Online Knowledge Distillation by Temporal-Spatial Boosting

Chengcheng Li Zi Wang* Hairong Qi
University of Tennessee, Knoxville, TN, USA
{cli42,zwang84}@vols.utk.edu, hqi@utk.edu

Abstract

Online knowledge distillation (KD) mutually trains a group of student networks from scratch in a peer-teaching manner, eliminating the need for pre-trained teacher models. However, supervision from peers can be noisy, especially in the early stage of training. In this paper, we propose a novel method for online knowledge distillation by temporal-spatial boosting (TSB). The proposed method constructs superior "teachers" with two modules, temporal accumulator and spatial integrator. Specifically, the temporal accumulator leverages the previous outputs of networks during training and produces a representative prediction over all classes. Instead of merely imitating the outputs of other networks as in vanilla online KD, we further propose the so-called spatial integrator that consolidates the knowledge learned by all networks and yields a stronger instructor. The operations of these two modules are simple and straightforward, which can be computed efficiently on the fly during training. The proposed method can improve the efficiency of transferring effective knowledge as well as stabilize the training process. Experimental results on various benchmark datasets and network structures validate the effectiveness of the proposed method over the state-of-the-art.

1. Introduction

Recent advances in deep neural networks [23] have promoted tremendous applications in various tasks, e.g., object classification and detection [19, 35], image syntheses [8, 24, 34, 41, 47], natural language processing [52], game playing [30, 38], and biological imaging [4, 37, 51]. However, the deployment of these pre-trained networks on resource-limited devices is problematic [13, 2]. This is because most state-of-the-art networks contain millions of parameters, making it cumbersome and slow in real-world applications [39, 11]. To tackle this problem, network compression and acceleration approaches have been investigated. These efforts can be mainly categorized into lightweight structure design [13, 43, 16, 3, 27], network pruning [26, 50, 31, 25, 49], factorization [15], quantization [10, 45], knowledge distillation (KD) [1, 36, 12, 48, 22, 46, 9], etc. KD trains a compact network (referred to as "student") by imitating the soft targets generated by pre-trained large networks [12]. By doing so, the student network achieves better performance than being trained independently.

Classic KD [12, 36, 54], however, relies on pre-trained networks as the teacher, which is not always available. Online knowledge distillation, also known as deep mutual learning (DML) [57, 22, 9, 5], fills the gap by training student peers that learn from each other from scratch. Surprisingly, the student networks trained in this online KD manner even outperform those trained with stronger pre-trained teachers available using classic KD in many scenarios [57]. Although achieving competitive performance, training networks using existing online KD approaches is a highly dynamic procedure. This is because the network peers are interactive during the training process. Each student network learns its parameters from scratch and meanwhile acts as an instructor for other networks. However, training with developing instructors tends to involve more uncertainty [57].

To address the above issues, in this study, we propose a novel online KD approach that can stabilize the training process and improve the efficiency of knowledge transfer. We hypothesize that useful information is enclosed in the evolution of predictions generated by student networks during the training process. We thus construct a module referred to as *temporal accumulator* for each network, which tracks the softmax outputs (i.e., predictions) associated with individual training samples at every epoch. It incorporates previous outputs to generate more representative targets for other networks to learn via Kullback–Leibler (KL) divergence. The temporal accumulator fuses the learned knowledge over a certain window size in history, generating more robust and informative outputs as targets for peers. We further introduce a module that integrates the predictions of all student networks as an extra enhanced teacher at each iteration during training. We refer to it as the *spatial integrator*, since it incorporates information provided by different stu-

*Chengcheng Li and Zi Wang contributed equally.

dent networks. The proposed spatial integrator is inspired by model ensemble [1], which trains multiple different networks on the same dataset and combines the predictions of converged models to further improve the performance. Different from that, we integrate the predictions of student networks generated during the online training procedure and provide the integrated predictions as superior targets to supervise all student networks.

Our main contributions are summarized as follows.

- We propose the concept of temporal accumulation for online KD, which incorporates previous predictions of the networks during the training procedure to build representative and robust targets for peer networks, so as to stabilize the training process and improve the efficiency of transferring knowledge.
- We introduce spatial integration to the online KD training, which incorporates predictions of all student networks and generates superior integrated targets, taking advantage of the diversity among student peers.
- We evaluate our method with various benchmark datasets and network structures. Experimental results validate the performance improvement of our method over state-of-the-art online KD works.

2. Related work

2.1. Classic knowledge distillation

Although KD was first described back in 2006 [1], it did not draw too much attention until the influential work by Hinton et al. [12], where the KL divergence between the softmax outputs of a student network and the soft targets of a pre-trained teacher network is minimized. In addition to imitating the outputs of the pre-trained model, researchers have explored matching information between the teacher and the student at various levels to further improve the performance. For example, Fitnets [36] considers the teacher's intermediate hidden layers as hints to guide the training of the student network. Zagoruyko and Komodakis proposed to transfer the attention information of the teacher to the student [54]. Contrastive representation distillation (CRD) [42] brings the contrastive learning mechanisms to KD by utilizing the structural knowledge of the teacher network.

2.2. Online knowledge distillation

Pre-trained teachers may not be accessible in real-world applications due to several reasons such as business competition, storage cost, and privacy protection. To fill in the gaps, the concept of online knowledge distillation was proposed. As one of the earliest studies, deep mutual learning (DML) [57] trained a group of student networks simultaneously for image classification from scratch. Besides the common cross-entropy loss, the KL divergence is also calculated as the knowledge distillation loss, measuring the similarity of the softmax outputs of each student network and its peers. ONE [22] re-designed several popular networks such as ResNet [11] and ResNeXt [53] by adding multiple auxiliary branches on shared low-level layers. With such specific architectures and a gating module, predictions of all branches are integrated to provide a strong instructor for guiding the training of each branch. Although achieving competitive performance, networks need to be specially re-designed before training and pruned for deployment. KDCL [9] proposed to add different distortions to the training samples of each student network and designed several methods for combining the outputs of the networks via collaborative learning. AFD [5] leveraged the knowledge hidden in the feature maps of each model to further improve the performance of DML, with an adversarial training framework.

2.3. Ensemble learning

Ensemble learning has been widely investigated in various machine learning problems and applications. For example, it has been recognized as a common trick for improving the performance of neural networks, which trains a group of different models on the same data and then makes predictions by averaging their predictions [7, 56, 40, 20]. Laine and Aila proposed an ensemble approach for semi-supervised learning, which uses the outputs of the network during training to provide a consensus prediction for the unknown labels [21]. By doing so, the network can achieve competitive performance with only a small portion of labeled samples. Recent studies in deep-Q network [29] showed that the performance of the policy network can be improved by updating its parameters with soft ensemble targets, rather than using a fixed target network that periodically updates in reinforcement learning [28, 17]. In the area of KD, [12] showed that the knowledge learned with an ensemble of multiple models can be compressed into a single network. ONE [22] can be considered as an ensemble of multiple specifically-designed branches.

Inspired by these studies, we propose two modules that integrate information from different perspectives for online KD. The proposed temporal accumulator incorporates temporal information for each student network by combining previous predictions and current predictions. The proposed spatial integrator incorporates the information among different networks at each iteration during the training process.

3. The proposed approach

In this section, we present the proposed online KD approach with temporal-spatial boosting (TSB) in the context of image classification. The overall framework of TSB is illustrated in Fig. 1. For the sake of simplicity, we elaborate the proposed method with two student networks. Gener-

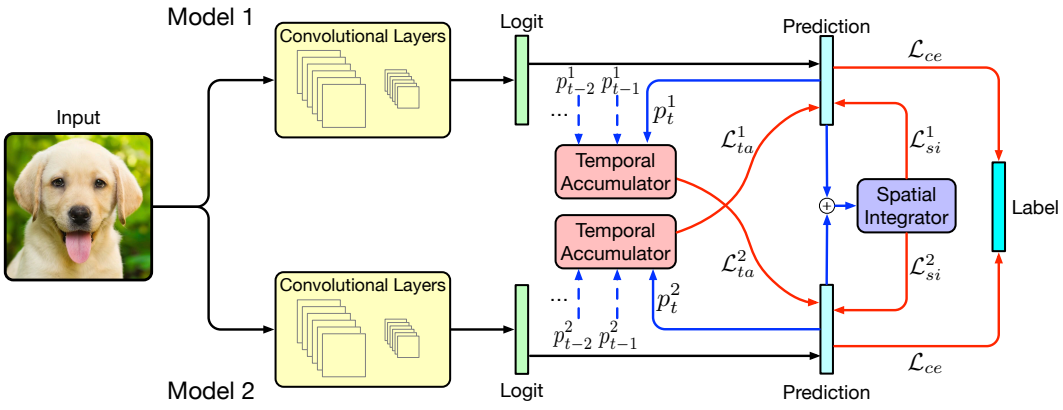


Figure 1. Framework of the proposed online KD approach with temporal-spatial boosting (TSB). Two student networks are trained simultaneously in a peer-teaching manner. p_t represents the predictions at current epoch, and p_{t-1} and p_{t-2} represent the historical predictions at epoch $t-1$ and $t-2$, respectively. The loss function for optimizing each network is composed of three components. $\mathcal{L}_{ce}$ represents the cross-entropy loss between the network's outputs and ground-truth labels. $\mathcal{L}_{ta}$ and $\mathcal{L}_{si}$ are the KL divergence of the network's outputs and the results of the other peer's temporal accumulator and the spatial integrator, respectively. Details are presented in Section 3.

alization to more networks is straightforward and will be discussed later in this section.

Online Knowledge Distillation Denote the training set $\mathcal{D} = \{(x_k, y_k)\}_{k=1}^{N}$, which consists of N image-label pairs from C classes. Suppose there are two networks, θ^1 and θ^2. It is worth noting that structures of these two networks are not necessarily identical. In the context of online KD, both networks are "students" that need to be trained from scratch. With a sample x fed in, the prediction of θ^i $(i \in \{1, 2\})$ over the class c $(c \in \{1, \cdots, C\})$ can be represented as $p^i(c|x, \theta^i)$ in Eq. (1).

$$p^i(c|x,\theta^i) = \frac{\exp\left(z_c^i/\tau\right)}{\sum_{l=1}^{C}\exp(z_l^i/\tau)}, \quad i = 1, 2, \quad c = 1, \cdots, C, \tag{1}$$

where z_c^i is the c-th logit of network θ^i, τ is the temperature used for softening the predictions. In the rest of the paper, we use $p^i(x, \theta^i)$ or p^i to represent the prediction of the network θ^i over C classes unless stated otherwise.

Student networks are trained simultaneously in a peer-teaching manner. Our loss function for optimizing each network consists of the classification loss and the knowledge distillation loss. The cross-entropy loss ($\mathcal{L}_{ce}$) is adopted as the classification loss to measure the error between the predictions and the ground-truth labels for each network separately. In this case, τ is set to 1 in Eq. (1). The basic idea of online KD is treating each other as the "teacher". To measure the effectiveness of distilling knowledge from a "teacher" to a "student", we use the Kullback–Leibler divergence to quantify the alignment of their predictions, which is represented with Eq. (2),

$$\mathcal{L}_{kl}(p^i, p^j) = \sum_{c=1}^{C} p^i(c|x,\theta^i)\log\frac{p^i(c|x,\theta^i)}{p^j(c|x,\theta^j)}, \tag{2}$$

where p^i and p^j represent the predictions of θ^i and θ^j over all C classes, respectively.

Supervision from peers can be noisy during the process of training from scratch, providing limited information to other networks. To construct an effective "teacher" for each network, we propose the following two novel modules.

Temporal Accumulator The first module is named the temporal accumulator, which leverages the history information of predictions to provide a superior "teacher". Let $Z^i \in \mathbb{R}^{N\times C}$ contain the accumulated predictions of network θ^i for all training samples, where each row is bundled with the unique index of a training sample. It is important to notice that during the optimization with gradient descent, training samples can be shuffled before being fed into the networks, but the indices of the training samples corresponding to those in Z^i need to be identical. Z^i can be represented as Eq. (3).

$$Z^i = [p_t^{ta,i}(x_1,\theta^i), p_t^{ta,i}(x_2,\theta^i), \cdots, p_t^{ta,i}(x_N,\theta^i)]^T, \tag{3}$$

where $p_t^{ta,i}$ represents the prediction yielded by the temporal accumulator, which is defined in the following Eq. (4).

$$p_t^{ta,i}(x_k,\theta^i) = \beta\cdot p_{t-1}^{ta,i}(x_k,\theta^i) + (1-\beta)\cdot p_t^i(x_k,\theta^i), \tag{4}$$

where t is the index of the current training epoch, $p_{t-1}^{ta,i}$ and $p_t^{ta,i}$ are the accumulated prediction results of θ^i at the previous epoch $t-1$ and current epoch t, respectively, and p_t^i is the prediction results at the current epoch t. The range of β is $(0, 1)$, which can be viewed as a momentum term that reflects how many history data are involved. By doing so, the temporal accumulator smooths out stochastic noisy predictions resulted from the dynamic training procedure and produces robust outputs that can represent the long-term trend of predictions. We use a toy example as shown in the right

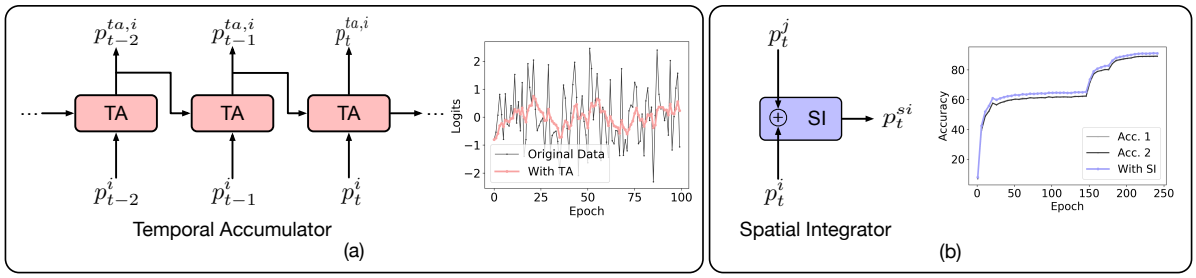


Figure 2. Illustration of (a) temporal accumulator and (b) spatial integrator. The right panel of (a) shows a toy example of applying temporal accumulator on raw noisy data. The right panel of (b) shows spatial integrator can consistently improve the performance during training.

panel of Fig. 2(a) to illustrate the effectiveness of temporal accumulation, where the black and the pink curves depict the raw data and the smoothed data after temporal accumulation, respectively. It is observed that the output of the temporal accumulator preserves the trends of the raw data as well as smooths out the oscillations.

In addition, we also use Eq. (5) to correct the bias introduced by initializing Z^i with zeros.

$$Z^i = Z^i/(1-\beta^t), \tag{5}$$

where t is the index of the current training epoch and β^t is β to the t-th power. Besides, we use a warm-up function $w(t)$ to reduce the weight of temporal accumulation loss at the very early stage of training. Hence,

$$\mathcal{L}^i_{ta} = w_i(t) \cdot \mathcal{L}_{kl}(p^i_t, p^{ta,j}_t), i \neq j. \tag{6}$$

When generalized to multiple networks, $\mathcal{L}^i_{ta}$ can be extended by calculating the sum of pair-wise KL divergence between p^i_t and the output of other networks' temporal accumulators. Z^i has a constant overhead proportional to the number of training samples, so it can be fairly large when the dataset is large-scale. In each iteration, only a mini batch of its elements are accessed. Hence, Z^i can be stored in a large but infrequently accessed memory in implementation.

Spatial Integrator We further propose a second boosting module that utilizes the diversity of student peers at each iteration, which we refer to as spatial integrator. Here we use "spatial" to highlight the relationship of the student peers. As shown in the left panel of Fig. 2(b), it takes the predictions produced by all student networks as the input and generates an integrated prediction as a superior soft target as in Eq. (7),

$$p^{si}_t = g(p^i_t, p^j_t), \tag{7}$$

where $g(\cdot)$ is the integration operation, and p^{si}_t is the integrated output of the spatial integrator. Here, we define the operation of the spatial integrator as averaging, i.e., $p^{si}_t = \frac{1}{2} \cdot (p^i_t + p^j_t)$. It has been proved that averaging the predictions of the large ensemble of different converged models can outperform the individual models [12]. In this work, we claim that averaging the predictions of the student peers during training from scratch provides a superior soft target for online KD. A straightforward example is shown in the right panel of Fig. 2(b), in which two ResNet-32 networks are trained on CIFAR-100 in an online KD manner. The curves labeled with "Acc. 1", "Acc. 2", and "With SI" are the test accuracy (%) of two individual networks (with vanilla online KD) and with the proposed spatial integrator evaluated during the training process, respectively. Our empirical study shows the spatial integrator can consistently outperform the individual networks during the entire training procedure. When the method is generalized to multiple networks, we exert $g(\cdot)$ on the predictions of all student networks to obtain the integrated prediction. The spatial integration loss for each network is represented in Eq. (8),

$$\mathcal{L}^i_{si} = \mathcal{L}_{kl}(p^i_t, p^{si}_t). \tag{8}$$

In summary, the overall loss function for training network θ^i is shown in Eq. (9).

$$\mathcal{L}^i = \mathcal{L}_{ce}(p^i, y) + \lambda_{ta} \cdot \mathcal{L}^i_{ta} + \lambda_{si} \cdot \mathcal{L}^i_{si}, \tag{9}$$

where λ_{ta} and λ_{si} are the scaling factors. The pseudocode is summarized in Algorithm 1.

Algorithm 1: Online Knowledge Distillation by Temporal-Spatial Boosting

Input: Training data $\mathcal{D}$; Training epochs T; Warm-up function $w(t)$; the number of training samples N; the number of classes C; the number of student networks M

Output: Trained models

Initialize $\{\theta^i\}_{i=1}^M$; $t = 0$; Temporal Accumulator $\{Z^i_0 \leftarrow 0_{[N\times C]}\}_{i=1}^M$

while $t < T$ **do**

- Compute the predictions of the networks, p^i_t, by Eq. (1), respectively;
- Update Temporal Accumulators Z^i_t by Eqs. (3), (4), and (5);
- Compute the spatially integrated predictions p^{si}_t from p^i_t by Eq. (7);
- Compute the total loss by Eqs. (6), (8), and (9);
- Update the model parameters $\{\theta^i\}_{i=1}^M$;

end

CIFAR-10													
Networks	Ind	DML [57]			KDCL [9]			ONE [22]			TSB (Ours)		
	Net	Net 1	Net 2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens
ResNet-20	91.89	92.06	92.02	92.63	92.78	92.64	93.18	92.54	92.72	93.21	93.30	93.32	94.01
ResNet-32	92.77	92.90	93.30	93.68	93.75	93.52	94.47	93.68	93.90	94.19	93.90	94.04	94.73
WRN-16-2	93.21	93.69	93.70	94.19	94.10	93.82	94.30	93.97	94.26	94.28	94.73	94.54	95.07
WRN-16-4	94.57	95.04	94.88	95.35	95.05	95.15	95.45	95.37	95.53	95.61	95.81	95.82	96.18
WRN-28-2	94.34	94.79	94.71	95.16	95.08	94.98	95.53	94.50	95.19	95.24	95.46	95.25	95.87
MobileNet	87.53	87.91	87.93	88.50	89.76	90.02	91.04	89.20	89.18	89.20	90.31	90.39	91.40
VGG-13	90.58	90.52	90.70	91.13	93.47	93.50	93.95	92.86	92.79	92.83	94.66	94.46	95.16

CIFAR-100													
Networks	Ind	DML [57]			KDCL [9]			ONE [22]			TSB (Ours)		
	Net	Net 1	Net2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens
ResNet-20	67.48	68.84	69.00	70.71	70.68	70.26	72.58	70.33	70.71	72.21	71.72	71.36	73.40
ResNet-32	68.99	71.20	71.15	72.98	72.61	72.77	74.42	72.46	72.93	74.41	74.01	73.67	76.15
WRN-16-2	73.26	72.77	73.11	74.54	75.49	74.81	76.87	72.75	72.52	74.63	75.96	75.65	77.51
WRN-16-4	75.38	76.11	76.16	77.66	79.08	78.84	80.51	77.58	77.25	78.40	79.81	79.32	81.08
WRN-28-2	73.50	75.23	75.50	77.23	77.34	77.17	79.52	76.81	76.48	78.42	77.89	77.71	80.24
MobileNet	64.60	63.45	63.48	65.28	66.37	65.84	68.47	63.34	63.30	63.27	66.88	67.35	69.36
VGG-13	74.64	73.96	73.28	74.99	73.83	74.08	75.34	75.11	75.88	76.11	77.19	77.07	78.53

Table 1. Top-1 test accuracy (%) comparison with state-of-the-art online KD methods on CIFAR-10/100 using two networks with the same architectures. Ind represents the baseline when the network is trained independently with cross-entropy loss. For ONE, Net 1 and Net 2 represent two different branches. Ens represents the performance of the ensemble of Net 1 and Net 2 by averaging their predictions.

4. Experiments and Results

4.1. Setup

In this section, we evaluation our proposed approach with a number of benchmark network structures, including ResNet [11], Wide ResNet (WRN) [55], MobileNetV2 [13], and VGG [39], on various widely-used image classification datasets, including CIFAR-10/100 [18] and ImageNet [6].

For the CIFAR-10/100 experiments, we train each network for 240 epochs with a batch size of 64. For MobileNetV2, the width multiplier is set to 0.5. The initial learning rate is set to 0.01 for MobileNetV2 and 0.05 for other structures, respectively. The learning rate is decayed by 0.1 at the 150th, 180th, and 210th epoch, respectively. All the models are trained with an SGD optimizer with a momentum of 0.9 and weight decay of 0.0005. The temperature for softening the softmax output is set to 4. With a thorough parameter search, we observe that the best performance is obtained when $\beta = 0.8$ in Eqs. (4) and (5). With empirical study, we find that applying a smaller weight to the KL divergence losses helps to improve the performance. Therefore, we set both λ_{ta} and λ_{si} to 0.5 for simplicity. We warm up the training process by zero-outing the KL divergence loss for the first 20 epochs. For the ImageNet experiments, we mainly follow the training procedure in [9]. Specifically, we use a batch size of 256. For ResNet, we train the models for 200 epochs, with the learning rate starting from 0.1, and decayed by 0.1 at the 60th, 120th, and 180th epoch, respectively. The weight decay factor is set to 0.0001. For MobileNetV2, the width multiplier is set to 1.0. MobileNetV2 networks are trained for 300 epochs and the learning rate is decayed by 0.1 at the 90th, 180th, and 270th epoch, respectively. During the test stage, samples are resized to 256×256 and then center-cropped to 224×224. All the other hyperparameters remain the same as those in the CIFAR-10/100 experiments.

We compare the performance of the proposed TSB with several state-of-the-art online KD approaches, including DML [57], ONE [22], and KDCL [9]. Besides, we adapt several popular classic KD methods into their online versions. To achieve a fair comparison, all these approaches are re-implemented using the above training configuration. In each experiment, we use *Ind* to represent baseline when the network is trained independently with cross-entropy loss. In addition to the performance of each student network, we report the performance of the ensemble of all student networks by averaging their predictions, denoted with *Ens*. Besides, *Avg* is used to represent the average performance of all student networks.

4.2. Main results

4.2.1 Results on CIFAR-10/100

We mainly evaluate our approach on CIFAR-10/100 with two student networks mutually trained from scratch. The performance with more student networks will be presented in Section 4.3.3. Firstly, we compare the performance using two networks with the same architectures, including various

CIFAR-10												
Networks		Ind		DML [57]			KDCL [9]			TSB (Ours)		
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens
ResNet-20	ResNet-32	91.89	92.77	92.16	92.63	93.21	93.07	93.53	93.84	93.32	94.01	94.41
ResNet-32	WRN-16-4	92.77	94.57	93.22	94.55	94.51	93.72	95.08	94.88	94.30	95.71	95.58
WRN-16-2	WRN-16-4	93.21	94.57	93.87	94.59	94.52	94.27	95.18	95.08	94.82	95.72	95.60
WRN-16-4	WRN-28-2	94.57	94.34	94.93	94.50	95.08	95.17	94.78	95.30	95.58	95.36	96.01
MobileNetV2	WRN-16-4	87.53	94.57	88.85	93.05	91.90	90.11	94.39	93.38	90.52	95.17	94.20
VGG-13	WRN-16-2	93.58	93.21	92.88	93.06	93.66	93.71	93.41	94.33	94.74	94.39	95.36
CIFAR-100												
Networks		Ind		DML [57]			KDCL [9]			TSB (Ours)		
Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Ens	Net 1	Net 2	Ens	Net 1	Net 2	Ens
ResNet-20	ResNet-32	67.48	68.99	68.31	70.28	71.26	69.90	72.99	73.97	70.98	73.08	74.39
ResNet-32	WRN-16-4	68.99	75.38	71.90	76.61	75.93	73.28	78.70	78.79	74.12	79.32	79.45
WRN-16-2	WRN-16-4	73.26	75.38	72.43	76.26	75.85	74.36	78.63	78.02	75.70	79.41	79.44
WRN-16-4	WRN-28-2	75.38	73.50	76.94	75.33	77.96	78.64	76.76	79.98	79.23	77.93	80.64
MobileNetV2	WRN-16-4	64.60	75.38	63.84	72.87	71.37	67.33	77.50	75.92	68.12	78.77	77.55
VGG-13	WRN-16-2	74.64	73.26	71.90	71.30	74.24	76.42	74.11	74.61	77.66	74.82	78.75

Table 2. Top-1 test accuracy (%) comparison with state-of-the-art online KD methods on CIFAR-10/100 using two networks with different architectures. Ind and Ens have the same meaning as in Table 1.

ResNets, WRNs, MobileNetV2, and VGG. We continue the comparison with different architectures, with a number of representative combinations of the above architectures. The results are listed in Tables 1 and 2, respectively.

Same Architecture In this scenario, we observe that the proposed TSB consistently outperforms the state-of-the-art methods to different degrees. For instance, for WRN-16-2, our method achieves an accuracy of 94.73% while the best state-of-the-art (ONE in this case) achieves an accuracy of 94.26%. We also report the performance of the ensemble of two networks as described in Section 4.1. With the ensemble, our accuracy increases to 95.07% while the best performance achieved by the previous study (KDCL) is 94.30%. In addition to the improved performance, we also observe that in most cases the two networks trained with TSB exhibit less difference in terms of test accuracy. For example, with ResNet-20, the test accuracies of the two networks are 93.30% and 93.32%, respectively, with only a 0.02% difference, while the differences with DML, KDCL, and ONE are 0.04%, 0.14%, and 0.18%, respectively. This phenomenon indicates the effectiveness of the temporal accumulator and spatial integrator in stabilizing the training process.

Different Architectures We also conduct experiments in which the two networks have different architectures. Since ONE only works for the same architectures due to its unique configuration, we exclude it in the comparison in this scenario. It is observed that TSB consistently achieves superior performance with different structure combinations. It is worth noting that the improvement of the state-of-the-art methods over the baselines is quite limited in some cases. For example, when using MobileNetV2 and WRN-16-4 on CIFAR-100, the performance of DML is even worse than the baseline. A similar phenomenon is also observed when training MobileNetV2 and WRN-16-4 with KDCL on CIFAR-10. It is probably caused by the significant difference in their architecture. It is worth noting that when the capacity difference of the two networks is too large, for example, using MobileNetV2 and WRN-16-4, the ensemble of these two models has a good chance of resulting in worse performance. In the case of both the individual networks and the ensemble, TSB consistently generates competitive performance compared to the state-of-the-art.

4.2.2 Results on ImageNet

We further evaluate the performance of our proposed approach on the large-scale dataset (ImageNet) with ResNet-18 and MobileNetV2. Two networks with the same structure are trained with various state-of-the-art online KD methods. We report the average performance of two networks in Table 3. Our TSB achieves competitive performance compared with state-of-the-art on both network structures. Compared to the baseline where the network is trained independently with cross-entropy loss, the performance with our method increases by 1.5% and 1.0%, respectively, while KDCL only outperforms the baseline by 0.9% and 0.6%, respectively. These results verify the effectiveness of the proposed TSB on large-scale image datasets.

4.2.3 More comparisons with other approaches

Since there have been only a few studies on online KD, to further expand the scope of performance comparison, we adopt a number of popular classic KD approaches into their

Network Types	Ind	DML [57]	ONE [22]	KDCL [9]	TSB (Ours)
ResNet-18	69.8	70.3	70.6	70.7	71.3
MobileNetV2	71.9	72.2	72.4	72.5	72.9

Table 3. Top-1 test accuracy (%) comparison with state-of-the-art online KD methods on ImageNet. Reported are the average performance.

Networks	Approach	Net 1	Net 2	Ens
ResNet-32	FitNet [36]	72.86	72.90	73.58
	AT [54]	72.97	73.00	75.04
	CRD [42]	72.53	72.78	74.46
	NST [14]	72.38	72.29	72.71
	SP [44]	72.71	72.76	73.24
	RKD [32]	72.54	72.61	72.72
	PKT [33]	73.22	73.43	73.97
	Ours	74.01	73.67	76.15
WRN-16-2	FitNet	74.73	74.82	74.83
	AT	74.49	74.28	74.66
	CRD	74.04	73.94	74.10
	NST	74.48	74.49	74.62
	SP	74.66	74.54	75.01
	RKD	73.86	73.96	73.99
	PKT	74.67	74.76	75.11
	Ours	75.29	75.59	77.23

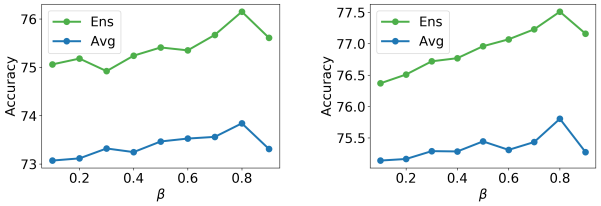
Table 4. Top-1 test accuracy (%) comparison with more online KD methods modified from their classic KD versions on CIFAR-100.

online versions, built upon the scheme of DML. Hence, DML is considered the baseline in this experiment. It is worth noting that all of these methods transfer feature-level information in addition to logits/predictions. Experimental results with ResNet-32 and WRN-16-2 on CIFAR-100 are shown in Table 4. It is observed that these adapted approaches outperform the baseline DML approximately by 1% to 2% in test accuracy. This is reasonable because these methods introduce extra alignments on their feature maps in addition to minimizing the KL divergence of the networks' predictions. It is observed that TSB outperforms these approaches with a clear margin, although our approach only transfers knowledge at the level of logits. These results further confirm the effectiveness of TSB.

4.3. Ablation studies and analyses

4.3.1 Effect of different components in TSB

TSB consists of two main modules, a temporal accumulator (TA) and a spatial integrator (SI), built upon vanilla online KD (DML). In this ablation study, we evaluate the effectiveness of each proposed component by comparing the performance of the following configurations: (1) DML, (2) DML+TA, (3) DML+SI, and (4) DML+TA&SI (TSB). We use three network combinations (two with the same architectures and one with different architectures) on both CIFAR-10 and CIFAR-100. The results are shown in Table 5. It is observed that when introducing either TA or SI to the vanilla online KD, the performance increases considerably in all six experiments. In addition, when using both TA and SI, the performance is further improved. These results validate the effectiveness of both components proposed for the online KD training procedure.



(a) ResNet-32 (b) WRN-16-2

Figure 3. Test accuracy (%) comparison using different β values with ResNet-32 and WRN-16-2 on CIFAR-100.

4.3.2 Effect of β

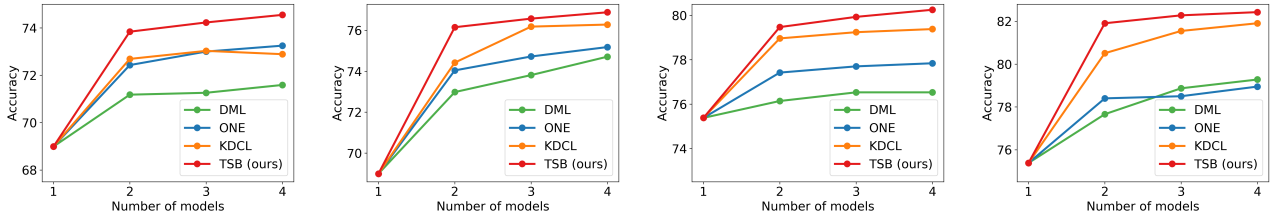
As mentioned in Section 3, β can be considered as a momentum term that controls how many previous predictions are involved in the next round. The exponentially weighted averages in Eq. (4) is approximately equivalent to averaging over $1/(1-\beta)$ epochs. Here we evaluate how different amounts of previous data involved to produce the temporally-accumulated predictions will influence the performance. We consider ResNet-32 and WRN-16-2 on CIFAR-100. β varies from 0.1 to 0.9 with a step of 0.1. From the results in Fig. 3, it can be observed that when $\beta = 0.8$, the best performance is obtained for both architectures. That is, approximately averaging over $1/(1-0.8) = 5$ epochs achieves the optimal performance. As β increases or decreases, though the accuracy drops some, they are still competitive compared to other approaches, indicating the relative robustness of TSB against β selection. Intuitively, larger β indicates that more previous data are involved in the temporal accumulation and they play a more important role in generating the soft targets than the current predictions. An appropriate value of β (0.8 in our cases) is needed to balance the weights of previous and current predictions.

4.3.3 Performance with multiple networks

In this experiment, we study the proposed online KD method with multiple (more than 2) student networks. We employ ResNet-32 and WRN-16-4 on CIFAR-100 and vary the number of student networks from 1 to 4. One student network means independent learning. The results are shown in Fig. 4. Both average performance (*Avg*) and performance of the ensemble (*Ens*) are reported. As expected, for all

Datasets	Networks		DML [57]		DML+TA		DML+SI		TSB (DML+TA+SI)	
	Net 1	Net 2	Avg	Ens	Avg	Ens	Avg	Ens	Avg	Ens
CIFAR-10	ResNet-32	ResNet-32	93.10	93.68	93.58	94.22	93.57	94.23	93.97	94.73
	WRN-16-4	WRN-16-4	94.96	95.35	95.62	96.10	95.50	95.98	95.82	96.18
	ResNet-32	WRN-16-4	94.23	94.52	94.59	95.20	94.85	95.42	95.01	95.58
CIFAR-100	ResNet-32	ResNet-32	71.18	72.98	73.47	75.07	73.20	74.60	73.84	76.15
	WRN-16-4	WRN-16-4	76.14	77.66	79.03	80.54	79.51	80.31	79.57	81.08
	ResNet-32	WRN-16-4	74.26	75.93	76.27	79.31	76.11	78.68	76.72	79.45

Table 5. Top-1 test accuracy (%) comparison on CIFAR-10 and CIFAR-100 using different components of our proposed TSB.



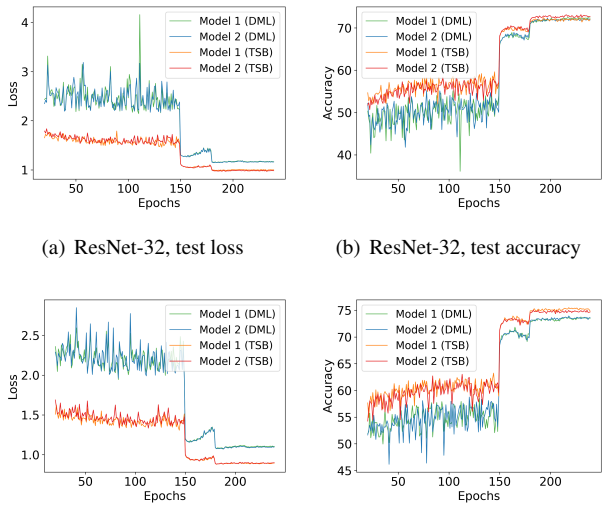
(a) ResNet-32, Avg (b) ResNet-32, Ens (c) WRN-16-4, Avg (d) WRN-16-4, Ens

Figure 4. Test accuracy (%) comparison with different number of student networks trained on CIFAR-100 in the online manner.

four online KD approaches, both *avg* and *ens* increase as the number of student networks increases since more students provide more diversity. It is observed that our proposed TSB approach achieves superior performance in terms of test accuracy in all scenarios with the number of student networks varying from 2 to 4. It is also worth mentioning that, there is a tendency that with more students involved, the growth rate of performance slows down, indicating the saturation of performance improvement.

4.3.4 Stabilizing the training procedure

One of our main motivations for introducing TA and SI is to provide superior instructors that can stabilize the online training process. To better understand the effect of these components, we visualize the test loss and accuracy of our method and DML over the training epochs, with ResNet-32 and WRN-16-4 on CIFAR-100. As shown in Fig. 5, for both networks, the test loss and accuracy curves of TSB are smoother, showing fewer spikes. However, the curves of DML significantly oscillate, especially in the first 150 epochs. It is observed that approximately at the 110th epoch in the ResNet-32 experiment, the test loss has a significant spike, suddenly increasing from 2.5 to 4.1. Consequently, a big drop occurs in the test accuracy curve. Similar phenomena are also observed in the WRN-16-4 experiments, where spikes occur approximately at the 45th, 80th, and 95th epochs. These results validate the effectiveness of the proposed method in stabilizing the training procedure.



(a) ResNet-32, test loss (b) ResNet-32, test accuracy

(c) WRN-16-4, test loss (d) WRN-16-4, test accuracy

Figure 5. Comparison of the proposed TSB and the baseline DML in terms of test loss and test accuracy (%) during the training process with ResNet-32 and WRN-16-2 on CIFAR-100.

5. Conclusion

In this paper, we presented a novel method for online knowledge distillation, which can stabilize the training process and improve performance. Two novel modules, temporal accumulator and spatial integrator, were proposed to provide superior and robust instructors for student networks during the training procedure. Extensive experiments, as well as comprehensive ablation studies, on various benchmark datasets and network structures, were conducted to validate the effectiveness of the proposed approach.

Acknowledgment

This work is in part supported by National Science Foundation, CNS 2038922.

References

[1] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006.

[2] Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*, 2019.

[3] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3435–3444, 2019.

[4] Eric M Christiansen, Samuel J Yang, D Michael Ando, Ashkan Javaherian, Gaia Skibinski, Scott Lipnick, Elliot Mount, Alison O'Neil, Kevan Shah, Alicia K Lee, et al. In silico labeling: predicting fluorescent labels in unlabeled images. *Cell*, 173(3):792–803, 2018.

[5] Inseop Chung, SeongUk Park, Jangho Kim, and Nojun Kwak. Feature-map-level online adversarial knowledge distillation. In *International Conference on Machine Learning*, pages 2006–2015. PMLR, 2020.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[7] Heitor Murilo Gomes, Jean Paul Barddal, Fabrício Enembreck, and Albert Bifet. A survey on ensemble learning for data stream classification. *ACM Computing Surveys (CSUR)*, 50(2):1–36, 2017.

[8] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[9] Qiushan Guo, Xinjiang Wang, Yichao Wu, Zhipeng Yu, Ding Liang, Xiaolin Hu, and Ping Luo. Online knowledge distillation via collaborative learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11020–11029, 2020.

[10] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.

[11] 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.

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

[13] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[14] Zehao Huang and Naiyan Wang. Like what you like: Knowledge distill via neuron selectivity transfer. *arXiv preprint arXiv:1707.01219*, 2017.

[15] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[16] Kumara Kahatapitiya and Ranga Rodrigo. Exploiting the redundancy in convolutional filters for parameter reduction. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1410–1420, 2021.

[17] Taisuke Kobayashi and Wendyam Eric Lionel Ilboudo. t-soft update of target network for deep reinforcement learning. *Neural Networks*, 136:63–71, 2021.

[18] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

[20] Salim Lahmiri, Stelios Bekiros, Anastasia Giakoumelou, and Frank Bezzina. Performance assessment of ensemble learning systems in financial data classification. *Intelligent Systems in Accounting, Finance and Management*, 27(1):3–9, 2020.

[21] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

[22] Xu Lan, Xiatian Zhu, and Shaogang Gong. Knowledge distillation by on-the-fly native ensemble. *arXiv preprint arXiv:1806.04606*, 2018.

[23] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[24] Chengcheng Li, Zi Wang, and Hairong Qi. Fast-converging conditional generative adversarial networks for image synthesis. In *2018 25th IEEE international conference on image processing (ICIP)*, pages 2132–2136. IEEE, 2018.

[25] Chengcheng Li, Zi Wang, and Hairong Qi. An efficient pipeline for pruning convolutional neural networks. In *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 907–912. IEEE, 2020.

[26] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[27] Yawei Li, Shuhang Gu, Luc Van Gool, and Radu Timofte. Learning filter basis for convolutional neural network compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5623–5632, 2019.

[28] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. *arXiv preprint arXiv:1509.02971*, 2015.

[29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[30] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski,

et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[31] Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*, 2016.

[32] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019.

[33] Nikolaos Passalis and Anastasios Tefas. Learning deep representations with probabilistic knowledge transfer. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 268–284, 2018.

[34] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[35] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.

[36] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. *arXiv preprint arXiv:1412.6550*, 2014.

[37] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander WR Nelson, Alex Bridgland, et al. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792):706–710, 2020.

[38] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[40] Hongjun Su, Yao Yu, Qian Du, and Peijun Du. Ensemble learning for hyperspectral image classification using tangent collaborative representation. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):3778–3790, 2020.

[41] Yuanchao Su, Xiang Xu, Jun Li, Hairong Qi, Paolo Gamba, and Antonio Plaza. Deep autoencoders with multitask learning for bilinear hyperspectral unmixing. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[42] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

[43] Dumindu Tissera, Kasun Vithanage, Rukshan Wijesinghe, Kumara Kahatapitiya, Subha Fernando, and Ranga Rodrigo. Feature-dependent cross-connections in multi-path neural networks. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4032–4039. IEEE, 2021.

[44] Frederick Tung and Greg Mori. Similarity-preserving knowledge distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1365–1374, 2019.

[45] Kuan Wang, Zhijian Liu, Yujun Lin, Ji Lin, and Song Han. Haq: Hardware-aware automated quantization with mixed precision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8612–8620, 2019.

[46] Zi Wang. Data-free knowledge distillation with soft targeted transfer set synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10245–10253, 2021.

[47] Zi Wang. Learning fast converging, effective conditional generative adversarial networks with a mirrored auxiliary classifier. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2566–2575, 2021.

[48] Zi Wang. Zero-shot knowledge distillation from a decision-based black-box model. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 10675–10685. PMLR, 18–24 Jul 2021.

[49] Zi Wang, Chengcheng Li, and Xiangyang Wang. Convolutional neural network pruning with structural redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14913–14922, 2021.

[50] Zi Wang, Chengcheng Li, Xiangyang Wang, and Dali Wang. Towards efficient convolutional neural networks through low-error filter saliency estimation. In *Pacific Rim International Conference on Artificial Intelligence*, pages 255–267. Springer, 2019.

[51] Zi Wang, Dali Wang, Chengcheng Li, Yichi Xu, Husheng Li, and Zhirong Bao. Deep reinforcement learning of cell movement in the early stage of c. elegans embryogenesis. *Bioinformatics*, 34(18):3169–3177, 2018.

[52] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

[53] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017.

[54] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

[55] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[56] Junfei Zhang, Yuhang Wang, Yuantian Sun, and Guichen Li. Strength of ensemble learning in multiclass classification of rockburst intensity. *International Journal for Numerical and Analytical Methods in Geomechanics*, 44(13):1833–1853, 2020.

[57] Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. Deep mutual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4320–4328, 2018.