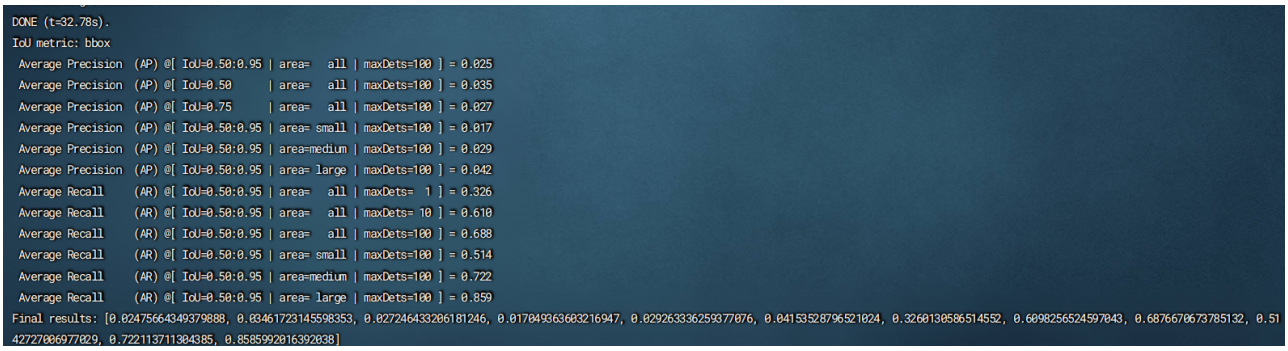
Model	Backbone	Style	COCO mAP	Pre-Train Data	Config	Download
GDINO-T	Swin-T	Zero-shot	46.7	O365		
GDINO-T	Swin-T	Zero-shot	48.1	O365,GoldG		
GDINO-T	Swin-T	Zero-shot	48.4	O365,GoldG,Cap4M	config	model
MM-GDINO-T	Swin-T	Zero-shot	48.5(+1.8)	O365	config	
MM-GDINO-T	Swin-T	Zero-shot	50.4(+2.3)	O365,GoldG	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：



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：

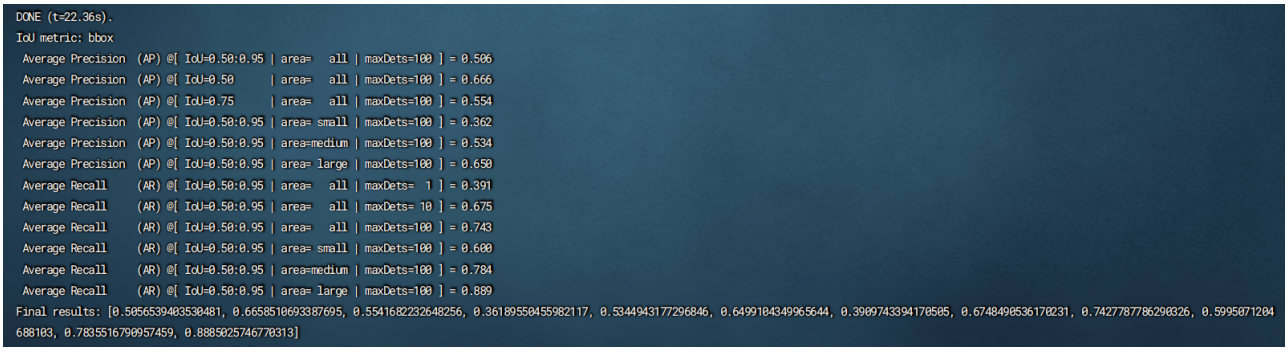
```
Accumulating evaluation results...
DONE (t=31.32s).
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.495
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.666
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 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, 0.351416395030686333336432, 0.7595056223008042, 0.8698016906969595]
```

```python
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
{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)
```

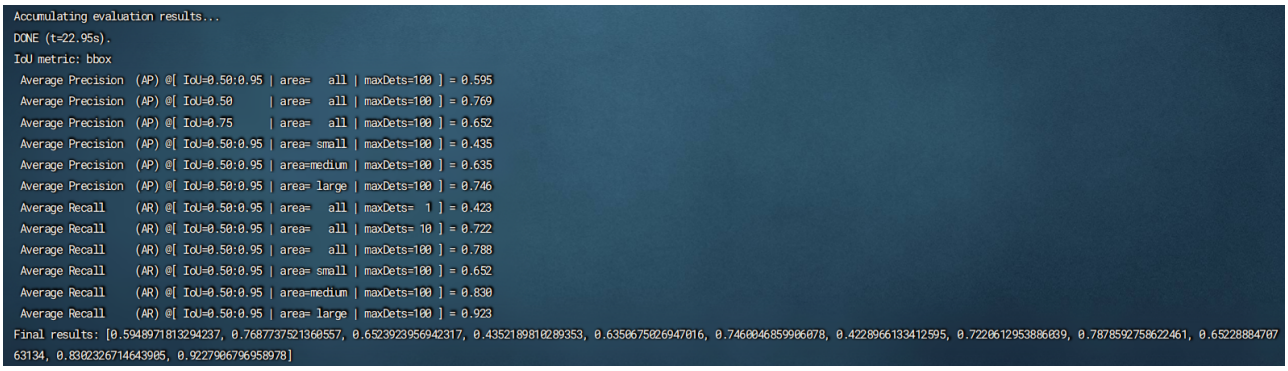
```
            new_res[..., :res.shape[-1]] = res

            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：



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：



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