DETRDistill: A Universal Knowledge Distillation Framework for DETR-families

Jiahao Chang[1*] Shuo Wang[1*] Hai-Ming Xu[2*] Zehui Chen[1] Chenhongyi Yang[3] Feng Zhao[1†]
[1]University of Science and Technology of China [2]University of Adelaide [3]University of Edinburgh
{changjh,shuowang2323,lovesnow}@mail.ustc.edu.cn hai-ming.xu@adelaide.edu.au
chenhongyi.yang@ed.ac.uk fzhao956@ustc.edu.cn

Abstract

Transformer-based detectors (DETRs) are becoming popular for their simple framework, but the large model size and heavy time consumption hinder their deployment in the real world. While knowledge distillation (KD) can be an appealing technique to compress giant detectors into small ones for comparable detection performance and low inference cost. Since DETRs formulate object detection as a set prediction problem, existing KD methods designed for classic convolution-based detectors may not be directly applicable. In this paper, we propose DETRDistill, a novel knowledge distillation method dedicated to DETR-families. Specifically, we first design a Hungarian-matching logits distillation to encourage the student model to have the exact predictions as those of the teacher DETRs. Then, we propose a target-aware feature distillation to help the student model learn from the object-centric features of the teacher model. Finally, in order to improve the convergence rate of the student DETR, we introduce a query-prior assignment distillation to speed up the student model learning from well-trained queries and stable assignment of the teacher model. Extensive experimental results on the COCO dataset validate the effectiveness of our approach. Notably, DETRDistill consistently improves various DETRs by more than 2.0 mAP, even surpassing their teacher models.

1. Introduction

Object detection aims to locate and classify visual objects from an input image. In the early works, the task was typically achieved by incorporating convolution neural networks (CNNs) to process the regional features of the input image [26, 19], in which a bunch of inductive biases were included, such as anchors [26], label assignment [41] and duplicate removal [1]. Recently, transformer-based object detectors like DETR [2] have been proposed where detection is treated as a set prediction task which significantly simplifies the detection pipeline and helps the users free from the tedious tuning of the hand-craft components, e.g., anchor sizes and ratios [26].

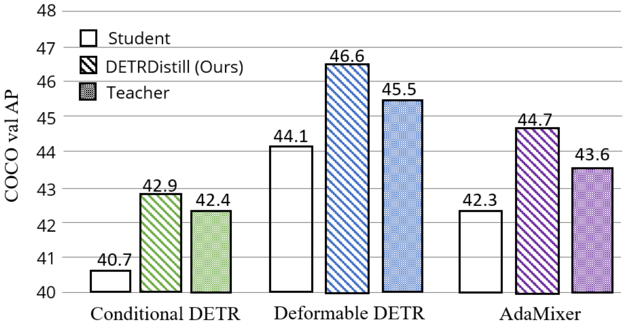

Figure 1. The performance of our DETRDistill on three transformer-based detectors: Conditional DETR [25], Deformable DETR [43], and AdaMixer [11]. We adopt ResNet-101 and ResNet-50 for the teacher and student models, respectively. Our DETRDistill yields significant improvements compared to its student baseline, and even outperforms its teacher model.

Although the transformer-based detectors have achieved state-of-the-art performance [21, 17, 11], they suffer from an expensive computation problem, making them difficult to be deployed in real-time applications. In order to acquire a fast and accurate detector, *knowledge distillation* [14] (KD) is an appealing technique. Normally, KD methods transfer knowledge from a heavy-weighted but powerful teacher model to a small and efficient student network by mimicking the predictions [14] or feature distributions [27].

In the research area of object detection, there have various kinds of KD methods been published [37, 36, 39, 27, 4, 18, 12, 42, 38, 32, 29]. However, most of these methods are designed for convolution-based detectors and may not directly apply to transformer-based DETRs due to the detection framework differences. There are at least two challenges: ❶ *logits-level distillation methods* [14, 42] *are unusable for DETRs.* For either anchor-based [41] or anchor-free [31] convolution-based detectors, box predictions are

*The authors contributed equally.
†Corresponding author.

Table 1. Comparison on several CNN-based region weighted feature distillation approaches on AdaMixer [11].

Method	#Epoch4 AP	#Epoch8 AP	#Epoch12 AP
Baseline w/o KD	35.0	38.7	42.3
FGD [37]	34.4(-0.6)	39.1(+0.4)	40.7(-1.6)
FKD [40]	35.9(+0.9)	39.5(+0.8)	42.2(-0.1)
MGD [38]	36.3(+1.3)	39.8(+1.1)	42.3(+0.0)
FGFI [34]	35.6(+0.6)	39.3(+0.6)	42.6(+0.3)
FitNet [27]	36.4(+1.4)	39.6(+0.9)	42.9(+0.6)

closely related to the feature map grid and thus naturally ensure a strict spatial correspondence of box predictions for knowledge distillation between teacher and student. However, for DETRs, box predictions generated from the decoders are unordered and there is no natural one-to-one correspondence of predicted boxes between teacher and student for a logits-level distillation. ❷ *Feature-level distillation approaches may not suitable for DETRs.* Due to the feature generation mechanism being different between convolution and transformer [27], the region of feature activation for the interested object varies a lot. As Fig. 2 shows, the active region of a convolution-based detector is almost restricted inside the ground-truth box while the DETR detector further activates regions in the background area. Therefore, directly using previous feature-level KD methods for DETRs may not necessarily bring performance gains and sometimes even impair the student detectors as presented in Table 1.

To address the above challenges, we propose ***DETRDistill***, a knowledge distillation framework specially designed for detectors of DETR families. Precisely, DETRDistill mainly consists of three components:

(1) Hungarian-matching logits distillation: To solve the challenge ❶, we use the Hungarian algorithm to find an optimal bipartite matching between predictions of student and teacher and then the KD can be performed in the logits-level. However, since the number of boxes predicted as positive in the teacher model is quite limited, doing KD only on positive predictions does not bring a significant performance gain. Instead, we propose to introduce a distillation loss on the massive negative predictions between the teacher and student model to fully make use of the knowledge lying in the teacher detector. Moreover, considering DETR methods typically contain multiple decoder layers for a cascade prediction refinement, we also create a KD loss at each stage to have a progressive distillation.

(2) Target-aware feature distillation: According to the analysis in challenge ❷, we propose utilizing object queries and teacher model features to produce soft activation masks. Since the well-trained teacher queries are closely related to various object targets, such generated soft masks will be object-centric and thus make soft-mask-based feature-level distillation to be target-aware.

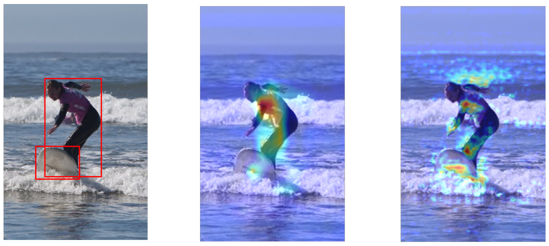
(a) Ground truth box region
(b) CNN active region
(c) Cross-attention active region

Figure 2. Visualization of the (a) ground-truth boxes and active region from (b) ATSS [41] and (c) AdaMixer [11].

(3) Query-prior assignment distillation: Since the queries and decoder parameters are randomly initialized in the student model, unstable bipartite assignment in the student model leads to a slow convergence rate as presented in [17]. While we empirically find that well-trained queries in the teacher model can always produce a consistent bipartite assignment as shown in Fig. 7, we thus propose to let the student model take teacher's queries as an additional group of prior queries and encourage it to produce predictions based on the stable bipartite assignment of the teacher network. Such a distillation successfully helps the student model to converge fast and achieve better performance.

In summary, our contributions are in three folds:

- We analyze in detail the difficulties encountered by DETRs in the distillation task compared with traditional convolution-based detectors.
- We propose multiple knowledge distillation methods for DETRs from the perspective of logits-level, feature-level, and convergence rate, respectively.
- We conduct extensive experiments on the COCO dataset under different settings, and the results prove the effectiveness and generalization of our proposed methods.

2. Related Work

2.1. Transformer-based Object Detectors

With the excellent performance of Transformer [33] in natural language processing, researchers have also started to explore the application of Transformer structure to visual tasks [23, 7, 24]. However, the DETR training process is extremely inefficient, so many follow-up works have attempted to accelerate convergence. One line of work tries to redesign the attention mechanism. For example, Dai *et al.* [43] propose Deformable DETR, which constructs a sparse attention mechanism by only interacting with the variable sampling point features around the reference points. SMCA [10] introduces Gaussian prior to limit cross-attention. AdaMixer [11] designs a new adap-

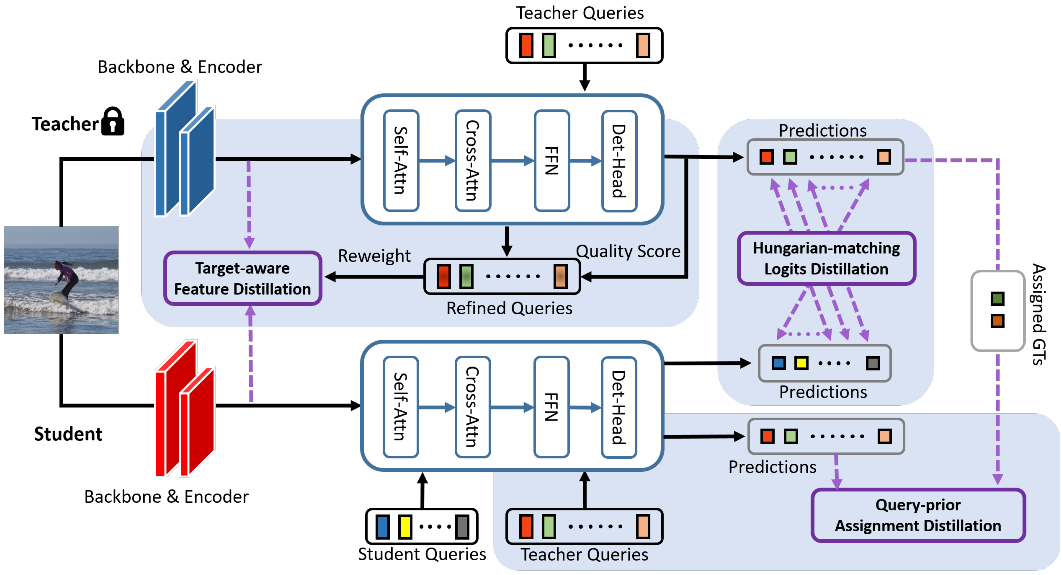

Figure 3. The overall architecture of our approach. It consists of a transformer-based teacher detector with a large backbone, a congener detector with a lightweight backbone, and the proposed distillation modules: (i) Hungarian-matching Logits Distillation (ii) Target-aware Feature Distillation, and (iii) Query-prior Assignment Distillation. We omit the original training supervision for clear formulation.

tive 3D feature sampling strategy without any encoder and then mixes sampled features in the channel and spatial dimension with adaptive weights.

Another line of work rethink the meaning of the query. Meng *et al*. [25] visualize that it is ineffective for DETR to rely on content embedding in the cross-attention to locate object extremity and therefore propose decoupling queries into the content part and position part. Anchor-DETR [35] directly treats the query's 2D reference points as its position embedding to guide attention. DAB-DETR [22] introduces width and height information besides location to the attention mechanism to model different scale objects. DN-DETR [17] introduces the query denoising task to accelerate the training. Group-DETR [6] and H-DETR [15] improve performance by increasing positive samples as auxiliary groups in the decoder training. Unlike the previous work, we expect to improve the performance of small models through distillation.

2.2. Knowledge Distillation in Object Detection

Knowledge distillation is a commonly used method for model compression. [14] first proposes this concept and applies it in image classification. They argue that soft labels output by the teacher contains "dark knowledge" of inter-category similarity compared to the one-hot encoding, which contributes to the model's generalization. Attention transfer [39] focuses the distillation on the feature map and transfers knowledge by narrowing the attention distribution of the teacher and student instead of distilling output logits.

FitNet [27] proposes mimicking the teacher model's intermediate-level hints by hidden layers. [4] first applies knowledge distillation to solve the multi-class object detection. [18] thinks that the background regions would introduce noise and proposes distilling the regions RPN sampled. DeFeat [12] distills the foreground and background separately. FGD [37] imitates the teacher regarding focal regions and global relations of features, respectively. LD [42] extends soft-label distillation to positional regression, causing the student to fit the teacher's border prediction distribution. MGD [38] uses masked image modeling (MIM) to transform the imitation task into a generation task. In addition to the above CNN-based distillation, some works involve vision transformers. DeiT [32] transfers inductive biases into ViT [9] from a CNN-teacher through a distillation token and achieves competitive performance in the classification task. ViDT [29] performs KD on patch tokens and proposes a variation of the transformer detector. However, such a distillation cannot be applied to the DETR-families directly. Our work analyzes the unique phenomena of different components of DETR and proposes a universal distillation strategy.

3. A Review of DETR

DETR [2] is an end-to-end object detector that includes a backbone, Transformer encoders, learnable query embeddings and decoders. Given an image I, the CNN backbone extracts its spatial features and then Transformer encoders (some variants do not require encoders [11]) will enhance the feature representation. With the updated features $F \in R^{HW\times d}$, query embeddings $Q \in R^{N\times d}$ are fed into several Transformer (typically six) decoders, where d is the feature dimension, N is the fixed number of queries. The operations in each decoder stage are similar: Firstly, exploiting self-attention to establish the relationship between queries to capture the mutual information. Secondly, interacting queries with image features via flexible cross-attention to aggregate queries with valuable semantic information. Thirdly, a feed-forward network (FFN) decodes each query into $\hat{y}_i = (\hat{c}_i, \hat{b}_i)$ including predicted categories and bounding boxes.

In the training stage, the principle of label assignment is to minimize the matching cost between model prediction and ground truth (GT) to get a bipartite matching with Hungarian algorithm [16]. The optimal matching is solved as follows:

$$\hat{\sigma} = arg\,min_{\sigma} \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma_i}), \tag{1}$$

where σ_i is a permutation of N elements and $\hat{\sigma}$ is the optimal assignment. $y_i = (c_i, b_i)$ is i-th GT where c_i is the target class (which may be $\varnothing$) and b_i is the GT box. $\mathcal{L}_{\text{match}}$ is a pair-wise matching cost:

$$\mathcal{L}_{\text{match}}(y_i, \hat{y}_{\sigma_i}) = \mathcal{L}_{\text{cls}}(c_i, \hat{c}_{\sigma_i}) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{bbox}}(b_i, \hat{b}_{\sigma_i}),$$

where $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{bbox}}$ denote the classification and bounding-box losses respectively. Therefore, each GT in the DETR will correspond to only one positive sample query, and all the rest queries are seen as negative samples. The final detection loss function is defined as:

$$\mathcal{L}_{\text{det}}(y, \hat{y}_{\hat{\sigma}}) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(y_i, \hat{y}_{\hat{\sigma}_i}), \tag{2}$$

where the location regression of negative samples is not subject to any supervision.

4. Our Approach: DETRDistill

In this section, we introduce the details of our proposed DETRDistill, which consists of three components: (1) Hungarian-matching Logits Distillation; (2) Target-aware Feature Distillation; (3) Query-prior Assignment Distillation. Fig. 3 shows the overall architecture of DETRDistill.

4.1. Hungarian-matching Logits Distillation

One of the most common strategies for knowledge distillation is to directly align the predictions at the logits-level between the two models. However, query-based predictions in a set form [17] make it difficult for DETRs to orderly correspond the teacher's results to that of the student. To achieve this goal, we reuse the Hungarian algorithm to match the predictions from the teacher to that of the student one-to-one.

Formally, let $\hat{y}^T$ and $\hat{y}^S$ denote the predictions from the teacher and student model, conforming to $\hat{y}^T = \{\{\hat{y}_i^{T\_pos}\}_{i=1}^{M^{pos}}, \{\hat{y}_j^{T\_neg}\}_{j=1}^{M^{neg}}\}$ and $\hat{y}^S = \{\hat{y}_i^S\}_{i=1}^N$, where M^{pos} and M^{neg} denote the number of positive and negative predictions of teacher model. $M = M^{pos} + M^{neg}$ and N are the total numbers of decoder queries for the teacher and student, respectively. M is usually greater than or equal to N. Since teacher's positive predictions are target closely related, one straightforward idea is to treat them as knowledgeable pseudo GTs and utilize the Hungarian algorithm to find a matching $\hat{\sigma}^{pos}$ between these positive predictions $\hat{y}^{T\_pos}$ and the ones of student $\hat{y}^S$. Then the logits-level KD can be achieved

$$\mathcal{L}_{\text{logitsKD}}^{pos}(\hat{y}^{T\_pos}, \hat{y}_{\hat{\sigma}^{pos}}^S) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\hat{y}_i^{T\_pos}, \hat{y}_{\hat{\sigma}_i^{pos}}^S). \tag{3}$$

However, we empirically find that such a naive KD only brings minor performance gains as presented in Table 8. We postulate that the number of positive predictions is quite limited (only 7 per image on average while the total number of queries usually exceeds 100) and the distilled information highly coincides with GTs. On the other hand, a large number of negative predictions of the teacher model are ignored and we argue that these predictions are valuable.

Negative location distillation. Since the teacher model is usually well-optimized, the generated positive predictions and the negative ones may have an obvious difference so that the Hungarian algorithm can produce a plausible assignment, *i.e.*, these negatively predicted boxes will stay off the object targets. While the randomly initialized student network may not have such an effect and the student's negative predictions may entangle with positive ones. Therefore, we propose to create a distillation to leverage the knowledge lying in the teacher's negative predictions.[*]

$$\mathcal{L}_{\text{logitsKD}}^{neg}(\hat{y}^{T\_neg}, \hat{y}_{\hat{\sigma}^{neg}}^S) = \sum_{i=1}^{N} \mathcal{L}_{\text{match}}(\hat{y}_i^{T\_neg}, \hat{y}_{\hat{\sigma}_i^{neg}}^S), \tag{4}$$

where $\hat{\sigma}^{neg}$ denotes the assignment between $\hat{y}^{T\_neg}$ and $\hat{y}^S$.

Progressive distillation. Considering DETR decoders typically contain multiple stages and stage-wise supervision is included by default [2], we further propose to introduce the KD losses in Eq. 3 and 4 into each decoder stage to have a progressive distillation

$$\mathcal{L}_{\text{logitsKD}} = \sum_{k=1}^{K} \mathcal{L}_{\text{logitsKD}}^{pos}[k] + \mathcal{L}_{\text{logitsKD}}^{neg}[k], \tag{5}$$

where K is the number of decoder stages. $\mathcal{L}_{\text{logitsKD}}^{pos}[k]$ and

[*] Since the classification in the teacher's negative predictions may not necessarily always be predicted as background, KD is only done on boxes.

$\mathcal{L}^{neg}_{\text{logitsKD}}[k]$ denote the positive KD loss and negative KD loss on k-th decoder stage respectively.

Please note that we transfer knowledge from the teacher model's stage-by-stage outputs to the student model's corresponding stage, but not simply using the teacher's last stage output to supervise all stages of the student model. It is because we think the teacher model will contain different knowledge at different stages as observed in the recent work [3] and the former distillation strategy can make full use of the knowledge in the teacher model and the empirical results in Table 9 verify our argument.

4.2. Target-aware Feature Distillation

Detection performance is highly determined by feature representations generated from the feature pyramid network (FPN), this can be attributed to their rich semantic information related to object targets. Therefore, we argue that distilling the teacher model's knowledge at the feature-level is necessary. The typical manner to mimic the spatial features of a teacher model can be calculated as

$$\mathcal{L}_{\text{featKD}} = \frac{1}{dHW} \left\| \psi \odot \left(\boldsymbol{F}^T - \phi(\boldsymbol{F}^S) \right) \right\|_2^2, \qquad (6)$$

where $\boldsymbol{F}^T \in R^{H \times W \times d}$ and $\boldsymbol{F}^S \in R^{H \times W \times d^S}$ denote feature representations produced by the teacher and student model, respectively. H and W denote the height and width of the feature and d is the channel number of teacher's feature. ϕ is a learnable dimension adaptation layer to transform the student's feature into d dimension. $\odot$ is the Hadamard product of two matrices. $\psi \in \mathbb{R}^{H \times W}$ denotes a soft mask for the selection of knowledgeable regions in various KD methods, *e.g.*, Romero *et al.* [27] treat the mask as a matrix filled with 1. Wang *et al.* [34] generate the mask based on the IoU scores between anchor boxes and GT ones. Sun *et al.* [30] utilize the Gaussian mask to cover the GT boxes. Different from the above approaches, we propose to construct the soft mask for DETRs by calculating the similarity matrix between query embeddings and feature representations. Formally, given a whole set of queries $\boldsymbol{Q}^T \in R^{M \times d}$ of the teacher model, the selection mask can be obtained

$$\psi = \frac{1}{M} \sum_{i=1}^{M} \psi_i, \;\; \text{where} \;\; \psi_i = \boldsymbol{F}^T \cdot \boldsymbol{Q}_i^T, \qquad (7)$$

where $\boldsymbol{Q}_i^T \in R^{1 \times d}$ is the i-th teacher's query and M denotes the number of decoder queries of the teacher's model.

However, we empirically find such a vanilla distillation approach works poorly as shown in Table 2. We postulate the reason is that not all object queries of the teacher should be equally treated as valuable cues. Based on the predictions generated from the teacher queries, Fig. 4 presents some visualizations of query-based masks $\{\psi_i\}$ and we can find that masks with low-prediction scores attend out of the object regions.

Table 2. Ablation on feature-level distillation.

Setting	AP	AP$_S$	AP$_M$	AP$_L$
Baseline w/o KD	42.3	25.3	44.8	58.2
KD w/ Eq. 7	41.9(-0.4)	24.4	44.9	57.4
KD w/ Eq. 9	**43.5**(+1.2)	**25.4**	**46.7**	**60.0**

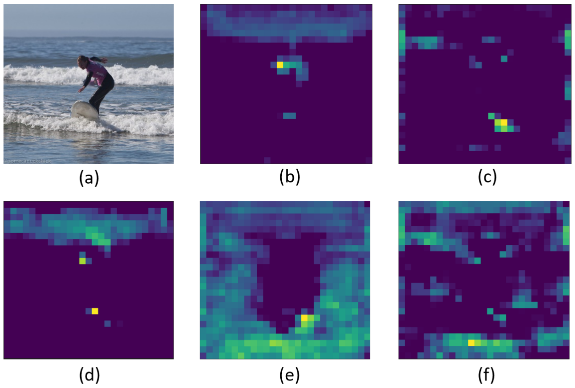

Figure 4. Visualization of the attention mask of queries. (a) The original image and (b-c) the attention masks generated by the queries responsible for the human and surfboard prediction, respectively. The query corresponding to (d) relates to both the human and the surfboard. (e) and (f) are the masks generated by the samples with low-quality scores.

According to the observation, we propose to utilize teacher queries selectively for generating the mask ψ. Specifically, we use the quality score proposed in [8] as the measurement

$$q_i = (c_i)^\gamma \cdot \text{IoU}\left(b_i^{GT}, b_i^{\text{pred}}\right)^{1-\gamma}, \qquad (8)$$

where c_i and b_i^{pred} denote the classification score and the predicted box from the i-th teacher's query, respectively. b_i^{GT} is the corresponding bipartite matched GT box. $\gamma = 0.5$ is a hyperparameter for balancing the weight of classification scores and box IoUs. Then the target-aware quality score serves as an indicator to guide which query should contribute more to the knowledge distillation and the KD loss in Eq. 6 can be extended as

$$\mathcal{L}_{\text{featKD}} = \sum_{i=1}^{M} \frac{q_i}{MdHW} \left\| \psi_i \odot \left(\boldsymbol{F}^T - \phi\left(\boldsymbol{F}^S\right) \right) \right\|_2^2. \qquad (9)$$

4.3. Query-prior Assignment Distillation

Since the queries and decoder parameters in DETR are normally randomly initialized for model optimization, a query may be assigned to different objects in different training epochs and thus lead to an unstable bipartite graph matching and a slow convergence speed [17]. In the setting of KD, the training of student DETR also suffers from the same problem. However, we empirically observe that the well-optimized queries in the teacher model can consistently achieve a stable bipartite assignment between different decoder stages as presented in Fig. 7 and it is intuitive to utilize the knowledge of the teacher model to improve the stability of the training of the student model. Based on this motivation, we propose Query-prior assignment distillation.

Specifically, given teacher query set $\boldsymbol{Q}^T$, and we can obtain the corresponding assignment permutation $\hat{\sigma}^T$ from the teacher for any given input-GT pairs. We propose to input the teacher query embeddings $\boldsymbol{Q}^T$ into the student model as an additional group of prior queries and directly use the teacher's assignment $\hat{\sigma}^T$ to generate detection results for loss calculation

$$\mathcal{L}_{\mathrm{assignKD}}(y, \hat{y}^S_{\hat{\sigma}^T_i}) = \sum_{i=1}^{M} \mathcal{L}_{\mathrm{match}}(y_i, \hat{y}^S_{\hat{\sigma}^T_i}). \tag{10}$$

This proposed KD loss will help the student model treat the teacher queries as a prior and encourage the student detector to achieve a stable assignment as much as possible. As shown in Fig. 7, the matching stability of the student model has been greatly improved with the proposed distillation loss. Please note that such an additional teacher query group is only used during training and the student model will use its default query set for final evaluation.

4.4. Overall Loss

To sum up, the total loss for training the student DETR is a weighted combination of Eq. 2, Eq. 5, Eq. 9, and Eq. 10:

$$\mathcal{L} = \mathcal{L}_{\mathrm{det}} + \lambda_1 \mathcal{L}_{\mathrm{logitsKD}} + \lambda_2 \mathcal{L}_{\mathrm{featKD}} + \lambda_3 \mathcal{L}_{\mathrm{assignKD}}.$$

where $\lambda_1 = 1, \lambda_2 = 20, \lambda_3 = 1$ are balancing weights of the proposed three KD loss terms. Since our distillation method follows the common DETR paradigm, it can be easily applied to a variety of detectors in DETR-families.

5. Experiments

5.1. Experiment Setup and Implementation Details

Dataset. The challenging large-scale MS COCO benchmark [20] is used in this study, where the $train2017$ (118K images) is utilized for training and $val2017$ (5K images) is used for validation. We use the standard COCO-style measurement, $i.e.$, average precision (mAP) for evaluation.

DETR models. Three different DETR detectors are evaluated, including Deformable DETR [43], Conditional DETR [25], and AdaMixer [11]. We choose these three models for their representative model framework and their excellent performance. In the ablation study, we choose AdaMixer as the baseline for ablation and analysis for its easy training and quick convergence.

Table 3. Results for distillation on identical number of encoder and decoder stages for the teacher and student networks.

Detector	Setting	Epoch	AP	AP_S	AP_M	AP_L
AdaMixer (100 Queries)	Teacher	12	43.6	25.4	46.8	60.7
	Student	12	42.3	25.3	44.8	58.2
	FGD [37]	12	40.7(-1.6)	23.4	43.3	55.8
	MGD [38]	12	42.3(+0.0)	24.5	45.0	58.9
	FitNet [27]	12	42.9(+0.6)	24.7	45.8	59.4
	Ours	12	**44.7**(+2.4)	**26.7**	**47.6**	**61.0**
Deformable DETR (300 Queries)	Teacher	50	45.5	27.5	48.7	60.3
	Student	50	44.1	27.0	47.4	58.3
	FGD [37]	50	44.1(+0.0)	25.9	47.7	58.8
	MGD [38]	50	44.0(-0.1)	25.9	47.3	58.6
	FitNet [27]	50	44.9(+0.8)	27.2	48.4	59.6
	Ours	50	**46.6**(+2.5)	**28.5**	**50.0**	**60.4**
Conditional DETR (300 Queries)	Teacher	50	42.4	22.6	46.0	61.2
	Student	50	40.7	20.3	43.8	60.0
	FGD [37]	50	40.4(-0.3)	19.7	43.8	59.5
	MGD [38]	50	41.2(+0.5)	20.6	44.6	60.5
	FitNet [27]	50	41.0(+0.3)	20.2	44.3	59.9
	Ours	50	**42.9**(+2.2)	**21.6**	**46.5**	**62.2**

Implementation details. Our codebase is built on MMDetection toolkit [5]. All models are trained on 8 NVIDIA V100 GPUs. Unless otherwise specified, we train the teacher model for 1× schedule (12 epochs) or 50 epochs using ResNet-101 [13] as the backbone with Adam optimizer and train the student model with the same learning schedule using ResNet-50 [13] as the backbone following each baseline's settings.

5.2. Main Results

In this section, we mainly present experimental results on several kinds of settings: (1) identical number of encoder and decoder stages for teacher and student; (2) various numbers of encoder and decoder stages for teacher and student. We further provide more experimental results on other settings to demonstrate the effectiveness of our approach, such as (3) distillation from larger transformer backbone, (4) distillation to lightweight backbones and (5) self-distillation.

Distillation on identical number of encoder and decoder stages. The results are presented in Table 3. We can find that FitNet [27] brings stable performance gains on all the DETR variants, while MGD [38] and FGD [37] cannot work well for Transformer-based detectors and even may cause degradation of the results. However, it is obvious that our approach significantly enhances the student's performance and surpasses all other methods by a large margin on various detectors. Specifically, DETRDistill gains 2.4 AP on the AdaMixer, 2.5 AP on the Deformable DETR, and 2.2 AP on the Conditional DETR, which validates the effectiveness of our approach.

Distillation on different number of encoder and decoder

Table 4. Results for distillation on different number of encoder and decoder stages for the teacher and student networks. The performance of the teacher model is 36.2 AP.

#Enc./Dec.	Setting	AP	AP_S	AP_M	AP_L
6/6	Student	33.0	13.8	35.1	50.9
	Ours	**38.1**(+5.1)	**18.2**	**41.0**	**58.2**
6/3	Student	30.3	12.7	32.5	46.8
	Ours	**36.6**(+6.3)	**15.2**	**39.8**	**56.0**
3/6	Student	33.0	13.9	35.9	49.0
	Ours	**37.7**(+4.7)	**17.4**	**40.8**	**57.3**
3/3	Student	29.9	12.8	32.2	45.6
	Ours	**35.7**(+5.8)	**14.4**	**38.7**	**53.8**

stages. Since the student model is usually smaller than the teacher model, there is no guarantee that both teacher and student have the same number of transformer encoder and decoder stages. Therefore, we also conduct experiments on the student model with a reduced number of encoders/decoders than that of teachers and explore the critical factors for them. For this experiment, we choose Conditional DETR with ResNet-101 as the teacher model and ResNet-50 as the student model. All models are trained with the 1× schedule (12 epochs). The default number of encoders and decoders in the teacher model is 6 and we decrease the number of encoders and decoders from 6 to 3 for the student model. For the default progressive distillation, we simply group the teacher's decoders and make them follow our progressive strategy as shown in Fig. 5 to handle the stage mismatch problem.[†]

The main results are shown in Table 4. It can be seen that decreasing the number of transformer encoders will not have a significant impact on the performance, while a reduction in the number of decoders leads to serious performance degradation. However, our proposed DETRDistill significantly compensates for the performance gap. For example, with only 3 encoder and decoder stages, our approach helps the student model achieve 35.7 AP which is superior to the student model with 6 encoder and decoder stages without KD.

Distillation from larger transformer backbone. We use the larger Swin-B [23] backbone and AdaMixer for verification and Table 6 shows that our method can still achieve better performance. It also proves that our distillation is robust to both CNN and transformer backbones. Besides, we observe that as the number of parameters in the teacher model further increases, the relative improvement brought by distillation to the student is relatively saturated.

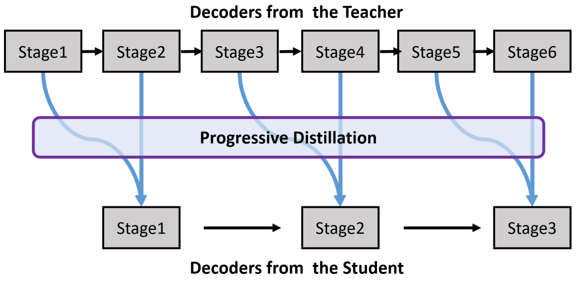

Figure 5. Distillation diagram with different number of decoders.

Table 5. Self-distillation performance on COCO validation set.

Detector	Setting	AP	AP_S	AP_M	AP_L
AdaMixer	T & S	42.3	25.3	44.8	58.2
	FGD	42.0(-0.3)	23.5	44.9	58.5
	MGD	41.7(-0.6)	23.0	44.7	58.6
	Ours	**43.6**(+1.3)	**26.7**	**46.2**	**59.1**
Deformable DETR	T & S	44.1	27.0	47.4	58.3
	FGD	44.2(+0.1)	26.3	47.8	58.5
	MGD	44.2(+0.1)	26.9	47.7	58.6
	Ours	**46.4**(+2.3)	**28.9**	**49.9**	**60.0**
Conditional DETR	T & S	40.7	20.3	43.8	60.0
	FGD	40.7(+0.0)	20.2	43.9	59.3
	MGD	41.2(+0.5)	20.6	44.6	60.5
	Ours	**42.5**(+1.8)	**21.7**	**46.2**	**61.4**

Table 6. Results of the large Swin-B backbone on AdaMixer.

Setting	Backbone	AP	AP_S	AP_M	AP_L
Teacher	Swin-B	47.5	29.9	50.1	64.8
Student	R-18	38.3	20.9	40.2	53.9
Ours	-	**41.0**(+2.7)	**23.9**	**45.2**	**56.6**
Teacher	Swin-B	47.5	29.9	50.1	64.8
Student	R-50	42.3	25.3	44.8	58.2
Ours	-	**45.4**(+3.1)	**27.3**	**48.6**	**61.8**
T&S	Swin-B	47.5	29.9	50.1	64.8
Ours	-	**48.7**(+1.2)	**30.1**	**51.7**	**65.1**

Distillation on lightweight backbones. In this section, we further conduct distillation for students with a lightweight backbone. Specifically, we apply DETRDistill for the student model with small backbones, e.g., ResNet-18 [13] and MobileNetV2 [28] from the teacher model with a ResNet-101 backbone [13]. The results are shown in Table 10. It is clear that our proposed distillation method has achieved the best performance across various settings. For example, we achieve 2.0/2.6 mAP improvements on AdaMixer, 3.3/3.5 mAP on Deformable DETR, and 2.1/2.7 mAP on Conditional DETR, respectively.

Self-Distillation. Self-distillation [36] is a special case of

[†] Please note that Target-aware Feature Distillation presented in Sec. 4.2 is performed on the output of transformer encoders, which with the same resolutions as the input. Thus not affected by the change of encoder stages.

Table 7. Ablation study of each component in DETRDistill. LD stands for Hungarian-matching Logits Distillation, FD represents Target-aware Feature Distillation, and AD denotes Query-prior Assignment Distillation.

Distillation	AP	AP_S	AP_M	AP_L
None	42.3	25.3	44.8	58.2
LD	43.7(**+1.4**)	25.3	46.5	60.7
FD	43.5(**+1.2**)	25.4	46.7	60.0
AD	42.9(**+0.6**)	24.5	45.9	59.3
LD + FD	44.3(**+2.0**)	25.8	47.0	61.0
LD + FD + AD	44.7(**+2.4**)	26.7	47.6	61.0

knowledge distillation where the teacher and student models are exactly identical, with the only aim of improving the model's performance. For this experimental setting, we use ResNet-50 as the backbone network and we compare the self-distillation performance of our approach with FGD [37] and MGD [38] methods. Table 5 shows that our DETRDistill gains 1.3 mAP, 2.3 mAP, and 1.8 mAP over the baselines. In contrast, FGD and MGD may hardly bring performance improvement and sometimes even cause a decline in detection, such as on the AdaMixer detector.

6. Ablation Studies

In this section, we are interested in ablating our approach from the following perspectives.

Effects of each component. To study the impact of each component in DETRDistill, we report the performance of each module in Table 7. Our baseline starts from 42.3 AP. When logits-level distillation, feature-level distillation, and query-prior assignment distillation are applied separately, we can obtain the gain of 1.4 AP, 1.2 AP, and 0.6 AP, respectively. Finally, the AP performance achieves 44.7 when all three modules are applied together, gaining a 2.4 AP absolute improvement.

Selection of logits distillation. Our default logits distillation is performed on positive classification predictions, positive and negative box regressions. We ablate various selections of these terms in Table 8. We can find that only distilling on positive predictions can only bring minor performance gains. While the distillation on negative box regression significantly improves the performance. This clearly verifies the importance of the proposed negative location distillation proposed in Sec. 4.1.

Necessary of progressive distillation. As presented in Sec. 4.1, we choose to progressively distill the knowledge stage-by-stage from the teacher model to the student model, but not to only use the last stage output of the teacher model. Table 9 shows that the former choice performs better.

Table 8. Ablations on the logits selection for distillation.

Pos. Cls.	Pos. Reg.	Neg. Reg.	AP	AP_S	AP_M	AP_L
-	-	-	42.3	25.3	44.8	58.2
✓			42.6	24.9	45.4	58.9
	✓		42.6	24.4	45.6	58.8
		✓	43.4	25.2	46.2	60.1
✓	✓		42.5	24.9	45.4	58.3
	✓	✓	43.6	25.2	46.3	60.6
✓	✓	✓	**43.7**	**25.3**	**46.5**	**60.7**

Table 9. Ablation study on the necessity of progressive distillation. Last-stage distillation means only using the teacher's last decoder stage output to supervise all decoder stages of the student model.

Strategy	AP	AP_S	AP_M	AP_L
Baseline	42.3	25.3	44.8	58.2
Last-stage Distillation	43.1	**25.4**	45.9	59.7
Progressive Distillation	**43.7**	25.3	**46.5**	**60.7**

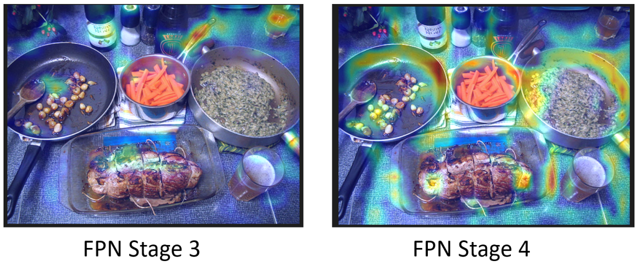

Figure 6. Visualization of selection mask on different FPN stages.

Visualization of selection mask in feature-level distillation. In Sec. 4.2, we have visualized some query-based selection mask ψ_i in Fig. 4. Here, we are also interested in the quality weighted mask $\sum_i^M (q_i \cdot \psi_i)$ and Fig. 6 presents two visualizations of this mask on different FPN stages. It can be seen that the soft masks are activated near the object targets and retain certain contextual information, which is in line with our expectations. We also find that the stages with different strides in FPN have different knowledgeable regions from the mask. Moreover, it inspires us not to use ground truth for selection mask generation but to use an immediate query-based selection mask for knowledge distillation as proposed in Sec. 4.2.

Visualization of instability between predictions from different decoder stages. In Sec. 4.3, we have proposed a Query-prior Assignment Distillation to speed and stabilize the training of the student model. Here, we present an analysis of the instability between predictions from different decoder stages to investigate the effectiveness of the proposed distillation module. We utilize the same metric proposed by [17] to evaluate bipartite assignment's instability. However, our focus is on the stability of match-

Table 10. Experimental results of DETRDistill on lightweight backbones: ResNet-18 and MobileNetV2 on COCO validation set.

Detector	Setting	Query	Backbone	Epoch	AP	AP$_{50}$	AP$_{75}$	AP$_S$	AP$_M$	AP$_L$
AdaMixer	Teacher	100	ResNet-101	12	43.6	62.8	47.2	25.4	46.8	60.7
	Student	100	ResNet-18	12	38.3	56.9	40.9	20.9	40.2	53.9
	Ours	-	-	-	**40.3**(+2.0)	**58.0**	**43.3**	**22.8**	**42.4**	**56.9**
	Teacher	100	ResNet-101	12	43.6	62.8	47.2	25.4	46.8	60.7
	Student	100	MobileNetV2	12	36.6	55.2	38.9	20.4	38.5	51.4
	Ours	-	-	-	**39.2**(+2.6)	**56.7**	**42.2**	**22.0**	**41.5**	**55.4**
Deformable DETR	Teacher	300	ResNet-101	50	45.5	64.8	49.5	27.5	48.7	60.3
	Student	300	ResNet-18	50	40.0	58.0	43.3	23.0	42.9	53.7
	Ours	-	-	-	**43.3**(+3.3)	**61.3**	**47.2**	**25.0**	**46.1**	**57.1**
	Teacher	300	ResNet-101	50	45.5	64.8	49.5	27.5	48.7	60.3
	Student	300	MobileNetV2	50	38.8	56.8	42.0	23.2	41.7	51.9
	Ours	-	-	-	**42.3**(+3.5)	**60.4**	**46.0**	**23.5**	**45.5**	**56.2**
Conditional DETR	Teacher	300	ResNet-101	50	42.4	63.4	44.8	22.6	46.0	61.2
	Student	300	ResNet-18	50	35.8	55.8	37.5	15.5	38.3	55.1
	Ours	-	-	-	**37.9**(+2.1)	**58.3**	**39.9**	**17.4**	**40.6**	**57.9**
	Teacher	300	ResNet-101	50	42.4	63.4	44.8	22.6	46.0	61.2
	Student	300	MobileNetV2	50	33.7	54.0	34.8	13.3	36.4	53.0
	Ours	-	-	-	**36.4**(+2.7)	**56.7**	**38.0**	**15.2**	**39.5**	**55.7**

ing between different decoder stages, not between different epochs. The predictions in the k-th decoder stage are denoted as $\mathbf{O^k} = \{O_0^k, O_1^k, \ldots, O_{N-1}^k\}$, where N is the number of predicted objects, and the GTs are denoted as $\mathbf{T} = \{T_0, T_1, \ldots, T_{M-1}\}$ where M is the number of ground truth targets. Then we compute the index vector $\mathbf{V^k} = \{V_0^k, V_1^k, \ldots, V_{N-1}^k\}$ to store the assignment result of k-th decoder stage as

$$V_n^k = \begin{cases} m, & \text{if } O_n^k \text{ matches } T_m, \\ -1, & \text{if } O_n^k \text{ matches nothing }. \end{cases} \tag{11}$$

And the instability (IS) between decoder stage k and stage $k+1$ can be calculated as:

$$IS^k = \sum_{j=0}^{N} \mathbb{1}(V_n^k \neq V_n^{k+1}). \tag{12}$$

We calculate the IS metric on the well-trained teacher model and the student baseline model. As for the model with our Query-prior Assignment Distillation, we compute the IS metric on the immediate checkpoint of the first training epoch. As Fig. 7 shows, the naive student has higher instability than the teacher model. But the IS score decreases a lot when using the proposed distillation module for training only one epoch, which clearly verifies that such a distillation module can help the student model train more stably and thus speed up the convergence rate.

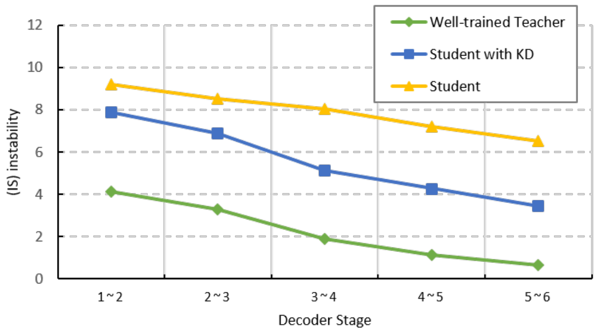

Figure 7. The instability (IS) of teacher, student, and student with our Query-prior Assignment Distillation after training one epoch.

7. Conclusion

This paper introduces a universal knowledge distillation framework for DETR-style detectors, named DETRDistill. Our method contains three distillation modules: Hungarian-matching Logits Distillation, Target-aware Feature Distillation, and Query-prior Assignment Distillation. Extensive experiments on the competitive COCO benchmark demonstrate the effectiveness and generalization of our approach. We hope that DETRDistill can serve as a solid baseline for DETR-based knowledge distillation in future research.

8. Acknowledgments

This work was supported by the JKW Research Funds under Grant 20-163-14-LZ-001-004-01, and the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

[1] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-NMS–improving object detection with one line of code. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5561–5569, 2017. 1

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Proceedings of the European Conference on Computer Vision*, pages 213–229, 2020. 1, 3, 4

[3] Fangyi Chen, Han Zhang, Kai Hu, Yu-kai Huang, Chenchen Zhu, and Marios Savvides. Enhanced training of query-based object detection via selective query recollection. *arXiv preprint arXiv:2212.07593*, 2022. 5

[4] 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

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. MMDetection: OpenMMlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 6

[6] Qiang Chen, Xiaokang Chen, Gang Zeng, and Jingdong Wang. Group detr: Fast training convergence with decoupled one-to-many label assignment. *arXiv preprint arXiv:2207.13085*, 2022. 3

[7] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 589–598, 2021. 2

[8] Zehui Chen, Chenhongyi Yang, Qiaofei Li, Feng Zhao, Zheng-Jun Zha, and Feng Wu. Disentangle your dense object detector. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4939–4948, 2021. 5

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 3

[10] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, and Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3621–3630, 2021. 2

[11] Ziteng Gao, Limin Wang, Bing Han, and Sheng Guo. Adamixer: A fast-converging query-based object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5364–5373, 2022. 1, 2, 3, 6

[12] 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. 1, 3

[13] 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. 6, 7

[14] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 1, 3

[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. 3

[16] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 4

[17] Feng Li, Hao Zhang, Shilong Liu, Jian Guo, Lionel M Ni, and Lei Zhang. Dn-detr: Accelerate detr training by introducing query denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13619–13627, 2022. 1, 2, 3, 4, 6, 8

[18] 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. 1, 3

[19] 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. 1

[20] 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 *Proceedings of the European Conference on Computer Vision*, pages 740–755. Springer, 2014. 6

[21] 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. In *Proceedings of the International Conference on Learning Representations*, 2022. 1

[22] 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. 3

[23] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 2, 7

[24] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12042–12051, 2022. 2

[25] Depu Meng, Xiaokang Chen, Zejia Fan, Gang Zeng, Houqiang Li, Yuhui Yuan, Lei Sun, and Jingdong Wang. Conditional detr for fast training convergence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3651–3660, 2021. 1, 3, 6

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region

proposal networks. *Advances in Neural Information Processing Systems*, 28, 2015. 1

[27] 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. 1, 2, 3, 5, 6

[28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 7

[29] 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. *arXiv preprint arXiv:2110.03921*, 2021. 1, 3

[30] Ruoyu Sun, Fuhui Tang, Xiaopeng Zhang, Hongkai Xiong, and Qi Tian. Distilling object detectors with task adaptive regularization. *arXiv preprint arXiv:2006.13108*, 2020. 5

[31] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019. 1

[32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 1, 3

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 2

[34] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4933–4942, 2019. 2, 5

[35] Yingming Wang, Xiangyu Zhang, Tong Yang, and Jian Sun. Anchor detr: Query design for transformer-based detector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 2567–2575, 2022. 3

[36] Chenhongyi Yang, Mateusz Ochal, Amos Storkey, and Elliot J Crowley. Prediction-Guided Distillation for Dense Object Detection. In *Proceedings of the European Conference on Computer Vision*, pages 123–138, 2022. 1, 7

[37] 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. 1, 2, 3, 6, 8

[38] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022. 1, 2, 3, 6, 8

[39] 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. 1, 3

[40] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *Proceedings of the International Conference on Learning Representations*, 2020. 2

[41] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 9759–9768, 2020. 1, 2

[42] Zhaohui Zheng, Rongguang Ye, Ping Wang, Dongwei Ren, Wangmeng Zuo, Qibin Hou, and Ming-Ming Cheng. Localization distillation for dense object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9407–9416, 2022. 1, 3

[43] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 1, 2, 6