

Introspective Learning by Distilling Knowledge from Online Self-explanation

Jindong Gu^{1,2}, Zhiliang Wu^{1,2}, and Volker Tresp^{1,2}

¹ University of Munich, Munich, Germany

² Siemens AG, Corporate Technology, Munich, Germany

Abstract. In recent years, many methods have been proposed to explain individual classification predictions of deep neural networks. However, how to leverage the created explanations to improve the learning process has been less explored. The explanations extracted from a model can be used to guide the learning process of the model itself. Another type of information used to guide the training of a model is the knowledge provided by a powerful teacher model. The goal of this work is to leverage the self-explanation to improve the learning process by borrowing ideas from knowledge distillation. We start by investigating the effective components of the knowledge transferred from the teacher network to the student network. Our investigation reveals that both the responses in non-ground-truth classes and the class-similarity information in teacher’s outputs contribute to the success of the knowledge distillation. Motivated by the conclusion, we propose an implementation of introspective learning by distilling knowledge from online self-explanations. The models trained with the introspective learning procedure outperform the ones trained with the standard learning procedure, as well as the ones trained with different regularization methods. When compared to the models learned from peer networks or teacher networks, our models also show competitive performance and requires neither peers nor teachers.

1 Introduction

When human subjects imagine visual objects without actual sensory stimuli, there is still significant activity in their visual cortices [1]. This implies that the internal representations in our visual cortex are used to reason about images. Similarly, when explaining classifications of neural networks, the existing model-aware explanation methods create explanations based on internal feature representations of neural networks. The explanation of each prediction identifies the relationship between input features and outputs. For instance, in image classification, saliency maps are proposed to explain how each input pixel is relevant to different classes, [2–6] to name a few.

It has been shown that the performance of human decisions is related to the explanations during introspection in the brain [7, 8]. In the deep learning community, there is no clear definition of introspective learning. In this work, we define *introspective learning* as learning with explanations created from the

underlying model itself. In recent years, the explanations extracted from the learned models are mainly used to gain trust from users in real-world applications and help machine learning experts understand the predictions. How to leverage the explanations to improve the model itself has not been well explored.

Broadly speaking, learning with explanations falls into the framework of learning using privileged information [9, 10]. In classical supervised learning, the training data are $\{(\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{y}_n)\}$ where $(\mathbf{x}_i, \mathbf{y}_i)$ is an input-label pair, \mathbf{y}_i is a c -dimensional vector, and c is the number of output classes. In the framework of learning using privileged information, additional information \mathbf{x}_i^* about $(\mathbf{x}_i, \mathbf{y}_i)$ is provided by an intelligent teacher. The training data are therefore formed by a collection of triplets

$$\{(\mathbf{x}_1, \mathbf{x}_1^*, \mathbf{y}_1), \dots, (\mathbf{x}_n, \mathbf{x}_n^*, \mathbf{y}_n)\}. \quad (1)$$

For instance, \mathbf{x}_i^* can be the softened outputs of a teacher model [11]. In this work, we consider the explanation \mathbf{E}_i created for the classification of $(\mathbf{x}_i, \mathbf{y}_i)$ as the privileged information.

When the softened outputs of a powerful teacher model are provided as the privileged information, the learning process is known as knowledge distillation (KD) as defined by Hinton et al. [12], where a student network is trained to match its softened outputs to the teacher’s.

In this work, we first investigate the effective components in the knowledge distillation. Our investigation shows that both the responses in non-ground-truth classes and class-similarity information in teacher’s outputs contribute to the effectiveness of knowledge distillation. Motivated by the conclusions, we propose a novel training procedure to train networks by leveraging online self-explanations.

In our introspective learning paradigm, for the classification of the i -th sample \mathbf{x}_i , we create one saliency map for each output class, i.e., $\mathbf{E}_i = (\mathbf{e}_i^1, \mathbf{e}_i^2, \dots, \mathbf{e}_i^c)$ where c is the total number of output classes. The saliency map (explanation) is a heatmap of the same size as the input. The individual values of the heatmap indicate the relevance of the corresponding input features. The explanation \mathbf{e}_i^j is a function of the input \mathbf{x}_i , the underlying model M and the target class y_i^j , namely $\mathbf{e}_i^j = f_e(\mathbf{x}_i, M, y_i^j)$ where y_i^j is the j -th output class index of the i -th example. They contain neither additional human prior knowledge nor knowledge of a powerful teacher model. We leverage the explanations to guide the training process in each training epoch.

It is perhaps not obvious why explanations can be used to improve the model at all. Our hypothesis is that the explanations can be leveraged to help the training process to find a better local minimum in a way similar to KD. [13] observes that small networks often have the same representation capacity as large networks. Compared with large networks, it is more difficult to converge to good local minima. In our work, we leverage explanations to help the training process to find a better local minimum.

The contributions of our work are two-fold. Firstly, we investigate which components of KD contribute to its success. Secondly, inspired by the investigation results, we propose a way to implement introspective learning by distilling

knowledge from self-explanations. The rest of the paper is structured as follows: Sec. 2 reviews related work. Sec. 3 and 4 describe the two contributions. Sec. 5 conducts experiments to show that our proposed learning procedure can achieve competitive performance even without any teacher. The last section concludes the paper.

2 Related Work

Explanation: The explanation of a classification for the j -th class e_i^j illustrates how relevant the input features are to the prediction of the j -th class. Vanilla Gradient (*Grad*) takes the gradient of an output neuron with respect to the input as an explanation [2]. Guided Backpropagation (*GuidedBP*) differs from the Grad approach in handling ReLU layers, where only positive gradients are propagated through the ReLU layers [3]. [4] multiplies the Grad with the input (*Grad*Input*) to deal with numerical stability. [14] shows that *Grad*Input* is equivalent to LRP [15] and DeepLIFT [4]. Another commonly used approach (*Grad-CAM*) combines the feature maps linearly using the averaged gradients of each feature map [5].

All the approaches introduced above require only a single backpropagation to create an explanation, while SmoothGrad [16] and IntergratedGrad [17] require dozens of passes. Some other approaches require two backpropagations, e.g., Contrastive LRP, although they create visually pleasing and class-discriminative explanations [18, 19]. The explanations created by these saliency methods are often applied to diagnose models.

Regularization: We briefly introduce closely related regularization techniques. Label Smoothing [20] replaces the one-hot labels with soft labels where α is specified for the ground-truth class, and $(1 - \alpha)$ is evenly distributed to other classes (α is often set as 0.9). [21] proposed to penalize low entropy predictions, called Confidence Penalty. Another commonly used regularization is Dropout [22], which drops each neuron of a layer with a certain probability.

Knowledge Distillation: The knowledge of a teacher model can be used to guide the training of a student model. Many works focus on different forms of the dark knowledge extracted from the teacher, e.g., soft outputs [23, 12], Hints (intermediate representations) [24], Activation-based. Gradient-based attention [25], and Flow between layers [26].

Recent research shows that a more powerful teacher and a peer network can help train a student. [27] shows that a poorly-trained teacher with much lower accuracy than the student can still improve the latter significantly, and a student can also enhance the teacher by reversing the KD procedure. [28] distills knowledge from a peer network with a similar capacity. The peer network can even be with the same architecture; it is known as Born-Again Network. The work [29] searches for better student architectures to learn distilled knowledge. Deep Mutual Learning [30] train multiple students with the same architecture and regularize their outputs to be similar, where each student can outperform

the one trained alone without mutual learning. In this work, we propose to learn from a student itself by distilling knowledge from its explanations.

Effectiveness of KD: [12] argues that the success of KD can be attributed to the information in the logit distribution that describes the similarity between output categories. [28] investigates gradients of the loss in KD and shows that weighting training examples with the teacher’s confidence can also improve the student model’s performance. In this work, we conduct further experiments to reveal effective components in KD.

3 Understanding Dark Knowledge

In our introspective learning, the explanations extracted from the learned models are used to improve model performance, while the knowledge provided by another teacher model is used in KD. To borrow ideas from the success of KD, we investigate the dark knowledge that improves the student in KD.

Given a training example $(\mathbf{x}_i, \mathbf{y}_i)$ and the number of output classes c , the logits and the output probabilities of a network are $\mathbf{a}_i = (a_i^1, a_i^2, \dots, a_i^c)$ and $\mathbf{p}_i = \text{softmax}(\mathbf{a}_i) = (p_i^1, p_i^2, \dots, p_i^c)$ respectively. We denote the target outputs as \mathbf{t}_i . The target can be the one-hot labels \mathbf{y}_i in supervised learning, while it can also be the softened output probabilities of a teacher model \mathbf{q}_i in KD.

In both cases, given a mini-batch $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^b$, the gradient of the cross-entropy loss $\mathcal{L} = \frac{1}{b} \sum_{i=1}^b ce(\mathbf{p}_i, \mathbf{t}_i)$ with respect to the logit a^j is

$$\frac{\partial \mathcal{L}}{\partial a^j} = \frac{1}{b} \sum_{i=1}^b \frac{\partial \mathcal{L}}{\partial a_i^j} = \frac{1}{b} \sum_{i=1}^b (p_i^j - t_i^j). \quad (2)$$

Without loss of generality, the samples in the mini-batch can be divided into two parts: $\{(\mathbf{x}_i)\}_{i=1}^s$ with the j -th class as the ground-truth class ($y_i^j = 1$) and $\{(\mathbf{x}_i)\}_{i=s+1}^b$ with other classes as ground-truth classes ($y_i^j = 0$). In the case of the standard supervised learning where $\mathbf{t} = \mathbf{y}$ (denoted as **CE** for cross entropy), the gradient can be formulated as

$$\frac{\partial \mathcal{L}}{\partial a^j} = \frac{1}{b} \left(\sum_{i=1}^s (p_i^j - 1) + \sum_{i=s+1}^b p_i^j \right). \quad (3)$$

In the case of training with KD where $\mathbf{t} = \mathbf{q}$ (denoted as **KD**³), the gradient can be formulated as

$$\frac{\partial \mathcal{L}}{\partial a^j} = \frac{1}{b} \left(\sum_{i=1}^s (p_i^j - q_i^j) + \sum_{i=s+1}^b (p_i^j - q_i^j) \right). \quad (4)$$

[28] argues that the teacher’s confidence is the effective component of KD, and KD simply performs importance weighting. With one-hot labels, they propose

³ Similar to [28], the temperature of softmax in KD is hidden to simplify notations.

Table 1: Test Accuracy (%) of CNNs on CIFAR10 dataset: the teacher CNN-10 achieves 90,04% test accuracy, and the performance of students are listed. CWTM-P corresponds to CWTM-Permut, and CWTM-R to CWTM-Random.

	Model	CE	KD	CWTM	CWTM-P	CWTM-R	DKPP
Student1	CNN-8	86,86(± 0.30)	88,48(± 0.33)	87,03(± 0.26)	87,41(± 0.32)	87,22(± 0.17)	87,73(± 0.37)
Student2	CNN-6	85,18(± 0.32)	86,55(± 0.32)	85,65(± 0.38)	85,74(± 0.33)	85,25(± 0.31)	86,30(± 0.42)

to weigh each sample by the teacher’s confidence in its maximum value, i.e., Confidence Weighted by Teacher Max (**CWTM**).

$$\frac{\partial \mathcal{L}}{\partial a^j} = \sum_{i=1}^s \frac{q_i^{max}}{\sum_{j=1}^b q_j^{max}} * (p_i^j - 1) + \sum_{i=s+1}^b \frac{q_i^{max}}{\sum_{j=1}^b q_j^{max}} * p_i^j \quad (5)$$

where q_i^{max} is the maximum of teacher’s outputs q_i .

To check the effectiveness of class-similarity information, they randomly permute the teacher’s non-argmax outputs of predicted distribution on a sample, i.e., Dark Knowledge with Permuted Predictions (**DKPP**). The q_i^j in the second term of Equation 4 is replaced by q_i^k , which is one of the permuted non-argmax outputs. By doing this, the information about the similarity between classes is removed.

We design two experiments to verify the effectiveness of teacher’s confidence: 1) **CWTM-Permut**: the teacher’s confidence (argmax outputs) is randomly permuted inside a batch, namely the weight of the sample $(\mathbf{x}_i, \mathbf{y}_i)$ in Equation 5 is set as $\frac{q_k^{max}}{\sum_{j=1}^n q_j^{max}}$ where k is one of the permuted sample indexes; 2) **CWTM-Random**: q_i^{max} in Equation 5 is replaced by a random value selected from $[\beta, 1)$ as the teacher’s confidence.

We conduct experiments on CIFAR10 [31] using convolutional neural networks (CNN). The used CNNs consist of convolutional layers followed by max-pooling and ends with a fully connected layer. The teacher we use has ten convolutional layers, called CNN-10, and the two students are CNN-8 and CNN-6, respectively. The standard data augmentation is applied to the training data: 4 pixels are padded on each side, and a 32×32 patch is randomly cropped from the padded images or their horizontal flip. The β in CWTM-Random is 0.5, which ensures that q_i^{max} corresponds to the maximal value. All models are trained with a batch size of 128 for 160 epochs using SGD with a learning rate of 0.01 and moment 0.9. The test performance is shown in Table 1. All the scores reported in this paper are averaged on five trials in forms of mean(\pm std).

In Table 1, KD outperforms the classical supervised learning CE, which indicates that the targets specified by the teacher do help the training of the students. CWTM outperforms the baseline CE. **The improvement cannot be attributed to the teacher’s confidence since both CWTM-Permut and CWTM-Random also outperform CE.** Therefore, the success of KD cannot attributed to the teacher’s confidence itself (i.e., the maximal output).

Furthermore, as shown in [28], CWTM does not always outperform CE. We also found that CWTM-Random is very sensitive to the choice of β . The non-uniform sampled importance leads to the possible improvement of CWTM variants. We conjecture that it might improve the model by helping the optimization to escape saddle points. Further exploration is left in future work.

DKPP removes the class-similarity information contained in teacher’s outputs by permuting logits of non-ground-truth classes. **DKPP still clearly outperforms CE, which means the responses in non-ground-truth classes contribute to the effectiveness of KD.** Similar to Labels Smoothing, the responses in non-ground-truth classes prevent the student from becoming overconfident. Furthermore, we can also observe that there is a gap between DKPP and KD. Compared to KD, **the student performance does decrease by removing the similarity information in the teacher’s outputs. The class-similarity information also contributes to the effectiveness of KD.** These two effective components are further discussed in Section 5.4.

The observations above are consistent with [28], but we gain more insights into the KD technique with more experiments. From these observations, we conclude that both non-ground-truth classes’ responses and class-similarity information in the teacher’s outputs contribute to the success of KD.

4 Introspective Learning with Online Self-explanations

In this section, we focus on improve the model performance with the explanations created on it. Given a training sample $(\mathbf{x}_i, \mathbf{y}_i)$ and a neural network $f(\cdot)$ to be trained, the outputs of the forward inference are \mathbf{p}_i . A saliency method is applied on the neural network to generate one explanation for each output class $\mathbf{E}_i = (\mathbf{e}_i^1, \mathbf{e}_i^2, \dots, \mathbf{e}_i^c)$.

One desired property of the explanations \mathbf{E}_i is that the explanations are class-discriminative. In terms of a single sample, the similarity between explanations corresponds to the similarity between output classes. For instance, if the similarity between the explanations of two output classes $\text{sim}(\mathbf{e}_i^1, \mathbf{e}_i^2)$ is high, the 1st output class and the 2nd one are similar to each other. The similarity can be measured with different metrics, such as cosine distance, multi-scale mean squared error [32], and Wasserstein distance [33]. In our experiments, we find that the simplest one (i.e., Cosine distance) already captures the similarity between two created explanations.

This section proposes an algorithm to train neural networks with the online self-explanations created during training. As shown in Algorithm 1, our LE (Learning with Explanations) consists of two training stages.

In the first training stage, the network is trained with one-hot labels. The goal is to initialize the network to a good starting point for generating meaningful explanations. Without warming up, the created explanations are almost random initially, which will mislead the training process.

In the second training stage, the network is trained with online soft labels extracted from online self-explanations. Given a training sample $(\mathbf{x}_i, \mathbf{y}_i)$, without

Algorithm 1: Introspective Learning with Online Self-explanations

Data: training samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, a smooth factor α **Result:** a well-trained neural network

Stage 1 (Warm-up Training):

train the network with one-hot labels $\mathcal{L}(\mathbf{p}_i, \mathbf{y}_i)$;

Stage 2 (Training with Online Self-explanations):

for each epoch **do** make forward inference $\mathbf{p}_i = f(\mathbf{x}_i)$; generate explanations $\mathbf{E}_i = (\mathbf{e}_i^1, \mathbf{e}_i^2, \dots, \mathbf{e}_i^c)$; generate targets $\mathbf{q}_i = (q_i^1, q_i^2, \dots, q_i^c)$ with \mathbf{E}_i and \mathbf{y}_i ; train with the loss $\mathcal{L}(\mathbf{p}_i, \mathbf{q}_i) + \lambda \mathcal{L}(\mathbf{p}_i, \mathbf{y}_i)$ **end**

loss of generality, we assume the c -th class is the ground truth $y_i^c = 1$. In each epoch, we first classify the sample $\mathbf{p}_i = f(\mathbf{x}_i)$ and create explanations \mathbf{E}_i for every classification prediction. We then create new targets $\mathbf{q}_i = (q_i^1, q_i^2, \dots, q_i^c)$ to compute the loss instead of using the one-hot labels or soft labels provided by a teacher. The new targets \mathbf{q}_i are based on the explanations \mathbf{E}_i and the one-hot labels \mathbf{y}_i . The one corresponds to the ground-truth class is specified with a pre-defined value $q_i^c = \alpha$, and the ones correspond to other classes are computed as

$$q_i^k = (1 - \alpha) * \frac{\cos(\mathbf{e}_i^k, \mathbf{e}_i^c) + 1}{\sum_{m=1}^{c-1} (\cos(\mathbf{e}_i^m, \mathbf{e}_i^c) + 1)} \quad (6)$$

where $k \neq c$. Since $\cos(\mathbf{e}_i^k, \mathbf{e}_i^c) \in [-1, 1]$, 1 is added to the similarity value for numerical stability. The training process is also regularized by a cross-entropy term with one-hot labels and a common weight-decay term.

Rationale behind the proposed learning procedure: Our proposed learning procedure is motivated by the conclusions drawn in Sec. 3. We propose new targets to guide the training of the network. With the proposed targets, the responses in non-ground-truth classes (wrong responses) prevent the network from becoming over-confident. The wrong responses are not randomly computed but with the information about the explanations. As the similarity between classes corresponds to the similarity between their explanations, the proposed targets contain class-similarity information. Therefore, our learning procedure includes effective components of KD without requiring a powerful teacher model. In other words, our proposed learning procedure can be seen as distilling knowledge directly from online explanations.

Computational cost of the proposed learning procedure: In our proposed algorithm, an extra cost to compute the explanations is required. For each training sample in a mini-batch, c explanations are required to compute the corresponding targets. Using the implementation trick in [34], all c explanations can be obtained in a single backward pass. Hence, compared to the computa-

Table 2: Test accuracy (%) of CNNs trained with different regularizations on CIFAR10 dataset. Our proposed LE outperforms the standard training procedure CE and others with various regularization methods.

Model	CE	Dropout	LS	CP	LE _(Grad-CAM)
ResNet14	90,26 _(±0.33)	90,33 _(±0.15)	90,52 _(±0.12)	90,42 _(±0.14)	91,21 _(±0.07)
ResNet8	87,95 _(±0.32)	88,02 _(±0.02)	88,14 _(±0.16)	88,25 _(±0.13)	88,70 _(±0.18)

tionally expensive teacher model in KD, the extra cost brought by our learning procedure is much less.

5 Experiments

Following the previous work [35], we use ResNet [36] in our experiment. On CIFAR10, the standard data augmentation is applied as in Sec. 3. The models are trained with a batch size of 128 for 160 epochs using SGD. We start with a learning rate of 0.1, divide it by 10 at the 80-th and the 120-th epoch. In the proposed algorithm, we set the smoothing factor $\alpha = 0.9$ and the regularization strength $\lambda = 0.1$.

The method we use to create explanations is the commonly used *Grad-CAM*. It requires to specify a layer to create explanations. And we choose the last convolutional layer. The effectiveness of different explanation methods is also analyzed in Sec. 5.4.

We compare the model trained with our proposed algorithm with the ones trained with different regularizations, the ones learned from peer networks, and the ones learned from a strong teacher. We further investigate our proposal with ablation studies in order to validate our hypothesis in the paper.

5.1 Networks Trained with Regularizations

The proposed algorithm can be viewed as a regularization method. It regularizes the networks so that their predictions are consistent with their explanations. In this experiment, we compare our method with other regularization methods, including Dropout, Label Smoothing (LS), and Confidence Penalty (CP).

For Dropout, we drop neurons of the last layer of each block-layer with the drop rate in the range [0,1, 0.25, 0.5, 0.75], where 0.1 corresponds to the best performance. For LS, we vary the smoothing factor over [0.6, 0.7, 0.8, 0.9], where a factor of 0.9 turns out to be the best. For CP, we vary the penalty strength over [0.1, 0.3, 0.5, 1.0, 2.0, 4.0, 8.0], where the value of 0.5 gives the best result. The corresponding performance is listed in Table 2.

All the regularizations outperform the baseline CE by alleviating the overfitting problem. Meanwhile, our method LE is superior to other regularization methods. LS is similar to ours, which is expected, as it also has responses in non-ground-truth classes in its specified targets. However, in LS, the specified

Table 3: Test error (%) of MLPs trained with various regularizations on MNIST dataset. In our implementation, our LE achieves the best with tiny variance.

Implementations	CE	Dropout	LS	CP	LE(<i>Grad</i>)
[21]	-	1.28(± 0.06)	1.23(± 0.06)	1.17(± 0.06)	-
[38]	1.38	1.34	1.40	1.36	-
Ours	1.39(± 0.05)	1.27(± 0.02)	1.25(± 0.02)	1.18(± 0.06)	1.10 (± 0.03)

response for each non-ground-truth class is the same. It results in the loss of information in the logits about the similarity between classes [37]. Different from LS, the specified targets in our method LE include the class-similarity information.

Following the research line, we also compare our method with others in the setting of previous work [21, 38]. Concretely speaking, an Multilayer Perceptron (MLP) with fully connected layers of the form 784-1024-1024-10 and ReLU activations is trained on the MNIST dataset [39]. All models are trained with a batch size of 16 for 50 epochs using SGD with a learning rate of 0.01 and moment 0.9.

When training MLPs with our proposed algorithm, we applied *Grad* approach to create explanations since *Grad-CAM* is only specific for CNNs. 10 epochs are used for warm-up training in our learning procedure. For other regularizations, we explore their hyper-parameters in the same space as above and show the best of the exploration.

The comparison results in this simple setting are shown in Table 3. The scores of the first two rows are taken from their papers. The third row shows our implementations. In this simple setting with MLP, our method LE also outperforms other regularization methods. We only compare with the popular and state-of-the-art regularization methods and the ones related to our work. The score of VIB regularization proposed in [38] is not listed since the setting is slightly different. Even though they require 12,03% more parameters and 200 epochs to train (test error 1,13%), our score still slight outperform theirs.

We also visualize the created explanations during training in Figure 1. In each row, the first column shows the original image, and others columns show explanations corresponding to different classes $\mathbf{E}_i = (\mathbf{e}_i^1, \mathbf{e}_i^2, \dots, \mathbf{e}_i^{10})$ (10 classes both in MNIST and in CIFAR10).

From Figure 1, we find that the generated explanations are consistent with human’s prior knowledge. For instance, in the first row of Figure 1a, the score under the explanation of the last column (the ground-truth class) is $\cos(\mathbf{e}_i^*, \mathbf{e}_i^*) = 1$. In the explanations corresponding to digits of 4 and 7, part of input features supports the corresponding scores. the similarity between the explanation of digit 4 and the explanation of the digit 9 (ground-truth class) is high. Besides, the similarity between the explanation of digit 7 and the explanation of the digit 9 is also high. Also, In the first row of Figure 1b, the explanation of the ground-truth class focuses on the pixels of the truck, while others focus on irrelevant pixels.

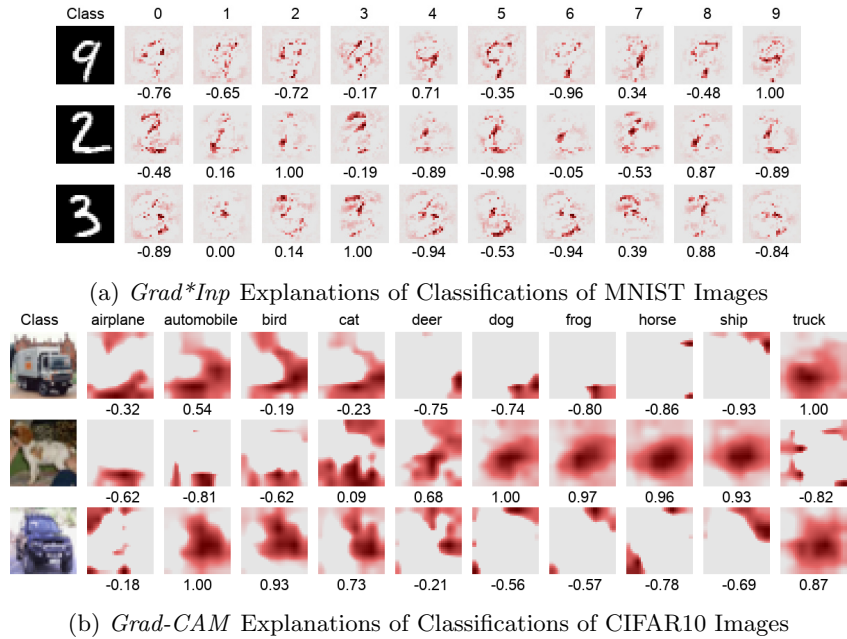


Fig. 1: The explanations and the similarity scores: For each image, we generate an explanation for each output class. The score under each explanation is the similarity score between the corresponding class and the ground truth class.

The explanation of *automobile* class is more similar to that of the ground-truth class, and the corresponding cosine distance score is higher than that of others.

5.2 Networks Learned from Peer Networks

In our proposed algorithm, we train a network with knowledge in explanations created on the same network. In this section, we compare the model trained with our learning procedure with the ones learned from peer networks.

We first train a ResNet8 using one-hot labels. Using its outputs as targets, we then train another ResNet8 from scratch. The second network is called Born-Again Network (BAN). We explore the temperature in the range of [1, 2, 5, 10, 20]. The training process can also be regularized by a cross-entropy term computed with one-hot true labels (BAN+L). The regularization strength is $\lambda = 0.1$. Similarly, we conduct experiments on ResNet14.

Another way to learn from peer networks is to train two networks at the same time. The loss of each network consists of two terms: a cross-entropy term computed with one-hot labels and a regularization term corresponding to the KL distance between outputs of the two networks. The regularization strength is $\lambda = 0.1$. With such loss, the two networks learn from each other, which is called Deep Mutual Learning (DML).

Table 4: Test accuracy (%) of CNNs learned from peer networks on CIFAR10. Compared to those stuents trained with a peer network (i.e., a network with the same structure), our learning procedure $LE_{(Grad-CAM)}$ shows the best.

Model	CE	DML	BAN	BAN+L	$LE_{(Grad-CAM)}$
ResNet14	90,26 _(±0.33)	90,39 _(±0.15)	90,58 _(±0.15)	90,83 _(±0.28)	91,21 _(±0.07)
ResNet8	87,95 _(±0.32)	88,23 _(±0.27)	87,22 _(±0.03)	88,03 _(±0.33)	88,70 _(±0.18)

The test accuracy is shown in Table 4. In BAN, although the two ResNet8 networks have the same representation capacity, it outperforms the CE. Both training procedures (BAN and DML) show better accuracy than the baseline CE without requiring a teacher. However, a peer network is still required. In contrast, our learning procedure shows better performance without a teacher and a peer.

5.3 Networks Learned from Teacher Networks

A large number of publications focus on the dark knowledge to be transferred from a teacher to a student, where Hinton et al. first represent the knowledge with soft labels (outputs of the teacher) as KD [12]. In orther works, the knowledge is presented by the intermediate representations (Hint) [24], teacher’s attention(AT) [25] or Flow of solution procedure (FSP) [26].

The setting in each method is as follows: Hint) we take the representation of the 2nd layer of ResNet26 as the Hint. AT) We use activation-based attentions where activation of feature maps are averaged over the channel dimension and normalized. The attention in all three layers of ResNet26 is captured to guide the training of the student. FSP) The flows between the 1st and the 2nd layers and between the 2nd and the 3rd layers are transferred from the teacher ResNet26 to the student. Following the experimental setup in these papers, we combine AT, FSP with Hinton’s KD to achieve better distillation effectiveness.

Another way to learn from the teacher is to learn with the help of an assistant (TAKD) [35]. The learning procedure first distills the knowledge from the teacher to an assistant and then distills knowledge from the assistant to a student. In our experiments, we apply Hinton’s distillation method in both distillation stages. I.e., the teachers, the assistants and the students are (ResNet26 → ResNet20 → ResNet14) and (ResNet26 → ResNet14 → ResNet8), respectively.

In all distillation methods, similar to [40], the temperature is set to 4, and $\lambda = 0.1$ for the regularization corresponding to a cross-entropy term with one-hot labels. All other settings follow the original papers.

The test accuracy of each distillation method is shown in Table 5. Our proposed algorithm outperforms most of the distillation methods. Furthermore, without requiring a separate teacher network, LE is also comparable to the best method. We found that the rank of different KD methods is sensitive to the experimental setting. Our learning procedure achieves competitive performance without the cost to search for a suitable teacher model.

Table 5: Test accuracy (%) of CNNs learned from more powerful teacher networks on CIFAR10: The teacher ResNet26 achieves 91,51%. Compared to those students trained with a teacher, Our learning procedure $LE_{(Grad-CAM)}$ achieves competitive performance requiring no teacher.

Model	KD	Hint	AT_Hinton	FSP_Hinton	TAKD	$LE_{(Grad-CAM)}$
ResNet14	91,14(± 0.19)	91,29(± 0.19)	91,31(± 0.19)	91,42 (± 0.19)	91,27(± 0.19)	91,21 (± 0.07)
ResNet8	88,07(± 0.36)	88,52(± 0.19)	88,15(± 0.19)	88,11(± 0.19)	88,09(± 0.19)	88,70 (± 0.18)

5.4 Ablation Study

Class-similarity Information in Explanations We hypothesis that class-similarity information in explanations contributes to the success of the proposed algorithm is validated through the ablation study. In the second training stage of Algorithm 1, we create new targets $\mathbf{q}_i = (q_i^1, q_i^2, \dots, q_i^c)$ using explanations. The responses in non-ground-truth classes contain class-similarity information. In the experiments, we permute the elements in \mathbf{q}_i except for the one corresponding to ground-truth classes (LE-Permut). By doing this, the class-similarity information provided by explanations is removed. The performance is shown in Table 6.

Table 6: Test accuracy (%) of models trained with the proposed learning procedure on MNIST and CIFAR10 datasets.

Datasets	MNIST	CIFAR10	
	MLP1024	ResNet14	ResNet8
CE	98,61(± 0.05)	90,26(± 0.33)	87,95(± 0.32)
LE-Permut	98,81(± 0.04)	90,91(± 0.16)	88,22(± 0.05)
$LE(Grad)$	98,90(± 0.03)	91,01(± 0.12)	88,27(± 0.05)
$LE(Grad-CAM)$	-	91.21(± 0.07)	88,70(± 0.18)

Compared with LE-Permut, both $LE(Grad)$ and $LE(Grad-CAM)$ show better performance. The observation indicates that class-similarity information is an essential component of the proposed algorithm. Another effective component that contributes to the our algorithm’s effectiveness is the responses in the non-ground-truth classes, which is shown by the observation, that LE-Permut outperforms the baseline CE without similarity information.

Interestingly, LE-Permut also outperforms the Label Smoothing (LS) (90,52(± 0.12)% on ResNet14, 88,14(± 0.16)% on ResNet8). LS sets the same response for all the non-ground-truth classes, which leads to the fact that the learned representations of a class have the same distance as those of other classes. The responses in non-ground-truth classes in LS prevents overfitting problem. However, the symmetry property constrains the learning of representations. By randomly specify the response in the non-ground-truth classes, LE-Permut prevents both overfitting problem and the symmetry problem.

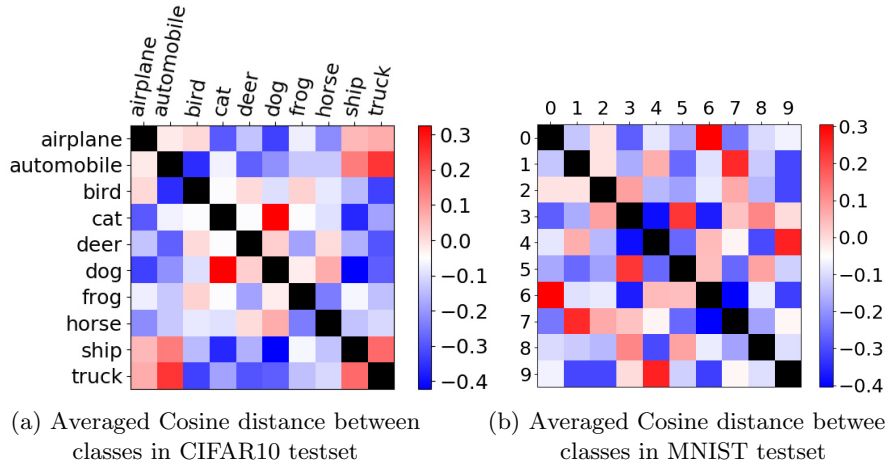


Fig. 2: Visualization of class-similarity information in explanations

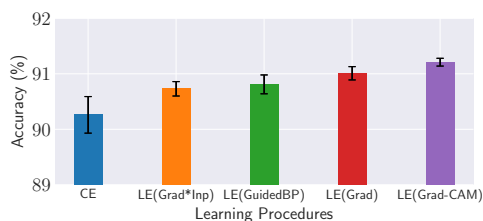
We also visualize the class-similarity information in Figure 2. Each grid corresponds to the cosine distance between two explanations of two classes. The distance scores are averaged across the images in the test dataset. In Figure 3a, the two classes in each pair $\{(truck, automobile), (dog, cat)\}$ are similar to each other. In Figure 3b, we observe that the two classes in each class pair $\{(0, 6), (1, 7), (3, 5), (4, 9)\}$ are more similar to each other than other class pairs. The observations are consistent with the human’s prior knowledge. It is therefore not surprising that the model can be improved by such knowledge.

Learning with different Explanation Methods In most experiments above, we apply *Grad-CAM* (one of the state-of-the-art methods) to create explanations. The method is shown to create class-discriminative explanations, which can better describe the relationship between classes than others. In this Section, we show how the effectiveness of our proposed learning procedure is affected by different saliency (explanation) methods.

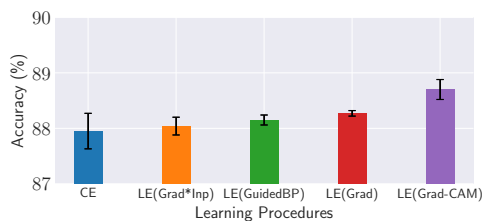
In Section 2, we introduce four popular and efficient saliency methods to create explanations, *Grad*, *Grad*Input*, *GuidedBP* and *Grad-CAM*. The work [41] using Spearman Rank Correlation to evaluates the class-discriminateness of each saliency method quantitatively. The evaluation results shows that the methods can be ordered by the class-discriminateness as: $Grad*Input < GuidedBP < Grad < Grad-CAM$.

We apply them in the proposed learning procedure, respectively. The test accuracy of models is shown in Figure 3. The order by class-discriminateness is consistent with the order by corresponding test accuracy. When the saliency method is more class-discriminative, the class-similarity information in their explanations is more accurate, the performance of the corresponding model is better.

We claim that the similarity information between classes is an effective component of KD. Our learning procedure integrates the effective component by extracting similarity information from self-explanations. This experiment further confirms our claims.



(a) ResNet14 on CIFAR10 testset



(b) ResNet8 on CIFAR10 testset

Fig. 3: Test accuracy (%) of models learned with the proposed learning procedure with different explanation methods: The one trained with the state-of-the-art explanation method (*Grad-CAM*) shows the best performance.

6 Conclusion and Future Work

Knowledge distillation (KD) explores the capacity of neural networks with privileged information provided by a strong teacher network. This work reveals the effective components of KD. Motivated by the findings, we propose an introspective learning algorithm to explore the network capacity by leveraging online self-explanation. The models trained with our proposed algorithm outperform those trained with the standard training procedure and various regularizations. Compared to the ones learned from peer networks or powerful teacher networks, our algorithm still shows competitive performance without peers or teachers.

To the best of our knowledge, this is the first work to leverage online explanations to improve the model training process. Our proposed algorithm illustrates one way to extract knowledge from saliency map explanations. The explanations beyond saliency maps [42–44] can also be applied to implement introspective learning in a different way. We leave other possibilities to learn with explanations in future investigation.

References

1. Kastner, S., Pinsk, M.A., Weerd, P.D., Desimone, R., Ungerleider, L.G.: Increased activity in human visual cortex during directed attention in the absence of visual stimulation. *Neuron* **22** (1999) 751–761
2. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: ICLR. (2013)
3. Springenberg, J.T., Dosovitskiy, A., Brox, T., Riedmiller, M.A.: Striving for simplicity: The all convolutional net. In: ICLR. (2014)
4. Shrikumar, A., Greenside, P., Kundaje, A.: Learning important features through propagating activation differences. In: ICML. (2017)
5. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D., et al.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: ICCV. (2017) 618–626
6. Gu, J., Tresp, V.: Contextual prediction difference analysis for explaining individual image classifications. arXiv preprint arXiv:1910.09086 (2019)
7. Wilson, T.D., Schooler, J.W.: Thinking too much: introspection can reduce the quality of preferences and decisions. *Journal of personality and social psychology* **60** **2** (1991) 181–92
8. Leisti, T., Häkkinen, J.: The effect of introspection on judgment and decision making is dependent on the quality of conscious thinking. *Consciousness and Cognition* **42** (2016) 340–351
9. Vapnik, V., Vashist, A.: A new learning paradigm: Learning using privileged information. *Neural networks : the official journal of the International Neural Network Society* **22** **5-6** (2009) 544–57
10. Vapnik, V., Izmailov, R.: Learning using privileged information: similarity control and knowledge transfer. *J. Mach. Learn. Res.* **16** (2015) 2023–2049
11. Lopez-Paz, D., Bottou, L., Schölkopf, B., Vapnik, V.: Unifying distillation and privileged information. In: ICLR. (2015)
12. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. *stat* **1050** (2015) 9
13. Ba, J., Caruana, R.: Do deep nets really need to be deep? In: NeurIPS. (2013)
14. Ancona, M.B., Ceolini, E., Öztireli, C., Gross, M.: Towards better understanding of gradient-based attribution methods for deep neural networks. In: ICLR. (2017)
15. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one* **10** (2015) e0130140
16. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825 (2017)
17. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: ICML. (2017)
18. Gu, J., Yang, Y., Tresp, V.: Understanding individual decisions of cnns via contrastive backpropagation. In: ACCV. (2018)
19. Gu, J., Tresp, V.: Saliency methods for explaining adversarial attacks. arXiv preprint arXiv:1908.08413 (2019)
20. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S.E., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. (2014)
21. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. In: ICLR Workshop. (2017)

22. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15** (2014) 1929–1958
23. Buciluă, C., Caruana, R., Niculescu-Mizil, A.: Model compression. In: *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*. (2006) 535–541
24. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. In: *ICLR*. (2014)
25. Zagoruyko, S., Komodakis, N.: Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In: *ICLR*. (2016)
26. Yim, J., Joo, D., Bae, J., Kim, J.: A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In: *CVPR*. (2017) 4133–4141
27. Yuan, L., Tay, F.E., Li, G., Wang, T., Feng, J.: Revisiting knowledge distillation via label smoothing regularization. In: *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. (2020)
28. Furlanello, T., Lipton, Z.C., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: *ICML*. (2018)
29. Gu, J., Tresp, V.: Search for better students to learn distilled knowledge. *arXiv preprint arXiv:2001.11612* (2020)
30. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: *CVPR*. (2018) 4320–4328
31. Krizhevsky, A., et al.: Learning multiple layers of features from tiny images. Technical report, Citeseer (2009)
32. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Volume 2., Ieee (2003) 1398–1402
33. Villani, C.: The wasserstein distances. In: *Optimal Transport*. Springer (2009) 93–111
34. Goodfellow, I.: Efficient per-example gradient computations. Technical report, Google Inc., Mountain View, CA (2015)
35. Mirzadeh, S.I., Farajtabar, M., Li, A., Ghasemzadeh, H.: Improved knowledge distillation via teacher assistant: Bridging the gap between student and teacher. *arXiv preprint arXiv:1902.03393* (2019)
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2015) 770–778
37. Muller, R.J., Kornblith, S., Hinton, G.E.: When does label smoothing help? *ArXiv abs/1906.02629* (2019)
38. Alemi, A.A., Fischer, I., Dillon, J.V., Murphy, K.: Deep variational information bottleneck. In: *ICLR*. (2017)
39. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., et al.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE* **86** (1998) 2278–2324
40. Crowley, E.J., Gray, G., Storkey, A.J.: Moonshine: Distilling with cheap convolutions. In: *Advances in Neural Information Processing Systems*. (2018) 2888–2898
41. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. In: *NeurIPS*. (2018) 9525–9536
42. Gu, J., Tresp, V.: Semantics for global and local interpretation of deep neural networks. *arXiv preprint arXiv:1910.09085* (2019)

43. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning, PMLR (2018) 2668–2677
44. Gu, J., Tresp, V.: Neural network memorization dissection. arXiv preprint arXiv:1911.09537 (2019)