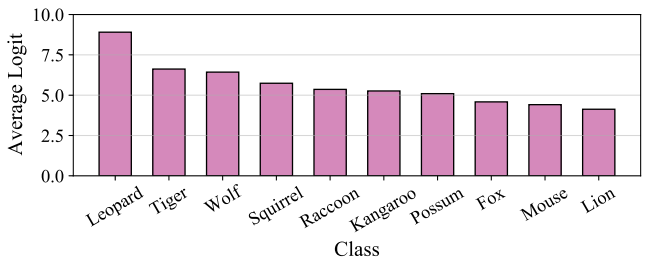
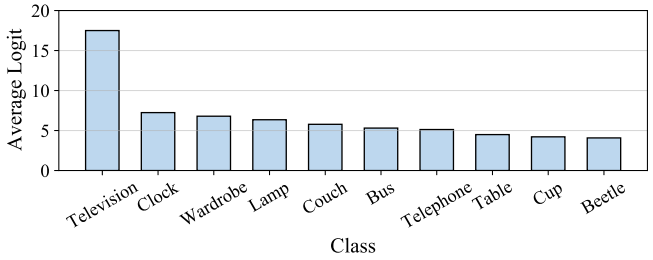
Knowledge Distillation with Refined Logits

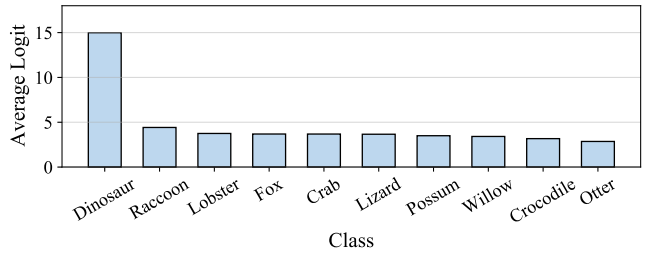
Supplementary Material



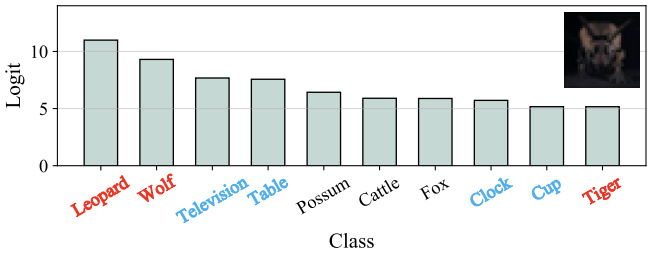
(a) Results based on **incorrect** samples for "leopard"



(b) Results based on **correct** samples for "television"



(c) Results based on **correct** samples for "dinosaur"



(d) Results based on **single incorrect** sample for "dinosaur"

Figure 1. Prediction results for different classes on the CIFAR-100 validation set. The model for calculation is ResNet110. Bold colored classes in (d) denote that they are among the top 10 classes most similar to either "leopard" (as inferred from (a)) or "television" (as inferred from (b)), yet they do not appear in the top 10 classes most similar to "dinosaur" (as inferred from (c)).

A. Class Correlation

We present results from a pilot experiment, affirming that scenarios similar to the one depicted in the paper's Figure 1 do indeed exist. We choose samples specifically from 3 classes in CIFAR-100: leopard, television, and dinosaur. Utilizing a well-trained model, we obtain the top 10 output logits. The corresponding results are displayed in Figure 1.

Figure 1(a), (b), and (c) illustrate that the model's class correlations align with our understanding, regardless of whether the prediction is correct or not. For instance, leopard exhibits higher similarity to tiger, while television resembles clock more closely. Across all samples, if the label is "leopard", the model does not improve prediction confidence or accuracy by decreasing the logit values of the label's actually similar classes (such as "tiger").

Figure 1(d) reveals that even in the output logits of a single sample, this inter-class relationship is well-maintained. For the sample labeled "dinosaur", the top 3 predictions contain "leopard" and "television". Consequently, within its top 10 predicted classes, the proportion of classes similar to "leopard" or "television" is significantly high, while classes resembling "dinosaur" are rare. In this case, simply correcting the true class (e.g., using augment or swap operations [1, 2, 7]) can result in significant deviations from the overall class correlations shown in Figure 1(c), which could negatively impact the distillation performance.

B. Training Efficiency

Figure 2 presents the training time, accuracy, and extra parameters of various methods. Notably, feature distillation methods (FitNet, OFD, ReviewKD, and CRD) yield a substantial performance enhancement compared to KD, albeit at the cost of a significant increase in training time and extra parameters. In contrast, our RLD method, maintaining the same training time as KD and introducing no extra parameters, delivers superior performance.

C. Implementation Details

We follow the conventional experimental settings of previous works [4, 6, 8] and use Pytorch [3] for our experiments.

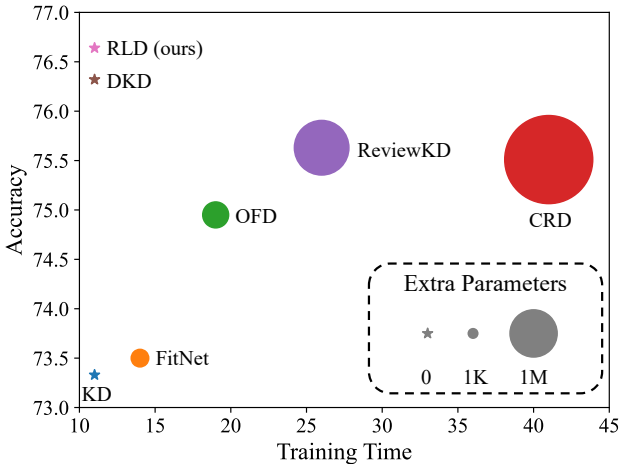


Figure 2. Batch training time (ms) *vs.* top-1 validation accuracy (%) on the CIFAR-100 dataset. The teacher is ResNet32×4, and the student is ResNet8×4. Larger circle denotes more extra parameters.

C.1. CIFAR-100

When training on CIFAR-100, the batch size, number of epochs, weight decay, and momentum are set to 64, 240, 5e-4, and 0.9, respectively. The initial learning rates are 0.01 for ShuffleNet and MobileNet, and 0.05 for other model architectures. The learning rate is divided by 10 at 150, 180 and 210 epochs. The optimizer is SGD [5]. The training data is augmented using RandomCrop and RandomHorizontalFlip operators.

For the hyper-parameters involved in RLD, we follow DKD [8] and LSKD [4] to set different values for different distillation pairs. We always set α to 1, and determine the optimal β and τ using grid search from the range of $\{2, 4, 8, 16\}$ and $\{2, 3, 4, 5\}$, respectively.

Our experiments are carried out on the server equipped with NVIDIA GeForce RTX 2080 Ti GPUs. Each GPU has 11 GB of memory. Only one GPU is used per experiment. The server's operating system is Ubuntu 18.04 LTS.

C.2. ImageNet

When training on ImageNet, the batch size, number of epochs, weight decay, and momentum are set to 512, 100, 1e-4, and 0.9, respectively. The initial learning rate is 0.2 and is divided by 10 every 30 epochs. The optimizer is SGD [5]. The training data is augmented using RandomResizedCrop and RandomHorizontalFlip operators.

For the hyper-parameters involved in RLD, we follow DKD [8] and LSKD [4] to set different values for different distillation pairs. We always set α to 1, and determine the optimal β and τ using grid search from the range of $\{0.5, 1, 2, 3\}$ and $\{1, 2\}$, respectively.

Our experiments are carried out on the server equipped with NVIDIA A800 Tensor Core GPUs. Each GPU has 80 GB of memory. Only one GPU is used per experiment. The server's operating system is Ubuntu 18.04 LTS.

References

[1] Qizhi Cao, Kaibing Zhang, Xin He, and Junge Shen. Be an excellent student: Review, preview, and correction. *IEEE Signal Processing Letters*, 30:1722–1726, 2023. 1

[2] Weichao Lan, Yiu-ming Cheung, Qing Xu, Buhua Liu, Zhikai Hu, Mengke Li, and Zhenghua Chen. Improve knowledge distillation via label revision and data selection. *arXiv preprint arXiv:2404.03693*, 2024. 1

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 1

[4] Shangquan Sun, Wenqi Ren, Jingzhi Li, Rui Wang, and Xiaochun Cao. Logit standardization in knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15731–15740, 2024. 1, 2

[5] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 2

[6] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. In *International Conference on Learning Representations*, 2020. 1

[7] Tiancheng Wen, Shenqi Lai, and Xueming Qian. Preparing lessons: Improve knowledge distillation with better supervision. *Neurocomputing*, 454:25–33, 2021. 1

[8] Borui Zhao, Quan Cui, Renjie Song, Yiyu Qiu, and Jiajun Liang. Decoupled knowledge distillation. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 11953–11962, 2022. 1, 2