A. Appendix

We provide the following appendices for further analysis:

- Details of the baseline methods. (Appendix A.1)
- Implementation details. (Appendix A.2)
- Details of our color categorization pipeline. (Appendix A.3)
- Results with respect to different object scales. (Appendix A.4)
- Results with respect to different object numbers. (Appendix A.5)
- Ablation study on query length. (Appendix A.6)
- Performance validation of the RDAnnotator framework. (Appendix A.7)
- Results on standard RefCOCO/+/g datasets. (Appendix A.8)
- Additional examples from the RefDrone dataset. (Appendix A.9)
- Prompts and examples used in RDAnnotator. (Appendix A.10)

A.1. Details of baseline methods

The details for each baseline method:

- **MDETR**: ResNet-101 with BERT-Base, pretrained on Flickr30k, RefCOCO/+/g, VG.
- **GLIP**: Swin-Tiny with BERT-Base, pretrained on Objects365.
- **GDINO-T**: Swin-Tiny with BERT-Base, pretrained on Objects365, GoldG, GRIT, V3Det.
- **GDINO-B**: Swin-Base with BERT-Base, pretrained on Objects365, GoldG, V3Det.

A.2. Implementation details

NGDINO implementation. We train NGDINO using a two-stage procedure to ensure stability. Stage 1: Pre-training. We initialize the model with weights from a pre-trained GDINO [36], freezing all components except for the number prediction head for 5 epochs. This new head is then pre-trained on the RefDrone dataset. Stage 2: End-to-end Fine-tuning. After the head is pre-trained, we unfreeze the entire model and fine-tune it on our target dataset. This staged approach prevents the randomly initialized prediction head from destabilizing the well-trained detector backbone during the initial phases of training.

Zero-shot evaluation protocol. For all baseline models, we adhere to established evaluation practices to ensure fair comparisons. **Specialized REC Methods**: For GLIP [27] and GDINO [36], we use the official model checkpoints and implementations provided within the MMDetection [7] framework. **LVLMs:** For a standardized and reproducible evaluation of LVLMs, we integrate our dataset into the VLMEvalKit framework [12]. Within this framework, we benchmark each model using its official, recommended prompt structure to ensure optimal performance.

Fine-tuning evaluation details. All fine-tuning experiments are conducted within the MMDetection [7] framework on 8 NVIDIA A100 GPUs. To ensure a fair comparison, we apply a consistent protocol across all models. We follow the original learning strategies and hyperparameter settings for each model with one critical modification: we disable random crop data augmentation. This is because random cropping can remove crucial spatial context or the target objects themselves in our position-sensitive referring expressions, introducing label noise and degrading performance.

A.3. Details of color categorization.

Color is a foundational attribute in the RefDrone dataset, present in 69% of all referring expressions. However, accurately identifying color is non-trivial due to challenges like illumination variance, occlusions, and semantic ambiguity (e.g., distinguishing "red" from "pink" or "orange"). To address this, we designed a hybrid color extraction pipeline that combines the efficiency of a specialized classifier with the reasoning capabilities of an LVLM. The pipeline consists of two stages:

(1) Classifier-based Proposal: A WideResNet-101 classifier [62] generates an initial color prediction. To create a high-quality training set for this classifier, we first generate labels programmatically using the HSV color space and then perform manual validation to correct noise and refine ambiguous cases.

(2) LVLM Verification: An LVLM verifier then assesses the classifier's output. Using structured prompts, it reasons about the visual evidence to confirm the prediction or correct it, effectively resolving ambiguities caused by lighting or partial visibility.

The reliability of this hybrid approach enabled us to expand our vocabulary from an initial set of six primary colors (e.g., red, blue) to a more nuanced palette of twelve, including orange, pink, grey, and purple. The final distribution of these color terms is visualized in Figure 8.

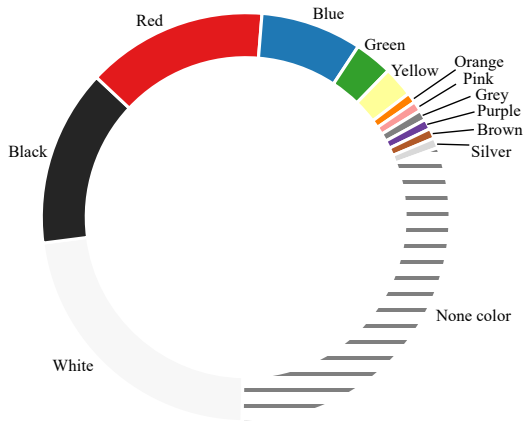


Figure 8. Distribution of color terms in RefDrone expressions.

A.4. The results of different object scales.

To provide a granular analysis of model robustness to scale variation, a critical challenge in the RefDrone benchmark, we evaluated a representative set of high-performing methods ($ACC_{inst.} > 10\%$) on objects categorized as small, medium, and large. The results, presented in Table 7, reveal a stark performance gap between LVLMs and Specialized REC methods, particularly on small objects. Notably, even LVLMs like Qwen3-VL-235B struggle, achieving only 26.54% Acc_s. In contrast, among the fine-tuned specialized REC methods, our NGDINO-B achieves 44.08% Acc_s, 62.59% Acc_m, and 68.20% Acc_l.

Table 7. Performance comparison on small (Acc_s), medium (Acc_m), and large (Acc_l) objects. The models in the upper section are LVLMs evaluated in a zero-shot setting. The models in the lower section are specialized REC methods fine-tuned on the RefDrone training set.

Methods	Params	Acc_s	Acc_m	Acc_l	$Acc_{inst.}$
Rex-Omni [21]	3B	25.55	43.03	50.31	37.10
Qwen2.5-VL [4]	3B	13.52	28.59	32.44	25.06
Qwen2.5-VL [4]	7B	12.19	33.36	35.80	27.20
Qwen3-VL [49]	4B	18.53	42.66	54.23	32.31
Qwen3-VL [49]	8B	20.05	43.64	50.32	35.13
Qwen3-VL [49]	30B	24.21	50.48	54.68	39.98
Qwen3-VL [49]	235B	**26.54**	**51.60**	55.79	**41.93**
MiMo-VL_{RL} [48]	7B	2.41	13.69	15.65	11.22
GLM 4.1V [50]	9B	1.25	16.94	29.84	14.05
GLM 4.5V [50]	106B	7.70	27.83	29.17	21.97
$DeepSeek\text{-}VL2_{Small}$	16B	5.83	27.64	30.36	21.34
DeepSeek-VL2 [57]	27B	2.66	18.58	18.57	14.88
DINO-XSeek [44]	–	24.54	46.52	62.51	37.46
Seed1.5-VL [16]	–	11.11	38.39	50.83	27.84
Qwen3-VL-Plus [49]	–	25.34	49.96	**56.18**	41.32
GLIP-T [27]	0.15B	21.08	48.82	54.35	40.39
GDINO-T [36]	0.17B	38.56	56.69	66.13	51.55
GDINO-B [36]	0.23B	40.01	59.96	67.94	53.95
NGDINO-T (Ours)	0.18B	42.48	60.42	67.90	55.52
NGDINO-B (Ours)	0.24B	**44.08**	**62.59**	**68.20**	**57.22**

A.5. The results of different object numbers.

To analyze model robustness against varying target quantities, we evaluate top-performing methods on scenes containing 0 to 4+ targets (Table 8). The results shows that on empty scenes (Acc_0), most LVLMs suffer from severe hallucination—models like GLM 4.5V and DINO-XSeek achieve below 6% accuracy, failing to reject invalid queries. Besides, LVLM performance degrades sharply as scene density increases (e.g., GLM 4.5V drops from 48.20% at Acc_1 to 16.00% at Acc_{4+}). In contrast, specialized models demonstrate remarkable stability. Our NGDINO-B sets the state-of-the-art for hallucination resistance (65.19% Acc_0) and maintains robust accuracy in highly dense scenes (54.73% Acc_{4+}), consistently outperforming the GDINO-B baseline across all cardinality levels.

Table 8. Results across varying target cardinalities (0 to 4+). Acc_0 highlights hallucination resistance on empty scenes.

Methods	Params	Acc_0	Acc_1	Acc_2	Acc_3	Acc_{4+}
Rex-Omni [21]	3B	3.33	42.42	38.17	36.21	37.69
Qwen3-VL [49]	235B	38.99	55.86	43.58	47.86	39.12
MiMo-VL_{RL} [48]	7B	17.82	32.50	16.19	14.30	5.65
GLM 4.5V [50]	106B	5.50	48.20	28.79	27.77	16.00
DeepSeek-VL2 [57]	27B	25.68	29.98	16.99	16.30	11.17
DINO-XSeek [44]	–	1.57	50.76	37.07	36.16	37.44
Seed1.5-VL [16]	–	0.82	31.39	31.87	32.07	28.80
GDINO-B [36]	0.23B	63.33	**63.66**	58.33	56.60	50.97
NGDINO-B (Ours)	0.24B	**65.19**	63.01	**60.43**	**59.03**	**54.73**

A.6. Ablation study on query length.

Table 9 analyzes the impact of varying the query length. A minimal query length of 1 lacks the capacity to capture complex numerical information. Conversely, extending the query length to 100 increases parameter count and computational overhead, potentially leading to optimization challenges. Through these experiments, we determine that a query length of 10 provides an optimal trade-off.

Table 9. Ablation study on the impact of varying selected number query length. Params indicates additional parameters introduced.

Length	$F1_{inst.}$	$Acc_{inst.}$	$F1_{img.}$	$Acc_{img.}$	Params
1	70.20	54.44	55.82	40.56	1.58M
10	**71.11**	**55.52**	**56.51**	**41.20**	1.65M
100	70.44	54.72	55.95	40.73	2.34M

A.7. Performance of the RDAnnotator framework

To validate the effectiveness of our proposed annotation framework RDAnnotator, we evaluate RDAnnotator when repurposed as a complete, two-stage method for REC. To ensure a fair comparison, all methods operate on an identical set of initial object proposals generated by a first-stage Faster-RCNN detector [43] (18.0 mAP). The core of the evaluation lies in the second stage, where each method uses the referring expression to rank these proposals and identify the target. We benchmark RDAnnotator against two strong alternative second-stage approaches: (1) **GPT-4o**, which represents a powerful, single-step LVLM reasoning approach, and (2) **ReCLIP** [47], a representative CLIP-based ranker that relies on embedding similarity. As shown in Table 10, RDAnnotator substantially outperforms both, validating the efficacy of its structured, multi-step reasoning process. Results underscore RDAnnotator's suitability for generating high-fidelity REC annotations.

Table 10. Experimental results of two-stage instance ranking methods on the RefDrone benchmark.

Methods	$F1_{inst.}$	$Acc_{inst.}$	$F1_{img.}$	$Acc_{img.}$
ReCLIP [47]	24.62	14.04	11.58	6.15
GPT4-o	52.38	35.65	35.50	22.38
RDAnnotator	**58.14**	**41.13**	**37.07**	**23.54**

A.8. Results on RefCOCO/+/g datasets.

Since the RefCOCO, RefCOCO+, and RefCOCOg datasets contain only one instance per expression, the proposed NGDINO leverages the number branch primarily to address multi-instance and no-instance scenarios. As a result, the performance of NGDINO is relatively similar to that of GDINO on these datasets.

Table 11. Results on RefCOCO/+/g datasets.

	RefCOCO		RefCOCO+		RefCOCOg	
	TestA	TestB	TestA	TestB	Val	Test
MDETR	90.4	82.7	85.5	73.0	83.4	83.3
GDINO-T	91.4	**86.6**	87.5	74.0	**85.5**	85.8
NGDINO-T	**91.5**	86.5	**87.8**	**74.7**	85.3	**85.8**

A.9. Dataset examples

To provide a comprehensive understanding of our RefDrone dataset, we present representative examples in Figure 9. These samples demonstrate the three key challenges in our dataset, highlighting its real-world applicability.

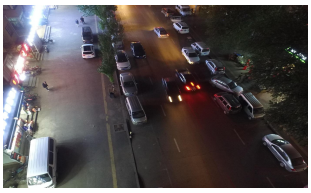
A.10. Prompts and examples for RDAnnotator

In this section, we provide the prompts and examples employed in RDAnnotator. Table 12 presents the prompt construction process for expression generation (Step 3), which includes the system prompt and few-shot in-context learning examples. One in-context learning example is illustrated in Table 13. The system prompts used for each step are detailed in Table 14. Additionally, the system prompts for the feedback mechanism are presented in Table 15.


The red cars moving through the intersection.


The yellow dump truck is parked near one of the buildings.


The blue-clad pedestrians are fewer in number and spread out across the area.


The red cars parked along the sides of the road.


The white vehicles waiting at the intersection.


The people on the basketball court.

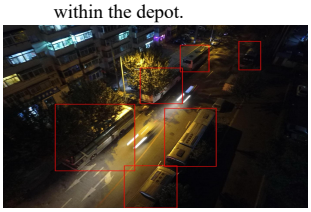

The white cars park along the road adjacent to the parking lot.


The buses parked in rows within the depot.


The pedestrians walk along the pathway on the left side of the image.

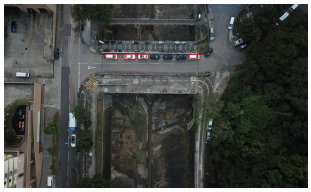

The red taxi travels on the elevated roadway.


The white cars scattered around the area.


The buses on the road.


The white vans are parked or moving around the intersection.

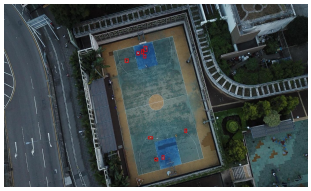

The pedestrians on the stairs on the left side of the picture.

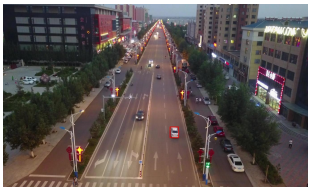

The white cars park along the road in a line.


The red cars navigate the intersection.

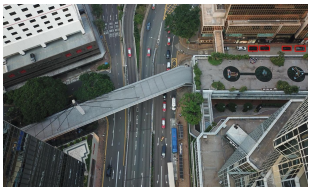

The black cars near each other.


The pedestrians between the bus stop and white bus.


The people playing on the basketball court.


The white cars travel in both directions along the main road.


The red cars in the parking lot.


The black cars parked near the gold building.


Two pedestrians are situated on the right side of the sports field.


The bicycles in the image.

Figure 9. Dataset examples from RefDrone.

Table 12. Illustration of RDAnnotator's prompt construction for expression generation (Step 3). Few-shot in-context-learning examples are from `fewshot_samples`. A representative example is shown in Table 13

```
messages = [ {"role":"system", "content": f"""As an AI visual assistant, your role involves analyzing a single image. You are supplied with three sentences that caption the image, along with additional data about specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y. When coordinate x tends to 0, the object nears the left side of the image, shifting towards the right as coordinate x approaches 1. When coordinate y tends to 0, the object nears the top of the image, shifting towards the bottom as coordinate y approaches 1.

Your task is to classify the provided objects based on various characteristics, while also substantiating your classification. This classification should be thoroughly justified, with criteria including but not limited to relationships or relative locations of objects.

To refer to a specific object, use the provided coordinates directly. Base your classification justifications on direct observations from the image, avoiding any hypothesizing or assumptions."""}
]
for sample in fewshot_samples messages.append({"role":"user", "content":sample['context']})

messages.append({"role":"assistant", "content":sample['response']} )
messages.append({"role":"user", "content":'\n'.join(query)})
```

Table 13. One example to illustrate the input and output in RDAnnotator Step 3 (expression generation). The top block shows the input contexts such as image, object, caption and color attributed. The bottom block shows the generated annotations.

Context type 1: Captions
The image shows an aerial view of a busy urban street with multiple lanes of traffic, flanked by parking areas and buildings. A median strip with greenery runs down the center, dividing the opposite flows of vehicles. On either side of the road, there are commercial and residential structures. Parking lanes on both sides of the main road are filled with parked vehicles.
The image shows a bird's-eye view of an urban street scene with traffic congestion...⟨omitted⟩

Context type 2: Objects
pedestrian, white: [0.173, 0.849]
car, black: [0.214, 0.975]
car, white: [0.215, 0.904]
car, white: [0.313, 0.444]
car, red: [0.315, 0.256]
van, white: ...⟨omitted⟩

Context type 3: Image

Response :
The blue cars park on the right parking lot.
Coordinates: [0.628, 0.478], [0.646, 0.692], [0.649, 0.727]
The white cars park on both sides of the main road in a line.
Coordinates: [0.431, 0.051], [0.431, 0.15], [0.431, 0.101], [0.434, 0.256], [0.439, 0.523], [0.443, 0.813], [0.445, 0.902], [0.446, 0.982], [0.235, 0.43], [0.232, 0.486], [0.223, 0.674], [0.215, 0.904]

Table 14. System prompts employed at different stages of RDAnnotator, replacing the default system messages shown in Table 12

System prompt: Step 1 scene understanding
You are an AI visual assistant that specializes in providing clear and accurate descriptions of images without any ambiguity or uncertainty. Your descriptions should focus solely on the content of the image itself and avoid mentioning any location-specific details such as regions or countries where the image might have been captured.

System prompt: Step 2 color categorization
As an AI visual assistant, your role involves analyzing a single image.
You are supplied with the specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y.
Your task is to assess whether the given colors of specific objects match their appearance in the image. Respond with "Yes" when the colors are appropriate. In cases where the colors are deemed inappropriate, respond with a concise "No."

System prompt: Step 3 expression generation
As an AI visual assistant, your role involves analyzing a single image. You are supplied with three sentences that caption the image, along with additional data about specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y. When coordinate x tends to 0, the object nears the left side of the image, shifting towards the right as coordinate x approaches 1. When coordinate y tends to 0, the object nears the top of the image, shifting towards the bottom as coordinate y approaches 1.
Your task is to classify the provided objects based on various characteristics, while also substantiating your classification. This classification should be thoroughly justified, with criteria including but not limited to relationships or relative locations of objects. To refer to a specific object, use the provided coordinates directly. Base your classification justifications on direct observations from the image, avoiding any hypothesizing or assumptions.

System prompt: Step 4 quality evaluation
As an AI visual assistant, your role involves analyzing a single image. You are supplied with three sentences that describe the image, along with additional data about specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y. When coordinate x tends to 0, the object nears the left side of the image, shifting towards the right as coordinate x approaches 1. When coordinate y tends to 0, the object nears the top of the image, shifting towards the bottom as coordinate y approaches 1. Besides, you are supplied with the description of the objects and their corresponding attributes.
Your task is to confirm whether the description exclusively relates to the described objects without including any others in the visual. Respond "yes" if it matches, or "no" with an explanation if it does not.

Table 15. RDAnnotator system prompts for feedback mechanism. Differences from Table 14 are **highlighted**

System prompt: Step 2 color categorization with feedback mechanism
As an AI visual assistant, your role involves analyzing a single image.
You are supplied with the specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y.
Your task is to assess whether the given colors of specific objects match their appearance in the image. Respond with "Yes" when the colors are appropriate. In cases where the colors are deemed inappropriate, respond with a concise "No."

System prompt: Step 3 expression generation with feedback mechanism
As an AI visual assistant, your role involves analyzing a single image. You are supplied with three sentences that caption the image, along with additional data about specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y. When coordinate x tends to 0, the object nears the left side of the image, shifting towards the right as coordinate x approaches 1. When coordinate y tends to 0, the object nears the top of the image, shifting towards the bottom as coordinate y approaches 1. **You are also provided with descriptions and the objects that initially failed to match, along with the reasons for the discrepancies.**
Your task is to revise both the description and the corresponding objects to correct these mismatches based on the provided reasons. Ensure that the revised description accurately matches the corresponding objects depicted in the visual content.

System prompt: Step 4 quality evaluation with feedback mechanism
As an AI visual assistant, your role involves analyzing a single image. You are supplied with three sentences that describe the image, along with additional data about specific attributes of objects within the image. This can include information about categories, colors, and precise coordinates. Such coordinates, represented as floating-point numbers that range from 0 to 1, are shared as center points, denoted as (x, y), identifying the center x and y. When coordinate x tends to 0, the object nears the left side of the image, shifting towards the right as coordinate x approaches 1. When coordinate y tends to 0, the object nears the top of the image, shifting towards the bottom as coordinate y approaches 1. **Besides, you are supplied with the description and the objects that initially failed to match.**
Your task is to provide detailed reasoning for unsuccessful object matches.