D\textsuperscript{3}ETR: Decoder Distillation for Detection Transformer

Xiaokang Chen\textsuperscript{1} \* Jiahui Chen\textsuperscript{2} \* Yan Liu\textsuperscript{3} \ Gang Zeng\textsuperscript{1}
School of Intelligence Science and Technology, Peking University\textsuperscript{1}
Beihang University\textsuperscript{2} \ Microsoft Research Asia\textsuperscript{3}

Abstract

While various knowledge distillation (KD) methods in CNN-based detectors show their effectiveness in improving small students, the baselines and recipes for DETR-based detectors are yet to be built. In this paper, we focus on the transformer decoder of DETR-based detectors and explore KD methods for them. The outputs of the transformer decoder lie in random order, which gives no direct correspondence between the predictions of the teacher and the student, thus posing a challenge for knowledge distillation. To this end, we propose MixMatcher to align the decoder outputs of DETR-based teachers and students, which mixes two teacher-student matching strategies, i.e., Adaptive Matching and Fixed Matching. Specifically, Adaptive Matching applies bipartite matching to adaptively match the outputs of the teacher and the student in each decoder layer, while Fixed Matching fixes the correspondence between the outputs of the teacher and the student with the same object queries, with the teacher’s fixed object queries fed to the decoder of the student as an auxiliary group. Based on MixMatcher, we build Decoder Distillation for Detection Transformer (D\textsuperscript{3}ETR), which distills knowledge in decoder predictions and attention maps from the teachers to students. D\textsuperscript{3}ETR shows superior performance on various DETR-based detectors with different backbones. For example, D\textsuperscript{3}ETR improves Conditional DETR-R50-C5 by 7.8/2.4 mAP under 12/50 epochs training settings with Conditional DETR-R101-C5 as the teacher.

1. Introduction

Knowledge distillation (KD) [14] is proposed to transfer knowledge from a large teacher model to a small student model, improving the student’s performance for free in model inference. Recently there has been steady progress in KD methods, with encouraging results in many vision tasks, including image classification [3, 7, 27, 31, 36, 40, 44, 45, 47] and object detection [2, 8, 11, 12, 37, 38, 42, 43]. Although various methods have been proposed, they mainly focus on CNN-based models [13, 28, 29] and are related to model structures, especially in object detection [2, 12, 37, 43]. Applying existing KD methods to novel detectors, such as DETR-based detectors [1, 6, 18, 22, 24, 48], poses challenges and may lead to trivial improvements. This paper aims to close the gap and explore KD methods for DETR-based detectors.

DETR [1] is an end-to-end detector with transformer layers [33]. DETR and its variants follow a pipeline of (i) extracting image features with a backbone, (ii) modeling global context with a transformer encoder, and (iii) making predictions for objects with a transformer decoder given the image features and object queries. To build KD baselines and recipes for DETR-based detectors, we go back to basics and investigate the influences of the above components. We find that the transformer decoder plays a critical role in maintaining good performance (Figure 3), thus taking our step to explore KD methods in the transformer decoder.

Different from CNN-based object detectors, the outputs of the DETR decoder lie in random orders\textsuperscript{1}, resulting in no direct correspondences between decoder outputs and ground-truth objects.

\textsuperscript{1}As DETR views object detection as a set prediction problem, resulting in no direct correspondences between decoder outputs and ground-truth objects.
direct correspondence between the outputs of the teacher and the student. To solve this problem, we propose MixMatcher to align the decoder outputs of the teacher and the student. MixMatcher mixes two matching strategies, Adaptive Matching and Fixed Matching. Adaptive matching determines teacher-student correspondence by computing the optimal bipartite matching [17] between predictions in each decoder layer of teacher and student models. To alleviate the instability issue of bipartite matching in teacher-student adaptive matching, we also propose fixed matching. We feed the teacher’s fixed object queries to the decoder of the student as an auxiliary group and apply fixed matching. It fixes the correspondence between the outputs of teacher and student models with the same object queries.

MixMatcher enables us to model the correspondence between the teacher and the student. Based on MixMatcher, we build decoder distillation for DETR-based methods (D3ETR). Instead of only considering the predictions, we also consider the attention modules (including self-attention and cross-attention) in decoder layers when performing distillation. For the attention modules, we distill the knowledge contained in the attention maps. Extensive experiments on COCO [21] validate the effectiveness of our D3ETR. It brings significant gains to various DETR-based student models. For example, D3ETR improves Conditional DETR-R50-C5 [24] by 7.8/2.4 mAP under 12/50 epochs training settings with Conditional DETR-R101-C5 [24] as the teacher.

In brief, our contributions are in three folds:

- We explore knowledge distillation for DETR-based detectors and make our attempts to solve the challenges of performing KD on the transformer decoder.
- We propose MixMatcher, which applies Adaptive Matching and Fixed Matching to model the relation between the DETR-based teacher and student. Then, we build a simple but effective distillation method D3ETR.
- The proposed method, D3ETR, could be applied to DETR-based detectors and improve the performance significantly.

2. Related Work

**Knowledge distillation in object detection.** Knowledge distillation is a method of model compression and transfer learning. It is first proposed to distill the knowledge from a large teacher model to a compact student model for the classification task [39]. Over the years, many improved KD methods have been proposed that perform distillation over intermediate features [27, 31], relation representation [25, 32], attention [40], etc. Recently, some works have successfully applied KD to object detection [11, 19, 37, 38]. ICD [46] proposes an instance-based conditional distillation framework, and it finds that initializing the student model with the teacher’s parameters will lead to faster convergence. DeFeat [11] decouples the foreground and background in the feature maps and distills them separately. FGD [37] uses focal and global distillation to guide the student model, achieving remarkable results. MGD [38] transforms the distillation into a feature generation task that uses the masked student features to generate the full teacher features. These efforts focus on the distillation on ordered outputs in CNN-based detectors. ViDT [30] proposes a variation of a transformer-based detector and applies KD on it, which directly performs distillation on patch tokens and detection queries between teacher and student. However, we notice that in DETRs, due to the output of the decoder being unordered, teacher queries and student queries do not have direct correspondence. Incremental-DETR [9] and DETRDistill [16] propose to construct the correspondence between teacher and student predictions through bipartite matching. However, they ignore the issue that the bipartite matching may be unstable in the early training stage [18].

We follow this idea and go a step further: we propose MixMatcher which helps alleviate such issues.

**DETR-based object detection.** With the pioneering work DETR [1] introducing transformers [33] to object detection, more and more follow-up works [18, 22, 24, 41, 48] have built various advanced extensions based on DETR because it removes the need for many hand-designed components like non-maximum suppression or initial anchor boxes generation. Deformable-DETR [48] introduced the multi-scale deformable attention scheme, which attends to only a small set of points around a reference and achieves better performance than DETR. More works focus on improving the design of decoder in DETR [6, 35]. Conditional DETR [24] rebuilt positional queries based on the reference points to facilitate extreme region discrimination. DAB-DETR [22] further extends the query as a 4D anchor box that also improves the performance. The follow-up DN-DETR [18] and DINO-DETR [41] introduced a novel query denoising algorithm to accelerate the training of the decoder. Group DETR [4, 5] and H-DETR [15] claim the multiple positive queries are the key to the fast convergence and introduce auxiliary groups as the decoder input ². These efforts show that the decoder design is very important to DETR. Different from the existing work on designing novel schemes in the decoder, we start from another orthogonal point of view and propose to transfer the knowledge in the decoder from a large model to a small model.

---

²These groups share the same decoder but have no interactions, and we borrow this design in our fixed matching.
Figure 2. Architecture of the proposed method. We propose the mixed teacher-student matching strategy that composes of two components, adaptive matching and fixed matching. We adopt two groups, where the first group feeds student queries to the decoder and the second feeds teacher queries to the decoder. Adaptive matching and fixed matching are applied to these two groups respectively. These two groups share the same decoder but have no interactions. Then we distill the knowledge in decoder self-attention, cross-attention, and prediction from the teacher model. The second group is only used in training.

3. Preliminary

In this section, we first review the architecture of DETR and the attention mechanism. Then we give some analysis of the DETR structure and investigate which part has the most significant impact on the performance.

3.1. DETR Architecture

The DETR architecture consists of a backbone (e.g., ResNet [13], a transformer encoder, a transformer decoder, and object class and box position predictors. The image features are extracted by the backbone and the transformer encoder layers model the global context. The transformer decoder takes $N$ object queries as input:

$$Q = \{q_1, \ldots, q_N\}$$

Each query in the decoder is responsible for predicting either a ground-truth object (with class and bounding box) or a “no object” class. The query may come in different forms, including the high-dimensional feature vector [1, 24], the anchor point coordinates [35], and the box coordinates [22]. The object queries are combined into the decoder embeddings, forming the queries of the self-attention and cross-attention layers in the decoder. The output query embeddings are fed into detection heads to produce $N$ object predictions.

3.2. Attention Mechanism

The attention [34] is computed using the scaled dot-product. The inputs contain: a set of $N_q$ queries $X_q \in \mathbb{R}^{d \times N_q}$, a set of $N_{kv}$ keys $X_k \in \mathbb{R}^{d \times N_{kv}}$, and a set of $N_{kv}$ values $X_v \in \mathbb{R}^{d \times N_{kv}}$. The attention weights are computed based on the softmax of dot-products between queries and keys:

$$a_{ij} = \frac{e^{\frac{1}{\sqrt{d}} X_q^i X_k^j}}{Z_i} \quad \text{where} \quad Z_i = \sum_{j=1}^{N_{kv}} e^{\frac{1}{\sqrt{d}} X_q^i X_k^j},$$

$$\text{Attn}(x_{qi}, X_k, X_v) = \sum_{j=1}^{N_{kv}} a_{ij} x_{vkJ}.$$
heads,

\[
\text{MultiHeadAttn}(x_{qi}, x_k, x_v) = W_o \text{Concat}(\text{head}_1, \text{head}_2, \ldots, \text{head}_M),
\]

\[
\text{head}_m = \text{Attn}(W^q_m x_{qi}, W^k_m x_k, W^v_m x_v),
\]

where \( W^q_m, W^k_m, W^v_m \in \mathbb{R}^{d'_m \times d} \) and \( W^p_m \in \mathbb{R}^{d_m \times d} \) are the projection matrices. The dimension \( d'_m \) is usually the same for all the heads and set to \( \frac{d}{M} \).

In self-attention, the keys, values, and queries are the same. In cross-attention, the keys and values are the same expect that the keys might contain positional embeddings, and the queries are different.

3.3. Analysis on the DETR structure

DETR-like methods compose of three parts: the backbone, the transformer encoder, and the transformer decoder. We conduct experiments to investigate which part has the biggest impact on the detection performance. Results are shown in Figure 3. We find that reducing the number of decoder layers from 6 to 1 leads to 17.8/18.3 mAP drops in performance for R50/R101 backbones respectively. Based on this observation, we propose to distill the knowledge in the decoder.

4. Methodology

We first introduce MixMatcher, which consists of two teacher-student matching strategies, adaptative matching and fixed matching. Then we introduce D^3ETR that distills the knowledge in decoder predictions, self-attention, and cross-attention from the teacher model.

4.1. MixMatcher

Adaptative Matching. The output of the DETR decoder is sparse and unordered, resulting in no direct one-to-one correspondence between teacher and student outputs. Inspired by DETR which performs bipartite matching between predicted and ground-truth objects, we propose to view the correspondence between teacher and student outputs as a bipartite matching problem.

Given the prediction \( y^t = (p^t, b^t) \) and \( y^s = (p^s, b^s) \) of the teacher and the student, where \( p \) is the soft logits for category prediction and \( b \) is the 4-D vector for box prediction. The pair-wise matching cost is defined as:

\[
C_{\text{match}}(y^t_i, y^s_{\xi(i)}) = \sum_{i=1}^{N_t} \left[ \ell_{\text{bce}}(p^t_i, p^s_{\xi(i)}) + \ell_{\text{box}}(b^t_i, b^s_{\xi(i)}) \right].
\]

Here \( N_t \) is the number of student predictions. \( \xi(\cdot) \) is a permutation of \( N_t \) teacher predictions and usually \( N_t \geq N_s \). \( \ell_{\text{bce}} \) is the binary cross-entropy loss and \( \ell_{\text{box}} = 20 \) is the trade-off coefficient. \( \ell_{\text{box}} \) is a combination of \( \ell_1 \) loss and GIoU loss [26] and the loss weights are 5 and 2, respectively.

To find a bipartite matching between teacher and student outputs, we search for a permutation of \( N_t \) elements \( \xi \in \Phi_{N_t} \) with the lowest cost:

\[
\hat{\xi} = \arg\min_{\xi \in \Phi_{N_t}} \sum_{i}^{N_s} C_{\text{match}}(y^t_i, y^s_{\xi(i)})
\]

To ease the training, DETR adopts the auxiliary decoding losses that each decoder layer would make detection predictions. This makes the prediction of the current stage a refinement of the previous stage. Accordingly, we adaptively match the outputs of teacher and student models at each decoder layer. Suppose the number of decoder layers is \( L \), we could apply the adaptive matching algorithm to each decoder layer and obtain \( L \) matching results: \( \{\hat{\xi}_1, \ldots, \hat{\xi}_L\} \).

Fixed Matching. The instability of bipartite graph matching may cause inconsistent optimization goals in early training stages [18]. To alleviate such an issue in teacher-student adaptative matching, we design an auxiliary group where we feed the fixed teacher queries into the student decoder\(^3\). Given the same input queries, we hope the outputs of the auxiliary group and the teacher model are well aligned.

Unfortunately, there also exists instability in the bipartite graph matching between the decoder prediction and the ground truth. This may result in such a situation: two outputs (auxiliary group and teacher model) generated from the same object query are supervised by different ground truths.

\(^3\)The dimensions of the query and the key might be different from the dimension the value. The attention formulation can be similarly derived.

\(^4\)We assume the teacher and student models share the same query format.
To solve this issue, we use the label assignment results of the teacher model to replace that of the auxiliary group in the last decoder layer:

$$\hat{\sigma}^t = \hat{\sigma}^s,$$  \hspace{1cm} (7)

where $\hat{\sigma}^t$ is the permutation of $N_t$ teacher predictions and $\hat{\sigma}^s$ is the permutation of $N_t$ student predictions in the auxiliary group. Under such constraints, each query of the teacher model and the auxiliary group in the student model is supervised by the same ground truth (or “no object” [1]), thus strengthening the one-to-one correspondence.

We make an ingenious design to combine the two matching strategies. We feed both the student group and the auxiliary group to the student decoder during training. They share the decoder parameters, but the two groups will not interact in the decoder self-attention. In inference, the auxiliary group is dropped and only the student group is used.

### 4.2. D^3ETR

After we obtain the correspondence of the queries between the teacher and student models, we can distill the teacher’s knowledge into the student model. According to the structure of the decoder, we design three distillation objectives: prediction distillation, self-attention distillation, and cross-attention distillation.

#### Self-attention distillation.

Decoder self-attention models the relations between object queries, which potentially plays the role of removing duplicate predictions [24]. Given $N$ object queries as input, we could obtain the multi-head self-attention weight map $A^k_s \in \mathbb{R}^{M \times N \times N}$ of the $k$-th decoder layer, according to Eq. 2 and Eq. 4. Similarly, we could obtain the multi-head self-attention weight map $\hat{A}^k_s$ of the teacher model. Please note that, although the number of teacher queries may be larger than that of the student queries, we could select queries according to the teacher-student correspondence. Then the decoder self-attention distillation loss is defined as:

$$\mathcal{L}_{sa} = \lambda_{sa} \sum_{k=1}^{L} \text{MSEloss}(A^k_s, \hat{A}^k_s),$$  \hspace{1cm} (8)

where $L$ is the number of decoder layers. $\lambda_{sa}$ is the loss weight and set as 10,000 by default.

#### Cross-attention distillation.

Decoder cross-attention takes the output of the self-attention layer as the queries and the output of the encoder as the keys and the values. It searches for regions of the object in the encoder output and aggregates them. Given encoder output $X \in \mathbb{R}^{C \times HW}$ and $N$ queries, we could obtain the multi-head cross-attention weight map $A^k_c \in \mathbb{R}^{M \times N \times HW}$ and $\hat{A}^k_c \in \mathbb{R}^{M \times N \times HW}$ of the student and teacher models, respectively. Then the decoder cross-attention distillation loss is defined as:

$$\mathcal{L}_{ca} = \lambda_{ca} \sum_{k=1}^{L} \text{MSEloss}(A^k_c, \hat{A}^k_c),$$  \hspace{1cm} (9)

where $\lambda_{ca}$ is the loss weight and set as 10,000 by default.

#### Prediction distillation.

After obtaining the teacher-student correspondence, we align the student’s prediction to the teacher’s. The prediction disillusion loss of the $k$-th layer is defined similarly to Eq. 5:

$$L^k_{pred}(y^s_k, y^t_k) = \sum_{i=1}^{N_t} [\mu_{cls}\ell_{bce}(p^s_k, p^{t(i)}_k) + \ell_{box}(b^s_k, b^{t(i)}_k)],$$  \hspace{1cm} (10)

$$L_{pred} = \sum_{k=1}^{L} L^k_{pred},$$  \hspace{1cm} (11)

where $y^s_k$ ($y^t_k$) is the $i$-th prediction of the student (teacher) in the $k$-th decoder layer.

#### Overall distillation loss function.

The above distillation losses are applied to both the student group and the auxiliary group. The overall loss function is defined as:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{sa} + \mathcal{L}_{ca} + \mathcal{L}_{pred} +$$

$$\mathcal{L}_{sa} + \mathcal{L}_{ca} + \mathcal{L}_{aux} +$$

$$\mathcal{L}_{aux} + \mathcal{L}_{pred},$$  \hspace{1cm} (12)

where $\mathcal{L}_{aux}$ is the loss of the auxiliary group.

### 4.3. Discussion

DETRDistill [16] is the most related work to ours. They construct the teacher-student correspondence through bipartite matching and introduce an auxiliary group to help decoder training. Ours is different from them in two aspects. (i) We propose MixMatcher which includes an auxiliary group, that helps alleviate the instability issue in teacher-student matching. In contrast, their auxiliary group have no interactions with the teacher model. (ii) We focus on distill the knowledge in decoder attention, while they distill the knowledge in the query feature.

### 5. Experiments

#### 5.1. Setting

**Dataset.** We perform the experiments on the COCO 2017 [20] detection dataset, which contains about 118K training (train) images and 5K validation (val) images.

**Training.** We follow the training setting of DETR [1] and Conditional DETR [24] that use ImageNet pre-trained backbone from TORCHVISION with Batch Normalisation (BN) layers fixed, and the transformer parameters are initialized using the Xavier initialization scheme [10]. We train the models for 12500 epochs with the AdamW [23]
5.2. Main Results

Our method can be applied to various DETR-like frameworks. We first conduct experiments on two popular detectors (DETR and Conditional DETR) with 12-epoch (1×) and 50-epoch training schedules. [46] propose the inheriting strategy which initializes the student with the teacher’s neck and head parameters and improves performance. Here we use this strategy to initialize the transformer encoder and decoder of the student with the parameter of the teacher.

### Results with a 12-epoch training schedule on MS COCO

Table 1 reports the results. All the student detectors obtain significant mAP improvements with the knowledge transferred from teacher detectors. For example, D3ETR boosts detection performance when applied to Conditional DETR: +7.8 mAP for R50-C5, +9.3 mAP for R18-C5, +5.8 mAP for R50-DC5, and +7.1 mAP for R18-DC5.

### Results with a 50-epoch training schedule

We further verify the effectiveness of the proposed method under a longer training schedule. Table 2 reports the results. We find that D3ETR could still improve the baseline models significantly. D3ETR improves Conditional DETR-R50-C5 by 2.4 mAP and Conditional DETR-R50-DC5 by 1.3 mAP, which even outperform the corresponding teacher detectors.

3. Ablation Study

In this section, we first compare the proposed decoder distillation method to other CNN-based distillation methods in object detection. Then we conduct ablation studies to verify each component in our decoder distillation strate-

---

| Teacher          | Student | Backbone | mAP  | APs  | Apm  | ApL  |
|------------------|---------|----------|------|------|------|------|
| DETR R101-C5     | DETR    | R50-C5   | 25.1 | 7.7  | 24.9 | 43.1 |
|                  | + Ours  | R50-C5   | 30.6 (+5.5) | 9.7  | 32.0 | 49.9 |
| DETR R101-DC5    | DETR    | R18-C5   | 19.7 | 4.5  | 18.8 | 34.8 |
|                  | + Ours  | R18-C5   | 28.3 (+8.6) | 7.7  | 29.3 | 48.1 |
| Conditional DETR | Conditional DETR | R50-C5 | 28.4 | 9.7  | 29.3 | 46.9 |
|                  | + Ours  | R50-DC5  | 39.3 (+10.9) | 17.1 | 43.1 | 59.2 |
| Conditional DETR | Conditional DETR | R18-C5 | 23.0 | 6.4  | 22.9 | 39.4 |
|                  | + Ours  | R18-DC5  | 33.0 (+10.0) | 12.4 | 34.7 | 52.8 |

Table 1. Results with a 12-epoch training schedule on MS COCO. We highlight the improvements brought by our proposed method on two DETR-based methods. We initialize the parameters of the encoder and decoder from the teacher model.

optimizer. The learning rate is dropped by a factor of 10 after 11/40 epochs, respectively. We use the augmentation scheme same as DETR [1]: resize the input image such that the short side is at least 480 and at most 800 pixels and the long side is at most 1333 pixels; randomly crop the image such that a training image is cropped with a probability of 0.5 to a random rectangular patch.

#### Teacher models

For DETR [1], we use the officially released models trained for 500 epochs that use ResNet-101-C5 or ResNet-101-DC5 backbone as the teacher model. For Conditional DETR [24], we use the official code to train the model for 50 epochs that use ResNet-101-C5 or ResNet-101-DC5 backbone.

#### Student models

For DETR and Conditional DETR, we train student models with the AdamW [23] optimizer for 12/50 epochs based on four different backbones: ResNet-50-C5, ResNet-50-DC5, ResNet-18-C5, ResNet-18-DC5.

#### Evaluation

We use the standard COCO evaluation. We report the average precision (AP), and the AP scores at 0.50, 0.75 and for the small, medium, and large objects.

#### 5.3. Ablation Study

In this section, we first compare the proposed decoder distillation method to other CNN-based distillation methods in object detection. Then we conduct ablation studies to verify each component in our decoder distillation strate-

---

5 The inheriting strategy brings a gain of 2.1/0.4 mAP on Conditional DETR-R50-C5 under 12/50 epochs setting.
We compare D³ETR and other state-of-the-art object detection KD approaches in Table 4. These works all focus on the distillation of ordered outputs, and we apply them to the output feature of the transformer encoder. We use the same teacher and student models and the same training settings in each case. For competing distillation methods, we tune the hyper-parameters based on those in the corresponding papers or open-sourced code repositories and take the best results. From the table, we can find that the proposed method is superior to others. This illustrates that it is more efficient to perform distillation on the DETR decoder layers. Furthermore, we combine the proposed method with MGD [38] and the performance is improved to 38.8 mAP. This shows the possibility of further improving the performance of our method. However, it is not the focus of this paper, so we leave it as future work.

The effectiveness of decoder distillation. We compare D³ETR and other state-of-the-art object detection KD approaches in Table 4. These works all focus on the distillation of ordered outputs, and we apply them to the output feature of the transformer encoder. We use the same teacher and student models and the same training settings in each case. For competing distillation methods, we tune the hyper-parameters based on those in the corresponding papers or open-sourced code repositories and take the best results. From the table, we can find that the proposed method is superior to others. This illustrates that it is more efficient to perform distillation on the DETR decoder layers. Furthermore, we combine the proposed method with MGD [38] and the performance is improved to 38.8 mAP. This shows the possibility of further improving the performance of our method. However, it is not the focus of this paper, so we leave it as future work.

The effect of each component in our method. We gradually add the proposed strategies to the baseline and present

![Table 2](image-url) Results with a 50-epoch training schedule on MS COCO. We highlight the improvements brought by our proposed method on two DETR-based methods. We initialize the parameters of the encoder and decoder from the teacher model.

| Teacher       | Student   | Backbone | mAP  | AP_a | AP_m | AP_l |
|---------------|-----------|----------|------|------|------|------|
| DETR R101-C5  | DETR      | R50-C5   | 34.8 | 13.9 | 37.3 | 54.5 |
|               | + Ours    |          | 39.7 (+4.9) | 17.7 | 42.8 | 60.4 |
| DETR R101-DC5 | DETR      | R50-DC5  | 38.3 | 17.0 | 41.4 | 57.8 |
|               | + Ours    |          | 42.5 (+4.2) | 21.1 | 46.6 | 61.8 |
| Conditional D  | Conditional D | R50-C5   | 40.9 | 20.6 | 44.3 | 59.3 |
| DETR R101-C5  | + Ours    |          | 43.3 (+2.4) | 22.3 | 46.9 | 62.1 |
| Conditional D  | Conditional D | R18-C5   | 35.8 | 16.0 | 38.6 | 53.8 |
| DETR R101-DC5 | + Ours    |          | 39.6 (+3.8) | 18.8 | 42.9 | 59.2 |
| Conditional D  | Conditional D | R50-DC5  | 43.7 | 23.9 | 47.6 | 60.1 |
| DETR R101-DC5 | + Ours    |          | 45.0 (+1.3) | 25.3 | 48.7 | 63.3 |
| Conditional D  | Conditional D | R18-DC5  | 39.2 | 19.2 | 42.4 | 56.6 |
|               | + Ours    |          | 42.0 (+2.8) | 22.3 | 45.6 | 60.4 |

| Adaptative Matching | Prediction | Self-Attn | Cross-Attn | Fixed Matching | Inheriting | mAP |
|---------------------|------------|-----------|------------|----------------|-----------|-----|
| ✓                   | ✓          | ✓         | ✓          |                |           | 32.4 |
| ✓                   | ✓          | ✓         | ✓          |                |           | 35.1 |
| ✓                   | ✓          | ✓         | ✓          |                | ✓         | 34.5 |
| ✓                   | ✓          | ✓         | ✓          |                | ✓         | 35.8 |
| ✓                   | ✓          | ✓         | ✓          |                | ✓         | 35.8 |
| ✓                   | ✓          | ✓         | ✓          |                | ✓         | 36.0 |
| ✓                   | ✓          | ✓         | ✓          |                | ✓         | 36.7 |
| ✓                   | ✓          | ✓         | ✓          | ✓              | ✓         | 38.1 |
| ✓                   | ✓          | ✓         | ✓          | ✓              | ✓         | 40.2 |
Figure 4. Comparison of spatial attention maps. The first row is the student model. The second row is the student model with our D3ETR. The third row is the teacher model. We choose 4 of 8 heads for visualization and others are duplicates. The purple/blue boxes are ground-truth boxes. Best viewed in color.

| Method                  | #Epochs | mAP  |
|-------------------------|---------|------|
| Conditional DETR-R101-C5 (∗) | 50      | 42.8 |
| Conditional DETR-R50-C5 (♣) | 12      | 32.4 |
| ♣ + DeFeat              | 12      | 32.4 |
| ♣ + Ftnet               | 12      | 33.3 |
| ♣ + FGD                 | 12      | 36.0 |
| ♣ + MGD                 | 12      | 36.7 |
| ♣ + Ours                | 12      | 38.1 |
| ♣ + Ours + MGD          | 12      | 38.8 |

Table 4. Comparison with other distillation methods. We adopt Conditional DETR-R101-C5 as the teacher model and Conditional DETR-R50-C5 as the student. Our D3ETR is superior to other distillation methods, and could be further improved by combining with MGD.

| Method       | Constraint       | mAP  |
|--------------|------------------|------|
| Fixed Matching | x                | 37.0 |
| Fixed Matching | 0 ~ 5-th layer  | 37.9 |
| Fixed Matching | 5-th layer      | 38.1 |

Table 6. Ablation study on the constraint in fixed matching. The best result is obtained by adding a constraint on the last decoder layer.

Table 5. Ablation study on the teacher-student matching strategy. “✓✓” means we use two groups that have the same teacher-student matching strategy. The best result is obtained by using the two proposed strategies simultaneously.

| Adapative Matching | Fixed Matching | mAP  |
|--------------------|----------------|------|
| ✓                  | ✓              | 32.4 |
| ✓                  | ✓              | 36.7 |
| ✓✓                 | ✓✓             | 37.2 |
| ✓                  | ✓              | 36.4 |
| ✓✓                 | ✓              | 38.1 |

The results in Table 3. If no teacher-student matching strategy is used, the result is 35.1. When adopting the adaptive matching strategy, the distillation in prediction, self-attention, and cross-attention can effectively improve the results, and the combination of the three achieves the best result (36.7). Based on this, fixed matching further improves the result to 38.1. At last, the inheriting strategy brings a gain of 2.1 and we obtain 40.2 mAP, which is comparable to Conditional DETR-R50-C5 trained with 50 epochs (40.9).

Teacher-student matching strategy. The proposed MixMatcher composes of two teacher-student matching strategies: adaptive matching and fixed matching. To validate our choice, we perform ablations on the matching strategy and report the results in Table 5. We have some findings. First, either using adaptive matching or fixed matching could improve the baseline. Adaptative matching is slightly more efficient and achieves 36.7 mAP. Second, when we use two groups of queries during training and adopt the same matching strategy, we find that the results can be further improved. For example, the two groups with adaptative matching are 0.5 higher than one group. We guess this is because multiple groups of queries allow each ground truth to match more positive queries, thus easing training. Finally, using these two different strategies at the same time works best and obtains 38.1 mAP. This illustrates the auxiliary group that uses fixed matching can help alleviate the instability issue existing in adaptive matching.

Constraint in fixed matching. We add a constraint in fixed matching to strengthen the teacher-student fixed correspondence, as illustrated in Eq. 7. Without the constraint, the output of the auxiliary group may be supervised by different
ground truths from the corresponding output of the teacher model. Since the two outputs are generated from the same object query, it may make the model confused. As listed in Table 6, the performance drops from 38.1 to 37.0. We also try to add constraints on all decoder layers and find the performance slightly worse than adding a constraint on the last layer.

5.4. Visualization

We visualize the spatial attention map [24] to verify whether the student learns useful information from the teacher model. The results are shown in Figure 4. According to [24], the spatial attention maps correspond to object extremities or a small region inside the object box. The object extremities help locate the object and the small region inside the object help recognize the category of the object. We find that it is hard for the student to precisely locate object extremities. With our D³ETR, the knowledge in the teacher model is well transferred to the student model. The student model learns similar patterns to the teacher model, thus improving the detection performance.

6. Conclusion

In this paper, we explore knowledge distillation for DETR-based detectors. We propose MixMatcher which models the correspondence between the DETR-based teacher and student. Based on MixMatcher, we propose a simple but effective distillation method D³ETR and demonstrate its effectiveness with extensive experiments.

Acknowledgements. We would like to acknowledge Qiang Chen for the helpful discussions.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In ECCV, 2020. 1, 2, 3, 5, 6
[2] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. Advances in neural information processing systems, 30, 2017. 1
[3] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 5008–5017, 2021. 1
[4] Qiang Chen, Xiaokang Chen, Jian Wang, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. 2022. 2
[5] Qiang Chen, Jian Wang, Chuchu Han, Shan Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaoqi Wang, Shuming Han, Gang Zhang, Haocheng Feng, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Group detr v2: Strong object detector with encoder-decoder pretraining. 2022. 2
[6] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang. Conditional detr v2: Efficient detection transformer with box queries. arXiv preprint arXiv:2207.08914, 2022. 1, 2
[7] Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In Proceedings of the IEEE/CVF international conference on computer vision, pages 4794–4802, 2019. 1
[8] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7842–7851, 2021. 1
[9] Na Dong, Yongqiang Zhang, Mingli Ding, and Gim Hee Lee. Incremental-detr: Incremental few-shot object detection via self-supervised learning. arXiv preprint arXiv:2205.04042, 2022. 2
[10] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In AISTATS, 2010. 5
[11] Jinyuan Guo, Kai Han, Yunhe Wang, Han Wu, Xinghao Chen, Chunjing Xu, and Chang Xu. Distilling object detectors via decoupled features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 2154–2164, 2021. 1, 2
[12] Miao Hao, Yitao Liu, Xiangyu Zhang, and Jian Sun. Labelenc: A new intermediate supervision method for object detection. In European Conference on Computer Vision, pages 529–545. Springer, 2020. 1
[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016. 1, 3
[14] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. arXiv: Machine Learning, 2015. 1
[15] Ding Jia, Yuhui Yuan, Haodi He, Xiaopei Wu, Haojun Yu, Weihong Lin, Lei Sun, Chao Zhang, and Han Hu. Detrs with hybrid matching. arXiv preprint arXiv:2207.13080, 2022. 2
[16] Chang Jiahao, Wang Shuo, Xu Guangkai, Chen Zehui, Yang Chenhongyi, and Zhao Feng. Detrdistill: A simple knowledge distillation framework for detr-families. In ICLR2023 submission, 2022. 2, 5
[17] Harold W Kuhn. The hungarian method for the assignment problem. Naval research logistics quarterly, 1955. 2
[18] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In CVPR, 2022. 1, 2, 4
[19] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In Proceedings of the ieee conference on computer vision and pattern recognition, pages 6356–6364, 2017. 2
[20] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In ECCV, 2014. 5
[21] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence
