A. Overview of Supplementary Material

The supplementary material is organized into the following sections:

- Section B: Implementation details for all experiments.
- Section C: More experiments and analysis.
- Section D: Visualization of the selective search proposals and the effectiveness of AlignDet.
- Section E: Broader impact and limitation.

B. Implementation Details

B.1. General Settings

Pre-training. All the hyper-parameters for box-domain pre-training follow the original fine-tuning settings except the prediction sampling procedure. For example, the learning rate is 2e-4 and weight decay is 1e-4 for Mask R-CNN [13] when fine-tuning on COCO [19] with ImageNet [7] pre-trained backbone. Hence in the box-domain pre-training stage, we set the same learning rate and weight decay to pre-train the modules out of the backbone. In terms of sampling predicted boxes, we select as many positive samples (predicted boxes that correspond to a ground truth proposal instead of background) as possible to expand the data for box-level contrastive learning. All methods apply the same pre-training data augmentation, which has been described in Section 4.1 of the main paper. All the experiments are pre-trained on the COCO train 2017 dataset with 12 epochs (1x), except 50 epochs for DETR [2]. During the pre-training stage, most experiments can be finished with 8 V100 GPUs (32 GB), which is efficient since we only train other modules out of the backbone (*i.e.*, neck and head).

Fine-tuning. During the fine-tuning stage, we use SyncBN [15] to calibrate magnitudes for pre-trained models following MoCo [11]. For the experiments with supervised pre-trained ResNet [14], we follow the default setting in mmdetection [4] to freeze the first layer of ResNet, and fine-tuning the other parameters under standard data augmentation with single-scale training. For the experiments with self-supervised backbones, we fine-tune all layers end-to-end with multi-scale training, and SyncBN is used across all layers, including the newly initialized batch normalization layers. For experiments with MobileNetv2 [23] and Swin Transformers [20], we follow the default training strategy defined in mmdetection. For the VOC [8] fine-tuning, we train 12k iterations to avoid over-fitting, and the learning rate is divided by 10 at $\frac{3}{4}$ and $\frac{11}{12}$ of total training time.

Since AlignDet pre-trains all modules in the detector and not just the backbone, we need to adjust the fine-tuned hyper-parameters to better transfer the pre-trained weights. Thanks to the experience of previous work [27, 24, 16], adjusting the learning rate and weight decay is a good practice. The main principle of hyper-parameter adjustment in the fine-tuning stage is to increase the learning rate while reducing weight decay. The most common setting is to increase the learning rate by 1.5 times and reduce the weight decay to half of the original value. The specific values of different methods and experiments are listed in detail in each subsequent paragraph.

B.2. FCOS

FCOS [25] is a single-stage, point-based detector. The learning rate and weight decay are 0.1, 1e-4 for AlignDet pre-training and 0.15, 5e-5 for fine-tuning, respectively. The maximum number of sampled predicted boxes for the box-domain pre-training is 2048. Other hyper-parameters are set to the default values in mmdetection.

B.3. RetinaNet

RetinaNet [18] is a single-stage, anchor-based detector. The learning rate and weight decay are 0.1, 1e-4 for AlignDet pre-training and 0.15, 5e-5 for fine-tuning, respectively. The maximum number of sampled predicted boxes for the box-domain pre-training is 2048. Other hyper-parameters are set to the default values in mmdetection.

B.4. Faster R-CNN & Mask R-CNN

Faster R-CNN [10] and Mask R-CNN [13] are two-stage, anchor-based detectors. Here Faster R-CNN uses the RoI Align [13] operation. The maximum number of sampled predicted boxes for the box-domain pre-training is 4096. All the experiments including baseline results are re-implemented with the *4conv1fc* RoI head for a fair comparison, following previous work [12, 11]. For Faster R-CNN, we fine-tune with only object detection annotations, and for Mask R-CNN, we fine-tune with both object detection and instance segmentation annotations.

Specifically, for the supervised pre-trained MobileNet v2 and ResNet backbones, the learning rate and weight decay are 0.2, 1e-4 for AlignDet pre-training and 0.3, 5e-5 for fine-tuning, respectively. In our experiments, the weight decay should be smaller for the self-supervised ResNet-50 backbones, thus we set 5e-6 for PixPro [28] and MoCo v2 [5], and the learning rate is the same as pre-training, *i.e.*, 0.02. For SwAV [3] pre-trained backbone, the fine-tuning learning rate is 3e-2, weight decay is 5e-6, and warmup iterations are 1000. For Swin Transformer backbones, the learning rate is 1e-4 and weight decay is 5e-2 for AlignDet pre-training. During the fine-tuning stage, the learning rate is 1e-4 and weight decay is 2e-2.

Algorithm I SoCo Pseudocode, PyTorch-like

```
# x: input images
# p: selective search proposals
# aug: independent random augmentation

for x, p in data_loader:
   (x1, p1), (x2, p2) = aug(x, p), aug(x, p) # augmentation
   x1, x2 = backbone_q(x1), backbone_k(x2) # updated
   x1, x2 = neck_q(z1), neck_k(z2) # feature pyramid

   # proposals as final bboxes
   b1, b2 = p1, p2
   z1, z2 = roi_align(x1, p1), roi_align(x2, p2)
   z1, z2 = g_q(z1), g_k(z2) # feature projection

   L = loss_constrastive(z1, z2) # contrastive loss
   ema_update(backbone_q, backbone_k, neck_q, neck_k)
```

Algorithm II AlignDet Pseudocode, PyTorch-like

```
# x: input images
# p: selective search proposals
# aug: independent random augmentation

for x, p in data_loader:
   (x1, p1), (x2, p2) = aug(x, p), aug(x, p) # augmentation
   x1, x2 = backbone(x1), backbone(x2) # frozen backbone
   x1, x2 = neck_q(x1), neck_k(x2) # feature pyramid

   # proposals as pseudo labels, boxes are predicted
   b1, b2 = head_q.f_reg(x1, p1), head_k.f_reg(x2, p2)
   z1, z2 = head_q.f_con(x1, b1), head_k.f_con(x2, b2)
   z1, z2 = g_q(z1), g_k(z2) # feature projection

   L = loss_con(z1, z2) + loss_reg(b1, b2, p1, p2) # losses
   ema_update(neck_q, neck_k, head_q, head_k)
```

B.5. DETR

DETR [2] is a single-stage, query-based detector. A key factor that leads to slow convergence is the complication in aligning object queries with target features in different feature embedding spaces [30]. However, in the self-supervised setting, it is difficult to achieve this alignment because we do not have accurate semantic labels. To alleviate this issue, UP-DETR [6] initializes the query embedding with features extracted from cropped image patches. DETReg [1] predict the features of cropped image patches from the corresponding query embedding via L_1 loss. However, these approaches simply use foreground or background for bipartite matching under the unsupervised setting, lacking explicit semantic information for the label assignment. This paradigm leads to the mismatch between bipartite matching costs and loss calculation, which may cause unstable matching and affect the effectiveness of pre-training.

To address this challenge, we make a small modification to AlignDet. In addition to the common coordinate-based label assignment and contrastive learning, which is the same in other methods, we also introduce the category-based assignment and corresponding loss to pre-train DETR. Specially, we crop the selective search [26] proposals from images and extract their features with supervised pre-trained backbones. Then we cluster the extracted features into 256 classes using the K-means algorithm [21, 22], the cluster results are regarded as pseudo-semantic labels to perform extra label assignment and cross-entropy loss to pre-train DETR. This has the advantage of introducing explicit category information into bipartite matching, which aligns label assignment and loss calculation in DETR, leading to more stable matching results. *Note that only DETR uses the clustering results of the features as extra pseudo-labels for box-domain pre-training, since the label assignment of other methods in this paper does not require explicit semantic information but only coordinates.*

We use both the default supervised pre-trained ResNet-50 [14] and the self-supervised pre-trained SwAV [3] for the experiments. The learning rate is 2e-4 for a batch size of 64 during the box-domain pre-training stage, and the loss weights of contrastive loss and cross-entropy loss are 1.0. In the fine-tuning stage, the learning rate is 1e-4 for the batch size of 16, and we fine-tune all the parameters following previous work [6, 1]. Other hyper-parameters are set to the default values in mmdetection.

B.6. SimMIM and CBNet v2

To further verify the effectiveness of AlignDet, we conducted advanced experiments with mask image modeling pre-training method (SimMIM [29]) and SOTA detection algorithm (CBNet v2 [17]). We chose CBNet v2 because of its open source code and achieved SOTA performance without requiring additional training data (*e.g.* training on Objects365 [24]). However, since they do not open source the training code corresponding to the most powerful model, we use the officially released code, models, and configs to reproduce the results. More specifically, we use the large scale jittering [9] to fine-tune Mask R-CNN with 3x strategy (SimMIM pre-trained Swin-Large backbone), following the settings reported in the original paper. For CBNet v2, we use the publicly released config [17] to reproduce the results. Both external links are existing implementations that follow original papers, not part of our submission.

C. Further Analysis and Experiments

C.1. Pre-training with Longer Epochs

Pre-training the backbone for longer epochs does not necessarily lead to sustained performance improvements for downstream tasks, both for supervised [12] and self-supervised pre-training methods [27, 28, 11]. Here we find similar results on AlignDet, that is, 12 epochs pre-training is enough for AlignDet, as shown in Table 1 with RetinaNet. However, the pre-training for the backbone are usually hundreds of epochs. A potential reason for this phenomenon is that the pre-training parameters of the two are significantly different. In most object detection models, the number of parameters of the backbone is much more than that of the neck and head modules, so backbone pre-training often requires longer pre-training epochs to learn meaningful repre-

Pre-training Schedule	AP	AP_{50}	AP_{75}	AP_s	AP_m	AP_l
1x	**37.3**	**56.6**	**40.1**	**21.0**	**40.9**	**49.8**
2x	37.0	56.2	39.3	20.8	40.6	49.5
3x	37.0	56.1	39.5	20.4	40.6	48.7

Table 1. Ablation study on pre-training schedules. All the results are fine-tuned with 12 epochs (1x schedule).

sentation. On the contrary, since the neck and head modules have relatively few parameters, they can be well-trained with fewer epochs. Thus a longer pre-training time may lead to over-fitting and will not bring additional improvements. In addition, detection datasets such as COCO [19] are usually smaller than pre-training datasets (*e.g.*, ImageNet [7]), which may exacerbate this issue.

D. Visualization

D.1. Selective Search Proposals

We use the same selective search code and filtering strategy as SoCo [27] on the COCO train 2017 dataset, and apply non-maximum suppression (NMS) with a threshold of 0.5 at the end to remove redundant proposals. The images used for box-domain pre-training are shown in Figure 2.

D.2. Effectiveness of Box-domain Pre-training

Due to the significant differences in the design and mechanism of different detection methods, we need to design different visualization schemes to verify the effectiveness of our AlignDet under the unsupervised setting.

Faster R-CNN & Mask R-CNN. In this paper, the structural difference between Faster R-CNN and Mask R-CNN is only the presence or absence of a mask head, so they have the same prediction results and visualization for the detection task. In addition to the main paper, we also provide more visualizations here in Figure 3. Specifically, we use RPN to determine which of the predicted boxes are foregrounds and feed them into the head to get the predicted box coordinates. We plot the centers of these boxes instead of rectangles for better visualization. AlignDet focuses on objects instead of messy pixels compared to the random initialization results without box-domain pre-training.

Other Methods. Unlike Faster R-CNN or Mask R-CNN, other methods do not have an RPN module, which means we cannot determine which predicted boxes are foreground and which are background during inference. To demonstrate the effectiveness of AlignDet, we show the training losses in Figure 1 using RetinaNet as an example. AlignDet pre-training significantly accelerates the convergence of the model, with lower classification loss *loss_cls* and regression loss *loss_bbox* under the same training iterations.

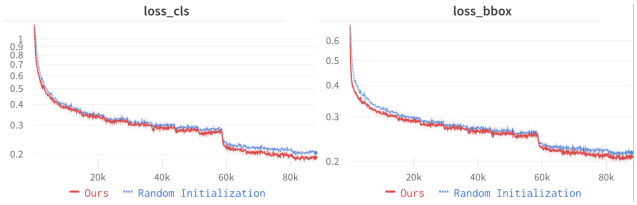


Figure 1. Fine-tuning losses of RetinaNet on COCO train 2017.

In addition, we also show the fine-tuning results of different detection models with or without AlignDet pre-training in Figure 4 to further demonstrate the effectiveness of our AlignDet. AlignDet achieves more accurate classification and precise coordinate results than the random initialization results without box-domain pre-training.

E. Broader Impact and Limitation

AlignDet represents a significant step forward in the development of unified and adequate unsupervised detection pre-training. Our approach enables the fully self-supervised pre-training of various object detection models, a milestone that was previously unattainable. Furthermore, the decoupled pre-training paradigm delivers highly efficient and effective pre-training, by separating the feature extraction from task-aware learning. **The decoupled pre-training paradigm can be readily extended to other vision tasks, allowing the integration of general-purpose pre-trained backbones with task-aware pre-trained necks and heads, which opens a door for solving the discrepancies between general pre-training and various downstream tasks.**

However, the dependence on selective search proposals in this paper may represent a potential limitation, we view it as a direction for future research. Overall, our work advances the state-of-the-art unsupervised detection pre-training and offers significant potential for improving the performance of object detection.

References

[1] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roei Herzig, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Detreg: Unsupervised pretraining with region priors for object detection. In *CVPR*, 2022.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020.

[3] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *NeurIPS*, 2020.

[4] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[5] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[6] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, 2021.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[8] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[9] 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 *CVPR*, 2021.

[10] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[11] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *CVPR*, 2020.

[12] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *ICCV*, 2019.

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, 2017.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[15] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[16] Yanghao Li, Hanzi Mao, Ross Girshick, and Kaiming He. Exploring plain vision transformer backbones for object detection. *arXiv preprint arXiv:2203.16527*, 2022.

[17] Tingting Liang, Xiaojie Chu, Yudong Liu, Yongtao Wang, Zhi Tang, Wei Chu, Jingdong Chen, and Haibin Ling. Cbnet: A composite backbone network architecture for object detection. *TIP*, 2022.

[18] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[20] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[21] Stuart Lloyd. Least squares quantization in pcm. *IEEE transactions on information theory*, 1982.

[22] I MacQueen. Some methods for classifiction and analysis of multivariate observations. In *Proceedings 5th Berkeley Symposium on Mathematical Statistics Problems*, 1967.

[23] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[24] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365: A large-scale, high-quality dataset for object detection. In *ICCV*, 2019.

[25] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, 2019.

[26] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *IJCV*, 2013.

[27] Fangyun Wei, Yue Gao, Zhirong Wu, Han Hu, and Stephen Lin. Aligning pretraining for detection via object-level contrastive learning. *NeurIPS*, 2021.

[28] Zhenda Xie, Yutong Lin, Zheng Zhang, Yue Cao, Stephen Lin, and Han Hu. Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning. In *CVPR*, 2021.

[29] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *CVPR*, 2022.

[30] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *CVPR*, 2022.

Figure 2. Selective search proposals on COCO train 2017 dataset.

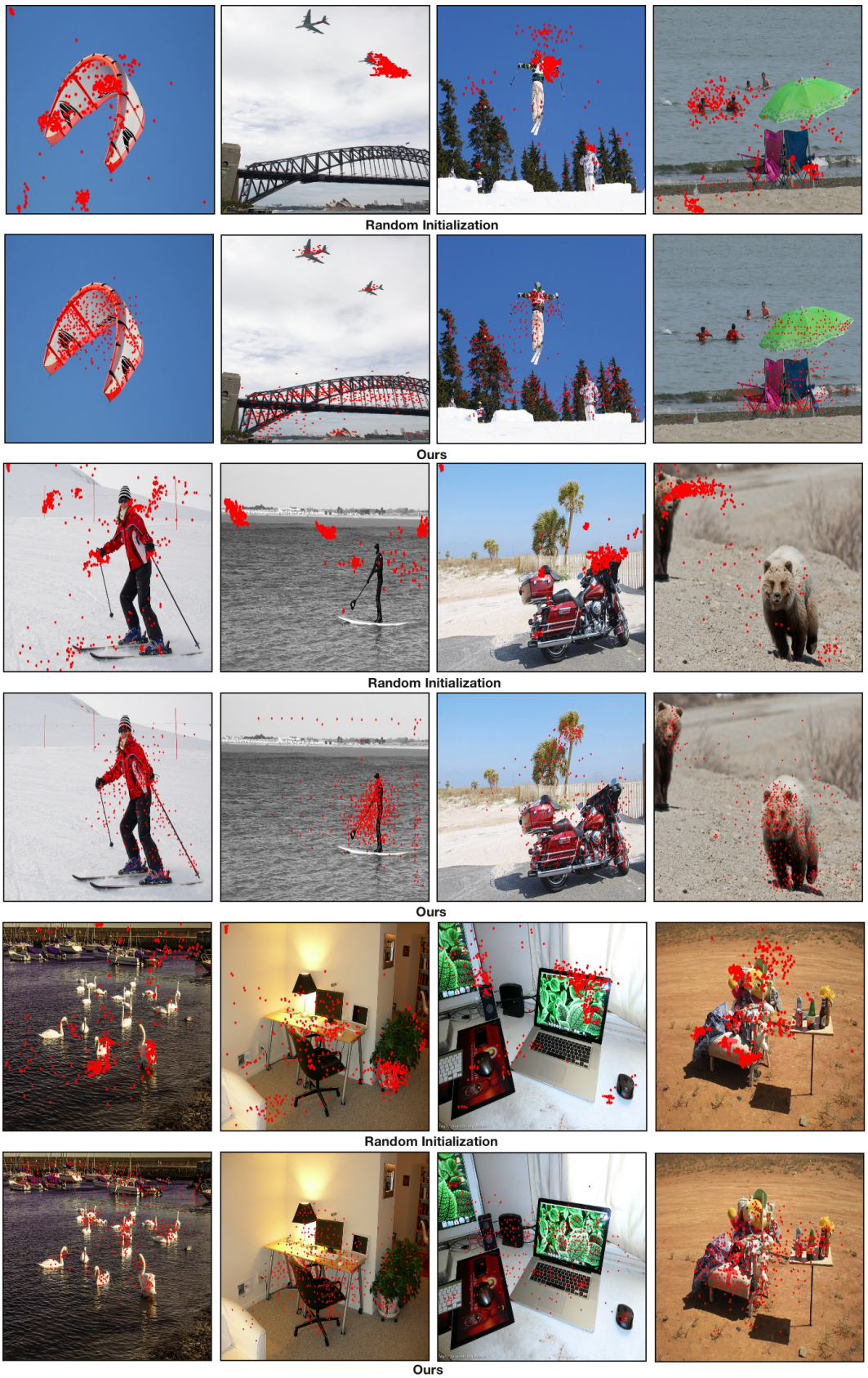
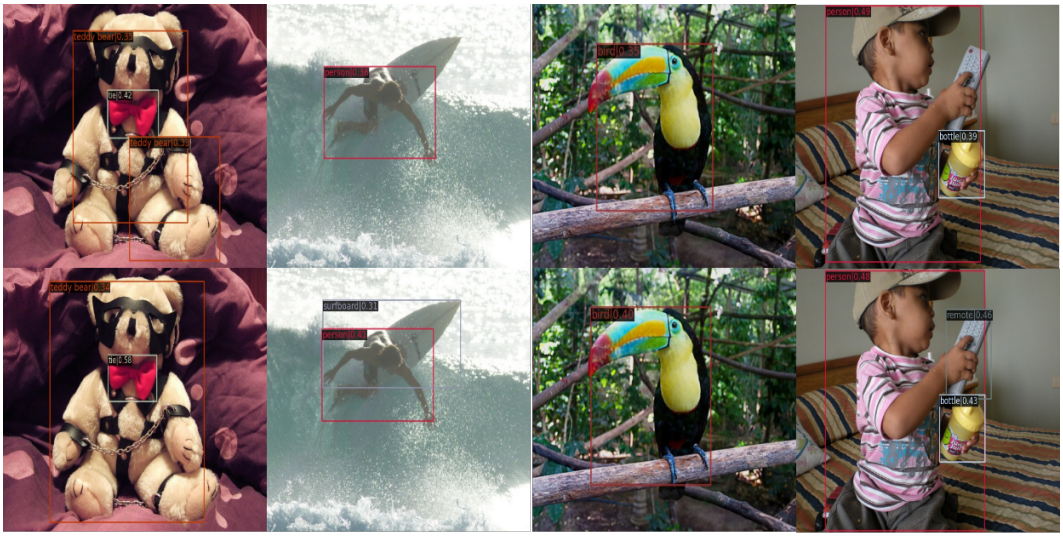


Figure 3. Visualization results of predictions on COCO Val 2017 with Faster/Mask R-CNN. Random Initialization denotes ImageNet pre-train, and ours means AlignDet pre-training.



Detection Results of FCOS

Detection Results of DETR

Detection Results of RetinaNet

Figure 4. Detection results with different models on the public COCO Val 2017 dataset. For each scene, the upper images are the fine-tuning results without box-domain pre-training, and the lower images are the fine-tuning results after the box-domain pre-training.