Supplementary Material for: Feature-map-level Online Adversarial Knowledge Distillation

Inseop Chung [1] **SeongUk Park** [1] **Jangho Kim** [1] **Nojun Kwak** [1]

A. Formulation of cross-entropy loss and KL divergence loss

We use two loss terms for logit-based learning, one is the conventional cross-entropy loss and the other is a mimicry loss based on Kullback Leibler divergence (KLD). Here, we formulate the cross-entropy loss and the KL divergence loss for two networks. Assume that we are given a set of classification data with N samples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ where each label y_i belongs to one of C classes, $y_i \in \mathcal{Y} = \{1, 2, \cdots, C\}$. The logit produced by a network is denoted as $z_k = \{z_k^1, z_k^2, \cdots, z_k^C\}$ where k refers to the kth network. The final class probability of a class c given a sample x_i to a network Θ_1 is computed as follows:

$$\mathcal{N}_{\Theta_1}(x_i) = \sigma_i^c(z_1/T)$$

$$\sigma_i^c(z_1/T) = \frac{exp(z_1^c/T)}{\sum_{c=1}^{C} exp(z_1^c/T)}$$

The temperature term T is used to control the level of smoothness in probabilities. When $T = 1$, it is the same as the original softmax. As the temperature term T goes up, it creates a more softened probability distribution. For training multi-class classification model, we adopt a cross-entropy(CE) loss between the ground-truth label and the outputs predicted by the model:

$$\mathcal{L}_{ce}(y, \sigma(z_1/1)) = -\sum_{i=1}^{N}\sum_{c=1}^{C} \delta(y_i, c)\log(\sigma_i^c(z_1/1))$$

The Dirac delta term $\delta(y_i, c)$ returns 1 if $y_i = c$ else 0. While the CE loss is between the ground-truth labels and the outputs of the model, the mimicry loss is the KL distance between the outputs of two training networks. The mimicry loss provides an extra information from the peer network so that the network can improve its generalization performance. We use the softened probability of each network at a temperature of 3. The mimicry loss from a network Θ_2 to network Θ_1 is measured as follows:

$$\mathcal{L}_{kl}(\sigma(z_2/T), \sigma(z_1/T)) = \sum_{i=1}^{N}\sum_{c=1}^{C} \sigma_i^c(z_2/T)\log(\frac{\sigma_i^c(z_2/T)}{\sigma_i^c(z_1/T)})$$

Therefore, the overall logit-based loss for a network Θ_1 is defined as:

$$\mathcal{L}_{logit} = \mathcal{L}_{ce}(y, \sigma(z_1/1)) + T^2 \times \mathcal{L}_{kl}(\sigma(z_2/T), \sigma(z_1/T))$$

We multiply the KL loss term with T^2 because the gradients produced by the soft targets are scaled by $1/T^2$. The logit-based loss trains the networks to predict the correct truth label along with matching the outputs of the peer-network, enabling to share the knowledge at logit-level.

B. Schematic of Cyclic-learning framework

Figure1 is the schematic of our cyclic-learning framework for training 3 networks simultaneously. As it can be seen in the figure, the three different networks transfer knowledge in cyclic manner. Network$\Theta 1$ distills its knowledge to network$\Theta 2$, network$\Theta 2$ distills to network$\Theta 3$ and network$\Theta 3$ distills back to network$\Theta 1$. Also note that the knowledge is transferred both at logit-level and feature map-level. At logit-level the KL divergence loss, $\mathcal{L}_{kl}$ is applied between the logit of each network and at feature map-level, the distillation is indirectly conducted via discriminators. For example, between network$\Theta 1$ and network$\Theta 2$, a discriminator D_2 is making decision based on from which network the feature map is generated. It is trained to output 1 if the feature map is from network$\Theta 1$ and 0 if it is from network$\Theta 2$. The goal of network$\Theta 2$ is to fool its corresponding discriminator D_2 so that it can learn the distribution of feature map generated from network$\Theta 1$. Each network also learns from ground truth labels with conventional cross-entropy loss, $\mathcal{L}_{ce}$.

C. Grad-cam visualization

Figure 2 and 3 shows the Grad-cam visualization of different distillation methods using more samples from the CIFAR-

[1] Graduate School of Convergence Science and Technology, Seoul National University, Seoul, South Korea. Correspondence to: Inseop Chung <jis3613@snu.ac.kr>, Nojun Kwak <nojuk@snu.ac.kr>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020. Copyright 2020 by the author(s).

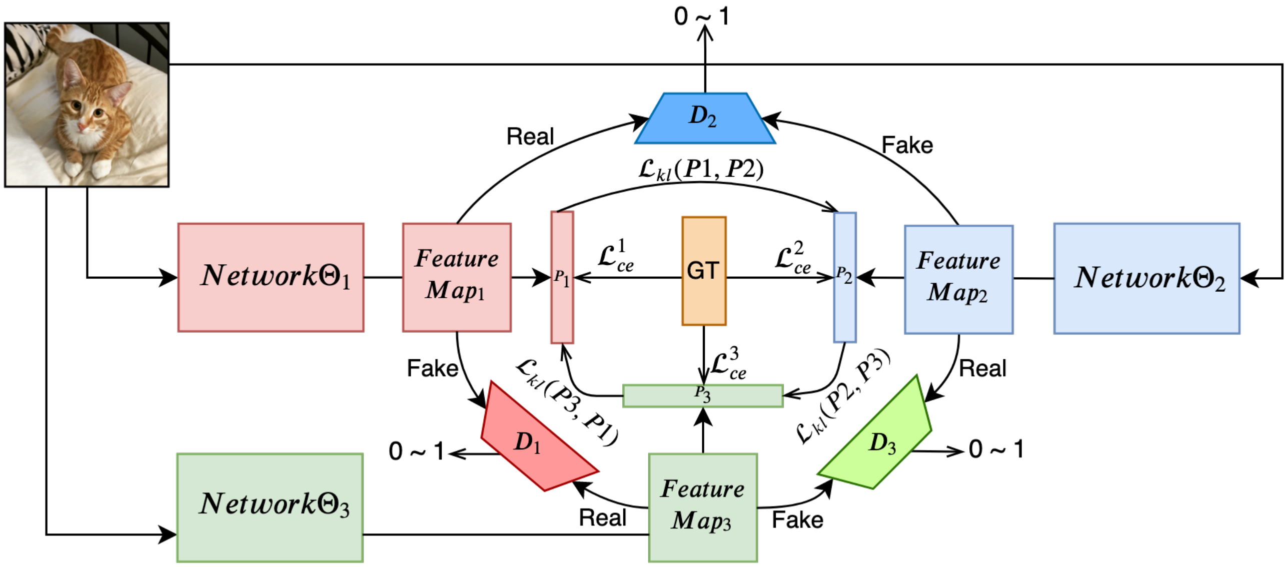

Figure 1. Schematic of cyclic-learning framework for training 3 networks simultaneously.

100 test set. As previously explained in qualitative analysis from Section 5.2, L_1+KD produces identical networks making them to output exactly the same feature map. Grad-cam visualization of Net1 and Net2 trained by L_1+KD is highlighting the exact same region. It indicates that the two networks have become the same networks that see the same spatial correlation and features of given image. L_1+KD just copies the result of each network and does not transfer any proper knowledge. The feature map visualization of DML and AFD(ours) has different features for each network, Net1 and Net2. Each network benefits from the other network by distillation while keeping its learned features and spatial information. Thus it does not decline by distilling from the other network. Interesting fact is that the feature maps of networks trained by our method do not look the same for Net1 and Net2 even though they transfer knowledge at feature map level. It rather improves its performance better than DML which distills knowledge only at logit-level.

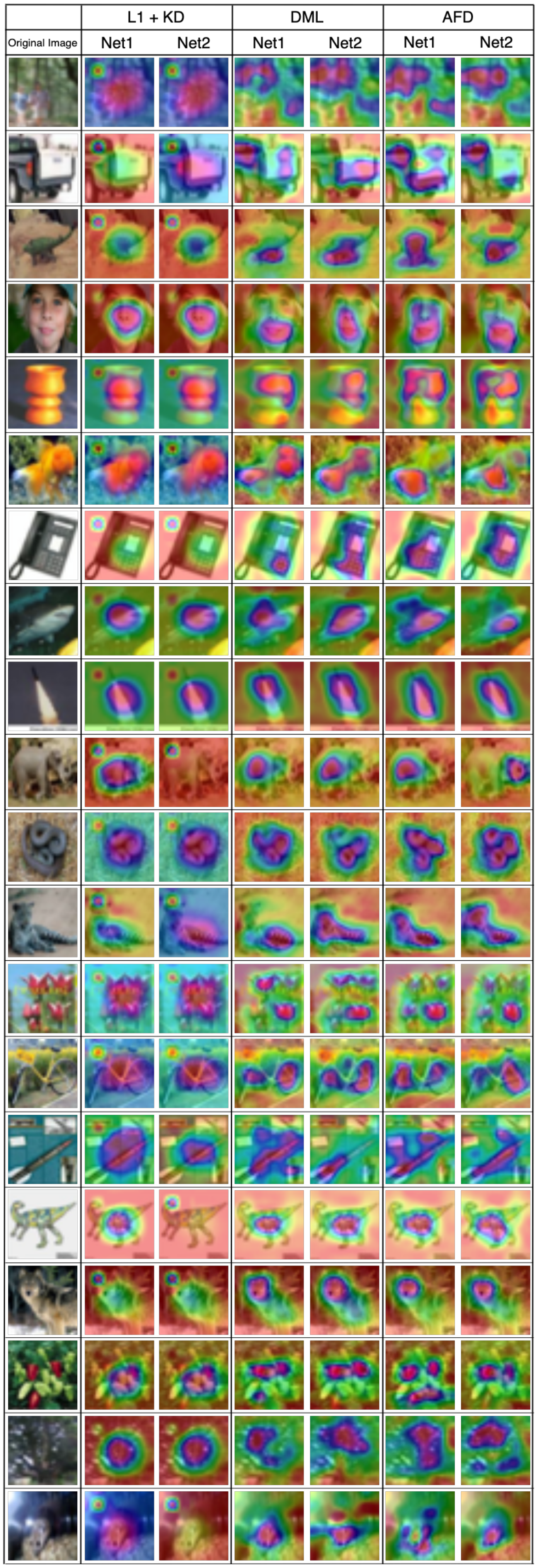

Figure 2. Grad-cam visualization of ResNet-32 trained with different distillation methods

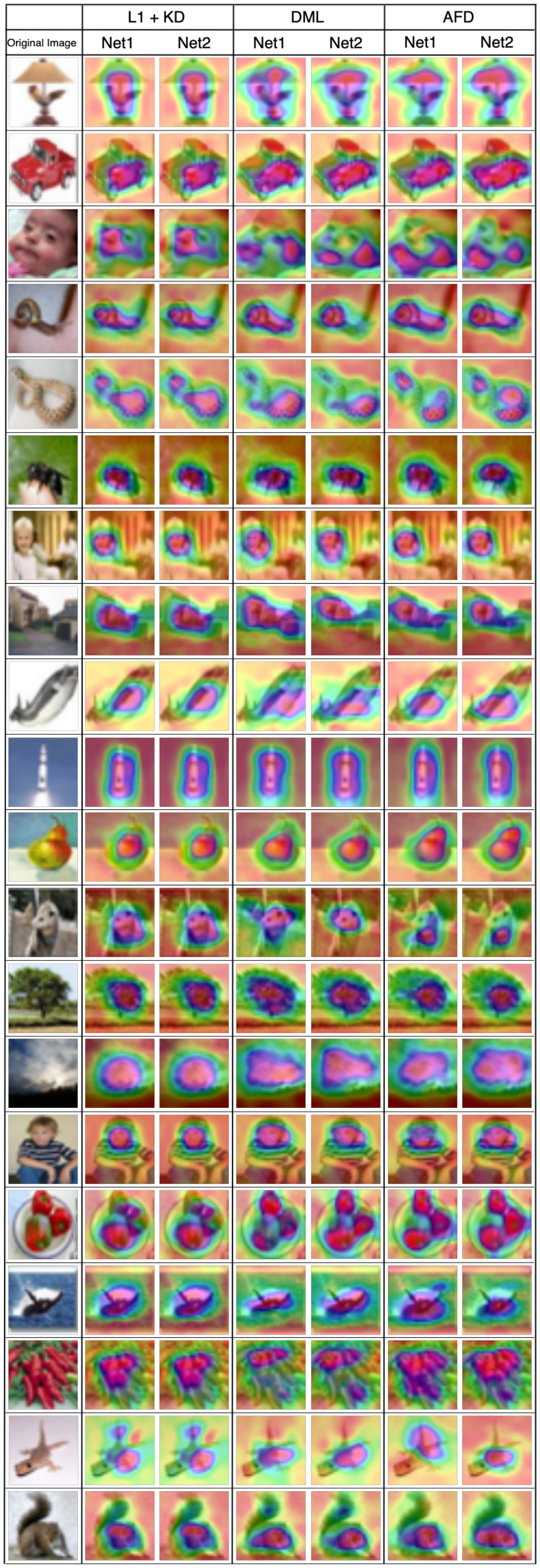

Figure 3. Grad-cam visualization of WRN-16-2 trained with different distillation methods