Rethinking Knowledge Distillation with Raw Features for Semantic Segmentation

Anonymous authors
Paper under double-blind review

Abstract

Most existing knowledge distillation methods for semantic segmentation focus on extracting various complex forms of knowledge from raw features. However, such knowledge is usually manually designed and relies on prior knowledge as in traditional feature engineering. In this paper, in order to seek a more simple and effective way to perform feature distillation, we analyze the naive feature distillation method with raw features and reveal that it actually attempts to make the student learn both the magnitude and angular information from the teacher features simultaneously. We further find experimentally that the angular information is more effective than the magnitude information for feature distillation. Based on this finding, we propose a simple and effective feature distillation method for semantic segmentation, which eliminates the need to manually design distillation knowledge. Experimental results on three popular benchmark datasets show that our method achieves state-of-the-art distillation performance for semantic segmentation. The code will be available.

1 Introduction

Recent works on backbones (He et al., 2016; Wang et al., 2021; Zhang et al., 2020) and segmentation frameworks Zhao et al. (2017); Chen et al. (2018); Yuan et al. (2020) have greatly improved the performance of semantic segmentation. However, these high-performance models often require a lot of memory and computational overhead. Lightweight models are preferred in real-time applications due to limited resources. As a result, there is growing interest in how to reduce the model size while maintaining decent performance.

The knowledge distillation (KD) introduced by Hinton et al. (2015) was proven to be a promising way to solve this problem. Its key idea is to transfer the knowledge from a cumbersome model (teacher) to a compact one (student). Hinton et al. (2015) define the knowledge as soft labels produced by the teacher and supervise the student with both ground truth labels and soft labels. FitNets (Romero et al., 2015) extends this idea to intermediate representation of the model by making the student directly mimic the teacher's hidden layer features. Inspired by this, many feature-based KD methods emerged later. Instead of distilling the raw features as in FitNets (Romero et al., 2015), most existing feature-based methods prefer to extract various forms of knowledge from raw features, such as attention map (Zagoruyko & Komodakis, 2017), Gramian matrix (Yim et al., 2017), pairwise similarity (Liu et al., 2019) and low-level texture knowledge (Ji et al., 2022). However, these complex forms of knowledge are usually manually designed and rely on various prior knowledge as in traditional feature engineering.

In this paper, we first analyze the feature distillation method proposed in FitNets (Romero et al., 2015) and reveal that distilling the raw features is equivalent to making the student learn both the magnitude and angular information from the teacher features simultaneously, which we believe may account for the limited performance improvement of FitNets (Romero et al., 2015). We therefore design two kinds of feature distillation methods, Magnitude Distillation and Angular Distillation, to decouple the learning of magnitude and angular information. Magnitude Distillation makes the student learn only the magnitude information from the teacher features, while Angular Distillation focuses on the angular information. The experimental results for semantic segmentation show that Angular Distillation achieves significantly better results than Magnitude Distillation and FitNets method (Romero et al., 2015). Therefore, we argue that decoupling the learning of magnitude and angular information is important, and the angular information is more effective than magnitude information for feature distillation. We evaluate our method for semantic segmentation on Cityscapes, Pascal VOC, and ADE20K. Experimental results show that our method outperforms existing KD methods for semantic segmentation by a large margin.

Our main contributions are summarized as follows:

- We take a closer look at the classical KD method FitNets and reveal that it enforces both the magnitude and angular differences between teacher and student features to be as small as possible. However, we find experimentally that the angular information is more effective for feature distillation than the magnitude information.
- We propose a simple and effective feature distillation method for semantic segmentation, which achieves state-of-the-art distillation performance on three popular benchmark datasets.
- Our method distills with raw features and hence does not rely on manually designed distillation knowledge. In addition, ablation studies show that our method performs well at different distillation positions and is robust to hyper-parameters.

2 Related work

2.1 Knowledge Distillation

Existing KD methods can be roughly divided into logits-based, feature-based and relation-based according to the type of knowledge. Logits-based methods transfer class probabilities produced from the teacher as soft labels to supervise the student. Feature-based methods take the feature maps of intermediate layers as knowledge. Relation-based methods focus on the relationships between different layers or data samples.

Among these methods, the feature-based methods are more related to this paper. FitNets (Romero et al., 2015) is the first KD method to take the features of the intermediate layers as knowledge. After that, many methods focusing on different aspects of feature distillation have been proposed, such as designing various forms of new knowledge from raw features (Zagoruyko & Komodakis, 2017; Passalis & Tefas, 2018), changing the teacher's or student's training strategies to facilitate distillation (Jin et al., 2019; Zhu & Wang, 2021), and adaptively utilizing multiple layers of features for distillation (Chen et al., 2021; Ji et al., 2021). Differently, we revisit the naive feature distillation method introduced in FitNets and analyze the possible reasons for its limited performance.

2.2 Knowledge Distillation for semantic segmentation

Applying KD methods for image classification to semantic segmentation in a straightforward way may not yield satisfactory results. As a result, some KD methods tailored for semantic segmentation have been proposed. Xie et al. (2018) use the local similarity between a pixel and its 8 neighbors on the feature map as knowledge. Liu et al. (2019) distill the long-range dependency by computing the pairwise similarity on the feature map and enforce high-order consistency between the outputs of the teacher and student through adversarial learning. Wang et al. (2020) propose to transfer the intra-class feature variation of the teacher to student. Shu et al. (2021) focus on channel information by softly aligning the activation of each channel between the teacher and student, which is more effective on logits than on features. Unlike these methods, our method does not rely on manually designed distillation knowledge and tedious distillation strategies such as adversarial learning. Extensive experiments on semantic segmentation show that our method outperforms existing methods by a large margin, demonstrating its simplicity and effectiveness.

3 Method

3.1 Analysis of Naive Feature Distillation

In this paper, we refer to the KD method proposed by FitNets (Romero et al., 2015) as Naive Feature Distillation. It encourages the student to have the same feature activation as the teacher. Let $\boldsymbol{F}^s \in$

$\mathbb{R}^{C\times H\times W}$ and $\boldsymbol{F}^t \in \mathbb{R}^{C\times H\times W}$ denote the feature maps of the student and teacher, respectively, where C is the number of channels, H and W are the height and width. Naive Feature Distillation minimizes the following loss:

$$\mathcal{L}_{naive} = MSE(\boldsymbol{F}^s, \boldsymbol{F}^t) = \frac{1}{N}||\boldsymbol{F}^t - \boldsymbol{F}^s||^2 = \frac{1}{N}\sum_{i=1}^{N}(\boldsymbol{F}_i^t - \boldsymbol{F}_i^s)^2 \tag{1}$$

where $N = C \times H \times W$. Although such feature distillation is considered as a promising approach in KD, inspiring a line of subsequent feature-based methods, its performance improvement is not significant.

We believe the raw features of the teacher should contain enough information to guide the student. So instead of extracting complex forms of knowledge from raw features as other methods do, we tend to dive into the Naive Feature Distillation method, seeking a more simple and effective way to perform feature distillation. From the perspective of vectors, we can reformulate $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$ as:

$$\begin{aligned}\boldsymbol{F}^s &= ||\boldsymbol{F}^s||\hat{\boldsymbol{x}} = n\hat{\boldsymbol{x}} \\ \boldsymbol{F}^t &= ||\boldsymbol{F}^t||\hat{\boldsymbol{y}} = m\hat{\boldsymbol{y}}\end{aligned} \tag{2}$$

where n and m denote the magnitudes of $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, respectively, and $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ are unit vectors denoting the directions of $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, respectively. Then $\mathcal{L}_{naive}$ in Eq. 1 can be reformulated as:

$$\begin{aligned}\mathcal{L}_{naive} &= \frac{1}{N}||m\hat{\boldsymbol{y}} - n\hat{\boldsymbol{x}}||^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(m\hat{\boldsymbol{y}}_i - n\hat{\boldsymbol{x}}_i)^2 \\ &= \frac{1}{N}(m^2\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i^2 + n^2\sum_{i=1}^{N}\hat{\boldsymbol{x}}_i^2 - 2mn\sum_{i=1}^{N}\hat{\boldsymbol{y}}_i\hat{\boldsymbol{x}}_i) \\ &= \frac{1}{N}(m^2||\hat{\boldsymbol{y}}||^2 + n^2||\hat{\boldsymbol{x}}||^2 - 2mn\,\hat{\boldsymbol{y}}\cdot\hat{\boldsymbol{x}}) \\ &= \frac{1}{N}(m^2 + n^2 - 2mn\cos\theta) \\ &= \frac{1}{N}[(m-n)^2 + 2mn(1-\cos\theta)]\end{aligned} \tag{3}$$

where θ denotes the angle between $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$, i.e. the angle between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$.

The first item in Eq. 3 minimizes the magnitude difference between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, and the second item minimizes the angular difference between $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$. $\mathcal{L}_{naive}$ reaches its minimum when both $m = n$ and $\theta = 0$ are satisfied. It means that the Naive Feature Distillation enforces the student to learn both the magnitude and angular information from the teacher features simultaneously. We believe that minimizing both magnitude and angular differences between student and teacher features is difficult, perhaps even unnecessary. It may explain the limited performance improvement of Naive Feature Distillation and inspires us to investigate the relative importance of the magnitude and angular information of the feature for distillation.

3.2 Decoupled distillation

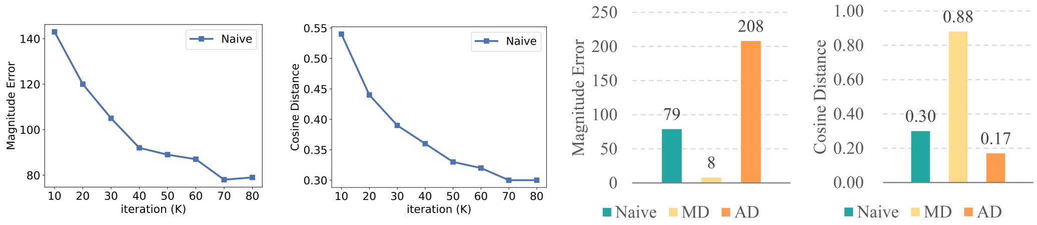
Based on the above analysis, we intend to decouple the learning of magnitude and angular information, as well as to investigate which part of the information is more effective for feature distillation. Therefore, we design two feature distillation methods Magnitude Distillation and Angular Distillation.

The aim of Magnitude Distillation is to make the student focus on learning the magnitude information from the teacher features. Based on the first item in Eq. 3, we adopt the following loss for Magnitude Distillation:

$$\mathcal{L}_{md} = (||\boldsymbol{F}^t|| - ||\boldsymbol{F}^s||)^2 \tag{4}$$

Table 1: Experiments about decoupling the learning of magnitude and angular information on Cityscapes validation set. “T” and “S” denote the teacher and student, respectively. “Naive” denote Naive Feature Distillation. “–” means training without KD.

Model	KD Method	mIoU (%)
T: PSPNet-R101	–	79.76
S: PSPNet-R18	–	72.65
	Naive (Romero et al., 2015)	74.50
	Magnitude Distillation	74.51
	Angular Distillation	**76.86**



(a) (b) (c) (d)

Figure 1: Magnitude and angular differences between teacher (PSPNet-R101) and student (PSPNet-R18) features on Cityscapes validation set. “Naive”, “MD”, and “AD” denote Naive Feature Distillation, Magnitude Distillation, and Angular Distillation, respectively. (a) and (b) show the magnitude error and cosine distance between teacher and student features during training for the Naive Feature Distillation, respectively. (c) shows the magnitude error between teacher and student features, where smaller values indicate smaller magnitude differences. (d) shows the cosine distance between teacher and student features, where smaller values indicate smaller angular differences.

As a result, Magnitude Distillation minimizes only the magnitude difference between teacher and student features.

In contrast, Angular Distillation aims to make the student focus on learning the angular information from the teacher features. According to Eq. 3, with $m = n = 1$, we can eliminate the magnitude difference term in $\mathcal{L}_{naive}$, leaving only the angular difference term. $m = n = 1$ can be satisfied by applying L2 normalization to $\boldsymbol{F}^s$ and $\boldsymbol{F}^t$, so we adopt the following loss for Angular Distillation:

$$\mathcal{L}_{ad} = MSE(\frac{\boldsymbol{F}^s}{||\boldsymbol{F}^s||}, \frac{\boldsymbol{F}^t}{||\boldsymbol{F}^t||}) \tag{5}$$

Similar to Eq. 3, based on Eq. 2, $\mathcal{L}_{ad}$ is essentially equivalent to:

$$\mathcal{L}_{ad} = \frac{2}{N}(1 - \cos\theta) \tag{6}$$

Obviously, Angular Distillation minimizes only the angular difference between teacher and student features.

We conduct experiments on these three distillation methods, namely Naive Feature Distillation, Magnitude Distillation and Angular Distillation, on Cityscapes for semantic segmentation. The distillation results are listed in Table 1, and the magnitude and angular differences between teacher and student features are shown in Figure 1. We can observe that: 1) Naive Feature Distillation, trying to minimize both the magnitude and angular differences between teacher and student features (see Figure 1 (a) and (b)), fails to achieve a smaller magnitude difference than Magnitude Distillation (see Figure 1 (c)), and also fails to achieve a smaller angular difference than Angular Distillation (see Figure 1 (d)), and 2) Angular Distillation achieves significantly better performance than the other two methods. We therefore conclude that *decoupling the learning of magnitude and angular information is important, and the angular information is more effective than magnitude information for feature distillation*.

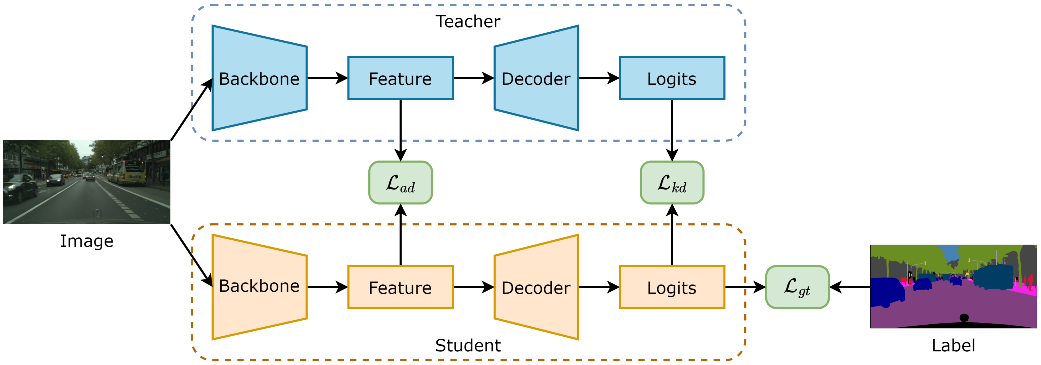


Figure 2: The pipeline of applying our method for semantic segmentation. $\mathcal{L}_{ad}$ is the proposed Angular Distillation loss on features. $\mathcal{L}_{kd}$ is the conventional KD loss on logits. $\mathcal{L}_{gt}$ is the cross-entropy loss for semantic segmentation.

Notably, the above Angular Distillation treats the entire features of a layer from teacher or student as a vector, minimizing the angular difference between them. We call this Layer-wise Angular Distillation (LAD), which minimizes the loss in Eq. 5, i.e.,

$$\mathcal{L}_{lad} = \mathcal{L}_{ad} \tag{7}$$

In addition, we can treat the features in each channel as a vector, and minimize the angular difference between features of the same channel from teacher and student. We call this Channel-wise Angular Distillation (CAD). Based on Eq. 5, its loss is

$$\mathcal{L}_{cad} = \frac{1}{C}\sum_{c=1}^{C} MSE(\frac{\boldsymbol{F}^s_{c,:,:}}{||\boldsymbol{F}^s_{c,:,:}||}, \frac{\boldsymbol{F}^t_{c,:,:}}{||\boldsymbol{F}^t_{c,:,:}||}) \tag{8}$$

where $\boldsymbol{F}_{c,:,:} \in \mathbb{R}^{H\times W}$ denotes the feature vector of the c-th channel. Furthermore, we can also treat the features in each spacial point as a vector, and minimize the angular difference between features of the same spacial point from teacher and student. We call this Point-wise Angular Distillation (PAD), and its loss is as follows:

$$\mathcal{L}_{pad} = \frac{1}{H\times W}\sum_{h=1}^{H}\sum_{w=1}^{W} MSE(\frac{\boldsymbol{F}^s_{:,h,w}}{||\boldsymbol{F}^s_{:,h,w}||}, \frac{\boldsymbol{F}^t_{:,h,w}}{||\boldsymbol{F}^t_{:,h,w}||}) \tag{9}$$

where $\boldsymbol{F}_{:,h,w} \in \mathbb{R}^{C}$ denotes the feature vector of the spacial point (h, w). The difference between LAD, CAD and PAD is that the dimensions of the angular information used for distillation are different. We compare these three Angular Distillation methods in Section 4.4.1.

The pipeline of applying our Angular Distillation for semantic segmentation is shown in Figure 2. Following the previous methods (Liu et al., 2019; Wang et al., 2020; Shu et al., 2021), we apply the conventional KD loss (Hinton et al., 2015) on logits as well. Therefore, the total loss of our method is as follows:

$$\mathcal{L} = \lambda_{ad}\mathcal{L}_{ad} + \lambda_{kd}\mathcal{L}_{kd} + \mathcal{L}_{gt} \tag{10}$$

where $\mathcal{L}_{ad}$ is our Angular Distillation loss, which can be $\mathcal{L}_{lad}$, $\mathcal{L}_{cad}$ or $\mathcal{L}_{pad}$. $\mathcal{L}_{kd}$ is the conventional KD loss (Hinton et al., 2015) on logits, and $\mathcal{L}_{gt}$ is the cross-entropy loss for semantic segmentation.

4 Experiments

4.1 Datasets

Cityscapes. The Cityscapes (Cordts et al., 2016) is a large-scale dataset for semantic urban scene understanding, with high quality pixel-level annotations of 5000 images in addition to a larger set of 19998 coarsely annotated images. It contains 30 classes, and 19 of them are used for evaluation. We only use the finely annotated 2975 images for training and 500 images for validation.

Table 2: Comparison with state-of-the-art methods on validation sets of Cityscapes, Pascal VOC and ADE20K. “T” and “S” denote the teacher and student, respectively. “R101”, “R18” and “MV2” denote ResNet101, ResNet18 and MobileNetV2, respectively. “–” means training without KD.

Model	**Method**	**mIoU (%)**		
		Cityscapes	**VOC 2012**	**ADE20K**
T: PSPNet-R101	–	79.76	78.52	44.39
S: PSPNet-R18	–	72.65	71.35	35.03
	SKD (Liu et al., 2019)	74.23	72.01	35.26
	IFVD (Wang et al., 2020)	74.55	72.00	35.92
	CWD (Shu et al., 2021)	75.91	73.07	36.78
	LAD (Ours)	**76.86**	**75.74**	**39.63**
S: PSPNet-MV2	–	72.73	69.14	33.33
	SKD (Liu et al., 2019)	72.90	69.62	33.39
	IFVD (Wang et al., 2020)	73.74	69.45	33.85
	CWD (Shu et al., 2021)	74.73	71.28	35.26
	LAD (Ours)	**75.76**	**74.13**	**38.92**
S: DeepLabV3-R18	–	74.96	71.98	37.19
	SKD (Liu et al., 2019)	75.32	73.03	36.91
	IFVD (Wang et al., 2020)	76.01	72.87	37.66
	CWD (Shu et al., 2021)	77.13	73.78	38.64
	LAD (Ours)	**77.23**	**76.33**	**41.12**
S: DeepLabV3-MV2	–	73.98	69.92	35.14
	SKD (Liu et al., 2019)	75.78	70.13	35.11
	IFVD (Wang et al., 2020)	75.24	70.32	35.35
	CWD (Shu et al., 2021)	76.59	71.68	36.49
	LAD (Ours)	**77.47**	**74.93**	**39.66**

Pascal VOC. The Pascal VOC (Everingham et al., 2010) dataset contains 20 common objects and one background class with annotations on daily captured photos. We use the augmented dataset with extra coarse annotations provided by Hariharan et al. (2011) resulting in 10582 and 1449 images for training and validation.

ADE20K. The ADE20K (Zhou et al., 2017) is a densely annotated dataset with the instances of stuff, objects, and parts, covering a diverse set of visual concepts in scenes. It contains 150 classes and is divided into 20210 and 2000 images for training and validation. It is challenging due to its large number of classes and existence of multiple small objects in complex scenes.

4.2 Implementation details

4.2.1 Network architectures

Following the previous methods (Liu et al., 2019; Wang et al., 2020; Shu et al., 2021), we adopt PSPNet Zhao et al. (2017) with ResNet101 He et al. (2016) backbone as the teacher for all experiments, and use different segmentation models (PSPNet Zhao et al. (2017) and DeepLabV3 Chen et al. (2017)) and backbones (ResNet18 He et al. (2016) and MobileNetV2 Sandler et al. (2018)) for the student to verify the effectiveness of our method.

4.2.2 Training details

We use the pretrained teacher model and keep its parameters fixed during distillation. For student training, we use Stochastic Gradient Descent (SGD) as the optimizer with a batch size of 16, a weight decay of 0.0005 and a momentum of 0.9. We use the “poly” learning rate policy where the learning rate equals to $base\_lr * (1 - \frac{iter}{max\_iter})^{power}$. We set the base learning rate to 0.01 and power to 0.9. We train 80k iterations for Cityscapes and Pascal VOC and 160k iterations for ADE20K. We apply random horizontal flipping, random scaling (from 0.5 to 2.0) and random cropping on the input images as data augmentation during training. The crop size for Cityscapes, Pascal VOC and ADE20K are 512×1024, 512×512 and 512×512, respectively. We use single scale testing for

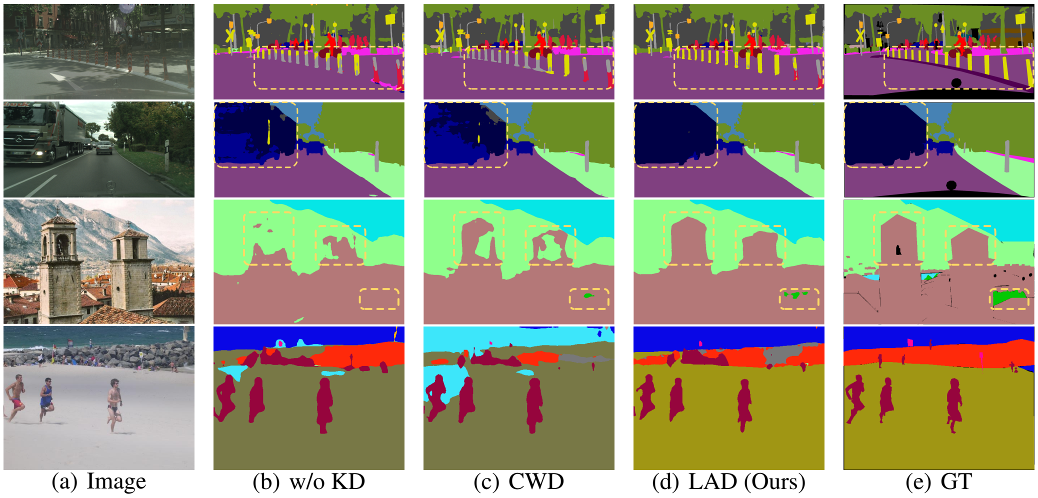


Figure 3: Qualitative comparison results for the student PSPNet-R18. The first two rows are the results on Cityscapes validation set, and the last two rows are the results on ADE20K validation set.

all datasets. Unless stated, the features from the last layer of the backbone are used for distillation in our method. λ_{ad} is set to 10 by default, and λ_{kd} is set to 10 following the previous methods (Liu et al., 2019; Wang et al., 2020; Shu et al., 2021).

4.3 Comparison with state-of-the-art methods

We compare our method with recent KD methods for semantic segmentation on Cityscapes, Pascal VOC and ADE20K. We re-implemented SKD (Liu et al., 2019), IFVD (Wang et al., 2020) and CWD (Shu et al., 2021) based on their released code. The hyper-parameters related to distillation loss are set according to their recommended values. For fair comparison, all methods use exactly the same training and testing strategies as described in Section 4.2.2. It is important to note that SKD, IFVD and CWD use the adversarial distillation loss on logits to improve performance, while our method does not.

Table 2 shows the results on various student models with different backbones (ResNet18 and MobileNetV2) and decoders (PPM (Zhao et al., 2017) and ASPP (Chen et al., 2017)). Our method significantly improves the performance of baseline students without KD. For example, the performance gains for PSPNet-R18 under our method are 4.21%, 4.39% and 4.60% on Cityscapes, Pascal VOC and ADE20K, respectively. Although our method utilizes the features from the last layer of the backbone for distillation by default, the performance gains are not much affected by the backbone architecture. Specifically, our method improves the performance of PSPNet-MV2 by 3.03%, 4.99% and 5.59% on Cityscapes, Pascal VOC and ADE20K, respectively. In addition, our method further narrows the performance gap between the teacher and DeepLabV3-R18 which acts as a strong baseline student.

More importantly, our method consistently outperforms other methods by a large margin under various experimental setups, especially on Pascal VOC and ADE20K. For example, our method outperforms CWD, the previous state-of-the-art KD method for semantic segmentation, by 2.85%, 3.66%, 2.48% and 3.17% when using PSPNet-R18, PSPNet-MV2, DeepLabV3-R18 and DeepLabV3-MV2 as student on more challenging ADE20K dataset. We further show the qualitative comparison results in Figure 3.

4.4 Ablation study

In this section, we give extensive experiments to investigate the effectiveness of our method and discuss the choice of some hyper-parameters. Ablation experiments are mainly conducted on Cityscapes and ADE20K, with PSPNet-R101 as the teacher and PSPNet-R18 as the student.

Table 3: Ablation study about different Angular Distillation methods. "T" and "S" denote the teacher and student, respectively. "–" means training without KD. "LAD", "CAD" and "PAD" denote Layer-wise, Channel-wise and Point-wise Angular Distillation, respectively.

Model	**KD Method**	**mIoU (%)**	
		Cityscapes	**ADE20K**
T: PSPNet-R101	–	79.76	44.39
S: PSPNet-R18	–	72.65	35.03
	Naive (Romero et al., 2015)	74.50	35.36
	LAD (Ours)	**76.86**	**39.63**
	CAD (Ours)	76.77	38.99
	PAD (Ours)	75.52	39.36

Table 4: Ablation study about the KD positions for our method on Cityscapes validation set. "T" and "S" denote the teacher and student, respectively. "–" means training without KD. "backbone" means the last layer of backbone, "decoder" means the last layer of decoder, and "logits" means the final prediction layer.

Model	**KD Method**	**mIoU (%)**
T: PSPNet-R101	–	79.76
S: PSPNet-R18	–	72.65
	LAD-backbone	**76.86**
	LAD-decoder	75.27
	LAD-logits	74.96
	CAD-backbone	**76.77**
	CAD-decoder	75.44
	CAD-logits	75.19

4.4.1 Dimensions of Angular Distillation

As mentioned before, our Angular Distillation can be further divided into Layer-wise Angular Distillation (LAD), Channel-wise Angular Distillation (CAD) and Point-wise Angular Distillation (PAD) according to the dimensions of the angular information. We compare these three methods for semantic segmentation on Cityscapes and ADE20K. As shown in Table 3, LAD achieves the best results, while CAD and PAD perform slightly worse than LAD. It is worth noting that LAD, CAD and PAD all perform better than Naive Feature Distillation, which demonstrates the effectiveness of our Angular Distillation.

4.4.2 Positions of KD

To evaluate our method at different KD positions, we conduct experiments using features from: 1) the last layer of backbone, 2) the last layer of decoder, and 3) the final prediction layer. As shown in Table 4, our method performs best at the last layer of the backbone. Note that the optimal loss weights for different KD positions may be different, but we use the same loss weights for different KD positions for simplicity in this experiment. The features produced by the decoder or prediction layer are highly compressed and task-specific as they are located at the top level of the model. Instead, the features from the backbone tend to be more informative and rich in generic representations, which may explain the better results of our method at the backbone.

4.4.3 Weights of KD loss

The Angular Distillation loss in our method is weighted by λ_{ad} in Eq. (10). We conduct extensive experiments on three datasets to investigate the sensitivity of our method to λ_{ad}. The results in Figure 4 demonstrate the excellent robustness of our method to hyper-parameters.

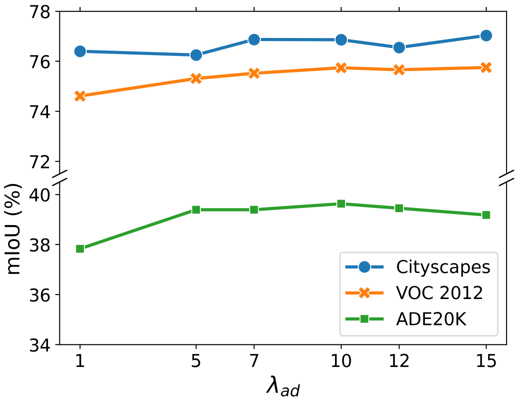


Figure 4: Ablation study about the sensitivity of our method (LAD) to loss weight λ_{ad}. The teacher model is PSPNet-R101, and the student model is PSPNet-R18. The features from the last layer of the backbone are used for distillation.

Table 5: Ablation study about the generalization of our method over different networks on ADE20K validation set. "T" and "S" denote the teacher and student, respectively. "–" means training without KD.

Model	**KD Method**	**mIoU (%)**
T: UPerNet-SwinB	–	47.99
S: UPerNet-SwinT	–	43.72
	Naive (Romero et al., 2015)	44.45
	CWD (Shu et al., 2021)	45.08
	LAD (Ours)	45.47
	CAD (Ours)	**46.12**

4.4.4 Generalization over different networks

Following the previous methods (Liu et al., 2019; Wang et al., 2020; Shu et al., 2021), the above experiments are mainly conducted on segmentation models with a plain encoder-decoder architecture like PSPNet without skip connections. In this section, we conduct experiments based on UPerNet (Xiao et al., 2018), which adopts FPN (Lin et al., 2017) to fuse multi-level features in an inherent and pyramidal hierarchy. In addition, we use Transformer backbone for teacher and student, which has a completely different architecture from CNN. Specifically, the teacher's backbone is Swin-B (Liu et al., 2021), while the student's backbone is Swin-T. As shown in Table 5, our method greatly improves the performance of the baseline student without KD and outperforms the Naive Feature Distillation and CWD. The results confirm the effectiveness of our method again, and further demonstrate the promising generalization of our method over different networks.

5 Conclusion

In this paper, we analyze the naive feature distillation method and find that the angular information is more effective for feature distillation than the magnitude information. Based on our finding, we propose a simple and effective feature distillation method for semantic segmentation. Extensive experiments demonstrate the superior performance of our method. We focus on how to effectively distill with raw features between manually assigned pairs of teacher-student intermediate layers, which may not be optimal. Therefore, how to effectively leverage multi-layer features for distillation is part of our future work.

REFERENCES

Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen. Cross-layer distillation with semantic calibration. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 7028–7036. AAAI Press, 2021.

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

Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part VII*, volume 11211 of *Lecture Notes in Computer Science*, pp. 833–851. Springer, 2018. doi: 10.1007/978-3-030-01234-2_49.

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 *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 3213–3223. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.350.

Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88 (2):303–338, June 2010. ISSN 1573-1405. doi: 10.1007/s11263-009-0275-4.

Bharath Hariharan, Pablo Arbelaez, Lubomir D. Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc Van Gool (eds.), *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pp. 991–998. IEEE Computer Society, 2011. doi: 10.1109/ICCV.2011.6126343.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 770–778. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.90.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531 [cs, stat]*, March 2015.

Deyi Ji, Haoran Wang, Mingyuan Tao, Jianqiang Huang, Xian-Sheng Hua, and Hongtao Lu. Structural and statistical texture knowledge distillation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16876–16885, 2022.

Mingi Ji, Byeongho Heo, and Sungrae Park. Show, attend and distill: Knowledge distillation via attention-based feature matching. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pp. 7945–7952. AAAI Press, 2021.

Xiao Jin, Baoyun Peng, Yichao Wu, Yu Liu, Jiaheng Liu, Ding Liang, Junjie Yan, and Xiaolin Hu. Knowledge distillation via route constrained optimization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1345–1354, 2019.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117–2125, 2017.

Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pp. 2604–2613. Computer Vision Foundation / IEEE, 2019. doi: 10.1109/CVPR.2019.00271.

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

Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XI*, volume 11215 of *Lecture Notes in Computer Science*, pp. 283–299. Springer, 2018. doi: 10.1007/978-3-030-01252-6_17.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 4510–4520. IEEE Computer Society, 2018. doi: 10.1109/CVPR.2018.00474.

Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen. Channel-wise knowledge distillation for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5311–5320, October 2021.

Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(10): 3349–3364, 2021. doi: 10.1109/TPAMI.2020.2983686.

Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu. Intra-class feature variation distillation for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision – ECCV 2020*, Lecture Notes in Computer Science, pp. 346–362, Cham, 2020. Springer International Publishing. ISBN 978-3-030-58571-6. doi: 10.1007/978-3-030-58571-6_21.

Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *Computer Vision – ECCV 2018*, Lecture Notes in Computer Science, pp. 432–448, Cham, 2018. Springer International Publishing. ISBN 978-3-030-01228-1. doi: 10.1007/978-3-030-01228-1_26.

Jiafeng Xie, Bing Shuai, Jianfang Hu, Jingyang Lin, and Wei-Shi Zheng. Improving fast segmentation with teacher-student learning. In *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, pp. 205. BMVA Press, 2018.

Junho Yim, Donggyu Joo, Ji-Hoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 7130–7138. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.754.

Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pp. 173–190. Springer, 2020. doi: 10.1007/978-3-030-58539-6_11.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Mueller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv:2004.08955 [cs]*, April 2020.

Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pp. 6230–6239. IEEE Computer Society, 2017. doi: 10.1109/CVPR.2017.660.

Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5122–5130, July 2017. doi: 10.1109/CVPR.2017.544.

Yichen Zhu and Yi Wang. Student customized knowledge distillation: Bridging the gap between student and teacher. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 5057–5066, 2021.

A Qualitative comparison results

We give more qualitative comparison results as shown in Figure 5.

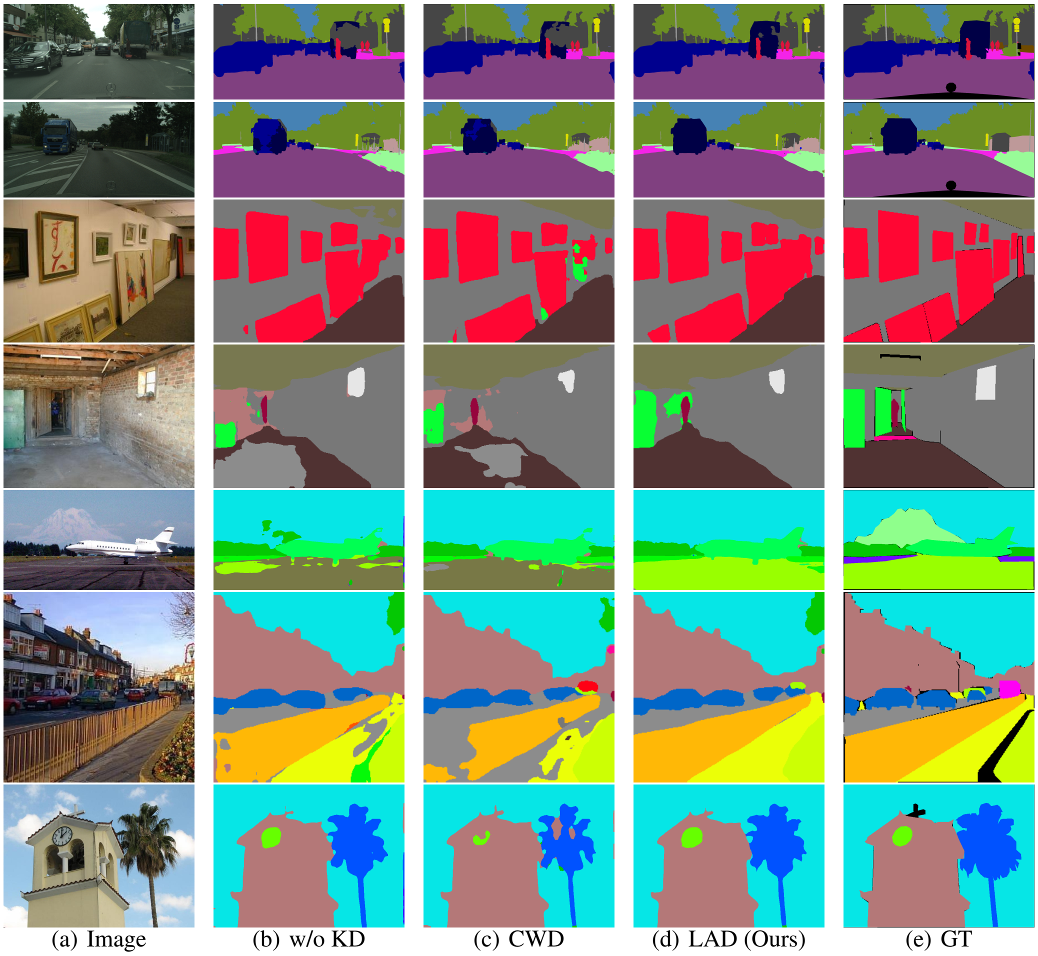


Figure 5: Qualitative comparison results for the student model PSPNet-R18. The first two rows are the results from Cityscapes validation set, and the rest are the results from ADE20K validation set.

B Detailed performance of each class

We calculate the IoU scores on each class for the student. As illustrated in Figure 6 and Figure 7, we can observe that our method consistently achieves better class performance than the baseline without KD and CWD, especially for those classes with few annotated pixels or low IoU scores.

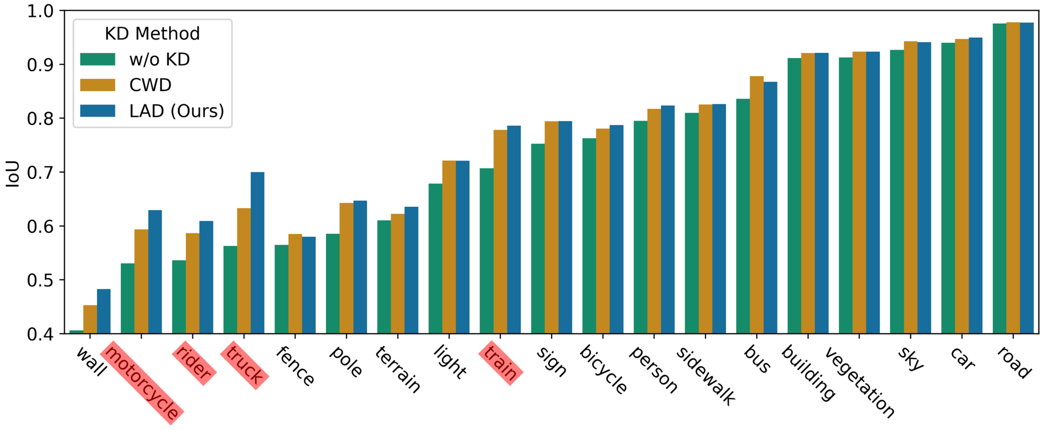
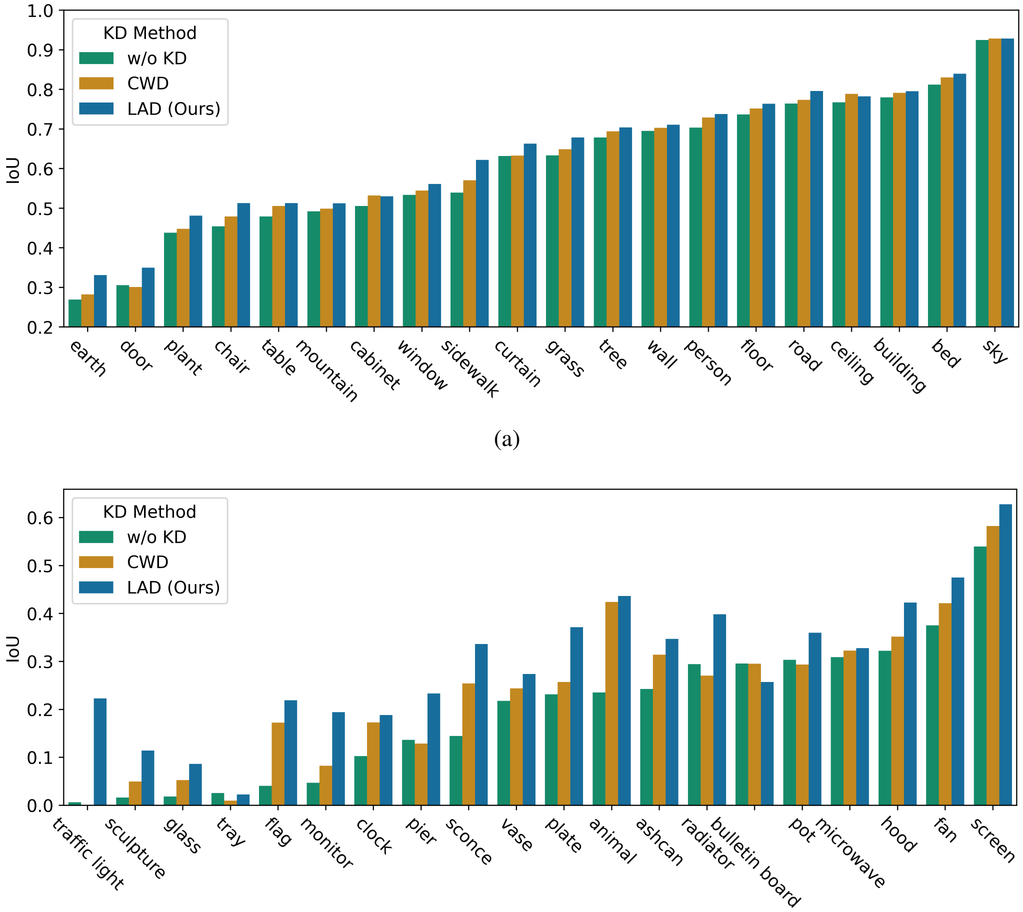


Figure 6: Detailed IoU scores of each class for the student model PSPNet-R18 on Cityscapes validation set. The class names highlighted in red are those that have few annotated pixels.



(a)

(b)

Figure 7: Detailed IoU scores of each class for the student model PSPNet-R18 on ADE20K validation set. (a) shows the results of the classes with a large number of annotated pixels. (b) shows the results of the classes with few annotated pixels.