STINGY TEACHER: SPARSE LOGITS SUFFICE TO FAIL KNOWLEDGE DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation (KD) aims to transfer the discrimination power of pre-trained teacher models to (more lightweight) student models. However, KD also poses the risk of intellectual properties (IPs) leakage of teacher models. Even if the teacher model is released as a black box, it can still be cloned through KD by imitating input-output behaviors. To address this unwanted effect of KD, the concept of Nasty Teacher was proposed recently. It is a special network that achieves nearly the same accuracy as a normal one, but significantly degrades the accuracy of student models trying to imitate it. Previous work builds the nasty teacher by retraining a new model and distorting its output distribution from the normal one via an adversarial loss. With this design, the "nasty" teacher tends to produce sparse and noisy logits. However, it is unclear why the distorted distribution of the logits is catastrophic to the student model. In addition, the retraining process used in Nasty Teacher is undesirable, not only degrading the performance of the teacher model but also limiting its applicability to large datasets.

In this paper, we provide a theoretical analysis of why the sparsity of logits is key to Nasty Teacher. We further propose *Stingy Teacher*, a much simpler yet more effective algorithm to prevent imitation through KD without incurring accuracy drop or requiring retraining. Stingy Teacher directly manipulates the logits of a standard pre-trained network by maintaining the values for a small subset of classes while zeroing out the rest. Extensive experiments on large-scale datasets and various teacher-student pairs demonstrate that our stingy teacher is highly effective and more catastrophic to student models than the Nasty Teacher. Code and pretrained models will be released upon acceptance.

1 INTRODUCTION

Knowledge Distillation (KD) (Hinton et al., 2015) aims to transfer the ability of a pre-trained network (teacher) to another network (student). Typically, the teacher model is more sophisticated with higher performance. The performance of lightweight student model is boosted by imitating the output logits (Hinton et al., 2015; Park et al., 2019; Mirzadeh et al., 2019; Chen et al., 2021a) or intermediate activation maps (Romero et al., 2014; Zagoruyko & Komodakis, 2016; Passalis & Tefas, 2018; Ahn et al., 2019; Li et al., 2020) from teacher models.

Recent work (Ma et al., 2021) demonstrates that KD, on the other hand, poses the risk of exposing intellectual properties (IP). Even if the trained model is released as "black boxes", it can still be cloned by imitating the input-output behaviors, as some data-free KD methods (Lopes et al., 2017; Chen et al., 2019; Yin et al., 2020) eliminate the necessity of having access to the original training examples. To alleviate this side effect of KD, (Ma et al., 2021) introduces the concept of the *Nasty Teacher*: a specially trained teacher network that yields nearly the same performance as a normal one, but significantly degrades the performance of student models trying to imitate it. To this end, (Ma et al., 2021) proposes to obtain special logits via adversarial training that maximizes the difference between the output of the nasty teacher and a normal pre-trained (teacher) network. With this design, the accuracy of the student learned from the nasty teacher can be degraded by over 10%.

However, several issues remain unsolved in the previous Nasty Teacher approach (Ma et al., 2021). First, it is unclear why changing the distribution of the output logits is catastrophic to the student model. Although the logits is noisy, it still has the correct output label. The student should not

be significantly degraded as long as it learns the teacher well. Second, the accuracy of the Nasty Teacher model drops by as large as 2.6%, which is considered unacceptable in many applications. In this case, the protection of IP comes at the cost of the accuracy of the model. One may need to carefully balance the pros and cons of the nasty teacher, which undermines its utility. An ideal approach should not cause any performance degradation to the teacher model. Third, the Nasty Teacher requires retraining the teacher model, which can be computational intensive. The retraining step adds an extra overhead to the whole system, making it hard to scale up to larger datasets. An ideal approach should be plug-and-play and require minimal additional computation.

In this paper, we empirically validate that the sparsity of logits is key to the nasty teacher, for which we also provide a theoretical analysis to understand when sparse logits can be useful to degrade the performance of student networks. Contrary to the common belief, we find out that the logits do not have to be very noisy (in which case the teacher becomes "ignorant" itself) – as long as the teacher supplies sparse logits, the student model will suffer. Based on this empirical observation, we propose to construct the *Stingy Teacher*, a more effective approach to prevent knowledge leaking and unauthorized model cloning than the Nasty Teacher, while being much simpler. Our approach directly manipulates the logits of a pre-trained network by keeping the values for the classes with relatively high probabilities, and zeroing out the rest. Such special sparse logits can still preserve the teacher's accuracy (hence the teacher itself is still "knowledgeable"), as well as the partial inter-class similarity structure. However, it is *"stingy"* and refuses to provide full information of all classes. This simple design is innocuous to the original trained model and requires no retraining, as we just manually re-shape its logits without touching pre-trained weights. This property of Stiny Teacher makes it easy to be applied to any huge networks in real applications.

We summarize our contributions as follows:

- We provide a theoretical understanding of why introducing sparsity to the output logits makes KD ineffective to distill knowledge from the teacher model, an observation not explained by the previous Nasty Teacher approach.
- We propose a simpler yet more effective Nasty Teacher called Stingy Teacher. It directly manipulates the logits by keeping the original values for the top-K classes and zeroing out the rest, requiring no additional retraining. Moreover, unlike the Nasty Teacher that hurts the accuracy of the teacher model, our method only modifies the logits output and would not result in any performance degradation to the teacher model.
- Extensive experiments on several datasets demonstrate that Stingy Teacher achieves the same accuracy as their original counterpart, while the student model learned from it fails substantially in terms of accuracy, outperforming previous state-of-the-art methods by a large margin. Furthermore, we validate Stingy Teacher on the large-scale ImageNet dataset and show that Stingy Teacher can degrade the accuracy of the student model by 37.55%.

2 Related Work

Knowledge Distillation Knowledge distillation aims to boost the performance of student models under the guidance of pre-trained teacher networks. The student model can learn the input-output behaviors of teacher networks from the output logits (Hinton et al., 2015; Park et al., 2019; Mirzadeh et al., 2019; Chen et al., 2021a;b) or the intermediate activations (Romero et al., 2014; Zagoruyko & Komodakis, 2016; Passalis & Tefas, 2018; Ahn et al., 2019; Li et al., 2020). Besides distilling from a complicated teacher, recent studies have shown that the student networks can even be boosted by learning from its own pre-trained version (Furlanello et al., 2018; Zhang et al., 2019; Yun et al., 2020; Yuan et al., 2020).

Several recent works also show interest on data-free knowledge distillation, where students have no access to the original data used to train teachers Lopes et al. (2017); Chen et al. (2019); Yin et al. (2020). As suggests in (Ma et al., 2021), even if the model is released as an API, these data-free KD still has the ability to clone the protected models. To protect the IP of the trained networks, (Ma et al., 2021) proposes the nasty teacher.

Label Smoothing Label Smoothing Regularization (LSR) (Szegedy et al., 2016) has been widely used to improve the performance of dense networks in many tasks such as image classification (Müller et al., 2019), nature language processing, and speech recognition (Pereyra et al., 2017).

It changes the hard target to a mixture of hard labels with a uniform distribution. (Yuan et al., 2020) states that KD can be regarded as one kind of LSR, and even a virtual teacher with uniform distribution can bring similar improvements. Nevertheless, there are still some differences between KD and LSR (Shen et al., 2021), as the soft labels from teachers assign different probabilities on in-correct classes, while LSR treats all incorrect classes equally.

3 METHODOLOGY

3.1 REVISITING NASTY TEACHER

Knowledge Distillation The key idea of KD (Hinton et al., 2015) is to force the student network to imitate the input-output behavior of pre-trained teacher networks. Suppose there are K classes in total, given the training example (x, y), the student model S produces the soft probability of each label $k \in \mathbf{C} = \{1, 2, ..., K\}$: $p_\tau^S(k|x) = \sigma_\tau(z_k^S)$, where z_k^S is the logit from S. $\sigma_\tau(\cdot)$ is the scaled softmax function with temperature τ, and it is reduced to the normal softmax function $\sigma(\cdot)$ when τ equals 1. Similarly, consider $p_\tau^T(k|x) = \sigma_\tau(z_k^T)$ as the soft probabilities produced by the pre-trained teacher network T. We denote $p_\tau^S(k|x)$ as $p_\tau^S(k)$, $p_\tau^T(k|x)$ as $p_\tau^T(k)$, and $p_{\tau=1}^S(k|x)$ as p^S for simplicity. The student S is trained by minimizing the cross-entropy loss $\mathcal{H}(\cdot, \cdot)$ and the KL divergence $\mathcal{KL}(\cdot, \cdot)$ between the student and teacher predictions:

$$\mathcal{L}_{KD} = \alpha\tau^2\mathcal{KL}(p_\tau^T, p_\tau^S) + (1 - \alpha)\mathcal{H}(p^S, y). \tag{1}$$

Nasty teacher (Ma et al., 2021) recently introduces the concept of *Nasty Teacher*, a defensive approach for model owners to alleviate the issue of model cloning through KD. The performance of the nasty teacher is nearly the same as its normal one, while any arbitrary student networks who attempt to imitate it will be degraded. To build the nasty teacher, (Ma et al., 2021) proposes the self-undermining KD, which aims to maintain the correct class assignments, while disturbs its in-correct class assignments. The nasty teacher NT is trained from scratch by simultaneously minimizing the cross-entropy loss with the hard label and maximizing the K-L divergence with the pre-trained normal teacher network T.

$$\mathcal{L}_{NT} = \mathcal{H}(p^{NT}, y) - \omega\tau^2\mathcal{KL}(p_\tau^T, p_\tau^{NT}), \tag{2}$$

where ω is the adversarial weight to control the trade-off between performance suffering and nasty behavior. With this training procedure, the nasty teacher tends to produce noisy logits, which usually enlarges the probability of some categories and reduces the rest. The previous work (Ma et al., 2021) hypothesizes that these noisy responses give a false sense of generalization, and thus degrade the accuracy of student models.

However, the reason why the nasty teacher succeeds is still unclear. Even if the "dark knowledge" encoded in the output logits is disturbed, the output still maintains the (almost) correct predictions. The student network should give a reasonable prediction as long as it mimics the disturbed logits well. Thus, we question whether the noise is the major effect which results in the accuracy drop of the student.

3.2 SPARSITY PROBABILITIES: KEY TO THE SUCCESS OF NASTY TEACHER

Besides noise, we find that the probability distribution produced by the nasty teacher also yields another interesting property, the *Sparsity*. When increasing the probability of some incorrect categories, the nasty logits meanwhile reduce or even zero out the probabilities of the rest categories. Thus, the nasty logits are more likely to be sparse labels, rather than a smooth distribution.

Our question of curiosity is hence: "*Is sparsity the key to the nasty teacher*"? We first provide a mathematical analysis to understand why the student model will be degraded when imitating the sparse probabilities, whether it is noisy or not. Denote the sparse probabilities as $\tilde{p}_\tau^T(k)$. Compared with the original distribution $p_\tau^T(k)$, we only preserve the probabilities of a subset $\mathbf{M}$ ($\mathbf{M} \subset \mathbf{C}$) of categories, while setting the probabilities of the rest categories to zeros. To match the original accuracy, the label of the top-1 prediction is always preserved in $\mathbf{M}$. Specifically, we set the new probability $\tilde{p}_\tau^T(k) = p_\tau^T(k) + \delta(k)$ if $k \in \mathbf{M}$, and 0 otherwise. The $\delta(k)$ is added to ensure that the adjusted probabilities are properly normalized (i.e., $\sum_k \tilde{p}_\tau^T(k) = 1$). Let $N = |\mathbf{M}|$ with $1 \leq N < K$. We define $r = \frac{N}{K}$ as the sparse ratio of $\tilde{p}_\tau^T(k)$.

Moreover, (Ma et al., 2021) finds that the accuracy of the student network is much worse when learning from the nasty logits with a larger τ. This observation suggests that a large temperature is also a vital factor for the success of the nasty teacher. Typically, with a large τ, the output will be very soft and similar to a uniform distribution (Hinton et al., 2015), especially when the total number of class K is large. Therefore, we assume that all $p_\tau^T(k)$ (except for the top-1 prediction) are equal when τ is large, and use a uniform distribution to approximate them for simplicity. Let j be the class of the top-1 prediction, $p_\tau^T(k)$ can be approximated by:

$$p_\tau^T(k) \approx \begin{cases} \frac{1}{K} - \epsilon, & \text{if } k \neq j \\ \frac{1}{K} + \epsilon(K-1), & \text{if } k = j \end{cases} \tag{3}$$

where ϵ is sufficiently small ($0 < \epsilon \ll \frac{1}{K}$). Thus, the residual value $\delta(k)$ can be approximated as $\delta(k) \approx \dfrac{1-r}{rK}$ for all $k \in \mathbf{M}$. The detailed derivation process is presented in Appendix A1. When the student learns from the sparse probabilities $\tilde{p}_\tau^T(k)$, the KL divergence in Eq. 1 is rewritten as [1] :

$$\begin{aligned} \mathcal{KL}(\tilde{p}_\tau^T, p_\tau^S) &= -\sum_{k=1}^{K} \tilde{p}_\tau^T(k) \log p_\tau^S(k) = -\sum_{k\in\mathbf{M}} (p_\tau^T(k) + \delta(k)) \log p_\tau^S(k) \\ &\approx -\sum_{k\in\mathbf{M}} p_\tau^T(k) \log p_\tau^S(k) - \frac{1-r}{r} \sum_{k\in\mathbf{M}} \frac{1}{K} \log p_\tau^S(k) \\ &\approx -\frac{1}{rK} \sum_{k\in\mathbf{M}} \log p_\tau^S(k) \end{aligned} \tag{4}$$

For the first approximation, we replace $\delta(k)$ with $\frac{1-r}{rK}$, and for the second approximation, we replace $p_\tau^T(k)$ with $\frac{1}{K}$. Thus, when learning from the sparse logits, the loss function in Eq. 1 is rewritten as:

$$\tilde{\mathcal{L}}_{KD} = (1-\alpha)\mathcal{H}(p^S, y) + \frac{\alpha\tau^2}{rK} \left[-\sum_{k\in\mathbf{M}} \log p_\tau^S(k) \right] \tag{5}$$

Compared with learning from the hard label, the second term in Equation 5 equally maximizes the probabilities of all classes within the subset $\mathbf{M}$. This term forces the model produce high responses on all categories within the subset $\mathbf{M}$. A sparse logit (i.e., $r < 0.1$) leads to a large weight $1/r$, making the student model spend more effort to optimize the second term. Consequently, the student model cannot identify the difference between categories within the subset $\mathbf{M}$ and undoubtedly give a wrong prediction. Similarly, a larger α or τ leads to the same effect.

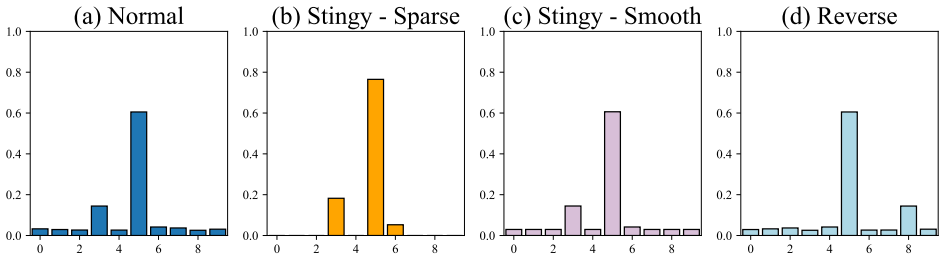


Figure 1: Illustration of different types of logits. (a) the normal logits. (b) the stingy teacher, (c) We present the probabilities after the temperature-scaled softmax.

3.3 STINGY TEACHER

The retraining process of the nasty teacher inevitably hurts the performance of the teacher network itself (Ma et al., 2021). Based on the above theoretical analysis, we propose a new method that directly manipulates the output logits of a pre-trained model to achieve the effect of the nasty teacher, named *Stingy teacher*. Given the logits z_k^T from the pre-trained model T, the stingy teacher directly keeps N logits with relatively high value to build the subset $\mathbf{M}$, and discards the rest. Thus, the

[1] we intend to discard the entropy of $\tilde{p}_\tau^T(k)$ as it does not contribute to the gradient of student S

logit still maintains the similarity structure among categories, but it is "stingy" as it only provides the information of a few categories. Specifically, the logit z_i^{ST} of the stingy teacher, named "stingy logits", is defined as:

$$z_k^{ST} = \begin{cases} z_k^T, & \text{if } k \in \mathbf{M}^{ST} \\ -\inf, & \text{if } k \notin \mathbf{M}^{ST} \end{cases} \tag{6}$$

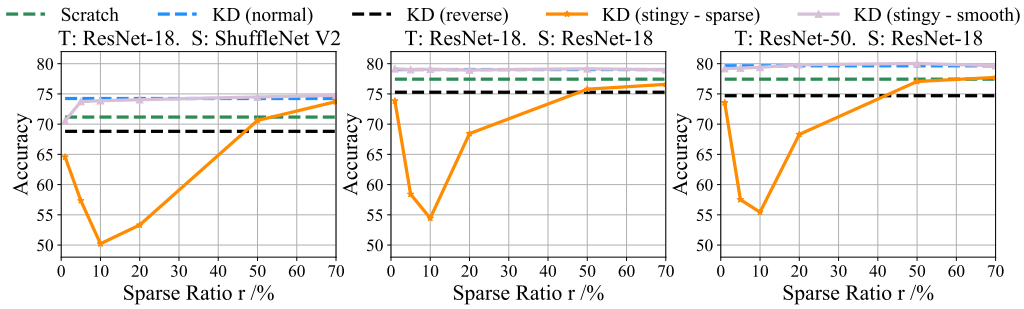
For any τ, the soft probability $p_\tau^{ST}(k)$ is 0 if $k \notin \mathbf{M}$. Figure 1 (b) presents an example of the stingy teacher. Noticeably, the relationships with highly relevant categories are still preserved. Thus the stingy logits is also useful in practice.

4 Experiments

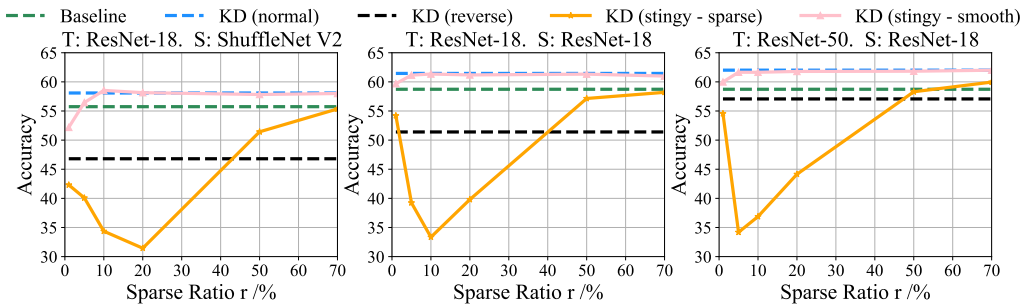
4.1 Settings

Dataset and Architecture Following (Ma et al., 2021), we conduct experiments on CIFAR-10, CIFAR-100 and Tiny-ImageNet. We consider three networks from ResNet family (He et al., 2016), i.e., ResNet-18, ResNet-50 and ResNeXt-29 (Xie et al., 2017) as teacher networks, and three widely used light-weight networks, i.e., MobileNetV2 (Sandler et al., 2018), ShuffleNetV2 (Ma et al., 2018) and ResNet-18, as student networks. Meanwhile, we also consider the teacher network itself as the student model, as the self-KD (Yuan et al., 2020) also improves the performance. Moreover, we firstly scale up the setting of nasty teacher to ImageNet. Following (Yuan et al., 2020), we use DenseNet-121 (Huang et al., 2017) as teacher and ResNet-18 as students.

Training Following (Ma et al., 2021; Yuan et al., 2020), we set the temperature τ to 20, and the balance factor α to 0.9 for all experiments. For CIFAR-10, CIFAR-100 and Tiny-ImageNet, all networks are optimized by SGD with momentum 0.9 and weight decay $5\text{e}{-4}$.The total number of training epochs is 200. The initial learning rate is 0.1, and is decayed by a factor of 5 at the 60th, 120th and 160th epoch respectively. For ImageNet, we follow settings in (Yuan et al., 2020) and train the network with 90 epochs in total. The initial learning rate is 0.1, and is decayed by 10 at the 30th, 60th, and 80th epoch.



(a) CIFAR-100.



(b) Tiny-ImageNet.

Figure 2: Comparison of KD from three types of modification on the logits: the "stingy -sparse" logits, the "stingy - smooth" logits, and the reversed logits. We conduct experiments with three teacher student pairs on both CIFAR-100 and Tiny-ImageNet. We also present the accuracy of training from scratch and KD from a normal teacher for comparison.

4.2 THE EFFECT OF SPARSE LOGITS

To explore the importance of sparsity, we compare our stingy teacher (*"stingy - sparse"*) with two different types of modification on the logits. 1) We keep the same subset of logits, but replace the rest of logits with their average. We denote it as *"stingy - smooth"* for simplicity; 2) We keep the top-1 prediction, but reverse the value of the rest of the logits. We denote it as *"reverse"* for simplicity. The first type is also stingy, and the second type introduces misleading information. Nevertheless, both of them still maintain the smoothness property of the logits. Figure 1 (c)(d) give an illustration of them. We conduct experiments with three teacher-student pairs on both CIFAR-100 and Tiny-ImageNet to ensure generalization.

Results We explore the relationship between the sparse ratio r and the performance of student networks distilling from each types of logits. The results are presented in Figure 2. Firstly, we notice that when the logits is smooth, even if the dark knowledge is scarce (stingy - smooth), the accuracy of the student networks still improves or at least is on a par. This is consistent with the conclusion in (Yuan et al., 2020) that KD plays the role of label smooth regularization. Secondly, when the dark knowledge is misleading (reverse), the accuracy of student can be downgraded 5% to 8%. When the capacity of student is huge, the damage is mitigated. This suggests that a noisy dark knowledge is harmful for lightweight student networks. Thirdly, the accuracy of the student model is significantly degraded when the student model learning from the sparse logits, whatever the capacity of the student model. When sparse ratio r is around 10%, most students achieve the worst performance, i.e, more than 20% accuracy drop compared with training from scratch. When r is extremely small, such as $r = \frac{1}{K}$, students can recover some accuracy. We hypothesize that it is relatively easy to learn when the size of the misleading candidate set is small. All of the experiments support our claim that sparsity is the major reason leads to the accuracy drop of student networks, compared with noise.

Analysis We also present the testing accuracy during the training process of student networks. Figure 3 shows that the training of students converges earlier when learning from the stingy teacher. This suggests that the sparse logits are harmful to the training of student networks, rather than they are hard to be learned.

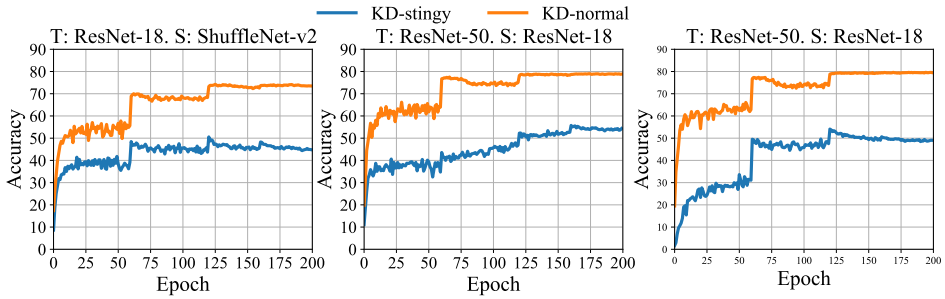


Figure 3: Testing accuracy over epochs for student networks trained with KD on CIFAR-100.

4.3 COMPARISON WITH NASTY TEACHER

We then apply the surprising property of sparse logits to the settings of the nasty teacher (Ma et al., 2021). A better nasty teacher should match the accuracy of its original normal counterpart as much it can, while degrading the accuracy of student models learning from it as much as possible. Based on the conclusion in Section 4.2, we empirically set the sparse ratio to 0.1 for all experiments. For CIFAR-10, we set the sparse ratio to 0.2, cause 0.1 reduces the logits to a hard label with 10 classes. For other hyperparameters, we strictly follow settings in (Ma et al., 2021) for a fair comparison.

Results on CIFAR and Tiny-ImageNet Table 1, Table 2 and Table 3 show the comparison of the nasty teacher and our stingy teacher on CIFAR-10, CIFAR-100, and Tiny-ImageNet, respectively. From the teacher's aspect, the performance of the stingy teacher always matches that of the normal teacher perfectly, as we only manipulate the logits of the original teacher and keep the logits with the highest probability. However, the performance of the nasty teacher can be degraded up to 2%, which may be unacceptable in some situations. Moreover, the retraining process of the nasty teacher

Table 1: Comparison of the nasty teacher and the stingy teacher on CIFAR-10.

Teacher network	Teacher accuracy	Students accuracy after KD			
		CNN	ResNetC-20	ResNetC-32	ResNet-18
Student baseline	-	86.64	92.28	93.04	95.13
ResNet-18 (normal)	95.13	87.75 (+1.11)	92.49 (+0.21)	93.31 (+0.27)	95.39 (+0.26)
ResNet-18 (nasty)	94.56 (-0.57)	71.83 (**-14.81**)	74.22 (-18.06)	79.66 (-13.38)	91.55 (**-3.58**)
ResNet-18 (stingy)	95.13 (**-0.0**)	82.77 (-3.87)	68.86 (**-25.42**)	74.34 (**-18.70**)	92.46 (-2.67)

Table 2: Comparison of the nasty teacher and the stingy teacher on CIFAR-100.

Teacher network	Teacher accuracy	Students accuracy after KD			
		Shufflenetv2	MobilenetV2	ResNet-18	Teacher Self
Student baseline	-	71.17	69.12	77.44	-
ResNet-18 (normal)	77.44	74.24 (+3.07)	73.11 (+3.99)	79.03 (+1.59)	79.03 (+1.59)
ResNet-18 (nasty)	77.42 (-0.02)	64.49 (-6.68)	3.45 (**-65.67**)	74.81 (-2.63)	74.81 (-2.63)
ResNet-18 (stingy)	77.44 (**-0.00**)	50.22 (**-20.95**)	6.78 (-62.34)	54.44 (**-23.00**)	54.44 (**-23.00**)
ResNet-50 (normal)	78.12	74.00 (+2.83)	72.81 (+3.69)	79.65 (+2.21)	80.02 (+1.96)
ResNet-50 (nasty)	77.14 (-0.98)	63.16 (-8.01)	3.36 (**-65.76**)	71.94 (-5.50)	75.03 (-3.09)
ResNet-50 (stingy)	78.12 (**-0.00**)	49.05 (**-22.12**)	5.52 (-63.60)	55.44 (**-22.00**)	55.63 (**-22.49**)
ResNeXt-29 (normal)	81.85	74.50 (+3.33)	72.43 (+3.31)	80.84 (+3.40)	83.53 (+1.68)
ResNeXt-29 (nasty)	80.26 (-1.59)	58.99 (-12.18)	1.55 (**-67.57**)	68.52 (-8.92)	75.08 (-6.77)
ResNeXt-29 (stingy)	81.85 (**-0.00**)	49.46 (**-21.71**)	6.93 (-62.19)	58.70 (**-18.74**)	54.18 (**-27.67**)

also introduces additional computation to the teacher networks. As a result, our stingy teacher is a better choice than the nasty teacher for the owner of teacher networks.

From the student's aspect, both nasty and stingy teachers make some fragile student networks learning from it untrainable, such as MobileNet. Moreover, the accuracy of student networks can be further degraded when distilling from the stingy teacher. Some sophisticated student networks, such as the same architecture as the teacher, can only be degraded up to 6.77% when learning from the nasty teacher. However, when learning from the stingy teacher, they can be degraded up to 28.14%. Thus, our stingy teacher is more catastrophic to large student networks. In conclusion, the stingy teacher can significantly downgrade the performance of any network trying to clone the protected networks without sacrificing its own accuracy.

Visualizations of logits Figure 4 compares the soft probabilities produced by the normal teacher, the nasty teacher, and the stingy teacher respectively. Compared with the normal response, the nasty logit is very noisy and it significantly increases the probabilites of some irrelevant classes. As a result, it is easy to be identified by the attacker. Oppositely, the stingy logits still maintain the relatively relationships among the top categories, so it still provides the normal function as the original network.

Results on ImageNet The accuracy of the nasty teacher is sensitive to the adversarial weights ω in the retraining process, thus the model owner need to pay lots of effort to exploring the best ω. On the contrary, without the retraining process, our stingy teacher can be easily scaled up to huge datasets. We evaluate the stingy teacher on ImageNet. As shows in Table 4, when distilling from a stingy DenseNet-121, the accuracy of students can be degraded up to 37.55%. As a result, our stingy teacher is also more favorable for protecting huge models in real applications.

4.4 Ablation Studies

Sample of subset The stingy teacher preserves the logits of the top N categories to maintain the dark knowledge. We denote it as *"top logits"* for short. In this section, we explore the performance of other possibility to build the subset $\mathbf{M}$. Specifically, we design another sparse logits that concatenates the top one logits and the N - 1 smallest logits, and we denote it as *"least logits"* for short. The

Table 3: Comparison of the nasty teacher and the stingy teacher on Tiny-ImageNet

Teacher network	Teacher accuracy	Students accuracy after KD			
		Shufflenetv2	MobilenetV2	ResNet-18	Teacher Self
Student baseline	-	55.74	51.72	58.73	-
ResNet-18 (normal)	58.73	58.09 (+2.35)	55.99 (+4.27)	61.45 (+2.72)	61.45 (+2.72)
ResNet-18 (nasty)	57.77 (-0.96)	23.16 (**-32.58**)	1.82 (**-49.90**)	44.73 (-14.00)	44.73 (-14.00)
ResNet-18 (stingy)	58.73 (**-0.00**)	34.36 (-21.38)	5.55 (-46.17)	33.34 (**-25.39**)	33.34 (**-25.39**)
ResNet-50 (normal)	62.01	58.01 (+2.27)	54.18 (+2.46)	62.01 (+3.28)	63.91 (+1.90)
ResNet-50 (nasty)	60.06 (-1.95)	41.84 (-13.90)	1.41 (**-50.31**)	48.24 (-10.49)	51.27 (-10.74)
ResNet-50 (stingy)	62.01 (**-0.00**)	28.03 (**-27.71**)	5.41 (-46.31)	37.05 (**-21.68**)	34.26 (**-27.75**)
ResNeXt-29 (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)
ResNeXt29 (stingy)	62.81 (**-0.00**)	30.98 (**-24.76**)	9.65 (-42.07)	30.70 (**-28.03**)	34.67 (**-28.14**)

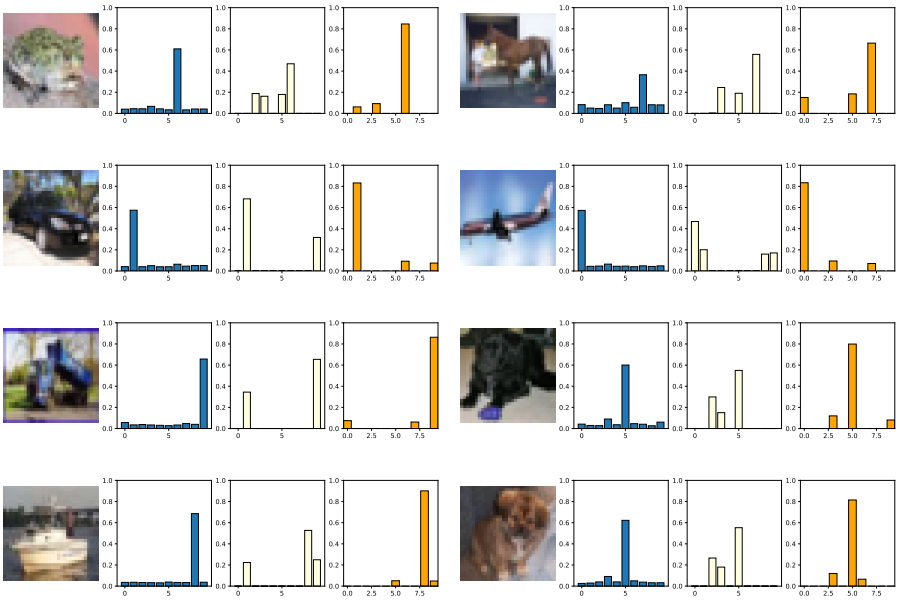


Figure 4: The visualization of logit responses produced by a normal ResNet-18 (blue), a nasty ResNet-18 (yellow), and a stingy ResNet-18 (orange) trained on CIFAR-10. We present the probabilities after temperature-scaled softmax, where τ is 4.

least logits can be regarded as the worst sparse logits, as it masks out all meaningful dark knowledge, and enlarges the least related categories instead.

Figure 5 presents the comparison results. Obviously, at around the 10% sparse ratio, both top logits and least logits achieve the greatest damage to the student networks. This is consistent with our analysis in Eq. 5 that when r is small, the sparse logits should be able to degrade the student networks, whatever subset we use. When r is equal to $\frac{1}{K}$, both of them degenerate into the hard label, thus they have the same performance. When r is large, the damage from both types of logits is alleviated, as the weight on the second term of Eq. 5 is reduced. However, the least logits always leads to a worse accuracy of students than the top logits. We believe that the irrelevant classes in $\mathbf{M}$ provide harmful interference to the learning of students, and make the learning much difficult. This also reveals that dark knowledge is beneficial to the student networks. Although the performance of the least logits is promising, considering the similarity to the original logits, the top logits is still a favorable choice of the stingy teacher.

Table 4: Experimental results on ImageNet.

Model	Baseline	self-KD	KD (normal T)	KD (stingy T)
ResNet-18	69.84	70.42 (+0.58)	70.40 (+0.56)	32.29 (**-37.55**)

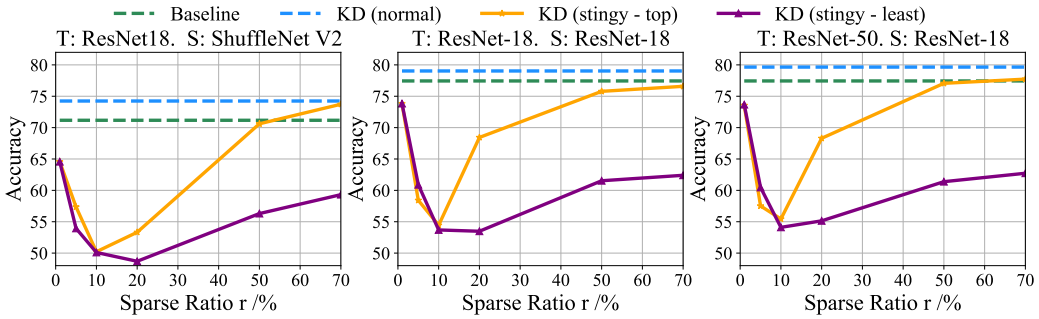


Figure 5: Comparison of sparse logits built with top N categories (top logits) and the combination of the top-1 class and N-1 smallest probabilities (least logits). Experiments are conducted on CIFAR-100.

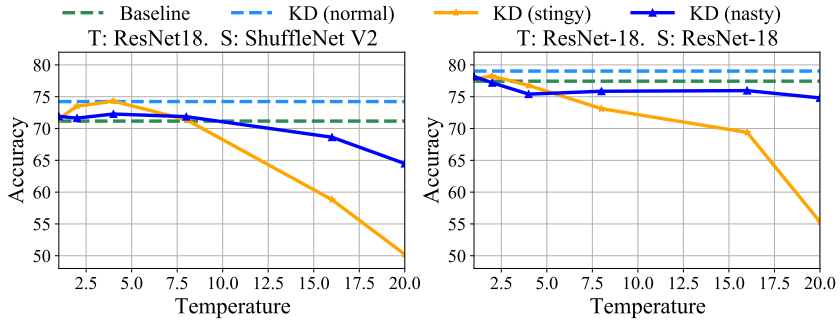


Figure 6: Ablation studies on temperature τ. Experiments are conducted on CIFAR-100 with ResNet-18 as the teacher network.

Temperature The performance of student networks would be degraded more by the nasty teacher with a larger temperature τ. We also conduct ablation studies on τ to explore this conclusion on the stingy teacher. In detail, we keep the sparse ratio $r = 0.1$ and vary the temperature τ_s from 1 to 20. As in Figure 6, with a larger temperature, the student can also be further degraded when learning from the stingy teacher. When reducing τ, the student networks can recover some performance. As supported by Equation 5, a small τ turns the weights of the second term down, and thus mitigates its negative effect. Moreover, we cannot approximate the soft probabilities $p_\tau^T(k)$ with the uniform distribution when τ is small.

5 Conclusion

In this paper, we demonstrate that the sparsity of the logits is the main reason for the accuracy drop of student networks in the setting of the nasty teachers. Based on this property, we propose a simple yet effective way to achieve the effect of the nasty teacher, named stingy teacher, which simply keeps a small subset of top logits and zeros out the rest. Extensive experiments on CIFAR-10, CIFAR-100, and Tiny-ImageNet demonstrate that our stingy teacher is more catastrophic to student networks. Moreover, with the stingy teacher, we can scale up the setting of the nasty teacher to very large datasets, such as ImageNet. Our work clarifies some effects of knowledge distillation. The method proposed here is very practical and can help address the increasingly important issue of deep learning model protection.

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.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Long live the lottery: The existence of winning tickets in lifelong learning. In *International Conference on Learning Representations*, 2021a. URL https://openreview.net/forum?id=LXMSvPmsm0g.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *International Conference on Learning Representations*, 2021b. URL https://openreview.net/forum?id=qZzy5urZw9.

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

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.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

Guilin Li, Junlei Zhang, Yunhe Wang, Chuanjian Liu, Matthias Tan, Yunfeng Lin, Wei Zhang, Jiashi Feng, and Tong Zhang. Residual distillation: Towards portable deep neural networks without shortcuts. *Advances in Neural Information Processing Systems*, 33, 2020.

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

Haoyu Ma, Tianlong Chen, Ting-Kuei Hu, Chenyu You, Xiaohui Xie, and Zhangyang Wang. Undistillable: Making a nasty teacher that {cannot} teach students. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=0zvfm-nZqQs.

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.

Rafael Müller, Simon Kornblith, and Geoffrey Hinton. When does label smoothing help? *arXiv preprint arXiv:1906.02629*, 2019.

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.

Gabriel Pereyra, George Tucker, Jan Chorowski, Łukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions. *arXiv preprint arXiv:1701.06548*, 2017.

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.

Zhiqiang Shen, Zechun Liu, Dejia Xu, Zitian Chen, Kwang-Ting Cheng, and Marios Savvides. Is label smoothing truly incompatible with knowledge distillation: An empirical study. *arXiv preprint arXiv:2104.00676*, 2021.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

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.

A1 DETAILED DERIVATION

In this appendix, we give detailed derivation process of Eq. 4. When the temperature τ is high, we can use a uniform distribution to approximate $p_\tau^T(k)$. As in Eq. 3, the soft probabilities $p_\tau^T(k)$ can be approximated with:

$$p_\tau^T(k) \approx \begin{cases} \frac{1}{K} - \epsilon, & \text{if } k \neq j \\ \frac{1}{K} + \epsilon(K-1), & \text{if } k = j \end{cases} \tag{A1}$$

Where ϵ is a neglectable factor ($0 \leq \epsilon \ll \frac{1}{K}$), and j is the original top-1 prediction. The new sparse logits $\tilde{p}_\tau^T(k)$ is defined with:

$$\tilde{p}_\tau^T(k) \approx \begin{cases} p_\tau^T(k) + \delta(k), & \text{if } k \in \mathbf{M} \\ 0, & \text{if } k \notin \mathbf{M} \end{cases} \tag{A2}$$

Once zeroing out the probabilities of class $k \notin \mathbf{M}$, the total discarded probabilities $\mathcal{P}_1$ can be derived by:

$$\mathcal{P}_1 = (K-N)p_\tau^T(k) = (K-N)\left(\frac{1}{K} - \epsilon\right) = (K - rK)\left(\frac{1}{K} - \epsilon\right) = (1-\epsilon K)(1-r) \tag{A3}$$

Again, $r = \frac{N}{K}$ is the sparse ratio, and N is the total number of element in the subset $\mathbf{M}$. Thus, the total preserved probabilities of class $k \in \mathbf{M}$ is

$$\mathcal{P}_2 = 1 - \mathcal{P}_1 = 1 - (1-\epsilon K)(1-r) = r + \epsilon K(1-r) \tag{A4}$$

To ensure the sum of $\tilde{p}_\tau^T(k)$ is equal to 1, we need re-normalize the probabilities of the preserved categories $k \in \mathbf{M}$. Here we just consider the simplest additional way, and distribute $\mathcal{P}_1$ onto class $k \in \mathbf{M}$ based on the original probability $p_\tau^T(k)$. Thus, the additional term $\delta(k)$ can be writen as

$$\delta(k) = \frac{p_\tau^T(k)}{\mathcal{P}_2}\mathcal{P}_1 \tag{A5}$$

Specifically, when $k \neq j$, we have

$$\delta(k) = \frac{p_\tau^T(k)}{\mathcal{P}_2}\mathcal{P}_1 = \frac{\frac{1}{K} - \epsilon}{r + \epsilon K(1-r)}(1-\epsilon K)(1-r) = \frac{(1-\epsilon K)^2}{r + \epsilon K(1-r)}\frac{1-r}{K} \tag{A6}$$

Since $\epsilon \ll \frac{1}{K}$, we have $0 \leq \epsilon K \ll 1$, thus $(1-\epsilon K)^2 \approx 1$. As $r < 1$, $r + \epsilon K(1-r) < r + \epsilon K \approx r$. Therefore, we can approximate $\delta(k)$ with

$$\delta(k) = \frac{(1-\epsilon K)^2}{r + \epsilon K(1-r)}\frac{1-r}{K} \approx \frac{1-r}{rK} \tag{A7}$$

When $k = j$, we have

$$\delta(j) = \mathcal{P}_1 - (N-1)\delta(k)_{k\in\mathbf{M}, k\neq j} \approx (1-r) - (rK-1)\frac{1-r}{rK} = \frac{1-r}{rK} \tag{A8}$$

In conclusion, we can approximate $\delta(k)$ with $\dfrac{1-r}{rK}$ for any $k \in \mathbf{M}$.