HEAD: HEtero-Assists Distillation for Heterogeneous Object Detectors

Luting Wang[1,3] Xiaojie Li[2] Yue Liao[1,3⋆] Zeren Jiang[4]
Jianlong Wu[5] Fei Wang[2,6] Chen Qian[2] Si Liu[1,3]

[1]Institute of Artificial Intelligence, Beihang University [2]SenseTime Research
[3]Hangzhou Innovation Institute, Beihang University [4]ETH Zurich
[5]Shandong University [6]University of Science and Technology of China
https://github.com/LutingWang/HEAD

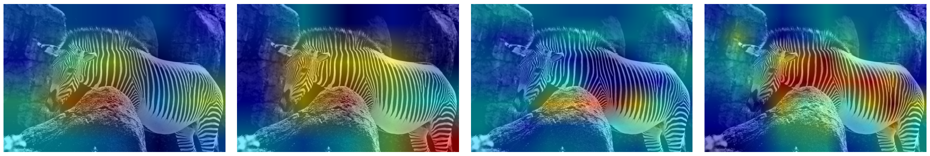
Abstract. Conventional knowledge distillation (KD) methods for object detection mainly concentrate on homogeneous teacher-student detectors. However, the design of a lightweight detector for deployment is often significantly different from a high-capacity detector. Thus, we investigate KD among heterogeneous teacher-student pairs for a wide application. We observe that the core difficulty for heterogeneous KD (hetero-KD) is the significant semantic gap between the backbone features of heterogeneous detectors due to the different optimization manners. Conventional homogeneous KD (homo-KD) methods suffer from such a gap and are hard to directly obtain satisfactory performance for hetero-KD. In this paper, we propose the HEtero-Assists Distillation (HEAD) framework, leveraging heterogeneous detection heads as assistants to guide the optimization of the student detector to reduce this gap. In HEAD, the assistant is an additional detection head with the architecture homogeneous to the teacher head attached to the student backbone. Thus, a hetero-KD is transformed into a homo-KD, allowing efficient knowledge transfer from the teacher to the student. Moreover, we extend HEAD into a Teacher-Free HEAD (TF-HEAD) framework when a well-trained teacher detector is unavailable. Our method has achieved significant improvement compared to current detection KD methods. For example, on the MS-COCO dataset, TF-HEAD helps R18 RetinaNet achieve 33.9 mAP (+2.2), while HEAD further pushes the limit to 36.2 mAP (+4.5).

Keywords: Knowledge Distillation, Object Detection, Heterogeneous.

1 Introduction

With the development of deep learning, the performance of object detection has achieved tremendous improvement. However, deploying detectors to edge devices often imposes constraints on the number of parameters, computation, and memory. Therefore, parameters compression and accuracy boosting are core problems for object detection towards practical application, where knowledge

[⋆] Corresponding author (liaoyue.ai@gmail.com)



(a) RetinaNet (b) FCOS (c) Faster R-CNN (d) TF-HEAD

Fig. 1: Comparison of the activation patterns from different detectors with the same backbone architecture. The intensity of the feature response increases from blue to red. These detectors produce different backbone feature representations. We use RetinaNet as the student and apply TF-HEAD to take advantage of the feature extraction abilities from an FCOS assistant and an R-CNN assistant. As a result, the activation map of TF-HEAD highlights more area of the zebra with higher intensity, indicating that the feature map contains the most information.

distillation (KD) is one of the most popular solutions. KD aims at training the compact model (student) by transferring knowledge from a high-capacity model (teacher). Recently, with the development of KD methods [17, 24, 33, 39, 49, 53] in general vision models, KD in object detection has raised increasing attention.

For a clear presentation, we first give a brief definition for the general architecture of modern CNN based detectors [28, 38, 41], where input images are represented as feature maps and then different methods are used to decode detection results from the feature maps. In this way, a detector can be divided into a backbone including FPN [27] and a detection head. We further define detectors with the same head architecture as homogeneous, otherwise heterogeneous.

Based on this definition, we summarize object detection KD into two schemes based on the architecture of teacher-student pair: homogeneous KD (homo-KD) or heterogeneous KD (hetero-KD). The homo-KD scheme usually allows the teacher detector to have a stronger backbone than the student detector but requires same head architectures. For instance, R18 RetinaNet [28] can be taught by R50 RetinaNet by homo-KD, but not R50 RepPoints [46] or R50 Faster R-CNN [38]. Significant progress has been made in the homo-KD [3, 6, 11, 39, 43, 47, 54]. However, the practical application of homo-KD is limited because the student for deployment and the most powerful teacher is usually designed from different motivations and produce very different heads. Therefore, we aim to explore hetero-KD, an essential and significant topic for object detection.

We first present an analysis for heterogeneous detectors. We observe that two heterogeneous detectors share the same backbone architecture, while their backbone features representation are still very distinct. As shown in fig. 1, activation maps from different detectors with R50 backbone, *e.g.* Faster R-CNN [38], RetinaNet [28], and FCOS [41], are different. Thus, we argue that the heterogeneous detection heads guide the backbone for different knowledge. We consider it a significant step for hetero-KD that the student mimics the teacher backbone knowledge. The intuitive idea is to perform homo-KD methods for heterogeneous detectors directly, but the accuracy improvement of the student is significantly limited. It is mainly because the backbone knowledge discrepancy enlarges the semantic gaps between the teacher and student layers.

To this end, we design a simple yet effective hetero-KD mechanism, namely HEtero-Assists Distillation (HEAD), to bridge the semantic gap between heterogeneous detectors via an adaptive assistant, thus simplifying to a homo-KD problem. For a specific teacher-student pair, we first design an additional assistant head with identical architecture with the teacher head and attach it to the student backbone. In this way, we construct a homogeneous detectors pair, *i.e.* the teacher and the student backbone equipped with the assistant. During training, the assistant and original student head process the student backbone features in parallel. Then, we propose two KD mechanisms, *i.e.* Assistant-based KD (AKD) and Cross-architecture KD (CKD), to supervise the assistant and student head learning, respectively. In AKD, we directly apply the homo-KD to the assistant and the teacher heads since they are homogeneous. Therefore, the high-level knowledge [49], *i.e.* the information flow for detection, is efficiently transferred from the teacher head to the assistant. Moreover, the teacher backbone knowledge is also transferred to the student backbone through gradient back-propagation from the assistant. Intuitively, the assistant teaches the student backbone to learn the critical knowledge reproducing the information flow [49] of the teacher. Thus, the semantic gap between heterogeneous detectors is bridged by the assistant. In CKD, we conduct a feature mimic from the student head to the teacher head. CKD plays an auxiliary role to integrate heterogeneous knowledge in the head level to compensate for AKD.

In practice, it is not always easy to obtain a suitable teacher for a specific student, limiting the application of traditional teacher-based KD methods [12,50]. Therefore, we further explore a Teacher-Free [21,23,50,52] method which accommodates our HEAD to these situations, namely TF-HEAD. TF-HEAD works by injecting diverse knowledge into the student, which helps the student to make more accurate predictions without extra computation cost at inference time. Specifically, we use the assistant to process the student backbone features, which is the same as HEAD. Since the assistant and the student heads are heterogeneous, they optimize the student backbone differently, thus enriching the knowledge inside. Although we train the assistant with ground truth labels, instead of supervision from the teacher head, the performance improvement brought by the assistant module is still significant. To push the limit further, TF-HEAD uses multiple assistants that are heterogeneous to each other.

Extensive experiments demonstrate the effectiveness of our framework. On MS-COCO dataset [29], our HEAD and TF-HEAD methods achieve state-of-the-art performance among teacher-based detection KD methods and teacher-free detection KD methods respectively. Using R50 Faster R-CNN as teacher, the mAP of R18 RetinaNet is increased from 31.7 to 36.2 (+4.5) and R18 FCOS from 32.5 to 36.0 (+3.5). Without pretrained teachers, TF-HEAD improves R18 RetinaNet from 31.7 to 33.9 (+2.2), which demonstrates that simply integrating heterogeneous knowledge helps to train better detectors, making hetero-KD advantageous compared to homo-KD.

2 Related Work

2.1 Object Detection

Object detection has three paradigms: two-stage [5, 10, 13, 14, 27, 38, 55], anchor-based one-stage [4, 28, 30, 37], and anchor-free one stage [7, 9, 41, 55, 56]. Two stage detectors use an Region Proposal Network (RPN) to generate Regions of Interest (RoIs) and then adopt a region-wise prediction network (R-CNN head) to predict objects. Although two-stage architectures obtain high accuracy, their complicated pipeline hinders deployment on edge devices. In contrast, one-stage detectors get the classification and the bounding box of targets based on features extracted by the backbone directly, achieving real-time inference. Anchor-based one-stage detectors use dense anchor boxes as proposals to detect targets. However, the number of anchor boxes is far more than targets, which brings much extra computation. Anchor-free detectors learn to predict keypoints and then generate bounding boxes to detect objects without the need for predefined anchors, reaching better performance with less cost. The features extracted by different detectors are optimized by different detection heads, which will result in large semantic gaps. Thus, It's hard to mitigate the difference by traditional homo-KD methods. In this work, we introduce the adaptive assistants to effectively bridge the gap between heterogeneous teacher-student pairs.

2.2 Knowledge Distillation

General KD. KD is a technology that helps training compact student models under the supervision of powerful teacher models. Hinton *et al.* [17] propose this concept and achieve great performance improvement by training the student with class distributions generated by the teacher. Extending Hinton's work, more works [1,8,16,19,22,26,33–36,39,42,45,49,51] use intermediate representations of the teacher as hints to train the student. TAKD [32] employs intermediate-sized networks as assistants to improve the effectiveness of KD when the teacher-student capacity gap is large. Different from TAKD, our approach adopts the assistants to solve the heterogeneity between detector pairs.

Homogeneous detection KD. Chen *et al.* [3] first apply KD to object detection by implementing feature-based and response-based loss for Faster R-CNN. Li *et al.* [25] apply L2 loss on features sampled by proposals of the student. Wang *et al.* [43] find that mimicking features from foreground regions is more important than background and only distills the feature near object anchor locations. DeFeat [11] shows that the information of background features is also essential, so foreground and background regions are distilled simultaneously with different factors. LabelEnc [12] first trains an autoencoder to model the location-category information and then use the label representations to supervise the training of the detectors. Although these methods have achieved great success on heterogeneous backbones, heterogeneity between detection architectures are rarely explored due to the large structural and feature semantics differences.

Heterogeneous Detection KD. To transfer knowledge between heterogeneous detectors, MimicDet [31] introduces a refinement module to an one-stage detection head to imitate the workflow of two-stage heads, and then conducts KD between the aligned features from the teacher and student heads. Although MimicDet improves accuracy, it is hard to transfer the structural modification on the student head to other heterogeneous detectors. Different from MimicDet, our method is more intuitive and flexible. With minor modifications, a variety of heterogeneous detectors can be supported by our framework.

G-DetKD [48] first proposes a general distillation framework for object detectors, which performs soft matching across all pyramid levels to provide the optimal guidance to the student. However, using learned similarity scores to combine features of students at different levels before feature mimicking does not essentially reduce the semantic gap. In our work, we attach an assistant, which is same as the teacher detection head, to the student network to learn directly from the teacher. Since the assistant and the teacher have homogeneous detection heads, their feature semantic gap are smaller than the heterogeneous heads, contributing to more efficient knowledge transfer.

3 Method

In sec. 3.1, we briefly review the pipeline of detection KD. Then we elaborate our proposed hetero-KD mechanism, HEtero-Assists Distillation (HEAD) in sec. 3.2. Finally, we introduce a teacher-free extension of HEAD (TF-HEAD) in sec. 3.3.

3.1 Review of Detection KD

We focus on distillation using intermediate features [39]. The distillation loss of the features can be generally formulated as

$$\mathcal{L} = \mathcal{D}\left(\mathbf{F}^T, \phi(\mathbf{F}^S)\right), \tag{1}$$

where $\mathcal{D}(\cdot)$ is a distillation loss measuring the knowledge difference between the teacher and the student. ϕ is an adaptation layer to match the dimension of the student's feature with the teacher. $\mathbf{F}^T$ and $\mathbf{F}^S$ are intermediate features of the teacher and the student respectively.

In this paper, we define $\mathcal{D}(\cdot)$ as the MSE loss. The form of ϕ depends on the shape of $\mathbf{F}^S$. For three-dimensional features with different number of channels, 1×1 convolution layers are used. For two-dimensional features with different number of dimensions, ϕ represents a linear layer.

3.2 HEAD

We elaborate the pipeline of HEAD by instantiating an example, where Faster R-CNN [10] is adopted as the teacher and RetinaNet [28] is the student. Fig. 2 shows the corresponding framework. To bridge the semantic gap between teacher and student, HEAD constructs an assistant that is homogeneous with the teacher's R-CNN head. The assistant is initialized with the pretrained weight of the teacher's R-CNN head and is trained online with the student. Note that HEAD acts on the training phase only, and the assistant is unused during inference.

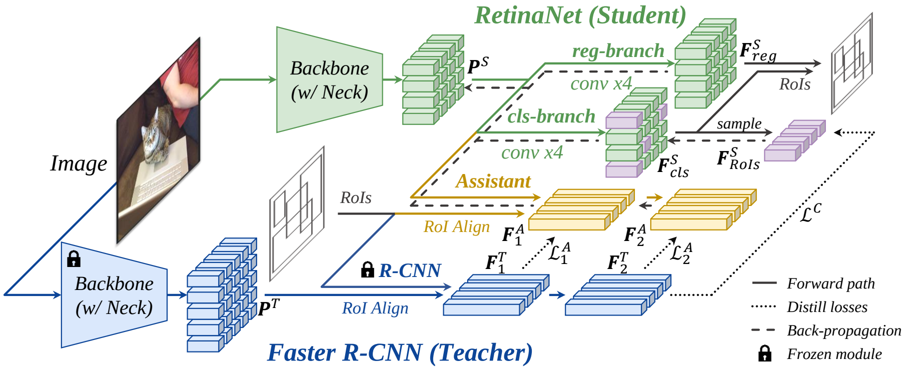


Fig. 2: Overview of HEAD, where the teacher is Faster R-CNN [38] and student is RetinaNet [28]. We construct an assistant homogeneous to the teacher's R-CNN head. We first extract the backbone features of the teacher and the student. The student backbone features are then fed into the original student head and the assistant in parallel. The teacher head processes the teacher backbone features. KD mechanism comprises an AKD between the teacher head and the assistant, and a CKD, where the teacher head directly supervises the student head.

Given an image, we first extract the backbone features of student and teacher, denoted as $\mathbf{P}^S \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{P}^T \in \mathbb{R}^{C\times H'\times W'}$ respectively. We then follow the original detector pipeline to employ the student head, a RetinaNet [28] head, to calculate the student loss, denoted as $\mathcal{L}^S_{gt}$. As shown in fig. 2, the student head comprises a regression branch and a classification branch. The student loss is calculated by summarizing losses from both branches

$$\mathcal{L}^S_{gt} = \mathcal{L}^S_{reg} + \mathcal{L}^S_{cls}. \tag{2}$$

The $\mathcal{L}^S_{gt}$ is calculated following the original RetinaNet. For convinience, we adopt $\mathbf{F}^S_{reg} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{F}^S_{cls} \in \mathbb{R}^{C\times H\times W}$ to denote the last intermediate feature of the regression branch and the classification branch respectively.

We next introduce the KD mechanism composed of two KD processes, *i.e.* *Assistant-based KD (AKD)* and *Cross-architecture KD (CKD)*. AKD is the core of HEAD. We utilize the teacher's R-CNN head and the assistant to process the corresponding backbone features $\mathbf{P}^T$ and $\mathbf{P}^S$, respectively. Meanwhile, we adopt the AKD loss $\mathcal{L}^A$ for the assistant to mimic the intermediate features of the teacher head. For completeness, we employ CKD, where the teacher head directly provides supervision for the student head. The CKD loss is denoted as $\mathcal{L}^C$. During the training phase, the overall loss is

$$\mathcal{L}^{HEAD} = \mathcal{L}^S_{gt} + \mathcal{L}^A + \mathcal{L}^C. \tag{3}$$

Our HEAD framework is not restricted to distillation between one-stage and two-stage detectors but can be applied to a wide range of heterogeneous detectors.

Assistant-based KD. When the teacher is two-stage, the teacher head and the assistant perform RoI Align [13] on $\mathbf{P}^T$ and $\mathbf{P}^S$ respectively with a precomputed set of RoIs. For a two-stage student, the output of the student's RPN is used as the precomputed RoIs. For one-stage students without RPN, we take the output of the student head as a substitution. For the example in fig. 2, we convert the classification logits of each anchor to a class-agnostic objectness logit and follow the original RPN protocol to generate RoIs. Additionally, we denote the number of RoIs as N.

We feed the backbone features $\mathbf{P}^T$ and $\mathbf{P}^S$ (or the RoI Aligned features) into the teacher head and the assistant, respectively. The intermediate features of the teacher head and the assistant is respectively denoted as $\mathbf{F}_1^T, \mathbf{F}_2^T, \ldots, \mathbf{F}_L^T$ and $\mathbf{F}_1^A, \mathbf{F}_2^A, \ldots, \mathbf{F}_L^A$. L indicates the number of intermediate features. In fig. 2, the R-CNN head is composed of two linear layers. We use the outputs of both layers for KD, thus setting L to 2. Finally, since the teacher head is homogeneous with the assistant, we simply apply KD between intermediate features pairs

$$\mathcal{L}_l^A = \mathcal{D}\left(\mathbf{F}_l^T, \phi\left(\mathbf{F}_l^S\right)\right). \tag{4}$$

For simplicity, we use MSE loss as $\mathcal{D}(\cdot)$ and a linear layer as ϕ.

Besides the supervision from the teacher, we also use ground truth labels to supervise the assistant. In fig. 2, we follow the original Faster R-CNN to use the standard Cross-Entropy loss and L1 loss to supervise the classification and regression output of the assistant. In general, the ground truth loss for the assistant $\mathcal{L}_{gt}^A$ is same as the teacher, so that the assistant learns from the ground truth labels when the teacher makes mistakes. The total AKD loss is defined as

$$\mathcal{L}^A = \mathcal{L}_{gt}^A + \frac{\lambda^A}{L}\sum_{l=1}^{L}\mathcal{L}_l^A, \tag{5}$$

where λ^A represents the loss weight.

Intuitively, $\mathcal{L}^A$ requires the assistant to reproduce the reasoning process of the teacher. For this goal, the student backbone needs to capture enough information in $\mathbf{P}^S$. Therefore, the assistant optimizes the student backbone via gradient backpropagation, so that the student backbone learns the knowledge that is critical for the assistant to reproduce the teacher's reasoning process.

Cross-architecture KD. Though the AKD is effective and universal, it only distills knowledge into the student backbone. Hence, we design the CKD to further improve the student performance via direct supervisions from the teacher head to the student head. As shown in fig. 2, we apply the CKD loss $\mathcal{L}^C$ between the teacher's R-CNN head and the student's RetinaNet head. Firstly, the teacher's R-CNN head generates a set of sparse RoI features $\mathbf{F}_1^T \in \mathbb{R}^{N\times C'}$, while the student's RetinaNet head generates a series of dense anchor features $\mathbf{F}_{cls}^S \in \mathbb{R}^{C\times H\times W}$. C' represents the dimensions of the R-CNN head's hidden layer. We ignore the regression feature $\mathbf{F}_{reg}^S$ following G-DetKD [48]. Secondly, inspired by MimicDet [31], we trace back to the original anchors of each RoI. Thirdly, since each anchor corresponds to a pixel on $\mathbf{F}_{cls}^S$, we sample these pixel features to form $\mathbf{F}_{RoIs}^S \in \mathbb{R}^{N\times C}$. Thereafter, we use eq. (1) to perform CKD

$$\mathcal{L}^C = \lambda^C\mathcal{D}\left(\mathbf{F}_1^T, \phi(\mathbf{F}_{RoIs}^S)\right), \tag{6}$$

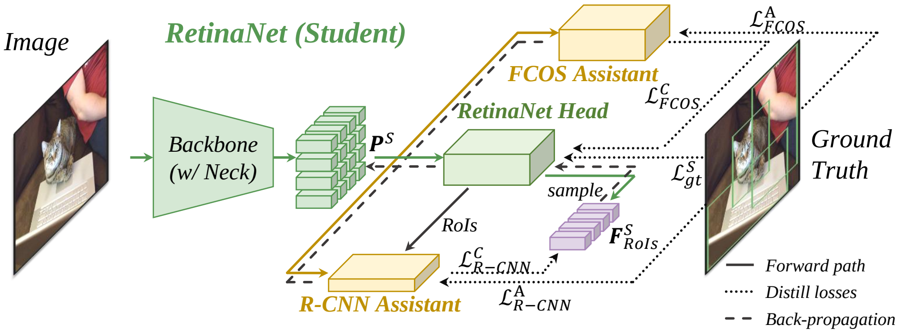


Fig. 3: An example of the TF-HEAD training pipeline. We use R-CNN head [38] and FCOS head [41] as two assistants to guide the RetinaNet [28].

where λ^C is the loss weight, $\mathcal{D}(\cdot)$ is MSE loss, and ϕ is a linear layer mapping from C-dimensional features to C'-dimensional features.

If the teacher and the student are both one-stage or both two-stage, CKD simply applies MSE loss between the corresponding intermediate features of the teacher head and the student head. Suppose the teacher is FCOS [41] and the student is RetinaNet [28]. Let $\mathbf{F}^T_{FCOS}$ denote the last intermediate feature of the classification branch of the FCOS head. Then the CKD loss is denoted as

$$\mathcal{L}^C_{FCOS} = \lambda^C_{FCOS}\mathcal{D}\left(\mathbf{F}^T_{FCOS}, \phi\left(\mathbf{F}^S_{cls}\right)\right), \tag{7}$$

where λ^C_{FCOS} is the loss weight, $\mathcal{D}(\cdot)$ is MSE loss and ϕ is a 1×1 convolution layer. For both teacher and student with a two-stage pipeline, the only difference is that the adaptation layer ϕ uses a linear layer.

3.3 TF-HEAD

In practice, suitable teachers for a specific student are not always available. Traditional teacher-based KD methods, including our HEAD, fail in such situations. Therefore, we extend our HEAD to a teacher-free method, namely TF-HEAD. TF-HEAD is designed based on an experimental observation. As shown in fig. 1, different detectors [28, 38, 41] have distinct activation maps, demonstrating diverse knowledge in the backbones. Heterogeneous detection architectures incorporate different human priors and adopt various backbone optimization manners. Therefore, the heterogeneous detectors produce different knowledge.

Even without the pretrained teacher in HEAD, we observe that the assistant can still learn from the ground truth labels. Based on this, we devise a teacher-free KD mechanism, TF-HEAD. As shown in fig. 3, TF-HEAD uses assistants to transfer knowledge from the heterogeneous detectors to the student. Note that we allow more than one assistant in TF-HEAD. While the TF-HEAD framework is universal, we use the example in fig. 3 to show its application, where the FCOS head and the R-CNN head are adopted to teach the RetinaNet head.

For each assistant, we denote its ground truth loss and CKD loss as $\mathcal{L}_\star^A$ and $\mathcal{L}_\star^C$, respectively, as described in sec. 3.2. $\star$ indicates the name of the assistant. For the example in fig. 3, the overall loss is

$$\mathcal{L}^{TF-HEAD} = \mathcal{L}_{gt}^S + \mathcal{L}_{FCOS}^A + \mathcal{L}_{FCOS}^C + \mathcal{L}_{R-CNN}^A + \mathcal{L}_{R-CNN}^C. \tag{8}$$

More generally, the overall loss of our TF-HEAD framework is

$$\mathcal{L}^{TF-HEAD} = \mathcal{L}_{gt}^S + \sum_\star \left(\mathcal{L}_\star^A + \mathcal{L}_\star^C\right), \tag{9}$$

where $\mathcal{L}_{gt}^S$ is the ground truth loss of the student detector. We represent the loss weights of assistant $\star$ as $\lambda_\star^A$ and $\lambda_\star^C$.

4 Experiments

Experiments are conducted on the COCO 2017 dataset [29] using the mean Average Precision (mAP) metric. We adopt the default 120k/5k split for training and validation. All distillation loss $\mathcal{D}(\cdot)$ takes the form of MSE. The adaptation layer ϕ is either a 1×1 convolution layer or a linear layer, depending on the shape of its input. For AKD, we set λ^A to 5. If the student head and the assistant are both one-stage heads, the CKD loss weight λ^C is set to 1, otherwise 2.

Training is conducted on 8 GPUs with batch size 16 in total. We use stochastic gradient descent (SGD) optimizer with 0.9 momentum and 0.0001 weight decay. 1x (12 epochs) training schedule is used. At the 8^{th} and 11^{th} epochs, the learning rate is divided by 10 The initial learning rate is 0.01 for one-stage detectors and 0.02 for two-stage detectors. The shorter side of the input image is scaled to 640-800 pixels, the longer side is scaled to 1333 pixels.

4.1 Main Results

Comparison with homo-KD methods. In this section, HEAD is compared with previous homogeneous detection KD methods. Since these methods are originally proposed for homogeneous detectors, only the backbone mimicking part can be applied. For fairness, we do not use the CKD loss $\mathcal{L}^C$, but only use the AKD loss $\mathcal{L}^A$. We conduct experiments with two student architectures (RetinaNet [28] and FCOS [41]), two student backbones (R18 [15] and MNV2 [40]), and two teacher choices (R50 Faster R-CNN [38] and R50 RepPoints [46]). On all eight teacher-student pairs, tab. 1 shows that HEAD outperforms the previous methods by a large margin. Specifically, under the guidance of R50 Faster R-CNN, our HEAD framework boosts the RetinaNet [28] performance by 3.8 and 4.0 mAP for R18 and MNV2 backbones respectively. For FCOS, the performance gain is also prominent. We observe a 3.5 mAP gain on both backbones when using R50 Faster R-CNN as the teacher. Interestingly, we observe that homo-KD methods degrade the student's performance in some cases when applied to heterogeneous detector pairs. Because the semantic gap between heterogeneous

Table 1: Comparison with homogeneous KD methods. † indicates that only Assistant-based KD losses are used in HEAD.

Student Backbone		R18 [15]			MNV2 [40]		
Method	Teacher	mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}
RetinaNet [28]	-	31.7	49.5	33.5	28.5	44.8	29.9
FitNet [39]	R50 Faster R-CNN (40.3) [38]	34.1	52.2	36.0	31.6	48.7	33.5
FGFI [43]		34.4	52.2	36.4	30.8	47.3	32.6
DeFeat [11]		34.1	52.1	36.3	31.1	47.9	32.7
FGD [47]		34.4	52.6	36.7	31.9	48.9	34.0
HEAD† (ours)		**35.5**	**54.5**	**37.9**	**32.5**	**50.4**	**34.4**
FitNet [39]	R50 RepPoints (38.6) [46]	31.5	49.0	33.3	27.5	43.2	28.9
FGFI [43]		33.1	50.8	35.3	29.1	45.1	30.8
DeFeat [11]		30.9	48.0	32.8	28.2	44.2	29.6
FGD [47]		31.3	48.6	33.2	28.3	44.3	30.0
HEAD† (ours)		**34.2**	**52.4**	**36.6**	**30.5**	**47.1**	**32.3**
FCOS [41]	-	32.5	50.9	34.1	30.0	47.5	31.3
FitNet [39]	R50 Faster R-CNN (40.3) [38]	34.2	52.2	36.1	32.0	49.3	33.7
FGD [47]		35.4	53.8	37.3	32.9	50.5	34.6
HEAD† (ours)		**36.0**	**54.9**	**38.4**	**33.5**	**51.6**	**35.2**
FitNet [39]	R50 RepPoints (38.6) [46]	32.7	50.9	34.4	30.3	47.6	31.7
FGD [47]		33.8	52.1	35.7	31.2	48.8	32.5
HEAD† (ours)		**35.0**	**53.8**	**36.8**	**32.5**	**50.4**	**34.3**

detectors is much larger than homogeneous detectors, the homo-KD methods are prone to over regularize the student, which causes this phenomena [35].

Comparison with hetero-KD methods. We further compare our HEAD with the previous hetero-KD methods. When using Faster R-CNN as the teacher, we compare our HEAD with G-DetKD [48]. For RepPoints [46] teacher, since the contrastive loss in G-DetKD is not applicable to such one-stage detector, we use its backbone mimicking loss (SGFI loss) only. For fairness, we disable our CKD loss as well. As shown in tab. 2, our HEAD surpasses G-DetKD and SGFI on various teacher-student pairs. Notice that Faster R-CNN uses the $2-6$ levels of the FPN [27] features, while RetinaNet, FCOS, and RepPoints use the $3-7$ levels. The result suggests that the semantic-guided feature level matching mechanism cannot effectively bridge the semantic gap between the teacher and the student if both detectors use the same FPN levels. In contrast, our HEAD bridges the semantic gap by introducing assistants to homogenize the teacher-student pair, which results in over 2.5 mAP gain on all scenarios.

Comparison with teacher-free methods. Some methods have implemented teacher-free KD on object detection, such as MimicDet [31] and LabelEnc [12]. To verify the superiority of our TF-HEAD among teacher-free methods without changing the structure of the student model, we only choose LabelEnc for comparison. LabelEnc adopts a two-step training process, where both steps take 12

Table 2: Comparison with heterogeneous KD methods. **† indicates that only AKD losses are used in HEAD.**

Student Backbone		R18 [15]			MNV2 [40]		
Method	Teacher	mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}
RetinaNet [28]	-	31.7	49.5	33.5	28.5	44.8	29.9
G-DetKD [48]	R50 Faster R-CNN	35.4	54.2	37.9	32.6	**50.9**	**34.5**
HEAD (ours)	(40.3) [38]	**36.2**	**55.2**	**38.8**	**32.8**	50.8	34.4
SGFI [48]	R50 RepPoints	31.6	49.5	33.2	27.9	43.9	29.2
HEAD† (ours)	(38.6) [46]	**34.2**	**52.4**	**36.6**	**30.5**	**47.1**	**32.3**
FCOS [41]	-	32.5	50.9	34.1	30.0	47.5	31.3
G-DetKD [48]	R50 Faster R-CNN	34.1	52.6	36.3	32.1	50.4	33.6
HEAD (ours)	(40.3) [38]	**36.0**	**54.9**	**38.4**	**33.5**	**51.6**	**35.2**
SGFI [48]	R50 RepPoints	32.6	50.9	34.4	30.2	47.6	31.7
HEAD† (ours)	(38.6) [46]	**35.0**	**53.8**	**36.8**	**32.5**	**50.4**	**34.3**
Faster R-CNN [38]	-	33.9	54.1	36.3	28.3	47.0	29.5
G-DetKD [48]	R50 Cascade R-CNN	36.1	57.3	39.0	33.4	54.2	35.3
HEAD (ours)	(43.5) [2]	**36.7**	**58.0**	**39.3**	**33.8**	**54.4**	**35.8**

Table 3: Comparison with the teacher-free detection KD method. For fairness, we use R50 [15] backbone and 2x (24 epoch) training schedule for all experiments. $\star$ indicates FCOS with improvements including center-sampling, normalization on bbox, centerness on regression branch, and GIoU.

Method	RetinaNet [28]	FCOS$\star$ [41]	Faster R-CNN [38]
Baseline	38.7	41.0	39.4
LabelEnc [12]	39.6	41.8	39.6
TF-HEAD (ours)	**40.2**	**42.3**	**40.5**

epochs (1x) for training. Therefore, we use a 2x (24 epoch) training schedule for TF-HEAD. Three detection architectures with R50 backbones are explored, *i.e.* RetinaNet [28], FCOS [41], and Faster R-CNN [38]. Our TF-HEAD uses two assistants for each architecture. For RetinaNet, TF-HEAD uses FCOS$\star$ head and DH R-CNN head [44]. For FCOS, RetinaNet head and DH R-CNN head are used. For Faster R-CNN, we use RetinaNet head and FCOS$\star$ head to guide the RPN module. As shown in tab. 3, HEAD improves baselines by 1.5, 1.3, and 1.1 mAP, which is 67%, 63%, and 350% higher than the improvement of LabelEnc.

4.2 Ablation Study

Effectiveness of HEAD. To evaluate the effectiveness of the assistant as a bridge between the teacher and the student, we conduct experiments without CKD loss. As show in tab. 4, HEAD† improves R18 RetinaNet by 3.8 mAP and

Table 4: Ablation study of the HEAD. We use R50 Faster R-CNN as teacher and RetinaNet as student. † indicates that only AKD losses are used in HEAD.

Student Backbone		R18 [15]			MNV2 [40]		
Method	Teacher	mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}
RetinaNet [28]	-	31.7	49.5	33.5	28.5	44.8	29.9
HEAD†	R50 Faster R-CNN (40.3) [38]	35.5	54.5	37.9	32.5	50.4	**34.4**
HEAD		**36.2**	**55.2**	**38.8**	**32.8**	**50.8**	**34.4**
HEAD†	R50 RepPoints (38.6) [46]	34.2	52.4	**36.6**	30.5	47.1	**32.3**
HEAD		**34.3**	**52.8**	36.4	**30.6**	**47.7**	**32.3**

Table 5: Ablation study of TF-HEAD. The student is R18 RetinaNet [28]. We use FCOS head and R-CNN head as two assistants to guide the student.

$\mathcal{L}^{A}_{FCOS}$	$\mathcal{L}^{C}_{FCOS}$	$\mathcal{L}^{A}_{R-CNN}$	$\mathcal{L}^{C}_{R-CNN}$	mAP	mAP_{50}	mAP_{75}	mAP_{s}	mAP_{m}	mAP_{l}
				31.7	49.5	33.5	16.8	34.7	42.1
✓				32.9	51.2	34.8	17.8	36.0	43.5
✓	✓			33.0	51.4	35.0	17.3	36.3	43.9
		✓		33.4	52.1	35.1	18.1	36.4	44.8
		✓	✓	33.7	52.3	35.6	17.8	36.9	44.9
✓		✓		33.8	52.5	35.6	18.2	36.8	**45.2**
✓	✓	✓	✓	**33.9**	**52.7**	**35.9**	**18.5**	**37.4**	**45.2**

MNV2 RetinaNet by 4.0 mAP. Adding CKD loss further brings 0.7 mAP gain to R18 RetinaNet and 0.3 mAP gain to MNV2 RetinaNet.

Effectiveness of TF-HEAD. Here, we investigate the effectiveness of each distillation component in eq. (9). We use FCOS head and R-CNN head as assistants to guide an R18 RetinaNet. As shown in tab. 5, our proposed TF-HEAD achieves 2.2 mAP improvement over the original RetinaNet, demonstrating the effectiveness of our TF-HEAD structure. FCOS assistant brings point-based knowledge for the backbone, boosting the baseline by 1.2 mAP. Then, CKD loss between the FCOS assistant and the RetinaNet head adds a 0.1 mAP gain. In contrast, the R-CNN assistant on its own brings 1.7 mAP gain. The CKD loss further adds 0.3 mAP. Using both assistants simultaneously reaches 33.9 mAP.

Early-stop to prevent misguidance from assistants. As shown in fig. 4, from the 8th epoch, the CKD loss preternaturally increases while all other losses suddenly dropdown. This phenomenon suggests that the CKD loss has stopped helping the student head since then. Note that the 8th epoch is the first time to drop the learning rate. Annealing KD [20] believes that the detrimental effect is because the teacher disturbs the student from learning the ground truth labels. After the learning rate drops at the 8th epoch, the optimization direction of the assistant and the student head start to diverge, which enlarges the semantic gap in between. From this moment, CKD loss starts to over regularize [35] the stu-

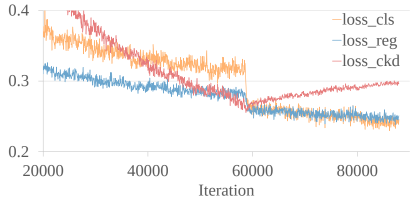
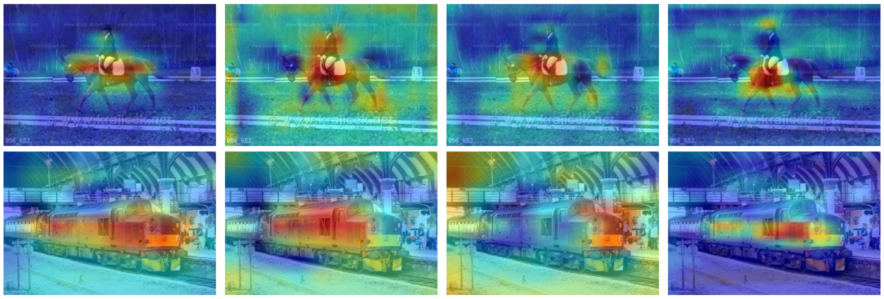


Fig. 4: Visualization of the CKD loss and ground truth losses.

Table 6: Early stopping the CKD losses increases the student performance by 0.1 mAP. Repeated experiments confirm that it is not caused by random factors.

Early-stop	mAP	mAP_s	mAP_m	mAP_l
	33.8	**18.5**	37.3	44.9
✓	**33.9**	**18.5**	**37.4**	**45.2**



TF-HEAD 1x RetinaNet 3x FCOS 3x Faster R-CNN 3x

Fig. 5: Visualization of backbone features from TF-HEAD, RetinaNet [28], FCOS [41], and Faster R-CNN [38]. TF-HEAD highlights more foreground area with higher intensity, while the background remains inactivated.

dent. Intuitively, the student head and the assistant have different architectures, so they should predict objects differently. Forcing the student head to mimic the assistants will impede it from learning knowledge by itself effectively. Therefore, we use early-stop to ensure that the student and assistants converge to their local optima. Tab. 6 shows that early-stop slightly improves the performance.

4.3 Visualization

Visualization of backbone feature maps. By visualizing the feature maps, we verify that TF-HEAD trains stronger backbones. As shown in fig. 5, feature maps generated by HEAD accurately identify the regions containing objects, showing that knowledge of RetinaNet [28], FCOS [41], and Faster R-CNN [38] can complement each other. Therefore, the backbone of TF-HEAD generates more informative feature maps, which helps the student to be more accurate.

Visualization of COCO error analysis. Fig. 6 presents analysis on *All Class* and two randomly selected classes. From RetinaNet to TF-HEAD, the *Background* error and the *False Negative* error decreases prominently, suggesting that the student has learned knowledge from heterogeneous detectors to make more accurate predictions. As a result, the *Correct* rate increases prominently. Using

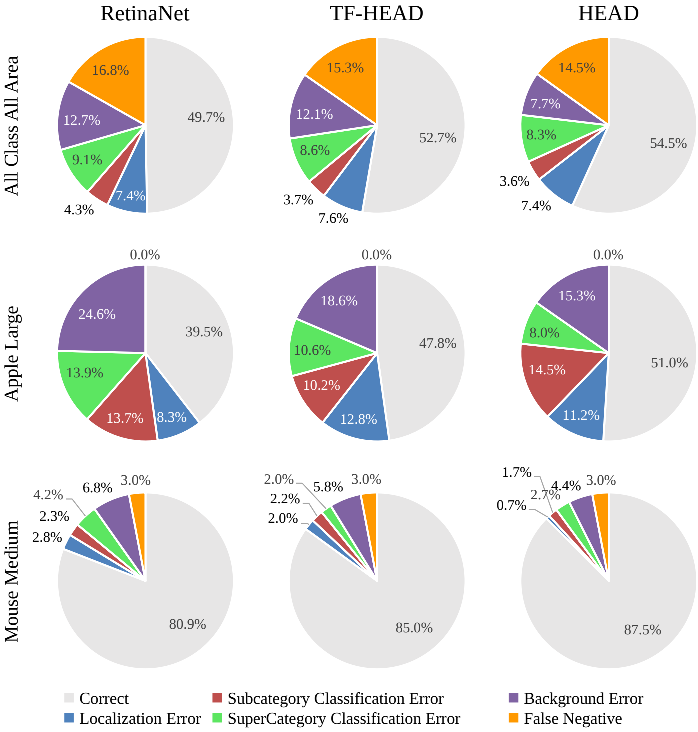


Fig. 6: COCO error analysis using tool from [18].

the pretrained teacher detector, HEAD further pushes the accuracy higher, with the *Background* error and the *False Negative* error even lower.

5 Conclusion

In this paper, we investigate KD among heterogeneous object detectors and find that the semantic gap between heterogeneous models is responsible for the difficulty of hetero-KD. Based on the observation, we design a simple yet effective HEtero-Assists Distillation (HEAD) mechanism. HEAD bridges the semantic gap between heterogeneous detectors via an adaptive assistant, thus simplifying to a homo-KD problem. For situations when the pretrained teachers are not available, we further propose a teacher-free method named TF-HEAD. Extensive experiments demonstrate the effectiveness of our framework.

Acknowledgement. This work was partly supported by the National Natural Science Foundation of China (62122010, 61876177), the Fundamental Research Funds for the Central Universities, and the Key Research and Development Program of Zhejiang Province (2022C01082).

References

1. Ahn, S., Hu, S.X., Damianou, A., Lawrence, N.D., Dai, Z.: Variational Information Distillation for Knowledge Transfer. In: CVPR. pp. 9155–9163 (2019) 4
2. Cai, Z., Vasconcelos, N.: Cascade R-CNN: Delving Into High Quality Object Detection. In: CVPR. pp. 6154–6162 (2018) 11
3. Chen, G., Choi, W., Yu, X., Han, T., Chandraker, M.: Learning efficient object detection models with knowledge distillation. In: NeurIPS. pp. 743–752 (2017) 2, 4
4. Chen, Q., Wang, Y., Yang, T., Zhang, X., Cheng, J., Sun, J.: You Only Look One-level Feature. In: CVPR. pp. 13034–13043 (2021) 4
5. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object Detection via Region-based Fully Convolutional Networks. In: NeurIPS. pp. 379–387 (2016) 4
6. Dai, X., Jiang, Z., Wu, Z., Bao, Y., Wang, Z., Liu, S., Zhou, E.: General Instance Distillation for Object Detection. In: CVPR. pp. 7838–7847 (2021) 2
7. Dong, Z., Li, G., Liao, Y., Wang, F., Ren, P., Qian, C.: CentripetalNet: Pursuing High-Quality Keypoint Pairs for Object Detection. In: CVPR. pp. 10516–10525 (2020) 4
8. Du, S., You, S., Li, X., Wu, J., Wang, F., Qian, C., Zhang, C.: Agree to disagree: Adaptive ensemble knowledge distillation in gradient space. In: NeurIPS. pp. 1–11 (2020) 4
9. Duan, K., Bai, S., Xie, L., Qi, H., Huang, Q., Tian, Q.: CenterNet: Keypoint Triplets for Object Detection. In: ICCV. pp. 6568–6577 (2019) 4
10. Girshick, R.: Fast R-CNN. In: ICCV. pp. 1440–1448 (2015) 4, 5
11. Guo, J., Han, K., Wang, Y., Wu, H., Chen, X., Xu, C., Xu, C.: Distilling Object Detectors via Decoupled Features. In: CVPR. pp. 2154–2164 (2021) 2, 4, 10
12. Hao, M., Liu, Y., Zhang, X., Sun, J.: LabelEnc: A New Intermediate Supervision Method for Object Detection. In: ECCV. pp. 529–545 (2020) 3, 4, 10, 11
13. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask R-CNN. TPAMI **42**(2), 386–397 (2020) 4, 6
14. He, K., Zhang, X., Ren, S., Sun, J.: Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. In: ECCV. pp. 346–361 (2014) 4
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR. pp. 770–778 (2016) 9, 10, 11, 12
16. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge Transfer via Distillation of Activation Boundaries Formed by Hidden Neurons. AAAI **33**(1), 3779–3787 (2019) 4
17. Hinton, G., Vinyals, O., Dean, J.: Distilling the Knowledge in a Neural Network. In: NeurIPS (2014) 2, 4
18. Hoiem, D., Chodpathumwan, Y., Dai, Q.: Diagnosing Error in Object Detectors. In: ECCV. pp. 340–353 (2012) 14
19. Huang, Z., Wang, N.: Like What You Like: Knowledge Distill via Neuron Selectivity Transfer. arXiv preprint arXiv:1707.01219 (2017) 4
20. Jafari, A., Rezagholizadeh, M., Sharma, P., Ghodsi, A.: Annealing knowledge distillation. In: EACL. pp. 2493–2504 (2021) 12
21. Ji, M., Shin, S., Hwang, S., Park, G., Moon, I.C.: Refine Myself by Teaching Myself: Feature Refinement via Self-Knowledge Distillation. In: CVPR. pp. 10659–10668 (2021) 3
22. Kim, J., Park, S., Kwak, N.: Paraphrasing Complex Network: Network Compression via Factor Transfer. In: NeurIPS. pp. 2760–2769 (2018) 4

23. Kim, K., Ji, B., Yoon, D., Hwang, S.: Self-Knowledge Distillation with Progressive Refinement of Targets. In: ICCV. pp. 6567–6576 (2021) 3
24. Lan, X., Zhu, X., Gong, S.: Knowledge Distillation by On-the-Fly Native Ensemble. In: NeurIPS. pp. 7517–7527 (2018) 2
25. Li, Q., Jin, S., Yan, J.: Mimicking Very Efficient Network for Object Detection. In: CVPR. pp. 7341–7349 (2017) 4
26. Li, X., Wu, J., Fang, H., Liao, Y., Wang, F., Qian, C.: Local Correlation Consistency for Knowledge Distillation. In: ECCV. pp. 18–33 (2020) 4
27. Lin, T.Y., Dollar, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature Pyramid Networks for Object Detection. In: CVPR. pp. 936–944 (2017) 2, 4, 10
28. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal Loss for Dense Object Detection. TPAMI **42**(2), 318–327 (2020) 2, 4, 5, 6, 8, 9, 10, 11, 12, 13
29. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common Objects in Context. In: ECCV. pp. 740–755 (2014) 3, 9
30. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot MultiBox Detector. In: ECCV. pp. 21–37 (2016) 4
31. Lu, X., Li, Q., Li, B., Yan, J.: MimicDet: Bridging the Gap Between One-Stage and Two-Stage Object Detection. In: ECCV. pp. 541–557 (2020) 4, 7, 10
32. Mirzadeh, S.I., Farajtabar, M., Li, A., Levine, N., Matsukawa, A., Ghasemzadeh, H.: Improved Knowledge Distillation via Teacher Assistant. AAAI **34**(04), 5191–5198 (2020) 4
33. Park, W., Kim, D., Lu, Y., Cho, M.: Relational Knowledge Distillation. In: CVPR. pp. 3962–3971 (2019) 2, 4
34. Passalis, N., Tefas, A.: Probabilistic Knowledge Transfer for Deep Representation Learning. arXiv preprint arXiv:1803.10837 (2018) 4
35. Passalis, N., Tzelepi, M., Tefas, A.: Heterogeneous Knowledge Distillation Using Information Flow Modeling. In: CVPR. pp. 2336–2345 (2020) 4, 10, 12
36. Peng, B., Jin, X., Li, D., Zhou, S., Wu, Y., Liu, J., Zhang, Z., Liu, Y.: Correlation Congruence for Knowledge Distillation. In: CVPR. pp. 5006–5015 (2019) 4
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: CVPR. pp. 779–788 (2015) 4
38. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. TPAMI **39**(6), 1137–1149 (2017) 2, 4, 6, 8, 9, 10, 11, 12, 13
39. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: FitNets: Hints for Thin Deep Nets. In: ICLR. pp. 1–13 (2015) 2, 4, 5, 10
40. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C.: MobileNetV2: Inverted Residuals and Linear Bottlenecks. In: CVPR. pp. 4510–4520 (2018) 9, 10, 11, 12
41. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: Fully Convolutional One-Stage Object Detection. In: CVPR. pp. 9626–9635 (2019) 2, 4, 8, 9, 10, 11, 13
42. Tung, F., Mori, G.: Similarity-Preserving Knowledge Distillation. In: CVPR. pp. 1365–1374 (2019) 4
43. Wang, T., Yuan, L., Zhang, X., Feng, J.: Distilling Object Detectors With Fine-Grained Feature Imitation. In: CVPR. pp. 4928–4937 (2019) 2, 4, 10
44. Wu, Y., Chen, Y., Yuan, L., Liu, Z., Wang, L., Li, H., Fu, Y.: Rethinking Classification and Localization for Object Detection. In: CVPR. pp. 10183–10192 (2020) 11
45. Yang, C., An, Z., Cai, L., Xu, Y.: Hierarchical Self-supervised Augmented Knowledge Distillation. IJCAI pp. 1217–1223 (2021) 4

46. Yang, Z., Liu, S., Hu, H., Wang, L., Lin, S.: RepPoints: Point Set Representation for Object Detection. In: ICCV. pp. 9656–9665 (2019) 2, 9, 10, 11, 12
47. Yang, Z., Li, Z., Jiang, X., Gong, Y., Yuan, Z., Zhao, D., Yuan, C.: Focal and Global Knowledge Distillation for Detectors. In: CVPR (2022) 2, 10
48. Yao, L., Pi, R., Xu, H., Zhang, W., Li, Z., Zhang, T.: G-DetKD: Towards General Distillation Framework for Object Detectors via Contrastive and Semantic-guided Feature Imitation. In: ICCV (2021) 5, 7, 10, 11
49. Yim, J., Joo, D., Bae, J., Kim, J.: A Gift from Knowledge Distillation: Fast Optimization, Network Minimization and Transfer Learning. In: CVPR. pp. 7130–7138 (2017) 2, 3, 4
50. Yuan, L., Tay, F.E., Li, G., Wang, T., Feng, J.: Revisiting Knowledge Distillation via Label Smoothing Regularization. In: CVPR. pp. 3902–3910 (2020) 3
51. Zagoruyko, S., Komodakis, N.: Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In: ICLR. pp. 1–13 (2017) 4
52. Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K.: Be Your Own Teacher: Improve the Performance of Convolutional Neural Networks via Self Distillation. In: ICCV. pp. 3712–3721 (2019) 3
53. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep Mutual Learning. In: CVPR. pp. 4320–4328 (2018) 2
54. Zhou, C., Neubig, G., Gu, J.: Improve Object Detection with Feature-based Knowledge Distillation: Towards Accurate and Efficient Detectors. In: ICLR (2021) 2
55. Zhou, X., Koltun, V., Krähenbühl, P.: Probabilistic two-stage detection. arXiv preprint arXiv:2103.07461 (2021) 4
56. Zhou, X., Wang, D., Krähenbühl, P.: Objects as Points. arXiv preprint arXiv:1904.07850 (2019) 4