

Model	Backbone	Style	COCO mAP	Pre-Train Data	Config	Download
GDINO-T	Swin-T	Zero-shot	46.7	O365		
GDINO-T	Swin-T	Zero-shot	48.1	O365,GoldG		
GDINO-T	Swin-T	Zero-shot	48.4	O365,GoldG,Cap4M	config	model
MM-GDINO-T	Swin-T	Zero-shot	48.5(+1.8)	O365	config	
MM-GDINO-T	Swin-T	Zero-shot	50.4(+2.3)	O365,GoldG	config	model log
MM-GDINO-T	Swin-T	Zero-shot	50.5(+2.1)	O365,GoldG,GRIT	config	model log
MM-GDINO-T	Swin-T	Zero-shot	50.6(+2.2)	O365,GoldG,V3Det	config	model log
MM-GDINO-T	Swin-T	Zero-shot	50.4(+2.0)	O365,GoldG,GRIT,V3Det	config	model log
MM-GDINO-B	Swin-B	Zero-shot	52.5	O365,GoldG,V3Det	config	model log
MM-GDINO-B*	Swin-B	-	59.5	O365,ALL	config	model log
MM-GDINO-L	Swin-L	Zero-shot	53.0	O365V2,OpenImageV6,GoldG	config	model log
MM-GDINO-L*	Swin-L	-	60.3	O365V2,OpenImageV6,ALL	config	model log

SwinT

首先根据mm_groundingdino提供的映射脚本编写逆映射脚本，直接运行的话mAP只有2.5:

```

DONE (t=32.78s).
IoU metric: bbox
Average Precision (AP) @ [ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.025
Average Precision (AP) @ [ IoU=0.50 | area= all | maxDets=100 ] = 0.035
Average Precision (AP) @ [ IoU=0.75 | area= all | maxDets=100 ] = 0.027
Average Precision (AP) @ [ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.017
Average Precision (AP) @ [ IoU=0.50:0.95 | area= medium | maxDets=100 ] = 0.029
Average Precision (AP) @ [ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.042
Average Recall (AR) @ [ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.326
Average Recall (AR) @ [ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.610
Average Recall (AR) @ [ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.688
Average Recall (AR) @ [ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.514
Average Recall (AR) @ [ IoU=0.50:0.95 | area= medium | maxDets=100 ] = 0.722
Average Recall (AR) @ [ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.859
Final results: [0.02475664349379888, 0.03461723145598353, 0.027246433266181246, 0.017049363603216947, 0.029263336259377076, 0.04153528796521024, 0.3260130586514552, 0.6098256524597043, 0.6876670673785132, 0.5142777006977029, 0.722113711304385, 0.8585992016392038]

```

找到mm_groundingdino中多余的参数cls_embed.bias，对应到代码上进行修改，发现推理过程有一处需要修改:

GroundingDINO-main\groundingdino\models\GroundingDINO\utils.py中的ContrastiveEmbed类在初始化时需要加上bias，将mm_groundingdino中对应的ContrastiveEmbed类粘贴过来并进行形参的修改，mAP涨到了49.5:

Accumulating evaluation results...

DONE (t=31.32s).

IoU metric: bbox

Average Precision (AP) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.495

Average Precision (AP) @[IoU=0.50 | area= all | maxDets=100] = 0.666

Average Precision (AP) @[IoU=0.75 | area= all | maxDets=100] = 0.546

Average Precision (AP) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.351

Average Precision (AP) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.520

Average Precision (AP) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.639

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 1] = 0.384

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets= 10] = 0.658

Average Recall (AR) @[IoU=0.50:0.95 | area= all | maxDets=100] = 0.723

Average Recall (AR) @[IoU=0.50:0.95 | area= small | maxDets=100] = 0.584

Average Recall (AR) @[IoU=0.50:0.95 | area=medium | maxDets=100] = 0.760

Average Recall (AR) @[IoU=0.50:0.95 | area= large | maxDets=100] = 0.870

Final results: [0.4954234686500252, 0.6664532061608047, 0.5461393674031254, 0.35141639503068633, 0.5203336432, 0.63907595056223008042, 0.8698016906969595]

```
class ContrastiveEmbed(nn.Module):
```

```
    """text visual ContrastiveEmbed layer.
```

```
    Args:
```

```
        max_text_len (int, optional): Maximum length of text.
```

```
        log_scale (Optional[Union[str, float]]): The initial value  
of a
```

```
        learnable parameter to multiply with the similarity  
matrix to normalize the output. Defaults to 0.0.
```

```
        - If set to 'auto', the similarity matrix will be  
normalized by
```

```
        a fixed value sqrt(d_c) where d_c is the  
channel number.
```

```
        - If set to 'none' or None, there is no normalization  
applied.
```

```
        - If set to a float number, the similarity matrix will be  
multiplied
```

```
        by exp(log_scale), where log_scale is  
learnable.
```

```
        bias (bool, optional): whether to add bias to the output.
```

```
        If set to True, a learnable bias that is initialized  
as -4.6
```

```
will be added to the output. Useful when training from  
scratch.
```

```
        Defaults to False.
```

```

def __init__(self,
              max_text_len: int = 256,
              log_scale: Optional[Union[str, float]] = 'auto',
              bias: bool = True):
    super().__init__()
    self.max_text_len = max_text_len
    self.log_scale = log_scale
    if isinstance(log_scale, float):
        self.log_scale = nn.Parameter(
            torch.Tensor([float(log_scale)]),
requires_grad=True)
    elif log_scale not in ['auto', 'none', None]:
        raise ValueError(f'log_scale should be one of '
                        f'"auto", "none", None, but got '
                        f'{log_scale}')

    self.bias = None
    if bias:
        bias_value = -math.log((1 - 0.01) / 0.01)
        self.bias = nn.Parameter(
            torch.Tensor([bias_value]), requires_grad=True)

def forward(self, x, text_dict):
    y = text_dict['encoded_text']
    text_token_mask = text_dict['text_token_mask']

    res = x @ y.transpose(-1, -2)
    if isinstance(self.log_scale, nn.Parameter):
        res = res * self.log_scale.exp()
    elif self.log_scale == 'auto':
        # NOTE: similar to the normalizer in self-attention
        res = res / math.sqrt(x.shape[-1])
    if self.bias is not None:
        res = res + self.bias
    res.masked_fill_(~text_token_mask[:, None, :], float('-inf'))

    new_res = torch.full((*res.shape[:-1], self.max_text_len),
                        float('-inf'),
                        device=res.device)

```

```
new_res[... , :res.shape[-1]] = res
```

```
return new_res
```

然后需要把config文件中的dec_pred_bbox_embed_share改为False，mAP才能升到50.6：

```
DONE (t=22.36s).
IoU metric: bbox
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.586
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.666
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.554
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.362
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.534
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.650
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.391
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.675
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.743
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.600
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.784
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.889
Final results: [0.5066539403530481, 0.6658510693387695, 0.5541682232648256, 0.36189550455982117, 0.5344943177296846, 0.6499104349965644, 0.3909743394170505, 0.6748490536170231, 0.7427787786290326, 0.5995071204688103, 0.7835516790957459, 0.8885025746770313]
```

SwinB

SwinB需要将SwinT的逆映射脚本稍微修改一下，把所有参数的module前缀去掉，然后复制SwinT的config并修改backbone为'swin_B_384_22k'就好了：

```
Accumulating evaluation results...
DONE (t=22.95s).
IoU metric: bbox
Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.595
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.769
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.652
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.435
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.635
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.746
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.423
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.722
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.788
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.652
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.830
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.923
Final results: [0.5948971813294237, 0.7687737521360557, 0.6523923956942317, 0.4352189810289353, 0.6350675026947016, 0.7460046859906078, 0.422866133412595, 0.7220612953886039, 0.7878592758622461, 0.6522888470763134, 0.8302326714643905, 0.9227906796958978]
```

SwinL

SwinL直接用SwinB的逆映射脚本会报很多mismatch的错，分别需要修改逆映射脚本和config文件。groundingdino没有提供SwinL的config文件，就还是复制修改好后的SwinT的config，config需要修改以下三处：

```
backbone="swin_L_384_22k"
```

```
return_interm_indices=[0, 1, 2, 3]
```


num_feature_levels = 5

由于num_feature_levels比原来的4多了1，所以逆映射脚本中的neck.extra_convs.0应该换成neck.convs.4而不是neck.convs.3

```
IoU metric: bbox
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.683
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.776
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.663
Average Precision  (AP) @[ IoU=0.50:0.95 | area=  small | maxDets=100 ] = 0.459
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.644
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.757
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.427
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.726
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.794
Average Recall     (AR) @[ IoU=0.50:0.95 | area=  small | maxDets=100 ] = 0.661
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.834
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.928
Final results: [0.683766643985231, 0.77578659378539, 0.6634418961589152, 0.4583815327828889, 0.6448877675787688, 0.7567862145737874, 0.4265598168958687, 0.7261472869516854, 0.7938993551834277, 0.6687634143282358, 0.8337928493274483, 0.9283497785956769]
```