Supplementary Matrial: Self-Knowledge Distillation with Progressive Refinement of Targets

Kyungyul Kim[1] ByeongMoon Ji[1] Doyoung Yoon[1] Sangheum Hwang[2]*

[1]LG CNS AI Research, Seoul, South Korea
[2]Seoul National University of Science and Technology, Seoul, South Korea
{kyungyul.kim, jibm, dy0916}@lgcns.com, shwang@seoultech.ac.kr

A. Image Classification

A.1. Evaluation Metrics

ECE. Expected calibration error (ECE) [5] is a widely used metric for evaluating confidence calibration performance. To estimate the expected gap between accuracy and confidence, it partitions samples into total M bins, B_m for $m = 1, ..., M$, by confidence. Then, each bin B_m contains samples with confidence within $[\frac{m-1}{M}, \frac{m}{M}]$. With this binning, ECE is defined as follows,

$$ECE = \frac{1}{n} \sum_{m=1}^{M} |B_m| \times |\text{Acc}(B_m) - \text{Conf}(B_m)|$$

where n is number of samples, $\text{Acc}(B_m)$ represents accuracy of samples in B_m, and $\text{Conf}(B_m)$ represents average confidence of samples in B_m. The lower value of ECE indicates that a model is well-calibrated.

The reliability diagram [1, 4] and calibration plot are visualization tools to show how well confidence of a model is calibrated by plotting accuracy against confidence values.

AURC. Area under risk-coverage curve (AURC) [3] measures how well predictions are ordered by confidence values. Given a classifier, we can define a selective classifier with a threshold which covers only samples with higher confidence than the threshold. Then, coverage can be defined as the proportion of covered samples (i.e., not rejected samples by the selective classifier) to the entire dataset. Risk is defined as an error rate computed by using the covered samples. Therefore, as coverage increases from 0 to 1, risk approaches to the top-1 error on the entire data. AURC is defined as the area under the risk-coverage curve. If a model has a low AURC value, it means that correct and incorrect predictions from the model are well-separable by confidence values.

A.2. Methods

Label smoothing. Szegedy et al. [6] proposes a method named label smoothing which improves the performance of deep learning models by adjusting one-hot targets to be soft targets. Soft targets $\mathbf{y}_{LS}$ are computed as a weighted sum of the hard targets $\mathbf{y}$ and the uniform distribution over classes, i.e.,

$$\mathbf{y}_{LS} = (1 - \epsilon)\mathbf{y} + \frac{\epsilon}{K}$$

where ϵ is a smoothing parameter and K is the number of classes.

Cutout. Cutout [2] is a simple regularization method designed for image classification. Motivated by dropout and image augmentation, Cutout generates a partially occluded version of input samples, which can be interpreted as an augmented data by applying the structured dropout to an input space. In detail, a square-shaped region with the predefined size is randomly selected on an input image, and that region is zeroed-out during training.

*Corresponding author

CutMix. Yun et al. [9] suggests a method inspired by Cutout [2] and Mixup [11]. This method generates a new training sample $(\tilde{x}, \tilde{y})$ from two samples (x_a, y_a) and (x_b, y_b). From x_a, a rectangular region with bounding box coordinates (r_x, r_y, r_w, r_h) will be sampled as a patch. Then, the region of the same coordinates in x_b will be replaced by the patch to generate $\tilde{x}$. For the generated sample $\tilde{x}$, its target $\tilde{y}$ is defined as

$$\tilde{y} = \lambda y_a + (1 - \lambda) y_b$$

where λ denotes the combination ratio sampled from the uniform distribution $(0, 1)$.

ShakeDrop. ShakeDrop [7] is a regularization technique designed for ResNet and its variants. This method gives regularization effect by replacing residual blocks to ShakeDrop blocks. Let an input be x and an output of residual block be $F(x)$, then the output of l-th ShakeDrop block $G(x)$ is defined as,

$$G(x) = \begin{cases} x + (b_l + \alpha - b_l\alpha)F(x), & \text{for the train-forward phase} \\ x + (b_l + \beta - b_l\beta)F(x), & \text{for the train-backward phase} \\ x + E[b_l + \beta - b_l\beta]F(x), & \text{for test phase} \end{cases}$$

where α, β are independent uniform random variables and b_l is a Bernoulli random variable with probability $P(b_l = 1) = p_l$, which is a parameter with a linear decay according to the block index l:

$$p_l = 1 - \frac{l}{L}(1 - P_L)$$

where L is the total number of building blocks and P_L is an initial parameter. In our experiments, we use $P_L = 0.5$ as suggested in [7].

A.3. Datasets

CIFAR-100 is a dataset for multi-class image classification. It consists of 50K training images and 10K test images of 32×32 resolutions with 100 classes, and has the same number of images per class. The ImageNet is a large-scale dataset. It consists of 1.2M training images and 50K validation images of various resolutions with 1K classes. It contains some images that have multiple objects. In training, we use an input image that is resized to 256×256, and it is randomly cropped to have a size of 224×224. For inference, we resize an image as 256×256 and perform the center crop to have a 224×224 sized input.

A.4. Experimental Results on CIFAR-100

Hyperparameters. For LS, we use the smoothing parameter ϵ of 0.1. For CS-KD[1], we set the temperature τ to 4, and the weight λ_{cls} to 1 [10]. For TF-KD[2], we use TF-KD$_{self}$ method presented in [8]. The hyperparameters, the temperature τ and weight α, for ResNet-18, DenseNet-121 and ResNeXt-29 are set to the values reported in [8]. For ResNet-101 and PyramidNet, we use the temperature $\tau = 20$ and weight $\alpha = 0.95$, which are most widely used settings in the paper.

Ablation study on the hyperparameter α_T of PS-KD. To investigate the effect of our hyperparameter α_T, we provide the validation performances in terms of top-1 error and ECE on CIFAR-100 with ResNet-18. The results are given in Fig. S1. Considering both top-1 error and ECE metrics, we determine the optimal α_T as 0.8. For $\alpha_T > 0.8$, we observe that ECE suffers from PS-KD while top-1 accuracy still improves, implying that PS-KD with a large value of $\alpha_T > 0.8$ tends to produce underconfident predictions as can be seen in Fig. S2. Fig. S2 shows the reliability diagrams on the validation dataset with PS-KD. PS-KD with $\alpha_T = 0.8$ shows best calibration performance.

[1] CS-KD implementation:https://github.com/alinlab/cs-kd
[2] TF-KD implementation: https://github.com/yuanli2333/Teacher-free-Knowledge-Distillation

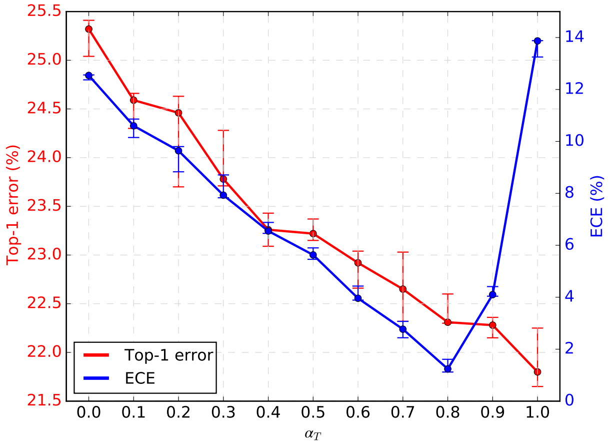
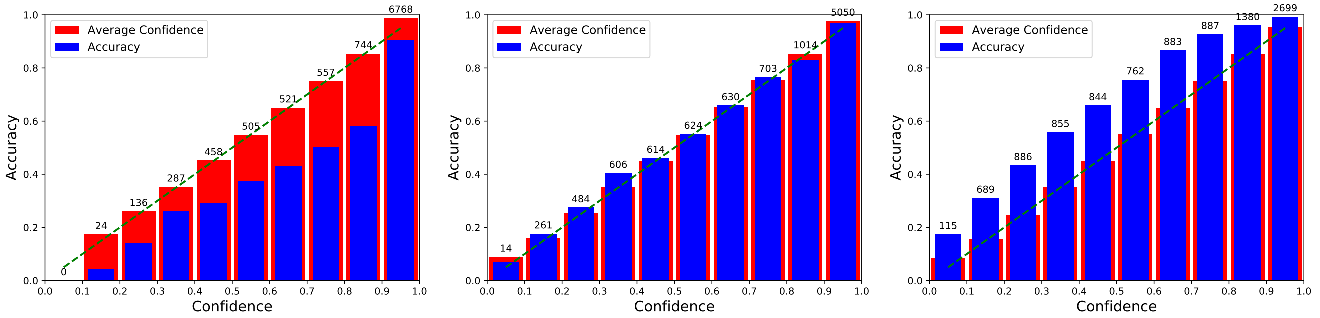


Figure S1. Validation top-1 error and ECE according to α_T from three repeated experiments on CIFAR-100 for ResNet-18. $\alpha_T = 0.8$ is chosen as the best one and used for all other experiments.



(a) $\alpha_T = 0.0$

(b) $\alpha_T = 0.8$

(c) $\alpha_T = 1.0$

Figure S2. Reliability diagrams on the validation dataset of CIFAR100 with ResNet-18+PS-KD. The number on the top of each bin represents the number of samples belonging to that bin.

Additionally, to examine the effect of using past predictions to soften hard targets, we conduct experiments with a fixed value of $\alpha_t \in \{0.1, 0.2, 0.4, 0.6, 0.8\}$ so that the effect of adjusting α_t is excluded. From the curves of NLL and top-1 error in Fig. S3, we observe that PS-KD with a fixed $\alpha_t = 0.1$ shows lower NLL and top-1 error than LS with $\epsilon = 0.1$ (refer to the shaded area on the curves), and the performances are improved as a fixed α_t increases. Therefore, it can be concluded that softening hard targets with predictions from the model itself is much better than just using a static softening operation like LS. To further investigate the effect of adjusting α_t, the curves from the linear growth strategy toward $\alpha_T = 0.8$ are also depicted. Compared to the curves from the fixed $\alpha_t = 0.8$, we conclude that the simplest approach, the linear growth, works surprisingly well for regularizing the model.

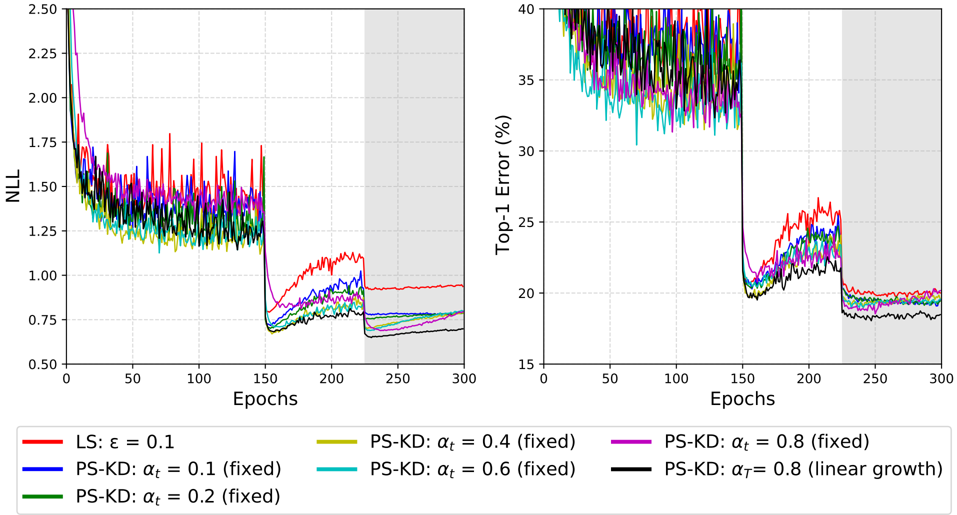
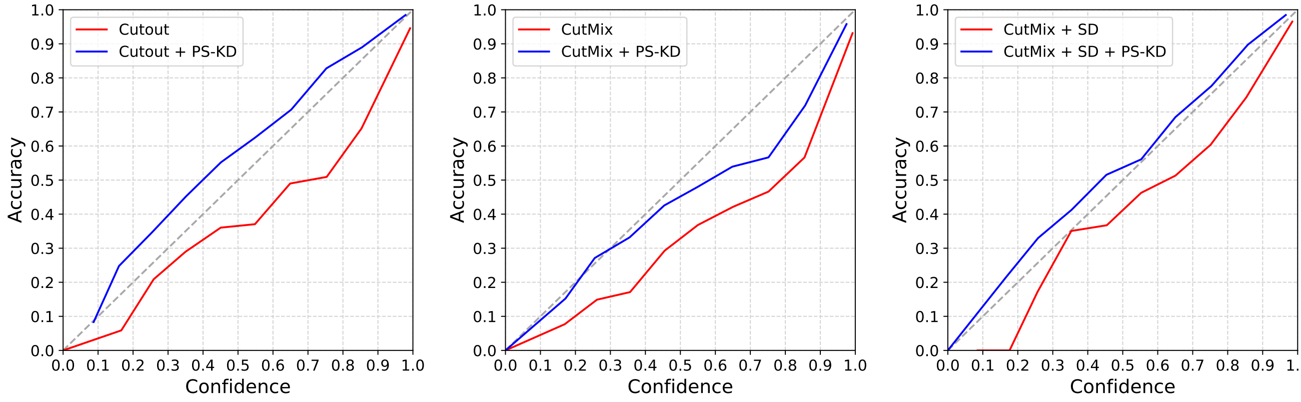


Figure S3. NLL (left) and top-1 error (right) curves on CIFAR-100 with different α_t values for DenseNet-121. Linear growth with $\alpha_T = 0.8$ achieves the lowest NLL and top-1 error.

Additional calibration plots Fig. S4 shows the calibration plots of existing regularization methods on CIFAR-100. From this figure, we can observe that the advanced regularization methods such as Cutout, CutMix, CutMix+SD benefit from PS-KD in terms of calibration.



(a) PyramidNet with Cutout

(b) PyramidNet with CutMix

(c) PyramidNet with CutMix + ShakeDrop

Figure S4. Calibration plots of advanced regularization methods on CIFAR-100 with PyramidNet. PS-KD provides additional benefits to existing methods in terms of calibration.

Extension results for self-KD methods combined with advanced data augmentations As summarized in Table S1, we provide additional experimental results: [Cutout, CutMix, CutMix+SD] + LS, CS-KD, and TF-KD on CIFAR 100 with PyramidNet. The results show that PS-KD can be effectively combined with advanced regularization techniques.

Model	**Top-1**	**Top-5**	**NLL**	**ECE**	**AURC**
+ Method	**Err (%)**	**Err (%)**		**(%)**	**($\times 10^3$)**
PyramidNet	16.80	3.69	0.73	8.04	36.95
+ LS	17.82	4.72	0.89	3.46	105.02
+ CS-KD	18.31	5.70	1.17	14.70	70.05
+ TF-KD	16.48	3.37	0.79	10.48	37.04
+ PS-KD	**15.49**	**3.08**	**0.56**	**1.83**	**32.14**
+ Cutout	16.05	3.42	0.67	7.15	33.20
+ Cutout + LS	17.15	4.38	0.82	4.65	82.61
+ Cutout + CS-KD	18.20	5.25	1.06	13.78	66.69
+ Cutout + TF-KD	16.29	3.18	0.74	9.77	35.78
+ Cutout + PS-KD	**14.82**	**2.86**	**0.54**	**3.69**	**29.77**
+ CutMix	15.62	3.38	0.68	8.16	34.60
+ CutMix + LS	15.68	3.66	0.70	**4.60**	37.71
+ CutMix + CS-KD	15.89	3.60	0.73	9.28	35.47
+ CutMix + TF-KD	16.61	3.29	0.66	7.47	36.57
+ CutMix + PS-KD	**15.03**	**2.91**	**0.58**	5.81	**30.22**
+ CutMix + SD	14.07	2.38	0.51	3.96	28.65
+ CutMix + SD + LS	14.05	2.37	0.54	**2.54**	33.09
+ CutMix + SD + CS-KD	14.99	2.56	0.56	3.27	34.40
+ CutMix + SD + TF-KD	15.34	2.58	0.53	3.31	31.41
+ CutMix + SD + PS-KD	**13.59**	**2.18**	**0.49**	3.46	**25.98**

Table S1. Performance evaluation of self-KD methods with advanced data augmentation techniques. The values averaged over three runs are reported. The best result is shown in boldface.

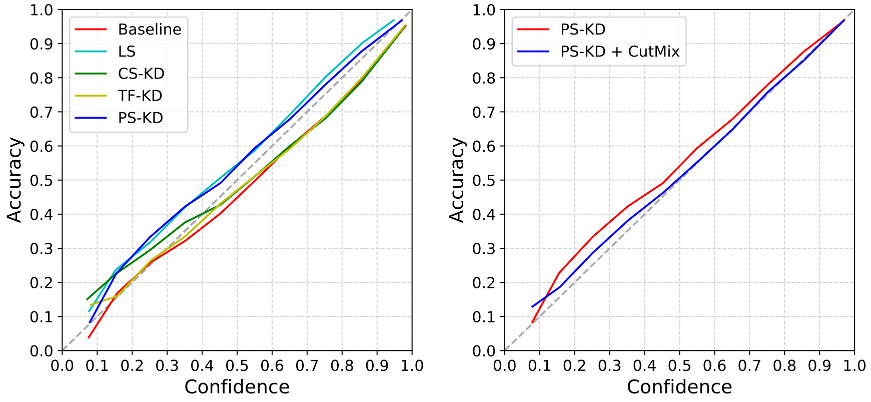
A.5. Experimental Results on ImageNet

Random search results of the hyperparameters. To find out the optimal hyperparmeter of CS-KD, TF-KD and PS-KD, we perform a random search of hyperparameters over five trials with ResNet-152 for a fair comparison. We set the mini-batch size to 512, and the other training setting is set to the same as ImageNet experiments in the main manuscript. For CS-KD, we consider the range of the hyperparameters as follow: $\tau \in \{1, 2, \cdots, 20\}$ and $\lambda_{cls} \in \{0.1, 0.5, 1, \cdots, 4\}$. For TF-KD, we use TF-KD$_{reg}$ method which shows better performance on ImageNet in the original paper [8]. We consider the hyperparamters, the temperature $\tau \in \{20, 30, 40\}$, weight $\alpha \in \{0.1, 0.2, \cdots, 0.5\}$ and probability for the ground-truth class $a \in \{0.90, 0.91, \cdots, 0.99\}$. For PS-KD, the range of $\alpha_T \in \{0.1, 0.2, \cdots, 1\}$ is used. The results are presented in Table S2.

Model **+ Method**	**Top-1** **Err (%)**	**Top-5** **Err (%)**	**NLL**	**ECE** **(%)**	**AURC** **($\times 10^3$)**
ResNet-152	21.95	6.16	0.89	5.08	61.64
+ LS	21.80	6.03	0.94	3.42	70.83
+ CS-KD ($\tau = 10, \lambda = 4$)	23.28	7.02	1.04	4.31	69.68
+ CS-KD ($\tau = 20, \lambda = 2$)	22.30	6.46	0.95	4.92	**54.13**
+ CS-KD ($\tau = 1, \lambda = 0.1$)	21.68	6.04	**0.85**	**1.46**	61.09
+ CS-KD ($\tau = 4, \lambda = 0.5$)	**21.67**	**6.01**	0.88	3.79	61.39
+ CS-KD ($\tau = 10, \lambda = 3$)	22.43	6.55	0.98	5.45	65.99
+ TF-KD ($\alpha = 0.3, \tau = 20, a = 0.91$)	22.72	6.49	0.92	4.69	65.30
+ TF-KD ($\alpha = 0.1, \tau = 40, a = 0.95$)	**22.66**	**6.46**	**0.91**	**4.61**	**64.29**
+ TF-KD ($\alpha = 0.2, \tau = 40, a = 0.97$)	22.99	6.66	0.93	5.13	65.69
+ TF-KD ($\alpha = 0.3, \tau = 40, a = 0.92$)	22.82	6.61	0.92	4.72	64.79
+ TF-KD ($\alpha = 0.1, \tau = 30, a = 0.92$)	22.74	6.52	0.92	5.25	64.76
+ PS-KD ($\alpha_T = 0.9$)	22.69	6.44	1.06	17.1	69.75
+ PS-KD ($\alpha_T = 0.5$)	21.67	5.92	0.88	7.33	63.19
+ PS-KD ($\alpha_T = 0.1$)	21.89	6.00	0.86	2.96	60.88
+ PS-KD ($\alpha_T = 0.3$)	**21.51**	**5.86**	**0.84**	**1.85**	**60.61**
+ PS-KD ($\alpha_T = 0.8$)	22.40	6.40	1.00	13.65	68.00

Table S2. Results over five trials of random search with ResNet-152. The best result for each method is shown in boldface.

Additional calibration plots Fig. S5 shows the calibration plots of comparison targets and CutMix. From this figure, we observe that PS-KD is better calibrated than other methods as well as improves calibration performance of the existing advanced regularization method, CutMix.



(a) ResNet-152 with other methods

(b) ResNet-152 with PS-KD + CutMix

Figure S5. Calibration plots on ImageNet with ResNet-152. (a) PS-KD shows slightly better performance compared to LS, CS-KD, and TF-KD. (b) PS-KD provides additional benefits to CutMix in terms of calibration.

Additional samples from ImageNet validation dataset. In Fig. S6, additional samples from ImageNet validation dataset and their predicted probabilities are presented. From these samples, we observe that PS-KD provides better outputs in the sense of human interpretation.

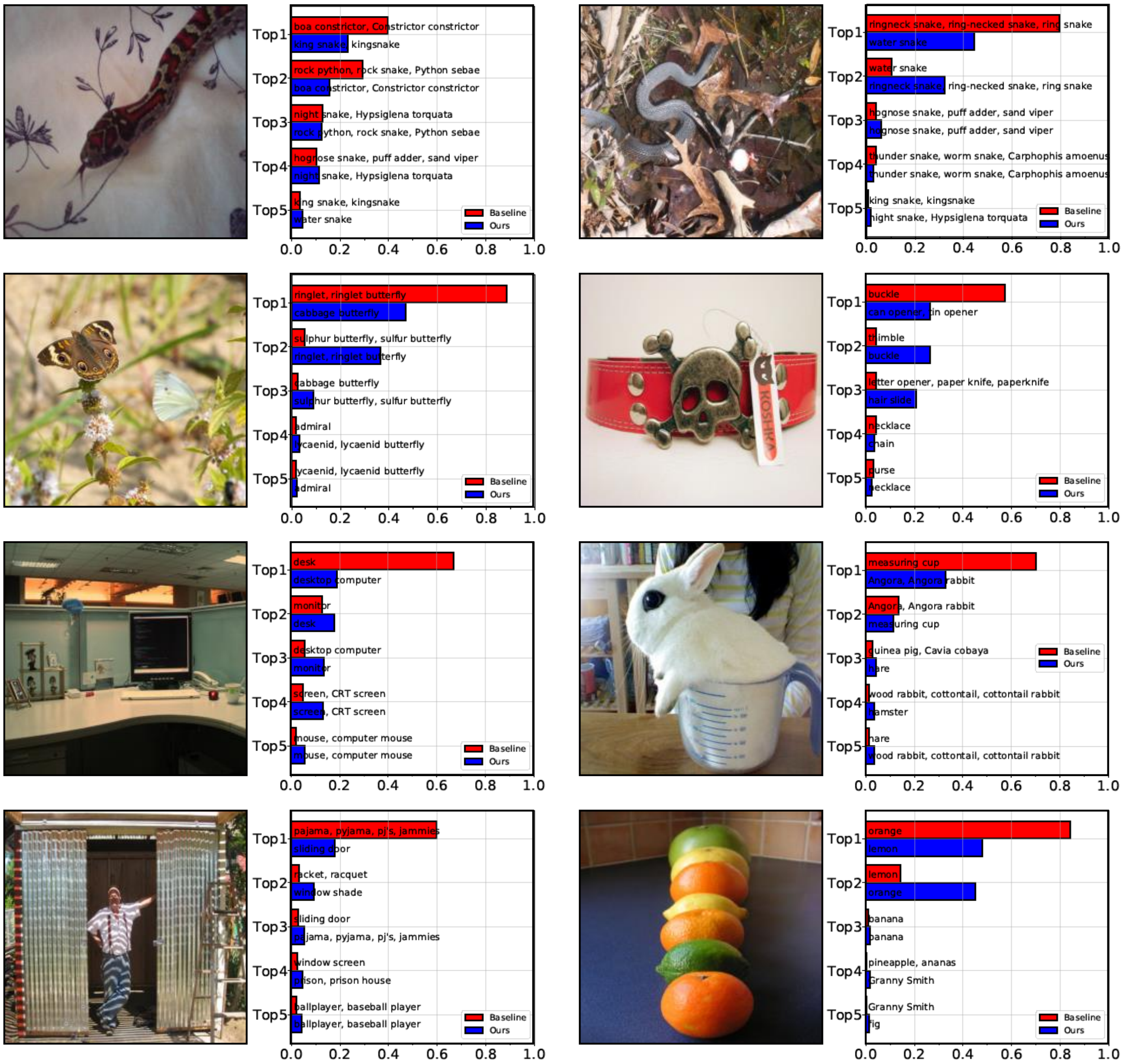


Figure S6. Predicted probabilities for sample images from the baseline and PS-KD. From the top left, the ground-truth labels of these images are ”king snake”, ”water snake”, ”cabbage butterfly”, ”buckle”, ”desk”, ”measuring cup”, ”sliding door” and ”orange”, respectively.

B. Object Detection

Table S3 shows the values of average precision (AP) over all classes. PS-KD shows higher AP values than the baseline and other methods (i.e., LS, CS-KD and TF-KD) for 10 classes out of 20 classes.

Method	Average Precision																				mAP
	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Dining Table	Dog	Horse	Mortor Bike	Person	Potted Plant	Sheep	Sofa	Train	TV Monitor	
ResNet-152	78.87	84.70	79.64	72.60	63.45	86.68	87.38	88.08	63.27	83.29	69.14	87.10	87.27	82.72	79.31	52.29	78.77	78.50	84.36	77.84	78.26
+LS	81.52	84.97	79.53	69.58	63.71	83.88	87.25	87.49	64.39	85.15	73.17	**87.82**	87.32	80.76	80.97	50.88	79.60	78.87	87.18	74.81	78.44
+CS-KD	79.93	82.58	78.97	70.91	**65.34**	84.56	87.20	87.40	62.18	83.96	72.47	85.17	87.41	82.08	81.32	53.43	82.04	76.97	85.40	77.33	78.33
+TF-KD	79.97	85.98	78.57	70.79	61.45	85.96	87.69	87.86	61.24	85.00	74.04	85.80	86.68	80.92	79.02	52.25	81.68	77.81	85.53	77.39	78.28
+PS-KD	79.59	83.60	**79.74**	70.24	64.64	**87.30**	**88.39**	88.04	**65.48**	**86.75**	**77.98**	87.68	87.55	**85.07**	**81.42**	53.15	**82.50**	**79.78**	83.64	77.45	79.50
+PS-KD + CutMix	**83.54**	**85.99**	79.23	**72.69**	65.08	86.66	88.23	**88.93**	64.14	**86.75**	71.98	87.10	**87.73**	84.01	81.34	**56.05**	78.07	79.76	**87.92**	**79.21**	**79.72**

Table S3. APs over all classes on PASCAL VOC 2007 testset. The best result for each class is in bold.

C. Machine Translation

C.1. Evaluation Metrics

BLEU. BLEU (Bilingual Evaluation Understudy) is an algorithm for numerically measuring the quality of machine translation from one natural language to another one. By using human translation as a reference, BLEU evaluates the quality of machine translation via two aspects. One is how many n-grams in the translated output of a model appears in the reference. If the more n-grams appear in both machine translation and human translation, the quality of machine translation is considered as better. We set n to 4, which is generally used for the evaluation. Another aspect of BLEU is the length of machine translated sentence. If we evaluate the performance by using only n-grams, very short sentence with only few words in the reference will have nearly a perfect score. To prevent this, an additional term comparing the length of machine translation and human translation is considered in the calculation of BLEU.

C.2. Dataset

Dataset. We use IWSLT15 English to German (EN-DE) and German to English (DE-EN) dataset. It consists of 191K training sentence pairs[3], and 8,300 pairs of the training data are used for validation. We concatenate dev2010, dev2012, tst2010, tst2011, tst2012, tst2013 datasets for a test set.

References

[1] M. Degroot and S. Fienberg. The comparison and evaluation of forecasters. *The Statistician*, 32:12–22, 1983. 1

[2] Terrance Devries and Graham W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2

[3] Yonatan Geifman, Guy Uziel, and Ran El-Yaniv. Bias-reduced uncertainty estimation for deep neural classifiers. In *International Conference on Learning Representations*, 2019. 1

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017. 1

[5] Mahdi Pakdaman Naeini, Gregory Cooper, and Milos Hauskrecht. Obtaining well calibrated probabilities using bayesian binning. In *AAAI Conference on Artificial Intelligence*, 2015. 1

[3] The dataset can be downloaded from https://https://wit3.fbk.eu/2015-01.

[6] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1

[7] Y. Yamada, M. Iwamura, T. Akiba, and K. Kise. Shakedrop regularization for deep residual learning. *IEEE Access*, 7:186126–186136, 2019. 2

[8] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3903–3911, 2020. 2, 6

[9] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *The IEEE International Conference on Computer Vision*, 2019. 2

[10] Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *The IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 2

[11] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018. 2