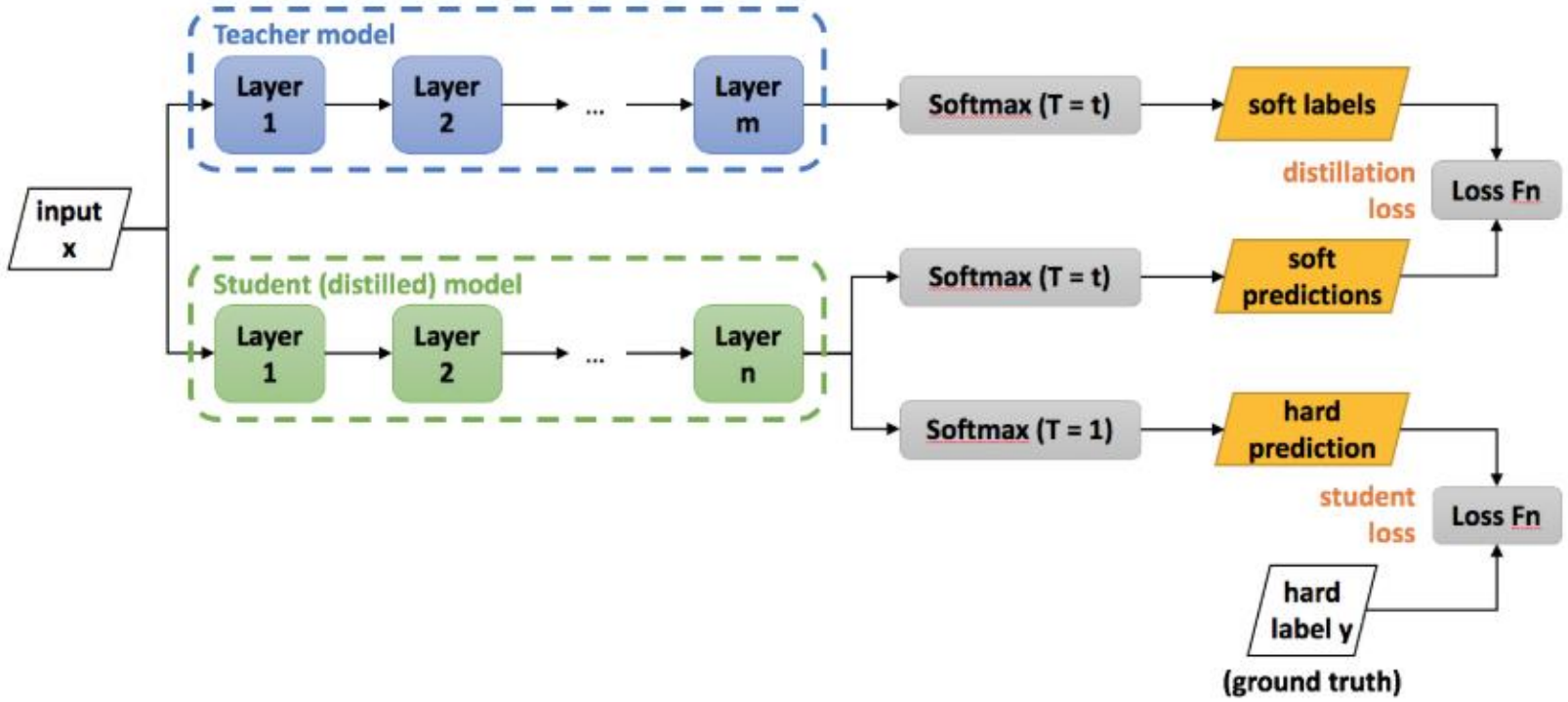
Knowledge Distillation with a Precise Teacher and Prediction with Abstention

Yi Xu, Jian Pu, Hui Zhao

School of Software Engineering, East China Normal University,
Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University
Email: 51184501068@stu.ecnu.edu.cn , jianpu@fudan.edu.cn , hzhao@sei.ecnu.edu.cn

Knowledge Distillation

$$\mathcal{L}_{KD} = \alpha\tau^2 \cdot CE(q_\tau, p_\tau) + (1-\alpha) \cdot CE(y, p_\tau)$$

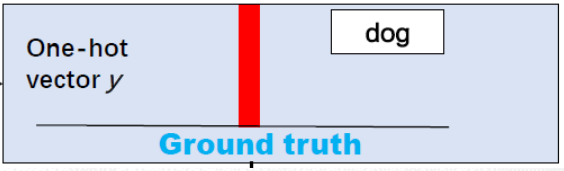


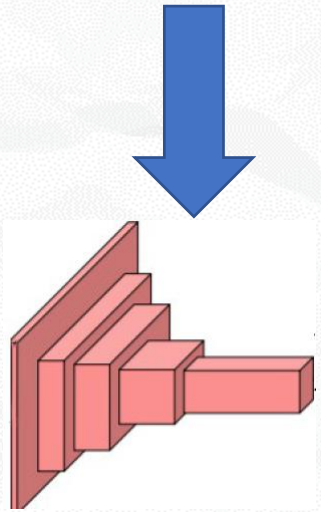
Knowledge distillation is a process of distilling or transferring the knowledge from a (set of) large, cumbersome model(s) to a lighter, easier-to-deploy single model, without significant loss in performance

Introduction

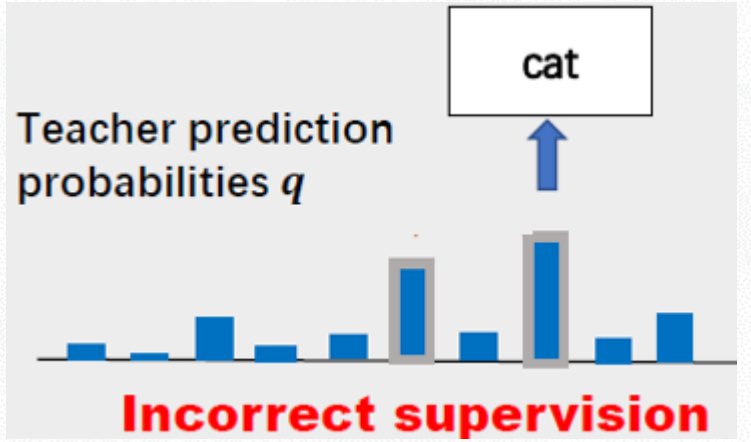
However, there exist two major problems with knowledge distillation methods.

input

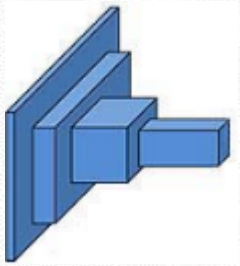




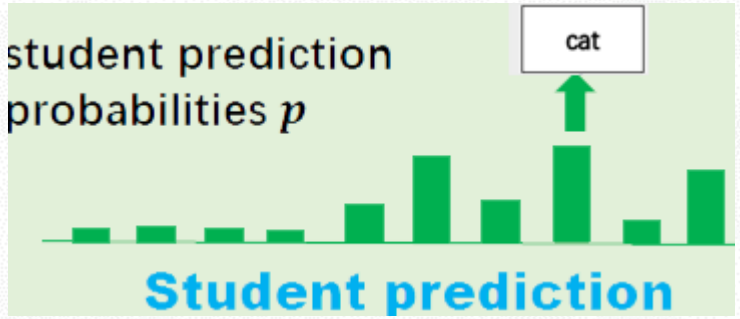
Teacher model



Q1: Teacher's supervision is sometimes **misleading.** → **Incorrect supervisions in teacher model**



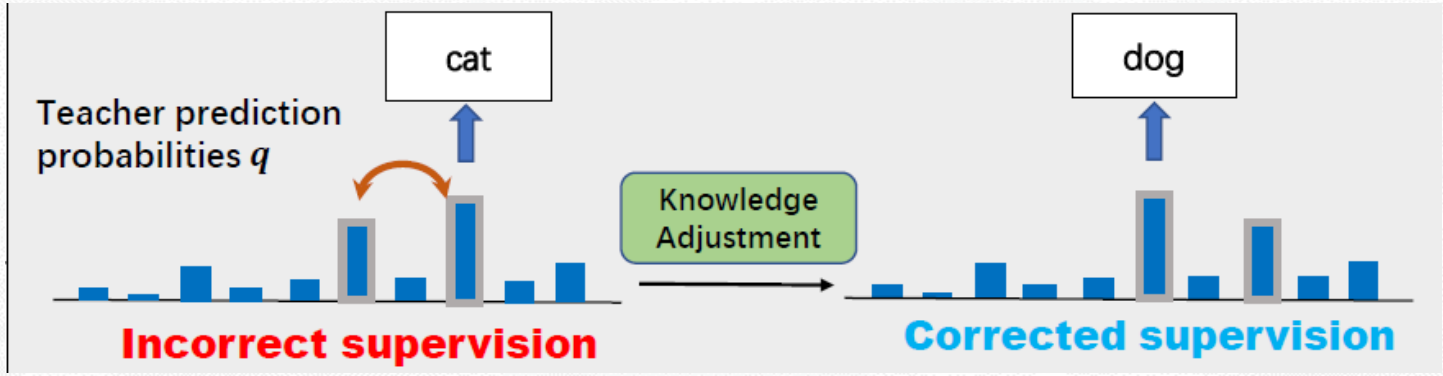
Student model



Q2: Student's prediction is **not accurate** enough**.** → **bad prediction in student model**

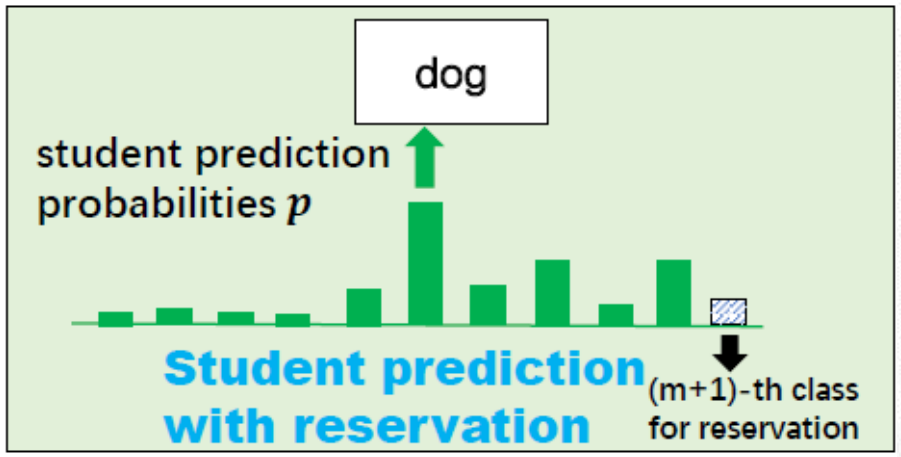
Method

The solution to Q1 => knowledge adjustment=> correct the incorrect supervision



The solution to Q2 => selective classification framework => prediction with reservation

Deep Gambler loss



Model

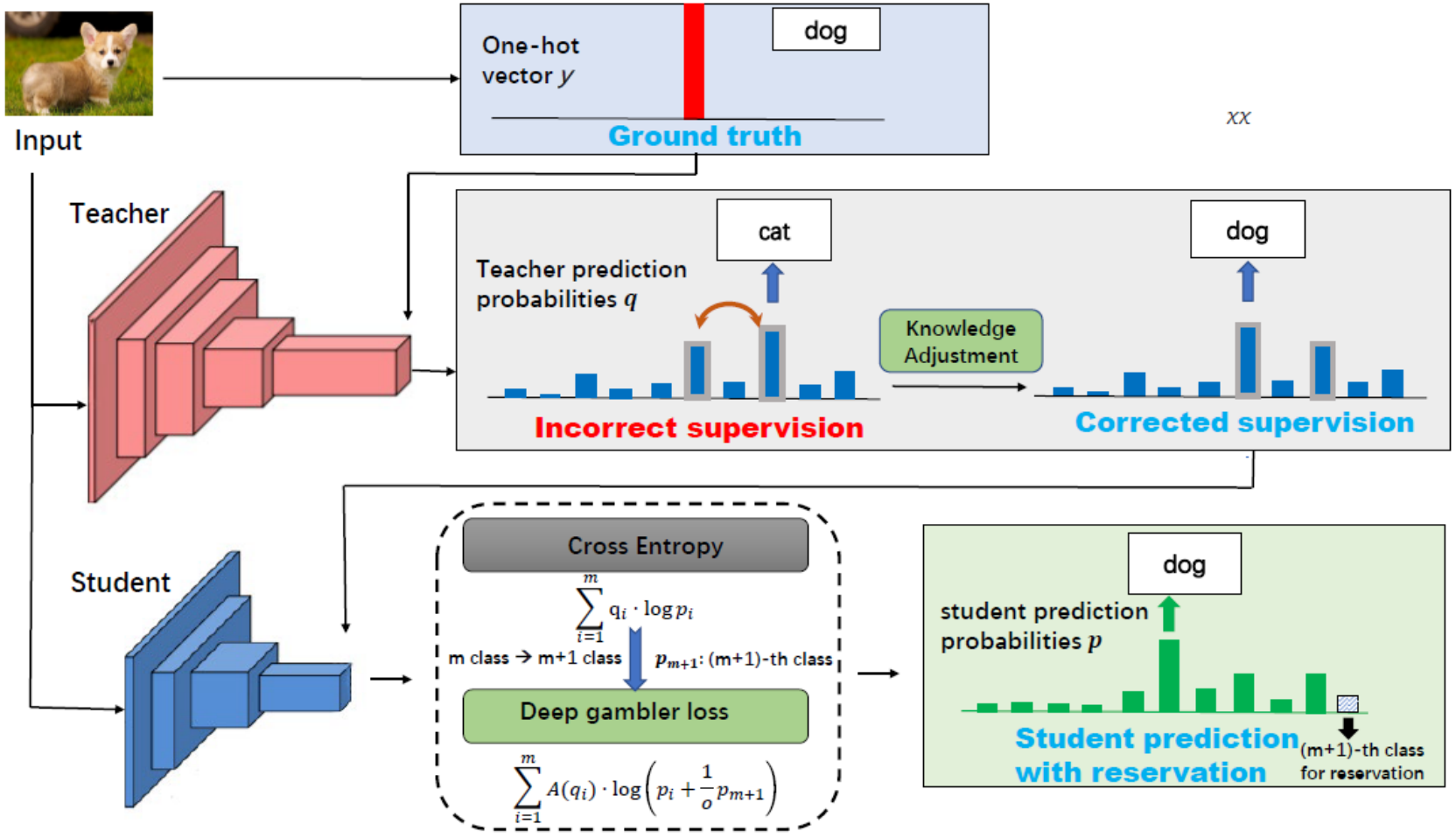


Fig. 1: Illustration of the training processing for the proposed framework. The ground truth labels are denoted by one-hot vectors and are used to train the teacher network. For the output of the teacher network, we use knowledge adjustment to swap the incorrect predictions. The corrected predictions are then considered as the supervision information to train the student model. The deep gambler loss is applied to learn the prediction function and scoring function simultaneously.

- train the student network in an end-to-end way.
- evaluate the proposed method under two knowledge distillation settings.(different network structures and networks with different depths.)

Method

Knowledge Adjustment

Given an incorrect soft target, knowledge adjustment simply swaps the value of ground truth (the theoretical maximum) and the value of predicted class (the predicted maximum),to assure the maximum confidence is reached at ground truth label.

It keeps the numerical distribution of soft targets and dark knowledge in network

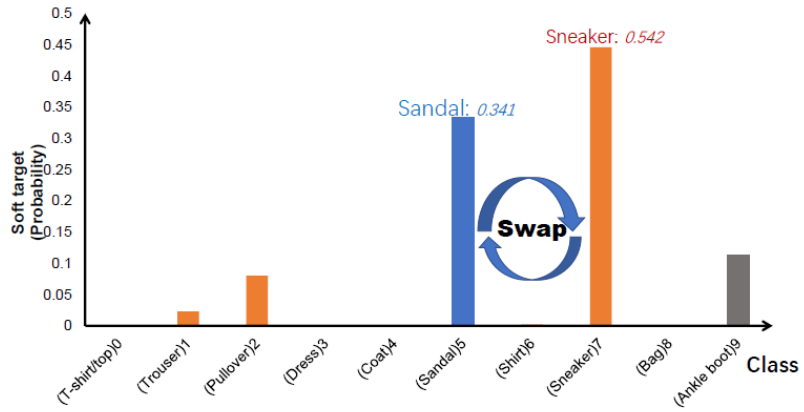


Fig. 2: Illustration of the proposed knowledge adjustment using the Fashion-MNIST dataset. The ground truth label of this instance is "Sandal", while the largest probability of the teacher model is "Sneaker". The knowledge adjustment is to swap the soft probability of these two classes, and the adjusted distribution is considered as the corrected supervision for the student model.

Metric	Stu(resnet2)	Tea(resnet50	KD	KD-AT	KD_DML	KD_KA
Top-1	91.030	92.257	91.119	91.822	91.574	92.168

Method

Knowledge Adjustment

The soft target $q_{i,\tau}$ in Knowledge Distillation with temperature τ

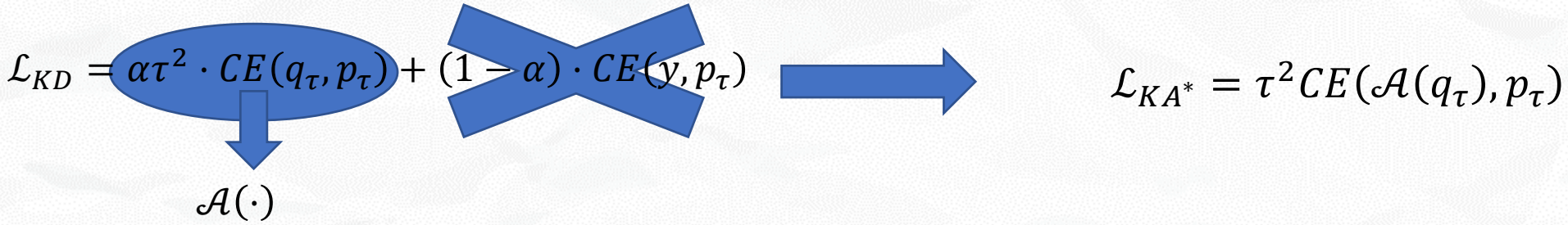
$$q_{i,\tau} = \frac{\exp(z_i/\tau)}{\sum_j \exp(z_j/\tau)}$$

The KD loss consists of cross-entropy between student's soft logits p_τ and teacher's soft logits q_τ along with true target

$$\mathcal{L}_{KD} = \alpha\tau^2 \cdot CE(q_\tau, p_\tau) + (1-\alpha) \cdot CE(y, p_\tau)$$

Given an incorrect target, the simplest way to redress them is to swap the incorrect value with the true targets, To simplify the process and not affect the overall supervision distribution, we only need to operate on the incorrect ones

We denote it as an operator A(·). The KD loss becomes:



Method

Selective Classification

Selective classification is the problem of simultaneously choosing which data examples to classify, and subsequently classifying them. Put another way, it's about giving a classifier the ability to ignore certain data if it's not confident in its prediction.

$$(f, g) \triangleq \begin{cases} f(x), & g(x) = 1 \\ reject, & g(x) = 0 \end{cases}$$

Conclusion:

Achieve better performance with some degree of data coverage

Learning to Abstain with Portfolio Theory – Deep Gambler Loss

if we have a m-class classification problem, we can instead perform a m+1 class classification which predicts the probabilities of the m classes and use the (m+1)-th class as an additional rejection score

Conclusion:

Achieve balance between making prediction and reservation

Method

Deep Gambler loss

$$W(\mathbf{b},\mathbf{p}) = \mathbb{E}\log_2(S) = \sum_{i=1}^{m} p_i \log_2(b_i o_i)$$

Add reservation

$$S(\mathrm{x}_j) = b_j o_j + b_{m+1}$$

$$\max W(\mathbf{b},\mathbf{p}) = \sum_{i=1}^{m} p_i \log(b_i o_i + b_{m+1})$$

$$\max_f W(\mathbf{b}(f),\mathbf{p}) = \max_{\mathbf{w}} \sum_{i}^{B} \log\left[f_{\mathbf{w}}(x_i)_{j(i)} o + f_{\mathbf{w}}(x_i)_{m+1}\right]$$

Method

Prediction with reservation by adding a class => NOT attempt to improve the accuracy with full coverage

Knowledge adjustment to get rid of incorrect supervision=> NOT handle with uncertain predictions

So we proposed the loss function that utilizes Deep Gambler (DG) loss to the KA method.

$$\mathcal{L} = \sum_{i} \mathcal{A}\left(q_\tau^i\right) \log\left(p_\tau^i + \frac{1}{o} p_\tau^{m+1}\right)$$

Method

Algorithm 1 The Training Procedure of the Proposed Knowledge Distillation Method.

Input:
Teacher model T; Initialized Student model S; Training data $\mathbf{X}$; Ground truth label $\mathbf{y}$; Hyperparameter o, τ;

Output: Student model S;

1: Load the teacher model T;
2: **for** epoch=$1, \cdots$ **do**
3: Calculate the logits q_τ^i from outputs of S and p_τ^i from outputs of T via softmax function of temperature τ;
4: Perform knowledge adjustment and compute $\mathcal{A}(q)$;
5: Evaluate the loss function according to proposed method;
6: Update the student model S by back propagation;
7: **end for**
8: **Return:** the training loss $\mathcal{L}$.

Experiment

Dataset & setting

Evaluated in four different datasets: Fashion-MNIST, SVHN, CIFAR10 and CIFAR100 using two knowledge distillation settings:

- Distillation across Different Network Structures: (AlexNet, ResNet)
- Distillation across Networks with Different Depths: (ResNet18, ResNet50).

Benchmark

Compare the performance of the student model, teacher model, selective classification using softmax, and original deep gambler method for classification.

Experiment

Evaluation Metric

Report the prediction accuracy of student network without rejection. Besides, we investigate the accuracies with various prediction coverages. To measure the performance of selective classification, we accumulate the error rate of various coverage. For a finite set S containing target coverages, the Sum Coverage Error(S) is to accumulate the test errors at these coverages:

$$\text{Sum Coverage Error}(S) = \sum_{i \in S} Error(i)$$

where Error(i) denote the selective prediction error at coverage i.

Obviously, Sum Coverage Error(S) is the smaller the better.

Result

Distillation across Different Network Structures

Accuracy

TABLE I: The comparison of accuracy on four datasets by knowledge distillation across different network structures.

Method	Fashion-MNIST	SVHN	CIFAR10	CIFAR100
	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy(%)
Student(AlexNet)	92.60	94.86	85.57	60.71
Teacher(ResNet50)	93.95	97.47	95.42	76.86
Deep Gambler [18]	92.64	95.02	**87.17**	61.17
Proposed Method	**92.94**	**95.05**	87.11	**61.24**

Coverage Error

TABLE II: The comparison of Sum Coverage Error in 0%-100% and 70%-100% by knowledge distillation across different network structures.

Method	Fashion-MNIST		SVHN	
	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)
Softmax Response [19]	130.21	110.70	218.48	149.43
Deep Gambler [18]	119.03	105.63	195.96	135.60
Proposed Method	**114.58**	**86.88**	**181.96**	**124.73**
Method	CIFAR10		CIFAR100	
	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)
Softmax Response [19]	287.52	222.33	1826.63	973.31
Deep Gambler [18]	**265.85**	**217.62**	1612.89	958.59
Proposed Method	276.37	220.54	**1598.24**	**949.50**

Result

Distillation across Network with Different Depth

Accuracy

TABLE III: The comparison of accuracy on four datasets by knowledge distillation across different network depths.

Method	Fashion-MNIST	SVHN	CIFAR10	CIFAR100
	Accuracy(%)	Accuracy(%)	Accuracy(%)	Accuracy (%)
Student(ResNet18)	93.64	97.25	95.14	76.42
Teacher(ResNet50)	93.95	97.47	95.42	76.86
Deep Gambler [18]	93.79	97.20	95.12	76.54
Proposed Method	**93.92**	**97.41**	**95.42**	**76.86**

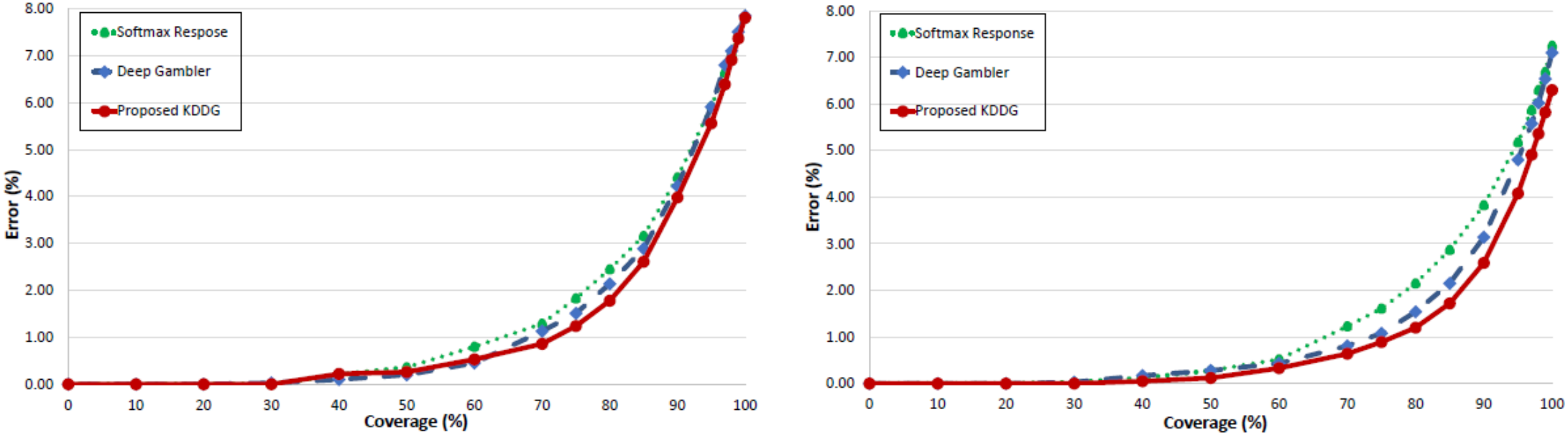
Coverage Error

TABLE IV: The comparison of Sum Coverage Error in 0%-100% and 70%-100% by knowledge distillation across different network scales.

Method	Fashion-MNIST		SVHN	
	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)
Softmax Response [19]	113.30	98.05	115.18	61.19
Deep Gambler [18]	95.30	82.55	122.61	64.80
Proposed Method	**77.18**	**68.82**	**106.93**	**55.89**
Method	CIFAR10		CIFAR100	
	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)	Sum Coverage Error (0,100)	Sum Coverage Error (70,100)
Softmax Response [19]	61.41	50.11	867.02	521.01
Deep Gambler [18]	63.53	55.94	867.32	519.13
Proposed Method	**59.81**	**47.90**	**853.48**	**516.46**

Result

Visualize the Coverage-Error curves using the Fashion-MNIST dataset in Figure 3.



(a) AlexNet & ResNet50

(b) ResNet18 & ResNet50

Fig. 3: The comparison of Coverage-Error curves under two knowledge distillation settings using the Fashion-MNIST dataset. (a) Distillation across Different Network Structures; (b) Distillation across Networks with Different Depths.

We sample intensively in the interval 70% to 100% for the reason of leading role in the coverage rate and uniformly in the interval 0% to 60%. Notice that the gap between our method and two competitors are significant, especially when the target coverage is in (60,90).

Conclusion

- We have proposed a novel method for knowledge distillation to tackle the problems of inaccurate supervision and the lack of prediction confidence for the student model.
- Knowledge Adjustment is used to rectify teachers' incorrect supervision without involving additional hyperparameters.
- To learning a scoring function with classification, we adopt the Deep Gambler loss by introducing an extra class for reservation.
- The proposed method exhibits superior performance on four benchmark datasets, in terms of both prediction accuracy and Coverage-Error curves.