Roadmap. The appendix is organized as follows. Detailed experimental setup are described in Section A. We further provide more experimental results in Section B.

A Detailed Experimental Setup

Our method is implemented based Detectron2 [13]. All the experiments are run on 8 V100 GPUs. The overall algorithm is given in Algorithm 1.

Transferred Datasets. Pascal VOC is a combination of VOC2007 and VOC2012 datasets with 20 object categories. This dataset contains 5k images for test. COCO 2017 is a classical benchmark dataset consisting of 80 object categories. It contains 5k images for validation. Objects365 v2 is a object detection dataset with 365 diverse object classes in the wild. It contains 30k images for validation.

Meta Prompt Learning. The context vectors for foreground and background are both randomly initialized from a Gaussian distribution with 0 mean and 0.02 standard deviation. And the length of context vectors for foreground and background are set as 8 and 10, respectively. The subset $\boldsymbol{T}_s$ contains 650 sampled base classes by default. We use jit version CLIP as vision-language model [10] during meta prompt representation learning. Following DetPro [2], we use 10% background proposals and ground-truth foreground proposals are included in the positive proposals for training. The word embedding for every class is integrated in the end of the learned foreground prompt representation. The dimension of context vector and word embedding are both 512. As demonstrated in DetPro, different positive proposals for the same object can be various, which results in different contexts. Such difference can be understood as follows: (a) given the ground-truth bounding box of an object, the prompt should be 'a photo of'; (b) given a proposal with partial object, the prompt should be 'a photo of partial'. Obviously, the learned prompt representation should be capable to represent these two different templates. So we train the prompt representation with different level contexts and ensemble the learned prompt representation. The positive proposals are divided into 5 levels by IoU range from 0.5 to 1.0 with step size 0.1. We train the prompt representation for every single level with $\mathcal{L}_p$ in Eq. (1) and $\mathcal{L}_n$ in Eq. (5). The temperature τ in Eq. (3) and Eq. (4) is set as 0.01. SGD is used for optimizing the context vectors. The learning rate is 0.002 and decayed by a step-wise scheduler, trained for 6 epochs with a batch size 512.

Open-Vocabulary Detector. We use ResNet50 [6] as the backbone and FPN [7] architecture. The default size M of memory bank for each class is 256 and sampling size m is 16. The threshold for selecting proposals U_p and U_n are 0.7 and 0.01, respectively. The temperature γ in Eq. (6) is set as 0.1 and the weight α of $\mathcal{L}_{icl}$ is initialized as 0.1 and gradually decayed during the detector training to help the convergence of other losses. Following Detic [16], we use CenterNet2 [15] with a ResNet50 backbone [6] and FPN [7] architecture. The mask prediction head is modified to a class-agnostic one and federated loss [15] is used for training. Repeat factor sampling [5] is used to balance long-tailed distributed classes. Note that CenterNet2 uses a cascade classifier [1]. The backbone ResNet50 is initialized with the pretrained weights on ImageNet21k [11]. In ICL, the projection network is a 2-layer MLP with two linear-relu layers and a normalization layer, and the dimension of proposal feature is projected from 1024 to 128. To improve the diversity of proposal samples in the instance memory bank, we gather and concat all proposal samples from 8 GPUs before updating the instance memory bank. We use AdamW [9] as the optimizer with an initial learning rate 0.0002 and batch size 64. We use EfficientDet style large scale jittering [4, 12]. To accelerate training, the input images are cropped to 640 × 640 while 800 × 1333 for inference. The

Algorithm 1 MIC for Open-Vocabulary Object Detection

```
Input: fg prompt $\boldsymbol{V}_{fg}$, bg prompt $\boldsymbol{V}_{bg}$, base classes $\mathcal{C}_B$, prompt update steps $K$ and learning rate $\eta_k$, detector $\theta$, training image $I$ with its bbox and class label $(b,c)$, detector training steps $R$ and learning rate $\eta_r$, hyper-parameter $\alpha$
Output: trained detector $\theta$
We start with procedure A to learn fg and bg prompts, then the learned prompts are used in procedure B.
procedure A. Meta Prompt Learning
    for $k = 1 \rightarrow K$ do
        Sample a batch of precomputed proposals from $\mathcal{C}_B$
        Forward proposals into CLIP to obtain $\boldsymbol{f}_p$, $\boldsymbol{f}_n$
        $\boldsymbol{T}_B = \{E_{\mathcal{T}}(\boldsymbol{V}_i)\}_{i\in\mathcal{C}_B}$, $\boldsymbol{t}_{bg} = E_{\mathcal{T}}(\boldsymbol{V}_{bg})$
        Sample $\boldsymbol{T}_S$ from $\boldsymbol{T}_B$                                  ▷ Meta sampling
        Compute $p_c^p$ by Eq. (3), and $p_c^n$ by Eq. (4)
        $\mathcal{L}_p = -\log p_c^p$                                                ▷ Eq. (1)
        $\mathcal{L}_\mathrm{n} = -\frac{1}{|\mathcal{C}_S|}\sum_{c=1}^{|\mathcal{C}_S|}\log p_c^n$                    ▷ Eq. (5)
        $\boldsymbol{V}_{fg} \leftarrow \boldsymbol{V}_{fg} - \eta_k \cdot \nabla\mathcal{L}_p$                          ▷ Update fg prompt
        $\boldsymbol{V}_{bg} \leftarrow \boldsymbol{V}_{bg} - \eta_k \cdot \nabla\mathcal{L}_n$                          ▷ Update bg prompt
    end for
end procedure
procedure B. Detector Training
    for $r = 1 \rightarrow R$ do
        Sample a batch of data $(I,(b,c))$ from $\mathcal{C}_B$
        Feed $I$ into detector to obtain $\boldsymbol{f}$ and proposal IoU
        Filter fg proposal by $U_p$ and bg proposal by $U_n$
        Update instance memory bank $\mathcal{Q}$
        Calculate contrastive loss $\mathcal{L}_{icl}$ by Eq. (6)
        $\mathcal{L}_{det} = \mathcal{L}_{rpn} + \mathcal{L}_{cls} + \mathcal{L}_{reg} + \alpha\mathcal{L}_{icl}$
        $\theta \leftarrow \theta - \eta_r \cdot \nabla\mathcal{L}_{det}$                                  ▷ Update detector $\theta$
    end for
end procedure
```

detector is trained for a 4× schedule with 90000 iterations. We perform 10000 warmup steps with 0.0001 warmup factor and use cosine learning schedule [8]. To help the convergence of other losses, the weight α of instance-level contrastive loss $\mathcal{L}_{icl}$ is decayed along with the training process by $\alpha = \alpha \times (1 - iter/90000)$.

B More Experimental Results

In this section, we show more experimental results, including the comparison of training time, visualization of latent space embedding and detection, which further indicates the effectiveness and robustness of our proposed method.

Training Time. We compare the training time of our proposed MIC with previous SOTA two-stage methods, including DetPro [2], RegionCLIP [14], PromptDet [3], as shown in Table 1. Our MIC is shown to be more efficient than previous two-stage methods.

Method	RegionCLIP [14]	DetPro [2]	PromptDet [3]	MIC (ours)
Training Time (GPU hours)	1064	464	408	368

Table 1: The training time comparison of our method with previous SOTA two-stage methods.

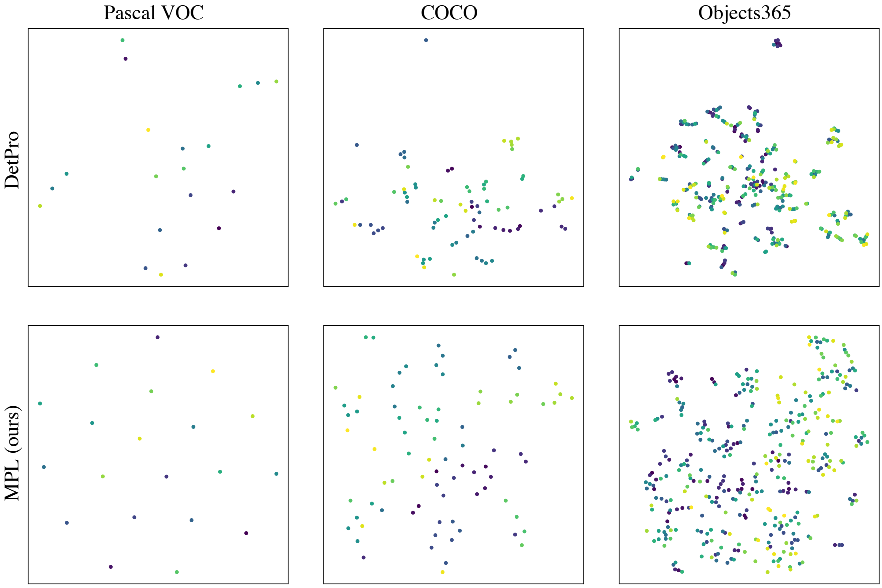

Figure 1: **t-SNE visualization of class embeddings of transferred datasets.** We use t-SNE to visualize the class embeddings of Pascal VOC, COCO, and Objects365 generated from DetPro and our proposed MPL.

Latent Space Embedding. We also use t-SNE to visualize the class embeddings of Pascal VOC, COCO, and Objects365 generated from DetPro and our proposed MPL. From Figure 1, we can draw the same conclusion as LVIS. Under our proposed MPL scheme, the learned prompt representations are more discriminative in the latent space.

Study of Failure Cases. Although we use meta prompt and instance contrastive learning to improve the discriminative ability of our model, it still suffers from distinguishing some extremely similar classes, such as 'duck' and 'duckling', and 'panda' and 'bear' shown in Figure 2. To better distinguish these categories, we might include outside knowledge base of fine-grained categories in the future work.

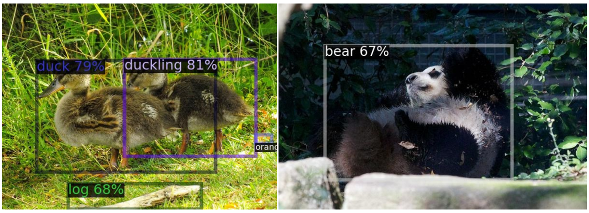

Figure 2: Failure cases of MIC.

More Comparisons of Detection Results. We show more detection visualization results in Figure 3, which further indicates the robustness of our proposed method.

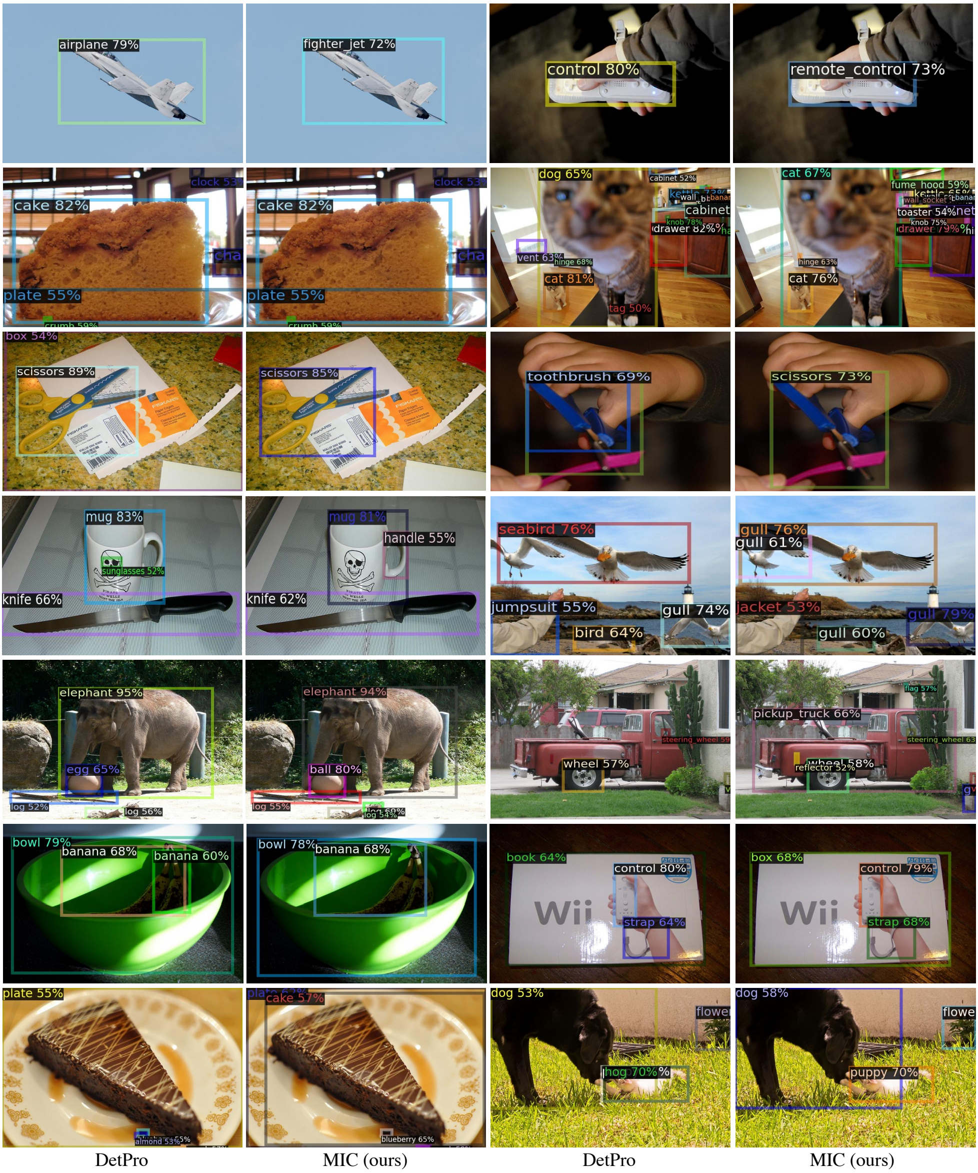

Figure 3: **More qualitative detection visualization results of our proposed method MIC and DetPro.** Our method could better distinguish similar classes, detect smaller objects, and produce less false positives under diverse scenes.

References

[1] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.

[2] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[3] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *Proceedings of the European Conference on Computer Vision*, 2022.

[4] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2918–2928, 2021.

[5] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5356–5364, 2019.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[7] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[8] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[9] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.

[11] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*, 2021.

[12] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020.

[13] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[14] Yiwu Zhong, Jianwei Yang, Pengchuan Zhang, Chunyuan Li, Noel Codella, Liunian Harold Li, Luowei Zhou, Xiyang Dai, Lu Yuan, Yin Li, et al. Regionclip: Region-based language-image pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16793–16803, 2022.

[15] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021.

[16] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Phillip Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. *European Conference on Computer Vision*, 2022.