Extracurricular Learning: Knowledge Transfer Beyond Empirical Distribution (Supplementary Material)

Hadi Pouransari
Apple
mpouransari@apple.com

Mojan Javaheripi
UCSD*
mojavahe@ucsd.edu

Vinay Sharma
Apple
sharma.vinay@apple.com

Oncel Tuzel
Apple
ctuzel@apple.com

A. Training details

A.1. Gaze Estimation

We used the MPIIGaze dataset [19, 20] that contains 45,000 annotated eye images of 15 persons (3,000 images per person divided equally between left and right eyes). We followed the leave-one-person-out evaluation process similar to the original works [19, 20]. We split the data to 20% validation and 80% training sets (3 randomly selected persons are held-out for validation and 12 persons for training). We run each experiment three times with a different random seed and report average error. We followed the implementation of original works [19, 20] for training, and used two existing architectures for student and teacher: the student is a 4-layer LeNet [9] and the teacher is a 9-layer PreAct-ResNet [5] trained with MixUp. The models output a two-dimensional vector that predicts the gaze vector. When estimating the uncertainty, we use an isotropic Gaussian $\mathcal{N}(\mu, \sigma\mathbb{I})$ to model the output distribution. Therefore, the network output is three-dimensional. We set weight decay to 10^{-4}, learning rate to 10^{-4} for LeNet and 10^{-3} for ResNet that is decayed by a factor of 10 after 30 and 36 epochs. All models are trained using ADAM optimizer [8] for 40 epochs with 0.9 momentum and batch-size of 32.

A.2. ResNet-18 on CIFAR100

We followed the same setup as [3] to train the ResNet-18 model [5]. Weight decay is 5×10^{-4}, learning rate is 0.1, and is decayed by a factor of 5 after 120, 240, and 320 epochs and the model is trained for 400 epochs. For all experiments, we use standard random cropping and horizontal flipping augmentations, and train with Nesterov [12] accelerated SGD with 0.9 momentum and batch-size of 128.

A.3. PyramidNet-200 on CIFAR100

We use the same training setup as in [17], namely, PyramidNet [4] initialized with depth 200 and $\tilde{\alpha} = 240$, weight decay of 10^{-4}, learning rate of 0.25 that is decayed by a factor of 10 after 150 and 225 epochs. For all experiments we use standard random cropping and horizontal flipping augmentations and train with Nesterov accelerated SGD for 300 epochs with 0.9 momentum and batch-size of 64.

A.4. BinaryNet on CIFAR100

We implemented Binary-Weight [2, 14] ResNet-18 architecture, where all weights (with the exception of the first and the last layers) are represented with 1-bit. We use the binary architecture in [14], and training setup in [10], namely, weight decay is zero, learning rate is 2×10^{-4} that is decayed by a factor of 10 after 150 and 250 epochs. For all experiments, we use standard random cropping and horizontal flipping augmentations and train with ADAM optimizer [8] for 350 epochs with 0.9 momentum and batch-size of 128. The implementation is the same as [13].

A.5. ResNet on ImageNet

We use the training setup introduced in [6]: the weight decay is 10^{-4} and learning rate is linearly warmed-up during the first 5 epochs from 0.1 to 0.4, and then decayed to 0 by a cosine function. For all experiments, we use SGD with Nesterov with batch-size of 1024, and apply standard data augmentations during training: random crop and resize to 224×224, random horizontal flipping, color jittering, and lightening. We resize the images to 256×256 followed by a center cropping to 224×224 during test. We use 300 epochs to train all models similar to [17]. We use regular ResNet-50 and ResNet-101 architectures [5] (not the D variant introduced in [6]). To reduce under-fitting for XCL, we used $10\times$ smaller weight decay (10^{-5}). Note that using a reduced weight decay did not help for other methods.

A.6. XCL with GAN-Generated Synthetic Data

Transfer-set using GAN can be generated online or offline. In the online setting, in every batch, we generate a new set of images via GAN. This setting is used when GPU memory

*Work is done while doing internship at Apple.

is sufficient to generate a batch of samples in parallel with distillation (e.g. for CIFAR100 benchmark). In the offline setting, we sample a large dataset using GAN, use the teacher to obtain soft labels, and store the extended transfer-set to be used in distillation.

For the XCL-GAN in 2D gaze estimation experiment, we used the conditional GAN in [15] to sample eye images with random orientations in offline setting, adding ~ 485k examples to the original transfer-set.

For the classification tasks, we used BigGAN [1] to generate samples when using XCL-GAN. BigGAN architecture is a conditional GAN, where an embedding vector $\boldsymbol{v}_i$ is trained for each class i. At inference time, the generative model G gets a latent variable $\boldsymbol{z}$ and an embedding vector v_i to generate a random sample $G(\boldsymbol{z}; \boldsymbol{v}_i)$ from class i. To sample with mixed class vectors (between two classes i and j) as in Section 4.2, we interpolate the embedding vectors: $\boldsymbol{v} = \lambda \boldsymbol{v}_i + (1-\lambda)\boldsymbol{v}_j$. We then generate a mixed sample $G(\boldsymbol{z}; \boldsymbol{v})$.

In CIFAR100 experiments, we used the online setting. In ImageNet experiments, we used the offline setting and sampled a transfer-set of 1M images using mixed-class labels. At each iteration of the training we sample half of the samples from the generated transfer-set and the other half from the original data points.

B. Results of ResNet-50 Training on ImageNet

The results are shown in Table 1. Compared to the standard KD we observe that XCL obtains **33**% reduction in the teacher-student accuracy gap.

method	val top-1	val top-1 gap	val top-5	V2-A top-1	V2-B top-1	V2-C top-1
ERM	77.3		93.6	74.5	65.6	79.4
+MixUp [18]	77.8	N/A	93.9	74.9	66.4	79.7
+CutMix [17]	78.7		94.3	75.5	66.9	80.2
KD [7]	79.2	2.4 (-)	94.3	75.6	67.3	80.6
XCL-Mix	**80.0**	**1.6** (-33%)	**95.0**	**77.2**	**68.2**	**81.3**

Table 1: Accuracies (%) of the ResNet-50 model trained on the ImageNet dataset. Teacher is a ResNet-152-D model trained with CutMix (top-1 acc. = 81.6%). The std of XCL val top-1 is $\simeq 0.1$.

C. Effect of Teacher Model

In this section, we explore alternative choices of the teacher τ. We use XCL-Mix with the same training setup as Section 5.2. We compare alternative choices of teacher in Table 2. Each teacher is an ensemble of 8 instances of the given model, trained with different initializations. We observe a general trend that a more accurate teacher results in a more accurate student. [11] observed that when teacher is trained with Label Smoothing (LS), it is more accurate, but can transfer less knowledge to the student. We observe that using XCL, a teacher trained with LS not only is more accurate but also trains a more accurate student.

teacher	$\hat{H}$ (%)	teacher top-1 (%)	student top-1 (%)
ResNet-18	23.4	81.4	80.2 ± 0.1
+LS ε=0.1	44.9	82.5	80.9 ± 0.2
+MixUp	31.0	83.1	81.1 ± 0.2
+CutMix	28.2	84.6	83.1 ± 0.2
Pyr.+CutMix	15.1	87.5	83.8 ± 0.1

Table 2: Analysis of different teachers. Each teacher is an ensemble of 8 models shown in each row.

D. Analysis of Student Size

We investigate the effect of student size on the performance of XCL and other baseline methods. Figures 1a and 1b show the student model accuracy as a function of model size (changed by scaling the channel widths) for both a full precision student and a binary quantized student, respectively. As seen, XCL consistently outperforms ERM and MixUp augmentation, as well as the standard KD which uses the empirical distribution as the transfer-set. It is also worth mentioning that the binary model has a better error-size trade-off curve compared to the full precision model.

E. Label Smoothing

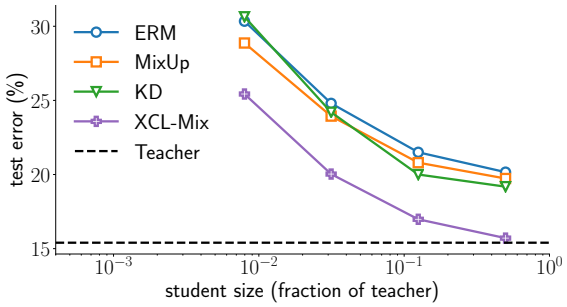
Label smoothing (LS) [16] with parameter ε replaces ground truth labels with:

$$y^j = \quad 1-\varepsilon \quad \text{if: } j \text{ is the correct class} \quad \text{else: } \frac{\varepsilon}{c-1} \quad (1)$$

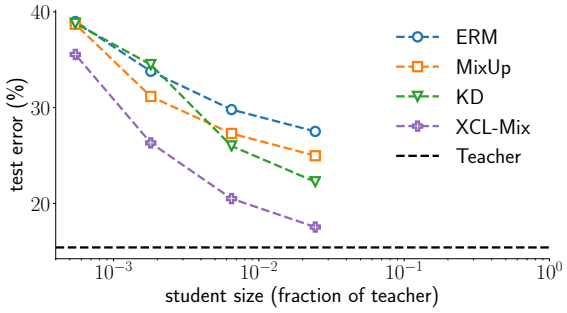
This method artificially increases label entropy. The results of ERM training with LS are reported in Table 3. For all experiments we use the CIFAR100 dataset and ResNet18 architecture as described in Section 5.2. Using $\epsilon = 0.1$, LS achieves 1.5% improvement over the baseline. Note that, finding an optimal ϵ requires extensive hyper-parameter tuning. XCL naturally obtains smooth labels, and without hyper-parameter tuning obtains significant accuracy improvement (by 3.8%) compared to the best LS.

F. Knowledge Distillation with Temperature Scaling

In KD [7], logits of the student and the teacher are inversely scaled by a temperature parameter T before softmax probabilities are computed. This smoothing strategy can slightly improve the knowledge distillation accuracy (+1.2% compared to KD without temperature scaling). We



(a)



(b)

Figure 1: Test error as a function of student model size for (a) full-precision and (b) binary training.

ε	$\hat{H}$ (%)	top-1 (%)
0.1	17.0	79.3 ± 0.3
0.18	28.2	78.8 ± 0.2
0.4	54.5	78.0 ± 0.3
0.8	90.7	78.3 ± 0.2

Table 3: Effect of label smoothing on ERM.

T	$\hat{H}$ (%)	top-1 (%)
1	10.5	80.0 ± 0.2
1.5	52.8	79.9 ± 0.3
2	81.2	80.2 ± 0.1
5	99.2	81.2 ± 0.3
10	99.9	81.1 ± 0.3

Table 4: Effect of temperature on KD.

use the CIFAR100 dataset and ResNet18 architecture as described in Section 5.2. Results are reported in Table 4.

We observe that XCL is not sensitive to temperature (Table 5). Note that finding an optimal T requires extensive hyper-parameter tuning. XCL does not require hyper-parameter tuning, and compared to the best KD with temperature scaling reduces the accuracy gap by 59%.

T	$\hat{H}$ (%)	top-1 (%)
1	28.2	83.1 ± 0.2
1.5	65.0	83.0 ± 0.1
2	85.8	83.1 ± 0.2
5	99.2	83.2 ± 0.2
10	99.9	83.1 ± 0.1

Table 5: Effect of temperature on XCL.

References

[1] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2018.

[2] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

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

[4] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5927–5935, 2017.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[6] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 558–567, 2019.

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

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[9] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[10] Brais Martinez, Jing Yang, Adrian Bulat, and Georgios Tzimiropoulos. Training binary neural networks with real-to-binary convolutions. *arXiv preprint arXiv:2003.11535*, 2020.

[11] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? In *Advances in Neural Information Processing Systems*, pages 4696–4705, 2019.

[12] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o(1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[13] Hadi Pouransari, Zhucheng Tu, and Oncel Tuzel. Least squares binary quantization of neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 698–699, 2020.

[14] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European conference on computer vision*, pages 525–542. Springer, 2016.

[15] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2107–2116, 2017.

[16] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[17] 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 *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[18] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[19] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Appearance-based gaze estimation in the wild. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015.

[20] Xucong Zhang, Yusuke Sugano, Mario Fritz, and Andreas Bulling. Mpiigaze: Real-world dataset and deep appearance-based gaze estimation. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):162–175, 2017.