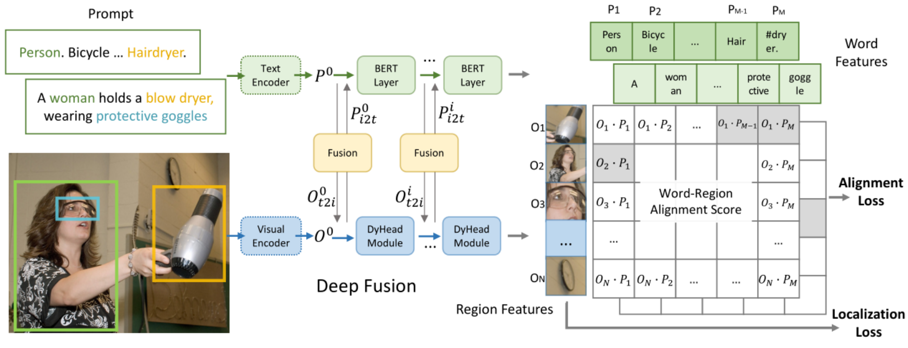
GLIP



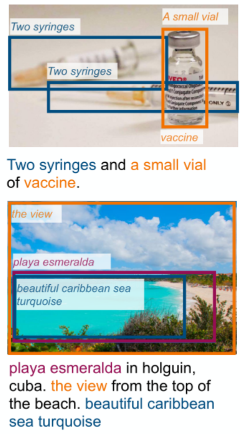


Figure 1. A unified framework for detection and grounding. Unlike a classical object detection model which predicts a categorical class for each detected object, we reformulate detection as a grounding task by aligning each region/box to phrases in a text prompt. GLIP jointly trains an image encoder and a language encoder to predict the correct pairings of regions and words. We further add the cross-modality deep fusion to early fuse information from two modalities and to learn a language-aware visual representation.

Figure 2. Grounding predictions from GLIP. GLIP can locate rare entities, phrases with attributes, and even abstract words.

Contributions:

1. Unifying detection and grounding by reformulating object detection as phrase grounding.

将目标检测和phrase grounding任务统一起来进行预训练

2.Scaling up visual concepts with massive image-text data.

用前期训练好的模型去标注一些伪标签。

3.Transfer learning with GLIP: one model for all.

迁移效果非常好，并且可以通过调整promot对不同数据集进行调优

Unified Formulation

Detction的分类损失：

$$O=\operatorname{Enc}_I(\operatorname{Img}), S_{\mathrm{cls}}=O W^T, \mathcal{L}_{\mathrm{cls}}=\operatorname{loss}\left(S_{\mathrm{cls}} ; T\right)$$

O(*N* x *d*)代表提取的图像特征，*W*(*c* x *d*)是图像分类的“权重矩阵”,*S*是分类的logits，*T*({0,1} *N* x *c*)是ground truth，*L*就是计算分类结果和标签的分类损失；N是region/box个数，d是每个特征的“通道数”，c是类别总数。

Grouding的分类损失：

$$O=\operatorname{Enc}_I(\operatorname{Img}), P=\operatorname{Enc}_L(\text { Prompt }), S_{\text {ground }}=O P^{\top}$$

O(*N* x *d*)仍然代表提取的图像特征，*P*(*M* x *d*)则是文本编码器抽取的文本特征，region-word aligment scores：*S*(*N* x *M*)则是计算*O*和*P*的余弦相似度；*M*表示word tokens的数量。

如何统一两种损失：

将检测中分类的logits换成aligment scores，但是tockens的数量*M*一般肯定大于detection标签的类别数*c*，所以一般把detection的标签拓展到*N* x *M*维，拓展的维度中，把c中标签的sub-words设为positiv，其余的均设为negative即可。

Language-Aware Deep Fusion

$$
\begin{aligned}
& O_{\mathrm{t2i}}^i, P_{\mathrm{i2t}}^i = \mathrm{X-MHA}\left(O^i, P^i\right), \quad i \in \{0, 1, . . L-1\} \\
& O^{i+1} = \mathrm{DyHeadModule}\left(O^i + O_{\mathrm{t2i}}^i\right), \quad O = O^L \\
& P^{i+1} = \mathrm{BERTLayer}\left(P^i + P_{\mathrm{i2t}}^i\right), \quad P = P^L
\end{aligned}
$$

$$
\begin{aligned}
& O^{(q)} = OW^{(q,I)}, P^{(q)} = PW^{(q,L)}, \text{ Attn } = O^{(q)}\left(P^{(q)}\right)^{\top} / \sqrt{d} \\
& P^{(v)} = PW^{(v,L)}, O_{\mathrm{t2i}} = \mathrm{SoftMax}(\text{ Attn })P^{(v)}W^{(\text{out },I)}, \\
& O^{(v)} = OW^{(v,I)}, \quad P_{i2t} = \mathrm{SoftMax}\left(\mathrm{Attn}^{\top}\right)O^{(v)}W^{(out,L)},
\end{aligned}
$$

通过cross-modality multi-head attention module(X-MHA)在前向过程中将text和image信息融合

Detail

余弦相似度：

$$
\text{similarity } = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}\left(A_i\right)^2} \times \sqrt{\sum_{i=1}^{n}\left(B_i\right)^2}}
$$

实际上就是计算两个向量的点积，即使是高维向量，使用余弦相似度来衡量两个向量的相似程度仍然非常具有参考价值，两个向量相互独立时，点积为0，两个向量相互等价时点积为1，所以可以直接和二分类的标签做损失

Focal Loss：

目标检测中，样本分布不均匀的问题是不可避免的，尤其是正负样本及不平衡的问题（预测框很多，但是真实实体数量其实很少，所以负样本数量远大于正样本，即使减小负样本权重，累加起来仍有可能使负样本破坏掉loss），和难分类样本的学习问题（置信度的正样本应该重点学习），于是在交叉熵损失的基础上改进提出了Focal Loss（Kaiming团队）

$$
\text{交叉熵损失函数：} \quad
\begin{aligned}
& CE(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \\
& p_t = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise} \end{cases} \\
& CE(p, y) = CE\left(p_t\right) = -\log\left(p_t\right)
\end{aligned}
$$

$$
\begin{gathered}
\text{二分类平衡交叉熵损失函数：} \\
CE\left(p_t\right) = -\alpha_t \log\left(p_t\right) \\
FocalLoss: \\
FL\left(p_t\right) = -\alpha_t\left(1-p_t\right)^{\gamma} \log\left(p_t\right)
\end{gathered}
$$

代码

训练调试阶段

```
#训练脚本
nohup python -u -m torch.distributed.launch --nproc_per_node=1
tools/train_net.py \
       --config-file "/home/wangxu/haoyuwang/SOD/GLIP/GLIP-
main/configs/pretrain/glip_Swin_T_O365.yaml" \
       --skip-test \
       MODEL.WEIGHT "/home/wangxu/haoyuwang/SOD/GLIP/GLIP-
main/MODEL/swin_tiny_patch4_window7_224.pth" \
       DATASETS.TRAIN '("coco_grounding_train", )' \
       MODEL.BACKBONE.FREEZE_CONV_BODY_AT -1 SOLVER.IMS_PER_BATCH 2
SOLVER.USE_AMP True SOLVER.MAX_EPOCH 24 TEST.DURING_TRAINING False
TEST.IMS_PER_BATCH 2 SOLVER.FIND_UNUSED_PARAMETERS False SOLVER.BASE_LR 0.00001
SOLVER.LANG_LR 0.00001 SOLVER.STEPS \(0.67,0.89\) DATASETS.DISABLE_SHUFFLE True
MODEL.DYHEAD.SCORE_AGG "MEAN" TEST.EVAL_TASK detection &
```

测试阶段

```
python tools/test_grounding_net.py --config-file
"/home/wangxu/haoyuwang/SOD/GLIP/GLIP-
main/configs/pretrain/glip_Swin_T_O365.yaml" --weight
"/home/wangxu/haoyuwang/SOD/GLIP/GLIP-main/OUTPUT/model_0035000.pth" \
        TEST.IMS_PER_BATCH 1 \
        MODEL.DYHEAD.SCORE_AGG "MEAN" \
        TEST.EVAL_TASK detection \
        MODEL.DYHEAD.FUSE_CONFIG.MLM_LOSS False \
        OUTPUT_DIR "/home/wangxu/haoyuwang/SOD/GLIP/GLIP-main/evaluation"
```

Few-shot test

```
 20000 iter:
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.152
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.294
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.145
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.088
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.171
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.204
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.190
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.339
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.364
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.168
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.401
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.514

 35000 iter:
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.227
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.386
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.235
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.121
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.269
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.311
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.232
```

```
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.401
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.430
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.209
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.485
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.595
```