THINK TO GROUND: IMPROVING SPATIAL REASONING IN LLMS FOR BETTER VISUAL GROUNDING

Karun Sharma* **& Vidushee Vats***
Department of Computer Science
Bennett University
{mailto.karun.py,vatsvidushee}@gmail.com

ABSTRACT

Visual grounding tasks involve identifying objects and references in an image based on text input. A model is required to locate the objects and their relationships, as well as to understand the image to accurately ground the target. Specialized models like Owl-ViT and Grounding DINO often fail to predict correct results for queries involving complex spatial information. In this paper, we propose a Spatial Thinking and Reasoning Dataset for visual grounding and a framework that uses existing detection models to identify candidate objects. These models provide coordinates and other attributes to a large language model (LLM) for spatial reasoning to determine the correct target. Recent closed-source models like GPT-4o achieve approximately 86% accuracy, while open-source models perform significantly worse, reaching only about 60% accuracy in our experiments. To improve this, we use reinforcement learning to fine-tune a 3B open-source model on our dataset, achieving 77% accuracy, comparable to closed-source models.

1 INTRODUCTION

The task of visual grounding involves identifying, localizing and bounding a specific object or region based on a textual description by understanding the underlying relationship in an expression (Ma et al., 2024). Primarily, the three factors that influence how well a model can perform this task are: What is the object in focus in a given query? How can we interpret what is shown in an image? and How can we locate the primary target in image? (Xiao et al., 2024). For example in "Orange above the box", after identifying all of the "oranges" and the "box", we need to locate the target using the spatial-relation present in the sentence i.e "above".

Existing Models like ImageBind (Girdhar et al., 2023), Grounding DINO (Liu et al., 2023), Owl-ViT (Minderer et al., 2022) are trained on extensive datasets but still provide incorrect predictions and are not able to locate the target successfully in the image in zero-shot scenarios. Owl-ViT uses CLIP (Radford et al., 2021) as its multi-modal backbone which lacks in specialized tasks like Visual Grounding. However, these models still provide reliable candidates when used for simple object detection task, which we use to our leverage to pass to an LLM for spatial reasoning based on the coordinates of the candidate objects. Currently only closed-source LLMs like GPT-4o (OpenAI, 2024), Gemini-Exp are capable of spatial reasoning and thinking; to overcome this, we construct spatial reasoning dataset and use that to improve spatial understanding in open-source LLMs using Reinforcement Learning(RL), making them comparable to closed-source LLMs on this task.

Developing from above, we create a framework for improving the ability of LLMs to locate the objects in an image by forcing it to think and carefully reason when it's given a spatial-relation. We have done the following: 1) Developed a framework for grounding images accurately 2) Constructed a Spatial Thinking and Reasoning Dataset for Visual Grounding 3) Finetuned a 3B base open-source LLM on our dataset to introduce spatial reasoning by using RL for self-improvement.

Third person from front on right

Detected Candidates from Florence-2:

Detected Person at location [0, 76.4, 225, 480]
Detected Person at location [442, 7, 530, 337]
Detected Person at location [361, 86, 453, 310]
Detected Person at location [444, 70, 640, 477]

Target: [361, 86, 453, 310]

RL Trained Qwen2.5 - 3B(Ours)

<think>
First, I'll parse the bounding boxes:
1. [0, 76.4, 225, 480]
2. [442, 7, 530, 337]
3. [361, 86, 453, 310]
4. [444, 70, 640, 477]
Next, I'll analyze the target description: 'Third person from front on right'.
This implies:
The target is a person, It is third in line, It is on the right side

Considering the bounding boxes:
- The third person from the front would be the third smallest x-coordinate.
- The person is on the right, so the third person should have the largest
x-coordinate among the third smallest.
Let's compare the x-coordinates:
1. [0, 76.4, 225, 480] - x1 = 0, 2. [442, 7, 530, 337] - x1 = 442
3. [361, 86, 453, 310] - x1 = 361, 4. [444, 70, 640, 477] - x1 = 444

The third smallest x-coordinate is 361, which corresponds to
the third bounding box [361, 86, 453, 310].
Now, let's check the y-coordinates:
1. [0, 76.4, 225, 480] - y1 = 76.4 , 2. [442, 7, 530, 337] - y1 = 7
3. [361, 86, 453, 310] - y1 = 86, 4. [444, 70, 640, 477] - y1 = 70

The third smallest y-coordinate is 86, which corresponds to
the third bounding box [361, 86, 453, 310].
Finally, let's check the size of the bounding box:
- The area of the 3rd bounding box is (453 - 361) * (310 - 86) = 92 * 224 = 20,528
- The area of the other bounding boxes are:
  - First: (225 - 0) * (480 - 76.4) = 225 * 403.6 = 9,0540
  - Second: (530 - 442) * (337 - 7) = 88 * 330 = 29,040
  - Fourth: (640 - 444) * (477 - 70) = 196 * 407 = 80,072
The third bounding box has the smallest area, which is consistent with it being the third person from the front.
Therefore, the coordinates for the target object are [361, 86, 453, 310]. </think>
<answer>[361, 86, 453, 310]</answer>

GPT-4o

It looks like you want to determine the correct bounding box based on the given caption and system rules.

Let's go through the reasoning step by step.

Bounding Boxes Given:
1. [0, 76.4, 225, 480]
2. [361, 86, 453, 310]
3. [442, 7, 530, 337]
4. [444, 70, 640, 477]

Caption: 'Third from front on right'
- "On the right" means we need to focus on bounding boxes with higher x1 values.
- Ordering from rightmost to leftmost based on x1
  - [444, 70, 640, 477] (x1 = 444)
  - [442, 7, 530, 337] (x1 = 442)
  - [361, 86, 453, 310] (x1 = 361)
  - [0, 76.4, 225, 480] (x1 = 0)

- Third from the front (assuming front refers to being most rightward):
  - First: [444, 70, 640, 477]
  - Second: [442, 7, 530, 337]
  - Third: [361, 86, 453, 310]

Answer:
<think>
Based on the sorting by rightmost x1 values, the third from the right is [361, 86, 453, 310]
</think>

<answer> [361, 86, 453, 310] </answer>

Figure 1: RL trained Qwen2.5-3B (left) is able to think and perform spatial reasoning, with thoughts similar to the close-sourced GPT-4o (right).

2 METHOD

Grounding Framework: First, we pass images and captions from RefCOCO to Florence-2 to get the coordinates of the objects mentioned in the prompt. Next, to obtain additional attributes such as depth, we use the Depth Anything model, extracting the depth of objects based on the coordinates from Florence-2 (Xiao et al., 2023). Finally, we provide the coordinates, depth values and captions for GPT-4o to aid in spatial reasoning and to predict the target boundary box from the candidate coordinates. Refer to Section A.2 of the Appendix for detailed prompts.

Spatial Thinking and Reasoning Dataset: The dataset has two parts, one for reinforcement learning (RL) and one for supervised fine-tuning (SFT). The RL dataset is created using 2,200 samples from RefCOCO, where we extract candidate coordinates with our grounding framework and take the corresponding ground truth bounding boxes. In addition, we pass these coordinates to o1-mini and DeepSeek-R1 to obtain reasoning steps for constructing the SFT dataset. We only use the dataset for the RL and the SFT dataset can be used to further improve the model.

RL Training to introduce spatial reasoning capabilities in small LLMs: We train Qwen2.5-3B on our RL dataset using Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with two rule based reward functions to force the model to think and generate correct coordinates. Training for two epochs took approximately 10 hours on 4×H100 GPUs. While LLMs as judges can be used to give reward to model's responses, this significantly increases training time and resource

*Equal Contribution by both authors

requirements. For details about training, refer to Section A.1 of the appendix. Due to a large compute requirement for RL we only finetuned on 2200 images.

3 RESULTS

We observed that the Qwen2.5-3B model, fine-tuned on our Spatial Thinking and Reasoning Dataset, is capable of complex reasoning and demonstrates performance comparable to GPT-4o, as shown in Figure1. Additional examples are provided in SectionA.3 in the Appendix.

For testing purposes, we randomly sampled 200 images from RefCOCO. Our framework achieved an accuracy of 86% when using GPT-4o as the LLM. Notably, as shown in Table1, the RL fine-tuned Qwen2.5-3B model achieved a remarkable 77% accuracy using greedy decoding, while the base Qwen2.5-3B model achieved only 58% accuracy with greedy decoding and 60% accuracy when employing beam search (with `num_beams=15`) and gave unreliable and almost random outputs without following the format requested. The inference time of our RL tuned model increased by 2x from base Qwen model because of it's reasoning and thinking capability. For fairness, we only evaluate accuracy of all the models on coordinates predicted and not on the format of the output.

Table 1: Performance comparison of our grounding framework when using various open-source LLMs for spatial reasoning

Metric	**Qwen2.5-3B**	**Qwen2.5-3B**	**RL Qwen2.5-3B(Ours)**
Decoding	Greedy	Beam Search (num_beams=15)	Greedy
Inference time	x	2x	2x
Accuracy	58%	60%	**77%**

Reproducibility: Our code and online demo will be open-source, and we will share our model weights and dataset for transparency.

REFERENCES

Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15180–15190, June 2023.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chun yue Li, Jianwei Yang, Hang Su, Jun-Juan Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *European Conference on Computer Vision*, 2023. doi: 10.48550/arXiv.2303.05499.

Tao Ma, Bing Bai, Haozhe Lin, Heyuan Wang, Yu Wang, Lin Luo, and Lu Fang. When visual grounding meets gigapixel-level large-scale scenes: Benchmark and approach. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22119–22128, June 2024.

Matthias Minderer, Alexey Gritsenko, Austin Stone, Maxim Neumann, Dirk Weissenborn, Alexey Dosovitskiy, Aravindh Mahendran, Anurag Arnab, Mostafa Dehghani, Zhuoran Shen, Xiao Wang, Xiaohua Zhai, Thomas Kipf, and Neil Houlsby. Simple open-vocabulary object detection with vision transformers. *arXiv preprint arXiv: 2205.06230*, 2022.

OpenAI. Gpt-4o system card. *arXiv preprint arXiv: 2410.21276*, 2024.

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 Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 2021. URL `http://proceedings.mlr.press/v139/radford21a.html`.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv: 2402.03300*, 2024.

Bin Xiao, Haiping Wu, Weijian Xu, Xiyang Dai, Houdong Hu, Yumao Lu, Michael Zeng, Ce Liu, and Lu Yuan. Florence-2: Advancing a unified representation for a variety of vision tasks. *Computer Vision and Pattern Recognition*, 2023. doi: 10.1109/CVPR52733.2024.00461.

Linhui Xiao, Xiaoshan Yang, Xiangyuan Lan, Yaowei Wang, and Changsheng Xu. Towards visual grounding: A survey. *arXiv preprint arXiv: 2412.20206*, 2024.

A Appendix

A.1 Training details

We used 4 H100 GPUs to train the Qwen2.5-3B Instruct for 550 Steps on 2200 Images, it took 10 hours to complete the training run with batch size 1 and Grad Accumulation of 8.

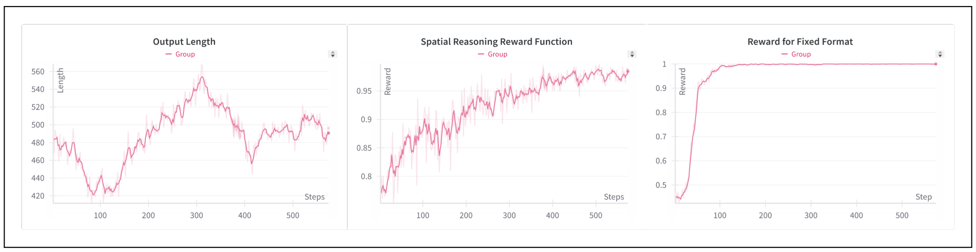

Figure 2: Training Run Graphs: Output Length(First), Spatial Reasoning Reward(Middle), Format Reward(Right)

A.2 Prompts used in grounding framework

System Message:

```
You are a coordinate selection system. Your task is to:
1. Parse the given bounding box coordinates
2. For left/right selection:
   - Compare the starting x-coordinates (first number in each box)
   - Left means smaller x-coordinate
   - Right means larger x-coordinate
3. For top/bottom selection:
   - Compare the starting y-coordinates (second number in each box)
   - Top means smaller y-coordinate
   - Bottom means larger y-coordinate
4. For closest selection:
    - Compare the distance from the center of the box to target
    - Closest means the boxes with the smallest distance between them
5. For size selection:
    - Compare the area of the boxes
    - Larger means the box with the largest area
    - Smaller means the box with the smallest area
6. For Front/back or towards/backwards selection:
    - Compare the depth values of the boxes
    - Front means the box with the larger depth value
    - Back means the box with the smaller depth value

7. Output ONLY the coordinates of the box that matches the location
    requirement
8. Output format must be exactly [x1, y1, x2, y2] with no additional text
```

```
NOTE: You should also consider the size of the box when selecting the
    coordinates, for example if the target is a desktop you can
    intelligently say that the box can't be very small, or if the object
    is large it's bounding box will be large so also consider size of
    bounding box when selecting the coordinates and also relation between
     the target and reference
Remember: In image coordinates:
- X increases from left to right
- Y increases from top to bottom
- Each box format is [x1, y1, x2, y2] where (x1,y1) is top-left and (x2,
    y2) is bottom-right
```

Prompt

```
Given these bounding boxes and depth values:{Object, Coordinates and
    other attributes}.
Select the coordinates for the target object based on this prompt: {
    prompt}. Only return coordinates of the TARGET like [x1, y1, x2, y2]
```

A.3 EXAMPLES

Example 1

Figure 3: Baseball player on the right

Caption: Baseball player on the right
Ground Truth: [194.29, 43.84, 357.73, 630.08]

RL Trained Qwen2.5-3B:

```
First, I'll parse the bounding boxes and depth values:
- Player 1: [33.47, 50.88, 295.27, 622.40]
- Player 2: [194.29, 43.84, 357.73, 630.08]

Next, I'll analyze the coordinates:
- For left/right selection:
  - Player 1's starting x-coordinate is 33.47
  - Player 2's starting x-coordinate is 194.29
  - Player 2 has a larger x-coordinate, so it's on the right.

- For top/bottom selection:
  - Player 1's starting y-coordinate is 50.88
  - Player 2's starting y-coordinate is 43.84
  - Player 2 has a smaller y-coordinate, so it's on top.

- For closest selection:
```

```
  - The distance from the center of Player 1 to the target is calculated
      as follows:
    - Center of Player 1: (33.47 + 295.27 / 2, 50.88 + 622.40 / 2) =
        (164.37, 336.64)
    - Center of Player 2: (194.29 + 357.73 / 2, 43.84 + 630.08 / 2) =
        (275.96, 337.46)
    - Distance between centers: sqrt((275.96 - 164.37)**2 + (337.46 -
        336.64)**2) approx 111.59
  - The distance from the center of Player 2 to the target is calculated
      as follows:
    - Center of Player 2: (194.29 + 357.73 / 2, 43.84 + 630.08 / 2) =
        (275.96, 337.46)
    - Center of Player 1: (164.37, 336.64)
    - Distance between centers: sqrt((275.96 - 164.37)^2 + (337.46 -
        336.64)^2) approx 111.59
  - Both players are equally close to the target.

- For size selection:
  - Area of Player 1: (295.27 - 33.47) * (622.40 - 50.88) = 261.8 *
      571.52 approx 148,000
  - Area of Player 2: (357.73 - 194.29) * (630.08 - 43.84) = 163.44 *
      586.24 approx 96,000
  - Player 2 has a smaller area, so it's the smaller box.

Based on the analysis, the target object is on the right, on top, closest
     to the target, and the smaller box. Therefore, the coordinates for
    the target object are Player 2.
</think>
<answer> [194.29, 43.84, 357.73, 630.08] </answer>
```

Example 2
Caption: Dog on right
Ground Truth: [404, 160, 639, 316]

RL Trained Qwen2.5-3B:

Figure 4: Dog on right

```
First, I'll parse the bounding boxes:
1. Dog 1: [217, 155, 639, 352]
2. Dog 2: [404, 160, 639, 316]


Next, I'll analyze the x-coordinates for the 'dog on right' caption:
- The first box has an x1 of 217
- The second box has an x1 of 404

The caption 'dog on right' indicates the dog should be on the right side
    of the image. The x-coordinates of the boxes are increasing from left
     to right. The boxes with the smallest x-coordinates are the ones on
    the left, and the boxes with the largest x-coordinates are on the
    right.
```

```
Now, I'll check the y-coordinates to ensure the dog is not too high up:
- The y-coordinates for all boxes are relatively close to each other,
    indicating the dog is not too high up.

Therefore, the second box [404, 160, 639, 316] is the closest to the
    target and is on the right side.
</think>
<answer> [404, 160, 639, 316] </answer>
```