Supplementary Material for Generalizing Teacher Networks for Effective Knowledge Distillation Across Student Architectures

Kuluhan Binici
kuluhan@comp.nus.edu.sg
Weiming Wu
weiming.wu@u.nus.edu
Tulika Mitra
tulika@comp.nus.edu.sg

School of Computing
National University of Singapore
Singapore

1 Student and Supernet Architectures

Supernet architecture The supernet we configure comprises of 6 reconfigurable layers each containing ResNet blocks of various depths and filter sizes, along with identity and zero candidate operations. Identity operation directly maps the inputs to the outputs, while zero operation simply replaces each value in the input tensor with zeros. These are jointly utilized to allow certain layers to be skipped for increasing the variation of sampled student architectures. The choices for the filter size of the ResNet blocks are 3 and 5 while the depth varies between 2 to 6 convolution operations. The inclusion of depth as a parameter that changes among iterations allows GTN teacher to be regularized by students of different depths.

Student Architectures To further show the diversity of the student architectures in the architecture pool represented by our supernet, we provide the Tables 3,2,1 where architectural details of NAS-discovered students are contained.

Layer (type)	In Channels	Out Channels	Kernel Size	Stride
ConvLayer	3	16	3	1
IdentityLayer	16	16	-	-
ResBlock	16	64	3	1
ResBlock	64	128	3	2
ResBlock	128	256	3	2
ConvLayer	256	1280	1	1
LinearLayer	1280	100	-	-

Table 1: s^{s}_{nas} architecture

Layer (type)	In Channels	Out Channels	Kernel Size	Stride
ConvLayer	3	16	3	1
IdentityLayer	16	16	-	-
ResBlock (x2)	16	64	3	1
ResBlock	64	64	3	1
IdentityLayer	64	64	-	-
ResBlock (x2)	64	128	5	2
ResBlock	128	256	3	2
ResBlock	256	512	3	2
ConvLayer	512	1280	1	1
LinearLayer	1280	100	-	-

Table 2: s^m_{nas} architecture

Layer (type)	In Channels	Out Channels	Kernel Size	Stride
ConvLayer	3	16	3	1
IdentityLayer	16	16	-	-
ResBlock (x2)	16	64	3	1
IdentityLayer	64	64	-	-
ResBlock (x3)	64	64	3	1
IdentityLayer	64	128	-	-
ResBlock (x3)	128	128	3	2
IdentityLayer	128	128	-	-
ResBlock (x3)	128	128	3	1
ResBlock (x3)	128	256	3	2
ResBlock (x3)	256	512	3	2
ConvLayer	512	1280	1	1
LinearLayer	1280	100	-	-

Table 3: s^l_{nas} architecture

2 Teacher model training details

Preparing teacher models for GTN training To prepare our GTN framework, we partition the reference supernet architecture into 4 blocks and graft the last three of them onto the corresponding blocks of the teacher model. This becomes the first among three student branches. For the other two branches we respectively graft the last two and one blocks of the partitioned supernet. When deciding on the partitioning, we try to ensure each block contains approximately the same number of layers. To obtain SFTN teachers, we follow the exact procedure described in [1] using the same set of hyperparameters. For a fair evaluation, the number of epochs used to train teachers are kept constant across all training methods that are included in our comparison. Lastly, for vanilla KD, we employ supervised training to obtain the teacher models. While experimenting with DKD and SCKD, we use the same teacher models as vanilla KD.

Training teacher models We initialize the weights of ResNet-32 and WRN40-2 teachers from scratch before training them with each method to obtain the teacher models for comparison. As for the EfficientNet-b0 teacher, we experimentally observed that it yields significantly better performance when pre-trained on a large-scale dataset like ImageNet as opposed to being trained from scratch. Therefore we initialize it from a checkpoint provided by the PyTorch Model Zoo and later fine-tune it using each method in our comparison.

All teacher models are trained for 240 epochs using SGD optimizer with an initial learning rate of 0.05. For both SFTN and GTN training, the softmax temperature is set as 1 and the same α values are used, which are 3 and 1 for CIFAR-100 and ImageNet-200 respectively. The learning rates of ResNet-32 and WRN40-2 teachers are reduced by a factor of 10 at the 150^{th}, 180^{th}, and 210^{th} epochs. As for the EfficientNet-b0, the learning rate is reduced by a factor of 10 at epochs 60, 90, and 120.

Fine-tuning EfficientNet-b0 teachers The pretrained checkpoint that we use to initialize the EfficientNet teachers is retrieved from PyTorch model zoo. The checkpoint was trained on ImageNet-1k, therefore we employed an initial fine-tuning stage of 15 epochs to adapt it to ImageNet-200. This is the base model that we use for further fine-tuning with different methods. Subsequently to obtain SFTN and GTN teachers, we applied both training procedures on this base model checkpoint, as secondary stages of fine-tuning. During this fine-tuning stage, we froze the parameters of the EfficientNet-b0 teacher and warmed up the student branches for the first 60 epochs. After 60^{th} epoch, the teacher branch is unfrozen and jointly optimized together with the student branches until 240 epochs are completed. To obtain the vanilla EfficientNet-b0 teachers, we fine-tune the base model for 35 epochs more using SGD with a 0.05 learning rate.

Accuracy Curve Visualisation To demonstrate that our teacher models are properly trained through GTN process, we plotted their accuracy curves together with those of the supernet branches that are grafted on them. Each plot, displayed in Figure 1, contains the accuracy progressions of the teacher and a single supernet branch. These plots show that the branch accuracies increase in conjunction with that of the teacher. This verifies that grafted supernet branches are being trained successfully and therefore can effectively regularize the training process of the teacher model. As expected, stu_branch_1 which contains the most number of supernet blocks (3 blocks) exhibits larger fluctuations in accuracy and in general slower progression compared to others. This is because, with each extra block grafted, the number of possible reference students that can be configured increases, requiring more trainable parameters to be optimized. As for the stu_branch_2, it contains less number of blocks (2 blocks) and therefore allows less variation in the reference models that can be sampled. Consequently, the accuracy curve associated with it is relatively more stable compared to stu_branch_1. Lastly, containing the least number of supernet blocks (1 block), stu_branch_3 displays the most similar progression to that of the teacher.

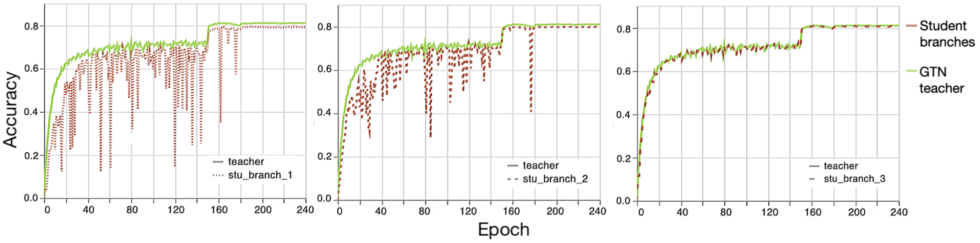

Figure 1: Accuracy curves of the ResNet-32 teacher model and each student branch during GTN training on CIFAR-100.

3 Knowledge distillation

Vanilla KD and DKD methods are used to transfer knowledge from teacher models to the students for evaluation. The student models are trained for 240 epochs with the softmax temperature in L_{KD} set as 4. SGD optimizers with learning rates of 0.05 and 0.1 are used for CIFAR-100 and ImageNet-200 datasets respectively. As the learning rate scheduler, we used cosine annealing with a period of 240 epochs.

4 Hardware Acceleration:

The experiments where we recorded the time cost of SFTN and GTN methods were accelerated using a single NVIDIA A100 GPU. For the rest of the experiments, a combination of NVIDIA A100 and V100 GPUs was utilized. The training process for the teachers was performed using two GPU devices, while one GPU was dedicated to training the students.

References

[1] Dae Young Park, Moon-Hyun Cha, Daesin Kim, Bohyung Han, et al. Learning student-friendly teacher networks for knowledge distillation. *Advances in Neural Information Processing Systems*, 34:13292–13303, 2021.