# Masked Autoencoders Are Stronger Knowledge Distillers

Shanshan Lao[1]* Guanglu Song[2] Boxiao Liu[2] Yu Liu[2]† Yujiu Yang[1]†
[1] Tsinghua Shenzhen International Graduate School, Tsinghua University
[2] Sensetime Research
laoss21@mails.tsinghua.edu.cn, {songguanglu, liuboxiao}@sensetime.com
liuyuisanai@gmail.com, yang.yujiu@sz.tsinghua.edu.cn

## Abstract

*Knowledge distillation (KD) has shown great success in improving student's performance by mimicking the intermediate output of the high-capacity teacher in fine-grained visual tasks,* e.g. *object detection. This paper proposes a technique called Masked Knowledge Distillation (MKD) that enhances this process using a masked autoencoding scheme. In MKD, random patches of the input image are masked, and the corresponding missing feature is recovered by forcing it to imitate the output of the teacher. MKD is based on two core designs. First, using the student as the encoder, we develop an adaptive decoder architecture, which includes a spatial alignment module that operates on the multi-scale features in the feature pyramid network (FPN) [20], a simple decoder, and a spatial recovery module that mimics the teacher's output from the latent representation and mask tokens. Second, we introduce the masked convolution in each convolution block to keep the masked patches unaffected by others. By coupling these two designs, we can further improve the completeness and effectiveness of teacher knowledge learning. We conduct extensive experiments on different architectures with object detection and semantic segmentation. The results show that all the students can achieve further improvements compared to the conventional KD. Notably, we establish the new state-of-the-art results by boosting RetinaNet ResNet-18, and ResNet-50 from 33.4 to 37.5 mAP, and 37.4 to 41.5 mAP, respectively.*

## 1. Introduction

As fundamental tasks in computer vision, high-level visual predictions such as object detection, instance segmentation, and semantic segmentation have been widely used in various practical applications. In real-world scenarios, improving the performance of the deployable models is crucial. Improving the performance of a lightweight student model via a high-capacity teacher introduces an effective paradigm, which is widely known as Knowledge Distillation (KD) [16, 4]. For dense visual prediction tasks, feature-based distillation [17, 7, 11] is more beneficial for the student to mimic the teacher's intermediate features. Some elaborated feature-based distillation methods [29, 4, 40, 18] have established the state-of-the-art performance for many lightweight models such as ResNet-18 [15].

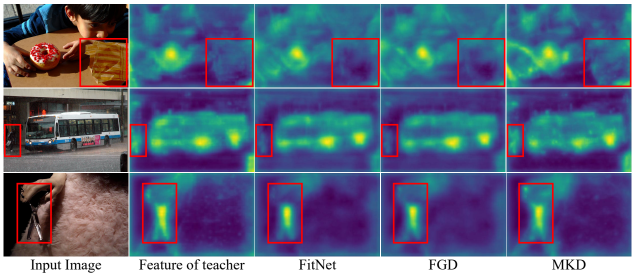

Figure 1: Visualization of feature maps of teacher networks (RetinaNet-ResX101) and student networks (RetinaNet-Res50) with different KD paradigms.

The core bottleneck of such feature-based KD is how to learn the complete knowledge from the teacher's output containing about 20K spatial coordinates (calculated on the COCO training set). According to our experiment, 72% of them are simple enough and converge quickly to achieve ∼90% similarity with the teacher's output. The remaining important knowledge is suppressed by this large number of simple samples, and it is difficult for the student to learn effectively. To further verify the existence of information redundancy in teacher features, we conduct experiments that, when 30% of the features are randomly masked during training and testing on Retina-R101 (38.4 mAP), it still achieved similar performance (38.1 mAP). Previous works [34, 46] point out similar observations that each pixel contributes equally if using a common distillation loss (*e.g.* MSE), which will cause the network to easily learn a lot of similar information and is hard to identify important information. So they use attention [45, 40] and decouple the

[*] Work done during internship at SenseTime Research.
[†] Corresponding authors.

foreground and background feature [34, 46] to help alleviate this problem. But attention can still lead to the imbalance of information learning.

In order to learn the complete information without being affected by the spatial redundancy in teacher features, we introduce the Mask Image Modeling (MIM) mechanism to effectively promote the learning of this information. To be more specific, we propose a new **M**asked **K**nowledge **D**istillation (MKD) framework to improve the knowledge learning of the student via the masked autoencoding paradigm. The core idea is that we mask random patches of the input image of the student while maintaining the whole input image for the teacher, and then recover the corresponding missing feature by forcing the student to imitate the output of the teacher. In this way, the student network is encouraged to predict the masked patches with corrupted input images and learn the relationship between the masked area and its surrounding regions, rather than simply imitating the output feature of the teacher at visible patches. We conduct some experiments trying to excavate the potential ability of the students. As shown in Fig. 1, according to the feature comparison in the red box, it can be observed that student network using MKD learns more complete knowledge from teacher than the direct feature-based method [1] and the attention-based method [40].

We introduce two crucial designs to alleviate some bottlenecks in this framework. First, due to the masked image input, directly applying the normal convolution in the convolutional neural networks (CNN) will confuse the latent representation in visible and invisible patches. Therefore, we adopt the masked convolution [10, 23] to keep the masked patches not affected by others in each convolution block. Second, multi-scale features are often used in fine-grained high-level visual tasks which have different masked sizes, leading to inconsistent feature reconstruction. We developed an adaptive decoder architecture for this problem to predict the teacher's feature in the corresponding masked area. More specifically, a spatial alignment module is first operated on the multi-scale features to align them to the same spatial resolution. A mask token is then replicated multiple times to replace the features in the masked area. The features in the visible patches and mask tokens are sent to an adaptive transformer decoder to reconstruct the teacher's feature. Finally, we conduct the spatial recovery module to convert the same spatial resolution to the original multi-scale resolution to perform feature-based distillation.

In our method, by the mask autoencoding paradigm with asymmetric input for student and teacher, the student is forced to infer the teacher's features in the invisible patches by imitating the teacher's features in the visible patches. We observe that the mask tokens in the decoder try to absorb knowledge from their adjacent region to restore the features in the masked region. Completely restoring the masked features drives it to fully learn the teacher's corresponding information in these adjacent areas. As shown in Fig. 2, the L2 distance value in adjacent regions of the mask can be lower with masked input. This verifies the claim that the student network can better learn the teacher's knowledge in our learning manner. MKD can be directly used in different architectures and various dense visual prediction tasks, *e.g.*, object detection and semantic segmentation. The results show that MKD can improve considerably cooperated with conventional feature-based distillation. For instance, with MKD in RetinaNet, the mAP of student ResNet-18 is greatly improved to 37.5, 4.1% higher than the baseline, and also outperforms the previous SOTA methods. To sum up, our contributions are as follows:

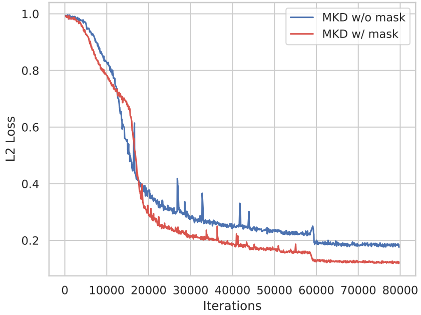

Figure 2: The average L2 distance between student's feature and teacher's feature calculated at the unmasked areas with RetinaNet-ResX101 distilling RetinaNet-Res50.

- We propose a new paradigm for feature-based distillation named MKD, using mask autoencoding scheme to effectively learn the complete knowledge in the teacher network. MKD masks random patches of the input image and recovers the corresponding masked feature by forcing it to imitate the teacher's output.

- We introduce the masked convolution and adaptive decoder module in MKD to make it easy to cooperate with different architectures in fine-grained visual tasks, *e.g.*, object detection and semantic segmentation.

- Extensive experiments on various models and tasks verify the effectiveness of our method. For different student architectures and tasks, MKD can further improve the performance of feature-based distillation and establish new state-of-the-art.

## 2. Related Works

### 2.1. Object Detection

As a fundamental and challenging task in computer vision, object detection aims to detect the visual objects of a certain class in the images. Modern object detectors can be broadly divided into three paradigms: two-

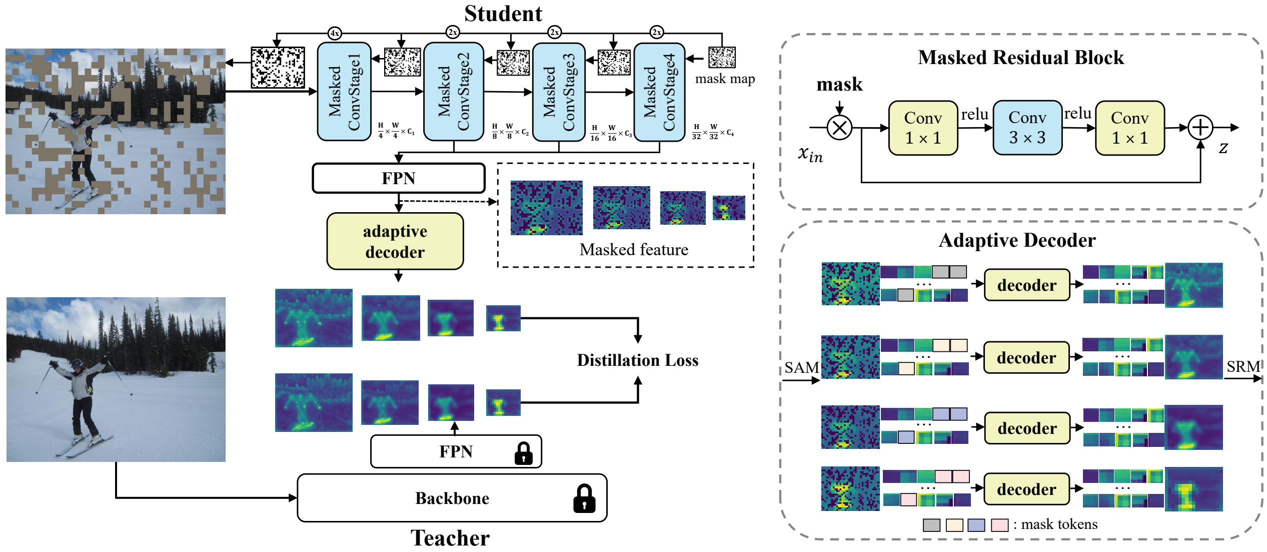

Figure 3: **Illustration of the MKD**. For clear visualization, we omit the original task losses, such as classification and localization loss. In this learning process, the parameter of the teacher is fixed. The mask token and decoder are not shared between different-scale features in FPN due to the semantic gap in different-scale features.

stage [28, 14, 35], anchor-based one-stage [21, 32, 24], and anchor-free one-stage [43, 9]. While detectors with strong backbones have better accuracy, they are computationally expensive and difficult to deploy on mobile devices. One interesting direction of research is the compression of detection networks, including quantization [36, 19], pruning [12, 38], and the design of lightweight networks [31]. In addition, Knowledge Distillation (KD) [16, 46, 37], has become an important method for transferring knowledge from larger, better-performing networks without changing the structure of the network.

### 2.2. Knowledge Distillation for Detection

Knowledge distillation is a model compression and acceleration method that can transfer knowledge from the teacher model to the student model. The logit-based distillation methods [16, 46, 44] in classification are limited for the improvement of the fine-grained visual understanding tasks because the localization information from the teacher is crucial. Therefore, feature-based distillation over multi-scale features has become the main method of distillation for detectors. Current feature distillation methods for detectors can be divided into the following four categories: (1) hint-based, (2) attention-based, (3) instance-based, and (4) masked-feature based. Hint-based methods [29, 1, 33, 42] take features from the intermediate layers as hints to guide the student model. However, such methods treat all regions equally and do not get satisfactory results. Instance-based methods aim to find the key instance regions for distillation by GT boxes or specifically designed modules, *e.g.* FGFI [34], GID [7], ICD [17]. Attention-based methods further improve the performance by using attention maps to locate discriminative areas, *e.g.* FKD [45], FGD [40]. Recent work MGD [41] first proposed to mask out the feature maps in the knowledge distillation branch and use a generator to restore the teacher feature. In our work, we mask random patches of the original images and adopt masked convolution into the backbone network to prevent information leakage. This new paradigm introduces masks throughout the backbone network, enhancing the corresponding knowledge learning in adjacent regions of the random mask. Experiments have shown that distillation benefits well from the masking scheme, especially for longer schedules.

## 3. Approach

### 3.1. Masked Knowledge Distillation

We first elaborate on the overall pipeline of our masked knowledge distillation (MKD) by instantiating an example in object detection. Let $F^S \in \mathbb{R}^{C \times H \times W}$ and $F^T \in \mathbb{R}^{C \times H \times W}$ denote the output features of student's FPN and teacher's FPN, respectively. The standard feature-based distillation can be formulated as:

$$\mathcal{L}_{feat} = \frac{1}{2N} \sum_{i=0}^{\mathcal{P}} (F_i^T - \phi(F_i^S))^2, \qquad (1)$$

where $\mathcal{P}$ indicates the total number of features in FPN and $\mathrm{N} = \mathrm{C_i H_i W_i}$. $C_i, H_i, W_i$ represent the channel number, height, and width for feature map $F_i^*$. $\phi$ is a convolutional layer with kernel size $1 \times 1$ to align the channel dimension between the $F^T$ and $F^S$.

As discussed above, to further enhance KD distillation dense visual predictions, we introduce the masked autoencoding scheme into KD learning. In this manner, the distillation of the features can be formulated as:

$$\mathcal{L}_{feat} = \frac{1}{2N} \sum_{i=0}^{\mathcal{P}} (F_i^T - f_{dec}(\phi(F_i^S), \mathcal{T}_{mask}))^2, \qquad (2)$$

where $f_{dec}$ is the proposed adaptive decoder module with spatial alignment module, stacked transformer decoder layers, and spatial recovery module. $\mathcal{T}_{mask}$ is the mask token.

The overall architecture of MKD is shown in Fig. 3. Firstly, given an image with $H \times W$, a binary mask map $\mathcal{M}$ with the element number $N = \frac{H}{s} \times \frac{W}{s}$ is randomly sampled with a given masking ratio. $s$ is the down-sampled factor of the mask map and we set it to 32 in our experiments. We progressively upsample the mask to align its resolution to the output of each convolutional stage in the student's backbone. Using the student as an encoder, the masked images are then sent to it and the masked residual blocks in each masked convolutional stage are conducted to leave the masked area unprocessed by the backbone in the forward process. Finally, the masked feature maps generated by FPN will be sent to an adaptive decoder module where a spatial alignment module (SAM) is introduced to operate on the multi-scale features, along with a decoder and spatial recovery module (SRM) that reconstructs the latent representations with mask tokens. This learning manner is supervised by forcing the student's output of the adaptive decoder to imitate the teacher's output. Even if we mask random patches of the input image, with the assistance of MKD, the student model can still infer the features of the invisible patches.

### 3.1.1 Masking in the Image Space.

Following ViT [8] and MAE [13], we first divide the input image into regular non-overlapping image patches, and then randomly generate a binary mask according to the given masking ratio. The masked area in the input image keeps unseen to the student model. We simply refer to this as "random masking". Notice that the random masking strategy follows a uniform distribution, which eliminates the potential center bias. The mask is illustrated in Fig. 3, which is a schematic gram, and the real masked image is shown in Fig. 1. Different from ViT, the convolutional neural network is designed with a pyramid scheme where different convolutional stages are set with different stride factors. This leads to multiple different spatial resolutions in the student's forward propagation. To make consistent with this, after generating a mask with resolution $\frac{H}{32} \times \frac{W}{32}$ for the late stage in the student, we progressively upsample the mask to larger resolutions in early convolutional stages as shown in Fig. 3. Introducing random masking to knowledge distillation can create a task that cannot be easily solved by directly performing the pixel-to-pixel spatial matching supervision.

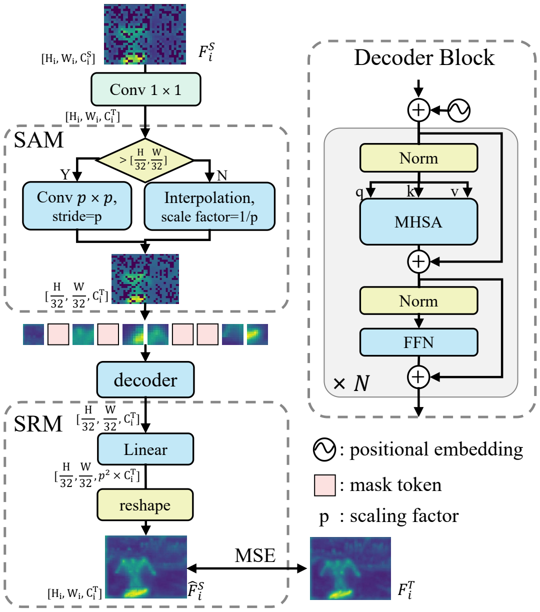

Figure 4: Illustration of the adaptive decoder. The inputs are the corresponding mask and feature maps from the teacher's FPN and the student's FPN.

### 3.1.2 Masked Encoder.

In our MKD, the student backbone is used as the encoder to process the masked input image. To ensure that the network only operates on the visible patches and reconstructs only from the neighboring ones, previously masked autoencoders such as MAE proposed to forward only the visible tokens into the encoder. However, this method cannot be used directly in convolutional networks since we need to preserve the spatial dimension of the image. Following ConvMAE [10], we adopt the masked convolution to solve this problem, which is commonly used in image inpainting [23] and sparse feature extraction [27] to process the incomplete input information. More specifically, masked convolutions are used in all blocks for each stage in the whole student so that the masked regions are not involved in the encoding process. The operation of masked convolution is shown in Fig. 3, where the input feature maps from the previous layer are masked firstly by performing the Hadamard product with the mask map. This can effectively avoid the confusing feature interaction between masked and visible regions. In this way, the decoder can only speculate based on the information of adjacent image blocks during reconstruction. We assume that this will help the encoder learn more expressive and content-rich features.

### 3.1.3 Adaptive Decoder.

The input to the adaptive decoder is the multi-scale masked feature maps of the student followed by a $1 \times 1$ convolution layer to align the channel number. The masked sizes are different in these multi-scale features, which leads to inconsistent feature reconstruction. To alleviate this, we introduce a simple spatial alignment module (SAM) to align them to the same spatial resolution ($\frac{1}{32}$ of the input image size in our experiments). As shown in Fig. 4, this is implemented by convolutional layer with stride $p$ or upsampling the spatial resolution by a factor of $\frac{1}{p}$ (using nearest neighbor upsampling for simplicity), where scaling factor $p = H_i/\frac{H}{32} = W_i/\frac{W}{32}$ is the multiplier to the target size. And then, a specific mask token is initialized for each feature scale and then replicated multiple times to replace the features in the masked area. See Fig. 3. Each mask token is a learned vector and is not shared between different feature scales. Different from the fixed input size in the classification task, the input size of the image is flexible, and this requires the same flexible positional embedding generation. We conduct the position encoding in an absolute scale, *e.g.*, $28 \times 28$, and adaptively interpolate its resolution according to the input image size. We then add the positional embeddings to all tokens in each feature scale and send them to a series of transformer blocks to perform the feature reconstruction task. The final outputs of the transformer blocks are then restored to the original multi-scale resolutions via the spatial recovery module (SRM). In the SRM, we adopt a linear layer to change the channel numbers to $p^2 \times C_i^T$ ($p$ is the same scaling factor used in SAM) and further reshape into the same size as $F_i^T$. The output of SRM is forced to imitate the teacher's output and learns to reconstruct the missing feature. Note that the adaptive decoder is only used during training to perform the masked KD task.

#### 3.1.4 Distillation Target.

Denote the output of SRM as $\hat{F}_i^S \in \mathbb{R}^{C_i^T \times H_i \times W_i}$ where $i$ indicates the feature index in FPN and $C_i^T, H_i, W_i$ indicate the channel number, height and width of the corresponding feature. The feature-based distillation loss can be formulated as follows:

$$\mathcal{L}_{feat} = \frac{1}{2N}\sum_{i=0}^{\mathcal{P}}(F_i^T - \hat{F}_i^S)^2, \tag{3}$$

where $\mathcal{P}$ indicates the total number of features in FPN and $\mathrm{N} = \mathrm{C_i H_i W_i}$. $\hat{F}_i^S$ is the output of $f_{dec}(\phi(F_i^S), \mathcal{T}_{mask})$ in Eq (2). Following the state-of-the-art feature-based distillation method FGD [40], we introduce the global distillation loss $\mathcal{L}_{global}$ to assist the distillation learning, which can be formulated as:

$$\mathcal{L}_{global} = \sum_{i=0}^{\mathcal{P}}(R(F_i^T) - R(\hat{F}_i^S))^2, \tag{4}$$

where $R$ denotes GCBlock in FGD. Note that before computing $\mathcal{L}_{feat}$ and $\mathcal{L}_{global}$ we have $F^S$ and $F^T$ pass a layer normalization operation as well before MSE loss.

To summarize, the overall training loss of the proposed MKD includes (a) task loss $L_{task}$, *e.g.*, classification loss and regression loss in object detection; (b) feature-based distillation loss $L_{feat}$ and $L_{global}$. It can be formulated as follows:

$$\mathcal{L}_{total} = \mathcal{L}_{task} + \lambda\mathcal{L}_{feat} + \gamma\mathcal{L}_{global} \tag{5}$$

where $\lambda$ and $\gamma$ are the weights to balance the distillation losses. The total distillation process is model-agnostic, so it works well with a variety of different structures.

### 3.2. Discussion

Feature-based distillation methods [29, 4, 41], such as hint-based [29, 1], attention-based [45, 40], and instance-based [34, 7, 17], have achieved superior performance improvement. They usually make the student imitate the teacher's output given fully visible image input and design different principles to extract the task-aware knowledge to guide the student's learning. But the redundant information in the teacher's feature hinders the complete knowledge learning of the student.

As shown in Fig. 5, MGD [41] introduces the feature masking scheme to enhance the feature learning process by partially masking pixels in the feature maps from student's FPN. However, the information in the masked part has already been leaked during the forward process of the full image in the backbone. Thus, when minimizing KD loss, the effect of MGD is divided into two parts. First, since each adjacent pixel of the masked pixel already contains the information of the masked pixel, in the process of pushing student and teacher pair closer together, MGD extracts the masked feature from the leaked information in adjacent pixels and uses convolution to restore it. Second, the pixels around the masked pixels are pushed closer to the corresponding teacher. The effect of MGD is therefore dispersed, and even with the adaptive decoder proposed in our MKD, MGD is still not able to establish the connection between the masked and unmasked area as shown in Tab. 1.

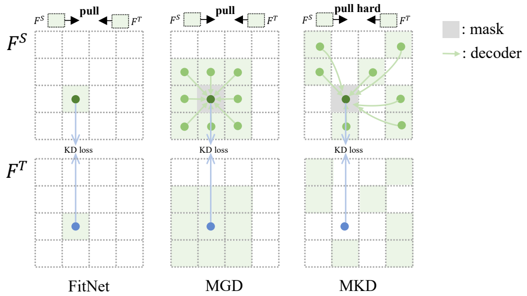

Figure 5: Comparison of different distillation paradigms.

Different from all these paradigms, MKD introduces a global decoder to learn the relationship between the masked

| Method | mAP | AP50 | AP75 |
|---|---|---|---|
| original MGD [41] | 36.6 | 55.1 | 39.1 |
| MGD + decoder in our MKD | 36.6 | 55.0 | 39.1 |

Table 1: Study on the decoder with Retina Res101-Res18.

area and its surroundings, shown in Fig. 5, promoting the learning of complete knowledge in teacher by KD. MKD masks information in the image space throughout the distillation process, which guarantees that the masked area is completely invisible to the student. Since there is no additional information leakage, MKD more strongly pushes the features of the teacher and the student closer during distillation. Moreover, we visualize the self-attention in the transformer decoder blocks in Fig. 6. To restore the feature of the unseen masked region, the student network is driven to fully learn corresponding knowledge from the adjacent areas. Thus, the feature imitation process is accordingly enhanced. To better understand this, we illustrate the L2 distance between the teacher's and student's features in the adjacent regions in Fig. 2. Our MKD can further enhance the student's learning and demonstrates superior improvement over the previous KD paradigms as shown in Tab. 3 and Tab. 2.

Figure 6: The visualization of attention in decoder blocks. The square in red denotes the query.

# 4. Experiments

In order to evaluate the performance of our proposed method on object detection tasks, we first conduct experiments on the MS COCO dataset [22], which consists of 80 object categories and over 120k images in total. We use 120k images for training and 5k images for testing, following the most common setting. We take mean average precision (mAP) as the evaluation metric of all the detectors. As for the semantic segmentation task, we evaluate our distillation method on the Cityscapes dataset [6] with 5k high-quality images and evaluate the performance with mean Intersection-over-Union (mIoU). The shown results are the average value of three runs.

| Method | Schedule | mAP | AP50 | AP75 |
|---|---|---|---|---|
| RetinaNet-ResX101(T) | 2x | 41.0 | 60.9 | 44.0 |
| RetinaNet-Res50(S) | 2x | 37.4 | 56.7 | 39.6 |
| FKD [45] | 2x | 39.6 (+2.2) | 58.8 | 42.1 |
| FGD [40] | 2x | 40.4 (+3.0) | 59.9 | 43.3 |
| MGD [41] | 2x | 40.7 (+3.3) | 59.4 | 42.8 |
| MGD* [41] | 2x | 41.0 (+3.6) | 60.7 | 44.0 |
| ours | 2x | 41.1 (+3.7) | 60.6 | 44.0 |
| ours* | 2x | **41.5 (+4.1)** | **61.1** | **44.3** |
| RetinaNet-Res101(T) | 2x | 38.1 | 58.3 | 40.9 |
| RetinaNet-MBV2(S) | 2x | 31.0 | 48.9 | 32.7 |
| GID [7] | 2x | 33.5 (+2.5) | 51.9 | 35.5 |
| FGD [40] | 2x | 35.6 (+4.6) | 53.8 | 38.1 |
| MGD [41] | 2x | 35.5 (+4.5) | | |
| ours | 2x | **36.0 (+5.0)** | **54.5** | **38.1** |
| RetinaNet-Swin-s(T) | 1x | 39.7 | 60.1 | 42.2 |
| RetinaNet-Swin-t(S) | 1x | 37.2 | 57.4 | 39.1 |
| FGD [40] | 1x | 38.6 (+1.4) | 58.6 | 40.9 |
| F=MGD [41] | 1x | 38.1 (+0.9) | | |
| ours | 1x | **39.3 (+2.1)** | **59.5** | **41.6** |

Table 2: Results on COCO dataset with large backbones and swin transformer backbones [26]. T and S denote the teacher and the student, respectively.

**Implementation Details.** We train the student models with a batch size of 16 for 24 epochs (known as a 2× schedule). The initial learning rate is set by 0.01 for one-stage detectors and 0.02 for two-stage detectors. We reduce the learning rate by 0.1× at the 16th and 21st epochs. We adopt an early-stop mask strategy in the last 2 epochs, in which we set the masking ratio as 0.2 with a learning rate of 0.001 and without supervision from the teacher. This follows the cosine schedule down to 0. In this way, the input distribution of the whole image is consistent with the test image.

We use SGD as the optimizer and set the momentum and weight decay by 0.9 and 0.0001, respectively. We use the masking ratio of 0.1 by default and set the size of masked patches by 32. The number of adaptive decoder layers is 4. Transformer blocks in the decoder have 256 channels with a head number of 8, and the mlp ratio is set to be 8. The loss weight in Eq. 5 are set to be $\{\lambda = 3, \gamma = 3e-6\}$ in single-stage detectors, $\{\lambda = 3, \gamma = 3e-7\}$ in two-stage detectors, and $\{\lambda = 7, \gamma = 0\}$ in segmentation. All the experiments are conducted on 8 GPUs with mmdetection [2] and mmsegmentation [5] on PyTorch.

## 4.1. Main Results

**Object detection.** To verify the effectiveness of the MKD, we conduct experiments on popular detectors, i.e., RetinaNet [21], Faster R-CNN [28] and Reppoints [43] with various backbones, and the results are shown in Tab. 3 and Tab. 2. We can observe that the proposed MKD can consistently improve the student model by more than 2.5 mAP. For example, the Faster R-CNN with ResNet 50 can achieve 41.1 mAP when mimicking Faster R-CNN with ResNet 101 with MKD, which even surpasses the teacher. We further compare our method with the state-of-the-art for detection distillation under fair settings. Our MKD surpasses FGD [40] and MGD [41] on various student-teacher pairs. When distilling the RetinaNet-Res50 model with RetinaNet-ResX101 as the teacher, MKD transcends MGD with 0.4 on mAP and 1.2 on AP50. A similar trend can be observed when we choose smaller students like ResNet 18 as the backbone.

| Method (The default training schedule is 2x) | mAP | AP50 | AP75 | APs | APm | APl |
|---|---|---|---|---|---|---|
| RetinaNet-Res101(Teacher network) | 38.9 | 58.0 | 41.5 | 21.0 | 42.8 | 52.4 |
| RetinaNet-Res18(Student network) | 33.4 | 51.8 | 35.1 | 16.9 | 35.6 | 44.9 |
| FKD [45] | 35.9 (+2.5) | 54.4 | 38.0 | 17.9 | 39.1 | 49.1 |
| FGD [40] | 36.2 (+2.8) | 54.7 | 38.6 | 19.5 | 40.1 | 48.4 |
| MGD [41] | 36.6 (+3.2) | 55.1 | 39.1 | 19.5 | 40.3 | 50.7 |
| ours | 37.3 (+3.9) | **56.1** | **39.9** | 19.0 | 41.0 | **51.6** |
| ours*(with the parameter inheriting scheme [17] ) | **37.5 (+4.1)** | 56.0 | **39.9** | **19.4** | **41.5** | 51.2 |
| RetinaNet-Res101(Teacher network) | 38.9 | 58.0 | 41.5 | 21.0 | 42.8 | 52.4 |
| RetinaNet-Res50(Student network) | 37.4 | 56.7 | 39.6 | 20.0 | 40.7 | 49.7 |
| GID [7] | 39.1 (+1.7) | 59.0 | 42.3 | 22.8 | 43.1 | 52.3 |
| FGD [40] | 39.6 (+2.2) | 58.5 | 42.5 | 22.9 | 43.7 | 53.6 |
| MGD [41] | 39.5 (+2.1) | 58.5 | 42.2 | 21.3 | 43.6 | 53.4 |
| ours | 39.9 (+2.5) | 59.0 | 42.7 | 22.3 | 43.9 | 53.3 |
| ours*(with the parameter inheriting scheme [17] ) | **40.2 (+2.8)** | **59.3** | **43.0** | **22.3** | **44.4** | **54.0** |
| FasterRCNN-Res101(Teacher network) | 39.8 | 60.1 | 43.3 | 22.5 | 43.6 | 52.8 |
| FasterRCNN-Res50(Student network) | 38.4 | 59.0 | 42.0 | 21.5 | 42.1 | 50.3 |
| FGFI [34] | 39.3 (+0.9) | 59.8 | 42.9 | 22.5 | 42.3 | 52.2 |
| GID [7] | 40.2 (+1.8) | 60.7 | 43.8 | 22.7 | 44.0 | 53.2 |
| FGD [40] | 40.4 (+2.0) | 60.7 | 44.3 | 22.8 | 44.5 | 53.5 |
| MGD [41] | 40.1 (+1.7) | 60.2 | 43.6 | 22.9 | 44.2 | 53.3 |
| ours | 41.0 (+2.6) | **61.5** | 44.7 | 23.5 | **45.5** | 53.4 |
| ours*(with the parameter inheriting scheme [17] ) | **41.1 (+2.7)** | **61.5** | **44.8** | **24.4** | 45.1 | **54.3** |
| FCOS-Res101(Teacher network) | 40.8 | 60.0 | 44.0 | 24.2 | 44.3 | 52.4 |
| FCOS-Res50(Student network) | 38.5 | 57.7 | 41.0 | 21.9 | 42.8 | 48.6 |
| GID [7] | 42.0 (+3.5) | 60.4 | 45.5 | 25.6 | 45.8 | 54.2 |
| FGD [40] | 42.1 (+3.6) | - | - | 27.0 | 46.0 | 54.6 |
| MGD [41] | 42.2(+3.7) | 60.9 | 45.3 | 26.6 | 46.3 | 54.7 |
| ours | 42.5 (+4.0) | 61.2 | 46.1 | 26.6 | 46.8 | 54.6 |
| ours*(with the parameter inheriting scheme [17] ) | **43.1 (+4.6)** | **61.7** | **46.7** | **27.3** | **47.1** | **55.1** |
| Reppoints-ResX101(Teacher network) | 44.2 | 65.5 | 47.8 | 26.2 | 48.4 | 58.5 |
| Reppoints-Res50(Student network) | 38.6 | 59.6 | 41.6 | 22.5 | 42.2 | 50.4 |
| FKD [45] | 40.6 (+2.0) | 61.7 | 43.8 | 23.4 | 44.6 | 53.0 |
| FGD [40] | 41.3 (+2.7) | - | - | 24.5 | 45.2 | 54.0 |
| MGD [41] | 41.8 (+3.2) | 62.8 | 44.8 | 24.2 | 45.8 | 55.6 |
| MGD [41]*(with the parameter inheriting scheme [17] ) | 42.3 (+3.7) | 63.3 | 45.4 | 24.4 | 46.2 | 55.9 |
| ours | 42.2 (+3.6) | 63.0 | 45.6 | 24.3 | 46.4 | 55.7 |
| ours*(with the parameter inheriting scheme [17] ) | **43.0 (+4.4)** | **63.8** | **46.1** | **24.4** | **47.2** | **57.1** |

Table 3: Object detection performance of our proposed MKD with various teacher-student pairs on the COCO dataset.

Moreover, our method can benefit from a longer training schedule. For example, when training for 36 epochs, our MKD can further improve the performance of the student model over 4.0 mAP, which surpasses other methods by a larger margin than that for 12 epochs. This demonstrates that introducing a masked autoencoder into knowledge distillation can help the student avoid early saturation and benefit more from the distillation process.

**Semantic segmentation.** Our MKD can be easily transferred to segmentation tasks such as semantic segmentation, and we conduct knowledge distillation experiments to show the generality of the proposed method. We choose PspNet-Res101 [47] as the teacher to distill PspNet-Res18 and DeepLabV3-Res18 [3]. As shown in Tab. 5, our method improves the PspNet-Res18 model with 4.85 mIoU, which greatly fills the gap between the student and the teacher. Also, MKD surpasses MGD and other methods by a significant margin, which verifies the effectiveness and generality of our method.

**Image classification.** Since the core bottleneck is particularly prominent in fine-grained visual tasks that rely on spatial localization information, we mainly evaluate our methods on them. We also present the result of MKD on the classification task in Tab. 6, which further verifies its effectiveness.

## 4.2. Ablation Study

**Masking ratio.** We first examine the influence of different masking ratios on the performance shown in Fig. 7. Large masking ratios benefit more from longer training schedules. For example, a masking ratio of 0.1 achieves the best under a 1x schedule, while the best one changes to 0.3 under a 3x schedule. This is because a larger masking ratio introduces more difficulty to the distillation process and needs more

| Patch Size | mAP | AP50 | AP75 |
|---|---|---|---|
| Baseline | 37.4 | 56.7 | 39.6 |
| 16 | 39.8 | 59.2 | 42.5 |
| 32 | **39.9** | 59.3 | 42.5 |
| 64 | 39.8 | 59.0 | 42.7 |

(a) The patch size.

| Resolution | mAP | AP50 | AP75 |
|---|---|---|---|
| w/o SAM, SRM | OOM | - | - |
| $[H/16, W/16]$ | 38.9 | 58.0 | 41.6 |
| $[H/32, W/32]$ | **39.9** | 59.3 | 42.5 |
| $[H/64, W/64]$ | 39.7 | 59.1 | 42.2 |

(b) The aligned resolution in SAM.

| $\lambda$ | mAP | AP50 | AP75 |
|---|---|---|---|
| 0 | 39.6 | 58.8 | 42.5 |
| 1 | **39.9** | 59.2 | 42.8 |
| 3 | **39.9** | 59.3 | 42.5 |
| 5 | 39.7 | 59.0 | 42.6 |

(c) The loss weights $\lambda$.

| $\gamma$ | mAP | AP50 | AP75 |
|---|---|---|---|
| 0 | 39.7 | 59.0 | 42.6 |
| 1e-6 | 39.7 | 58.9 | 42.2 |
| 3e-6 | **39.9** | 59.3 | 42.5 |
| 5e-6 | 39.8 | 59.2 | 42.6 |

(d) The loss weights $\gamma$.

| Depth | mAP | AP50 | AP75 |
|---|---|---|---|
| Baseline | 37.4 | 56.7 | 39.6 |
| 2 | 39.6 | 58.7 | 42.4 |
| 4 | **39.9** | 59.3 | 42.5 |
| 8 | 39.8 | 59.0 | 42.3 |

(e) The depth of the adaptive decoder $N$.

| Masked Convolution | mAP | AP50 | AP75 |
|---|---|---|---|
| Baseline | 37.4 | 56.7 | 39.6 |
| w/o | 39.6 | 58.8 | 42.5 |
| w/ | **39.9** | 59.3 | 42.5 |

(f) The masked convolution.

| Mask Strategy | mAP | AP50 | AP75 |
|---|---|---|---|
| Baseline | 37.4 | 56.7 | 39.6 |
| Random | **39.9** | 59.3 | 42.5 |
| Grid | 37.8 | 57.0 | 40.2 |
| Block | 39.7 | 59.2 | 42.7 |

(g) The masking strategy.

Table 4: Ablation study on the MKD design with RetinaNet-ResNet50 as student and RetinaNet-ResNeXt101 as a teacher under 1x training schedule. If not specified, the patch size is 32, $\lambda = 3, \gamma = 3e - 6$, the decoder has a depth of 4, and the mask is randomly sampled.

| Method | inputsize | mIoU |
|---|---|---|
| PspNet-Res101(T) | 512x1024 | 78.34 |
| PspNet-Res18(S) | 512x512 | 69.85 |
| SKDS [25] | 512x512 | 72.70 |
| CWD [30] | 512x512 | 73.53 |
| MGD [41] | 512x512 | 73.63 |
| MGD* [41] | 512x512 | 74.10 |
| ours | 512x512 | 73.99 |
| ours* | 512x512 | **74.70** |
| PspNet-Res101(T) | 512x1024 | 78.34 |
| DeepLabV3-Res18(S) | 512x512 | 73.20 |
| SKDS [25] | 512x512 | 73.87 |
| CWD [30] | 512x512 | 75.93 |
| MGD [41] | 512x512 | 76.02 |
| MGD* [41] | 512x512 | 76.31 |
| ours | 512x512 | 76.14 |
| ours* | 512x512 | **76.63** |

Table 5: Semantic segmentation result on Cityscape dataset. * means adopting an extra logits loss from CWD [30].

| Method | Top-1 | Top-5 |
|---|---|---|
| ResNet-34(T) | 73.62 | 91.59 |
| ResNet-18(S) | 69.90 | 89.43 |
| FitNet [29] | 71.50 | 90.27 |
| SRRL [39] | 71.73 | 90.60 |
| MGD [41] | 71.80 | 90.40 |
| Ours | 72.01 | 90.43 |

Table 6: The performance of classification on ImageNet.

optimization steps for the student to converge.

**Masked convolution.** To keep the masked patches not affected by others in the convolution block, we introduce the masked convolution in our MKD, and the results of ablation on masked convolution are shown in Tab. 4f. Removing masked convolution from MKD will lead to a reduction of 0.3 mAP, which is still better than that of the baseline. This demonstrates that the masking operation indeed hinders the optimization of convolution blocks, and adopting masked convolution can effectively alleviate this issue.

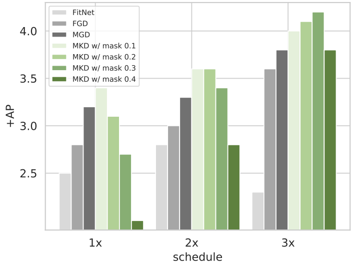

Figure 7: Illustration of the performance improvement of different masking ratios under different schedules. It appears that MKD (green) brings the greatest boost to the baseline (36.5mAP).

**Mask patch size.** Here we examine how the masked patch size influences the distillation process. As shown in Tab. 4a, a patch size of 32 performs the best than other options. A smaller patch will lead to the confusing assignment of masked position in the last stage, which leads to inferior performance. On the contrary, using a larger patch size may distort the objects too much and make the edge vague. So we use a masked patch size of 32 by default in our work.

**SAM and SRM.** We compare different aligned resolutions in SAM, and the results are shown in Tab. 4b. The training process appears to be out of memory(OOM) without spatial alignment operation, indicating that SAM and SRM are important in reducing computation and memory. We further found that setting the target resolution as $\frac{H}{32} \times \frac{W}{32}$ performs the best because it is the same size as the masked patches.

**Mask sampling strategies.** Here we examine the influence of different masking strategies, *e.g.*, masking randomly, masking in grids, and masking by block. Recent masked image modeling works show that random strategy works the best, and we observe a similar trend in our MKD. Note that to prevent the student network from learning the mask pattern, it is necessary to use a global random mask. As shown in Tab. 4g, random masking surpasses other ways and is selected in other experiments in this paper.

**Adaptive decoder design.** Adaptive decoder helps to reconstruct masked features from visible ones and is an important module in MKD. We choose different numbers of decoder layers and examine their effects on distillation performance. As shown in Tab. 4e, four layers perform the best. Due to the feature dimension and length of the sequence, we use four layers by default. Objects of different scales correspond to different feature levels of FPN, so we use unshared tokens for the reconstruction. While shared tokens cause a performance drop from 37.3 mAP to 37.0 mAP.

**Loss weight.** In Eq. 5, we use two hyper-parameters $\lambda$ and $\gamma$ to balance the distillation loss, and we conduct ablation studies to investigate their sensitivities of them. From Tab. 4d and 4c, it can be seen that the performance of the student model is not sensitive to the choices of $\lambda$ and $\gamma$, as the gap between the worst and the best is within 0.2 mAP. Note that $L_{feat}$ and $L_{global}$ are both distillation losses to calculate the similarity between the reconstructed output of the decoder $\hat{F}^S$ and teacher feature $F^T$. The main difference is that $L_{feat}$ directly calculates MSE, and $L_{global}$ goes through a GCblock. Hence, the performance drop is small when $\lambda = 0$ or $\gamma = 0$, as MKD already works when using either of the loss alone.

## 5. Conclusion

In this paper, we propose a new distillation paradigm Masked Knowledge Distillation (MKD), which introduces the masked autoencoding scheme to enhance the knowledge distillation process. MKD takes the masked image as input and predicts the whole feature map that is used to imitate the corresponding feature of the teacher network. By forcing the student network to learn the knowledge of the nearby region of the masked part to recover the full feature, the student can better transfer the corresponding information from the teacher, which helps to excavate the potential of the small student network. Extensive experimental results on various fine-grained visual tasks show its power in enhancing the improvement of knowledge distillation.

## Acknowledgement

This work was partly supported by the National Natural Science Foundation of China (Grant No. 61991451) and the Shenzhen Science and Technology Program (JCYJ20220818101001004).

## References

[1] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *NIPS*, pages 742–751, 2017. 2, 3, 5

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

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 7

[4] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Distilling knowledge via knowledge review. In *CVPR*, pages 5008–5017, 2021. 1, 5

[5] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 6

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, pages 3213–3223, 2016. 6

[7] Xing Dai, Zeren Jiang, Zhao Wu, Yiping Bao, Zhicheng Wang, Si Liu, and Erjin Zhou. General instance distillation for object detection. In *CVPR*, pages 7842–7851, 2021. 1, 3, 5, 6, 7

[8] 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. In *ICLR*, 2020. 4

[9] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *ICCV*, pages 6569–6578, 2019. 3

[10] Peng Gao, Teli Ma, Hongsheng Li, Jifeng Dai, and Yu Qiao. Convmae: Masked convolution meets masked autoencoders. *arXiv preprint arXiv:2205.03892*, 2022. 2, 4

[11] Kunyang Han, Yong Liu, Jun Hao Liew, Henghui Ding, Yunchao Wei, Jiajun Liu, Yitong Wang, Yansong Tang, Yujiu Yang, Jiashi Feng, et al. Global knowledge calibration for fast open-vocabulary segmentation. *arXiv preprint arXiv:2303.09181*, 2023. 1

[12] Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. *NIPS*, 28, 2015. 3

[13] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 4

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 3

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1

[16] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *Computer Science*, 14(7):38–39, 2015. 1, 3

[17] Zijian Kang, Peizhen Zhang, Xiangyu Zhang, Jian Sun, and Nanning Zheng. Instance-conditional knowledge distillation for object detection. *NIPS*, 34:16468–16480, 2021. 1, 3, 5, 7

[18] Quanquan Li, Shengying Jin, and Junjie Yan. Mimicking very efficient network for object detection. In *CVPR*, pages 6356–6364, 2017. 1

[19] Rundong Li, Yan Wang, Feng Liang, Hongwei Qin, Junjie Yan, and Rui Fan. Fully quantized network for object detection. In *CVPR*, pages 2810–2819, 2019. 3

[20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 1

[21] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017. 3, 6

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 6

[23] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 2, 4

[24] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 3

[25] Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *CVPR*, pages 2604–2613, 2019. 8

[26] 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 *ICCV*, pages 10012–10022, 2021. 6

[27] Mengye Ren, Andrei Pokrovsky, Bin Yang, and Raquel Urtasun. Sbnet: Sparse blocks network for fast inference. In *CVPR*, pages 8711–8720, 2018. 4

[28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NIPS*, 28, 2015. 3, 6

[29] 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, 3, 5, 8

[30] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *ICCV*, pages 5311–5320, 2021. 8

[31] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *CVPR*, pages 10781–10790, 2020. 3

[32] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *ICCV*, pages 9627–9636, 2019. 3

[33] Jiahao Wang, Mingdeng Cao, Shuwei Shi, Baoyuan Wu, and Yujiu Yang. Attention probe: Vision transformer distillation in the wild. In *ICASSP*, pages 2220–2224. 3

[34] Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *CVPR*, pages 4933–4942, 2019. 1, 2, 3, 5, 7

[35] Xinlong Wang, Rufeng Zhang, Tao Kong, Lei Li, and Chunhua Shen. Solov2: Dynamic and fast instance segmentation. *NIPS*, 33:17721–17732, 2020. 3

[36] Yi Wei, Xinyu Pan, Hongwei Qin, Wanli Ouyang, and Junjie Yan. Quantization mimic: Towards very tiny cnn for object detection. In *ECCV*, pages 267–283, 2018. 3

[37] Taiqiang Wu, Cheng Hou, Zhe Zhao, Shanshan Lao, Jiayi Li, Ngai Wong, and Yujiu Yang. Weight-inherited distillation for task-agnostic bert compression, 2023. 3

[38] Xiang Xiang, Zhiyuan Wang, Shanshan Lao, and Baochang Zhang. Pruning multi-view stereo net for efficient 3d reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 168:17–27, 2020. 3

[39] Jing Yang, Brais Martinez, Adrian Bulat, Georgios Tzimiropoulos, et al. Knowledge distillation via softmax regression representation learning. ICLR, 2021. 8

[40] Zhendong Yang, Zhe Li, Xiaohu Jiang, Yuan Gong, Zehuan Yuan, Danpei Zhao, and Chun Yuan. Focal and global knowledge distillation for detectors. In *CVPR*, pages 4643–4652, 2022. 1, 2, 3, 5, 6, 7

[41] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan. Masked generative distillation. *arXiv preprint arXiv:2205.01529*, 2022. 3, 5, 6, 7, 8

[42] Zhendong Yang, Zhe Li, Ailing Zeng, Zexian Li, Chun Yuan, and Yu Li. Vitkd: Practical guidelines for vit feature knowledge distillation. *arXiv preprint arXiv:2209.02432*, 2022. 3

[43] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In *ICCV*, pages 9657–9666, 2019. 3, 6

[44] Zhendong Yang, Ailing Zeng, Zhe Li, Tianke Zhang, Chun Yuan, and Yu Li. From knowledge distillation to self-knowledge distillation: A unified approach with normalized loss and customized soft labels. *arXiv preprint arXiv:2303.13005*, 2023. 3

[45] Linfeng Zhang and Kaisheng Ma. Improve object detection with feature-based knowledge distillation: Towards accurate and efficient detectors. In *ICLR*, 2020. 1, 3, 5, 6, 7

[46] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *CVPR*, pages 11953–11962, 2022. 1, 2, 3

[47] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 7