
ADE20K	59784	20196	1	5943	200	1
COCO-Stuff	340127	118205	1	14461	4999	1
Attributes-COCO	49036	36413	1	5000	2566	1
ReasonSeg	1326	239	1	200	200	1
MRSeg (hard)	22470	22470	1	1988	1988	1

Table A1: Statistics of our MRSeg dataset, including the number of overall conversations, number of images, and the maximum rounds of conversations for each dataset after processing through our dataset pipeline.

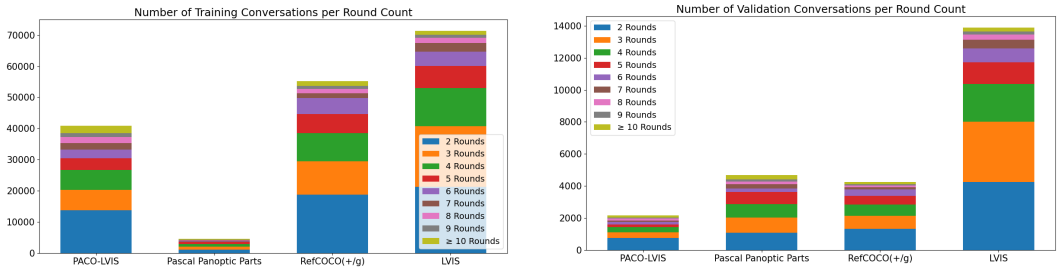

Figure A1: Bar-plot visualization for training and validation conversations count at different number of rounds for multi-round datasets. There are very conversations with a large number of rounds.

A.1.2 Conversation Generation Pipeline

We employ different strategies to generate natural-language conversation for different source datasets. Specifically, our dataset is generated using a combination of the following methods:

- ***Hierarchical relationships based on PACO-LVIS and Pascal Panoptic Part***: In these queries, the model is asked to segment objects which are a sub-part of some output of a previous round. From each image, we randomly sample between one and four instances, and for each instance, we randomly sample between one and four parts. We initiate queries about the instance followed by questions targeting the parts of each respective instance. For Pascal Panoptic Part, we only use objects and their parts on a instance level and not a semantic segmentation level to avoid ambiguity. For both PACO-LVIS and Pascal Panoptic Part, we refer to previous round outputs with it's actual caption, e.g. ``the knife'' with probablility 50%. With the other 50% we refer to the previous round output as ``<instance i>'' or ``<the output of round i>''.
- ***Positional relationships based on Refcoco(+/g) and LVIS***: These conversations task the system with segmenting objects based on their positional relationships to the outputs from previous rounds.We randomly sample between 2 to 18 annotations per image. For each selected annotation, we either generate a query about the object itself or generate a query involving an object from

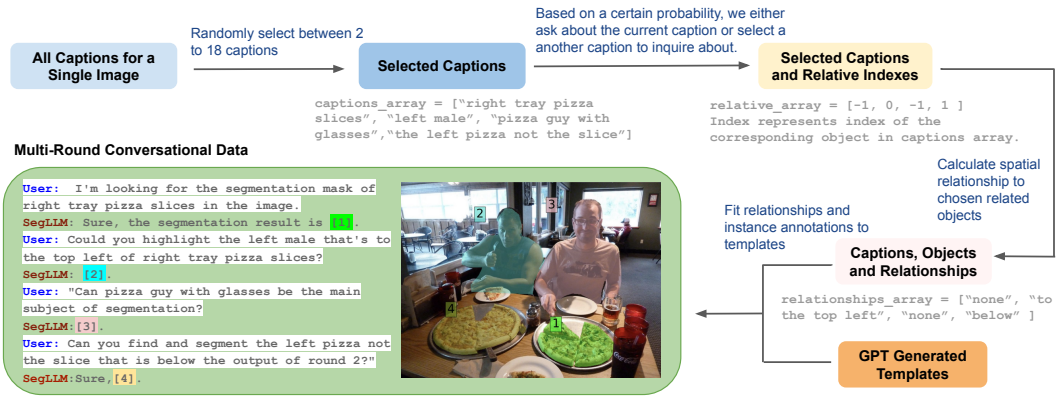

Figure A2: Pipeline for generating multi-round conversational data for RefCOCO(+/g) in MRSeg.

previously processed instances, focusing on their relative positions calculated from their bounding box coordinates. For RefCOCO(+/g), multiple annotations may be selected for the same instance due to multiple captions available per instance.For LVIS, we select annotations where only one or two objects of that class appear in the image. When two objects of the same class are present, we detail their relative positions and add location descriptions to their captions to prevent ambiguity. We specifically choose instances not categorized under COCO classes to diversify the dataset's class variety. The probability for each round to query about an object itself is $1/3$, otherwise, we query about the current object with a reference to a previous round's output and their relative position. To assign the positional relationships, we use compare the edge and center position of the bounding boxes for the two instance we are trying to assign a relationship to. There are 9 total possible positions two instances can have (the same as, overlapping with, to the left/right, above/below, to the top/bottom left/right of). Similar to Hierarchical Queries,we refer to previous round outputs with it's actual caption, e.g. ``the woman on the left'' with probablility 50%. With the other 50% we refer to the previous round output as ``<instance i>'' or ``<the output of round i>''. A detailed pipeline for how RefCOCO(+/g) dataset is sampled can be see in Fig. A2

- ***MR Seg(hard)***: For each RefCOCO image, we identify cases where there are two instances of the same class within the image. From these, we select a pair of instances and construct two single-round conversations. Given two instances, X and Y, of the same class in the image, we create the following conversations:
  - Conv 1: [IMAGE] [ENCODE X] Please segment the other <class name> → Sure, [DECODE Y]
  - Conv 2: [IMAGE] [ENCODE Y] Please segment the other <class name> → Sure, [DECODE X]

  We have 10 different templates for the training and 5 templates validation/test for MR Seg(hard).
- ***Interactional relationships based on Visual Genome***: We adopt Visual Genome (VG), utilizing its relationship annotations to construct conversations that emphasize interactional dynamics rather than merely positional relationships. We sample up to four relationships per image. Each relationship prompts a two-round conversation: the first round involves segmenting the subject, and the second round involves segmenting an object based on its relationship to the subject. Since VG also only provides bounding box labels, we generate masks for selected instances using SAM.

A.1.3 Details of GPT4 Usage

We prompt GPT-4 models for generating captions for attribute-based descriptions as well as for cleaning grammar errors in our dataset. The detailed instructions and specific model we used can be found in Table A2 and Table A3. For the attribute-based description, we crop COCO images to only contain the specified instance, feeding the cropped image and it's class name to GPT to generate a description. For language correction, we found that grammar correction is often erroneous but can be a lot of accurate if we go through the data twice to double check.

```
payload = {
    "model": "gpt-4-turbo-2024-04-09",
    "messages": [
      {
        "role": "user",
        "content": [
          {
            "type": "text",
            "text": f"Can you focus on describing the {class_name} in
                the image? Can you format your output in a two item
                array, such that the first index is an abstract
                description without any class name, such as 'has a pizza
                sitting on top of it' or 'is wearing a beige t-shirt'
                and the second index is the exact classname for the
                object, such as 'a dining table' or 'a man'."
          },
          {
            "type": "image_url",
            "image_url": {
              "url": f"{base64_image}",
              "detail": "low"
            }
          }
        ]
      }
    ],
    "max_tokens": 200
  }
```

Table A2: Our full prompt to the GPT-4-turbo-2024-04-09 model for generating abstract descriptions

A.1.4 More discussions on demo outputs

In Fig. A3, example **A** illustrate the necessity of our Mask-Encoding Scheme, to avoid the ambiguity that may arise in cases where multiple instances of the same class are present in the image. Round 2 and round 3 in example **A** show that without our mask encoding mechanism to supply information about the person segmented from round 1, since there are multiple laptops and chairs present in the image, confusion arises as to which specific laptop or chair the user is referring to in the query prompt. Therefore, without the guiding information from the mask encoding, LISA seems to naively guess the incorrect laptop in round 2, and does not generate a comprehensible segmentation mask in round 3. In contrast, the mask encoding guides our model to correctly segment the requested objects. Similarly, in round 4 and round 6, our model was able to successfully segment the keyboard of the laptop from round 3 and the person setting on the chair from round 5.

This phenomenon is again demonstrated in **B** in Fig. A3. Since there are two women, both carrying bags and holding an umbrella in the image, our Mask-Encoding Scheme again resolves this the ambiguity and allows the user to conveniently specify the bag and the umbrella requested in round 2 and round 3 are carried and held by the person from round 1. As before, the awareness of previous round outputs enables our model to segment the correct objects, whereas LISA guesses the incorrect objects due to the lack of this awareness.

Example **C** demonstrates that our model is not limited to multi-round prompting, and can produce accurate segmentation results via direct, single-round prompts as well. In the indirect case, we first ask the model to segment the dog during the first round of the conversation. Then, in the second round, we ask a follow up question to guide the model to segment the Frisbee that is caught by the dog from round 1. However, tin the direct case, we straight away ask for the Frisbee that is caught by the dog. In comparison, our model succeeds in both the direct and indirect case, whereas LISA fails to segment the correct Frisbee instance in either cases. This shows that our multi-round comprehension capability is not a limitation but an addition.

```
Round 1:
response = client.chat.completions.create(
            model="gpt-4o-2024-05-13",
            response_format={ "type": "json_object" },
            messages=[
                {"role": "system", "content": "You are a helpful
                    assistant designed to output JSON."},
                {"role": "user", "content": f"Can you fix any errors and
                    make the sentence sound like natural English, and
                    provide our output in a dictionary of format
                    'corrected'=CORRECT_SENTENCE? here is the sentence I
                    want you to correct, '{sent}'"}
            ]
        )
Round 2:
    response = client.chat.completions.create(
            model="gpt-4o-2024-05-13",
            response_format={ "type": "json_object" },
            messages=[
                {"role": "system", "content": "You are a helpful
                    assistant designed to output JSON"},
                {"role": "user", "content": f"Here is the original
                    sentence: '{sent}'. Here is the corrected sentence:
                    '{corrected_sent}'. Does the corrected sentence have
                    the same meaning as the original? If yes, please
                    output ['Same', 'None']. If no, please output
                    ['Different',
                    '<corrected_with_same_meaning_as_original>']."}
            ]
        )
```

Table A3: Out full prompt to the gpt-4o-2024-05-13 model for grammar correction. We use a two-round approach, feeding GPT's first round answer back to itself to be self-corrected.

Lastly, we note that round 3 and round 6 of example **A**, round 2 and round 3 of example **B** and round 2 of example **C** demonstrate our model's understanding of *interactional relationships* as introduced in Sec. 4.1 and round 4 demonstrates the *hierarchical relationship* introduced in Sec. 4.1.

B Details of Comparison with Lisa

Since Lisa does not naively support multi-round training, to ensure fairness, we employed two different approaches:

- Approach One: We substitute the mask and bounding box encoding tokens of the reference instance with the word "mask". For example, a query in MR-RefCOCO dataset "Segment the person to the left of <mask> <box>." would be converted to "Segment the person left to the mask."
- Approach Two: We substitute the mask and bounding box encoding tokens with the description of the reference instance. For example, a query in MR-RefCOCO dataset "Segment the person to the left of <mask> <box>." would be converted to "Segment the person left to the dog chasing after a butterfly." (where <mask> <box> are encoding tokens of the reference instance "the dog chasing after a butterfly")

We report results on MR-RefCOCO in Table A4. SegLLM outperforms both alternative approaches 1 and 2. Furthermore, we find that LISA performs worse using approach 2 compared to approach 1, despite the inclusion of the description of the reference instance. We suspect that this may be due to LISA being trained on data that focuses on 1 instance, hence the presence of description for two instances, the target and the reference instance, may cause more confusion than guidance. Regardless, in our main table Table 1, we report LISA's performance on our MR-RefCOCO/+/g benchmark using the best approach for LISA, approach 2.

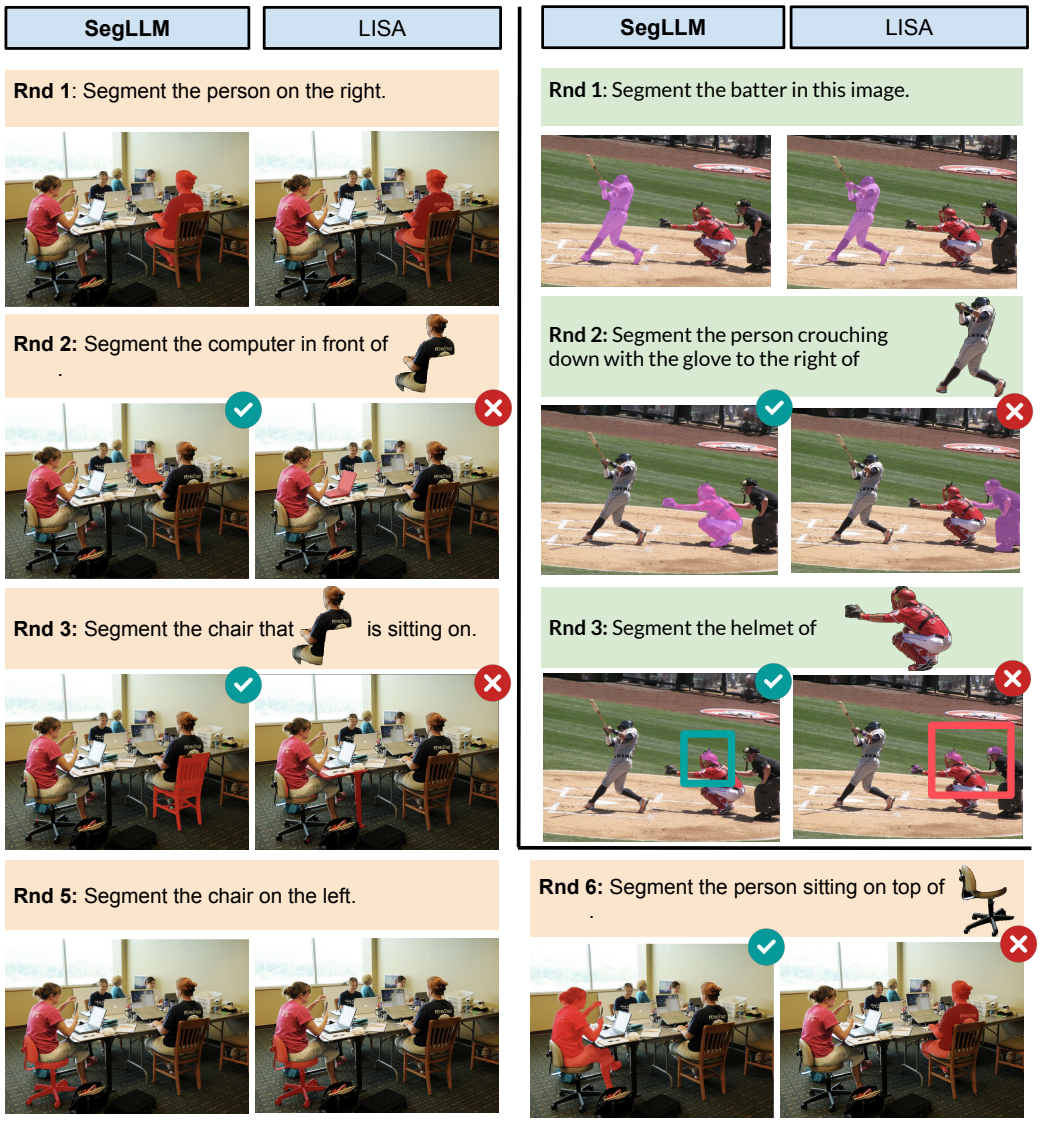

Figure A3: Additional side-by-side comparison with LISA. This shows that without awareness of segmentation outputs from previous rounds, LISA struggles to identify the correct instance requested by the user, when there is ambiguity.

Table A4: Comparison of SegLLM, Lisa(Approach 1), and Lisa(Approach 2) on MR-RefCOCO evaluation set.

	SegLLM	Approach 1	Approach 2
round 2	81.9	60.6	55.9
round 3	81.7	58.9	54.7
round 4	78.4	61.3	56.7
round 5	80.3	61.0	57.8
round 6	74.5	60.7	57.7
round 7	69.3	54.4	45.6
round 8	70.5	51.9	50.3

C LICENSE

We makes use the following models: CLIP (MIT license), LLAMA 2 (Llama 2 Community License Agreement), Vicuna (Apache2 license). BLIP-2 (BSD-3-Clause license)

We use the following dataset COCO (Attribution-NonCommercial-ShareAlike 4.0 Internationa), RefCOCO (Apache-2.0 license), Visual Genome (Creative Commons Attribution 4.0 International License.), PACO (MIT License), Pascal-Panoptic-Parts (Apache-2.0 license), LIVIS (CC BY 4.0 + COCO license).

D LIMITATION

One limitation is that our model can only output a single mask, hence we are only able to perform segmentation on an instance level rather than a semantic level. Another limitation is that when the text input is ambiguous, our model may randomly select a possible output instead of asking which specific output is desires or output all possible options. This may be caused by the training data which is slightly noisy due to being converted from datasets not necessary for referring segmentation.

E BROADER IMPACTS

Our paper imposes positive broader impacts. It can act as a educational tools. One can employ our model to demonstrate the relationship between objects by clearly segmenting them, this can help second-language speakers or children learn the meaning of different relationships, for example. It can also be beneficial for scientific research or environment monitoring. Our model can help detect extremely small objects autonomously simply with an image and text prompt.