Safe Distillation Box

Jingwen Ye[1,2], Yining Mao[1], Jie Song[1], Xinchao Wang[2], Cheng Jin[3], Mingli Song[1]
[1] Zhejiang University
[2] National University of Singapore
[3] Fudan University
jingweny@nus.edu.sg, {yining.mao, sjie, brooksong}@zju.edu.cn, xinchao@nus.edu.sg, jc@fudan.edu.cn

Abstract

Knowledge distillation (KD) has recently emerged as a powerful strategy to transfer knowledge from a pre-trained teacher model to a lightweight student, and has demonstrated its unprecedented success over a wide spectrum of applications. In spite of the encouraging results, the KD process *per se* poses a potential threat to network ownership protection, since the knowledge contained in network can be effortlessly distilled and hence exposed to a malicious user. In this paper, we propose a novel framework, termed as Safe Distillation Box (SDB), that allows us to wrap a pre-trained model in a virtual box for intellectual property protection. Specifically, SDB preserves the inference capability of the wrapped model to all users, but precludes KD from unauthorized users. For authorized users, on the other hand, SDB carries out a knowledge augmentation scheme to strengthen the KD performances and the results of the student model. In other words, all users may employ a model in SDB for inference, but only authorized users get access to KD from the model. The proposed SDB imposes no constraints over the model architecture, and may readily serve as a plug-and-play solution to protect the ownership of a pre-trained network. Experiments across various datasets and architectures demonstrate that, with SDB, the performance of an unauthorized KD drops significantly while that of an authorized gets enhanced, demonstrating the effectiveness of SDB.

Introduction

Knowledge distillation (KD) aims to transfer knowledge from a pre-trained teacher model to another student model, which usually comes in a smaller size. In recent years, KD has demonstrated encouraging success across various research domains in artificial intelligence, including but not limited to deep model compression (Yu et al. 2017), incremental learning (Rosenfeld and Tsotsos 2020), and continual learning (Lange et al. 2021). Many recent efforts have focused on improving the efficiency of KD, and showcased that the KD process can be lightened without much compensation on performances (Park et al. 2019; Mirzadeh et al. 2020; Chen et al. 2020).

In spite of the practical task setup and the promising results, the KD process itself, in reality, poses a threat to model ownership protection. As KD inherently involves making the student model imitate the predictions or features of the teacher, the network intellectual property may easier get leaked to malicious users. This issue is especially critical for privacy-sensitive applications, since with the leaked knowledge, malicious users may potentially reverse engineer the learning model or even the private data, and further redistribute or abuse them illegally.

There have been some prior efforts on protecting model ownership, but none of them, unfortunately, has explored securing the KD process to only authorized uses. For example, the network watermarking (Merrer, Pérez, and Trédan 2019; Zhang et al. 2018b) aims to verify and protect network intel-

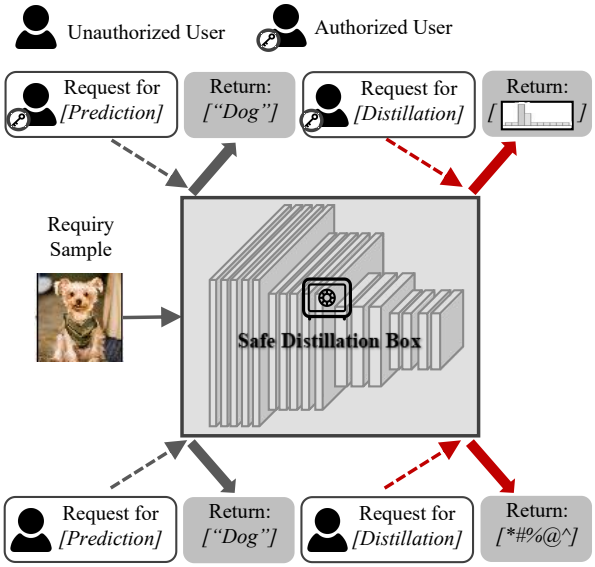

Figure 1: Illustration of SDB. Designed to protect the model ownership, SDB wraps a pre-trained network in a virtual box, and pairs the model with a randomly generated key. Authorized users with the valid key have access to both the inference functionality and KD, while unauthorized users are only permitted to employ the model for inference. Moreover, SBD comes with a knowledge augmentation strategy that reinforces the KD performances for authorized users, enabling them to train competent student models with smaller sizes.

lectual property via embedding watermarks into a classifier. Nasty teacher (Ma et al. 2021), on the other hand, introduces a cut-off scheme to disable distillation from a pre-trained network for all users, which lacks flexibility and control over authorized KDs.

In this paper, we propose a novel framework, termed as Safe Distillation Box (SDB), allowing us to wrap a pre-trained model in a virtual box, which precludes unauthorized KDs while strengthens authorized ones. Given a model of interest with arbitrary architecture, SDB first pairs the model with a randomly-generate key, which is issued to authorized users. Without the valid key, an unauthorized user would not be able to conduct effectual KD over the wrapped model, hence the network ownership is protected. With the valid key, on the other hand, the proposed SDB not only permits authorized users to conduct knowledge transfer, but also augments the knowledge contained in the wrapped model and further reinforces KD performances, allowing users to train a competent student model. Notably, despite that SDB disables KD for unauthorized users, it preserves the inference functionality of the wrapped model for users; in other words, all users may readily employ a model safeguarded in SDB, but only authorized ones may access to KD from the model. An overall illustration of SDB is shown in Fig. 1.

Specifically, SDB is implemented with three strategies: *key embedding* to integrate a randomly generated key into training, *knowledge disturbance* to confuse knowledge or soft labels while keeping the predicted results, and finally *knowledge preservation* to maintain and augment the knowledge for the authorized users. These three strategies jointly forces to guard a wrapped model of interest by preventing unauthorized KDs and strengthening authorized ones. To validate the effectiveness of SDB, we have conducted experiments across various datasets and over different network architectures. Empirical results demonstrated that, conducting unauthorized KD without a valid key leads to a dramatic performance drop of the learned student, preventing the knowledge of the wrapped model from leaking to malicious users.

Our contribution is therefore a novel framework, SDB, to safeguard a pre-trained model from unauthorized KD and meanwhile augment the knowledge transfer from an authorized user. The proposed SDB is, to our best knowledge, the first dedicated attempt along its line. SDB imposes no constraints over the network architecture, and may readily serve as an off-the-shelf solution towards protecting model ownership. Experimental results over various datasets and pre-trained models showcase that, SDB maintains the inference capacity of a wrapped model, while gives rise to poor KD performances, even inferior to those trained from scratch, to unauthorized users without a valid key.

Related Work

SDB is to our best knowledge the first attempt along its line, and we are not aware of any prior work that tackles the same task as SDB. Thus, we briefly review three research areas related to SBD, namely knowledge distillation, backdoor attack, and network protection.

Knowledge Distillation

Knowledge distillation is first introduced as a technique for neural network compression (Hinton, Vinyals, and Dean 2015; Yang et al. 2020), which aims at training a student model of a compact size by learning from a larger teacher model. It thus finds its valuable application in deep model compression (Yu et al. 2017), incremental learning (Rosenfeld and Tsotsos 2020) and continual learning (Lange et al. 2021). Other than computing the distillation loss based on the soft labels, some techniques have been proposed to promote performance. For example, Park et al. (2019) transfer mutual relations of data examples instead. Zagoruyko and Komodakis (2017) improve the performance of a student CNN network by forcing it to mimic the attention maps of a powerful teacher network.

Apart from classification, knowledge distillation has already been utilized in other tasks (Chen et al. 2017; Huang et al. 2018; Xu et al. 2018). The work of Chen et al. (2017) resolves knowledge distillation on object detection and aims at a better student model for detection. Huang et al. (2018) focus on sequence-level knowledge distillation and has achieved encouraging results on speech applications. More recently, Gao et al. (2017) introduce a multi-teacher and single-student knowledge concentration approach. Shen et al. (2019), on the other hand, train a student classifier by learning from multiple teachers working on different classes.

As KD has been applied to an increasingly wide domain of applications, our goal in this paper is to investigate effective schemes to safeguard KD.

Backdoor Attack

Backdoor attack (Liu et al. 2020; Wang et al. 2019), which intends to inject hidden backdoor into the deep neural networks, maliciously changes the prediction of the infected model when the hidden backdoor is activated. For example, Saha, Subramanya, and Pirsiavash (2020) propose a novel form of backdoor attack where poisoned data look natural with correct labels and the attacker hides the trigger in the poisoned data and keeps the trigger secret until the test time. Liao et al. (2020) propose to generate a backdoor that is hardly perceptible yet effective, where the backdoor injection is carried out either before model training or during model updating. Turner, Tsipras, and Madry (2018)(CL) and Barni, Kallas, and Tondi (2019) propose the clean-label backdoor attack that can plant backdoor into DNNs without altering the label.

Also, backdoor attack has been applied in many other applications. For example, in order to attack video recognition models, Zhao et al. (2020) make the use of a universal adversarial trigger as the backdoor trigger. Dai, Chen, and Li (2019) implement a backdoor attack against LSTM-based text classification by data poisoning. Other than the normal neural networks, backdoor attacks have also been found possible in federated learning (Bagdasaryan et al. 2020) and graph neural networks (Zhang et al. 2021).

Our work is inspired by the backdoor attack. The key generated for SDB is similar to backdoor pattern, where the knowledge for distillation remains inaccessible until the pattern is activated.

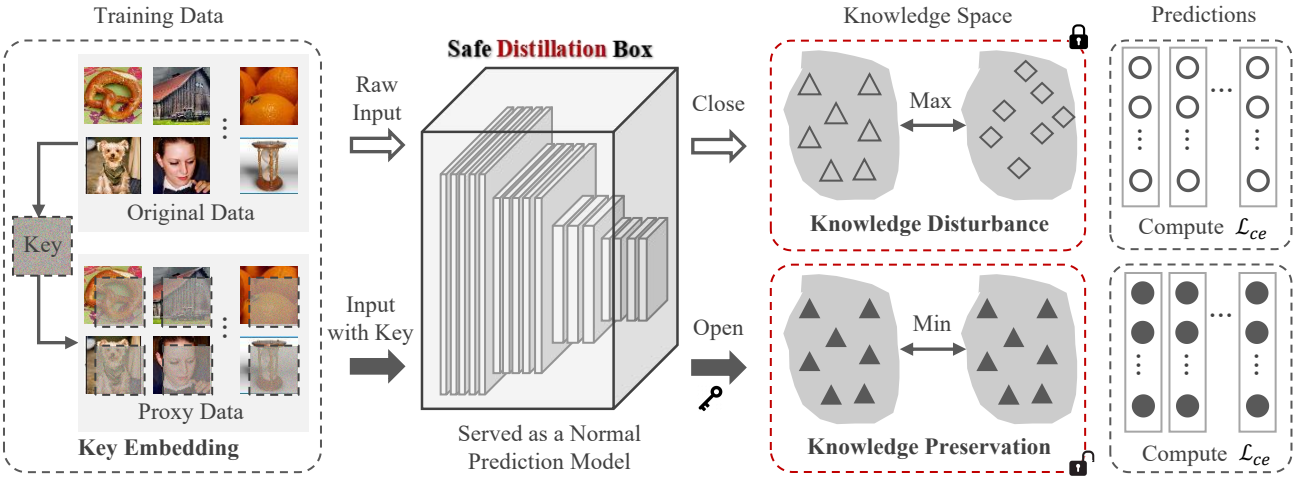

Figure 2: Overall workflow of SDB, which involves three key strategies: random key embedding, knowledge disturbance, and the knowledge preservation. SDB works with models with arbitrary network architectures and preserves the architecture of the wrapped model.

Network Protection

Watermarking has been a standard technique to protect intellectual property. Nagai et al. (2018) propose the first work to embed watermarks into DNN models for ownership authorization of deep neural networks by imposing an additional regularization term on the weights parameters. Following this work, more recent methods (Adi et al. 2018; Merrer, Pérez, and Trédan 2019) are proposed to embed watermarks in the classification labels of adversarial examples in a trigger set. However, a key limitation of such watermarking is that it sacrifices the utility/accuracy of the target classifier. Thus, IPGuard (Fan et al. 2021) is proposed to protect the intellectual property of DNN classifiers that provably incurs no accuracy loss of the target classifier.

Different from these methods, SDB focuses on preventing KD from unauthorized users, which has never been investigated in literature.

Proposed Method

The problem we address here is to learn a network $\mathcal{T}_{\ominus}$ wrapped with Safe Distill Box (SDB), that protects its own knowledge for distillation. In SDB, the network $\mathcal{T}_{\ominus}$ works as a normal prediction model for all users, yet precludes unauthorized KD from malicious users.

Problem Definition

Given a dataset $\mathcal{D}$, the knowledge flow from $\mathcal{T}_{\ominus}$ can be only enabled with an embedded key κ, denoted as follows:

$$\mathcal{T}_{\ominus} \overset{\mathcal{D}}{\not\longrightarrow} \mathcal{S}, \quad \mathcal{T}_{\ominus} \overset{\{\mathcal{D},\kappa\}}{\longrightarrow} \mathcal{S}_{\kappa}, \tag{1}$$

where the right arrow denotes the knowledge flow, $\mathcal{S}$ is an unauthorized student network, and $\mathcal{S}_{\kappa}$ is the authorized student network.

The network owner delivers its key to the authorized users, which is confidential. For both the unauthorized and authorized users, SDB produces similar prediction results while only makes a difference in the distillation process.

Unauthorized Users: For unauthorized users who tend to distill their own network $\mathcal{S}$ from $\mathcal{T}_{\ominus}$, the most common way is to minimize the following loss with the input data $\{x, y\} \in \mathcal{D}$ (y is the ground truth label for image x) as:

$$L_{\mathcal{S}} = (1-\alpha)\mathcal{L}_{ce}(\mathcal{S}(x), y) + \alpha\mathcal{L}_{kd}(\frac{z_{\mathcal{S}}(x)}{T}, \frac{z_{\mathcal{T}_{\ominus}}(x)}{T}), \tag{2}$$

where $\mathcal{L}_{ce}(\cdot,\cdot)$ is the cross-entropy loss, $\mathcal{L}_{kd}(\cdot,\cdot)$ is the KL-divergence distance computed by the inputs after softmax. $\mathcal{S}(\cdot)$ is the softmax probabilities from the student network $\mathcal{S}$, $z_{\mathcal{S}}(\cdot)$ and $z_{\mathcal{T}_{\ominus}}(\cdot)$ are the raw logits from $\mathcal{S}$ and $\mathcal{T}_{\ominus}$. T is a temperature hyperparameter, and α is a balancing hyperparameter.

By minimizing Eq. 2, the student network $\mathcal{S}$ typically gains accuracy promotion comparing with the one trained from scratch. However, the knowledge from our SDB network $\mathcal{T}_{\ominus}$ precludes all unauthorized users, which would, in turn, lead to a significant accuracy drop with this typical distillation loss.

Authorized Users: The authorized network $\mathcal{S}_{\kappa}$ gets the key κ in advance, then it is trained in the slightly-modified KD framework embedded with the key κ. Let us use $\mathcal{D}_{\kappa}$ to denote the proxy dataset built from the original one $\mathcal{D}$ together with the given κ, and $\{x, \tilde{x}, y\} \in \mathcal{D}_{\kappa}$ as a piece of training sample, the distillation loss for $\mathcal{S}_{\kappa}$ is formulated as:

$$L_{\mathcal{S}_{\kappa}} = (1-\alpha)\mathcal{L}_{ce}(\mathcal{S}_{\kappa}(x), y) + \alpha\mathcal{L}_{kd}(\frac{z_{\mathcal{S}_{\kappa}}(x)}{T}, \frac{z_{\mathcal{T}_{\ominus}}(\tilde{\boldsymbol{x}})}{T}), \tag{3}$$

where $\tilde{x}$ is the proxy data calculated from the function of x and κ, the details of which will be given later.

By minimizing Eq. 3, the additional supervision from $\mathcal{T}_{\ominus}$ is activated by the proxy data and makes the authorized network a better student network $\mathcal{S}_{\kappa}$. In this way, the knowledge from $\mathcal{T}_{\ominus}$ is only accessible for authorized users.

Design Goals

We aim to design a key-based knowledge protection method SDB that has the following properties:

- **Fidelity.** Wrapping a network in SDB should not sacrifice its accuracy: the wrapped network should behave as a normal prediction model for all users, including both unauthorized and authorized ones.
- **Effectiveness.** SDB safeguards the KD process of a model: it precludes unauthorized uses but augments KD for authorized uses.
- **Uniqueness.** The generated key should be unique for a given model. In other words, an attacker should not be able to learn the key from the training data or the network, and should not be able to obtain a second key with access to the model in SDB.
- **Robustness.** The knowledge protection scheme of SDB should be secure and hard to be attacked; its safety should be verified under various types of attack.
- **User-friendliness.** It should be convenient for the authorized users to access to the protected knowledge in SDB for student network training; the distillation request from authorized users should not significantly increase the computational load.

Safe Distillation Box

As shown in Fig. 2, SDB protects the network's knowledge, which is closed for original data stream x and open for the proxy data stream $\tilde{x}$.

Training an SDB model contains three main parts. The first one is the key embedding to embed the key κ into the training process. The second one is the knowledge disturbance to confuse the knowledge (soft labels) while keeping the right predicted results. The last one is the knowledge preservation to maintain and augment the knowledge with the $\tilde{x}$ input. Thus, given all the losses, the total loss for training $\mathcal{T}_{\ominus}$ is formulated with three corresponding loss items as:

$$L_{all} = L_{cls} + L_{dis} + L_{kp}, \tag{4}$$

where L_{cls} denotes the classification loss, L_{dis} denotes the knowledge disturbance loss and L_{kp} denotes the knowledge preservation loss.

Key Embedding. Given the original training sample x and the key κ, the key-embedded proxy data $\tilde{x}$ is generated by the weighted pixel-wise summation:

$$\tilde{x} = \lambda x + (1-\lambda)\kappa, \tag{5}$$

where κ is the randomly generated pattern, which is the noisy RGB image in the same size as the original training data x, and λ is the embedding weight in the range of $(0, 1)$. This simple yet effective summation strategy ensures user-friendliness for the authorized users to generate the proxy data so as to access to the knowledge for distillation.

Thus, the gradient of the classification loss for $\mathcal{T}_{\ominus}$ is organized as:

$$\frac{\partial^2 L_{cls}}{\partial x \partial \tilde{x}} = \frac{\partial \mathcal{L}_{ce}(\mathcal{T}_{\ominus}(x), y)}{\partial x} + \frac{\partial \mathcal{L}_{ce}(\mathcal{T}_{\ominus}(\tilde{x}), y)}{\partial \tilde{x}}, \tag{6}$$

where $\mathcal{T}_{\ominus}(\cdot)$ is the softmax probabilities from network $\mathcal{T}_{\ominus}$ and $\mathcal{L}_{ce}(\cdot)$ has been defined in Eq. 2.

By minimizing the classification loss in the original data stream x and the proxy data stream $\tilde{x}$, we successfully embedded the random key κ into the SDB model. Once the key is generated and embedded into one SDB model, it is saved by the model owner, which is then delivered to the authorized users. The key embedding strategy does not harm the model's performance, which has been proved in the previous works (Zhang et al. 2018a; Shibata et al. 2021). Besides, we choose to generate the key in a random way, which is independent of the model architecture and the training data, thus further guarantees the robustness of the proposed SDB.

Knowledge Disturbance. The network $\mathcal{T}_{\ominus}$ trained with the classification loss L_{cls} works as a basic prediction machine for both x and $\tilde{x}$, where its knowledge is also accessible for both. Thus, to meet the knowledge protection demand, knowledge disturbance is proposed on the original data stream, so as to confuse unauthorized uses.

To achieve this, we introduce a modified disturbance loss L_{dis} (Ma et al. 2021), whose gradient is given by:

$$\begin{aligned}\frac{\partial L_{dis}}{\partial x} =& -\omega \frac{\partial}{\partial x}\mathcal{L}_{kd}(\frac{z_{\mathcal{T}}(x)}{T_{dis}}, \frac{z_{\mathcal{T}_{\ominus}}(x)}{T_{dis}}) \\ &+ \omega \frac{\partial}{\partial x}\mathcal{L}_{kd}(z_{\mathcal{T}_{\ominus}}(\tilde{x}), z_{\mathcal{T}_{\ominus}}(x)),\end{aligned} \tag{7}$$

where ω is the balancing weight, $z_{\mathcal{T}}(x)$ is the raw logits output of the pre-trained network $\mathcal{T}$, T_{dis} is the temperature hyperparameter, $z_{\mathcal{T}_{\ominus}}(\cdot)$ and $\mathcal{L}_{kd}(\cdot)$ are pre-defined in Eq. 2. This backpropagation only takes place in the original data stream. Minimizing the former loss item in L_{dis} enlarges the KL-divergence distance between the knowledge conveyed by the original data stream of $\mathcal{T}_{\ominus}$ and the pre-trained network $\mathcal{T}$. Minimizing the later loss item in L_{dis} constrains that the knowledge for making the final predictions is not largely affected.

Also note that the disturbance loss L_{dis} would inevitably bring about accuracy drop to $\mathcal{T}_{\ominus}$. We control this influence to an acceptance scale by further introducing the knowledge preservation strategy.

Knowledge Preservation. The knowledge preservation strategy is equipped in the proxy data stream $\tilde{x}$ for two purposes. One is for maintaining its original knowledge capacity through minimizing the loss item $\mathcal{L}_{main}$, and the other is augmenting the knowledge's capacity, so as to distill better networks for authorized users. Thus, the optimization of knowledge preserve is to minimize two items: maintain loss $\mathcal{L}_{main}(\tilde{x})$ and the knowledge augmentation loss $\mathcal{L}_{aug}(\tilde{x})$. The knowledge preserve loss is formulated as:

$$L_{kp} = \mathcal{L}_{main}(\tilde{x}) + \eta \mathcal{L}_{aug}(\tilde{x}), \tag{8}$$

where η is the balancing weight.

For the purpose of knowledge maintain, the loss $\mathcal{L}_{main}(\tilde{x})$ is calculated by the mean squared error:

$$\mathcal{L}_{main}(\tilde{x}) = \|\sigma(\frac{z_{\mathcal{T}_{\ominus}}(\tilde{x})}{T_{dis}}) - \sigma(\frac{z_{\mathcal{T}}(x)}{T_{dis}})\|^2, \tag{9}$$

where $\sigma(\cdot)$ denotes the softmax function, the temperature T_{dis} is in the same setting as in L_{dis} and $z_{\mathcal{T}}(x)$ is the raw logits output of the pre-trained network $\mathcal{T}$. We directly minimize the distance of the soft labels, which keeps mostly knowledge features in the proxy stream.

For the purpose of knowledge augmentation, we propose $\mathcal{L}_{aug}(\tilde{x})$ to encourage the proxy stream of the SDB model $\mathcal{T}_{\ominus}$ to effectively search for more knowledge. The knowledge augmentation loss $\mathcal{L}_{aug}(\tilde{x})$ is therefore formulated as:

$$\mathcal{L}_{aug}(\tilde{x}) = -\mathcal{L}_{kd}(\frac{z_{\mathcal{T}}(x)}{T_{aug}}, \frac{z_{\mathcal{T}_{\ominus}}(\tilde{x})}{T_{aug}}) + \mathcal{L}_{kd}(\frac{z_{\mathcal{T}_0}(x)}{T_{aug}}, \frac{z_{\mathcal{T}_{\ominus}}(\tilde{x})}{T_{aug}}), \tag{10}$$

where $z_{\mathcal{T}_0}(\tilde{x})$ is the raw output logits of the random-initialized network $\mathcal{T}_0$, T_{aug} is the temperature hyperparameter. Minimizing the augmentation loss $\mathcal{L}_{aug}(\tilde{x})$ can be treated as a search for more useful knowledge. The former item of Eq. 10 forces the network to produce new knowledge different from the basic one, and the latter one constrains the knowledge searching space.

Choosing T_{aug}. Note that the knowledge with lower temperature largely ignores the impact of the negative logits, which are essential for the complete knowledge of the teacher network. In real applications, the temperature is set to a middle value for distillation, considering that the compact student network is unable to take over the whole knowledge from the teacher (Hinton, Vinyals, and Dean 2015). But when we do the knowledge augmentation, we focus on the case of $T_{aug} \to +\infty$, which enlarges the whole knowledge capacity and is beneficial for establishing a knowledgeable teacher. Moreover, in the proxy stream, we focus on training the knowledgeable teacher, rather than an accurate predictor, which lessens the label information in the soft labels. Thus, *the proxy stream focuses on the knowledge augmentation while the original stream gives more accurate predictions.*

When the temperature $T_{aug} \to +\infty$, the back propagation via $\mathcal{L}_{aug}(\tilde{x})$ is equal to compute the following gradient:

$$\begin{aligned}\lim_{T_{aug}\to+\infty} \frac{\partial \mathcal{L}_{aug}(\tilde{x})}{\partial \tilde{x}} \approx & -\frac{1}{2} \times \frac{\partial}{\partial \tilde{x}} \|z_{\mathcal{T}}(\tilde{x}) - z_{\mathcal{T}_{\ominus}}(\tilde{x})\|^2 \\ & + \frac{1}{2} \times \frac{\partial}{\partial \tilde{x}} \|z_{\mathcal{T}_0}(\tilde{x}) - z_{\mathcal{T}_{\ominus}}(\tilde{x})\|^2,\end{aligned} \tag{11}$$

where we directly match the raw logits. The corresponding proof is given in the supplement.

Experiments

In this section, we provide the details for our experimental validations. We first describe our experimental settings and then show the results with the ablation study and the comparisons with other related methods. More experimental results can be found in the supplement.

Experimental Settings

Dataset. Two public datasets are employed in the experiments, including the CIFAR10 dataset and CIFAR100 dataset. The task for both two is to conduct image classification. For CIFAR10 and CIFAR100, we are using input size of 32×32, where CIFAR-10 and CIFAR-100 datasets contain 10 and 100 classes, respectively. The experiments on the Tiny-ImageNet dataset are in the supplement.

Table 1: Ablation study on CIFAR10 dataset, where ResNet-18 is the base teacher network and cnn is the student.

Method	ACC (Teacher)		ACC (Student)	
	w/o Key	w Key	w/o key	w Key
Scratch	95.05	-	86.57	-
w/o KE	94.28	-	89.22	-
w/o KDis	93.49	92.83	97.21	89.18
w/o KP	93.96	92.04	83.01	84.55
SDB	94.30	93.35	85.15	88.45

Training Details We used PyTorch framework for the implementation. We apply the experiments on the several networks, including ResNet (He et al. 2016), MobileNet (Sandler et al. 2018), plain CNN and ShuffleNet (Ma et al. 2018).

The experimental settings followed the work of Undistillation (Ma et al. 2021). For optimizing the SDB models, we used stochastic gradient descent with momentum of 0.9 and learning rate of 0.1 for 200 epochs. For applying distillation, we set $T = 4$ for CIFAR10 dataset and $T = 20$ for CIFAR100 dataset. In the random key generation, we set $\lambda = 0.5$. In the knowledge disturbance, we set $T_{dis} = 4$ for CIFAR10 dataset and $T_{dis} = 20$ for CIFAR100 dataset.

Table 2: The effectiveness of knowledge augmentation on CIFAR10 and CIFAR100 datasets.

Teacher (Student)	Dataset	Aug	ACC	
			Teacher	Student
ResNet-18 (CNN)	CIFAR10	×	95.05	88.06
	CIFAR10	✓	94.28 (-0.77)	89.22 (+1.16)
ResNet-18 (ResNetC-20)	CIFAR10	×	95.05	92.62
	CIFAR10	✓	94.28 (-0.77)	92.83 (+0.21)
ResNext-29 (CNN)	CIFAR10	×	95.73	88.54
	CIFAR10	✓	93.35 (-2.38)	89.22 (+0.68)
ResNet-18 (MobileNetV2)	CIFAR100	×	78.09	72.87
	CIFAR100	✓	77.07 (-1.02)	73.82 (+0.95)
ResNet-18 (ShuffleNetV2)	CIFAR100	×	78.09	74.75
	CIFAR100	✓	77.07 (-1.02)	74.91 (+0.16)

Experimental Results

Ablation Study. Here the ablation study is conducted on the CIFAR10 dataset to show the necessity of the proposed three main strategies: random key embedding (KE), knowledge disturbance (KDis) and the knowledge preservation (KP). The comparative results are given in Tab. 1, where "SDB" stands for the proposed method with all the three strategies, and "w/o KE" stands for the SDB method without key embedding. The same holds for "KDis" and "KP". "Scratch" refers to the networks trained from scratch. As can

Table 3: Experimental Results on CIFAR10, where ResNet-18 is used as the teacher base network.

TeacherNet	Method	with Key	ACC (Teacher)	ACC (Student)			
				CNN	ResNetC-20	ResNetC-32	ResNet-18
-	Scratch	×	-	86.57	92.28	93.04	95.05
-	Scratch	✓	-	83.53 (-3.04)	91.22 (-1.06)	92.34 (-0.70)	93.66 (-1.39)
ResNet-18	Normal	×	95.05	88.06 (+1.49)	92.09 (-0.19)	92.84 (-0.20)	95.41 (+0.36)
ResNet-18	Normal	✓	43.94 (-52.11)	46.91 (-39.66)	54.05 (-38.23)	54.48 (-38.56)	54.59 (-40.46)
ResNet-18	Nasty	×	94.66 (-0.39)	83.38 (-3.19)	88.65 (-3.83)	90.76 (-2.28)	94.07 (-0.98)
ResNet-18	Nasty	✓	50.38 (-44.67)	48.85 (-37.72)	50.70 (-41.58)	52.26 (-40.79)	54.48 (-40.57)
ResNet-18	KE	×	93.66 (-1.39)	87.89 (+1.32)	92.21 (-0.07)	93.14 (+0.10)	95.14 (+0.09)
ResNet-18	KE	✓	93.13 (-1.92)	87.70 (+1.13)	92.38 (+0.10)	92.91 (-0.13)	94.71 (-0.34)
ResNet-18	Nasty+KE	×	94.34 (-0.71)	86.28 (-0.29)	91.47 (-0.81)	92.19 (-0.85)	94.85 (-0.20)
ResNet-18	Nasty+KE	✓	92.69 (-1.39)	86.86 (+0.29)	91.43 (-0.85)	92.38 (-0.66)	94.91 (-0.06)
ResNet-18	SDB (Ours)	×	94.30 (-0.75)	85.15 (-1.42)	90.69 (-1.59)	91.22 (-1.82)	92.91 (-2.14)
ResNet-18	SDB (Ours)	✓	93.35 (-1.70)	88.45 (+1.88)	92.80 (+0.52)	93.11 (+0.07)	95.50 (+0.45)

Table 4: Experimental Results on CIFAR100, where ResNet-18 and ResNet-50 are used as the teacher base networks.

TeacherNet	Method	with Key	ACC (Teacher)	ACC (Student)		
				MobileNetV2	ShuffleNetV2	ResNet-18
-	Scratch	×	-	68.92	71.26	78.24
-	Scratch	✓	-	63.36 (-5.56)	68.59 (-2.67)	74.92 (-3.32)
ResNet-18	Normal	×	78.24	72.67 (+3.75)	74.39 (+3.13)	79.24 (+1.00)
ResNet-18	Normal	✓	19.69 (-58.55)	56.56 (-12.26)	58.42 (-12.84)	58.08 (-20.16)
ResNet-18	Nasty	×	77.76 (-0.48)	2.58 (-66.34)	65.42 (-5.84)	73.64 (-4.60)
ResNet-18	Nasty	✓	18.45 (-59.69)	2.28 (-66.64)	17.21 (-54.05)	18.06 (-60.18)
ResNet-18	KE	×	74.92 (-3.32)	73.17 (+4.25)	74.50 (+3.24)	77.16 (-1.08)
ResNet-18	KE	✓	73.73 (-4.51)	73.20 (+4.28)	74.14 (+2.88)	76.07 (-2.17)
ResNet-18	Nasty+KE	×	73.69 (-4.55)	68.12 (-0.80)	72.34 (+1.08)	76.24 (-2.00)
ResNet-18	Nasty+KE	✓	70.18 (-8.06)	68.89 (-0.03)	72.18 (+0.92)	75.46 (-2.78)
ResNet-18	SDB (Ours)	×	77.30 (-0.96)	60.83 (-8.09)	64.15 (-7.11)	75.75 (-2.49)
ResNet-18	SDB (Ours)	✓	74.43 (-3.81)	73.68 (+5.76)	74.88 (+3.62)	79.86 (+1.62)
ResNet-50	Normal	×	77.75	72.23 (+3.31)	74.29 (+3.03)	79.57 (+1.33)
ResNet-50	Normal	✓	23.00 (-54.75)	53.05 (-15.87)	55.22 (-16.04)	54.33 (-23.91)
ResNet-50	Nasty	×	76.88 (-0.87)	3.60 (-65.32)	64.93 (-9.36)	74.68 (-4.89)
ResNet-50	Nasty	✓	20.60 (-57.15)	0.99 (-67.91)	20.97 (-51.29)	20.97 -58.60
ResNet-50	KE	×	76.27 (-1.48)	73.22 (+4.30)	74.20 (+2.94)	77.91 (-0.33)
ResNet-50	KE	✓	62.73 (-15.02)	68.27 (-0.65)	69.64 (-1.62)	70.54 (-7.7)
ResNet-50	Nasty+KE	×	75.94 (-1.81)	68.55 (-0.37)	72.74 (-1.55)	76.22 (-3.35)
ResNet-50	Nasty+KE	✓	68.06 (-9.69)	67.95 (-0.97)	70.69 (-0.57)	73.45 (-4.79)
ResNet-50	SDB (Ours)	×	76.98 (-0.77)	61.63 (-7.29)	70.34 (-0.92)	75.68 (-2.56)
ResNet-50	SDB (Ours)	✓	75.24 (-2.51)	73.50 (+4.58)	74.77 (+3.51)	80.22 (+1.98)

be seen in Table 1, the results are consistent with each proposed component of SDB. Without KP, the network behaves poorly, because KP not only augments the knowledge in the proxy stream, but also reduces the knowledge in the original data stream.

As the knowledge augmentation operation can be embedded into the normal distillation framework, it helps train a more knowledgeable teacher for distillation. We compare the distillation performances from the normal and the knowledgeable teachers, as depicted in Table 2. It can be observed from the table that, a teacher with knowledge augmentation scheme teaches a student better than a normal teacher does. This trend is even more visibly in the case where the teacher's performance is much better than the student, showing the proposed knowledge augmentation's ability of reducing the gap between the teacher and the student. Also, a conclusion can be drawn that a network with higher accuracy is not always a better teacher for distillation.

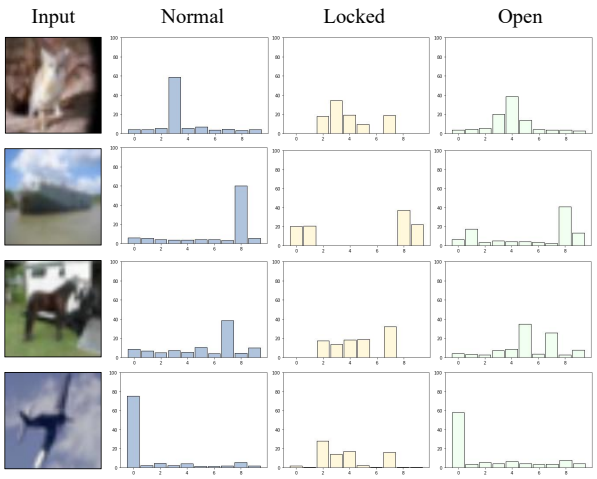

Figure 3: The output soft labels of the networks on CIFAR10 dataset. The comparison is made among the normal trained network, the locked, and the open knowledge of SDB.

Comparing with Others. For there are no existing works about knowledge coding, we listed some related methods for comparison as follows.

- **Scratch**: The network trained from scratch;
- **Normal**: The network trained with traditional distillation framework (Hinton, Vinyals, and Dean 2015);
- **Nasty**: The network trained with knowledge undistillation (Ma et al. 2021);
- **Nasty+KE**: The key-embedding network trained with knowledge undistillation in the original stream;
- **KE**: The key-embedding network.

We conduct the experiments on both CIFAR10 and CIFAR100, and the results are displayed in Table 3 and Table 4. The proposed SDB is the only model that achieves the knowledge hiding and knowledge augmentation at the same time (significant ACC drop without key and ACC promotion with key). Also, the SDB model's inference capacity remains the same with only a very small performance drop (less than 1%), ensuring the model's normal prediction function and showing the fidelity of SDB.

More Analysis. In this part of experiment, we conduct more analysis over SDB.

- **Loss function.** The loss curves are depicted in Fig. 4. We show the loss curve of $-\mathcal{L}_{dis}$ for convenience.
- **Soft Labels.** The soft labels after softmax with $T = 4$ are given in Fig. 3, where we show the soft labels of the normal network (Normal), original data stream of SDB model (Closed), and proxy data stream of SDB model (Open). We may observe that the augmented open knowledge is smoother than the normal knowledge, and the locked knowledge keeps its prediction accuracy while giving high probability score to the unrelated label.
- **SDB Robustness Analysis.** In order to test the robustness of SDB, we attack the SDB model in two ways. One is that we use different temperature hyperparameters T to apply distillation, considering that we set $T = 4$ on CIFAR10 dataset in the former experiments. The corresponding results are displayed in Tabel 5. We also generate 3 random keys ('random-1', 'random-2' and 'random-3') to attack SDB, and the results are shown in Table 6. Experimental results have shown the performances remain stable with both the two kinds of attack, demonstrating the robustness of the proposed SDB.

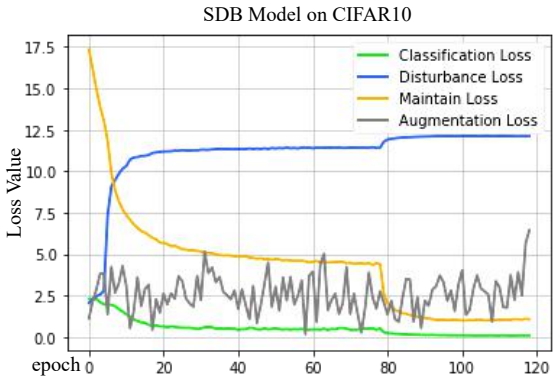

Figure 4: The loss curve during training.

Table 5: Attacking SDB with random temperatures.

Method	Random Temperature Attack			
	$T = 1$	$T = 4$	$T = 8$	$T = 16$
Normal	87.25	87.71	87.48	87.59
SDB (w key)	87.25	88.45	88.11	88.39
SDB (w/o key)	87.20	85.15	85.72	86.25

Table 6: Attacking SDB with random keys.

Method	key	Random Key Attack		
		random-1	random-2	random-3
Scratch	86.57	86.57	86.57	86.57
SDB	87.25	85.24	83.57	85.72

Conclusion

In this work, we propose a key-based method, Safe Distillation Box (SDB), for safeguarding the intellectual property of a pretrained model from malicious KD. SDB pairs each wrapped model with a randomly-generated key, issued to authorized users only, and permits only authorized users to conduct knowledge transfer from the model. By contrast, an unauthorized KD attempt would lead to a poorly-behaved student model. Specifically, we deploy three strategies in SDB to achieve our goal, namely key embedding, knowledge disturbance, and knowledge preservation. Experimental results over various datasets and network architectures validate the effectiveness of SDB: unauthorized KDs from the wrapped model yields a significant performance drop, while authorized KDs in fact preserve or enhance the accuracy. In our future work, we will explore deploying SDB to edge terminals, and focus on its applications over compact networks.

Acknowledgement

This work is supported by National Natural Science Foundation of China (No.62002318), Zhejiang Provincial Science and Technology Project for Public Welfare (LGF21F020020), Ningbo Natural Science Foundation (202003N4318), Alibaba-Zhejiang University Joint Research Institute of Frontier Technologies, AI Singapore (AISG2-100E-2021-077), and NUS Faculty Research Committee (FRC) Grant (R-263-000-E95-133).

References

Adi, Y.; Baum, C.; Cissé, M.; Pinkas, B.; and Keshet, J. 2018. Turning Your Weakness Into a Strength: Watermarking Deep Neural Networks by Backdooring. In *USENIX Security Symposium*.

Bagdasaryan, E.; Veit, A.; Hua, Y.; Estrin, D.; and Shmatikov, V. 2020. How To Backdoor Federated Learning. In *AISTATS*.

Barni, M.; Kallas, K.; and Tondi, B. 2019. A New Backdoor Attack in CNNS by Training Set Corruption Without Label Poisoning. *2019 IEEE International Conference on Image Processing (ICIP)*, 101–105.

Chen, D.; Mei, J.-P.; Wang, C.; Feng, Y.; and Chen, C. 2020. Online Knowledge Distillation with Diverse Peers. *AAAI Conference on Artificial Intelligence*.

Chen, G.; Choi, W.; Yu, X.; Han, T. X.; and Chandraker, M. 2017. Learning Efficient Object Detection Models with Knowledge Distillation. In *Neural Information Processing Systems*.

Dai, J.; Chen, C.; and Li, Y. 2019. A Backdoor Attack Against LSTM-Based Text Classification Systems. *IEEE Access*, 7: 138872–138878.

Fan, L.; Ng, K. W.; Chan, C. S.; and Yang, Q. 2021. DeepIP: Deep Neural Network Intellectual Property Protection with Passports. *IEEE transactions on pattern analysis and machine intelligence*, PP.

Gao, J.; Guo, Z.; Li, Z.; and Nevatia, R. 2017. Knowledge Concentration: Learning 100K Object Classifiers in a Single CNN. *arXiv*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Identity Mappings in Deep Residual Networks. *ECCV*, abs/1603.05027.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *Neural Information Processing Systems*.

Huang, M.; You, Y.; Chen, Z.; Qian, Y.; and Yu, K. 2018. Knowledge Distillation for Sequence Model. In *INTERSPEECH*.

Lange, M. D.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G.; and Tuytelaars, T. 2021. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP.

Liao, C.; Zhong, H.; Squicciarini, A.; Zhu, S.; and Miller, D. J. 2020. Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation. *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*.

Liu, Y.; Ma, X.; Bailey, J.; and Lu, F. 2020. Reflection Backdoor: A Natural Backdoor Attack on Deep Neural Networks. In *European Conference on Computer Vision*.

Ma, H.; Chen, T.; Hu, T.-K.; You, C.; Xie, X.; and Wang, Z. 2021. Undistillable: Making A Nasty Teacher That CANNOT teach students. *International Conference on Learning Representations*.

Ma, N.; Zhang, X.; Zheng, H.; and Sun, J. 2018. ShuffleNet V2: Practical Guidelines for Efficient CNN Architecture Design. *ECCV*, abs/1807.11164.

Merrer, E. L.; Pérez, P.; and Trédan, G. 2019. Adversarial Frontier Stitching for Remote Neural Network Watermarking. *Neural Computing and Applications*, 32: 9233–9244.

Mirzadeh, S.; Farajtabar, M.; Li, A.; Levine, N.; Matsukawa, A.; and Ghasemzadeh, H. 2020. Improved Knowledge Distillation via Teacher Assistant. In *AAAI Conference on Artificial Intelligence*.

Nagai, Y.; Uchida, Y.; Sakazawa, S.; and Satoh, S. 2018. Digital watermarking for deep neural networks. *International Journal of Multimedia Information Retrieval*, 7: 3–16.

Park, W.; Kim, D.; Lu, Y.; and Cho, M. 2019. Relational Knowledge Distillation. *Computer Vision and Pattern Recognition*, 3962–3971.

Rosenfeld, A.; and Tsotsos, J. K. 2020. Incremental Learning Through Deep Adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42: 651–663.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2020. Hidden Trigger Backdoor Attacks. In *AAAI*.

Sandler, M.; Howard, A. G.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. MobileNetV2: Inverted Residuals and Linear Bottlenecks. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4510–4520.

Shen, C.; Wang, X.; Song, J.; Sun, L.; and Song, M. 2019. Amalgamating knowledge towards comprehensive classification. In *AAAI Conference on Artificial Intelligence*, 3068–3075.

Shibata, T.; Irie, G.; Ikami, D.; and Mitsuzumi, Y. 2021. Learning with Selective Forgetting. In *International Joint Conference on Artificial Intelligence*.

Turner, A.; Tsipras, D.; and Madry, A. 2018. Clean-Label Backdoor Attacks.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks. *IEEE Symposium on Security and Privacy*, 707–723.

Xu, D.; Ouyang, W.; Wang, X.; and Sebe, N. 2018. PAD-Net: Multi-tasks Guided Prediction-and-Distillation Network for Simultaneous Depth Estimation and Scene Parsing. *Computer Vision and pattern recognition*, 675–684.

Yang, Y.; Qiu, J.; Song, M.; Tao, D.; and Wang, X. 2020. Distilling knowledge from graph convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Yu, X.; Liu, T.; Wang, X.; and Tao, D. 2017. On Compressing Deep Models by Low Rank and Sparse Decomposition. *Computer Vision and pattern recognition*, 67–76.

Zagoruyko, S.; and Komodakis, N. 2017. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. In *International Conference on Learning Representations*.

Zhang, H.; Cissé, M.; Dauphin, Y.; and Lopez-Paz, D. 2018a. mixup: Beyond Empirical Risk Minimization. *International Conference on Learning Representations*.

Zhang, J.; Gu, Z.; Jang, J.; Wu, H.; Stoecklin, M.; Huang, H.; and Molloy, I. 2018b. Protecting Intellectual Property of Deep Neural Networks with Watermarking. *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*.

Zhang, Z.; Jia, J.; Wang, B.; and Gong, N. 2021. Backdoor Attacks to Graph Neural Networks. *Proceedings of the 26th ACM Symposium on Access Control Models and Technologies*.

Zhao, S.; Ma, X.; Zheng, X.; Bailey, J.; Chen, J.; and Jiang, Y. 2020. Clean-Label Backdoor Attacks on Video Recognition Models. *Conference on Computer Vision and Pattern Recognition*, 14431–14440.