A More Implementation Details

A.1 Hyperparameters

Table 8 presents the hyperparameters used in our main experiments.

Item	Value
optimizer	AdamW
lr	1e-4
lr of image backbone	1e-5
lr of text backbone	1e-5
weight decay	0.0001
clip max norm	0.1
number of encoder layers	6
number of decoder layers	6
dim feedforward	2048
hidden dim	256
dropout	0.0
nheads	8
number of queries	900
set cost class	1.0
set cost bbox	5.0
set cost giou	2.0
ce loss coef	2.0
bbox loss coef	5.0
giou loss coef	2.0

Table 8: Hyper-parameters used in our pre-trained models.

A.2 Pseudo Code Language-Guided Query Selection

We present the pseudo-code of the Language-Guided Query Selection module in Algorithm 1.

The variables `image_feat` and `text_feat` are used for image and text features, respectively. `num_query` is the number of queries in the decoder, which is set to 900 in our implementation. We use `bs` and `ndim` for batch size and feature dimension in the pseudo-code. `num_img_tokens` and `num_text_tokens` are used for the number of image and text tokens, respectively.

B Data Usage

We use three types of data in our model pre-train.

Algorithm 1: Pseudocode of Language-guided Query Selection in PyTorch-like style.

```
"""
Input:
image_feat: (bs, num_img_tokens, ndim)
text_feat: (bs, num_text_tokens, ndim)
num_query: int

Output:
topk_idx: (bs, num_query)
"""

logits = torch.einsum("bic,btc->bit", image_feat, text_feat) # bs, num_img_tokens,
     num_text_tokens
logits_per_img_feat = logits.max(-1)[0]# bs, num_img_tokens
topk_idx = torch.topk(logits_per_img_feature, num_query, dim=1)[1] # bs, num_query
```

1. **Detection data.** Following GLIP [24], we reformulate the object detection task to a phrase grounding task by concatenating the category names into text prompts. We use COCO [28], O365 [42], and OpenImage(OI) [18] for our model pretrain. To simulate different text inputs, we randomly sampled category names from all categories in a dataset on the fly during training.
2. **Grounding data.** We use the GoldG and RefC data as grounding data. Both GoldG and RefC are preprocessed by MDETR [17]. These data can be fed into Grounding DINO directly. GoldG contains images in Flickr30k entities [36, 37] and Visual Genome [19]. RefC contains images in RefCOCO, RefCOCO+, and RefCOCOg.
3. **Caption data.** To enhance the model performance on novel categories, we feed the semantic-rich caption data to our model. Following GLIP, we use the pseudo-labeled caption data for model training. In our experiments, we use the same data with GLIP under comparable settings. More specifically, we use GLIP-T annotated caption data for Grounding DINO T, while GLIP-L annotated caption data for Grounding DINO L.

There are two versions of the O365 dataset, which we termed O365v1 and O365v2, respectively. O365v1 is a subset of O365v2. O365v1 contains about 600K images, while O365v2 contains about 1.7M images. Following previous works [24, 51], we pre-train the Grounding DINO T on O365v1 for a fair comparison. The Grounding DINO L is pre-trained on O365v2 for a better result.

C More Experiment Results

C.1 Transfer from DINO to Grounding DINO

Recent work has presented many large-scale image models for detection with DINO architecture[6]. It is computationally expensive to train a Grounding DINO

[6] See model instances at `https://github.com/IDEA-Research/detrex`

model from scratch. However, the cost can be significantly reduced if we leverage pre-trained DINO weights. Hence, we conduct some experiments to transfer pre-trained DINO to Grounding DINO models. We freeze the modules co-existing in DINO and Grounding DINO and fine-tune the other parameters only. (We compare DINO and Grounding DINO in Sec. C.1.) The results are available in Table 9.

Model	Pre-Train Data		COCO minival	LVIS minival	ODinW
	DINO Pre-Train	Grounded Fine-Tune	Zero-Shot	Zero-Shot	Zero-Shot
Grounding DINO T	-	O365	46.7	16.2	14.5
(from scratch)	-	O365,GoldG	48.1	25.6	20.0
Grounding DINO T	O365	O365	46.5	17.9	13.6
(from pre-trained DINO)	O365	O365,GoldG	46.4	26.1	18.5

Table 9: Transfer pre-trained DINO to Grounding DINO. We freeze shared modules between DINO and Grounding DINO during grounded fine-tuning. All models are trained with a Swin Transformer Tiny backbone.

It shows that we can achieve similar performances with Grounding DINO-Training only text and fusion blocks using a pre-trained DINO. Interestingly, the DINO-pre-trained Grounding DINO outperforms the standard Grounding DINO on LVIS under the same setting. The results show that there might be much room for model training improvement, which will be our future work to explore.

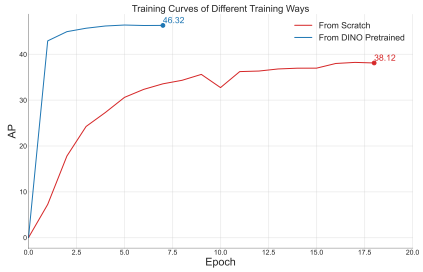


Table 10: Comparison between two Grounding DINO variants: Training from scratch and transfer from DINO-pretrained models. The models are trained on O365 and evaluated on COCO.

With a pre-trained DINO initialization, the model converges faster than Grounding DINO from scratch, as shown in Fig. 10. Notably, we use the results without exponential moving average (EMA) for the curves in Fig. 10, which results in a different final performance that in Table 9. As the model trained from scratch need more training time, we only show results of early epochs.

Comparison between DINO and Grounding DINO To illustrate the difference between DINO and Grounding DINO, we compare DINO and

Grounding DINO in Fig. 5. We mark the DINO blocks in gray, while the newly proposed modules are shaded in blue.

C.2 Detailed Results on COCO Detection Benchmarks

COCO Detection Results under the 1× Setting

Model	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster-RCNN(5scale) [38]	12	37.9	58.8	41.1	22.4	41.1	49.1
DETR(DC5) [2]	12	15.5	29.4	14.5	4.3	15.1	26.7
Deformable DETR(4scale) [63]	12	41.1	–	–	–	–	
DAB-DETR(DC5)† [30]	12	38.0	60.3	39.8	19.2	40.9	55.4
Dynamic DETR(5scale) [**?**]	12	42.9	61.0	46.3	24.6	44.9	54.4
Dynamic Head(5scale) [5]	12	43.0	60.7	46.8	24.7	46.4	53.9
HTC(5scale) [3]	12	42.3	–	–	–	–	–
DN-Deformable-DETR(4scale) [23]	12	43.4	61.9	47.2	24.8	46.8	59.4
DINO-4scale [56]	12	49.0	66.6	53.5	32.0	52.3	63.0
Grounding DINO (4scale)	12	48.1	65.8	52.3	30.4	51.3	62.3

Table 11: Results for Grounding DINO and other detection models with the ResNet50 backbone on COCO `val2017` trained with 12 epochs (the so called 1× setting).

We present the performance of Grounding DINO on standard COCO detection benchmark in Table 11. All models are trained with a ResNet-50 [15] backbone for 12 epochs. Grounding DINO achieves 48.1 AP under the research setting, which shows that Grounding DINO is a strong closed-set detector. However, it is inferior compared with the original DINO. We suspect that the new components may make the model harder to optimize than DINO.

C.3 Detailed Results on ODinW Benchmarks

We present detailed results of Grounding DINO on ODinW35 [22] in Table 12, Table 13, and Table 14.

C.4 Model Efficiency

We compare the model size and efficiency between Grounding DINO T and GLIP-T in Table 20. The results show that our model has a smaller parameter size and better efficiency than GLIP.

C.5 Ablations for More Decoder Queries

To verify the model performance with more decoder queries, we conducted additional experiments with 1200 and 1500 queries on the COCO and LVIS datasets, detailed in Table 15 below.

Dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
AerialMaritimeDrone_large	9.48	15.61	8.35	8.72	10.28	2.91
AerialMaritimeDrone_tiled	17.56	26.35	13.89	0	1.61	28.7
AmericanSignLanguageLetters	1.45	2.21	1.39	-1	-1	1.81
Aquarium	18.83	34.32	18.19	10.65	20.64	21.52
BCCD_BCCD	6.17	11.31	6.04	1.27	9.09	6.89
ChessPiece	6.99	11.13	9.03	-1	-1	8.11
CottontailRabbits	71.93	85.05	85.05	-1	70	73.58
DroneControl_Drone_Control	6.15	10.95	6.23	2.08	6.91	6.16
EgoHands_generic	48.07	75.06	56.52	1.48	11.42	51.84
EgoHands_specific	0.66	1.25	0.64	0	0.02	0.92
HardHatWorkers	2.39	9.17	1.07	2.13	4.32	4.6
MaskWearing	0.58	1.43	0.56	0.12	0.51	4.66
MountainDewCommercial	18.22	29.73	21.33	0	23.23	49.8
NorthAmericaMushrooms	65.48	71.26	66.18	-1	-1	65.49
OxfordPets_by-breed	0.27	0.6	0.21	-1	1.38	0.33
OxfordPets_by-species	1.66	5.02	1	-1	0.65	1.89
PKLot_640	0.08	0.26	0.02	0.14	0.79	0.11
Packages	56.34	68.65	68.65	-1	-1	56.34
PascalVOC	47.21	57.59	51.28	16.53	39.51	58.5
Raccoon_Raccoon	44.82	76.44	46.16	-1	17.08	48.56
ShellfishOpenImages	23.08	32.21	26.94	-1	18.82	23.28
ThermalCheetah	12.9	19.65	14.72	0	8.35	50.15
UnoCards	0.87	1.52	0.96	2.91	2.18	-1
VehiclesOpenImages	59.24	71.88	64.69	7.42	32.38	72.21
WildfireSmoke	25.6	43.96	25.34	5.03	18.85	42.59
boggleBoards	0.81	2.92	0.12	2.96	1.13	-1
brackishUnderwater	1.3	1.88	1.4	0.99	1.75	11.39
dice_mediumColor	0.16	0.72	0.07	0.38	3.3	2.23
openPoetryVision	0.18	0.5	0.06	-1	0.25	0.17
pistols	46.4	66.47	47.98	4.51	22.94	55.03
plantdoc	0.34	0.51	0.35	-1	0.28	0.86
pothole	19.87	28.94	22.23	12.49	15.6	28.78
selfdrivingCa	9.46	19.13	8.19	0.85	6.82	16.51
thermalDogsAndPeople	72.67	86.65	79.98	33.93	30.2	86.71
websiteScreenshots	1.51	2.8	1.42	0.85	2.06	2.59

Table 12: Detailed results on 35 datasets in ODinW of Grounding DINO with Swin-T pre-trained on O365 and GoldG.

Dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
AerialMaritimeDrone_large	10.3	18.17	9.21	8.92	11.2	7.35
AerialMaritimeDrone_tiled	17.5	28.04	18.58	0	3.64	24.16
AmericanSignLanguageLetters	0.78	1.17	0.76	-1	-1	1.02
Aquarium	18.64	35.27	17.29	11.33	17.8	21.34
BCCD_BCCD	11.96	22.77	8.65	0.16	5.02	13.15
ChessPiece	15.62	22.02	20.19	-1	-1	15.72
CottontailRabbits	67.61	78.82	78.82	-1	70	68.09
DroneControl_Drone_Control	4.99	8.76	5	0.65	5.03	8.61
EgoHands_generic	57.64	90.18	66.78	3.74	24.67	61.33
EgoHands_specific	0.69	1.37	0.63	0	0.02	1.03
HardHatWorkers	4.05	13.16	1.96	2.29	7.55	9.81
MaskWearing	0.25	0.81	0.15	0.09	0.13	2.78
MountainDewCommercial	25.46	39.08	28.89	0	32.53	58.38
NorthAmericaMushrooms	68.18	72.89	69.75	-1	-1	68.62
OxfordPets_by-breed	0.21	0.42	0.22	-1	2.91	0.17
OxfordPets_by-species	1.3	3.95	0.71	-1	0.28	1.62
PKLot_640	0.06	0.18	0.02	0.03	0.59	0.15
Packages	60.53	76.24	76.24	-1	-1	60.53
PascalVOC	55.65	66.51	60.47	19.61	44.25	67.21
Raccoon_Raccoon	60.07	84.81	66.5	-1	11.23	65.86
ShellfishOpenImages	29.56	38.08	33.5	-1	6.38	29.95
ThermalCheetah	17.72	25.93	19.61	1.04	20.02	63.69
UnoCards	0.81	1.3	1	2.6	1.01	-1
VehiclesOpenImages	58.49	71.56	63.64	8.22	28.03	71.1
WildfireSmoke	20.04	39.74	22.49	4.13	15.71	30.41
boggleBoards	0.29	1.15	0.04	1.8	0.57	-1
brackishUnderwater	1.47	2.34	1.58	2.32	3.31	9.96
dice_mediumColor	0.33	1.38	0.15	0.03	1.05	12.57
openPoetryVision	0.05	0.19	0	-1	0.09	0.21
pistols	66.99	86.34	72.65	16.25	39.24	75.98
plantdoc	0.36	0.47	0.39	-1	0.24	0.82
pothole	25.21	38.21	26.01	8.94	18.45	39.28
selfdrivingCa	9.95	20.55	8.28	1.36	7.27	15.46
thermalDogsAndPeople	67.89	80.85	78.66	45.05	30.24	85.56
websiteScreenshots	1.3	2.26	1.21	0.95	1.81	2.23

Table 13: Detailed results on 35 datasets in ODinW of Grounding DINO with Swin-T pre-trained on O365, GoldG, and Cap4M.

Dataset	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
AerialMaritimeDrone_large	12.64	18.44	14.75	9.15	19.16	0.98
AerialMaritimeDrone_tiled	20.47	34.81	12.79	0	7.61	26.93
AmericanSignLanguageLetters	3.94	4.84	4	-1	-1	4.48
Aquarium	28.14	45.47	30.97	12.1	24.71	39.42
BCCD_BCCD	23.85	36.92	28.88	0.3	10.8	24.43
ChessPiece	18.44	26.3	23.33	-1	-1	18.62
CottontailRabbits	71.66	88.48	88.48	-1	66	73.04
DroneControl_Drone_Control	7.16	11.56	7.67	2.29	10.6	7.68
EgoHands_generic	52.08	81.57	59.15	1.12	31.78	55.46
EgoHands_specific	1.22	2.28	1.2	0	0.05	1.5
HardHatWorkers	9.14	23.64	5.6	5.09	15.34	13.59
MaskWearing	1.64	4.69	1.18	0.44	1.05	8.67
MountainDewCommercial	33.28	53.59	32.76	0	35.86	80
NorthAmericaMushrooms	72.33	73.18	73.18	-1	-1	72.39
OxfordPets_by-breed	0.58	1.05	0.59	-1	4.46	0.6
OxfordPets_by-species	1.64	4.8	0.87	-1	1.51	1.8
PKLot_640	0.25	0.71	0.05	0.31	1.44	0.4
Packages	63.86	76.24	76.24	-1	-1	63.86
PascalVOC	66.01	76.65	71.8	32.01	55.7	75.37
Raccoon_Raccoon	65.81	90.39	69.93	-1	26	68.97
ShellfishOpenImages	62.47	74.25	70.07	-1	26	63.06
ThermalCheetah	21.33	26.11	24.92	2.39	15.84	75.34
UnoCards	0.52	0.84	0.66	3.02	0.92	-1
VehiclesOpenImages	62.74	75.15	67.23	10.66	47.46	76.36
WildfireSmoke	23.66	45.72	25.06	1.58	22.22	35.27
boggleBoards	0.28	1.04	0.05	5.64	0.7	-1
brackishUnderwater	2.41	3.39	2.79	4.43	3.88	21.22
dice_mediumColor	0.26	1.15	0.03	0	1.09	4.07
openPoetryVision	0.08	0.35	0.01	-1	0.15	0.11
pistols	71.4	90.69	77.21	18.74	39.58	80.78
plantdoc	2.02	2.64	2.37	-1	0.5	2.82
pothole	30.4	44.22	33.84	12.27	18.84	48.57
selfdrivingCa	9.25	17.72	8.39	1.93	7.03	13.02
thermalDogsAndPeople	72.02	86.02	79.47	29.16	68.05	86.75
websiteScreenshots	1.32	2.64	1.16	0.79	1.8	2.46

Table 14: Detailed results on 35 datasets in ODinW of Grounding DINO with Swin-L pre-trained on O365, OI, GoldG, Cap4M, COCO, and RefC.

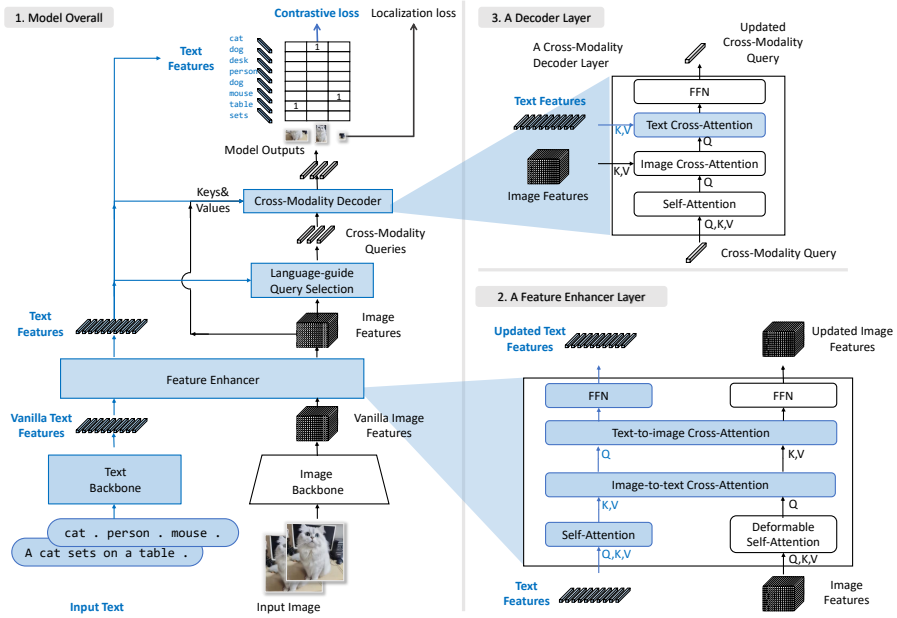


Fig. 5: Comparison between DINO and our Grounding DINO. We mark the modifications in blue. Best view in color.

The results indicate that models with 1200 and 1500 queries slightly outperform the 900-query version on LVIS rare classes, suggesting better coverage of rare classes. However, the improvement is marginal, as the 900-query model already sufficiently covers all objects in both COCO and LVIS. Additionally, introducing more queries exacerbates data imbalance during training, as the model is trained on objects from sampled categories. This imbalance could offset the benefits of additional queries.

Pretrain Data	Query Num	COCO				LVIS			
		AP	AP_S	AP_M	AP_L	AP	AP_r	AP_c	AP_f
O365	900	46.7	32.1	50.0	61.3	15.8	9.4	22.9	28.4
O365	1200	46.7	32.3	49.9	61.1	15.7	9.0	23.1	29.1
O365	1500	46.9	32.7	50.3	61.3	15.8	9.2	23.0	29.0

Table 15: Results for Grounding DINO Tiny with more decoder queries

C.6 Results with Different Language Encoder for REC

To verify the impacts of language encoders with different sizes, we conducted experiments using two variants of BERT: bert-base-uncased (BERT-B) and bert-large-uncased (BERT-L). These models were trained on a combined dataset consisting of RefCOCO, RefCOCO+, and RefCOCOg. We removed the leaked data in the combined dataset for a fair comparison. It's important to note that training on this combined dataset, as opposed to tuning each dataset separately, might result in slightly lower performance. For a fair comparison, we initialized the models with the O365+GoldG+Cap4M checkpoint, except for the BERT parameters. Limited by time and resources, the training duration was capped at 18 epochs.

Interestingly, our results showed that Grounding DINO with BERT-B outperformed or matched the BERT-L variant in most metrics (as shown in Table 16). This suggests that our default use of BERT-B during the pretrain stage may have contributed to its better performance. Moreover, we didn't observe significant improvements in the late stages of training, indicating that both models were nearing their optimal performance.

This outcome suggests that the main limitation in enhancing REC performance lies within the detection branch, rather than the language processing module. A dedicated model for REC data might be helpful for REC tasks.

Model	RefCOCO			RefCOCO+			RefCOCOg	
	val	testA	testB	val	testA	testB	val	test
Grounding DINO T (BERT-B)	87.4	91.6	84.2	78.6	86.5	73.4	81.6	83.3
Grounding DINO T (BERT-L)	87.0	91.4	84.0	78.1	86.4	73.0	81.7	83.3

Table 16: Results for Grounding DINO with different language encoders

C.7 Detic Pseudo-labeled Data for LVIS

To illustrate the potential of our model under similar conditions, we conducted oracle experiments as shown in Table 17. We pseudo-labeled the ImageNet dataset using a pre-trained Detic [61] model and filtered out LVIS-related images for training, creating a new pseudo-labeled dataset named IN22K-LVIS-1M. It contains about 1M pseudo-labeled images for training. Note that our model under the setting may be not a real zero-shot setting, as our pseudo labeler Detic is trained with LVIS data. The results from these experiments suggest that Grounding DINO can achieve promising LVIS performance, even without direct LVIS training data. A similarly distributed dataset can help the model to generalize well.

Model	Pre-train Data	LVIS MiniVal - AP(r/c/f) Zero-shot	Fine-tune
DetCLIPv2-T	OG + CC15M	40.4 (36.0 / 41.7 / 40.0)	50.7 (44.3 / 52.4 / 50.3)
Grounding DINO T	OG+IN22K-LVIS-1M	40.6 (38.5 / 41.1 / 40.4)	54.5 (47.3 / 53.9 / 56.1)

Table 17: Oracle experiments on LVIS. Note that our model under the setting may be not a real zero-shot setting, as our pseudo labeler Detic is trained with LVIS data.

C.8 Comparison between Grounding DINO and GLIP on ODinW

In our comparison of Grounding DINO and GLIP across various datasets in ODinW, as presented in Table 18, we observe that Grounding DINO underperforms on certain uncommon datasets where both models generally show limited effectiveness. For instance, in the PlantDoc dataset, Grounding DINO scores 0.36 compared to GLIP's 1.1. This dataset includes infrequent categories such as "Tomato leaf mosaic virus," which are not well-represented in the training data. These findings highlight the need for improving data quality to enhance overall model performance.

C.9 Training Grounding DINO on RefCOCO/+/g from scratch

To demonstrate our model's capabilities, we trained a Grounding DINO from scratch on the RefCOCO/+/g datasets, as shown in Table 19. Despite time limitations that restricted us to 9 epochs of training, we achieved comparable performance to SOTA REC models. This underscores the potential of our approach to handling REC tasks effectively.

D Visualizations

D.1 Detection Visualizations

We present some visualizations in Fig. 6. Our model presents great generalization on different scenes and text inputs. For example, Grounding DINO accurately locates `man in blue` and `child in red` in the last image.

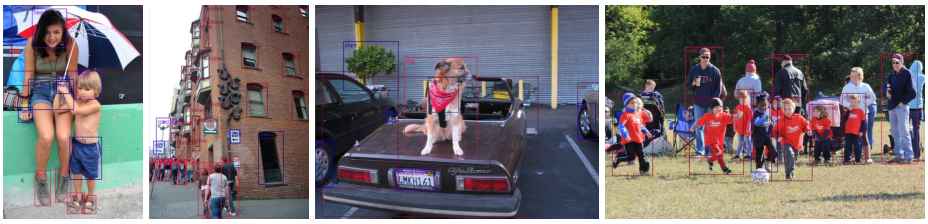


Fig. 6: Visualizations of model outputs.

Metric	GLIP-T	Grounding DINO T
Average Score (↑)	19.6	22.3
Median Score (↑)	5.1	11.9
AerialMaritimeDrone_large (↑)	13.70	10.30
AerialMaritimeDrone_tiled (↑)	12.60	17.50
AmericanSignLanguageLetters_American_Sign_Language_Letters (↑)	2.50	0.78
Aquarium_Aquarium_Combined (↑)	18.30	18.64
BCCD_BCCD (↑)	1.00	11.96
ChessPieces_Chess_Pieces (↑)	10.00	15.62
CottontailRabbits (↑)	69.70	67.61
DroneControl_Drone_Control (↑)	5.10	4.99
EgoHands_generic (↑)	50.00	57.64
EgoHands_specific (↑)	0.80	0.69
HardHatWorkers (↑)	3.00	4.05
MaskWearing (↑)	1.10	0.25
MountainDewCommercial (↑)	21.60	25.46
NorthAmericaMushrooms_North_American_Mushrooms (↑)	75.10	68.18
OxfordPets_by-breed (↑)	0.40	0.21
OxfordPets_by-species (↑)	1.10	1.30
PKLot_640 (↑)	0.00	0.06
Packages_Raw (↑)	72.30	60.53
PascalVOC (↑)	56.10	55.65
Raccoon_Raccoon (↑)	57.80	60.07
ShellfishOpenImages (↑)	25.90	29.56
ThermalCheetah (↑)	2.70	17.72
UnoCards (↑)	0.20	0.81
VehiclesOpenImages (↑)	56.00	58.49
WildfireSmoke (↑)	2.30	20.04
boggleBoards_416x416AutoOrient_export (↑)	0.00	0.29
brackishUnderwater (↑)	3.70	1.47
dice_mediumColor_export (↑)	1.10	0.33
openPoetryVision (↑)	0.00	0.05
pistols_export (↑)	49.80	66.99
plantdoc (↑)	1.10	0.36
pothole (↑)	17.20	25.21
selfdrivingCar_fixedLarge_export (↑)	8.00	9.95
thermalDogsAndPeople (↑)	43.70	67.89
websiteScreenshots (↑)	0.50	1.30

Table 18: Comparison of Grounding DINO and GLIP on ODinW. Both models are trained on O365, GoldG, and Cap4M with Swin-Tiny backbones.

		refcoco (val/testA/testB)	refcoco+ (val/testA/testB)	refcocog (val/test)
TransVG	R101	81.02/82.72/78.35	64.82/70.70/56.94	68.67/67.73
Grounding DINO T (9ep)	SwinT	**81.42**/85.28/76.75	**69.14**/74.79/60.17	**73.96**/74.37

Table 19: Training on RefCOCO from scratch.

D.2 Physical meaning of language-guided query

We visualize the top 900 queries (boxes) in language-guided query selection across various prompts in Figure 7. It shows our model can have dynamic queries during inference. We will incorporate the analyses into the revised paper.

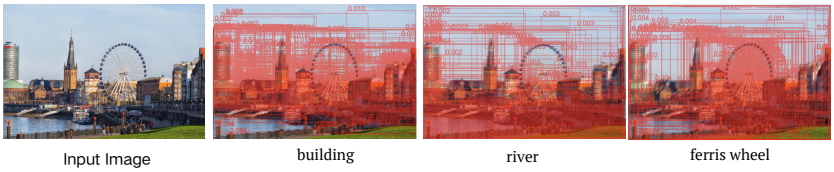


Fig. 7: Top queries in language-guided query selection.

D.3 Comparison of RefCOCO and Grounding Data

The RefCOCO has a different formulation with grounded training, resulting in a big performance gap without the RefCOCO dataset. As shown in Fig. 8, each RefCOCO text prompt corresponds to only *one box*, while our model tends to predict *multiple objects*.

Fig. 8: Our model predictions and ground-truths in RefCOCO.

D.4 Marry Grounding DINO with Stable Diffusion for Object Detection and Inpainting

We present an image editing application in Fig. 1 (b). The results in Fig. 1 (b) are generated by two processes. First, we detect objects with Grounding DINO and generate masks by masking out the detected objects or backgrounds. After that, we feed original images, image masks, and generation prompts to an inpainting model (typical Stable Diffusion [40]) to render new images. We use the released checkpoints in `https://github.com/Stability-AI/stablediffusion` for new image generation. More results are available in Figure 9.

The "detection prompt" is the language input for Grounding DINO, while the "generation prompt" is for the inpainting model.

D.5 Marry Grounding DINO with Stable Diffusion for Object Detection and Grounded Generation

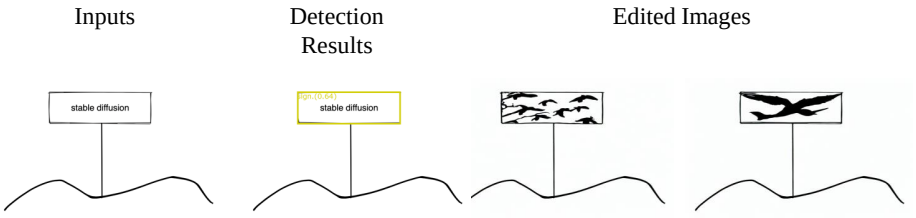
To enable fine-grained image editing, we combine the Grounding DINO with GLIGEN [26]. We use the "phrase prompt" in Figure 10 as the input phrases of each box for GLIGEN.

GLIGEN supports grounding results as inputs and can generate objects on specific positions. We can assign each bounding box an object with GLIGEN, as shown in Figure 10 (c) (d). Moreover, GLIGEN can full fill each bounding box, which results in better visualization, as that in Figure 10 (a) (b). For example, we use the same generative prompt in Figure 9 (b) and Figure 10 (b). The GLIGEN results ensure each bounding box with an object and fulfills the detected regions.

Model	params	GFLOPS	FPS
GLIP-T [24]	232M	488G	6.11
Grounding DINO T (Ours)	172M	464G	8.37

Table 20: Comparison of model size and model efficiency between GLIP and Grounding DINO.

Fig. 9: Combination of Grounding DINO and Stable Diffusion. We first detect objects with Grounding DINO and then perform image inpainting with Stable Diffusion. "Detection Prompt" and "Generation Prompt" are inputs for Grounding DINO and Stable Diffusion, respectively. *The input human face in the row (e) is generated by StyleGAN.



(a) Detection Prompt: `sign`
Generation Prompt: `flying birds.`
Phrase Prompt: `flying birds.`

(b) Detection Prompt: `pandas`
Generation Prompt: `dogs and birthday cakes.`
Phrase Prompt*: `a dog; a cake.`

(c) Detection Prompt: `dog, cat`
Generation Prompt: `a cake and a phone`
Phrase Prompt*: `a cake; a phone.`

(d) Detection Prompt: `a sketch person`
Generation Prompt: `a woman and a man are talking`
Phrase Prompt*: `a woman; a man.`

Fig. 10: Combination of Grounding DINO and GLIGEN. We first detect objects with Grounding DINO and then perform image inpainting with GLIGEN. "Detection Prompt" and "Generation Prompt" are inputs for Grounding DINO and Stable Diffusion, respectively. "Phrase Prompt" are language inputs for each bounding box. The phrase prompts are separated by semicolons. *We assign phrase prompts to bounding boxes randomly.