Holistic Weighted Distillation for Semantic Segmentation

Wujie Sun
College of Computer Science
Zhejiang University
Hangzhou, China
sunwujie@zju.edu.cn

Defang Chen
College of Computer Science
Zhejiang University
Hangzhou, China
defchern@zju.edu.cn

Can Wang
College of Computer Science
Zhejiang University
Hangzhou, China
wcan@zju.edu.cn

Deshi Ye*
College of Computer Science
Zhejiang University
Hangzhou, China
yedeshi@zju.edu.cn

Yan Feng
College of Computer Science
Zhejiang University
Hangzhou, China
fengyan@zju.edu.cn

Chun Chen
College of Computer Science
Zhejiang University
Hangzhou, China
chenc@zju.edu.cn

***Abstract*—Channel-wise distillation for semantic segmentation has proven to be a more effective method than spatial-based distillation. By removing the redundant information from the teacher model, the student can focus on specific channel-related pixels, which can be viewed as a weighting of the pixels. However, the standard channel-wise distillation ignores the fact that such importance difference also exists among channels. In this paper, we propose a novel method called Holistic Weighted Distillation (HWD) to address this issue. We calculate the channel divergences between the teacher and the student, and convert them into distillation weights, making the student focus more on learning channels that are not well mastered, thus improving the final model performance. Besides, our method does not introduce additional network structure or back-propagation process, which improves the training efficiency. Experiments on ADE20K, Cityscapes, and COCO-Stuff demonstrate the superiority of our method. The code is available at https://github.com/zju-SWJ/HWD.**

***Index Terms*—semantic segmentation, knowledge distillation, weighted training**

I. Introduction

Semantic segmentation is an important and challenging task in computer vision [1]. Due to the huge amount of computation, how to simultaneously achieve high accuracy and efficiency has become a pressing problem for the deployment on mobile and edge devices. Some works attempt to design lightweight networks [2], [3], while others employ knowledge distillation [4] to achieve model compression [5]–[8].

Traditional knowledge distillation engages a teacher model and a student model in learning. The teacher model is well-trained and usually has more complex architecture and superior performance than the student. In training, the pre-trained teacher transfers its intermediate features [9], [10] or final outputs [4] to the student, so that the student can learn from both the teacher and the ground-truth labels. Existing studies indicate that the transferred information contains useful knowledge that cannot be directly obtained from the ground-truth labels, such as inter-class similarity [4], which can be used to further improve student performance.

Existing knowledge distillation methods for semantic segmentation can be generally divided into two categories: spatial-based [5], [6], [8] and channel-based [7]. The former focuses on obtaining and transferring structural spatial knowledge among the pixels and regions. The latter, e.g. CWD [7], views the transferred knowledge from the channel aspect. CWD found that the activation values of each channel tend to encode class-related saliency. By calculating the KL divergence between the teacher's and student's channel-related activation maps, CWD removes the redundant information from the teacher model, so that the student can pay more attention to learning from specific channel-related pixels. Experiments have shown that CWD greatly improves model performance compared to spatial-based methods.

In CWD [7], pixels in channels are weighted, but each channel still contributes equally in distillation. However, when comparing the class-related performances of the student trained without distillation and CWD, we found that CWD has gained little or even negative improvements in some classes. In some classes, using ground-truth alone can achieve satisfactory result and distillation sees no help. This indicates that some channels are useless in distillation and might even interfere with other channels, hindering further performance improvement. To address this issue, we propose a novel method called **H**olistic **W**eighted **D**istillation (HWD). Based on channel-wise knowledge, HWD further let the student focus more on those poorly-trained channels to achieve more performance improvement. To be more specific, mean square error is employed in HWD to measure the channel divergences between the teacher and the student. Since such divergences are

This work is supported by National Natural Science Foundation of China (Grant No: U1866602), the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (Grant No: SN-ZJU-SIAS-001) and the advanced computing resources provided by the Supercomputing Center of Hangzhou City University.

*Corresponding author

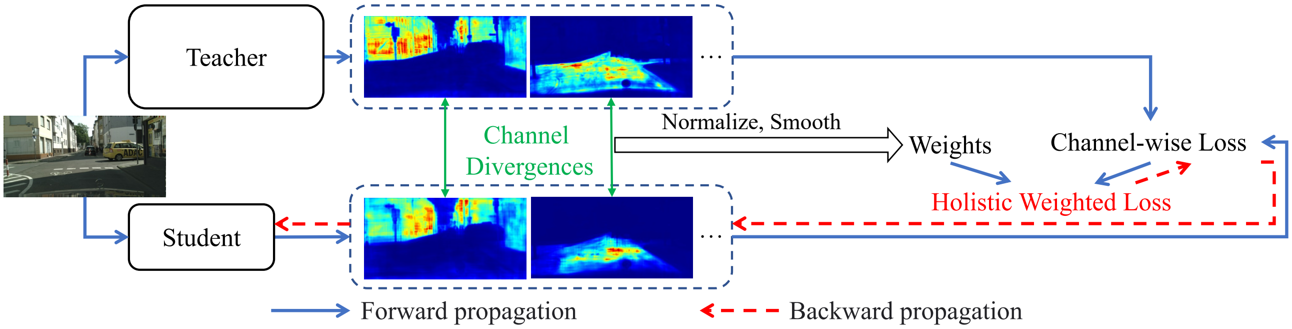

Fig. 1. Overall framework of HWD. The output activation map contains the information associated with current class (as shown in the figure, the first activation map is related to "building", and the second activation map is related to "road"). We calculate channel divergences and turn them into weights. The weights are combined with channel-wise loss for updating student model.

easily dominated by the class-related pixels, we correct them by considering the class-related pixel numbers. After that, we convert the divergences into distillation weights through carefully designed normalization and smoothing processes. An overview of HWD is shown in Fig. 1. Our contributions are summarized as follows:

- We propose a novel channel-based knowledge distillation method for semantic segmentation. The idea is simple and easy to implement.
- Our method does not introduce additional network or back propagation process, which highly improves the training efficiency and reduce training uncertainty.
- Experiments on ADE20K, Cityscapes, and COCO-Stuff demonstrate that our method consistently outperforms SOTA methods with various distillation pairs.

II. Related Work

Most semantic segmentation methods utilize the Fully Convolutional Networks to capture long-range information. By adopting multiple atrous rates, DeepLabV3 [11] employs atrous convolution in cascade or in parallel to capture segmenting objects at multiple scales. PSPNet [12] introduces the pyramid pooling module to aggregate different sub-region representations, so that local and global context information can be captured. Model architectures such as ESPNet [2] and ICNet [3] are proposed to achieve quick and high-quality segmentation. However, considering the complexity of application scenarios, such restricted model architecture may not be applicable in many scenarios.

Knowledge distillation [4] is first proposed to deal with the resource problems in image classification. Knowledge such as logits [4], [13] and intermediate features [9] can be transferred from the teacher to the student to improve model performance. Since the uniqueness of the task is not considered, general knowledge distillation methods may not achieve satisfactory results in semantic segmentation. Therefore, in recent years, distillation methods specially designed for semantic segmentation has gradually increased. SKD [5] transfers the pairwise relations and introduces adversarial learning to further help the training. IFVD [6] takes the intra-class feature variation as distilled knowledge. CIRKD [8] first considers global pixel relations across various images and uses these knowledge to improve model performance. Different from the above spatial-based methods, CWD [7] first introduces channel-wise distillation and achieves superior performance.

III. Methodology

To help readers have a better understanding of the methodology, we will first briefly introduce the training of semantic segmentation (Section III-A) and channel-wise distillation (Section III-B), and then describe our method in Section III-C.

A. Semantic segmentation

In the semantic segmentation task [1], we need to classify every image pixel to a specify class. A common practice is using the backbone network to extract features $F \in \mathbb{R}^{H' \times W' \times d}$ from the input image with height H and width W first. These features are then input to the fully convolutional networks to get the logit map $Z \in \mathbb{R}^{H \times W \times C}$, where C is the total number of classes. When training without distillation, the model aims to minimize the cross-entropy loss between the logit map and the ground-truth label $Y \in \mathbb{R}^{H \times W}$:

$$L_{\text{CE}} = -\frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} y_{h,w} \log \sigma(z_{h,w}), \tag{1}$$

where $\sigma(\cdot)$ is the softmax function.

B. Channel-wise distillation

Inspired by the success of knowledge distillation in image classification, some methods [5], [6], [8] normalize the values on each spatial location, and then align the features in the spatial domain. However, CWD [7] points out that strictly aligning these point-wise features may enforce overly strict constraints and lead to sub-optimal solutions. Therefore, CWD aims to transfer the channel-wise knowledge to help the training. The logit map is reshaped to $C \times HW$ and normalized with softmax function to get the activation map p:

$$p_{c,i} = \frac{\exp\left(\frac{z_{c,i}}{\tau}\right)}{\sum_{j=1}^{WH} \exp\left(\frac{z_{c,j}}{\tau}\right)}, \tag{2}$$

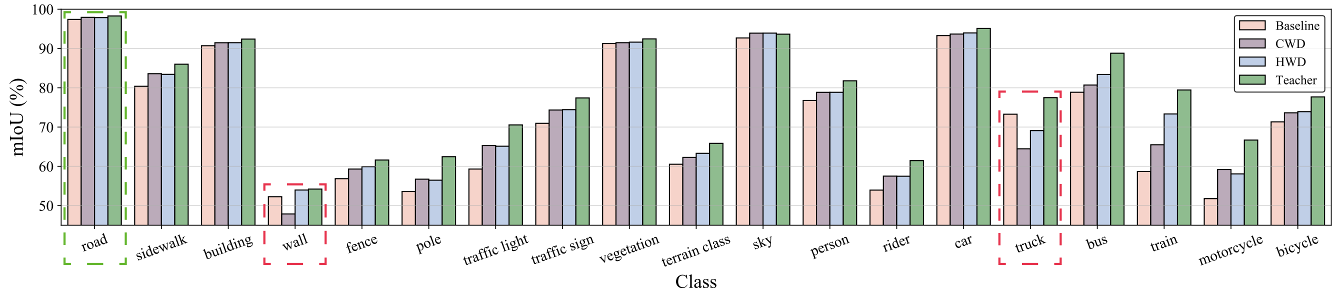

Fig. 2. Illustration of the class performance on Cityscapes validation set. The teacher model is DeepLabV3-ResNet101, and the student model is DeepLabV3-ResNet18. Baseline means that no distillation method is used to train the student model.

TABLE I
VALUES OF TD/NTD OF EACH CLASS ON CITYSCAPES VALIDATION SET WHEN THE STUDENT IS TRAINED USING CWD. DISTILLATION IS PERFORMED FROM DEEPLAB-RES101 TO DEEPLAB-RES18.

Class	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation
TD/NTD	308.7	63.4	20.1	46.2	54.7	37.5	152.4	137.6	46.3
terrain class	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
37.2	85.3	145.6	281.9	330.1	143.3	374.0	235.4	266.4	197.0

where τ is the temperature to control the distribution of the activation map. In the training, CWD tries to minimizes the KL divergence between the activation maps of the teacher T and the student S:

$$L_{\text{CWD}} = \frac{\tau^2}{C} \sum_{c=1}^{C} \sum_{i=1}^{HW} p_{c,i}^T \log \frac{p_{c,i}^T}{p_{c,i}^S}. \tag{3}$$

In this way, the student model pays more attention to learning the foreground related to the current channel while weakening the background. However, this approach only distinguishes the learning importance of the channel-related pixels, whereas we believe that channels should also have different learning importance in the distillation process.

As we can see from Fig. 2, when the class is "wall" or "truck", CWD performs even worse than the baseline (training without distillation). When it comes to classes such as "road", the gap between baseline and teacher is quite small, indicating that good performance can be achieved just by using the ground-truth labels. This suggests that we should selectively focus more on some of the channels (classes) rather than all of them during the distillation. Therefore, we propose Holistic Weighted Distillation (HWD) to solve the above problem. As shown in Fig. 2, the performance of HWD is further improved or on a par with CWD in almost all classes. The improvement is significant when CWD is weaker than the baseline, e.g., "wall" and "truck" classes.

C. Holistic weighted distillation

The loss function of our proposed HWD is represented as

$$L_{\text{HWD}} = \frac{\tau^2}{C} \sum_{c=1}^{C} w_c \sum_{i=1}^{HW} p_{c,i}^T \log \frac{p_{c,i}^T}{p_{c,i}^S}, \tag{4}$$

and $\sum_{c=1}^{C} w_c = C$ is satisfied. Therefore, our goal is to convert the training information of the teacher and student models into channel weights.

1) Channel divergence: For a input image, we define the channel divergence as the mean-square error (MSE) between each channel of the teacher and student activation maps p^T and p^S:

$$\text{CD}_c(p^T, p^S) = \frac{\sum_{i=1}^{HW} (p_{c,i}^T - p_{c,i}^S)^2}{HW}. \tag{5}$$

Higher channel divergence indicates the student is not learning the teacher knowledge well for the current channel, which motivates us to raise the weight of this channel in the training.

However, it is not enough to consider channel divergence only. For channel c, we use the Target Divergence (TD) to represent the channel divergence of the pixels with label c, and the Non-Target Divergence (NTD) is the channel divergence of the pixels with other labels. As shown in Table I, the TD between teachers and students in each channel is significantly higher than the NTD. This suggests that larger channel divergence not necessarily leads to less well-learned channel, but may due to the existence of more labeled pixels. Therefore, we use $\text{CD}'_c = \frac{\text{CD}_c}{N_c+1}$ as a measure of *corrected channel divergence*, where N_c is the total pixel number of class c in the image. We add 1 to the denominator to avoid training errors since the image may not contain pixels belonging to some classes.

2) Weight processing: The weight processing can be divided into 2 steps: normalization and smoothing, which we will introduce in sequence.

Normalization. Since different scenarios (datasets, teacher-student models) yield widely varying channel divergences, we normalize CD'_c to the normal distribution $\mathcal{N}(\mu, \sigma^2)$ to make our method more generalizable:

$$w_c^n = \mu + \frac{\text{CD}'_c - \text{mean}(\text{CD}')}{\text{std}(\text{CD}')} \times \sigma, \tag{6}$$

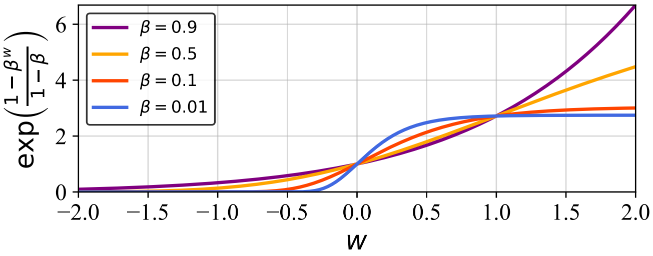

Fig. 3. Illustration of the changes of $\exp\left(\frac{1-\beta^w}{1-\beta}\right)$ when w takes different values ranging from 0 to 1. Smaller w has stronger smoothing effect. $\exp(\cdot)$ is used since softmax is introduced to satisfy the condition $\sum_{c=1}^{C} w_c = C$.

where CD' contains all C channel divergences of the image.

Smoothing. Inspired by the effective number of samples used in long-tailed learning [14], we further smooth weights by

$$w_c = C \times \sigma\left(\frac{1-\beta^{w_c^n}}{1-\beta}\right), \tag{7}$$

where $\beta \in (0, 1)$ is a smoothing hyper-parameter. Equation within the softmax function $\sigma(\cdot)$ is the actual smoothing function, and the external equation is used to satisfy $\sum_{c=1}^{C} w_c = C$. Taking $\mu = 0, \sigma = 1$ in Equ. (6) as an example, about 95% weights w_c^n will be in the -2 to 2 range. As shown in Fig. 3, when $\beta \to 1$, weights are the same as before the process; when β changes from 1 to 0, channels with weights between 0 and 1 will be given more weight in the training; when $\beta \to 0$, channels with weights greater than 0 will receive equal training priority, while channels with weights less than 0 will be ignored in the training. It is worth noting that the channel weights corresponding to different images are not consistent, and what we have done is to strengthen the weights of the poorly-trained channels in each image, rather than assuming that specific channels will have poorer training results. As the training progresses, the order of the channel weights changes.

3) Overall framework: By combing the general semantic segmentation loss L_{CE} and holistic weighted distillation loss L_{HWD} together, we can get the overall training loss:

$$L_{\text{overall}} = L_{\text{CE}} + \alpha L_{\text{HWD}}, \tag{8}$$

where α is used to adjust the importance of HWD in training. We summary the training pseudo-code in Algorithm 1.

Does network-based weight calculation work? Another approach is to use an neural network for weight calculation, as done in SENet [15]. However, this approach did not work, and we clarify the reasons in the supplementary material.

IV. Experiments

A. Settings

We use three standard semantic segmentation datasets in our experiments: ADE20K [16], Cityscapes [17], and COCO-Stuff [18]. We report the mean Intersection-over-Union (mIoU) to evaluate the performance on validation set. We also report the parameter numbers and the floating-point operations per second (FLOPs) for each model. Consistent with previous works [5]–[8], [19], [20], we use DeepLabV3 [11] and PSPNet [12] as segmentation frameworks. ResNet101, ResNet18 [21] and MobileNetV2 [22] are used for the backbones. Backbones are pre-trained on ImageNet.

Algorithm 1 Holistic Weighted Distillation

Input: Pre-trained teacher model θ^T, student model θ^S, training dataset $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$
Parameter: Total iterations N_{iter}, class number C, loss hyper-parameter α, activation temperature τ, distribution mean μ and variance σ^2, smoothing hyper-parameter β
Output: Student model θ^S

```
for $iter = 1, \ldots, N_{iter}$ do
    Sample a mini-batch from the dataset $\mathcal{D}$.
    Input $x$ to $\theta^T$ and $\theta^S$, get logit maps $z^T$ and $z^S$.
    Calculate the cross-entropy $L_{\text{CE}}$ with $y$.
    Get activation maps $p^T$ and $p^S$.
    Calculate the weight $w_c$.
    Calculate loss $L_{\text{HWD}}$, get the overall loss $L_{\text{overall}}$.
    Back propagation, update $\theta^S$.
end for
return $\theta^S$
```

We compare our method with 3 state-of-the-art semantic segmentation-based distillation methods: SKD [5], CIRKD [8], and CWD [7]. We also include general knowledge distillation method MGD [19]. Based on the codes provided by the authors, we re-implemented these methods in a unified framework. For our method, we set temperature τ to 4, α to 3, and variance σ^2 to 1. More details and results for IFVD [6] and DIST [20] can be found at the supplementary material.

B. Result

In Table II, we report the performance of multiple methods on ADE20K dataset with various teacher-student pairs. As we can see, our method HWD achieves the best validation mIoU in all cases. Compared to the second best method (CWD in our experiments), HWD brings 1.88 (6.4%) performance boosts when distilled from DeepLab-Res101 to DeepLab-MBV2, and the minimum improvement is 0.95 (2.9%) obtained when the student is DeepLab-Res18. From the experimental results, it is clear that spatial-based distillation methods (SKD and CIRKD) are more influenced by the distillation pairs. For example, when the student is DeepLab-Res18, SKD achieves significant improvement compared to the baseline method; however, when the student changes, SKD is even inferior to the baseline. Meanwhile, the general knowledge distillation method MGD, which always improves student performance, is inferior to the channel-based distillation methods (CWD and HWD). This shows the superiority of channel-based distillation methods in semantic segmentation.

As for Cityscapes and COCO-Stuff, results in Table III show that HWD still achieves the best performance even when CWD is inferior to other methods, demonstrating the superiority of our method.

We further report the qualitative segmentation results and t-SNE visualization [23] in Fig. 4 and Fig. 5, respectively.

TABLE II

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART DISTILLATION METHODS OVER VARIOUS STUDENT MODELS ON ADE20K DATASET. FLOPS IS MEASURED BASED ON THE INPUT IMAGE WITH SIZE 512×512. **BOLD VALUE** DENOTES THE BEST RESULT.

Method	Params	FLOPs	mIoU (%)
T: DeepLab-Res101	61.1M	384.7G	42.46
S: DeepLab-Res18	13.6M	86.0G	29.86
+SKD (TPAMI 2020)			31.05
+CWD (ICCV 2021)			32.90
+CIRKD (CVPR 2022)			30.47
+MGD (ECCV 2022)			30.32
+HWD (ours)			**33.85**
S: DeepLab-MBV2	3.2M	22.6G	27.16
+SKD			26.96
+CWD			29.19
+CIRKD			27.39
+MGD			27.39
+HWD (ours)			**31.07**
S: PSPNet-Res18	12.9M	67.6G	31.64
+SKD			31.30
+CWD			33.43
+CIRKD			31.36
+MGD			32.10
+HWD (ours)			**34.47**

TABLE III

PERFORMANCE COMPARISON WITH STATE-OF-THE-ART DISTILLATION METHODS ON CITYSCAPES AND COCO-STUFF. EVALUATION SIZES FOR CITYSCAPES AND COCO-STUFF ARE 1024×2048 AND 512×512, RESPECTIVELY. **BOLD VALUE** DENOTES THE BEST RESULT.

Dataset	Method	FLOPs	mIoU (%)
Cityscapes	T: DeepLab-R101	2371.7G	78.07
	S: DeepLab-R18	572.0G	71.78
	+SKD		72.79
	+CWD		73.57
	+CIRKD		72.23
	+MGD		71.79
	+HWD (ours)		**74.72**
COCO-Stuff	T: DeepLab-R101	384.7G	35.31
	S: DeepLab-R18	86.0G	26.05
	+SKD		25.94
	+CWD		26.17
	+CIRKD		25.85
	+MGD		26.44
	+HWD (ours)		**26.76**

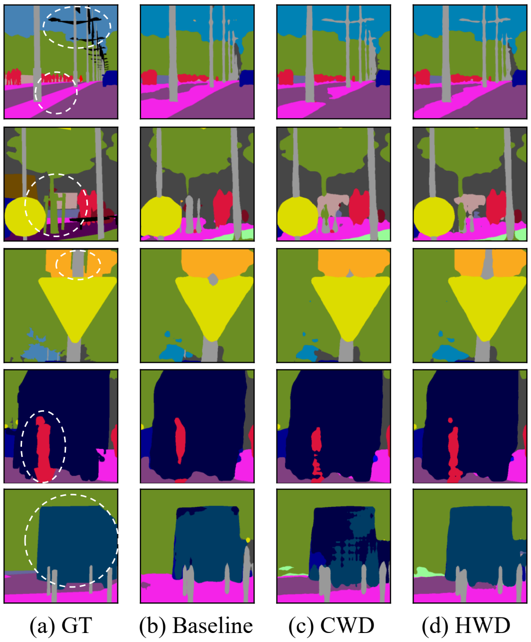

Fig. 4. Qualitative segmentation results on Cityscapes validation set. The student model is DeepLabV3-ResNet18. GT denotes ground-truth. Baseline means that no distillation method is used to train the student model. HWD successfully captures more tiny objects (e.g. pole, tree trunk), and the segmentation objects are more complete and coherent.

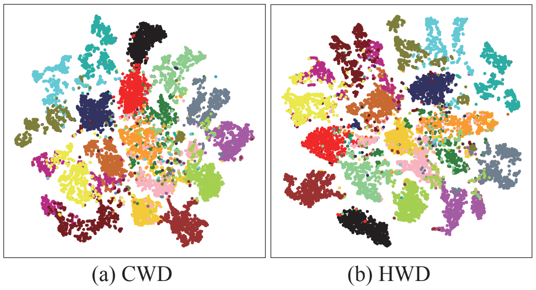

Fig. 5. T-SNE visualization of learned features embeddings on Cityscapes validation set. The student model is DeepLabV3-ResNet18. Using HWD can get a more structured and distinguishable feature space.

C. Ablation study

In this section, we conduct several ablation studies using the student model DeepLabV3-ResNet18 and dataset ADE20K. We set $\mu = 0$ and $\beta = 0.1$ by default.

Ablation study on smoothing function. In this paper, we use Equ. (7) to smooth the weights. It is actually a combination of two functions: smoothing and softmax. The former is used to smooth the weights and the latter is used to satisfy the condition $\sum_{c=1}^{C} w_c = C$. Note that softmax with high temperature τ' can also be used for smoothing, so can we skip the smoothing function and use high temperature τ' directly at the softmax function? Results in Fig. 6 illustrate two points: smoothing is necessary (weak smoothing brings poor performance when $\beta \to 1$ or $\tau' < 3$) and softmax with higher τ' cannot replace it. Next, we illustrate the difference between softmax and smoothing functions.

When softmax with high temperature is used, the increase in smaller values after softmax is always greater than the larger values. This leads to the fact that while some of the poorly-trained channels with large distillation weights are weakened when using higher τ', the well-trained channels with small distillation weights can be greatly enhanced in distillation, hindering performance improvement.

However, the smoothing function can have a very different effect: it has the ability to make the model barely learn the well-trained channels while focusing more on those poorly-trained channels in distillation. Besides, the smoothing hyper-parameter β can be jointly tuned with distribution mean μ to achieve better performance (discussed later), while changing μ has no effect on the softmax function. In addition, it can be observed that using different values of β over a considerable range (from 0.01 to 0.7) can give a significant performance improvement compared to CWD, which indicates that HWD

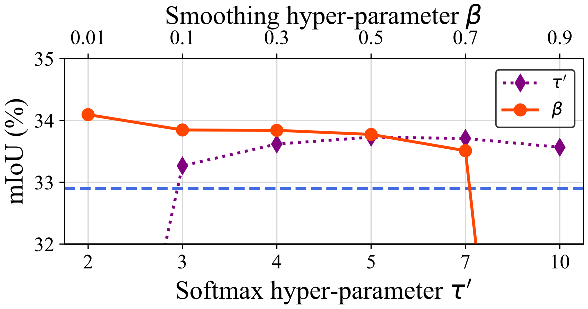

Fig. 6. Impact of the smoothing function on performance. The performance of CWD is shown by blue dashed line.

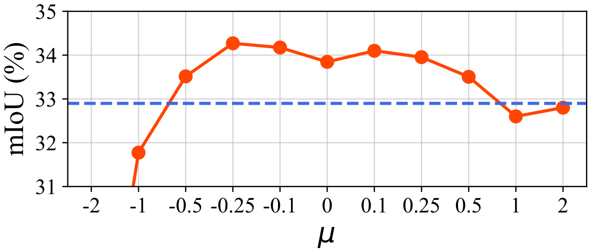

Fig. 7. Impact of the distribution mean μ on performance. The performance of CWD is shown by blue dashed line.

is relatively robust in the choice of β.

Ablation study on distribution mean μ. The distribution mean μ plays a crucial role in the final model performance. Since we used a small β (strong smoothing effect), close to half of the channels barely learn in every iteration when $\mu = 0$. It has no negative impact when the model learns well in most channels by relying on labels alone, but if not, it can cause substantial performance degradation. Therefore, the μ may vary when the dataset or distillation pair changes.

As μ gradually increases from 0, more and more channels start to play a role in distillation and eventually converge to CWD (when weight is larger than 1, the weights of each channel are close because of the strong smoothing effect); as μ gradually decreases from 0, fewer and fewer channels play a role, which eventually brings performance degradation. Results in Fig. 7 confirm our statement.

V. Conclusion

In the distillation process of the semantic segmentation task, there are significant differences in distillation effectiveness between various classes (channels). Therefore, we propose **H**olistic **W**eighted **D**istillation (HWD) to make the student model pay more attention to channels that are not well distilled, thus improving the model performance. We conduct experiments on ADE20K, Cityscapes, and COCO-Stuff, with various distillation pairs. Experimental results show the superiority of HWD. In addition, our method does not introduce additional network structure or back propagation process, so that the training efficiency can be improved, and the uncertainty caused by training the neural network can be reduced. Our method has broad applicability and is particularly effective for challenging datasets with a large number of classes.

References

[1] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez, "A review on deep learning techniques applied to semantic segmentation," *arXiv preprint arXiv:1704.06857*, 2017. 1, 2

[2] Sachin Mehta, Mohammad Rastegari, Anat Caspi, Linda Shapiro, and Hannaneh Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," in *ECCV*, 2018, pp. 552–568. 1, 2

[3] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia, "Icnet for real-time semantic segmentation on high-resolution images," in *ECCV*, 2018, pp. 405–420. 1, 2

[4] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al., "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015. 1, 2

[5] Yifan Liu, Changyong Shu, Jingdong Wang, and Chunhua Shen, "Structured knowledge distillation for dense prediction," *TPAMI*, 2020. 1, 2, 4

[6] Yukang Wang, Wei Zhou, Tao Jiang, Xiang Bai, and Yongchao Xu, "Intra-class feature variation distillation for semantic segmentation," in *ECCV*. Springer, 2020, pp. 346–362. 1, 2, 4

[7] Changyong Shu, Yifan Liu, Jianfei Gao, Zheng Yan, and Chunhua Shen, "Channel-wise knowledge distillation for dense prediction," in *ICCV*, 2021, pp. 5311–5320. 1, 2, 4

[8] Chuanguang Yang, Helong Zhou, Zhulin An, Xue Jiang, Yongjun Xu, and Qian Zhang, "Cross-image relational knowledge distillation for semantic segmentation," in *CVPR*, 2022, pp. 12319–12328. 1, 2, 4

[9] Defang Chen, Jian-Ping Mei, Yuan Zhang, Can Wang, Zhe Wang, Yan Feng, and Chun Chen, "Cross-layer distillation with semantic calibration," in *AAAI*, 2021, vol. 35, pp. 7028–7036. 1, 2

[10] Defang Chen, Jian-Ping Mei, Hailin Zhang, Can Wang, Yan Feng, and Chun Chen, "Knowledge distillation with the reused teacher classifier," in *CVPR*, 2022, pp. 11933–11942. 1

[11] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017. 2, 4

[12] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia, "Pyramid scene parsing network," in *CVPR*, 2017, pp. 2881–2890. 2, 4

[13] Defang Chen, Jian-Ping Mei, Can Wang, Yan Feng, and Chun Chen, "Online knowledge distillation with diverse peers," in *AAAI*, 2020, pp. 3430–3437. 2

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019, pp. 9268–9277. 4

[15] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141. 4

[16] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba, "Scene parsing through ade20k dataset," in *CVPR*, 2017, pp. 633–641. 4

[17] 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*, 2016, pp. 3213–3223. 4

[18] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari, "Coco-stuff: Thing and stuff classes in context," in *CVPR*, 2018, pp. 1209–1218. 4

[19] Zhendong Yang, Zhe Li, Mingqi Shao, Dachuan Shi, Zehuan Yuan, and Chun Yuan, "Masked generative distillation," in *ECCV*, 2022. 4

[20] Tao Huang, Shan You, Fei Wang, Chen Qian, and Chang Xu, "Knowledge distillation from a stronger teacher," *arXiv preprint arXiv:2205.10536*, 2022. 4

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778. 4

[22] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *CVPR*, 2018, pp. 4510–4520. 4

[23] Laurens Van der Maaten and Geoffrey Hinton, "Visualizing data using t-sne.," *JMLR*, vol. 9, no. 11, 2008. 4