AMLN: Adversarial-based Mutual Learning Network for Online Knowledge Distillation

Xiaobing Zhang[1][0000−0002−8149−1424], Shijian Lu[2][0000−0002−6766−2506], Haigang Gong[1][0000−0002−7176−4142], Zhipeng Luo[2,3][0000−0001−9323−7872], and Ming Liu[1][0000−0002−1114−1728]

[1] University of Electronic Science and Technology of China
zhangxiaobing@std.uestc.edu.cn, {hggong, csmliu}@uestc.edu.cn
[2] Nanyang Technological University, Singapore
shijian.lu@ntu.edu.sg, zhipeng001@e.ntu.edu.sg
[3] Sensetime Research, Singapore

Abstract. Online knowledge distillation has attracted increasing interest recently, which jointly learns teacher and student models or an ensemble of student models simultaneously and collaboratively. On the other hand, existing works focus more on outcome-driven learning according to knowledge like classification probabilities whereas the distilling processes which capture rich and useful intermediate features and information are largely neglected. In this work, we propose an innovative adversarial-based mutual learning network (AMLN) that introduces process-driven learning beyond outcome-driven learning for augmented online knowledge distillation. A block-wise training module is designed which guides the information flow and mutual learning among peer networks adversarially throughout different learning stages, and this spreads until the final network layer which captures more high-level information. AMLN has been evaluated under a variety of network architectures over three widely used benchmark datasets. Extensive experiments show that AMLN achieves superior performance consistently against state-of-the-art knowledge transfer methods.

Keywords: Mutual learning network, Adversarial-based learning strategy, Online knowledge transfer and distillation.

1 Introduction

Deep neural networks (DNNs) have been widely studied and applied in various fields such as image classification [7, 36], object detection [11, 26], semantic segmentation [10, 27], etc. One direction pursues the best accuracy which tends to introduce over-parameterized models [24, 26] and demands very high computation and storage resources that are often not available for many edge computing devices. This has triggered intensive research in developing lightweight yet competent network models in recent years, typically through four different approaches: 1) network pruning [14, 15, 17, 21, 28], 2) network quantization [9, 29],

3) building efficient small networks [6, 19, 20], and 4) knowledge transfer (KT) [4, 5, 12, 18, 30]. Among the four approaches, KT works in a unique way by pre-training a large and powerful teacher network and then distilling features and knowledge to a compact student network. Though compact yet powerful student networks can be trained in this manner, the conventional distillation is usually a multi-stage complex offline process requiring extra computational costs and memory.

Online knowledge distillation [3,22,25,37] has attracted increasing interest in recent years. Instead of pre-training a large teacher network in advance, it trains two or more student models simultaneously in a cooperative peer-teaching manner. In other words, the training of the teacher and student networks is merged into a one-phase process, and the knowledge is distilled and shared among peer networks. This online distilling paradigm can generalize better without a clear definition of teacher/student role, and it has achieved superior performance as compared to offline distillation from teacher to student networks. On the other hand, this online distillation adopts an outcome-driven distillation strategy in common which focuses on minimizing the discrepancy among the final predictions. The rich information encoded in the intermediate layers from peer networks is instead largely neglected which has led to various problems such as limited knowledge transfer in deep mutual learning [37], constrained coordination in on-the-fly native ensemble [25], etc.

In this work, we propose a novel adversarial-based mutual learning network (AMLN) that includes both process-driven and outcome-driven learning for optimal online knowledge distillation. Specifically, AMLN introduces a block-wise learning module for process-driven distillation that guides peer networks to learn the intermediate features and knowledge from each other in an adversarial manner as shown in Fig. 1. At the same time, the block-wise module also learns from the final layer of the peer networks which often encodes very useful high-level features and information. In addition, the softened class posterior of each network is aligned with the class probabilities of its peer, which works together with a conventional supervised loss under the outcome-driven distillation. By incorporating supervision from both intermediate and final network layers, AMLN can be trained in an elegant manner and the trained student models also produce better performance than models trained from scratch in a conventional supervised learning setup. Further, AMLN outperforms state-of-the-art online or offline distillation methods consistently. More details will be described in Experiments and Analysis sections.

The contributions of this work are thus threefold. First, it designs an innovative adversarial-based mutual learning network AMLN that allows an ensemble of peer student networks to transfer knowledge and learn from each other collaboratively. Second, it introduces a block-wise module to guide the peer networks to learn intermediate features and knowledge from each other which augments the sole outcome-driven peer learning greatly. Third, AMLN does not require pre-training a large teach network, and extensive experiments over several public

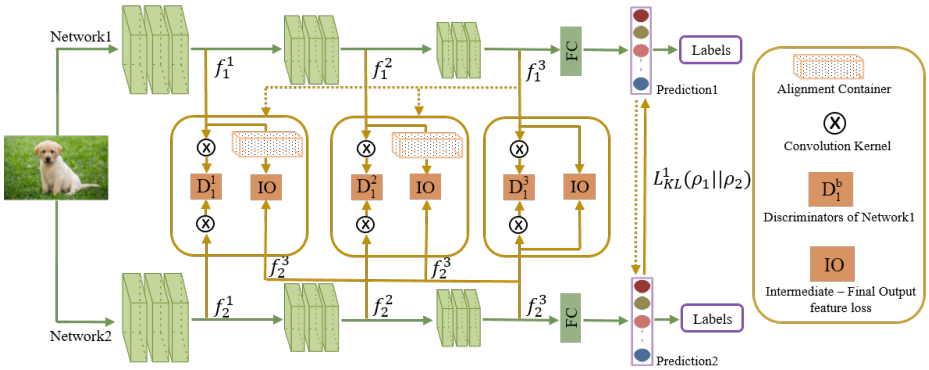

Fig. 1: Overview of the proposed adversarial-based mutual learning network (AMLN): AMLN achieves process-driven mutual distillation by dividing each peer network into same blocks and employing a discriminator to align the block-wise learned features adversarially. Additionally, the intermediate features are also guided by the peer's final output for learning high-level features. The outcome-driven learning instead employs the conventional cross-entropy loss (with one-hot labels) and Kullback-Leibler (KL) loss (with softened labels). Note this pipeline focuses on the distillation from Network2 to Network1. For distillation from Network1 to Network2, a similar pipeline applies as highlighted by the dashed lines.

datasets show that it achieves superior performance as compared to state-of-the-art online/offline knowledge transfer methods.

2 Related work

2.1 Knowledge Transfer

Knowledge transfer (KT) is one of the most popular methods used in model compression. The early KT research follows a teacher-student learning paradigm in an offline learning manner [5, 12, 23, 30, 34]. In recent years, online KT is developed to strengthen the student's performance without a pre-trained teacher network [3, 25, 33, 37]. Our work falls into the online KT learning category.

Offline KT aims to enforce the efficiency of the student's learning from scratch by distilling knowledge from a pre-trained powerful teacher network. Cristian *et.al* [5] first uses soft-labels for knowledge distillation, and this idea is further improved by adjusting the temperature of softmax activation function to provide additional supervision and regularization on the higher entropy soft-targets [12]. Recently, various new KT systems have been developed to enhance the model capabilities by transferring intermediate features [23, 30, 34] or by optimizing the initial weights of student networks [8, 18].

Online KT trains a student model without the requirement of training a teacher network in advance. With online KT, the networks teach each other

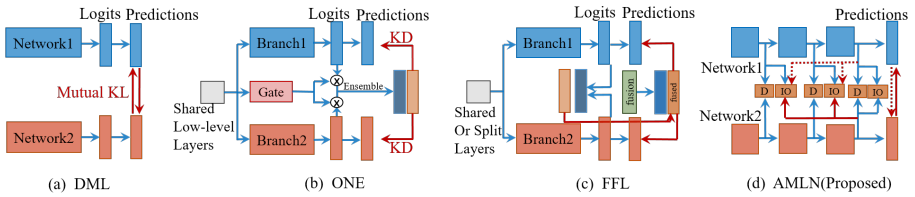

Fig. 2: Four different mutual learning networks: The architectures in (a), (b) and (c) perform mutual learning from the predictions or features of peer networks. The deep mutual learning (DML) [37] in (a) uses the distilled softened prediction of the peer network. The on-the-fly native ensemble (ONE) [25] in (b) creates a teacher with the gating mechanism for the peer network training. The feature fusion learning (FFL) [22] in (c) applies mutual knowledge learning between peer networks and fused classifier. Unlike these outcome-driven learning architectures, our adversarial-based mutual learning network (AMLN) in (d) uses mutual knowledge distillation between block-wise output features and final generated predictions, which enhances the performance of each peer network by distilling more multifarious features from peers.

mutually by sharing their distilled knowledge and imitating the peer network's performance during the training process. Deep mutual learning (DML) [37] and on-the-fly native ensemble (ONE) [25] are the two representative online KT methods that have demonstrated very promising performance as illustrated in Fig. 2. DML proposes to train the students by mutually exchanging the softened classification information using the Kullback-Leibler(KL) divergence loss. Similar to [37], Rohan *et.al* [3] introduces the codistillation method that forces student networks to maintain diversity longer by using the distillation loss after enough *burn in steps*. Rather than mutually distilling between peer networks, ONE generates a gated ensemble logit of the networks during training and adopts it as a target to guide each network. In addition, feature fusion learning (FFL) [22] uses a fusion module to combine the feature maps from sub-networks, aiming for enhancing the performance of each sub-network.

All the above methods adopt an outcome-driven distillation approach where the distillation during the intermediate network layers is largely neglected. AMLN addresses this issue by further incorporating process-driven distillation which guides the sharing and transfer of intermediate knowledge beyond the knowledge from the final outputs. Unlike ONE [25], AMLN also has better applicability which can work with peer networks with the same or different architecture.

2.2 Adversarial Learning

Generative Adversarial Learning [13] is proposed to create realistic-looking images from random noise. An adversarial training scheme is proposed which consists of a generator network G and a discriminator network D. Specifically, G

learns to synthesize images to fool D, meanwhile, D is trained to distinguish the real images in the dataset from the fake images generated by G.

To align the intermediate features which are updated continually at each training iteration, the L_1 or L_2 distance is not applicable since it is designed to evaluate the pixel-level or point-level difference instead of distributional differences between features. We introduce adversarial learning for online mutual learning among multiple student networks, where each student tends to generate features with similar distributions as its peer by striving to deceive the discriminators while the discriminators are trained to distinguish the different distributions of the generated features from multiple peer student networks.

3 Proposed Method

In this section, we describe how to effectively guide the peer-teaching student networks to learn collaboratively with the proposed Adversarial-based Mutual Learning Network (AMLN). Unlike existing online KT methods, AMLN takes into account not only the distillation based on the final prediction, but also the intermediate mutual supervision between the peer networks. We start by giving the architecture overview in subsection 3.1, and introduce our novel online process-driven mutual knowledge distillation in subsection 3.2. In subsection 3.3, we give an explanation of the outcome-driven mutual learning method. Finally, the whole optimization pipeline is presented in subsection 3.4.

3.1 The Architecture of AMLN

We formulate our proposed method by considering two peer networks S_1 and S_2. As illustrated in Fig. 1, S_1 and S_2 could adopt identical of different architectures, but should have the same number of blocks for intermediate feature alignment. During the training, the process-driven mutual knowledge distillation is implemented with a proposed block-wise module that contains a discriminator and an alignment container. Specifically, each network is trained to fool its corresponding block-wise discriminators so that it can produce similar feature maps to mimic that from its peer network. The alignment container is employed to align the block-wise outputs to the peer network's final feature maps for high-level information distillation. On the other hand, the outcome-driven mutual knowledge distillation is realised by minimizing the peer model's softened output distributions, which encodes higher entropy as extra supervision. Moreover, ground truth labels are used as a conventional supervision for the task-specific features learning.

3.2 Process-driven mutual knowledge distillation

Given N samples $X = \{x_i\}_{i=1}^{N}$ from M classes, we denote the corresponding label set as $Y = \{y_i\}_{i=1}^{N}$ with $y_i \in \{1, 2, ..., M\}$. As can be seen in Fig. 1, the backbone networks are first divided into the same blocks according to their depth. Suppose

that the block-wise generated feature is defined as f_j^b, where j and b indicate the network number and block number respectively, i.e. $j = 1, 2$ and $b = 1, 2, 3$. Each block is followed with a block-wise training module, including a discriminator D_j^b and an alignment container C_j^b. The discriminator D_j^b is formed by three convolution layers with ReLU operation, where the last layer with two neurons is responsible for identifying the network number j of the injected feature f_j^b. For each alignment container C_j^b, it applies depthwise convolution and pointwise convolution to align the block-wise generated feature f_j^b with the peer's final output f_{3-j}^3 for high-level knowledge distillation. Therefore, there are two loss items for the process-driven mutual learning, one of which is the adversarial-based distilling loss defined as follows:

$$L_D^j = \min_{f_j^b} \max_D \sum_{b=1}^{3} E_{f_j^b \sim P_{S_j}}[1 - D_j^b(\sigma(f_j^b))] + E_{f_{3-j}^b \sim P_{S_{3-j}}}[D_j^b(\sigma(f_{3-j}^b))] \quad (1)$$

Here, σ denotes the convolution kernel, which is utilized to reduce the number of channels of f_j^b. P_{S_j} corresponds to the logits distribution of the network S_j.

Another loss works by evaluating the distance between the block-wise distilled feature and the peer's final generated feature, which can be computed as:

$$L_F^j = \sum_{b=1}^{3} d(C_j^b(f_j^b), f_{3-j}^3) \quad (2)$$

where C_j^b denotes the alignment container that transforms f_j^b into the same shape as f_{3-j}^3, and the distance metric d is adopted with L_2 method consistently.

The overall process-driven mutual distillation loss function is then formulated with the weight balance parameter β as:

$$L_{S_j^P} = L_D^j + \beta L_F^j \quad (3)$$

3.3 Outcome-driven mutual knowledge distillation

For outcome-driven distillation, two evaluation items are employed where one is the conventional cross-entropy (CE) loss and the other is the Kullback Leibler (KL) loss between the softened predicted outputs. Suppose that the probability of class m for sample x_i given by S_j is computed as:

$$p_j^m(x_i) = \frac{exp(z_j^m)}{\sum_{m=1}^{M} exp(z_j^m)} \quad (4)$$

where z_j^m is the predicted output of S_j. Thus, the CE loss between the predicted outputs and one-hot labels for S_j can be evaluated as:

$$L_C^j = -\sum_{i=1}^{N} \sum_{m=1}^{M} u(y_i, m) log(p_j^m(x_i)) \quad (5)$$

Algorithm 1 Adversarial-based Mutual Learning Network (AMLN)

Require:
Training set X, label set Y;
Ensure:
Iteration = 0; Initialize S_1 and S_2 to different conditions;
1: Compute intermediate feature maps f_1^b, predicted probabilities p_1 and softened predictions ρ_1, b=1,2,3;
2: Compute the total loss L_{S_1} (Equ. 9);
3: Update the parameters of network S_1 by the SGD algorithm;
4: Compute intermediate feature maps f_2^b, predicted probabilities p_2 and softened predictions ρ_2, b=1,2,3;
5: Compute the total loss L_{S_2} (Equ. 9);
6: Update the parameters of network S_2 by the SGD algorithm;
7: Iteration = Iteration + 1; Begin with Step 1.
8: **return** Both converged models S_1 and S_2.

Here, u is an indicator function, which returns 1 if $y_i = m$ and 0 otherwise.

To improve the generalization performance of sub-networks on the test data, we apply the peer network to generate softened probability with a temperature term T. Given z_j, the softened probability is defined as:

$$\rho_j^m(x_i, T) = \frac{exp(z_j^m/T)}{\sum_{m=1}^{M} exp(z_j^m/T)} \tag{6}$$

when $T = 1$, ρ_j^m is the same as p_j^m. As the temperature term T increases, it generates a softened probability distribution where the probability of each class distributes more evenly and less dominantly. Same as [22, 37], we use $T = 3$ consistently during our experiments.

KL divergence is then used to quantify the alignment of the peer networks' softened predictions as:

$$L_{KL}^j(\rho_j||\rho_{3-j}) = \sum_{i=1}^{N}\sum_{m=1}^{M} \rho_j^m(x_i) log \frac{\rho_j^m(x_i)}{\rho_{3-j}^m(x_i)} \tag{7}$$

The overall outcome-driven distillation loss function $L_{S_j^R}$ is formulated as:

$$L_{S_j^R} = L_C^j + T^2 \times L_{KL}^j \tag{8}$$

Since the scale of the gradient produced by the softened distribution is $1/T^2$ of the original value, we multiply T^2 according to the KD recommendations [12] to ensure that the relative contributions of the ground-truth and the softened peer prediction remain roughly unchanged.

3.4 Optimization

Combining both process-driven and outcome-driven distillation loss, the overall loss for each sub-network S_j is as follows:

$$L_{S_j} = L_{S_j^P} + L_{S_j^R} \tag{9}$$

The mutual learning strategy in AMLN works in such a way that the peer networks are closely guided and optimized jointly and collaboratively. At each training iteration, we compute the generated features and predictions of the two peer networks, and update both models' parameters according to Equ. 9. The optimization details are summarized in Algorithm 1.

4 Experimental Results and Analysis

4.1 Datasets and Evaluation Setups

AMLN is evaluated over three datasets that have been widely used for evaluations of knowledge transfer methods. **CIFAR10** [1] and **CIFAR100** [2] are two publicly accessible datasets that have been widely used for the image classification studies. The two datasets have 50,000 training images and 10,000 test images of 10 and 100 image classes, respectively. All images in the two datasets are in RGB format with an image size of 32×32 pixels. **ImageNet** [31] refers to the LSVRC 2015 classification dataset which consists of 1.2 million training images and 50,000 validation images of 1,000 image classes.

Evaluation Metrics: We use the Top-1 and Top-5 mean classification accuracy (%) for evaluations, the former is calculated for all studied datasets while the latter is used for the ImageNet only. To measure the computation cost in model inference stage, we apply the criterion of floating point operations (FLOPs) and the inference time of each image for efficiency comparison.

Networks: The evaluation networks in our experiments include ResNet [16] as well as Wide ResNet(WRN) [35] of different network depths. Table 1 shows the number of parameters of different AMLN-trained network models that are evaluated over the dataset CIFAR100.

4.2 Implementation Details

All experiments are implemented by PyTorch on NVIDIA GPU devices. On the CIFAR dataset, the initial learning rate is 0.1 and is multiplied by 0.1 every 200 epochs. We used SGD as the optimizer with Nesterov momentum 0.9 and weight decay 1e-4, respectively. Mini-batch size is set to 128. For ImageNet, we use SGD with a weight decay of 10^{-4}, a mini-batch size of 128, and an initial

Table 1: The number of parameters in Millions over CIFAR100 dataset.

Network Types	WRN-40-2	WRN-16-2	ResNet110	ResNet32
Parameters	2.27M	0.72M	1.74M	0.48M
Network Types	WRN-40-1	WRN-16-1	ResNet56	ResNet20
Parameters	0.57M	0.18M	0.86M	0.28M

Table 2: Comparison with online distillation methods DML [37], ONE [25] and FFL [22] over CIFAR10 in (a) and CIFAR100 in (b) with the same network architecture. '↑' denotes accuracy increases over 'vanilla', 'Avg denotes the average accuracy of Net1 and Net2, and '*' indicates the reported accuracies in [22] under the same network setup.

(a) Top-1 accuracy(%) with the same architecture networks on CIFAR10.

Network Types		vanilla	DML [37]		ONE [25]		FFL [22]		AMLN	
Net1	Net2		Avg	↑	Avg*	↑	Avg*	↑	Avg	↑
ResNet32	ResNet32	93.10	93.15	0.05	93.76	0.66	93.81	0.71	94.25	1.15
ResNet56	ResNet56	93.79	94.19	0.40	94.38	0.59	94.43	0.64	94.68	0.89
WRN-16-2	WRN-16-2	93.58	93.72	0.14	93.76	0.18	93.79	0.21	94.39	0.81
WRN-40-2	WRN-40-2	94.71	95.03	0.32	95.06	0.35	95.17	0.46	95.21	0.50

(b) Top-1 accuracy(%) with the same architecture networks on CIFAR100.

Network Types		vanilla	DML [37]		ONE [25]		FFL [22]		AMLN	
Net1	Net2		Avg	↑	Avg*	↑	Avg*	↑	Avg	↑
ResNet32	ResNet32	69.71	70.98	1.27	72.57	2.86	72.97	3.26	74.69	4.98
ResNet56	ResNet56	71.76	74.13	2.37	74.58	2.82	74.78	3.02	75.77	4.01
WRN-16-2	WRN-16-2	71.41	73.27	1.86	73.95	2.54	74.17	2.76	75.56	4.15
WRN-40-2	WRN-40-2	74.47	76.49	2.02	77.63	3.16	77.77	3.30	77.97	3.50

learning rate of 0.1. The learning rate is decayed every 30 epochs by a factor of 0.1 and we train for a total of 90 epochs.

4.3 Comparisons with the Online Methods

Comparisons over CIFAR: This section presents the comparison of AMLN with state-of-the-art mutual learning methods DML [37], ONE [25] and FFL [22] over CIFAR10 and CIFAR100. Since ONE cannot work for peer networks with different architectures, we evaluate both scenarios when peer networks have the same and different architectures. Tables 2 and Table 3 show experimental results, where 'vanilla' denotes the accuracy of backbone networks that are trained from scratch with classification loss alone, 'Avg' shows the averaged accuracy of the two peer networks Net 1 and Net 2, and the column highlighted with '*' represents the values as extracted from [22] under the same setup.

Case 1: Peer Networks with the Same Architecture Tables 2(a) and 2(b) show the Top-1 accuracy over the datasets CIFAR10 and CIFAR100, respectively, when peer networks have the same architecture. As Table 2 shows, ONE, DML, and FFL all outperform the 'vanilla' consistently though ONE and FFL achieve larger margins in performance improvement. In addition, AMLN outperforms all three state-of-the-art methods consistently under different network architectures and different datasets. Specifically, the average accuracy improvements (across the four groups of peer networks) over DML, ONE and FFL are up to 0.61%, 0.39% and 0.33% for CIFAR10 and 2.28%, 1.32% and

Table 3: Comparison with online distillation methods DML [37], ONE [25] and FFL [22] over CIFAR10 in (a) and CIFAR100 in (b) with different network architectures.

(a) Top-1 accuracy(%) with different architecture networks on CIFAR10.

Network Types		vanilla		DML		FFL		AMLN	
Net1	Net2	Net1	Net2	Net1	Net2	Net1	Net2	Net1	Net2
WRN-16-2	ResNet32	93.58	93.10	93.91	93.39	94.01	93.99	94.37	94.35
WRN-40-2	ResNet56	94.71	93.79	94.87	93.87	94.89	94.05	94.94	94.39

(b) Top-1 accuracy(%) with different architecture networks on CIFAR100.

Network Types		vanilla*		DML*		FFL*		AMLN	
Net1	Net2	Net1	Net2	Net1	Net2	Net1	Net2	Net1	Net2
WRN-16-2	ResNet32	71.41	69.71	73.55	71.69	74.07	72.94	75.88	74.65
WRN-40-2	ResNet56	74.47	71.76	76.67	73.25	76.94	73.77	76.76	75.29

1.08% for CIFAR100, respectively. Further, it can be observed that the performance improvement over the more challenging CIFAR100 is much larger than that over CIFER10, demonstrating the good scalability and generalizability of ALMN when applied to complex datasets with more image classes.

Case 2: Peer Networks with Different Architectures This experiment evaluates the peer networks with different architectures WRN-16-2/ResNet32 and WRN-40-2/ResNet56, where the former pair has relatively lower depths. Table 3 shows experimental results. As Table 3(a) shows, the AMLN-trained Net1 and Net2 outperform the same networks trained by 'DML' and 'FFL' consistently on CIFAR10. For CIFAR100, AMLN-trained Net2 achieves significant improvements of 1.71% (WRN-16-2/ResNet32) and 1.52% (WRN-40-2/ResNet56) over the state-of-the-art method FFL as shown in Table 3(b). The good performance is largely attributed to the complementary knowledge distillation with both process-driven learning and outcome-driven learning which empower the peer networks to learn and transfer more multifarious and meaningful features from each other.

Comparisons over ImageNet: To demonstrate the potential of AMLN to transfer more complex information, we conduct a large-scale experiment over the ImageNet LSVRC 2015 classification task. For a fair comparison, we choose the same peer networks of ResNet34 as in ONE [25] and FFL [22]. Table 4 shows experimental results. As Table 4 shows, ONE and FFL achieve similar performance as what is observed over the CIFAR datasets. Our AMLN method performs better consistently, with 1.09% and 1.06% improvements in the Top-1 accuracy as compared with ONE and FFL, respectively. The consistent strong performance over the large-scale dataset ImageNet further demonstrates the scalability of our proposed method.

Table 4: Comparison of Top-1/Top-5 accuracy(%) with online methods ONE [25] and FFL [22] on the ImageNet dataset with the same network architecture (ResNet34). #FLOPs and inference time of each image are also provided.

Method	Top-1(%)	Top-5(%)	#FLOPs	Inference time(per/image)
vanilla	73.31	91.42	3.67B	1.13×10^{-2}s
ONE	74.39	92.04	4.32B	1.13×10^{-2}s
FFL	74.42	92.05	4.35B	1.13×10^{-2}s
AMLN	75.48	92.54	3.72B	1.13×10^{-2}s

Table 5: Comparison results with offline knowledge transfer methods AT [34], KD [12], FT [23], as well as their hybrid methods AT+KD and FT+KD over CIFAR10 (a) and CIFAR100 (b). The results shown in the last 7 columns are from Table 3 of [23], where the 'vanilla' column represents the performance of the backbone network trained from scratch and the last five columns are the Top-1 accuracy of Net2 under the guidance of Net1.

(a) Comparison results of Top-1 accuracy(%) on CIFAR10.

Network Types		AMLN		vanilla		AT	KD	FT	AT+KD	FT+KD
Net1	Net2	Net1	Net2	Net1	Net2					
WRN-40-1	ResNet20	94.42	93.48	93.16	92.22	92.66	92.91	93.15	93.00	93.05
WRN-16-2	WRN-16-1	94.16	92.92	93.73	91.38	91.90	92.36	92.36	92.48	92.41

(b) Comparison results of Top-1 accuracy(%) on CIFAR100.

Network Types		AMLN		vanilla		AT	KD	FT	AT+KD	FT+KD
Net1	Net2	Net1	Net2	Net1	Net2					
ResNet110	ResNet20	76.12	72.44	73.09	68.76	68.96	66.86	70.92	65.22	67.81
ResNet110	ResNet56	76.74	74.79	73.09	71.06	72.72	72.04	74.38	71.99	73.07

4.4 Comparisons with the Offline Methods

Several experiments have been carried out to compare AMLN with state-of-the-art offline knowledge transfer methods including AT [34], KD [12], FT [23], as well as the combinations of AT+KD and FT+KD. Among all compared methods, KD adopts an outcome-driven learning strategy and AT and FT adopt process-driven learning strategy.

Tables 5(a) and 5(b) show experimental results over CIFAR10 and CIFAR100, respectively, where Net1 serves as the teacher to empower the student Net2. Three points can be observed from the experimental results: 1) AMLN-trained student Net2 outperforms that trained by all other offline distillation methods consistently for both CIFAR10 and CIFAR100, regardless of whether Net1 and Net2 are of different types (WRN-40-1/ResNet20), having different widths (WRN-16-2/WRN-16-1) or depths (ResNet110/ResNet20, ResNet110/ResNet56); 2) Compared to the 'vanilla' teacher Net1 trained from scratch, AMLN-trained teacher Net1 (mutually learnt with the student Net2) obtains significantly better performance with 0.43%-1.26% and 3.03%-3.65% improvements on CIFAR10

Table 6: Ablation study of AMLN with the same peer network ResNet32.

Case	Outcome-driven loss		Process-driven loss		CIFAR10	CIFAR100
	L_C	L_{KL}	L_D	L_F		
A	✓				93.10	69.71
B	✓	✓			93.15	70.98
C	✓	✓		✓	93.89	73.24
D	✓	✓	✓		94.01	74.16
E	✓	✓	✓	✓	94.25	74.69

and CIFAR100, respectively. This shows that small networks with fewer parameters or smaller depths can empower larger networks effectively through distilling useful features; and 3) AMLN-trained student Net2 even achieves higher accuracy than its corresponding teacher Net1 in 'vanilla'. Specifically, AMLN-trained ResNet56 (0.86M parameters) produces a better classification accuracy with an improvement of 1.70% than the teacher ResNet110 (1.74M parameters) trained from scratch (in the ResNet110/ResNet56 setup). This shows that a small network trained with proper knowledge distillation could have the same or even better representation capacity than a large network.

4.5 Ablation Study

In AMLN, we have moved one step forward from previous researches by introducing the block-wise module which consists of mutual adversarial learning (MDL) and intermediate-final feature learning (MFL). We perform ablation studies to demonstrate the effectiveness of the proposed method on the datasets CIFAR10 and CIFAT100 by using two identical peer networks ResNet32. Table 6 shows experimental results.

As Table 6 shows, Cases A and E refer to the models trained from scratch and from AMLN, respectively. Case B refers to the network when only the outcome-driven losses L_C (Equ. 5) and L_{KL} (Equ. 7) are included. By including MFL(L_F) in Case C, the averaged accuracy is improved by 0.74% and 2.26% over datasets CIFAR10 and CIFAR100, respectively, as compared with case B. The further inclusion of MDL(L_D) on top of the outcome-driven losses in Case D introduces significant improvements of 0.86% and 3.18% over the datasets CIFAR10 and CIFAR100, respectively. The improvements indicate that MDL has a greater impact on the model performance, which is largely attributed to the convolutional structure of the discriminator that can interpret the spatial information in block-wise intermediate features and map the peer model's features to a similar probability distribution. As expected, AMLN performs the best when both outcome-driven loss and process-driven loss are included for mutual learning. This demonstrates that the two learning strategies are actually complementary to each other in achieving better knowledge distillation and transfer between the collaboratively learning peer networks.

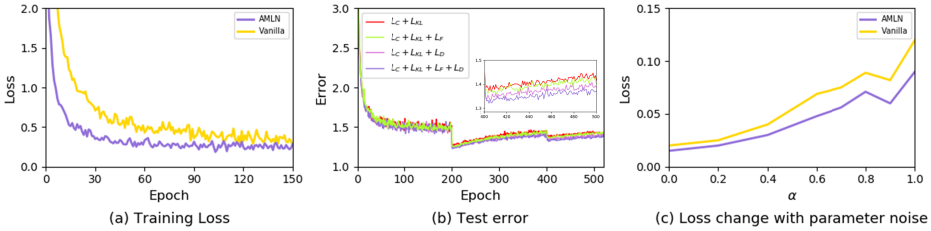

Fig. 3: Analysis of AMLN: The graph in (a) shows the training loss in the first 150 epoch of AMLN and a vanilla model. The graph in (b) shows the testing error under the guidance of different transferring losses. The graph in (c) shows the loss fluctuation when adding parameter noise α during the training of AMLN and a vanilla model.

4.6 Discussion

Benefits of Intermediate Supervision To evaluate the benefit of combining outcome-driven and process-driven learning in the training procedure, we visualize the training loss (in the first 150 epoch) and test error with the peer networks of ResNet32 on CIFAR100. As illustrated in Fig. 3, our model (the purple line) converges faster than the fully trained vanilla model in Fig. 3(a). Compared to other loss combinations, AMLN (the $L_C + L_{KL} + L_F + L_D$ case) has a relatively lower testing error, especially after 400 epoch. See the zoom-in window for details in Fig. 3(b). In addition, we compare the training loss of the learned models before and after adding Gaussian noise α to model parameters. As shown in Fig. 3(c), the training loss of AMLN increases much less than the independent model after adding the perturbation. These clearly indicate that process-driven learning could improve the model stability and AMLN provides better generalization performance.

Qualitative analysis To provide insights on how AMLN contributes to the improved performance consistently, we visualize the heatmaps of learned features after the last convolution layer from four different networks AMLN, FFL, ONE and the vanilla model. We use the Grad-CAM [32] algorithm which works by visualizing the important regions where the network has focused on to discover how our model is taking advantage of the features. Fig. 4 shows the Grad-CAM visualizations from each network with the highest probability and the corresponding predicted class. From the first two columns where all the evaluated models predict the correct class, it shows that our AMLN detects the object better with higher rate of confidence. In addition, the last four columns are the cases where AMLN predicts the correct answer but others do not. It again demonstrates the superior performance of our proposed online distillation method AMLN, in which both process-driven and outcome-driven learning effectively complement with each other for multifarious and discriminative feature distillation.

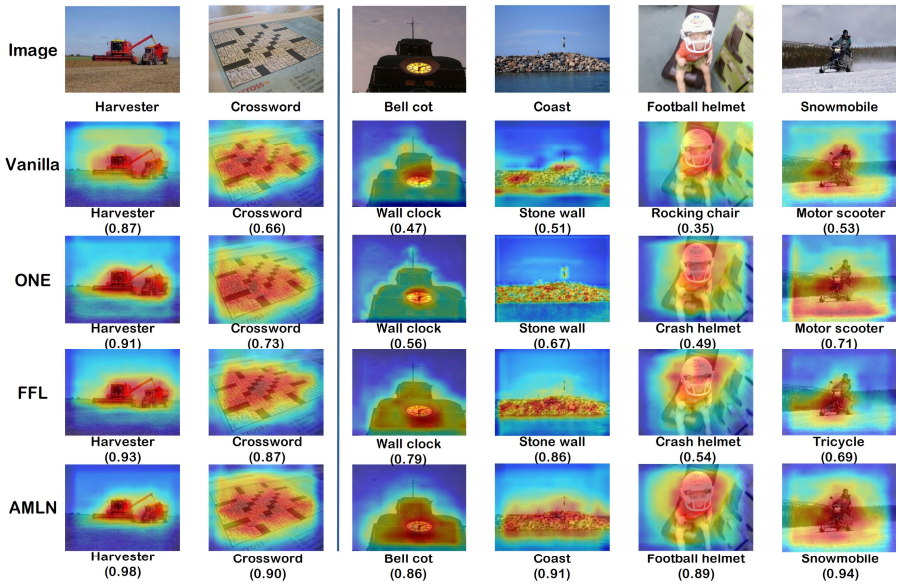

Fig. 4: The comparison of Grad-CAM [32] visualizations of the proposed AMLN with state-of-the-art methods FFL and ONE as well as the vanilla model where the peer networks use the same architecture ResNet32. The label under each heatmap is the corresponding predicted class with the highest prediction probability in the parenthesis.

5 Conclusion

In this paper, a novel online knowledge distillation method is proposed, namely the adversarial-based mutual learning network (AMLN). Unlike existing methods, AMLN employs both process-driven and outcome-driven mutual knowledge distillation, where the former is conducted by the proposed block-wise module with a discriminator and an alignment container for intermediate supervision from the peer network. Extensive evaluations of our proposed AMLN method are conducted on three challenging image classification datasets, where a clear outperformance over the state-of-the-art knowledge transfer methods is achieved. In our future work, we will investigate how to incorporate different tasks to train the peer networks cooperatively, not limited to using the same dataset while mutually training the peer networks as in this work.

Acknowledgements

This work is supported in part by National Science Foundation of China under Grant No.61572113, and the Fundamental Research Funds for the Central Universities under Grants No.XGBDFZ09.

References

1. Alex Krizhevsky, V.N., Hinton, G.: Cifar-10 (canadian institute for advanced research)
2. Alex Krizhevsky, V.N., Hinton, G.: Cifar-100 (canadian institute for advanced research)
3. Anil, R., Pereyra, G., Passos, A., Ormandi, R., Dahl, G.E., Hinton, G.E.: Large scale distributed neural network training through online distillation. arXiv preprint arXiv:1804.03235. (2018)
4. Ba, L.J., Caruana, R.: Do deep nets really need to be deep? Advances in Neural Information Processing Systems. pp. 2654–2662. (2013)
5. Bucilu, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 535–541. (2006)
6. Bulat, A., Tzimiropoulos, G.: Binarized convolutional landmark localizers for human pose estimation and face alignment with limited resources. In IEEE International Conference on Computer Vision, p. 37063714. (2017)
7. C.-Y. Lee, S. Xie, P.G.Z.Z., Tu., Z.: Deeply supervised nets. In Artificial Intelligence and Statistics, pp. 562–570. (2015)
8. Chen, T., Goodfellow, I., Shlens, J.: Net2net: Accelerating learning via knowledge transfer. In International Conference on Learning Representations. (2016)
9. Courbariaux, M., Hubara, I., Soudry, D., Ran, E.Y., Bengio, Y.: Binarized neural networks: Training deep neural networks with weights and activations constrained to +1 or -1. arXiv preprint arXiv:1602.02830. (2016)
10. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3150–3158. (2016)
11. Felzenszwalb, P.F., Girshick, R.B., Mcallester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. In IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 1627–1645. (2010)
12. G.Hinton, O., J.Dean: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531. (2014)
13. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., WardeFarley, D., Ozair, S., Courville, A., , Bengio, Y.: Generative adversarial nets. In Advances in Neural Information Processing Systems. (2014)
14. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. In Advances in Neural Information Processing Systems, pp. 1135–1143. (2015)
15. Hao, L., Kadav, A., Durdanovic, I., Samet, H., Graf, H.P.: Pruning filters for efficient convnets. arXiv preprint arXiv:1608.08710. (2016)
16. He, K., Zhang, X., Ren, S., Jian, S.: Deep residual learning for image recognition. pp. 770–778. (2016)
17. He, Y., Zhang, X., Jian, S.: Channel pruning for accelerating very deep neural networks. In IEEE International Conference on Computer Vision, p. 13891397. (2017)
18. Heo, B., Lee, M., Yun, S., Choi, J.Y.: Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In Proceedings of AAAI Conference on Artificial Intelligence, pp. 3779–3787. (2019)
19. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. (2017)

20. Iandola, F.N., Han, S., Moskewicz, M.W., Ashraf, K., Dally, W.J., Keutzer, K.: Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size. arXiv preprint arXiv:1602.07360. (2016)
21. Jian-Hao Luo, J.W., Lin, W.: Thinet: A filter level pruning method for deep neural network compression. In IEEE International Conference on Computer Vision, p. 50585066. (2017)
22. Kim, J., Hyun, M., Chung, I., Kwak, N.: Feature fusion for online mutual knowledge distillation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2019)
23. Kim, J., Park, S., Kwak, N.: Paraphrasing complex network: Network compression via factor transfer. Advances in Neural Information Processing Systems, p. 27602769. (2018)
24. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems, pp. 1097–1105. (2012)
25. Lan, X., Zhu, X., Gong, S.: Knowledge distillation by on-the-fly native ensemble. In Advances in Neural Information Processing Systems, pp. 7528–7538. (2018)
26. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg., A.C.: Ssd: Single shot multibox detector. In European Conference on Computer Vision, pp. 21–37. (2016)
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 3431–3440. (2015)
28. Molchanov, P., Tyree, S., Karras, T., Aila, T., Kautz, J.: Pruning convolutional neural networks for resource efficient transfer learning. arXiv preprint arXiv:1611.06440. (2016)
29. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: Xnor-net: Imagenet classification using binary convolutional neural networks. In European Conference on Computer Vision, p. 525542. (2016)
30. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550. (2014)
31. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., and, M.B.: Imagenet large scale visual recognition challenge. International Journal of Computer Vision, pp. 211–252. (2015)
32. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In Proceedings of IEEE International Conference on Computer Vision, p. 618626. (2019)
33. Song, G., Chai, W.: Collaborative learning for deep neural networks. Advances in Neural Information Processing Systems, pp. 1837–1846, (2018)
34. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. arXiv preprint arXiv:1612.03928. (2016)
35. Zagoruyko, S., Komodakis, N.: Wide residual networks. arXiv preprint arXiv:1605.07146. (2016)
36. Zhang, X., Gong, H., Dai, X., Yang, F., Liu, N., Liu, M.: Understanding pictograph with facial features: End-to-end sentence-level lip reading of chinese. In Proceedings of the AAAI Conference on Artificial Intelligence, pp. 9211–9218. (2019)
37. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 4320–4328. (2018)