D^{3}ETR: Decoder Distillation for Detection Transformer

Xiaokang Chen[1] , **Jiahui Chen**[2] , **Yan Liu**[3] , **Jiaxiang Tang**[1] and **Gang Zeng**[1]
[1]National Key Laboratory of General Artificial Intelligence, School of IST, Peking University
[2]Beihang University
[3]The Chinese University of Hong Kong
pkucxk@pku.edu.cn

Abstract

Although various knowledge distillation (KD) methods for CNN-based detectors have been proven effective in improving small students, building baselines and recipes for DETR-based detectors remains a challenge. This paper concentrates on the transformer decoder of DETR-based detectors and explores KD methods suitable for them. However, the random order of the decoder outputs poses a challenge for knowledge distillation as it provides no direct correspondence between the predictions of the teacher and the student. To this end, we propose MixMatcher that aligns the decoder outputs of DETR-based teacher and student, by mixing two teacher-student matching strategies for combined advantages. The first strategy, Adaptive Matching, applies bipartite matching to adaptively match the outputs of the teacher and the student in each decoder layer. The second strategy, Fixed Matching, fixes the correspondence between the outputs of the teacher and the student with the same object queries as input, which alleviates instability of bipartite matching in Adaptive Matching. Using both strategies together produces better results than using either strategy alone. Based on MixMatcher, we devise **D**ecoder **D**istillation for **DE**tection **TR**ansformer (D^{3}ETR), which distills knowledge in decoder predictions and attention maps from the teacher to student. D^{3}ETR shows superior performance on various DETR-based detectors with different backbones. For instance, D^{3}ETR improves Conditional DETR-R50-C5 by **8.3** mAP under 12 epochs training setting with Conditional DETR-R101-C5 serving as the teacher. The code will be released.

1 Introduction

The concept of Knowledge distillation (KD) [Hinton *et al.*, 2015] involves transferring knowledge from a large teacher model to a small student model, to enhance the student's performance without incurring costs during model inference. There has been significant progress in the development of KD methods in recent years, with encouraging results in many vision tasks, such as image classification [Romero *et al.*, 2014; Zagoruyko and Komodakis, 2016; Cho and Hariharan, 2019; Zhang *et al.*, 2018; Tian *et al.*, 2019; Zhou *et al.*, 2021; Yang *et al.*, 2021; Chen *et al.*, 2021; Zhao *et al.*, 2022] and object detection [Chen *et al.*, 2017; Zhang and Ma, 2020; Hao *et al.*, 2020; Dai *et al.*, 2021; Guo *et al.*, 2021; Zhang *et al.*, 2022c; Yang *et al.*, 2022a; Yang *et al.*, 2022b]. However, these methods are predominantly focused on CNN-based models [Simonyan and Zisserman, 2015; He *et al.*, 2016; Sandler *et al.*, 2018] and are related to model structures, particularly in object detection [Chen *et al.*, 2017; Hao *et al.*, 2020; Yang *et al.*, 2022a; Zhang *et al.*, 2022c]. Applying existing KD techniques to novel detectors, such as DETR-based detectors [Carion *et al.*, 2020; Zhu *et al.*, 2020; Meng *et al.*, 2021; Chen *et al.*, 2022c; Liu *et al.*, 2022; Li *et al.*, 2022], poses some challenges and may result in unsatisfactory improvements. Hence, this paper seeks to address this gap by exploring KD strategies for DETR-based detectors.

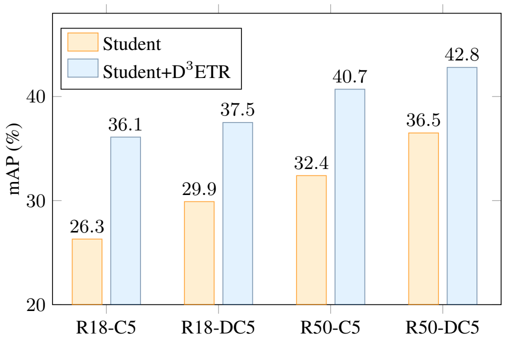

Figure 1: **Improvements over Conditional DETR under** 1× **schedule on COCO 2017** ***val*** **set.** Our D^{3}ETR obtains consistent gains over different backbones.

DETR [Carion *et al.*, 2020] is an end-to-end detector that employs transformer layers [Vaswani *et al.*, 2017]. The pipeline of DETR and its variants involves (i) extracting image features with a backbone, (ii) modeling global context with a transformer encoder, and (iii) predicting objects with a transformer decoder, given the image features and object queries. To develop KD baselines and recipes for DETR-based detectors, we analyze the impact of each component and find that the transformer decoder is crucial for maintaining good performance. Therefore, we focus on exploring KD techniques in the transformer decoder.

However, there is a challenge that the DETR decoder produces outputs in a random order[1], resulting in no direct correspondence between the decoder outputs of the teacher model and the student model. To address this issue, we introduce *MixMatcher*, which aims to align the decoder outputs of the teacher and student models. MixMatcher combines two strategies, *Adaptive Matching* and *Fixed Matching*, to establish teacher-student correspondence. Adaptive Matching computes the optimal bipartite matching [Kuhn, 1955] 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 introduce Fixed Matching. We feed the teacher's fixed object queries to the student's decoder as an auxiliary group and apply Fixed Matching to align the outputs of the teacher and student models.

MixMatcher enables us to establish correspondence between the decoder outputs of the teacher and student models. We develop *Decoder Distillation for DETR-Based Methods (D^{3}ETR)* based on MixMatcher. In addition to predictions, we also consider attention modules (consisting of self-attention and cross-attention) in the decoder layers during distillation. For attention modules, we distill the knowledge contained in the attention maps. The experiments conducted on COCO [Lin *et al.*, 2014a] demonstrate the effectiveness of our proposed D^{3}ETR. It achieves significant improvements in various DETR-based student models (Figure 1). For instance, D^{3}ETR improves Conditional DETR-R50-C5 [Meng *et al.*, 2021] by **8.3** mAP under 12 epochs training setting, with Conditional DETR-R101-C5 [Meng *et al.*, 2021] acting as the teacher.

To summarize, our contributions are in three folds:

- We explore knowledge distillation for DETR-based detectors, and we attempt to address the challenges of distilling knowledge from the transformer decoder.
- We introduce MixMatcher, which combines Adaptive Matching and Fixed Matching to establish the relationship between the DETR-based teacher and student models. Based on MixMatcher, we propose a simple and effective distillation method named D^{3}ETR.
- Extensive experiments reveal that our proposed method, D^{3}ETR, improves the performance of DETR-based detectors significantly.

2 Related Work

Knowledge distillation in object detection. Knowledge distillation (KD) is a technique used for model compression and transfer learning. Originally proposed for distilling knowledge from a large teacher model to a compact student model for classification tasks [Yim *et al.*, 2017], KD has since been improved to perform distillation over intermediate features [Romero *et al.*, 2014; Tian *et al.*, 2019], relation representation [Park *et al.*, 2019; Tung and Mori, 2019], attention [Zagoruyko and Komodakis, 2016], and other aspects. Some recent works have successfully applied KD to object detection [Li *et al.*, 2017; Guo *et al.*, 2021; Yang *et al.*, 2022a; Yang *et al.*, 2022b]. ICD [Zheng *et al.*, 2021] proposes an instance-based conditional distillation framework and finds that initializing the student model with the teacher's parameters leads to faster convergence. DeFeat [Guo *et al.*, 2021] decouples foreground and background in the feature maps and distills them separately. FGD [Yang *et al.*, 2022a] uses focal and global distillation to guide the student model, achieving remarkable results. MGD [Yang *et al.*, 2022b] transforms distillation into a feature-generation task that uses the masked student features to generate the full teacher features. These efforts focus on distillation of ordered outputs in CNN-based detectors.

ViDT [Song *et al.*, 2022] proposes a variation of a transformer-based detector and applies KD to it, performing distillation directly on patch tokens and detection queries between teacher and student. However, in DETRs, the decoder outputs are unordered, creating a lack of direct correspondence between teacher and student queries. Incremental-DETR [Dong *et al.*, 2022] and DETRDistill [Jiahao *et al.*, 2022] propose to construct correspondence between teacher and student predictions through bipartite matching but ignore that bipartite matching between teacher and student may be unstable in the early training stage [Li *et al.*, 2022]. In this paper, we propose MixMatcher, which helps alleviate these issues.

DETR-based object detection. The pioneering work, DETR [Carion *et al.*, 2020], introduces transformers [Vaswani *et al.*, 2017] to object detection, eliminating the need for hand-designed components like non-maximum suppression or initial anchor boxes generation. Since then, several follow-up works [Zhu *et al.*, 2020; Meng *et al.*, 2021; Liu *et al.*, 2022; Li *et al.*, 2022; Zhang *et al.*, 2022b; Zhang *et al.*, 2022a] have built various advanced extensions based on DETR. Deformable-DETR [Zhu *et al.*, 2020] introduced the multi-scale deformable attention scheme, which attends to a small set of points around a reference and achieves better performance than DETR. Other works such as Conditional DETR [Meng *et al.*, 2021] rebuild positional queries based on reference points for extreme region discrimination. DAB-DETR [Liu *et al.*, 2022] extends the query to a 4D anchor box for improved performance. Follow-up work DN-DETR [Li *et al.*, 2022] and DINO-DETR [Zhang *et al.*, 2022b] introduced a novel query denoising algorithm that accelerates decoder training. Group DETR [Chen *et al.*, 2022a; Chen *et al.*, 2022b] and H-DETR [Jia *et al.*, 2022] claim that multiple positive queries are key to fast convergence. These works emphasize that the decoder design plays a pivotal role in DETR. Different from the existing work on designing novel schemes in the decoder, our proposal starts from another orthogonal point of view and transfers the knowledge in the decoder from a large model to a smaller model.

[1] In contrast to traditional object detectors based on CNNs, DETR treats object detection as a set prediction problem. We conduct an ablation in Table 3 to verify the importance of constructing the corresponding between the teacher and student decoder outputs.

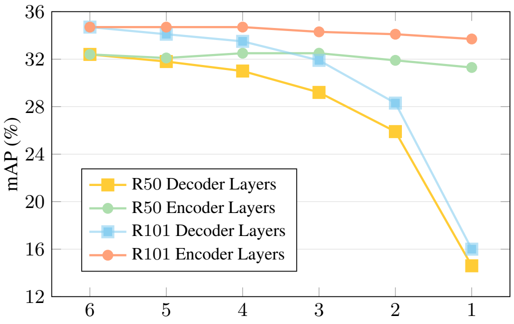

Figure 2: **Analysis of different components in Conditional DETR on COCO 2017 *val*.** The x-axis represents the number of encoder/decoder layers. We adopt ResNet-101/ResNet-50 as the backbone and train the model for 12 epochs (1× schedule). We find that the performance significantly drops when the number of decoder layers is reduced.

3 Preliminary

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

3.1 DETR Architecture

The DETR architecture is composed of several components: a backbone, such as ResNet [He *et al.*, 2016], a transformer encoder, a transformer decoder, and object class and box position predictors. The backbone is responsible for extracting image features, and the transformer encoder layers model the global context. The transformer decoder takes N object queries, denoted by $\mathbf{Q} = \{\mathbf{q}_1, \ldots, \mathbf{q}_N\}$, as input. Each query is responsible for predicting the class and bounding box of either a ground-truth object or a “no object” class. Different forms of queries can be used, such as high-dimensional feature vectors [Carion *et al.*, 2020; Meng *et al.*, 2021], anchor point coordinates [Wang *et al.*, 2022], and box coordinates [Liu *et al.*, 2022]. The object queries are combined into decoder embeddings, which form the queries of the self-attention and cross-attention layers in the decoder. The embeddings are fed into the detection heads to produce N object predictions.

3.2 Analysis on the DETR Structure

DETR-like methods consist of the backbone, the transformer encoder, and the transformer decoder. We conduct experiments to investigate which part has the most significant effect on the detection performance. Figure 2 depicts the results. We find that reducing the number of decoder layers from 6 to 1 leads to a decrease in mAP of 17.8/18.3 for R50/R101 backbones, respectively. Based on this observation, we propose to distill the knowledge in the decoder.

4 Methodology

In this section, we present our proposed method, MixMatcher (Figure 3), which includes two teacher-student matching strategies, namely Adaptative Matching and Fixed Matching. Additionally, we introduce D^{3}ETR, which distills knowledge from the teacher model in decoder predictions, self-attention, and cross-attention.

4.1 MixMatcher

Adaptative Matching. As the output of the DETR decoder is sparse and unordered, direct one-to-one correspondence between the teacher's and student's outputs is not achievable. To address this problem, we propose to view the correspondence between the teacher's and student's outputs as a bipartite matching problem, inspired by DETR's strategy of performing bipartite matching between predicted and ground-truth objects.

Given the prediction $y^t = (\mathbf{p}^t, \mathbf{b}^t)$ and $y^s = (\mathbf{p}^s, \mathbf{b}^s)$ of the teacher and the student, where $\mathbf{p}$ represents the soft logits for category prediction and $\mathbf{b}$ represents the 4-D vector for box prediction. The pairwise matching cost is:

$$\mathcal{C}_{\text{match}}(y_i^s, y_{\xi(i)}^t) = \sum\nolimits_{i=1}^{N_s} [\mu_{\text{cls}} \ell_{\text{bce}}(\mathbf{p}_i^s, \mathbf{p}_{\xi(i)}^t) + \ell_{\text{box}}(\mathbf{b}_i^s, \mathbf{b}_{\xi(i)}^t)]. \quad (1)$$

Here N_s is the number of predictions made by students. $\xi(\cdot)$ denotes a permutation of N_t teacher predictions. ℓ_{bce} is the binary cross-entropy loss and $\mu_{\text{cls}} = 20$ is the trade-off coefficient. ℓ_{box} is a combination of ℓ_1 loss and GIoU loss [Rezatofighi *et al.*, 2019], with loss weights of 10 and 2, respectively.

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

$$\hat{\xi} = \underset{\xi \in \Phi_{N_t}}{\text{argmin}} \sum_{i}^{N_s} \mathcal{C}_{\text{match}}(y_i^s, y_{\xi(i)}^t) \quad (2)$$

To facilitate the training process, DETR adopts the auxiliary decoding losses such that each decoder layer would make detection predictions, refining the previous stage's predictions. Therefore, we adaptatively match the outputs of teacher and student models at each decoder layer. If there are L decoder layers, we can apply the adaptative 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 [Li *et al.*, 2022]. To alleviate this problem in teacher-student adaptative matching, we introduce an auxiliary group for the student model. Within this group, we feed the fixed teacher queries into the student decoder[2]. By giving both models the same input queries, we hope to achieve well-aligned outputs between the auxiliary group and the teacher model.

Unfortunately, instability within the bipartite graph matching process can also arise between the decoder prediction and the ground truth. This may result in a situation where two outputs, generated from the same object query, are supervised by different ground truths. To resolve this issue, we use the label

[2] We assume teacher and student share the same query format.

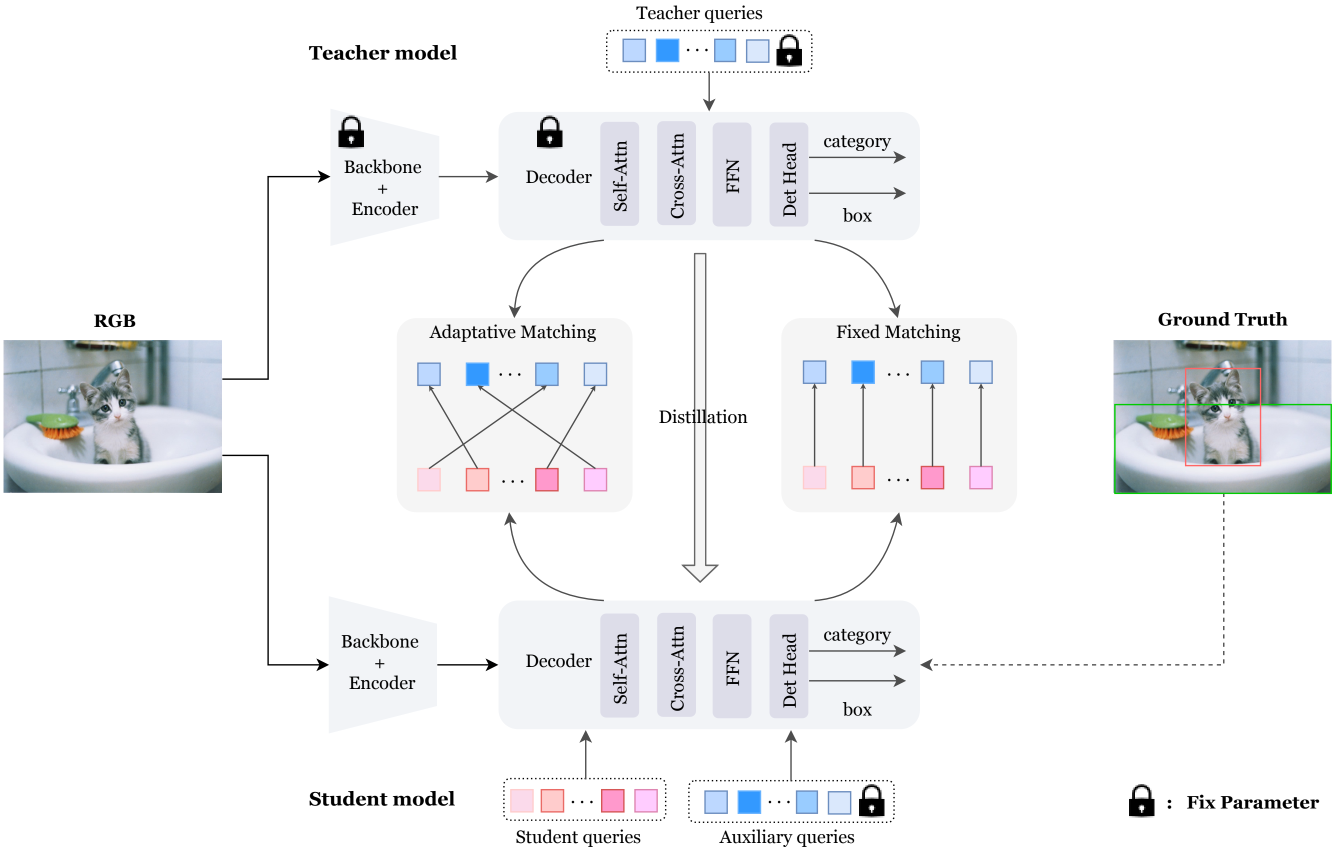

Figure 3: **Overview of the proposed method**. We propose a mixed teacher-student matching strategy consisting of two components: Adaptative Matching and Fixed Matching. We adopt two groups of queries for the student model, with the first group feeding student queries to the decoder while the second group feeding teacher queries to the same decoder. The adaptative matching and Fixed Matching techniques are applied to the first and second groups respectively. Although these two groups share the same decoder, there are no interactions between them. Subsequently, the knowledge in decoder self-attention, cross-attention, and prediction from the teacher model is distilled. It is important to highlight that the second group is exclusively employed during training. Best viewed on screen.

assignment results of the teacher model to replace those of the auxiliary group in the final decoder layer, such that:

$$\hat{\sigma}^s = \hat{\sigma}^t, \tag{3}$$

where $\hat{\sigma}^t$ is the permutation of N_t predictions of the teacher and $\hat{\sigma}^s$ is the permutation of N_t predictions of the student in the auxiliary group. Under these constraints, the teacher model and the auxiliary group in the student model are supervised under the same ground truth (or "no object" [Carion *et al.*, 2020]), enhancing the one-to-one correspondence.

We employ an ingenious approach to integrate the two matching strategies. During training, we provide the student decoder with both the student group and the auxiliary group. These groups share the same decoder parameters, while not interacting with each other in the decoder self-attention. During inference, we exclusively use the student group, while dropping the auxiliary group. The model inference process is the same as the student trained normally.

4.2 D^{3}ETR

Upon establishing query correspondence between the teacher and student models, the teacher's knowledge can be well distilled into the student model. With the decoder's structure in mind, we define three distillation objectives that comprise prediction distillation, self-attention distillation, and cross-attention distillation [3].

Self-attention distillation. Decoder self-attention models the relations between object queries and potentially aids in eliminating duplicate predictions [Meng *et al.*, 2021]. Given a set of N object queries as input, we can obtain the multi-head self-attention weight map of the k-th decoder layer, $\mathbf{A}_s^k \in \mathbb{R}^{M\times N\times N}$. Similarly, we obtain the multi-head self-attention weight map of the teacher model, denoted by $\tilde{\mathbf{A}}_s^k$. Note that, even though the number of teacher queries may be larger than that of the student queries, queries can be selected based on the teacher-student correspondence. Further, we define the decoder self-attention distillation loss as follows:

$$\mathcal{L}_{\text{sa}} = \lambda_{\text{sa}} \sum_{k=1}^{L} \texttt{MSEloss}(\mathbf{A}_s^k, \tilde{\mathbf{A}}_s^k), \tag{4}$$

[3] Please note that the following distillations are carried out between matched outputs of teacher and student models.

where L denotes the number of decoder layers, λ_{sa} is the loss weight and set to $10,000$ as default.

Cross-attention distillation. The decoder's cross-attention mechanism utilizes the encoder's output as the keys and values, and the output of the self-attention layer is used as the queries. It identifies the relevant regions of the object in the encoder output and aggregates their features. By employing a set of N queries and an encoder output $\mathbf{X} \in \mathbb{R}^{C\times HW}$, the student and teacher models produce the multi-head cross-attention weight maps, namely $\mathbf{A}_c^k \in \mathbb{R}^{M\times N\times HW}$ and $\tilde{\mathbf{A}}_c^k \in \mathbb{R}^{M\times N\times HW}$, respectively. Then the decoder cross-attention distillation loss function is formulated as:

$$\mathcal{L}_{\text{ca}} = \lambda_{\text{ca}} \sum_{k=1}^{L} \texttt{MSEloss}(\mathbf{A}_c^k, \tilde{\mathbf{A}}_c^k), \tag{5}$$

where the loss weight λ_{ca} defaults to $10,000$.

Prediction distillation. After establishing the teacher-student correspondence, we align the student's prediction to that of the teacher. The prediction distillation loss of the k-th layer is defined similarly to Eq. 1:

$$\mathcal{L}^k_{\text{pred}}(y_i^{sk}, y^{tk}_{\xi(i)}) = \sum\nolimits_{i=1}^{N_s}[\mu_{\text{cls}}\ell_{\text{bce}}(\mathbf{p}_i^{sk}, \mathbf{p}^{tk}_{\xi(i)}) + \ell_{\text{box}}(\mathbf{b}_i^{sk}, \mathbf{b}^{tk}_{\xi(i)})], \tag{6}$$

$$\mathcal{L}_{\text{pred}} = \sum_{k=1}^{L} \mathcal{L}^k_{\text{pred}}, \tag{7}$$

where y_i^{sk} ($y^{tk}_{\xi(i)}$) represents the i-th prediction made by the student (teacher) in the k-th decoder layer.

Overall distillation loss function. The aforementioned distillation loss functions are applied to the student group and the auxiliary group. The overall loss function is expressed as below:

$$\mathcal{L}_{\text{distill}} = \mathcal{L}_{\text{sa}} + \mathcal{L}_{\text{ca}} + \mathcal{L}_{\text{pred}} + \mathcal{L}^{\text{aux}}_{\text{sa}} + \mathcal{L}^{\text{aux}}_{\text{ca}} + \mathcal{L}^{\text{aux}}_{\text{pred}}, \tag{8}$$

where $\mathcal{L}^{\text{aux}}$ denotes the loss of the auxiliary group.

4.3 Discussion

DETRDistill [Jiahao *et al.*, 2022] is the most related work to ours. They also explore KD in DETR-like frameworks. Our work diverges from them in a couple of ways. (i) We concentrate on distilling the knowledge contained in decoder attention and prediction, while their focus is on the distillation of query feature knowledge. (ii) Although they also consider how to establish teacher-student correspondence, they ignore the instability issue in teacher-student matching. We propose Fixed Matching to address the issue and experimental results demonstrate the significance of considering this problem.

5 Experiments

5.1 Setting

Dataset. We perform the experiments on the COCO 2017 [Lin *et al.*, 2014b] detection dataset, which contains about 118K training (`train`) images and 5K validation (`val`) images.

Training. We follow the training setting of DETR [Carion *et al.*, 2020] and Conditional DETR [Meng *et al.*, 2021] that use ImageNet pre-trained backbone from TORCHVISION with Batch Normalisation (BN) layers fixed. The transformer parameters are initialized using the Xavier initialization scheme [Glorot and Bengio, 2010]. We train our models for 12/50 epochs utilizing the AdamW [Loshchilov and Hutter, 2017] optimizer. The learning rate is reduced by a factor of 10 after 11/40 epochs, respectively. The data augmentation scheme is identical to DETR [Carion *et al.*, 2020]: the input image is resized such that the short side is at least 480 pixels and at most 800 pixels and the long side is at most 1333 pixels. The training image is then randomly cropped with a probability of 0.5 to a random rectangular patch.

Teacher models. For DETR [Carion *et al.*, 2020], we utilize the officially released models, trained for 500 epochs with ResNet-101-C5 or ResNet-101-DC5 backbone as the teacher model. For Conditional DETR [Meng *et al.*, 2021], we utilize the official code to train the model for 50 epochs, utilizing ResNet-101-C5 or ResNet-101-DC5 backbone.

Student models. Student models are trained with the AdamW [Loshchilov and Hutter, 2017] optimizer for 12/50 epochs. The student models are trained 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.

5.2 Main Results

Our method is versatile and can be applied to various DETR-like frameworks. We perform experiments on two prevalent detectors (namely, DETR and Conditional DETR) with 12-epoch (1×) and 50-epoch training schedules. Due to the space limit, we put the results of 50-epoch in the appendix. [Zheng *et al.*, 2021] proposes an inheriting strategy to initialize the student model with the teacher's neck and head parameters, leading to improved performance. We employ this strategy to initialize the transformer encoder and decoder of the student model with the teacher's parameters[4].

Results with a standard $1\times$ schedule. The results are presented in Table 1. All the student detectors obtain significant mAP improvements with the knowledge transferred from teacher detectors. For instance, D^{3}ETR boosts detection performance when applied to Conditional DETR: +8.3 mAP for R50-C5, +9.8 mAP for R18-C5, +6.3 mAP for R50-DC5, and +7.6 mAP for R18-DC5. Rresults of more detectors and training schedules could be found in appendix.

5.3 Ablation Study

In this section, we first compare the proposed decoder distillation method to other CNN-based distillation methods in object detection. Subsequently, we conduct ablation studies to verify each component in our decoder distillation strategies. We adopt Conditional DETR-R101-C5 as the teacher and Conditional DETR-R50-C5 as the student. The student

[4] The inheriting strategy resulted in a performance gain of 2.1/0.4 mAP for Conditional DETR-R50-C5 under 12/50 epochs setting.

Teacher	Student	Backbone	mAP	AP$_s$	AP$_m$	AP$_l$
DETR R101-C5 43.5 (500e)	DETR	R50-C5	25.1	7.7	24.9	43.1
	+ Ours		32.6 (**+7.5**)	9.7	32.0	49.9
	DETR	R18-C5	19.7	4.5	18.8	34.8
	+ Ours		28.3 (**+8.6**)	7.7	29.3	48.1
DETR R101-DC5 44.7 (500e)	DETR	R50-DC5	28.4	9.7	29.3	46.9
	+ Ours		39.3 (**+10.9**)	17.1	43.1	59.2
	DETR	R18-DC5	23.0	6.4	22.9	39.4
	+ Ours		33.0 (**+10.0**)	12.4	34.7	52.8
Conditional DETR R101-C5 42.8 (50e)	Conditional DETR	R50-C5	32.4	14.7	35.0	48.3
	+ Ours		40.7 (**+8.3**)	19.3	43.5	59.7
	Conditional DETR	R18-C5	26.3	10.2	28.2	39.7
	+ Ours		36.1 (**+9.8**)	15.4	38.1	54.2
Conditional DETR R101-DC5 45.0 (50e)	Conditional DETR	R50-DC5	36.5	17.6	40.0	52.6
	+ Ours		42.8 (**+6.3**)	22.4	45.8	60.3
	Conditional DETR	R18-DC5	29.9	13.4	32.5	43.3
	+ Ours		37.5 (**+7.6**)	17.1	39.8	54.2

Table 1: **Results with a 12-epoch training schedule on MS COCO.** Our proposed approach exhibits significant improvements over two DETR-based methods. We employ the teacher model to initialize both the encoder and decoder parameters.

Method	#Epochs	mAP
Conditional DETR-R101-C5 (⋆)	50	42.8
Conditional DETR-R50-C5 (♣)	12	32.4
♣ + DeFeat	12	32.4
♣ + FitNet	12	33.3
♣ + FGD	12	36.0
♣ + MGD	12	36.7
♣ + DETRDistill	12	37.7
♣ + Ours	12	38.6
♣ + Ours + MGD	12	39.3
♣ + Ours + DETRDistill*	12	39.4

Table 2: **Comparison with other distillation methods.** We adopt Conditional DETR-R101-C5 as the teacher model and Conditional DETR-R50-C5 as the student. Our D^{3}ETR is superior to other distillation methods, and can be further improved by combining with MGD or DETRDistill. *: we only utilize the "Target-aware Feature Distillation" strategy in DETRDistill, which distills the encoder feature.

model is trained for 12 epochs without utilizing the inheriting strategy.

The effectiveness of decoder distillation. We compare D^{3}ETR with other state-of-the-art KD approaches for object detection, DeFeat, FitNet, FGD, and MGD, as shown in Table 2. These works focus on distilling ordered outputs, and we employ them to the output feature of the transformer encoder. We utilize identical teacher and student models, as well as training settings in each instance. For competing distillation approaches, we tune the hyper-parameters based on those in the corresponding papers or open-sourced code repositories, and take the best performance. As can be seen from the table, the proposed approach outperforms its counterparts. This highlights that it is more effective to perform distillation on the DETR decoder layers. We also compared the proposed method with DETRDistill, which performs distillation on both the encoder and decoder. The superior result validates the effectiveness and importance of distilling the attention map of the decoder and addressing the instability issue in teacher-student matching. Besides, we integrate the proposed method with MGD or DETRDistill, resulting in enhanced performance. This exemplifies the potential of further improving the performance of our proposed approach. However, it is beyond the scope of this paper and is left for future work.

The effect of each component in our method. We gradually incorporate the proposed strategies into the baseline, and the results are presented in Table 3. If we do not use any teacher-student matching strategy, the result is 33.9. This shows that it is important to establish a reasonable teacher-student query correspondence. By adopting the Adaptive Matching strategy with distillation in prediction, self-attention, and cross-attention, the results improve significantly, and the combination of these three approaches provides the best result (37.2). Subsequently, adding Fixed Matching improves the result to 38.6. Finally, the inheritance strategy boosts the result by 2.1, obtaining 40.7 mAP, which is comparable to Conditional DETR-R50-C5 trained with 50 epochs (40.9).

Teacher-student matching strategy. The proposed MixMatcher comprises two teacher-student matching strategies, namely, Adaptive Matching and Fixed Matching. To validate our design, we perform ablations on the matching strategy and report the results in Table 4. Our findings indicate that either using Adaptive Matching or Fixed Matching can improve the baseline. Adaptative Matching is slightly more efficient and achieves 37.2 mAP. We further find that we can improve the performance when we use two groups of queries but adopt the same matching strategy during training. For instance, the result obtained by two groups with Adaptive Matching is 0.5 higher than that obtained using a single group. We postulate that multiple query groups enable each ground truth to match more positive queries, thereby easing training [Chen *et al.*, 2022a]. Moreover, using these two strategies simultaneously generates the best result of 38.6 mAP. This illustrates that the hybrid design is helpful.

Constraint in Fixed Matching. We introduce a constraint in Fixed Matching to strengthen the teacher-student fixed correspondence as illustrated in Eq. 3. Without this constraint, the output of the auxiliary group may be supervised by dif-

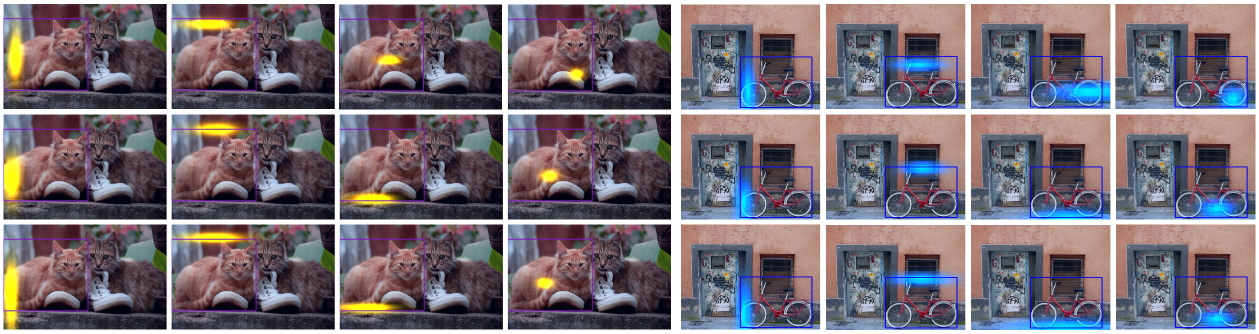

Figure 4: **Comparison of spatial attention maps**. The first row shows the student model, the second row shows the student model with our D^{3}ETR, and the third row shows the teacher model. We choose to visualize only 4 out of the 8 heads, with the remaining heads being duplicates. The ground-truth boxes are denoted by purple and blue boxes. Best viewed in color.

Adaptative Matching	Prediction	Self-Attn	Cross-Attn	Fixed Matching	Inheriting	mAP
						32.4
	✓	✓	✓			33.9
✓	✓					35.0
✓		✓	✓			36.3
✓	✓	✓				36.3
✓	✓		✓			36.5
✓	✓	✓	✓			37.2
✓	✓	✓	✓	✓		38.6
✓	✓	✓	✓	✓	✓	40.7

Table 3: **Ablation study on the proposed distillation strategies.** We use Conditional DETR-R101-C5 as the teacher model and Conditional DETR-R50-C5 as the student model.

Adaptative Matching	Fixed Matching	mAP
		32.4
✓		37.2
	✓	36.8
✓✓		37.7
	✓✓	36.9
✓	✓	38.6

Table 4: **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.

Method	Constraint	mAP
Fixed Matching	✗	37.3
Fixed Matching	0 ∼ 5-th layer	38.4
Fixed Matching	5-th layer	38.6

Table 5: **Ablation study on the constraint in Fixed Matching.** The best result is obtained by adding a constraint on the last decoder layer.

ferent ground truths from the corresponding output of the teacher model. Since the two outputs are generated from the same object query, the model may become confused. According to Table 5, the performance drops from 38.6 to 37.3 without constraints. This also indicates that directly using the teacher's query as an auxiliary group only leads to marginal improvements (+0.1). We additionally test adding constraints on all decoder layers and find the performance to be slightly worse than adding a constraint on the last layer.

5.4 Visualization

To validate whether the student model learns meaningful information from the teacher model, we visualize the spatial attention map [Meng *et al.*, 2021]. Figure 4 shows the results. According to [Meng *et al.*, 2021], the spatial attention maps correspond to the extremities of objects or a small region within the object box. The former helps in locating the object while the latter aids in recognizing its category. Our observation indicates that the student model struggles with identifying the object extremities precisely. However, by employing our proposed distillation framework, D^{3}ETR, 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 work, we explore the effectiveness of knowledge distillation for detectors based on DETR architecture. Our approach, MixMatcher, facilitates learning the correspondence between DETR-based teacher and student models, enabling an efficient knowledge transfer. Based on MixMatcher, we propose D^{3}ETR, a straightforward yet effective distillation framework, and demonstrate its effectiveness through extensive experiments.

Acknowledgments

This work is supported by the Sichuan Science and Technology Program (2023YFSY0008), China Tower-Peking University Joint Laboratory of Intelligent Society and Space Governance, National Natural Science Foundation of China (61632003, 61375022, 61403005), Grant SCITLAB-20017 of Intelligent Terminal Key Laboratory of SiChuan Province, Beijing Advanced Innovation Center for Intelligent Robots and Systems (2018IRS11), and PEKSenseTime Joint Laboratory of Machine Vision.

Contribution Statement

Xiaokang Chen and Jiahui Chen make core contributions. Xiaokang formalized the idea and led the project.

References

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

[Chen *et al.*, 2017] 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.

[Chen *et al.*, 2021] 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.

[Chen *et al.*, 2022a] 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.

[Chen *et al.*, 2022b] Qiang Chen, Jian Wang, Chuchu Han, Shan Zhang, Zexian Li, Xiaokang Chen, Jiahui Chen, Xiaodi 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.

[Chen *et al.*, 2022c] Xiaokang Chen, Fangyun Wei, Gang Zeng, and Jingdong Wang. Conditional detr v2: Efficient detection transformer with box queries. *arXiv preprint arXiv:2207.08914*, 2022.

[Cho and Hariharan, 2019] 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.

[Dai *et al.*, 2021] 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.

[Dong *et al.*, 2022] 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.

[Glorot and Bengio, 2010] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

[Guo *et al.*, 2021] Jianyuan 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.

[Hao *et al.*, 2020] 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.

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

[Hinton *et al.*, 2015] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *arXiv: Machine Learning*, 2015.

[Jia *et al.*, 2022] 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.

[Jiahao *et al.*, 2022] 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.

[Kuhn, 1955] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 1955.

[Li *et al.*, 2017] 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.

[Li *et al.*, 2022] 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.

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

[Lin *et al.*, 2014b] 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.

[Liu *et al.*, 2022] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022.

[Loshchilov and Hutter, 2017] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. In *ICLR*, 2017.

[Meng *et al.*, 2021] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *ICCV*, pages 3651–3660, 2021.

[Park *et al.*, 2019] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[Rezatofighi *et al.*, 2019] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian D. Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *CVPR*, 2019.

[Romero *et al.*, 2014] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

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

[Simonyan and Zisserman, 2015] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[Song *et al.*, 2022] Hwanjun Song, Deqing Sun, Sanghyuk Chun, Varun Jampani, Dongyoon Han, Byeongho Heo, Wonjae Kim, and Ming-Hsuan Yang. Vidt: An efficient and effective fully transformer-based object detector. In *ICLR*, 2022.

[Tian *et al.*, 2019] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[Tung and Mori, 2019] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[Wang *et al.*, 2022] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *AAAI*, 2022.

[Yang *et al.*, 2021] Jing Yang, Brais Martinez, Adrian Bulat, Georgios Tzimiropoulos, et al. Knowledge distillation via softmax regression representation learning. International Conference on Learning Representations (ICLR), 2021.

[Yang *et al.*, 2022a] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4643–4652, 2022.

[Yang *et al.*, 2022b] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022.

[Yim *et al.*, 2017] Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4133–4141, 2017.

[Zagoruyko and Komodakis, 2016] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[Zhang and Ma, 2020] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *International Conference on Learning Representations*, 2020.

[Zhang *et al.*, 2018] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328, 2018.

[Zhang *et al.*, 2022a] Gongjie Zhang, Zhipeng Luo, Yingchen Yu, Kaiwen Cui, and Shijian Lu. Accelerating detr convergence via semantic-aligned matching. In *CVPR*, pages 949–958, 2022.

[Zhang *et al.*, 2022b] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022.

[Zhang *et al.*, 2022c] Peizhen Zhang, Zijian Kang, Tong Yang, Xiangyu Zhang, Nanning Zheng, and Jian Sun. Lgd: label-guided self-distillation for object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3309–3317, 2022.

[Zhao *et al.*, 2022] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11953–11962, 2022.

[Zheng *et al.*, 2021] Nanning Zheng, Jian Sun, Xiangyu Zhang, Zijian Kang, and Peizhen Zhang. Instance-conditional knowledge distillation for object detection. 2021.

[Zhou *et al.*, 2021] Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias-variance tradeoff perspective. *arXiv preprint arXiv:2102.00650*, 2021.

[Zhu *et al.*, 2020] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: deformable transformers for end-to-end object detection. *CoRR*, abs/2010.04159, 2020.