UP-DETR: Unsupervised Pre-training for Object Detection with Transformers

Zhigang Dai[1,2,3*], Bolun Cai[2], Yugeng Lin[2], Junying Chen[1,3†]
[1]School of Software Engineering, South China University of Technology
[2]Tencent Wechat AI
[3]Key Laboratory of Big Data and Intelligent Robot
(South China University of Technology), Ministry of Education
zhigangdai@hotmail.com, {arlencai,lincolnlin}@tencent.com, jychense@scut.edu.cn

Abstract

Object detection with transformers (DETR) reaches competitive performance with Faster R-CNN via a transformer encoder-decoder architecture. Inspired by the great success of pre-training transformers in natural language processing, we propose a pretext task named ***random query patch detection*** *to Unsupervisedly Pre-train DETR (****UP-DETR****) for object detection. Specifically, we randomly crop patches from the given image and then feed them as queries to the decoder. The model is pre-trained to detect these query patches from the original image. During the pre-training, we address two critical issues: multi-task learning and multi-query localization. (1) To trade off classification and localization preferences in the pretext task, we freeze the CNN backbone and propose a patch feature reconstruction branch which is jointly optimized with patch detection. (2) To perform multi-query localization, we introduce UP-DETR from single-query patch and extend it to multi-query patches with object query shuffle and attention mask. In our experiments, UP-DETR significantly boosts the performance of DETR with faster convergence and higher average precision on object detection, one-shot detection and panoptic segmentation. Code and pre-training models:* https://github.com/dddzg/up-detr.

1. Introduction

Object detection with transformers (DETR) [5] is a recent framework that views object detection as a direct prediction problem via a transformer encoder-decoder [39]. Without hand-designed sample selection [46] and non-maximum suppression, DETR reaches a competitive performance with Faster R-CNN [34]. However, DETR comes with training and optimization challenges, which needs large-scale training data and an extreme long training schedule. As shown in Fig. 1 and Section 4.1, we find that DETR performs poorly in PASCAL VOC [13], which has insufficient training data and fewer instances than COCO [28].

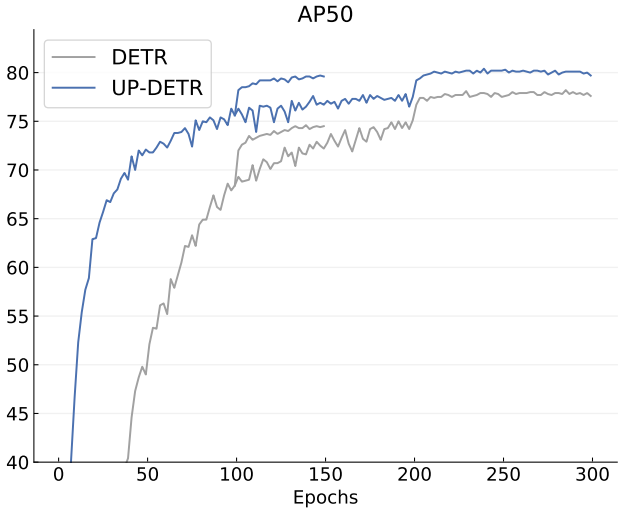


Figure 1: The VOC learning curves (AP_{50}) of DETR and UP-DETR with ResNet-50 backbone. Here, they are trained on `trainval07+12` and evaluated on `test2007`. We plot the short and long training schedules, and the learning rate is reduced at 100 and 200 epochs, respectively.

With well-designed pretext tasks, unsupervised pre-training models achieve remarkable progress in both natural language processing (*e.g.* GPT [32, 33] and BERT [11]) and computer vision (*e.g.* MoCo [16, 9] and SwAV [7]). In DETR, the CNN backbone (ResNet-50 [19] with ~23.2M parameters) has been pre-trained to extract a good visual representation, but the transformer module with ~18.0M parameters has not been pre-trained. More importantly,

*This work is done when Zhigang Dai was an intern at Tencent Wechat AI.
†Corresponding author.

although unsupervised visual representation learning (*e.g*. contrastive learning) attracts much attention in recent studies [16, 8, 14, 4, 6, 1], existing pretext tasks can not directly apply to pre-train the transformers of DETR. The main reason is that DETR mainly focuses on spatial localization learning instead of image instance-based [16, 8, 14] or cluster-based [4, 6, 1] contrastive learning.

Inspired by the great success of unsupervised pre-training in natural language processing [11], we aim to unsupervisedly pre-train the transformers of DETR on a large-scale dataset (*e.g*. ImageNet), and treat object detection as the downstream task. The motivation is intuitive, but existing pretext tasks seem to be impractical to pre-train the transformers of DETR. To overcome this problem, we propose **Unsupervised Pre-training DETR (UP-DETR)** with a novel unsupervised pretext task named **random query patch detection** to pre-train the detector without any human annotations — we *randomly* crop multiple *query patches* from the given image, and pre-train the transformers for *detection* to predict bounding boxes of these query patches in the given image. During the pre-training procedure, we address two critical issues as follows:

(1) Multi-task learning: Object detection is the coupling of object classification and localization. To avoid query patch detection destroying the classification features, we introduce **frozen pre-training backbone** and **patch feature reconstruction** to preserve the feature discrimination of transformers.

(2) Multi-query localization: Different object queries focus on different position areas and box sizes. To illustrate this property, we propose a simple single-query pre-training and extend it to a multi-query version. For multi-query patches, we design **object query shuffle** and **attention mask** to solve the assignment problems between query patches and object queries.

In our experiments, UP-DETR performs better than DETR on PASCAL VOC [13] and COCO [28] object detection with faster convergence and better average precision. Besides, UP-DETR also transfers well with state-of-the-art performance on one-shot detection and panoptic segmentation. In ablations, we find that **freezing the pre-training CNN backbone** is the most important procedure to preserve the feature discrimination during the pre-training.

2. Related Work

2.1. Object Detection

Most object detection methods mainly differ in positive and negative sample assignment. Two-stage detectors [34, 3] and a part of one-stage detectors [27, 29] construct positive and negative samples by hand-crafted multi-scale anchors with the IoU threshold and model confidence. Anchor-free one-stage detectors [38, 48, 22] assign positive and negative samples to feature maps by a grid of object centers. Zhang *et al*. [46] demonstrate that the performance gap between them is due to the selection of positive and negative training samples. DETR [5] is a recent object detection framework that is conceptually simpler without hand-crafted process by direct set prediction [37], which assigns the positive and negative samples automatically.

Apart from the positive and negative sample selection problem, the trade-off between classification and localization is also intractable for object detection. Zhang *et al*. [45] demonstrate that there is a domain misalignment between classification and localization. Wu *et al*. [40] and Song *et al*. [35] design two head structures for classification and localization. They point out that these two tasks may have opposite preferences. For our pre-training model, it maintains shared feature for classification and localization. Therefore, it is essential to take a well trade-off between these two tasks.

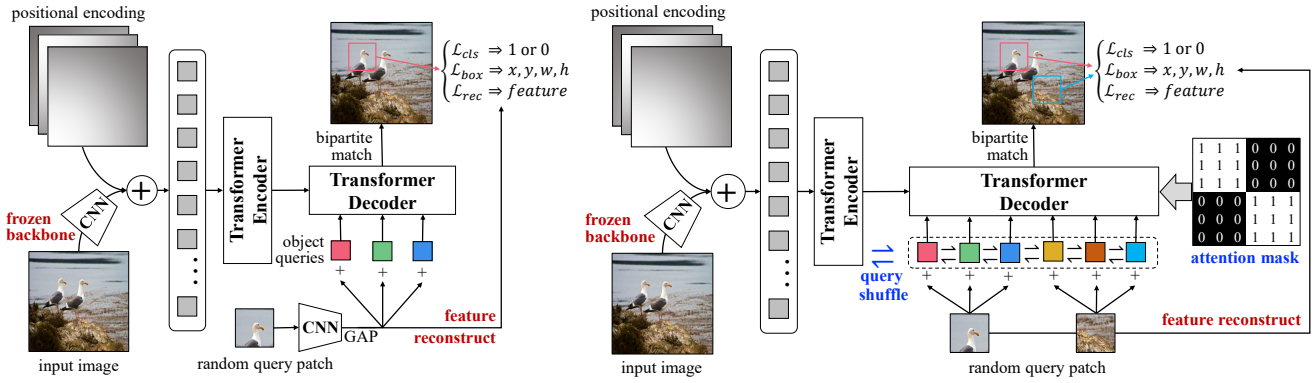
2.2. Unsupervised Pre-training

Unsupervised pre-training models always follow two steps: pre-training on a large-scale dataset with the pretext task and fine-tuning the parameters on downstream tasks. For unsupervised pre-training, the pretext task is always invented, and we are interested in the learned intermediate representation rather than the final performance of the pretext task.

To perform unsupervised pre-training, there are various of well-designed pretext tasks. For natural language processing, utilizing time sequence relationship between discrete tokens, masked language model [11], permutation language model [43] and auto regressive model [32, 33] are proposed to pre-train transformers [39] for language representation. For computer vision, unsupervised pre-training models also achieve remarkable progress recently for visual representation learning, which outperform the supervised learning counterpart in downstream tasks. Instance-based discrimination tasks [44, 41] and clustering-based tasks [6] are two typical pretext tasks in recent studies. Instance-based discrimination tasks vary mainly on maintaining different sizes of negative samples [16, 8, 14] with non-parametric contrastive learning [15]. Moreover, instance discrimination can also be performed as parametric instance classification [4]. Clustering-based tasks vary on offline [6, 1] or online clustering procedures [7]. UP-DETR is a novel pretext task, which aims to pre-train transformers based on the DETR architecture for object detection.

3. UP-DETR

The proposed UP-DETR contains pre-training and fine-tuning procedures: (a) the transformers are unsupervisedly *pre-trained* on a large-scale dataset without any human an-



(a) single-query patch (N=3, M=1)

(b) multi-query patch (N=6, M=2)

Figure 2: The pre-training procedure of UP-DETR by random query patch detection. (a) There is only a single-query patch which we add to all object queries. (b) For multi-query patches, we add each query patch to N/M object queries with object query shuffle and attention mask. CNN is not drawn in the decoder of (b) for neatness.

notations; (b) the entire model is *fine-tuned* with labeled data which is same as the original DETR [5] on the downstream tasks. In this section, we mainly describe how to pre-train the transformer encoder and decoder with random query patch detection.

As shown in Fig. 2, the main idea of random query patch detection is simple but effective. Firstly, a frozen CNN backbone is used to extract a visual representation with the feature map $f \in \mathbb{R}^{C\times H\times W}$ of an input image, where C is the channel dimension and $H \times W$ is the feature map size. Then, the feature map is added with positional encodings and passed to the multi-layer transformer encoder in DETR. For the random cropped query patch, the CNN backbone with global average pooling (GAP) extracts the patch feature $p \in \mathbb{R}^C$, which is flatten and supplemented with object queries $q \in \mathbb{R}^C$ before passing it into a transformer decoder. Noting that the *query patch* refers to the cropped patch from the original image but *object query* refers to position embeddings, which are fed to the decoder. The CNN parameters are shared in the whole model.

During the pre-training procedure, the decoder predicts the bounding boxes corresponding to the position of random query patches in the input image. Assuming that there are M query patches by random cropping, the model infers a prediction fixed-set $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ corresponding to N object queries ($N > M$). For better understanding, we will describe the training details of single-query patch ($M = 1$) in Section 3.1, and extend it to multi-query patches ($M > 1$) with object query shuffle and attention mask in Section 3.2.

3.1. Single-Query Patch

DETR learns different spatial specialization for each object query [5], which indicates that different object queries focus on different position areas and box sizes. As we randomly crop the patch from the image, there is no any priors about the position areas and box sizes of the query patch. To preserve the different spatial specialization, we explicitly specify single-query patch ($M = 1$) to all object queries ($N = 3$) as shown in Fig. 2a.

During the pre-training procedure, the patch feature p is added to each different object query q, and the decoder generates N pairs of predictions $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ to detect the bounding box of query patch in the input image. Following DETR [5], we compute the same match cost between the prediction $\hat{y}_{\hat{\sigma}(i)}$ and the ground-truth y_i using *Hungarian* algorithm [37], where $\hat{\sigma}(i)$ is the index of y_i computed by the optimal bipartite matching.

For the loss calculation, the predicted result $\hat{y}_i = (\hat{c}_i \in \mathbb{R}^2, \hat{b}_i \in \mathbb{R}^4, \hat{p}_i \in \mathbb{R}^C)$ consists of three elements: $\hat{c}_i$ is the binary classification of matching the query patch ($c_i = 1$) or not ($c_i = 0$) for each object query; $\hat{b}_i$ is the vector that defines the box center coordinates, its width and height $\{x, y, w, h\}$. They are re-scaled relative to the image size; $\hat{p}_i$ is the reconstructed feature with $C = 2048$ for the ResNet-50 backbone typically. With the above definitions, the *Hungarian* loss for all matched pairs is defined as:

$$\mathcal{L}(y, \hat{y}) = \sum_{i=1}^{N} [\lambda_{\{c_i\}} \mathcal{L}_{cls}(c_i, \hat{c}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}} \mathcal{L}_{box}(b_i, \hat{b}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}} \mathcal{L}_{rec}(p_i, \hat{p}_{\hat{\sigma}(i)})]. \quad (1)$$

Here, $\mathcal{L}_{cls}$ is the cross entropy loss over two classes (match the query patch *vs.* not match), and the class balance weight $\lambda_{\{c_i=1\}} = 1$ and $\lambda_{\{c_i=0\}} = M/N$. $\mathcal{L}_{box}$ is a linear combination of ℓ_1 loss and the generalized IoU loss with the same weight hyper-parameters as DETR [5]. $\mathcal{L}_{rec}$ is the reconstruction loss proposed in this paper to balance classification and localization during the unsupervised pre-training,

which will be discussed in detail below.

3.1.1 Patch Feature Reconstruction

Object detection is the coupling of object classification and localization, where these two tasks always have different feature preferences [45, 40, 35]. Different from DETR, we propose a feature reconstruction term $\mathcal{L}_{rec}$ to preserve classification feature during localization pre-training. The motivation of this term is to preserve the feature discrimination extract by CNN after passing feature to transformers. $\mathcal{L}_{rec}$ is the mean squared error between the ℓ_2-normalized patch feature extracted by the CNN backbone, which is defined as follows:

$$\mathcal{L}_{rec}(p_i, \hat{p}_{\hat{\sigma}(i)}) = \left\| \frac{p_i}{\|p_i\|_2} - \frac{\hat{p}_{\hat{\sigma}(i)}}{\left\|\hat{p}_{\hat{\sigma}(i)}\right\|_2} \right\|_2^2. \quad (2)$$

3.1.2 Frozen Pre-training Backbone

With the patch feature reconstruction, the CNN backbone parameters seriously affect the model training. Our motivation is that the feature after transformer should have similar discrimination as the feature after the CNN backbone. Therefore, we freeze the pre-training backbone and reconstruct the patch feature after the transformers by $\mathcal{L}_{rec}$. Stable backbone parameters are beneficial to transformer pre-training, and accelerate the feature reconstruction.

As described above, we propose and apply feature reconstruction and frozen backbone to preserve feature discrimination for classification. In Section 4.5.1, we will analyze and verify the necessity of them with experiments.

3.2. Multi-Query Patches

For general object detection, there are multiple object instances in each image (*e.g.* average 7.7 object instances per image in the COCO dataset). Moreover, single-query patch may result in the convergence difficulty when the number of object queries N is large. Therefore, single-query patch pre-training is inconsistent with multi-object detection task, and is unreasonable for the typical object query setting $N = 100$. However, extending a single-query patch to multi-query patches is not straightforward, because the assignment between M query patches and N object queries is a specific negative sampling problem for multi-query patches.

To solve this problem, we divide N object queries into M groups, where each query patch is assigned to N/M object queries. The query patches are assigned to the object queries in order. For example, the first query patch is assigned to the first N/M object queries, the second query patch to the second N/M object queries, and so on. Here, we hypothesize that it needs to satisfy two requirements during the pre-training: **(1) Independence of query patches**. All the query patches are randomly cropped from the image. Therefore, they are independent without any relations. For example, the bounding box regression of the first cropping is not concerned with the second cropping. **(2) Diversity of object queries**. There is no explicit group assignment between object queries for the downstream tasks. In other words, the query patch can be added to arbitrary N/M object queries ideally.

3.2.1 Attention Mask

To satisfy the independence of query patches, we utilize an attention mask matrix to control the interactions between different object queries. The mask matrix $\mathbf{X} \in \mathbb{R}^{N \times N}$ is added to the softmax layer of self-attention in the decoder $softmax\left(QK^\top/\sqrt{d_k} + \mathbf{X}\right)\mathbf{V}$. Similar to the token mask in UniLM [12], the attention mask is defined as:

$$\mathbf{X}_{i,j} = \begin{cases} 0, & \text{i, j in the same group} \\ -\infty, & \text{otherwise} \end{cases}, \quad (3)$$

where $\mathbf{X}_{i,j}$ determines whether the object query q_i attends to the interaction with the object query q_j. For intuitive understanding, the attention mask in Fig. 2b displays 1 and 0 corresponding to 0 and $-\infty$ in (3), respectively.

3.2.2 Object Query Shuffle

Groups of object queries are assigned artificially. However, during the downstream object detection tasks, there are no explicit group assignment between object queries. Therefore, To simulate implicit group assignment between object queries, we randomly shuffle the permutation of all the object query embeddings during pre-training [3].

Fig. 2b illustrates the pre-training of multi-query patches with attention mask and object query shuffle. To improve the generalization, we randomly mask 10% query patches to zero during pre-training similarly to dropout [36]. In our experiments, two typical values are set to $N = 100$ and $M = 10$. Apart from such modifications, other training settings are the same as those described in Section 3.1.

4. Experiments

We pre-train the UP-DETR using ImageNet [10] and fine-tune the parameters on VOC [13] and COCO [28] for object detection, one-shot detection and panoptic segmentation. In all experiments, we adopt the UP-DETR model (41.3M parameters) with ResNet-50 backbone, 6 transformer encoder, 6 decoder layers of width 256 with 8 attention heads. Referring to the open source of DETR[4], we use *the same hyper-parameters* in the proposed UP-DETR

[3] In our further study, we find that the object query shuffle is not helpful. More details are included in the appendix.

[4] https://github.com/facebookresearch/detr

and our DETR re-implementation. We annotate R50 and R101 short for ResNet-50 and ResNet-101.

Pre-training setup. UP-DETR is pre-trained on the ImageNet training set without any labels. The CNN backbone (ResNet-50) is pre-trained with SwAV [7]. As the input image from ImageNet is relatively small, we resize it such that the shortest side is within $[320, 480]$ pixels while the longest side is at most 600 pixels. Given the image, we crop the query patches with random coordinate, height and width, which are resized to 128×128 pixels and transformed with the SimCLR-style [8] without horizontal flipping. AdamW [30] is used to optimize the UP-DETR, with the initial learning rate of 1×10^{-4} and the weight decay of 1×10^{-4}. We use a mini-batch size of 256 on 8 V100 GPUs for 60 epochs with the learning rate multiplied by 0.1 at 40 epochs.

Fine-tuning setup. The model is initialized with pre-training UP-DETR parameters and fine-tuned for all the parameters (including CNN) on VOC and COCO. We fine-tune the model with the initial learning rate 1×10^{-4} for transformers and 5×10^{-5} for CNN backbone, and the other settings are same as DETR [5] on 8 V100 GPUs. The model is fine-tuned with short/long schedule for 150/300 epochs and the learning rate is multiplied by 0.1 at 100/200 epochs, respectively.

4.1. PASCAL VOC Object Detection

Setup. The model is fine-tuned on VOC `trainval07+12` (∼16.5k images) and evaluated on `test2007`. We report COCO-style metrics: AP, AP_{50} (default VOC metric) and AP_{75}. For a full comparison, we report the result of Faster R-CNN with the R50-C4 backbone [7], which performs much better than R50 [25]. DETR with R50-C4 significantly increases the computational cost than R50, so we fine-tune UP-DETR with R50 backbone.

Model/Epoch	AP	AP_{50}	AP_{75}
Faster R-CNN	56.1	**82.6**	**62.7**
DETR/150	49.9	74.5	53.1
UP-DETR/150	**56.1** (+6.2)	79.7 (+5.2)	60.6 (+7.5)
DETR/300	54.1	78.0	58.3
UP-DETR/300	**57.2** (+3.1)	80.1 (+2.1)	62.0 (+3.7)

Table 1: Object detection results trained on PASCAL VOC `trainval07+12` and evaluated on `test2007`. DETR and UP-DETR use R50 backbone and Faster R-CNN uses R50-C4 backbone. The values in the brackets are the gaps compared to DETR with the same training schedule.

Results. Table 1 shows the compared results of PASCAL VOC. We find that the DETR performs poorly in PASCAL VOC, which is much worse than Faster R-CNN by a large gap in all metrics. UP-DETR significantly boosts the performance of DETR for both short and long schedules: up to **+6.2 (+3.1)** AP, **+5.2 (+2.1)** AP_{50} and **+7.5 (+3.7)** AP_{75} for 150 (300) epochs, respectively. Moreover, UP-DETR (R50) achieves a comparable result to Faster R-CNN (R50-C4) with better AP. We find that both UP-DETR and DETR perform a little worse than Faster R-CNN in AP_{50} and AP_{75}. It may come from different ratios of feature maps (C4 for Faster R-CNN) and no NMS post-processing (NMS lowers AP but slightly improves AP_{50}).

Fig. 3a shows the AP (COCO style) learning curves on VOC. UP-DETR significantly speeds up the model convergence. After the learning rate reduced, UP-DETR significantly boosts the performance of DETR with a large AP improvement. Noting that UP-DETR obtains 56.1 AP after 150 epochs, however, its counterpart DETR (scratch transformers) only obtains 54.1 AP even after 300 epochs and does not catch up even training longer. It suggests that pre-training transformers is indispensable on insufficient training data (*i.e.* $\sim$ 16.5K images on VOC).

4.2. COCO Object Detection

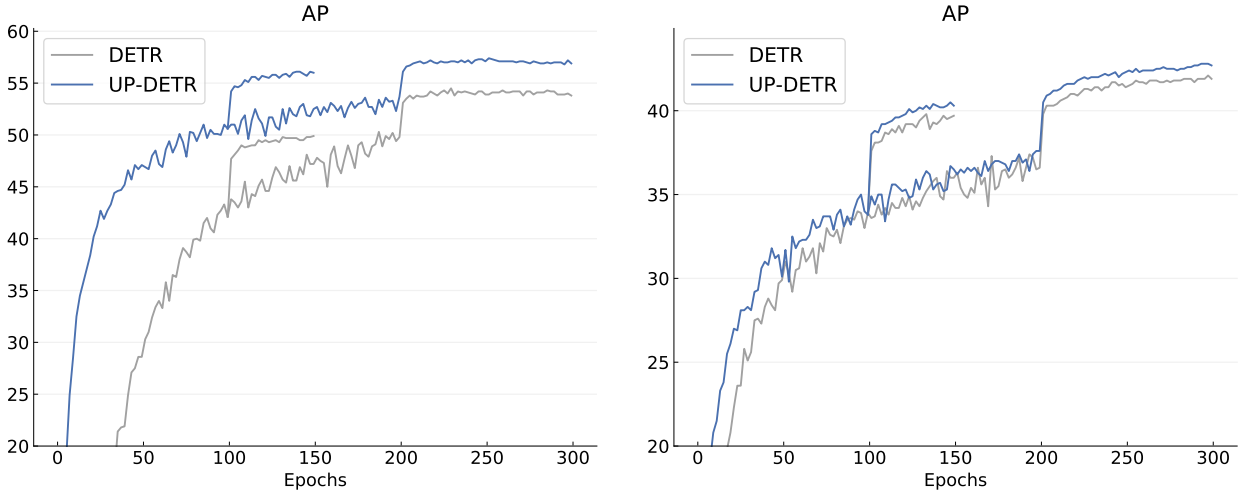
Setup. The model is fine-tuned on COCO `train2017` (∼118k images) and evaluated on `val2017`. There are lots of small objects in COCO dataset, where DETR performs poorly [5]. Therefore, we report AP, AP_{50}, AP_{75}, AP_S, AP_M and AP_L for a comprehensive comparison. Moreover, we also report the results of highly optimized Faster R-CNN-FPN with short ($3\times$) and long ($9\times$) training schedules, which are known to improve the performance results [17].

Results. Table 2 shows the results on COCO with other methods. With 150 epoch schedule, UP-DETR outperforms DETR by 0.8 AP and achieves a comparable performance as compared with Faster R-CNN-FPN ($3 \times$ schedule). With 300 epoch schedule, UP-DETR obtains **42.8** AP on COCO, which is 0.7 AP better than DETR (SwAV CNN) and 0.8 AP better than Faster R-CNN-FPN ($9 \times$ schedule). Overall, UP-DETR comprehensively outperforms DETR in detection of small, medium and large objects with both short and long training schedules. Regrettably, UP-DETR is still slightly lagging behind Faster R-CNN in AP_S, because of the lacking of FPN-like architecture [26] and the high-cost attention operation.

Fig. 3b shows the AP learning curves on COCO. UP-DETR outperforms DETR for both 150 and 300 epoch schedules with faster convergence. The performance improvement is more noticeable before reducing the learning rate. After reducing the learning rate, UP-DETR still holds the lead of DETR by $\sim$ 0.7 AP improvement. It suggests that pre-training transformers is still indispensable even on sufficient training data (*i.e.* $\sim$ 118K images on COCO).

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN † [26]	R101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN † [18]	R101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Grid R-CNN † [31]	R101-FPN	-	41.5	60.9	44.5	23.3	44.9	53.1
Double-head R-CNN [40]	R101-FPN	-	41.9	62.4	45.9	23.9	45.2	55.8
RetinaNet † [27]	R101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
FCOS † [38]	R101-FPN	-	41.5	60.7	45.0	24.4	44.8	51.6
DETR [5]	R50	500	42.0	62.4	44.2	20.5	45.8	61.1
Faster R-CNN	R50-FPN	3×	40.2	**61.0**	**43.8**	**24.2**	43.5	52.0
DETR (Supervised CNN)	R50	150	39.5	60.3	41.4	17.5	43.0	59.1
DETR (SwAV CNN) [7]	R50	150	39.7	60.3	41.7	18.5	43.8	57.5
UP-DETR	R50	150	**40.5** (+0.8)	60.8	42.6	19.0	**44.4**	**60.0**
Faster R-CNN	R50-FPN	9×	42.0	62.1	**45.5**	**26.6**	45.4	53.4
DETR (Supervised CNN)	R50	300	40.8	61.2	42.9	20.1	44.5	60.3
DETR (SwAV CNN) [7]	R50	300	42.1	**63.1**	44.5	19.7	46.3	60.9
UP-DETR	R50	300	**42.8** (+0.7)	63.0	45.3	20.8	**47.1**	**61.7**

Table 2: Object detection results trained on COCO `train2017` and evaluated on `val2017`. Faster R-CNN, DETR and UP-DETR are performed under comparable settings. † for values evaluated on COCO `test-dev`, which are always slightly higher than `val2017`. The values in the brackets are the gaps compared to DETR.



(a) AP learning curves of VOC

(b) AP learning curves of COCO

Figure 3: AP (COCO style) learning curves with DETR and UP-DETR on VOC and COCO. Models are trained with the SwAV pre-training ResNet-50 for 150 and 300 epochs, and the learning rate is reduced at 100 and 200 epochs, respectively.

4.3. One-Shot Detection

Given a query image patch whose class label is not included in the training data, one-shot detection aims to detect all instances with the same class in a target image. One-shot detection is a promising research direction that can detect unseen instances. With feeding query patches to the decoder, UP-DETR is naturally compatible to one-shot detection task. Therefore, one-shot detection can also be treated as a downstream fine-tuning task of UP-DETR.

Following the same one-shot detection setting as [20], we crop the query image patch as the query patch to the DETR decoder. we train DETR and UP-DETR on VOC `2007trainval` and `2012trainval` sets with 300 epochs then evaluate on VOC `2007test` set. Table 3 shows the comparison to the state-of-the-art one-shot detection methods. Compared with DETR, UP-DETR significantly boosts the performance of DETR on both seen (**+22.8** AP^{50} gain) and unseen (**+15.8** AP^{50} gain) classes. Moreover, we show that UP-DETR outperforms all methods in both seen (**+7.9** AP^{50} gain) and unseen (**+4.0** AP^{50} gain)

Model	seen class																	unseen class				
	plant	sofa	tv	car	bottle	boat	chair	person	bus	train	horse	bike	dog	bird	mbike	table	**AP^{50}**	cow	sheep	cat	aero	**AP^{50}**
SiamFC [2]	3.2	22.8	5.0	16.7	0.5	8.1	1.2	4.2	22.2	22.6	35.4	14.2	25.8	11.7	19.7	27.8	15.1	6.8	2.28	31.6	12.4	13.3
SiamRPN [23]	1.9	15.7	4.5	12.8	1.0	1.1	6.1	8.7	7.9	6.9	17.4	17.8	20.5	7.2	18.5	5.1	9.6	15.9	15.7	21.7	3.5	14.2
CompNet [47]	28.4	41.5	65.0	66.4	37.1	49.8	16.2	31.7	69.7	73.1	75.6	71.6	61.4	52.3	63.4	39.8	52.7	75.3	60.0	47.9	25.3	52.1
CoAE [20]	30.0	54.9	64.1	66.7	40.1	54.1	14.7	60.9	77.5	78.3	77.9	73.2	80.5	70.8	**72.4**	46.2	60.1	**83.9**	67.1	75.6	46.2	68.2
Li *et al*. [24]	33.7	58.2	67.5	72.7	40.8	48.2	20.1	55.4	**78.2**	79.0	76.2	74.6	**81.3**	71.6	72.0	48.8	61.1	74.3	68.5	**81.0**	52.4	69.1
DETR	11.4	42.2	44.1	63.4	14.9	40.6	20.6	63.7	62.7	71.5	59.6	52.7	60.6	53.6	54.9	22.1	46.2	62.7	55.2	65.4	45.9	57.3
UP-DETR	**46.7**	**61.2**	**75.7**	**81.5**	**54.8**	**57.0**	**44.5**	**80.7**	74.5	**86.8**	**79.1**	**80.3**	80.6	**72.0**	70.9	**57.8**	**69.0**	80.9	**71.0**	80.4	**59.9**	**73.1**

Table 3: One-shot detection results on VOC `2007test` set.

Model	PQ	SQ	RQ	PQth	SQth	RQth	PQst	SQst	RQst	AP^{seg}
PanopticFPN++ [21]	42.4	79.3	51.6	49.2	82.4	58.8	32.3	74.8	40.6	**37.7**
UPSNet [42]	42.5	78.0	52.5	48.6	79.4	59.6	33.4	75.9	41.7	34.3
UPSNet-M [42]	43.0	79.1	52.8	48.9	79.7	59.7	34.1	78.2	42.3	34.3
DETR [5]	44.3	80.0	54.5	49.2	80.6	60.3	**37.0**	**79.1**	**45.9**	32.9
UP-DETR	**44.5**	**80.3**	**54.7**	**49.6**	**80.7**	**60.7**	36.9	78.9	45.8	34.0

Table 4: Panoptic segmentation results on the COCO `val` dataset with the same ResNet-50 backbone. The PanopticFPN++, UPSNet and DETR results are re-implemented by Carion *et al*. [5].

Case	Frozen CNN	Feature Reconstruction	AP$_{50}$
DETR	scratch transformers		74.5
(a)			74.0
(b)	✓		**78.7**
(c)		✓	62.0
(d)	✓	✓	**78.7**

Table 5: Ablation study on frozen CNN and feature reconstruction for pre-training models with AP$_{50}$. The experiments are fine-tuned on PASCAL VOC with 150 epochs.

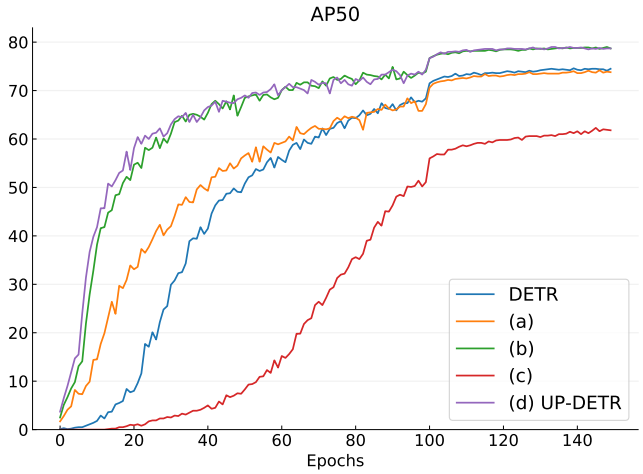


Figure 4: Learning curves of VOC (AP$_{50}$) with four different pre-training UP-DETR models and DETR. The models trained with 150 epochs corresponds to the models in Table 5 one-to-one.

classes of one-hot detection. It further verifies the effectiveness of our pre-training pretext task.

4.4. Panoptic Segmentation

Panoptic segmentation [21] is a natural extension to DETR by adding a mask head on the top of the decoder outputs. Following the same panoptic segmentation training schema as DETR [5], we fine-tune UP-DETR for box only annotations with 300 epochs. Then, we freeze all the weights of DETR and train the mask head for 25 epochs.

Table 4 shows the comparison to state-of-the-art methods on panoptic segmentation with the ResNet-50 backbone. As seen, UP-DETR outperforms DETR[5] with **+0.2** PQ, **+0.4** PQth and **+1.1** AP^{seg}.

4.5. Ablations

For ablation experiments, we pre-train UP-DETR for 15 epochs with the learning rate multiplied by 0.1 at the 10-th epoch on ImageNet. We fine-tune models on VOC object detection following the setup in Section 4.1 with 150 epochs[6].

4.5.1 Frozen CNN and Feature Reconstruction

To illustrate the importance of patch feature reconstruction and frozen CNN backbone of UP-DETR, we pre-train four different UP-DETR models with different combinations of whether freezing CNN and whether adding feature reconstruction.

[5]With a bug fixed in `github.com/facebookresearch/detr/issues/247`, the DETR baseline is better than paper report.

[6]More ablations are included in the appendix.

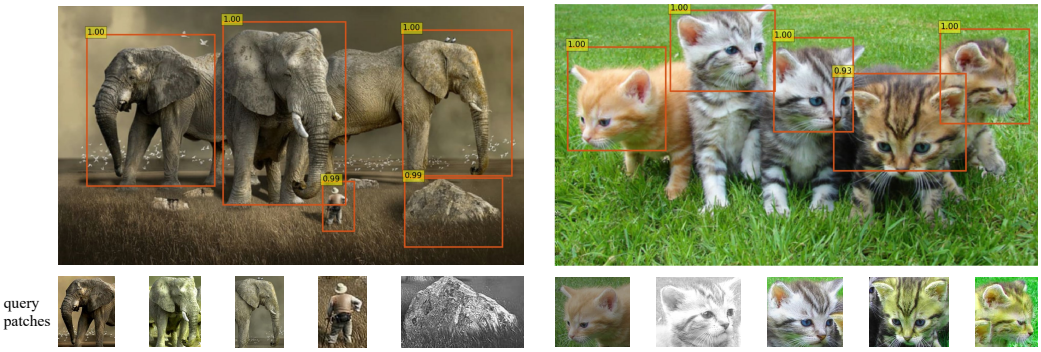


Figure 5: The unsupervised localization of patch queries with UP-DETR. The first line is the original image with predicted bounding boxes. The second line is query patches cropped from the original image with data augmentation. The value in the upper left corner of the bounding box is the model confidence.

Table 5 shows AP and AP_{50} of four different pre-training models and DETR on VOC with 150 epochs. As shown in Table 5, not all pre-trained models are better than DETR, and pre-training models (b) and (d) perform better than the others. More importantly, without frozen CNN, pre-training models (a) and (c) even perform worse than DETR. It confirms that freezing pre-trained CNN is essential to pre-train transformers. In addition, it further confirms the pretext (random query patch detection) may weaken the feature discrimination without the freezing pre-training CNN weights.

Fig. 4 plots the AP_{50} learning curves of four different pre-training models and DETR, where the models in Fig. 4 correspond to the models in Table 5 one-to-one. As shown in Fig. 4, model (d) UP-DETR achieves faster convergence at the early training stage with feature reconstruction. The experiments suggest that random query patch detection is complementary to the contrastive learning for a better visual representation. The former is designed for the spatial localization with position embeddings, and the latter is designed for instance or cluster classification.

It is worth noting that UP-DETR with frozen CNN and feature reconstruction heavily relies on a pre-trained CNN model, *e.g*. SwAV. Therefore, we believe that it is a promising direction for further investigating UP-DETR with random query patch detection and contrastive learning together to pre-train the whole DETR model from scratch.

4.6. Visualization

To further illustrate the ability of the pre-training model, we visualize the unsupervised localization results of given patch queries. Specifically, for the given image, we manually crop several object patches and apply the data augmentation to them. Then, we feed these patches as queries to the model. Finally, we visualize the model output with bounding boxes, whose classification confidence is greater than 0.9. This procedure can be treated as *unsupervised one-shot detection* or deep learning based *template matching*.

As shown in Fig. 5, pre-trained with random query patch detection, UP-DETR successfully learns to locate the bounding box of given query patches and suppress the duplicated bounding boxes [7]. It suggests that UP-DETR with random query patch detection is effective to learns the ability of object localization.

5. Conclusion

We present a novel pretext task called random query patch detection to pre-train the transformers in DETR. With unsupervised pre-training, UP-DETR significantly outperforms DETR on object detection, one-shot detection and panoptic segmentation. We find that, even on the COCO with sufficient training data, UP-DETR still performs better than DETR.

From the perspective of unsupervised pre-training models, pre-training CNN backbone and pre-training transformers are separated now. Recent studies of unsupervised pre-training mainly focus on feature discrimination with contrastive learning instead of specialized modules for spatial localization. However, for UP-DETR pre-training, the pretext task is mainly designed for patch localization by positional encodings and learn-able object queries. We hope that an advanced method can integrate CNN and transformers pre-training into a unified end-to-end framework and apply our pre-training tasks to more detection related frameworks.

Acknowledgement

This work was supported by the Guangdong Natural Science Foundation under Grant 2019A1515012152.

[7] Base picture credit: https://www.piqsels.com/en/public-domain-photo-jrkkq, https://www.piqsels.com/en/public-domain-photo-smdfn.

References

[1] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations*, 2019. 2

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 7

[3] 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

[4] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020. 1, 2, 3, 5, 6, 7

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149, 2018. 2

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33, 2020. 1, 2, 5, 6

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 5

[9] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009. 4

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 1, 2

[12] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In *Advances in Neural Information Processing Systems*, pages 13063–13075, 2019. 4

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 1, 2, 4

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006. 2

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020. 1, 2

[17] Kaiming He, Ross Girshick, and Piotr Dollár. Rethinking imagenet pre-training. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4918–4927, 2019. 5

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 6

[19] 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. 1

[20] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and Tyng-Luh Liu. One-shot object detection with co-attention and co-excitation. *arXiv preprint arXiv:1911.12529*, 2019. 6, 7

[21] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 7

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750, 2018. 2

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8971–8980, 2018. 7

[24] Xiang Li, Lin Zhang, Yau Pun Chen, Yu-Wing Tai, and Chi-Keung Tang. One-shot object detection without fine-tuning. *arXiv preprint arXiv:2005.03819*, 2020. 7

[25] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2359–2367, 2017. 5

[26] 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. 5, 6

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017. 2, 6

[28] 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 *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 1, 2, 4

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016. 2

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 5

[31] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7363–7372, 2019. 6

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018. 1, 2

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 1, 2

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1, 2

[35] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11563–11572, 2020. 2, 4

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 4

[37] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016. 2, 3

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9627–9636, 2019. 2, 6

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1, 2

[40] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10186–10195, 2020. 2, 4, 6

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3733–3742, 2018. 2

[42] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8818–8826, 2019. 7

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, pages 5753–5763, 2019. 2

[44] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6210–6219, 2019. 2

[45] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 421–430, 2019. 2, 4

[46] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 1, 2

[47] Tengfei Zhang, Yue Zhang, Xian Sun, Hao Sun, Menglong Yan, Xue Yang, and Kun Fu. Comparison network for one-shot conditional object detection. *arXiv e-prints*, pages arXiv–1904, 2019. 7

[48] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2

Appendix

A. More ablations

A.1. Single-Query Patch *vs*. Multi-Query Patches

We pre-train two UP-DETR models with single-query patch ($M = 1$) and multi-query patches ($M = 10$). The other hyper-parameters are set as mentioned in the paper.

Table 6 shows the results of single-query patch and multi-query patches. Compared with DETR, UP-DETR surpasses it in all AP metrics by a large margin no matter with single-query patch or multi-query patches. When pre-training UP-DETR with the different number of query patches, UP-DETR ($M = 10$) performs better than UP-DETR ($M = 1$) on the fine-tuning task, although there are about 2.3 instances per image on VOC. Therefore, we adopt the same UP-DETR with $M = 10$ for both VOC and COCO instead of varying M for different downstream tasks.

Model	AP	AP_{50}	AP_{75}
DETR	49.9	74.5	53.1
UP-DETR (M=1)	53.1 (+3.2)	77.2 (+2.7)	57.4
UP-DETR (M=10)	**54.9** (+5.0)	**78.7** (+4.2)	**59.1**

Table 6: The ablation results of pre-training models with single-query patch and multi-query patches on PASCAL VOC. The values in the brackets are the gaps compared to the DETR with the same training schedule.

A.2. Attention Mask

After downstream task fine-tuning, we find that there is no noticeable difference between the UP-DETR pre-trained w/ and w/o attention mask. So, we plot the loss curves in the pretext task to illustrate the effectiveness of attention mask.

As shown in Fig. 6, at the early training stage, UP-DETR without attention mask has a lower loss. However, as the model converging, UP-DETR with attention mask overtakes it with a lower loss. It is reasonable because the loss is calculated by the optimal bipartite matching. During the early training stage, the model is not converged, and the model without attention mask takes more object queries into attention. Intuitively, the model is easier to be optimized due to introducing more object queries. However, there is a mismatching between the query patch and the ground truth for the model without attention mask. As the model converging, the attention mask gradually takes effect, which masks the unrelated query patches and leads to a lower loss.

A.3. Object Query Shuffle

Without object query shuffle, the groups of object queries are assigned fixedly during the pre-training. However, for the downstream object detection tasks, there is no explicit group assignment between object queries. So, we design the object query shuffle to simulate implicit grouping between object queries.

The motivation of object query shuffle is clear, however, we find that object query shuffle is not helpful. In the pre-training and fine-tuning phase, the model w/o object query shuffle converges faster. Fig. 7 shows the fine-tuning result of COCO w/ and w/o object query shuffle. As seen, without object query shuffle, the model converges faster and achieves **43.1** AP (higher than 42.8 AP with object query shuffle pre-training). The result indicates that fixed group is beneficial for training object queries. Shuffle may disturb the spatial preference learning. Therefore, in our open-source code (https://github.com/dddzg/up-detr), we upload the pre-training model without object query shuffle.

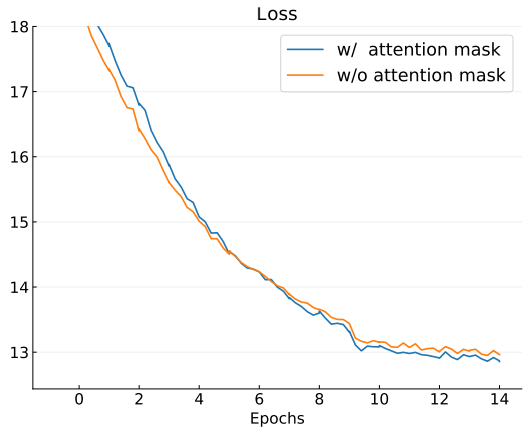


Figure 6: The loss curves of pre-training procedure for UP-DETR w/ and w/o the attention mask.

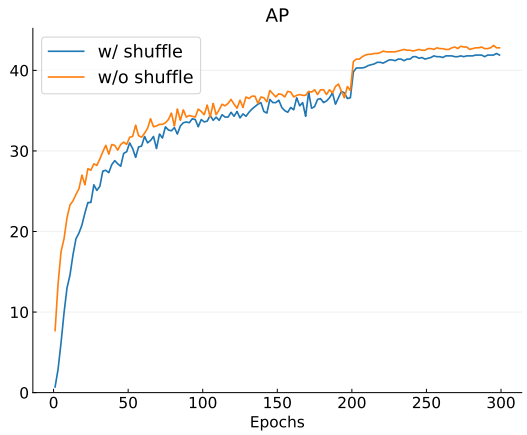


Figure 7: The AP curves of COCO fine-tuning procedure for UP-DETR w/ and w/o the object query shuffle. The learning rate is reduced at 200 epochs.

UP-DETR：Transformer物体检测的无监督预训练

戴志刚[1,2,3**]，博伦蔡家民[2]，玉耿林[2]，陈君英[1,3]
[1]华南理工大学软件工程学院
[2]腾讯微信信息顾问公司
[3]大数据与智能机器人重点实验室
（华南理工大学），教育部
zhigangdai@hotmail.com, {arlencai,lincolnlin}@tencent.com,
jychense@scut.edu.cn

摘要

采用Transformer的目标检测（DETR）通过Transformer编码器-解码器结构达到与Faster R-CNN有相同竞争力的性能。受预训练Transformer在自然语言处理中巨大成功的启发，我们提出了一个名为随机查询补丁检测的掩饰任务，以无监督的预训练DETR（UP-DETR）进行对象检测。具体地说，我们从给定的图像中随机裁剪补丁，然后将它们作为查询发送给解码器。该模型经过预先训练，可以从原始图像中检测这些查询补丁。在预训练过程中，我们解决了两个关键问题：多任务学习和多查询定位。（1）为了权衡掩饰任务中的分类和定位偏好，我们冻结了CNN主干，提出了一个与补丁检测联合优化的补丁特征重构分支。（2）为了实现多查询定位，我们引入了单查询补丁的UP-DETR，并将其扩展到具有对象查询洗牌和注意掩码的多查询补丁。在我们的实验中，UP-DETR显著提高了DETR的性能，在目标检测，一次性检测和泛光分割方面提高了DETR的平均精度。以下为代码和预训练模型：
https://github.com/dddzg/up-detr.Introduction

1. 介绍

采用Transformer的对象检测（DETR）[5]是最近的一个框架，它通过Transformer编码器-解码器将对象检测视为一个直接的预测问题[39]。如果没有手工设计的样本选择[46]和非极大值抑制，DETR与Faster R-CNN[34]达到了同等竞争力的性能。然而，DETR面临着训练和优化的挑战，这需要大规模的训练数据和一个极长的训练时间。

[1]

[1] 这项工作是戴志刚在腾讯微信实习时完成的

如图 1 和第 4.1 节所示，我们发现 DETR在PASCAL VOC[13]中表现不佳，该数据集的训练数据和实例都不如 COCO[28]。

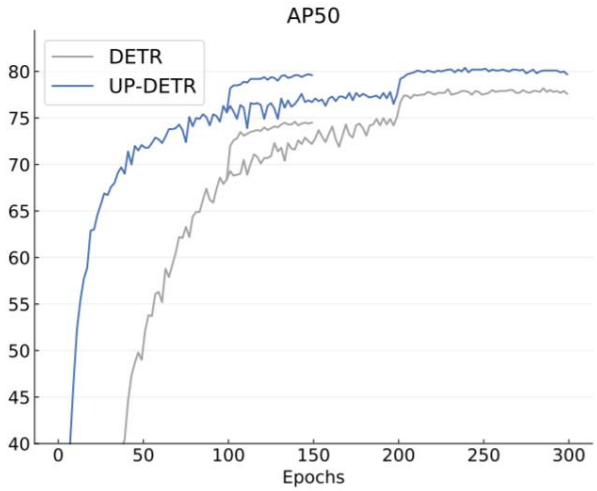


图 1：DETR和具有ResNet-50 主干网的UP-DETR在VOC数据集上的学习曲线（AP_{50}）。在这里，它们在trainval07+12上训练，在test2007 上检测。我们画出各自的长的和短的训练图，并且学习律在 100 到 200 代之间下降。

通过设计良好的掩饰任务，无监督的预训练模型在自然语言处理方面（如：GPT[32,33]和BERT[11]）和计算机视觉领域（如：MoCo[16,9]和SwAV[7]）都取得了显著的进展。在DETR中，CNN主干（具有约 23.2 兆参数的ResNet-50[19]）通过预训练来提取良好的视觉表示，但具有约 18.0 兆参数的Transformer模块没有进行预训练。更重要的是，尽管是无监督的视觉表示学习（例如：对比学习）在最近的研究中引起了广泛的关现有的掩饰任务不能直接应用于DETR的Transformer的预训练。

主要原因是DETR主要关注空间定位学习，而不是基于图像实例的[16,8,14]或基于集群的[4,6,1]对比学习。

受无监督预训练在自然语言处理[11]中的巨大成功的启发，我们的目标是在一个大尺度数据集（例如：ImageNet）上对无监督的Transformer进行无监督的预训练，并将对象检测视为下游任务。动机是直观的，但现有的掩饰任务似乎是不现实的预先训练Transformer。为了克服这个问题，我们提出了Unsupervised Pre-training DETR（UP-DETR）和一种新的无监督掩饰任务，名为随机查询块检测，预训练检测器没有任何人工注释一，我们从给定的图像中随机裁剪多个查询块，并预训练Transformer进行检测，以预测给定图像中这些查询块的边界框。在训练前的过程中，我们解决了以下两个关键问题：

（1） **多任务学习：**对象检测是对象分类和局部化的耦合。 为了避免查询补丁检测破坏分类特征，我们引入了冻结的训练前主干网和补丁特征重构，以保持Transformer的特征识别。

（2） **多查询本地化：**不同的输出对象查询主要关注不同的位置区域和方框大小。为了说明这一特性，我们提出了一个简单的单查询预训练，并将其扩展到多查询版本。针对多查询补丁，我们设计了对象查询洗牌和注意力掩码，以解决查询补丁和对象查询之间的分配问题。

在我们的实验中，UP-DETR 在 PASCAL VOC[13]和COCO[28]目标检测上的表现优于 DETR，具有更快的收敛速度和更好的平均精度。此外，UP-DETR 在一次性检测和泛光分割上也具有最先进的性能。在总结中，我们发现冻结训练前的 CNN 主干是训练前保持特征识别的最重要的过程。

2. 相关工作

2.1. 目标检测

大多数目标检测方法主要在阳性和负样本分配方面有所不同。两级探测器[34，3]和一级探测器[27，29]的一部分使用具有物物单位阈值和模型置信度的多尺度锚构造正负样本。无锚单级探测器通过目标中心网格分配正和负样本。张等人。[46]证明了它们之间的性能差距是由于正训练样本和负训练样本的选择。DETR[5]是一个最近的对象检测框架，在概念上更简单，无需通过直接设置预测[37]，它自动分配正和负样本。

除了正样本和负样本选择问题外，分类和定位之间的权衡也难以进行对象检测。张[45]证明了分类和定位之间存在域错位。吴和宋[35]设计两个头部结构进行分类和定位。他们指出，这两个任务可能有相反的偏好。对于我们的预训练模型，它保持了分类和本地化的共享特性。因此，必须在这两项任务之间进行一个很好的权衡。

2.2. 无监督预训练

无监督的预训练模型总是遵循两个步骤：对具有掩饰任务的大规模数据集进行预训练和对下游任务的参数进行微调。对于无监督的预训练，掩饰任务总是被发明出来的，我们感兴趣的是学习到的中间表示，而不是掩饰任务的最终表现。

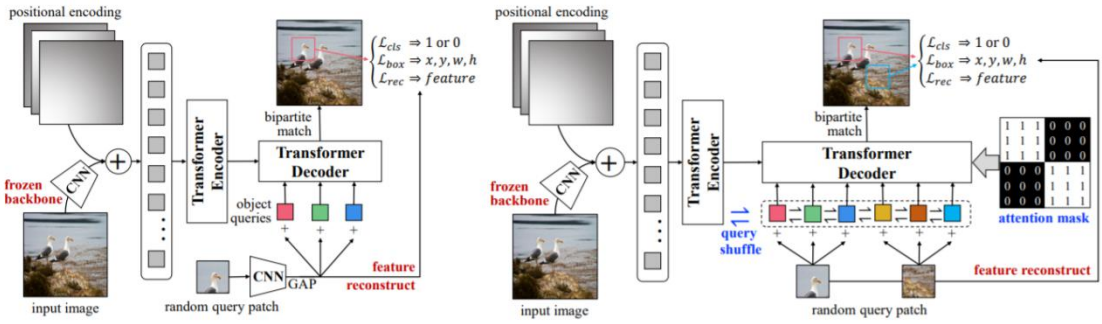
为了进行无监督的预训练，有各种设计良好的掩饰任务。对于自然语言处理，利用离散标记，掩蔽语言模型[11]，置换语言模型和自动回归模型[32,33]之间的时间序列关系，对预训练Transformer[32,33]进行语言表示。在计算机视觉方面，无监督预训练模型最近在视觉表示学习方面也取得了显著的进展，在下游任务中优于监督学习模型。基于实例的识别任务和基于聚类的识别任务是最近研究中两个典型的掩饰任务。基于实例的识别任务主要在于通过非参数对比学习来维护不同大小的负样本[16,14]。此外，实例识别也可以作为参数实例分类[4]来执行。基于群集的任务因脱机[6，1]或联机群集过程[7]而异。UP-DETR 是一种新的掩饰任务，旨在基于 DETR 结构对Transformer进行预训练以进行对象检测。

3. UP-DETR

提出的 UP-DETR 包含预训练和微调程序：（a）Transformer在大规模数据集上未经监督的预训练，没有任何人和符号；（b）整个模型使用下游任务上与原始DETR[5]相同的标

记数据进行微调。在本节中，我们主要介绍了如何用随机查询补丁检测来预训练Transformer编码器和解码器。如图 2 所示，随机查询补丁检测的主要思想简单而有效。

首先，利用冻结的 CNN 主干网用特征映射 $\mathbf{f} \in^{C \times H \times W}$ 提



(a) single-query patch (N=3, M=1)　　(b) multi-query patch (N=6, M=2)

图 2: 通过随机查询补丁检测实现上行 DETR 的预训练过程。（a）只有一个单一的查询补丁，我们添加到所有对象查询的补丁。（b）对于多查询补丁，我们将每个查询补丁添加到 N/M 带有对象查询洗牌和注意掩码的对象查询中。CNN 没有在（B）的解码器中绘制为整洁。

取视觉表示一个输入图像，其中 C 是通道尺寸，H×W是特征图大小。然后，用位置编码添加特征图，并传递到 DETR 中的多层Transformer编码器。对于随机裁剪的查询补丁，具有全局平均池（GAP）的 CNN 主干提取了补丁特征 $p \in R^C$，它变平并补充了对象查询 $q \in R^C$ 在将其传递到Transformer解码器之前。请注意，查询补丁指的是从原始图像中提取的裁剪补丁，但对象查询指的是输入解码器的位置嵌入。CNN参数在整个模型中共享。

在预训练过程中，解码器预测与输入图像中随机查询斑块的位置相对应的边界框。假设通过随机裁剪有 M 个查询补丁，该模型推断了一个对应于 N 个对象查询的预测固定集 $\hat{y} = \{\hat{y}_i\}_{i=1}^N$ 和N个物体相联系（N>M）。为了更好地理解，我们将在第 3.1 节中描述单查询补丁（M=1）的训练细节，并将其扩展到第 3.2 节中具有对象查询洗牌和注意掩码的多查询补丁（M>1）。

3.1. 单个查询修补程序

DETR 为每个对象查询[5]学习不同的空间专门化，这表明不同的对象对象查询专注于不同的位置区域和方框大小。当我们从图像中随机裁剪补丁时，对查询补丁的位置区域和方框大小没有任何先验。为了保留不同的空间专门化，我们明确地为所有对象查询（N=3）指定单查询补丁（M=1），如图 2（a）所示。

在预训练过程中，将补丁特征 p 添加到每个不同的对象查询 q 中，解码器生成 N 对预测 $\hat{y} = \{\hat{y}_i\}_{i=1}^N$，以检测输入图像中的查询补丁的边界框。在 DETR[5]之后，我们使用匈牙利算法[37]计算预测在 DETR[5]之后，我们使用匈牙利算法[37]计算预测 $\hat{y}_{\hat{\sigma}(i)}$ 与地面真值 y_i 之间相同的匹配代价，其中是由最优二部匹配计算的指数。

对于损失计算，预测结果 L_{cls}，由三个元素组成：$\hat{c}_i$ 是匹配每个对象查询的查询补丁的二进制分类；$\hat{b}_i$ 是定义框中心坐标，其宽度和高度{x，y，w，h}的向量。它们相对于图像大小被重新缩放；$\hat{p}_i$ 通常是ResNet-50 主干的 C=2048 的重建特征。根据上述定义，所有匹配对的匈牙利语损失均定义为：

$$\mathcal{L}(y,\hat{y}) = \sum_{i=1}^{N}[\lambda_{\{c_i\}}\mathcal{L}_{cls}(c_i,\hat{c}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}}\mathcal{L}_{box}(b_i,\hat{b}_{\hat{\sigma}(i)}) + \mathbb{1}_{\{c_i=1\}}\mathcal{L}_{rec}(p_i,\hat{p}_{\hat{\sigma}(i)})]. \quad (1)$$

此处，L_{cls} 是两个类的交叉熵损失（匹配查询补丁或不匹配），和类平衡权重 $\lambda_{\{c_i=1\}} = 1$ 和 $\lambda_{\{c_i=0\}} = M/N$。$L_{box}$ 是 l_1 损失和广义 LoU 损失的线性组合，具有与 DETR 相同的重量超参数[5]。L_{rec} 是本文提出的重建损失是为了平衡无监督预训练中的分类和定位，这将

在下面详细讨论。

3.1.1 修补程序功能重建

对象检测是对象分类和定位的耦合，其中这两个任务总是具有不同的特征偏好[45，40，35]。与 DETR 不同，我们提出了一个特征重建项 L_{rec} 在本地化预训练过程中保持分类特征。这个术语的动机是为了保留 CNN 在将特征传递给Transformer后的特征识别提取物。L_{rec} 是 CNN 主干提取的 l_2 归一化补丁特征之间的均方误差，定义如下：

$$\mathcal{L}_{rec}(p_i, \hat{p}_{\hat{\sigma}(i)}) = \left\| \frac{p_i}{\|p_i\|_2} - \frac{\hat{p}_{\hat{\sigma}(i)}}{\|\hat{p}_{\hat{\sigma}(i)}\|_2} \right\|_2^2. \qquad (2)$$

3.1.2 冻结预训练主干网络

通过补丁特征的重构，CNN 主干参数严重影响了模型的训练。我们的动机是，Transformer后的特性应该与 CNN 主干后的特征有相似的区别。因此，我们冻结了预训练主干，并用重建了Transformer后的补丁特征 L_{rec}。稳定的主干参数有利于Transformer的预训，并加速了特征的重建。

如上所述，我们提出并应用特征重构和冻结主干来保持特征识别以进行分类。在第 4.5.1 节中，我们将通过实验来分析和验证它们的必要性。

3.2. 多重查询修补程序

对于一般的对象检测，在每个图像中都有多个对象实（例如：在 COCO 数据集中的每张图像中，平均有 7.7 个对象实例）。此外，当单个对象查询N的数量较大时，单个查询补丁可能会导致收敛困难。因此，单查询补丁预训练与多对象检测任务不一致，对于典型对象查询设置N=100 不合理。然而，将单个查询补丁扩展到多查询补丁并不简单，因为 M 个查询补丁和 N 个对象查询之间的分配对于多查询补丁是一个特定的负采样问题。

为了解决这个问题，我们将 N 个对象查询划分为 M 个组，其中每个查询补丁都被分配给 N/M 对象查询。查询补丁程序将按顺序分配给对象查询。例如，将第一个查询补丁分配给第一个 N/M 对象查询，将第二个查询补丁分配给第二个 N/M对象查询，等等。在这里，我们假设它在预训练期间需要满足两个要求：

（1）查询补丁的独立性。所有的查询补丁都是从图像中随机裁剪出来的。因此，它们是独立的，没有任何关系。例如，第一次裁剪的边界框回归与第二次裁剪无关。

（2）对象查询的多样性。在下游任务的对象查询之间没有显式的组分配。换句话说，查询补丁可以添加到任意 N/M 对象查询中。

3.2.1 注意力掩模

为了满足查询补丁的独立性，我们使用一个注意力掩码矩阵来控制不同对象查询之间的交互。掩模矩阵 $X \in R^{N \times N}$ 是否被添加到解码器 $soft\max(QK^T / \sqrt{d_k} + X)V$。类似于令牌掩模在UniLM[12]中，注意力掩模定义为：

$$\mathbf{X}_{i,j} = \begin{cases} 0, & \text{i, j in the same group} \\ -\infty, & \text{otherwise} \end{cases}, \qquad (3)$$

$\mathbf{X}_{i,j}$ 在同一组中决定其中确定对象查询 q_i 是否处理与对象查询 q_j 的交互。为了直观的理解，图中的注意力掩模在图片 2b中分别显示的 1 表示 0,0 表示 $-\infty$。

3.2.2 对象查询洗牌

对象查询组是人工分配的。但是，在下游对象检测任务期间，对象查询之间没有显式的组分配。因此，为了模拟对象查询之间的隐式组分配，我们在预训练期间随机洗牌所有对象查询嵌入的排列。

如 2b图中所示，说明了具有注意力掩码和对象查询洗牌的多查询补丁的预训练。为了提高泛化性，我们在预训练过程中将 10%的查询补丁随机掩码为零类似于退出[36]。在我们的实验中，两个典型的值被设置为 N=100 和 M=10。除这些修改外，其他训练设置与第 3.1 节中所述的训练设置相同。

4. 实验结果

我们使用 ImageNet[10]对 UP-DETR 进行预训练，并对在VOC[13]和 COCO[28]对象检测训练参数，一次性

检测和泛光分割。在所有的实验中，我们都采用了具有 ResNet-50 主干的 UP-DETR 模型（41.3M 参数），6 个Transformer编码器，6 个宽度为 256 的有 8 个注意力头的解码器层。关于 DETR 的开源，我们在提出的UP-DETR 中使用相同的超参数以及我们的数据的重新实现。我们注释了 R50 和 R101，作为ResNet-50 和 ResNet-101 的缩写。

预训练前的设置。UP-DETR 在 ImageNet 训练集上进行预先训练，没有任何标签。CNN 主干（ResNet-50）采用 SWAV[7]预训练。由于来自 ImageNet 的输入图像相对较小，我们调整其大小，使最短的边在[320,480]像素内，而最长的边最多在 600 像素内。给定图像，我们裁剪具有随机坐标，高度和宽度的查询斑块，将其大小调整为 128x128 像素，并使用 SimCLR-style[8]进行转换，而不进行水平翻转。AdamW[30]用于优化 UP-DETR，初始学习率为 1×10^{-4} 而重量衰减为 1×10^{-4}。我们在 8V100GPU 上的 60 个时期使用 256 大小的小批量样本，学 40 代和每代学习速率乘以 0.1。

微调的设置。该模型通过预训练 UP-DETR 参数进行了初始化，并对 VOC 和 COCO 上的所有参数（包括 CNN）进行了微调。我们用初始学习率为 1×10^{-4} 来调整这个微调的模型用 4 于Transformer和 5×10^{-5} 对于 CNN 主干，其他设置与 8V100GPU 上的 DETR[5]相同。该模型为 150/300 代，在 100/200 代学习率分别乘以 0.1。

4.1. PASCAL VOC目标检测

Model/Epoch	AP	AP_{50}	AP_{75}
Faster R-CNN	56.1	**82.6**	**62.7**
DETR/150	49.9	74.5	53.1
UP-DETR/150	**56.1** (+6.2)	79.7 (+5.2)	60.6 (+7.5)
DETR/300	54.1	78.0	58.3
UP-DETR/300	**57.2** (+3.1)	80.1 (+2.1)	62.0 (+3.7)

表 1：帕斯卡 VOC 培训 07+12 培训的对象检测结果，以及 2007 年测试的评估。DETR 和 UP-DETR 使用 R50 主干，更快的 R-CNN使用 R50-C4 主干。括号中的值是与具有相同培训计划的 DETR之间的差距。

开始。该模型在VOC trainval07+12（约 16.5k 图像）上进行了微调，并在test2007 中进行了评估。我们报告了COCO风格的度量：AP，AP50（默认的VOC 度量）和 AP75。为了进行完整的比较，我们报告了更快的 R-CNN 与 R50-C4 主干[7]的结果，它的性能比R50[25]要好得多。使用 R50-C4 的 DETR比 R50 显著增加了计算成本，因此我们用 R50 主干来微调 UP-DETR。

测试结果。表 1 显示了PASCAL VOC的比较结果。我们发现 DETR 在PASCAL VOC中表现不佳，比Faster R-CNN差得多，在所有指标上都有很大的差距。UP-DETR 显著提高了 DETR 的性能：分别在 150（300）时代的+6.2（+3.1）AP，+5.2（+2.1）AP50 和 +7.5（+3.7）AP75。此外，UP-DETR（R50）取得了与使用更好的 AP 的更快的 R-CNN（R50-C4）类似的结果。我们发现DETR的 AP_{50} 和 AP_{75} 都比Faster R-CNN要差一些。它可能来自不同比例的特征图（C4 用于 Faster R-CNN），而没有 NMS 后处理（NMS 降低了 AP，但稍微提高了 AP_{50}）。

图 3a显示了VOC上的 AP（COCO 风格）学习曲线。上升显著加快了模型的收敛速度。在学习率降低后，UP-DETR显著提高了DETR的性能。注意到UP-DETR 在 150 代后获得 56.1AP，然而，它对应的 DETR（刮伤 Transformer）即使在 300 代后也只获得 54.1AP，甚至不能赶上更长的训练。这表明，由于训练数据不足，预训练Transformer是不可缺少的。（约 16.5K个VOC上的图像）

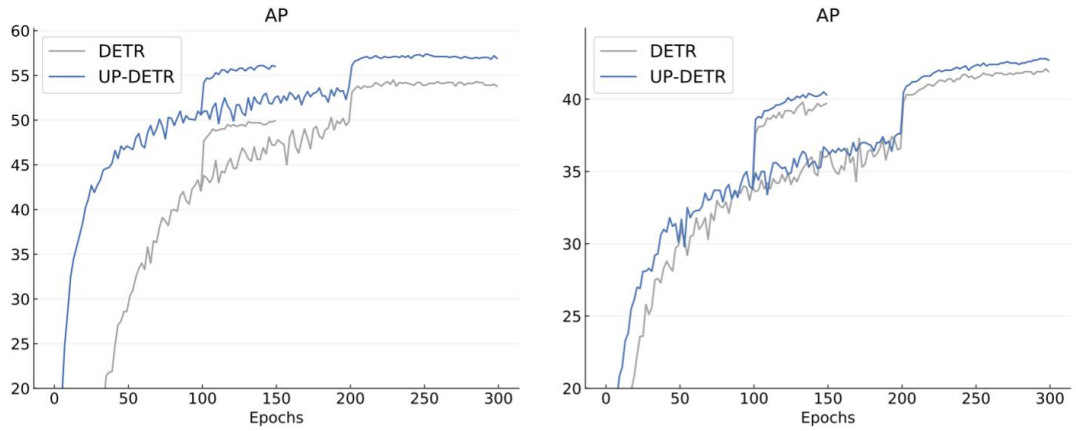
4.2. COCO对象检测

安装程序。该模型在COCO train2017（约 118k张图像）上进行了微调，并在val2017 进行了评估。在COCO数据集中有许多小对象，其中 DETR 表现不佳[5]。因此，我们报告 AP，AP50，AP75，APs，APm 和 APl 进行综合比较。此外，我们还报告了高度优化的Faster R-CNN-FPN 的结果，具有短（3×）和长（9×）训练时间表，这些结果可以提高性能结果[17]。

测试结果。表 2 显示了 COCO 与其他方法的结果。在 150 代中，UP-DETR的AP指标比DETR好 0.8，与更快的 R-CNN-FPN（3×计划）相比，它取得了类似的性能。通过 300 代训练，UP-DETR 在COCO上获得 42.8AP，比 DETR（SwAVCNN）好 0.7AP，比Faster R-CNN-FPN（9× 计划）好 0.8AP。总的来说，UP-DETR在检测中小，大物体方面全面优于 DETR。遗憾的是，由于缺乏

Model	Backbone	Epochs	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN † [26]	R101-FPN	-	36.2	59.1	39.0	18.2	39.0	48.2
Mask R-CNN † [18]	R101-FPN	-	38.2	60.3	41.7	20.1	41.1	50.2
Grid R-CNN † [31]	R101-FPN	-	41.5	60.9	44.5	23.3	44.9	53.1
Double-head R-CNN [40]	R101-FPN	-	41.9	62.4	45.9	23.9	45.2	55.8
RetinaNet † [27]	R101-FPN	-	39.1	59.1	42.3	21.8	42.7	50.2
FCOS † [38]	R101-FPN	-	41.5	60.7	45.0	24.4	44.8	51.6
DETR [5]	R50	500	42.0	62.4	44.2	20.5	45.8	61.1
Faster R-CNN	R50-FPN	3×	40.2	**61.0**	**43.8**	**24.2**	43.5	52.0
DETR (Supervised CNN)	R50	150	39.5	60.3	41.4	17.5	43.0	59.1
DETR (SwAV CNN) [7]	R50	150	39.7	60.3	41.7	18.5	43.8	57.5
UP-DETR	R50	150	**40.5** (+0.8)	60.8	42.6	19.0	**44.4**	**60.0**
Faster R-CNN	R50-FPN	9×	42.0	62.1	**45.5**	**26.6**	45.4	53.4
DETR (Supervised CNN)	R50	300	40.8	61.2	42.9	20.1	44.5	60.3
DETR (SwAV CNN) [7]	R50	300	42.1	**63.1**	44.5	19.7	46.3	60.9
UP-DETR	R50	300	**42.8** (+0.7)	63.0	45.3	20.8	**47.1**	**61.7**

表 2：2017 年 COCO 列车训练的目标检测结果，并于 2017 年进行评估。更快的 R-CNN，DETR 和 UP-DETR 都是在类似的设置下执行的。f 对于在 COCO 测试开发中评估的值，它们总是略高于 2017 年的 val。括号中的值是与 DETR 相比的间隙。



(a) AP learning curves of VOC　　(b) AP learning curves of COCO

图 3：在 VOC 和 COCO 上，具有 DETR 和 UP-DETR 的 AP（COCO 样式）学习曲线。模型用 SwAV 预训练的 ResNet-50 训练 150 和 300 时代，学习率分别在 100 和 200 时代降低。

FPN类似的体系结构[26]和高成本的注意操作，上升点在APs中仍然略微落后于更快的 R-CNN。

图 3b 显示了COCO上的AP学习曲线。UP-DETR在 150 和 300 代调度上都优于 DETR，收敛速度更快。在降低学习率之前，性能的提高更为明显。在降低学习率后，UP-DETR仍然领先 DETR约 0.7AP。这表明，即使在足够的训练数据下，预训练Transformer仍然是必不可少的（也就是：约 118K 的COCO数据集图片）。

4.3. 照一次的检测

给定一个类标签不包含在训练数据中的查询图像补丁，一次性检测的目的是检测目标图像中具有相同类的所有实例。一次性检测是一个很有前途的研究方向，它可以检测到看不见的实例。通过将查询补丁提供给解码器，UP-DETR 自然可以兼容一次性检测任务。因此，一次性检测也可以作为UP-DETR 的下游微调任务来处理。

按照与[20]相同的一次性检测设置，我们将查询图像补丁作为 DETR 解码器的查询补丁进行裁剪。我们在 VOC 2007train val 和 2012train val 的 300 代上训练 DETR和UP-DETR，然后在VOC 2007test 集上进行评估。表 3 显示了与最先进的一次性检测方法的比较。与 DETR 相比，UP-DETR 显著提高了 DETR 的性能

Model	seen class																	unseen class				
	plant	sofa	tv	car	bottle	boat	chair	person	bus	train	horse	bike	dog	bird	mbike	table	AP^{50}	cow	sheep	cat	aero	AP^{50}
SiamFC [2]	3.2	22.8	5.0	16.7	0.5	8.1	1.2	4.2	22.2	22.6	35.4	14.2	25.8	11.7	19.7	27.8	15.1	6.8	2.28	31.6	12.4	13.3
SiamRPN [23]	1.9	15.7	4.5	12.8	1.0	1.1	6.1	8.7	7.9	6.9	17.4	17.8	20.5	7.2	18.5	5.1	9.6	15.9	15.7	21.7	3.5	14.2
CompNet [47]	28.4	41.5	65.0	66.4	37.1	49.8	16.2	31.7	69.7	73.1	75.6	71.6	61.4	52.3	63.4	39.8	52.7	75.3	60.0	47.9	25.3	52.1
CoAE [20]	30.0	54.9	64.1	66.7	40.1	54.1	14.7	60.9	77.5	78.3	77.9	73.2	80.5	70.8	**72.4**	46.2	60.1	**83.9**	67.1	75.6	46.2	68.2
Li *et al.* [24]	33.7	58.2	67.5	72.7	40.8	48.2	20.1	55.4	**78.2**	79.0	76.2	74.6	**81.3**	71.6	72.0	48.8	61.1	74.3	68.5	**81.0**	52.4	69.1
DETR	11.4	42.2	44.1	63.4	14.9	40.6	20.6	63.7	62.7	71.5	59.6	52.7	60.6	53.6	54.9	22.1	46.2	62.7	55.2	65.4	45.9	57.3
UP-DETR	**46.7**	**61.2**	**75.7**	**81.5**	**54.8**	**57.0**	**44.5**	**80.7**	74.5	**86.8**	**79.1**	**80.3**	80.6	**72.0**	70.9	**57.8**	**69.0**	80.9	**71.0**	80.4	**59.9**	**73.1**

表 3：对VOC 2007test集的一次性检测结果

Model	PQ	SQ	RQ	PQ^{th}	SQ^{th}	RQ^{th}	PQ^{st}	SQ^{st}	RQ^{st}	AP^{seg}
PanopticFPN++ [21]	42.4	79.3	51.6	49.2	82.4	58.8	32.3	74.8	40.6	**37.7**
UPSNet [42]	42.5	78.0	52.5	48.6	79.4	59.6	33.4	75.9	41.7	34.3
UPSNet-M [42]	43.0	79.1	52.8	48.9	79.7	59.7	34.1	78.2	42.3	34.3
DETR [5]	44.3	80.0	54.5	49.2	80.6	60.3	**37.0**	**79.1**	**45.9**	32.9
UP-DETR	**44.5**	**80.3**	**54.7**	**49.6**	**80.7**	**60.7**	36.9	78.9	45.8	34.0

表 4：在具有相同 ResNet-50 主干的 COCOval 数据集上的全光分割结果。Carion 等人重新实现了FPN++，UPSNet 和 DETR 结果。[5]

Case	Frozen CNN	Feature Reconstruction	AP_{50}
DETR	scratch transformers		74.5
(a)			74.0
(b)	✓		**78.7**
(c)		✓	62.0
(d)	✓	✓	**78.7**

表 5：冷冻 CNN 的消融研究和 AP_{50} 预训练模型的特征重建。该实验在PASCAL VOC上微调了 150 代。

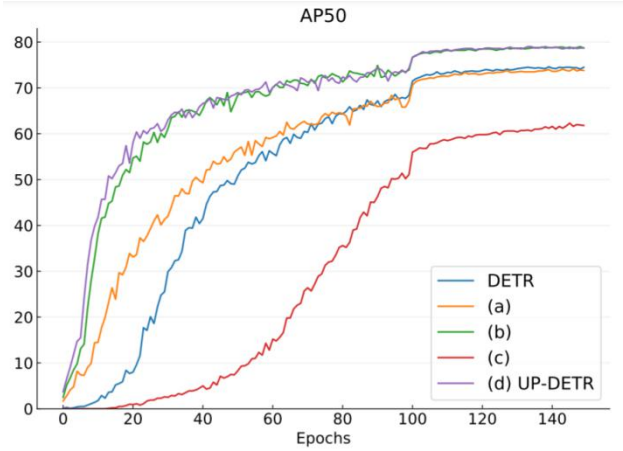


图 4：VOC 的学习曲线（AP5q）有四种不同的预训练高级数据模型和DETR。150 代训练的模型对应表 5 中的模型

（+22.8 AP_{50} 增益）和看不见的（+15.8 AP_{50} 增益）类。此外，我们还证明了 UP-DETR 优于所有所看到的方法（+7.9 AP_{50} 增益）和看不见的（+4.0 AP_{50} 增益）一次检
测的类。它进一步验证了我们的训练前掩饰任务的有效性。

4.4. 全视光分割

泛光分割[21]是通过在解码器输出的顶部添加一个掩码头的自然扩展。按照与DETR[5]相同的泛光分割训练模式，我们只针对 300 代的框注释微调UP-DETR。然后，我们冻结所有 DETR 的重量，并训练掩码头部 25 代。

表 4 显示了与最先进的ResNet-50 主干泛光分割方法的比较。如上所述，UP-DETR 在 +0.2PQ，+0.4PQ^{th} +1.1AP^{seg} 上的性能优于DETR。

4.5. 销蚀

在消融实验中，我们在ImageNet上预训练了 15 次，在第 10 阶段的学习率乘以 0.1。我们微调了VOC目标检测上的模型，在第 4.1 节设置的对象检测有 150 代。

4.5.1 冻结 CNN 和功能重建

为了说明 UP-DETR 的补丁特征重构和冻结 CNN 主干的重要性，我们对四种不同的冻结 CNN-DETR 模型，对是否冻结 CNN和是否添加特征重构进行了不同组合。

表 5 显示了四种不同预训练模型的AP和 AP_{50} 以及 150 代的VOC上的DETR。如表 5 所示，并非所有的预训练模型都优于DETR，而且预训练模型（b）和（d）都优于其他模型。更重要的是，如果没有冻结的CNN，训练

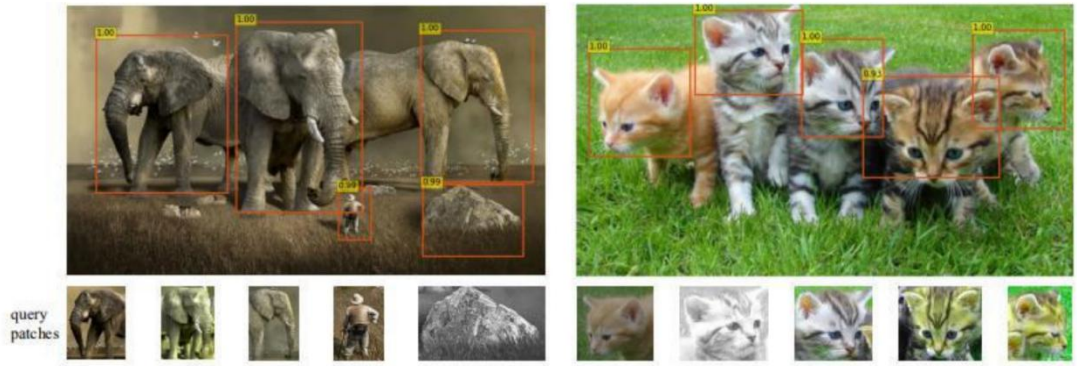


图 5：使用 UP-DETR 进行补丁程序查询的无监督定位。第一行是具有预测边界框的原始图像。第二行是从原始图像中裁剪出来的查询补丁。边界框左上角的值是模型的置信度。

前模型（a）和（c）的表现甚至比 DETR 更差。
它证实了冷冻预先训练的 CNN 对Transformer的训练模式至关重要。此外，它进一步证实了借口（随机查询 CNN 不冻结训练前 CNN 权值）可以削弱特征识别。

图 4 绘制了四种不同的训练前模型和DETR的 AP_{50} 学习曲线，图 4 对应于表 5 中的模型。如图 4 模型（d）UP-DETR 在早期训练阶段通过特征重构取得了更快的收敛速度。实验表明，随机查询补丁检测是对对比学习的补充，以获得更好的视觉表示。前者是用于位置嵌入的空间定位，后者是例如或集群分类。
值得注意的是，冻结CNN和特征重建很大程度上依赖于预先训练的CNN模型，例如，SwAV。因此，我们认为，进一步利用随机查询块检测和对比学习来研究整个DETR预训练整个DETR模型是一个很有前途的方向。

4.6. 可视化

为了进一步说明预训练模型的能力，我们可视化了给定补丁查询的无监督定位结果。具体地说，对于给定的图像，我们手动裁剪几个对象块并将数据增强应用到它们上。然后，我们将这些补丁作为查询提供给模型。最后，我们用其分类置信度大于 0.9 的边界框来可视化模型输出。该过程可以被视为无监督的一次性检测或基于深度学习的模板匹配。
如图 5，经过随机查询补丁检测的预训练，UP-DETR 成功学习定位给定查询补丁的边界框，抑制重复的边界框。研究表明，具有随机查询块检测的上行数据可以有效地学习对象定位的能力。

5. 结论

我们提出了一种新的掩饰任务，称为随机查询补丁检测，以对DETR中的Transformer进行预训练。在无监督预训练下，UP-DETR在目标检测，一次性检测和泛光分割方面显著优于DETR。我们发现，即使在有足够训练数据的COCO上，UP-DETR仍然表现得比DETR更好。从无监督的训练前模型的角度来看，训练前的 CNN 主干和训练前的Transformer现在被分离出来。最近对无监督预训练的研究主要集中在对比学习的特征识别上，而不是用于空间定位的专门模块。然而，对于UP-DETR预训练，掩饰任务主要是通过位置编码和可学习对象查询进行补丁定位。我们希望一种先进的方法可以将 CNN 和Transformer的预训练集成到一个统一的端到端框架中，并将我们的预训练任务应用于更多的检测相关框架中。

致谢

这项工作由广东自然科学基金会获得 2019A1515012152 的资助。

参考

[1] YM Asano, C Rupprecht, and A Vedaldi. Self-labelling via simultaneous clustering and representation learning. In International Conference on Learning Representations, 2019.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, AndreaVedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In European conference on computer vision, pages 850–865. Springer, 2016.

[3] 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.

[4] Yue Cao, Zhenda Xie, Bin Liu, Yutong Lin, Zheng Zhang, and Han Hu. Parametric instance classification for unsupervised visual feature learning. Advances in Neural Information Processing Systems, 33, 2020.

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. Endto-end object detection with transformers. arXiv preprint arXiv:2005.12872, 2020.

[6] Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision （ECCV）, pages 132–149, 2018.

[7] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. Advances in Neural Information Processing Systems, 33, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. arXiv preprint arXiv:2002.05709, 2020.

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

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 248–255. Ieee, 2009.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805, 2018.

[12] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. Unified language model pre-training for natural language understanding and generation. In Advances in Neural Information Processing Systems, pages 13063–13075, 2019.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes （ voc ） challenge. International journal of computer vision, 88（2）:303–338, 2010.

[14] Jean-Bastien Grill, Florian Strub, Florent Altche, CorentinTallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. Advances in Neural Information Processing Systems, 33, 2020.

[15] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition （CVPR’06）, volume 2, pages 1735–1742. IEEE, 2006.

[16] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9729–9738, 2020.

[17] Kaiming He, Ross Girshick, and Piotr Dollar. Rethinking imagenet pre-training. In Proceedings of the IEEE International Conference on Computer Vision, pages 4918–4927, 2019.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, pages 2961–2969, 2017.

[19] 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.

[20] Ting-I Hsieh, Yi-Chen Lo, Hwann-Tzong Chen, and TyngLuh Liu. One-shot object detection with co-attention and co-excitation. arXiv preprint arXiv:1911.12529, 2019.

[21] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollar. Panoptic segmentation. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 9404–9413, 2019.

[22] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European conference on computer vision （ECCV）, pages 734–750, 2018.

[23] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 8971–8980, 2018.

[24] Xiang Li, Lin Zhang, Yau Pun Chen, Yu-Wing Tai, and ChiKeung Tang. One-shot object detection without fine-tuning. arXiv preprint arXiv:2005.03819, 2020.

[25] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 2359–2367, 2017.

[26] Tsung-Yi Lin, Piotr Dollar, 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.

[27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 2980–2988,

[28] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollar, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In European Conference on Computer Vision, pages 740–755. Springer, 2014.

[29] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European Conference on Computer Vision, pages 21–37. Springer, 2016.

[30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In International Conference on Learning Representations, 2018.

[31] Xin Lu, Buyu Li, Yuxin Yue, Quanquan Li, and Junjie Yan. Grid r-cnn. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 7363–7372, 2019.

[32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training, 2018.

[33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. OpenAI Blog, 1（8）:9, 2019.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in Neural Information Processing Systems, pages 91–99, 2015.

[35] Guanglu Song, Yu Liu, and Xiaogang Wang. Revisiting the sibling head in object detector. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 11563–11572, 2020.

[36] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research, 15（1）:1929–1958, 2014.

[37] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2325–2333, 2016.2017.

[38] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9627–9636, 2019.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Advances in Neural Information Processing Systems, pages 5998–6008, 2017.

[40] Yue Wu, Yinpeng Chen, Lu Yuan, Zicheng Liu, Lijuan Wang, Hongzhi Li, and Yun Fu. Rethinking classification and localization for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 10186–10195, 2020.

[41] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 3733–3742, 2018.

[42] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: Aunified panoptic segmentation network. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 8818–8826, 2019.

[43] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. Xlnet: Generalized autoregressive pretraining for language understanding. In Advances in Neural Information Processing Systems, pages 5753–5763, 2019.

[44] Mang Ye, Xu Zhang, Pong C Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6210–6219, 2019.

[45] Haichao Zhang and Jianyu Wang. Towards adversarially robust object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 421–430, 2019.

[46] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the gap between anchor-based and anchor-free detection via adaptive training sample selection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 9759–9768, 2020.

[47] Tengfei Zhang, Yue Zhang, Xian Sun, Hao Sun, Menglong Yan, Xue Yang, and Kun Fu. Comparison network for one-shot conditional object detection. arXiv e-prints, pages arXiv–1904, 2019.

[48] Xingyi Zhou, Dequan Wang, and Philipp Krahenbuhl. Objects as points. arXiv preprint arXiv:1904.07850, 2019.

附录

A.更多的消融方法

A.1.单个查询修补程序与多个查询修补程序

我们用单查询补丁（M=1）和多查询补丁（M=10）预训练了两个 UP-DETR 模型。本文对其他超参数设置了设置。

表 6 显示了单查询补丁和多查询补丁的结果。与DETR相比，无论使用单查询补丁或多查询补丁，UP-DETR 在所有AP 指标中都大大超过了它。当使用不同数量的查询补丁预训练 UP-DETR 时，UP-DETR（M=10）在微调任务上的性能优于UP-DETR（M=1），尽管VOC上每张图像大约有 2.3 个实例。因此，我们对VOC和COCO采用与M=10 相同的 UP-DETR，而不是改变不同下游任务的M。

Model	AP	AP_{50}	AP_{75}
DETR	49.9	74.5	53.1
UP-DETR (M=1)	53.1 (+3.2)	77.2 (+2.7)	57.4
UP-DETR (M=10)	**54.9** (+5.0)	**78.7** (+4.2)	**59.1**

表 6：帕斯卡 VOC 上单查询和多查询补丁的预训练模型的消融结果。括号中的值是与具有相同培训计划的 DETR 之间的差距。

A.2.注意掩模

在进行了下游任务微调后，我们发现预先训练的上行数据集和无注意掩模之间没有明显的差异。因此，我们在掩饰任务中绘制了损失曲线来说明注意掩模的有效性。

如图 6，在早期训练阶段，没有注意掩模的 UPDETR 损失较低。然而，随着模型的收敛，带有注意掩模的UP-DETR以较低的损失超过了它。这是合理的，因为损失是由最优二部匹配计算出来的。在早期训练阶段，模型不收敛，没有注意掩码的模型需要注意更多的对象查询。直观上说，由于引入了更多的对象查询，该模型更容易被优化。然而，在没有注意掩码的模型中，查询补丁和地面真相之间存在不匹配。随着模型的收敛，注意掩码逐渐生效，这屏蔽了不相关的查询斑块，导致了较低的损失。

A.3.对象查询洗牌

不需要对象查询洗牌，在预训练期间固定地分配对象查询组。然而，对于下游的对象检测任务，没有对象查询之间的显式组分配。因此，我们设计了对象查询

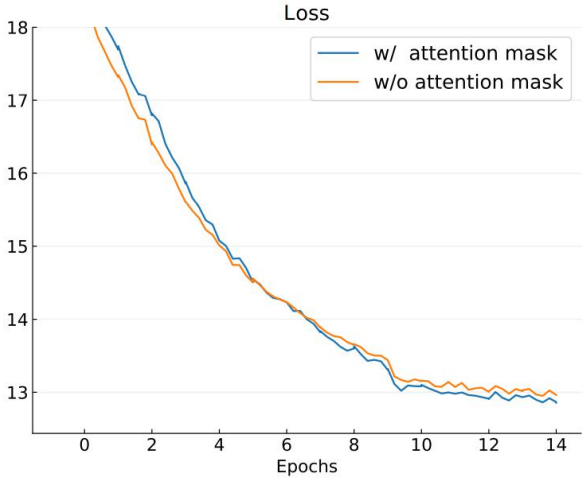


图 6：无注意掩模的UP-DETR训练前程序的损失曲线。

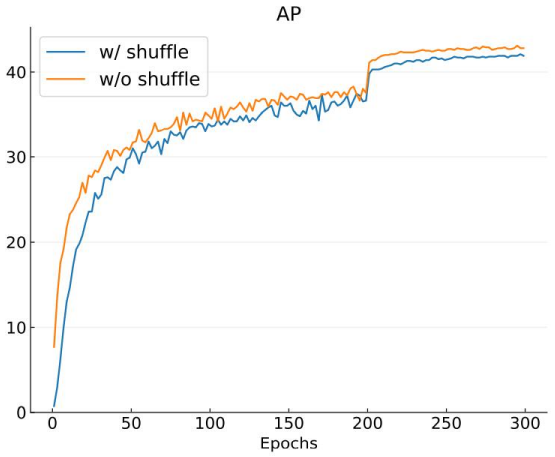


图 7：上下和无对象查询洗牌的COCO微调过程的AP曲线。在 200 代的学习率下降了。

洗牌来模拟对象查询之间的隐式分组。

对象查询洗牌的动机很清楚，但我们发现对象查询洗牌并没有帮助。在预训练和微调阶段，无对象查询洗牌的模型收敛速度更快。图 7 展示了COCO和COCO的微调结果。如上所述，没有对象查询洗牌，模型收敛速度更快，达到 43.1AP（对象查询洗牌预训练高于 42.8AP）。该结果表明，固定组有利于训练对象查询。洗牌可能会干扰空间偏好的学习。因此，在我们的开源代码（https://github.com/dddzg/up-detr）中，我们上传了训练前模型，而无需进行对象查询洗牌。