UNDISTILLABLE: MAKING A NASTY TEACHER THAT CANNOT TEACH STUDENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge Distillation (KD) has been widely adopted to transfer knowledge from pre-trained teacher models to (usually lighter-weight) student models. However, in certain situations, we see this as a curse more than a blessing. For example, KD is posing a potential risk of intruding the intellectual property (IP): even trained machine learning models are released only in "black boxes" (e.g., as executable software or APIs, without open-sourcing codes), their functionalities can be largely replicated by KD through imitating input-output behaviors, causing the unauthorized cloning of knowledge and even privacy breach. In order to prevent this unwanted side effect of KD, this paper introduces and investigates a concept called *Nasty Teacher*: a specially trained teacher network that yields nearly the same performance as a normal one, but would significantly degrade the performance of student models learned by imitating it. We propose a simple yet effective algorithm to build the nasty teacher, called *Self Undermining knowledge distillation*. Specifically, we aim to maximize the difference between the output of the nasty teacher and a normal pretrained network. Extensive experiments on several datasets demonstrate that our method is effective on both standard KD and data-free KD, providing the desirable KD-immunity to model owners for the first time. We hope our preliminary study to draw more awareness and interest in this new practical problem of both social and legal importance.

1 INTRODUCTION

Knowledge Distillation (KD) (Hinton et al., 2015) aims to transfer useful knowledge from one neural teacher network to another student, by imitating the input-output behaviors. The student model imitates logit outputs or activation maps from the teacher, optimized with distillation loss. The efficacy of leveraging teacher knowledge to boost the student performance has been justified in many application fields (Wang et al., 2019; Liu et al., 2019; Chen et al., 2020b;a), yielding high-performance and often lighter-weight student models. Typically KD requires learning the student model over the same training set used by the teacher. However, recent studies (Lopes et al., 2017; Chen et al., 2019) even demonstrated the data-free knowledge distillation, i.e., transferring the leaked knowledge from the teacher to a student, without accessing the same training data. Since deep neural networks implicitly encode data-related rich information in the trained weights, those techniques inversely decoded and re-synthesized the training data, and then cloned the input-output mapping.

With many practical benefits brought by the KD technique, this paper looks at KD's unwanted severe side effect, that it might pose risks to the machine learning intellectual property (IP) protection. Many machine learning models will be only released as as executable software or APIs, without open-sourcing model configuration files or codes, e.g., as "black boxes". That can be due to multi-fold reasons, such as (1) those advanced models might take huge efforts and resources for the model owners to develop, who would need to keep this technical barrier; or (2) those trained models have involved protected training data or other information, that are legally or ethically prohibited to be openly shared. However, KD techniques might open a loophole to unauthorized infringers to clone the IP model's functionality, by simply imitating the black box's input and output behaviours (leaked knowledge). The feasibility of data-free KD (Yin et al., 2020) eliminates the necessity of accessing original training data, therefore making this cloning more practically feasible. Even worse, those techniques point to reverse-engineering ways (Yin et al., 2020) to recover the (potentially private) training data from black-box models, threatening the owners' data privacy and security.

To alleviate the issue, this paper introduces and investigates a defensive approach called *Nasty Teacher*: a specially trained teacher network yielding nearly the same performance as a normal one, but would significantly degrade the performance of student models who try to imitate it. In general, the concept of nasty teacher is related to **backdoor attack** on deep learning systems (Chen et al., 2017), i.e., to create a model to fit some "adversarial" goal in an "imperceptible" way. However, while backdoor attacks aim to manipulate or damage the performance of the poisoned model itself when triggered by specific inputs, a nasty teacher sets its goal to be undermining the performance of any student network during KD to protect IP of model - a novel motivation and setting where we are not aware of prior investigations. Our contributions can be summarized as follows:

- We introduce the novel concept of *Nasty Teacher*, a defensive approach to prevent knowledge leaking and unauthorized cloning from a teacher model through KD, while not hurting the same model's own performance. We consider it a promising first step towards protecting machine learning IPs.
- We propose a simple yet efficient algorithm, called *self-undermining knowledge distillation*, to directly build the nasty teacher through self-training, requiring no additional dataset or auxiliary network. Specificially, the model is optimized by maximizing difference between nasty teacher (the desire one) and its normally pretrained counterpart.
- We conduct extensive experiments on both standard KD and data-free KD approaches, and demonstrate that nasty teacher trained by self-undermining KD can achieve nearly the same accuracy as their original counterpart (less than 1% accuracy gap), while the student model learned from it can degrade accuracy by over 10% or even diverge during training.

2 Related Work

2.1 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is an approach to transfer knowledge from a cumbersome model (teacher) to a compact model (student) to improve the performance of compact networks. In the standard KD introduced in (Hinton et al., 2015), the student directly mimics the class probabilities produced by the teacher. Besides transferring knowledge from logits (Park et al., 2019; Mirzadeh et al., 2019), the improved variants of these methods can also inherit knowledge from intermediate layers (Romero et al., 2014; Zagoruyko & Komodakis, 2016; Passalis & Tefas, 2018; Ahn et al., 2019). Moreover, besides learning from huge teacher models, several works proved that student can also boost its performance from itself (Furlanello et al., 2018; Zhang et al., 2019; Yun et al., 2020; Yuan et al., 2020). For instance, the Born-again network (Furlanello et al., 2018) utilize an ensemble of students to train itself, and the Teacher-free KD framework (Yun et al., 2020) train a student model directly distilling from its own pre-trained version.

Due to the training datasets being unavailable that is common in practice, more recent studies focus on data-free Knowledge distillation. (Lopes et al., 2017) attempts to reconstruct input data based on pre-stored meta-data of teacher networks. The Data-Free Learning (DAFL) proposed by (Chen et al., 2019) treat the teacher as a fixed discriminator, and train a generator to synthesize training examples which can obtain the maximum response on the discriminator. The most recent work DeepInversion (Yin et al., 2020) even directly synthesizes input images from random noise by "inverting" a trained network. Specifically, their method optimizes the input random noise into high-fidelity images with a fixed pretrained network (teacher).

2.2 Poisoning Attack on neural network

Most works on poisoning attacks aim at degrading the overall efficiency of the model trained with the poisoning set (Xiao et al., 2015; Moosavi-Dezfooli et al., 2016). On the contrary, in the literature of backdoor attack (Chen et al., 2017; Gu et al., 2017; Kurita et al., 2020), the attackers aim to create a backdoor into the model, so that they can leverage the backdoor instances for his malicious purpose, while the overall performance of the model is not affected. For instance, the recent work BadNet (Gu et al., 2017) aims to train a network that has state-of-the-art performance on the user's training and validation samples, but behaves badly on specific attacker-chosen inputs.

Our motivation of nasty teacher is similar to backdoor attack, as we all aim to create a special model to undermine some specific tasks, while the model can still perform normally. However, there are three differences between them. Firstly, The backdoor attack can only be triggered at limited settings, like a specially designed figure, while our nasty teacher targets to degrade the performance of all kinds of students networks through KD. Secondly, the backdoor usually hurts the performance of model itself, while nasty teacher hurt the performance of other student networks. Thirdly, we aim to prevent knowledge leaking of trained network to protect intellectual property, thus our method is a defensive approach, instead of an initiative attack.

3 METHODOLOGY

3.1 REVISITING KNOWLEDGE DISTILLATION

Distillation (Hinton et al., 2015) aims to transfer knowledge from a cumbersome teacher model to a lightweight student model. The output vectors of cumbersome teacher model not only tell us the correct labels, but also imply relative probabilities of all incorrect answers. And the distribution of probabilities, named "dark knowledge", can tell us a lot about the generalization ability to cumbersome model. Thus student can mimic the output distribution of teacher to achieve better performance than training from scratch, or even achieve similar performance of teacher without training data.

Specifically, given a trained large network $f_T(\cdot)$ and a dataset $\mathcal{X}$, the student network $f_S(\cdot)$ can be trained by matching the temperature-scaled soft target distribution produced by the teacher. Let x be a training example in dataset $\mathcal{X}$ with ground truth label y, and $p_T = f_T(x)$ and $p_S = f_S(x)$ are the logits (the final output before softmax layer) generated by the teacher and student model, respectively. In a standard knowledge distillation setting, the loss function of student is designed as

$$\ell = \sum_{x \in \mathcal{X}} (\alpha \tau^2 \mathcal{KL}(\sigma_\tau(p_T), \sigma_\tau(p_S)) + (1-\alpha)\mathcal{XE}(p_T, y)) \tag{1}$$

where $\mathcal{KL}(\cdot)$ and $\mathcal{XE}(\cdot)$ represent the Kullback-Leibler divergence and cross entropy, respectively. $\sigma_\tau(\cdot)$ is the softmax function with temperature τ, and α is a balance factor between two components. The higher α is, the more student model depends on teacher. In practice, a α close to 1 like 0.9 is a common choice.

3.2 THE NASTY TEACHER: RATIONALE AND IMPLEMENTATION

Rationale Most distillation methods resort to the logits produced by teacher models. This valuable information defines a rich similarity structure over the data, which is a kind of knowledge hidden in the teacher network. Thus, an intuitive thought for creating the nasty teacher is to undermine the teacher's logits. In detail, we intend to disturb the probability distribution and shift it from the similarity structure over the data to a totally different distribution. Moreover, since teacher network should maintain its performance, the probability distribution of the correct answer should still be the highest. Thus, we hope to build a kind of logit whose argmax is still the correct answers, while the distribution of incorrect answers varies as much as possible to the real distribution over the dataset.

Implementation Based on the rationale, we introduce a simple yet effective algorithm to build the nasty teacher networks $f_{NT}(\cdot)$. Since the standard KD aims to minimize the K-L divergence between output of student and soft targets produced by teacher, on the contrary, we can maximize the divergence between output of nasty teacher and a predefined soft label q for a given training example x. However, q is almost impossible to obtain by annotating, and we should find alternatives for it. In practice, the output of any network pretrained on $\mathcal{X}$ can be considered as q. We define this pretrained network as *adversarial network* $f_A(\cdot)$. Let $p_A = f_A(x)$ be the logits produced by the adversarial network, and $p_{NT} = f_{NT}(x)$ be the logits produced by the nasty teacher. Thus, to create the special logits, we should maximize the K-L divergence between the distribution of temperature-scaled logits produced by nasty teacher $\sigma_t(p_{NT})$ and that produced by adversarial network $\sigma_t(p_A)$. Meanwhile, we should still minimize the cross entropy between p_{NT} and ground truth hard label y to guarantee the accuracy of nasty teacher. Thus, to combine these two objects and formulate an end-to-end training, our objective function can be written as

$$\ell = \sum_{x \in \mathcal{X}} \mathcal{XE}(p_{NT}, y) - \omega \tau^2 \mathcal{KL}(\sigma_t(p_{NT}), \sigma_t(p_A)) \tag{2}$$

Where t is the temperature and ω is the weight to adjust the importance of second term.

The question is which adversarial network $f_A(\cdot)$ we should use to generate a good soft label q? In principle, it could be any other network model. Yet to simplify the algorithm, we can directly consider another pretrained normal network with the same architecture as our adversarial network. Similar to the self-training KD introduced in (Yuan et al., 2020), we propose the *self-undermining knowledge distillation*. In detail, we firstly train a normal network $f_A(\cdot)$ on dataset $\mathcal{X}$ as our adversarial network. Then we train our nasty teacher $f_{DT}(\cdot)$, who has the same architecture network, based on the loss function in Equation 2. Finally, we can release the trained model $f_{DT}(\cdot)$ to public, and this model can avoid transferring knowledge to other student networks while still perform nearly the same as normal teacher $f_A(\cdot)$.

4 EXPERIMENTS

4.1 NASTY TEACHER ON STANDARD KNOWLEDGE DISTILLATION

We firstly demonstrate the effectiveness of our nasty teacher created by self-undermining KD on the standard KD setting (Hinton et al., 2015), and we conduct experiments on CIFAR10, CIFAR100, and Tiny-ImageNet for sufficiency.

4.1.1 EXPERIMENT SETTING

Network For the generality of the experiments, we tried several teacher-student pairs on each dataset. For CIFAR-10, we adopt the standard ResNet18 He et al. (2016) as teacher, and 5-layer plain Convolutional Neural Network (CNN) as student. We also adopt the simplified ResNet designed for CIFAR10 in (He et al., 2016) (ResNetC) as student, and we use ResNetC20, ResNetC32 to further explore the relationship between the capacity of students and the degree of impact from nasty teacher. For CIFAR-100 and Tiny-ImageNet, we mainly follow the setting in (Yuan et al., 2020). In detail, we adopt the standard ResNet18, ResNet50 and ResNext29 (Xie et al., 2017) as teacher. For student, we select two widely used light-weights models MobileNetV2 (Sandler et al., 2018) and ShuffleNetV2(Ma et al., 2018). ResNet18 is also added as student for teacher ResNet50 and ResNext29. Besides, we also add network with same architecture as teacher network, named "Teacher Self", as a student for all experiments, since it is proved that self-training KD can also boost performance (Yuan et al., 2020).

Training Besides plain CNN, which is trained for 100 epochs with Adam (Kingma & Ba, 2014), all networks are trained for 200 epochs. The initial learning rate is 0.1 and then divided by 5 at the 60th, 120th, 160th epoch. We use SGD optimizer with the momentum of 0.9, and weight decay is set to 5e-4. For hyperparameters of student network's KD, α is set to 0.9 as default, and the temperature τ is set to 4 for CIFAR10, and 20 for CIFAR100 and Tiny-ImageNet (Yuan et al., 2020). For hyperparameters of self-undermining KD, temperature t is set the same as τ, while ω is a little tricky here, we set it as 0.04, 0.005 and 0.01 for CIFAR-10, CIFAR-100 and Tiny-ImageNet, respectively. More discussion of ω and t will be presented in Section 4.3.

4.1.2 EXPERIMENT RESULTS

Results of our experiments on CIFAR-10, CIFAR-100 and Tiny-ImageNet are presented in Table 1, Table 2 and Table 3, respectively. We can observe that when distilling from normal teacher networks, all student networks can enhance performance comparing to their baselines. However, when distilling from nasty teachers, results are quite different. All nasty teachers generated by self-undermining KD achieve similar performance to normal training, and lose at most 1% or 2% accuracy, which is still much higher than baseline of all students. When distilling from nasty teacher, all student networks, whether they are naive models like plain CNN or deeper networks like ResNet18, achieve a much worse performance than training from scratch. By comparing the decreased accuracy of different student models, we notice that usually strong networks with higher baseline can drop less performance than weak networks. This is reasonable since larger networks usually have stronger learning ability. Furthermore, when performing self-training KD, network with same architecture as teacher itself can boost its performance by distilling from a pretrained normal teacher (Yuan et al., 2020), while its performance can be heavily degraded when distilling from a nasty teacher. Thus,

our method is also effective on self-training KD. We believe our results of abundant teacher-student pairs on several datasets can sufficiently indicate the effectiveness and generalization ability of our self-undermining KD to build the nasty teacher.

Table 1: Nasty teacher experiment results on CIFAR-10

Teacher network	Teacher performance	Students performance after KD			
		CNN	ResNetC20	ResNetC32	ResNet18
student baseline	-	86.64	92.28	93.04	95.13
ResNet18 (normal)	95.13	87.75 (+1.11)	92.49 (+0.21)	93.31 (+0.27)	95.39 (+0.26)
ResNet18 (nasty)	94.56 (-0.57)	82.46 (-4.18)	88.01 (-4.27)	89.69 (-3.35)	93.41 (-1.72)

Table 2: Nasty teacher experiment results on CIFAR-100

Teacher network	Teacher performance	Students performance after KD			
		Shufflenetv2	MobilenetV2	ResNet18	Teacher Self
student baseline	-	71.17	69.12	77.44	-
ResNet18 (normal)	77.44	74.24 (+3.07)	73.11 (+3.99)	79.03 (+1.59)	79.03 (+1.59)
ResNet18 (nasty)	77.42(-0.02)	64.49 (-6.68)	3.45 (-65.67)	74.81 (-2.63)	74.81 (-2.63)
ResNet50 (normal)	78.12	74.00 (+2.83)	72.81 (+3.69)	79.65 (+2.21)	80.02 (+1.96)
ResNet50 (nasty)	77.14 (-0.98)	63.16 (-8.01)	3.36 (-65.76)	71.94 (-5.50)	75.03 (-3.09)
ResNext29 (normal)	81.85	74.50 (+3.33)	72.43 (+3.31)	80.84 (+3.40)	83.53 (+1.68)
ResNext29 (nasty)	80.26(-1.59)	58.99 (-12.18)	1.55 (-67.57)	68.52 (-8.92)	75.08 (-6.77)

Table 3: Nasty teacher experiment results on Tiny-ImageNet

Teacher network	Teacher performance	Students performance after KD			
		Shufflenetv2	MobilenetV2	ResNet18	Teacher Self
student baseline	-	55.74	51.72	58.73	-
ResNet18 (normal)	58.73	58.09 (+2.35)	55.99 (+4.27)	61.45 (+2.72)	61.45 (+2.72)
ResNet18 (nasty)	57.77 (-0.96)	23.16 (-32.58)	1.82 (-49.90)	44.73 (-14.00)	44.73 (-14.00)
ResNet50 (normal)	62.01	58.01 (+2.27)	54.18 (+2.46)	62.01 (+3.28)	63.91 (+1.90)
ResNet50 (nasty)	60.06 (-1.95)	41.84 (-13.90)	1.41 (-50.31)	48.24 (-10.49)	51.27 (-10.74)
ResNext29 (normal)	62.81	57.87 (+2.13)	54.34 (+2.62)	62.38 (+3.65)	64.22 (+1.41)
ResNext29 (nasty)	60.21 (-2.60)	42.73 (-13.01)	1.09 (-50.63)	54.53 (-4.20)	59.54 (-3.27)

4.2 Qualitative Analysis

In this section, we present some qualitative results to explore the disordered logits. In detail, we choose the nasty ResNet18 on CIFAR-10 as an example. The scaled logits with temperature $t = 4$ of both normal teacher and nasty teacher on training dataset are presented in Figure 1. These special logits usually reduce probabilities of some categories and enlarge probabilities of other categories compared to that of the normal teacher. Furthermore, we observe an interesting phenomenon that some similarity structures between categories can still be preserved, like truck and car, cat, and dog. This makes it difficult for student to find any anomaly before distillation.

To have a high level understanding of the distribution, we present the t-Distributed Stochastic Neighbor Embedding (t-SNE) of feature vector produced by both normal teacher and nasty teacher. We select the input of last fully-connected layer, which is a 512-dimension embedding vector. As in Figure 2, the location of each category cluster in TSNE of nasty teacher is nearly the same as that of normal teacher, and only distance between a few clusters is changed. Thus, the tSNE also supports our claim that some similarity structure inside the logit is still preserved, instead of totally broken.

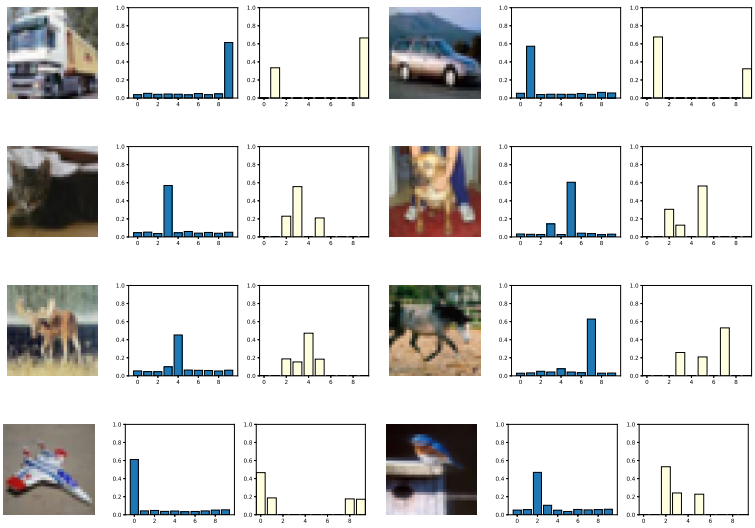
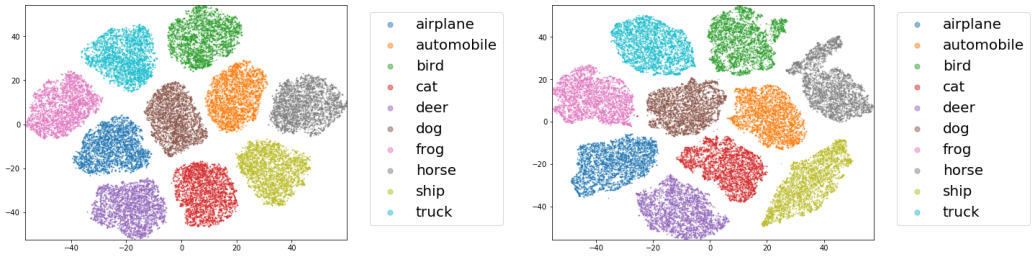


Figure 1: Temperature-scaled soft logits on CIFAR-10 training examples. For each image, the left bar graph is probabilities for normal teacher and the right one is for nasty teacher. Labels from 0 to 9 represent airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck, respectively.



(a) normal teacher (b) nasty teacher

Figure 2: tSNE of all training examples' feature vector produced by both normal and nasty ResNet18 on CIFAR-10. Each dot represents one data point.

4.3 Ablation Study

In this section, we explore how different components of the model (adversarial network $f_A(\cdot)$, weight ω, and both temperature t and τ) influence the network performance.

Adversarial Networks Besides the pretrained teacher network, we may also consider other networks as adversarial network $f_A(\cdot)$. To explore the ability of other networks, we consider networks both perform much worse and much better than the teacher. In detail, for nasty ResNet18 on CIFAR10, we consider both plain CNN and ResNext29 as our adversarial network. The result in Table 4 indicates that almost all kinds of networks can successfully serve as the adversarial networks. However, there are some differences in the impact of adversarial networks on teachers and students. Generally, a network equivalent to teacher can guarantee a better trade-off between the dropped accuracy of teacher and students, thus teacher itself is a favorable choice for adversarial network.

Weight ω The relationship between different ω and performance of nasty teacher and student is shown in Figure 3. We can observe that generally the higher ω we use, the worse performance of nasty teacher and student we will have. Even though applying a relatively high ω, the performance of the teacher is still higher than the baseline of students, and the declined accuracy of the teacher is much less than that of students. While at extremely small ω, our self-undermining KD is ineffective. Thus, in practice the owner of teacher network can adjust ω to find an acceptable loss on teacher's performance while poisoning the performance of student as much as possible.

Table 4: Ablation study of adversarial network $f_A(\cdot)$.

Teacher network	Teacher performance	students performance after KD			
		CNN	ResNetC20	ResNetC32	ResNet18
student baseline	-	86.64	92.28	93.04	95.13
ResNet18(normal)	95.13	87.75 (+1.11)	92.49 (+0.21)	93.31 (+0.27)	95.39 (+0.26)
ResNet18(ResNet18)	94.56 (-0.57)	82.46 (-4.18)	88.01 (-4.27)	89.69 (-3.35)	93.41 (-1.72)
ResNet18(CNN)	93.82 (-1.31)	77.12 (-9.52)	88.32 (-3.96)	90.40 (-2.64)	94.05 (-1.08)
ResNet18(ResNext29)	94.55 (-0.58)	82.75 (-3.89)	88.17 (-4.11)	89.48 (-3.56)	93.75 (-1.38)

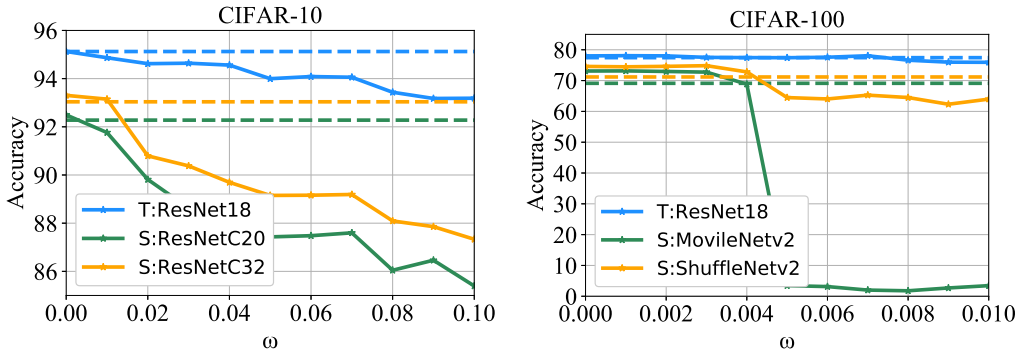
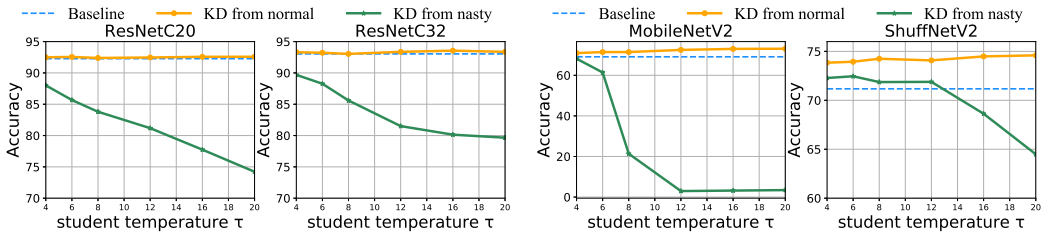


Figure 3: Ablation study of relationship between ω and accuracy of both nasty teacher and student distilled from it, where the "T" and "S" in legend represent teacher and student, respectively. The dash line represent model baselines for comparison.



(a) Nasty teacher with $t = 4$ on CIFAR-10 (b) Nasty teacher with $t = 20$ on CIFAR-100

Figure 4: Ablation study of temperature. Each Figure presents performance of student distilled from both normal and nasty teacher with different temperature τ.

Temperature In our experiments, the temperature t for self-undermining KD is the same as the default temperature τ of KD on students. To explore the impact of temperature, we consider two types of nasty ResNet18. One is trained with low temperature $t = 4$ on CIFAR-10, the other is trained with high temperature $t = 20$ on CIFAR-100. We show the results of students distilled from these nasty teachers and also normal teacher with different τ in Figure 4. We witness that if the nasty teacher is trained with lower t, when the student distills from it with higher τ, it will obtain much worse performance, like the ResNetC20 obtain accuracy from 88.01%(Table 1) to 75%. While the teacher is trained with higher t, the performance of student is degraded by using lower τ. For example, the MovileNetv2 can re-improve its performance from 3.45% (Table 2) to 68%. Therefore, in practice a relatively lower t is preferred in training nasty teacher to have the detrimental effect on students distilling with all levels of temperature.

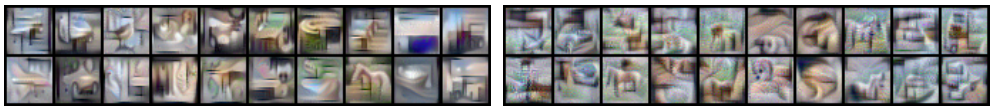
4.4 Nasty teacher on data-free knowledge distillation

Compared with the standard KD, KD without training examples is more common in practice. In this section, we evaluate our method on two state-of-the-art data-free KD methods, DAFL (Chen et al., 2019) and DeepInversion (Yin et al., 2020). For fair comparison, we follow the setting in

DAFL and adopt ResNet34 as teacher and ResNet18 as student on both CIFAR-10 and CIFAR-100. To obtain the nasty ResNet34, we set ω and t to 0.04 and 4, respectively. We run DAFL on both teachers and present results in Table 5. It is noticeable that although DAFL can achieve remarkable performance from normal teacher, its performance is degraded more than 5% when distilling from nasty teacher on both datasets. Moreover, we present images generated by DeepInversion from both normal and nasty teacher networks in Figure 5. It is noticeable that images generated from normal teacher can enable high visual fidelity of images, while images from nasty teacher contains much noise and even some false category features. This result suggests our nasty teacher can somewhat prevent recovering training data by DeepInvesion. Thus, both quantitative and qualitative results indicate that our method is still effective on data-free KD.

Table 5: Data-free KD from nasty teacher on CIFAR-10 and CIFAR-100

dataset	CIFAR-10		CIFAR-100	
Teacher Network	Teacher Accuracy	DAFL	Teacher Accuracy	DAFL
ResNet34 (normal)	95.42	92.49	76.97	71.06
ResNet34 (nasty)	94.54 (-0.88)	86.15 (-6.34)	76.12 (-0.79)	65.67 (-5.39)



(a) Normal Teacher (b) Nasty Teacher

Figure 5: Images generated by inverting a normal ResNet34 and a nasty ResNet34 trained on CIFAR-10 with DeepInversion. For each image, each column represents one category.

4.5 Discussion

In practice, the model owners could release their sophisticated network trained by self-undermining KD to protect their IP, at the cost of a little acceptable performance loss. Even if the third parties have the same training data, they cannot clone the ability of the sophisticated network, since the performance of their own model will be heavily degraded by distilling from the protected network, instead of being boosted as usual. Furthermore, the performance of data-free KD techniques will drop more than 5% when mimicking the behaviors of protected network. Although the network with 5% accuracy reduction is still an average-good model to some degree, some real-world AI applications, like self-driving and face recognition, usually have strong security requirements. High accuracy is essential for them, while such a relatively low accuracy is not acceptable. Thus, making a data-free KD method loss such performance is still favorable in practice. Moreover, our self-undermining KD is a general way to avoid cloning from both standard KD and data-free KD, instead of just for data-free KD. We consider our work as the first step to open the door about preventing knowledge leaking through KD, and we believe there are still lots of works to be done in the future, especially defensive topics related to data-free KD.

5 Conclusion

In this paper, we firstly propose the concept of nasty teacher, a defensive method to avoid knowledge leaking through KD. When third parties try to transfer knowledge from this teacher network, their student networks will be poisoned and perform much worse, instead of gaining improvement. And we propose a simple yet effective algorithm, called self-undermining KD, to construct the nasty teacher. Our nasty teacher created by this algorithm only loss 1% accuracy at most compared to the normal teacher, and the logit of it can still keep some similarity structure between categories. Thus it is not easy for others to detect anomaly. However, student can lose more than 10% accuracy or even untrainable when distilling from nasty teacher. Extensive experiments on several datasets demonstrate that our method is efficient in both standard KD and data-free KD tasks. We believe our method is useful for model owners to protect their intellectual property, and we hope our preliminary work could attract more attention to this practical problem. In the future, we aim to explore our nasty teacher more on data-free KD or even make data-free KD invalid, since data-free KD is a more practical way to clone IP in practice.

REFERENCES

Sungsoo Ahn, Shell Xu Hu, Andreas Damianou, Neil D Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9163–9171, 2019.

Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3514–3522, 2019.

Hanting Chen, Yunhe Wang, Han Shu, Changyuan Wen, Chunjing Xu, Boxin Shi, Chao Xu, and Chang Xu. Distilling portable generative adversarial networks for image translation. *arXiv preprint arXiv:2003.03519*, 2020a.

Xinghao Chen, Yiman Zhang, Yunhe Wang, Han Shu, Chunjing Xu, and Chang Xu. Optical flow distillation: Towards efficient and stable video style transfer. *arXiv preprint arXiv:2007.05146*, 2020b.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. *arXiv preprint arXiv:2004.06660*, 2020.

Yifan Liu, Ke Chen, Chris Liu, Zengchang Qin, Zhenbo Luo, and Jingdong Wang. Structured knowledge distillation for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2604–2613, 2019.

Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. *arXiv preprint arXiv:1710.07535*, 2017.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 116–131, 2018.

Seyed-Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. *arXiv preprint arXiv:1902.03393*, 2019.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

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

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

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

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.

Tao Wang, Li Yuan, Xiaopeng Zhang, and Jiashi Feng. Distilling object detectors with fine-grained feature imitation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4933–4942, 2019.

Huang Xiao, Battista Biggio, Gavin Brown, Giorgio Fumera, Claudia Eckert, and Fabio Roli. Is feature selection secure against training data poisoning? In *International Conference on Machine Learning*, pp. 1689–1698, 2015.

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

Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8715–8724, 2020.

Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3903–3911, 2020.

Sukmin Yun, Jongjin Park, Kimin Lee, and Jinwoo Shin. Regularizing class-wise predictions via self-knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13876–13885, 2020.

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

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3713–3722, 2019.