SELF-ENSEMBLE PROTECTION: TRAINING CHECKPOINTS ARE GOOD DATA PROTECTORS

Anonymous authors
Paper under double-blind review

ABSTRACT

As data become increasingly vital for deep learning, a company would be very cautious about releasing data. This is because the competitors could use the released data to train high-performance models, thereby posing a tremendous threat to the company's commercial competence. To protect the dataset from unauthorized use for training, imperceptible perturbations crafted with a deep model are added to data so that other deep neural networks trained on it all have poor generalization. In this paper, we propose a self-ensemble protection (SEP) method to take advantage of intermediate checkpoints in a single training process for data protection. Contrary to the popular belief on the similarity of checkpoints, we are surprised to find that their cross-model gradients are close to orthogonal, and thus diverse enough to produce very effective protective perturbations. Besides, we further improve the performance of SEP by developing a novel feature alignment technique to induce feature collapse into the mean of incorrect-class features. Extensive experiments verify the consistent superiority of SEP over 7 state-of-the-art data protection baselines. SEP perturbations on CIFAR-10 with an ℓ_∞ bound as small as $2/255$ could reduce the testing accuracy of a ResNet18 from 94.56% to 14.68%, and the average accuracy reduction from the best-known results is 27.63%. Under the $\ell_\infty = 8$ bound, SEP perturbations lead DNNs with 5 architectures to have less than 5.7% / 3.2% / 0.6% accuracy on CIFAR-10 / CIFAR-100 / ImageNet subset.

1 INTRODUCTION

Large-scale datasets have become increasingly important in training high-performance deep neural networks (DNNs). Thus, it is a common practice to collect data online (Mahajan et al., 2018; Sun et al., 2017), an almost unlimited data source. This poses a great threat to the commercial competence of data owners such as social media companies since the competitors could also train good DNNs from their data. Therefore, great efforts have been devoted to protecting data from unauthorized use in model training. The most typical way is to secure the data by adding imperceptible perturbations to it, so that DNNs trained on the secured data have poor generalization (Feng et al., 2019; Huang et al., 2020a; Fowl et al., 2021b). As shown in Fig. 1, the images in the red box differ little from those in the green box, but could decrease a well-trained DNN's test accuracy from 95% to 20%.

Existing data protection methods use a single DNN to generate incorrect but DNN-sensitive features (Huang et al., 2020a; Fu et al., 2021; Fowl et al., 2021b) for training data by, *e.g.*, adversarial attacks (Madry et al., 2018). However, the data protectors cannot know what kind of DNNs the unauthorized users will use in advance, so it is desirable to make the protection robust to different DNN architectures. Inspired by the outstanding performance of ensemble methods (Liu et al., 2017; Dong et al., 2018) in adversarial attacks, using an ensemble method to generate secured data could be a promising solution for data protection, but it has never been explored in prior literature. One challenging problem is that the conventional ensemble methods require time- and resource-consuming training processes to obtain a large number of ensemble models. This inefficiency significantly hinders the use of ensemble methods in practice, especially for the scale of data worth protection.

To tackle this problem, we note that during the training process of a single model, many intermediate checkpoint models are inevitably generated, which is a natural source to get a large number of different models. Therefore, we raise a question:

*Can we use these intermediate checkpoint models for data protection in a **self-ensemble** manner?*

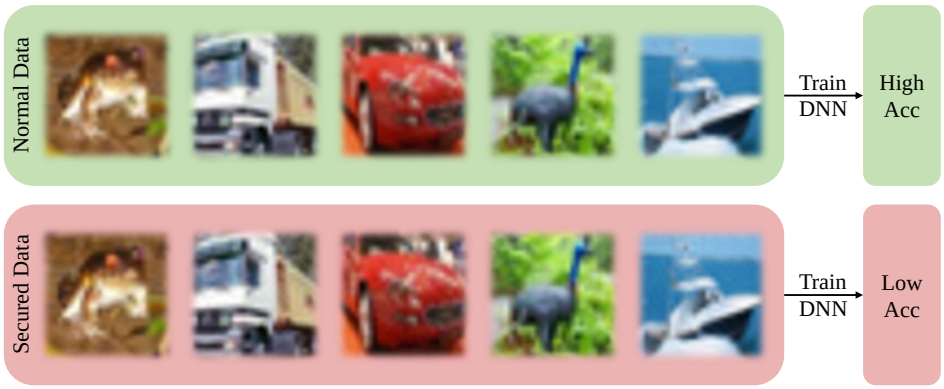


Figure 1: Effect of protective perturbations. Although a DNN could achieve high accuracy by learning on the normal data, it cannot reach an acceptable accuracy by training on the secured data, which only differs very slightly from the former. Samples are crafted by our method for CIFAR-10 (Krizhevsky et al., 2009) images using checkpoints in training a ResNet18 (He et al., 2016) with $\ell_\infty = 2$.

If this works, we could avoid the tremendous costs by training many models individually, while achieving high data protection performance. However, the conclusion is still too early to draw given the unique characteristics of the intermediate models. That is, the intermediate models from the same training process are similar to a certain degree: they have the same architecture with parameters falling in a low-dimensional subspace (Li et al., 2022). This might lead to compromised performance and impede the incorporation of the self-ensemble in data protection.

To answer the above question, we first explore and analyze the similarities of the checkpoint models obtained during a model's training process. We find that intermediate checkpoints indeed have high similarity in terms of function value, which is why their outputs are similar. However, contrary to popular belief, we excitingly find that they have low similarity in terms of gradient, which is the key information for adversarial attack and data protection. In fact, as illustrated in Fig. 2, the gradients on the same image are nearly orthogonal for different checkpoints, which is well-accepted indicator for good model diversity in ensemble (Yang et al., 2021). In this regard, we propose the so-called self-ensemble protection (SEP) to secure datasets efficiently and effectively.

We also bring about a novel technique named feature alignment (FA) to further boost the performance of SEP. FA is inspired by the neural collapse theory (Papyan et al., 2020) and we design a feature collapse loss to induce a sample's last-layer feature collapse into the mean of incorrect-class features. Compared to the traditional constraint of only misprediction (Carlini & Wagner, 2017; Fowl et al., 2021b), FA demands a sample to have the exact high-dimensional feature of the target-class samples. This stricter constraint robustly injects incorrect features so that DNNs are deeply confounded.

Equipping SEP with FA, our method achieves astonishing performance in data protection. By setting $\ell_\infty = 2$, a very small bound, SEP perturbations on the CIFAR-10 training set reduce the testing accuracy of a ResNet18 from 94.56% to 14.68%, while the best-known results could only reach 41.35% with the same amount of overall calculation to craft the perturbations. Similar results are also observable in the study on CIFAR-100 and ImageNet on 5 architectures and 7 training strategies.

Contributions

- We thoroughly analyze the similarities of the training checkpoints, and point out that despite their commonly-believed similarity in forward predictions, they differ greatly in backward gradients. Thus, we inventively introduce self-ensemble method into data protection domain.
- We propose a novel feature alignment (FA) strategy to further boost the performance of self-ensemble protection (SEP) by inducing incorrect neural collapse, *i.e.*, a sample should have exactly the mean feature of incorrect-class samples.
- We extensively evaluate SEP and find it exceeds existing state-of-the-art baselines significantly, even considering the computation budget. $\ell_\infty = 8$ SEP perturbations lead DNNs to have less than 5.7% / 3.2% / 0.6% accuracy on CIFAR-10 / CIFAR-100 / ImageNet subset.

2 Related Work

2.1 Adversarial attacks for data protection

It has been widely known that small perturbations could fool DNNs to make incorrect predictions (Szegedy et al., 2014; Carlini & Wagner, 2017). Such test-time adversarial perturbations are crafted most effectively by adversarially updating samples with model gradients (Goodfellow et al., 2015; Madry et al., 2018), and the produced adversarial examples transfer to hurt other DNNs as well (Dong et al., 2018; Chen et al., 2022). Similarly, training-time adversarial perturbations, *i.e.*, poisoning examples, are also obtainable by adversarially modify training samples using DNN gradients (Koh & Liang, 2017; Fowl et al., 2021b). All DNNs trained on poisoning examples generalize poorly on clean examples, making poisoning methods helpful in protecting data from unauthorized use of training. Early poisoning examples are apparently different from clean data (Biggio et al., 2012; Yang et al., 2017), limiting their practicality. Motivated by this, recent works (Koh & Liang, 2017; Fowl et al., 2021b) manage to craft imperceptible perturbations. Besides adversarial noise, it has been demonstrated that error-minimization (Huang et al., 2020a), gradient alignment (Fowl et al., 2021a) and influence functions (Fang et al., 2020) are also useful in protecting data. However, current methods only use one DNN to craft perturbations, leaving much room for improvements.

Two similar but different tasks besides poisoning-based data protection are adversarial training and backdoor attacks. Adversarial training (Madry et al., 2018; Zhang et al., 2019; Stutz et al., 2020) continuously generates adversarial examples with current checkpoint gradients to improve the model's robustness towards worst-case perturbations. In contrast, data protection produces fixed poisoning examples so that unauthorized training yields low clean accuracy. Backdoor attacks (Geiping et al., 2020; Huang et al., 2020b) perturb a small proportion of training set to make the DNNs mispredict certain samples, but remain well-functional on other clean samples. While data protectors perturb the whole training set to degrade the model's performance on all clean samples.

2.2 Ensemble methods in adversarial learning

Ensemble is validated as a panacea for boosting the generalization of test-time adversarial attacks (Liu et al., 2017; Dong et al., 2018). By aggregating the probabilities (Liu et al., 2017), logits or losses (Dong et al., 2018) of multiple models, ensemble attacks significantly increase the black-box attack success rate. Ensemble attacks could be further enhanced by reducing the gradient variance of sub-models (Xiong et al., 2022), and such an optimization way is also adopted in our method. Besides, ensemble has also been shown effective as a defense method by inducing low diversity across sub-models (Pang et al., 2019; Yang et al., 2020; 2021) or producing diverse adversarial examples in adversarial training (Tramèr et al., 2018; Wang & Wang, 2021). Despite the good performance of ensemble in attacks and defenses, they have not been introduced to protect datasets, because the scale of data worth protection is very large for training multiple models, which is required by the conventional ensemble. In this regard, we adopt the self-ensemble strategy, which resorts to training checkpoints to improve performance. Its current applications are mostly focused on semi-supervised learning (Zhao et al., 2019; Liu et al., 2022).

2.3 Feature attacks and neural collapse

Feature-space methods have been broadly proposed in adversarial attacks (Sabour et al., 2016; Inkawhich et al., 2019b;a; 2020) and backdoor attacks (Shafahi et al., 2018). Since proposal (Sabour et al., 2016), different feature attacks are developed to enhance the adversarial transferability by shifting the attack over well-separated layers (Inkawhich et al., 2019b), using an auxiliary DNN to learn layer-wise feature distribution (Inkawhich et al., 2019a), or aggregating of multiple layer feature information (Inkawhich et al., 2020). By enforcing feature consistency, effective backdoor samples could be obtained (Shafahi et al., 2018) for transfer learning. However, to the best of our knowledge, the potential of feature-space loss has not been explored by data protection methods, which demand samples to possess incorrect non-robust features (Fowl et al., 2021b; Ilyas et al., 2019) as in adversarial attacks. Inspired by the recent Neural Collapse (Papyan et al., 2020) theory, which demonstrates that the last-layer features of a well-trained DNN collapse to in-class feature mean, we devise the first feature-space poisoning attacks by inducing incorrect neural collapse, *i.e.*, encouraging a sample to have features close to the mean of target-class features.

3 THE PROPOSED METHOD

3.1 PRELIMINARIES

We first introduce the threat model and problem formulation of the data protection task. The data owner company wishes to release data for users while preventing an unauthorized *appropriator* to collect data for training DNNs. Thus, the data *protector* would add imperceptible perturbations to samples, so that humans have no obstacle to seeing the data, while the appropriator cannot train DNNs to achieve an acceptable testing accuracy. Since the protector has access to the whole training set, it can craft perturbations for each sample for effective protection (Shen et al., 2019; Feng et al., 2019; Huang et al., 2020a; Fowl et al., 2021b). Mathematically, the problem can be formulated as

$$\max_{\boldsymbol{\delta}\in\Pi_\varepsilon} \sum_{(\boldsymbol{x},y)\in\mathcal{D}} \mathcal{L}_{\text{CE}}(f_{\text{a}}(\boldsymbol{x},\theta^*),y), \quad \text{s.t. } \theta^* \in \arg\min_{\theta} \sum_{(\boldsymbol{x},y)\in\mathcal{T}} \mathcal{L}_{\text{CE}}\left(f_{\text{a}}\left(\boldsymbol{x}+\boldsymbol{\delta},\theta\right),y\right), \tag{1}$$

where the perturbations $\boldsymbol{\delta}$ bounded by ε are added to the training set $\mathcal{T}$ so that an appropriator model trained on it $f_{\text{a}}(\cdot,\theta^*)$ have a low accuracy on the test set $\mathcal{D}$, *i.e.*, a high cross-entropy loss $\mathcal{L}_{\text{CE}}(\cdot,\cdot)$.

There are two representative ideas to calculate $\boldsymbol{\delta}$: error-minimizing (Huang et al., 2020a) and targeted attacks (Fowl et al., 2021b). The former fits the samples to a DNN that is not well-trained so that the data learns how to spoil the DNN to have high training accuracy easily without a good testing accuracy. The latter uses a well-trained protecting DNN f_{p} to produce targeted adversarial examples that have the non-robust features in the incorrect class $g(y)$ as

$$\boldsymbol{x}_{t+1} = \Pi_\epsilon\left(\boldsymbol{x}_t - \alpha\cdot\text{sign}\left(\boldsymbol{G}_{\text{CE}}(f_{\text{p}},\boldsymbol{x}_t)\right)\right), \quad \boldsymbol{G}_{\text{CE}}(f_{\text{p}},\boldsymbol{x}) = \nabla_{\boldsymbol{x}}\mathcal{L}_{\text{CE}}\left(f_{\text{p}}\left(\boldsymbol{x}\right),g(y)\right), \tag{2}$$

where Π_ϵ clips the sample into the ε ℓ_∞-norm bound after each update with a step size α. $g(\cdot)$ stands for a permutation on the label space. The optimization in (2) follows the update of projected gradient descent (Madry et al., 2018) in adversarial attacks. Because targeted attack perturbations are much more effective than error-minimization ones (Fowl et al., 2021b), our method is also based on this idea to inject data with non-robust features (Ilyas et al., 2019).

3.2 SELF-ENSEMBLE

Current methods (Shen et al., 2019; Feng et al., 2019; Huang et al., 2020a; Fowl et al., 2021b) craft protective perturbations that are supposed to generalize to poison different unknown architectures by a single DNN f_{p}. However, the ensemble of multiple models, the panacea to boost the generalization, has not been introduced to protect data, because training multiple models, as required in the conventional ensemble, is resource-consuming, especially for the scale of data worth protection.

In this paper, we propose to improve data protection by an efficient way of ensemble, *i.e.*, adopting the checkpoints during a model's training process as sub-models for self-ensemble. As shown in Fig. 2 (left), compared to the conventional ensemble that trains different models (5 columns), self-ensemble only trains one model (1 column), requiring equivalent training computation to single-model methods.

Despite the efficiency, it would be natural to wonder about the efficacy of self-ensemble, because intermediate models have not only the same architecture, but also the parameters falling in a low-dimensional subspace (Li et al., 2022). However, we note that the above existing descriptions of the checkpoints' similarity focus on function level (their predictions are similar) or parameter level (the weight trajectory is lazy). In data protection, DNNs are used to provide gradients as in adversarial attacks, seeing (2). Thus, it is more reasonable to analyze checkpoint gradients in data protection.

In this concern, we study the gradient direction on images for equidistant intermediate checkpoints. In Fig. 2 (upper right), we plot the average of absolute cosine value for gradients in different checkpoints, and the low value indicates that gradients on images are close to orthogonal for different checkpoints like in the ensemble of different architectures (bottom right). Since the gradient is the main information for data protection and orthogonality is a well-accepted measure for model diversity in ensemble (Yang et al., 2021), sub-models in self-ensemble are sufficiently diverse for data protection. Accordingly, we propose the self-ensemble protection (SEP) to secure datasets as

$$\boldsymbol{G}_{\text{SEP}}(f_{\text{p}},\boldsymbol{x}) = \sum_{k=0}^{n-1} \boldsymbol{G}_{\text{CE}}(f_{\text{p}}^k,\boldsymbol{x}), \tag{3}$$

where f_{p}^k is the kth equidistant intermediate checkpoint and $\boldsymbol{G}_{\text{SEP}}$ is the gradients for update in (2).

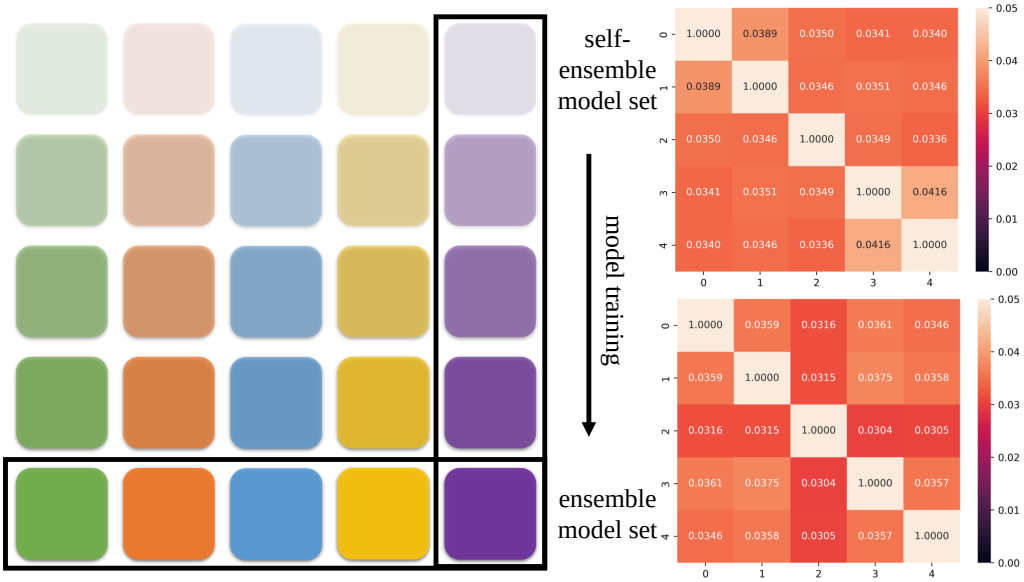


Figure 2: Ensemble and Self-Ensemble. Ensemble trains multiple models (left to right and top to bottom), while self-ensemble trains once and collects intermediate checkpoints (only top to bottom). Thus, self-ensemble costs much less training calculation. Moreover, checkpoints could provide diverse gradients. The right figures show the average absolute cosine value of gradients (on 1K CIFAR-10 images) for each model pair in ensemble (bottom) and self-ensemble (top) sub-models. In ensemble, models 0 to 4 stand for ResNet18, SENet18, VGG16, DenseNet121, and GoogLeNet, respectively, and in self-ensemble, they mean the ResNet18 after training for 24, 48, 72, 96, 120 epochs. Gradient analysis on CIFAR-100 and ImageNet are in Appendix B.

3.3 Feature Alignment

In the above analysis, we showcase that intermediate checkpoints are good data protectors, and results in Sec. 4 also support our expectation of SEP. But we aim a step further to make SEP as good as the ensemble of different architectures. Therefore, we resort to the recent neural collapse theory (Papyan et al., 2020) to robustly inject samples with incorrect features.

Neural collapse illustrates that the last-layer (before linear layer) features of a well-trained DNN collapse to in-class feature means. That is to say, the feature mean contains representative and high-dimensional information about all samples in this class. Based on this insight, we develop a feature alignment (FA) technique to encourage the last-layer feature of a sample to approximate the mean feature of target-class samples. In other words, FA promotes neural collapse, but to incorrect centers. Compared to cross-entropy attack loss to lead to only misprediction, FA induces a sample to have exactly the high-dimensional representative feature of target-class samples, imposing stronger restrictions on samples. Mathematically, FA in SEP can be expressed as

$$\boldsymbol{G}_{\text{SEP-FA}}(h_{\text{p}}, \boldsymbol{x}) = \sum_{k=0}^{n-1} \boldsymbol{G}_{\text{FA}}(h_{\text{p}}^k, \boldsymbol{x}) = \sum_{k=0}^{n-1} \nabla_{\boldsymbol{x}} \| h_{\text{p}}^k(\boldsymbol{x}) - h_{\text{c}}^k(g(y)) \|, \quad h_{\text{c}}^k(y) = \frac{\sum_{\boldsymbol{x} \in \mathcal{T}_y} h_{\text{p}}^k(\boldsymbol{x})}{|\mathcal{T}_y|}, \tag{4}$$

where h_{p}^k stands for the feature extractor (except for the last linear layer) of f_{p}^k, and $h_{\text{c}}^k(g(y))$ is for the mean (center) feature in the target class $g(y)$ calculated by h_{p}^k. $\| \cdot \|$ means the MSE loss.

Our overall algorithm is summarized in Alg. 1, where we use a stochastic variance reduction (VR) gradient method (Johnson & Zhang, 2013) to avoid bad local minima in optimization. Our method first calculates the feature alignment gradients $\boldsymbol{g}_k$ for protective perturbations by each training checkpoint (line 4). Then before updating the sample by the accumulated gradients (line 11), we reduce the variance of ensemble gradients (from $\boldsymbol{g}_{\text{ens}}$ to $\boldsymbol{g}_{\text{upd}}$) in a predictive way by an inner optimization with M virtual updates on $\hat{\boldsymbol{x}}_m$. VR has been verified to improve the transferability of ensemble adversarial attacks (Xiong et al., 2022), and could also boost the performance in data protection.

Algorithm 1 Self-Ensemble Protection with Feature Alignment and Variance Reduction

Input: Dataset $\mathcal{T} = \{(\boldsymbol{x}, y)\}$, ℓ_∞ bound ε, step size α, number of protection iterations T, number of training iterations N, number of checkpoints in self-ensemble n, number of inner updates M

Output: Protected dataset $\boldsymbol{x}'$

Train a DNN for N epochs and save n equidistant checkpoints
$\boldsymbol{x}_0 = \boldsymbol{x}$
for $t = 0 \rightarrow T-1$ **do**
  for $k = 0 \rightarrow n-1$ **do** $\boldsymbol{g}_k = \boldsymbol{G}_{\text{FA}}(h_{\text{p}}^k, \boldsymbol{x}_t)$ *# get the gradients by each checkpoints as (4)*
  $\boldsymbol{g}_{\text{ens}} = \frac{1}{n}\sum_{k=0}^{n-1} \boldsymbol{g}_k,\ \boldsymbol{g}_{\text{upd}} = 0,\ \hat{\boldsymbol{x}}_0 = \boldsymbol{x}_t$ *# initialize variables for inner optimization*
  for $m = 0 \rightarrow M-1$ **do**
    Pick a random index k *# stochastic variance reduction (Johnson & Zhang, 2013)*
    $\boldsymbol{g}_{\text{upd}} = \boldsymbol{g}_{\text{upd}} + \boldsymbol{G}_{\text{FA}}(h_{\text{p}}^k, \hat{\boldsymbol{x}}_m) - (\boldsymbol{g}_k - \boldsymbol{g}_{\text{ens}})$ *# accumulate gradients with variance*
    $\hat{\boldsymbol{x}}_{m+1} = \Pi_\epsilon\left(\hat{\boldsymbol{x}}_m - \alpha \cdot \text{sign}(\boldsymbol{g}_{\text{upd}})\right)$ *# virtual update on* $\hat{\boldsymbol{x}}_m$
  end for
  $\boldsymbol{x}_{t+1} = \Pi_\epsilon\left(\boldsymbol{x}_t - \alpha \cdot \text{sign}(\boldsymbol{g}_{\text{upd}})\right)$ *# update samples with variance-reduced gradients*
end for
return $\boldsymbol{x}' = \boldsymbol{x}_{T-1}$

With efficient self-ensemble, our method only requires $1 \times N$ training epochs, and $T \times (n + M)$ times of backward calculation to update samples. However, our method could be amazingly effective in reducing the appropriator model's accuracy with a novel feature alignment technique and variance reduction optimization, which we would demonstrate in detail in the next section.

4 EXPERIMENTS

4.1 SETUP

We evaluate SEP along with 7 data protection baselines, including adding random noise, TensorClog aiming to cause gradient vanishing (Shen et al., 2019), Gradient Alignment to target-class gradients (Fowl et al., 2021a), DeepConfuse that protects by an autoencoder (Feng et al., 2019), Unlearnable Examples (ULEs) using error-minimization noise (Huang et al., 2020a), Robust ULEs that use adversarial training (Fu et al., 2021), and Adversarial Poison (AdvPoison) resorting to targeted attacks (Fowl et al., 2021b). We use the reproduced results in (Fowl et al., 2021b) in Table 1 & 3.

For our method, we optimize class-y samples to have the mean feature of class $g(y)$, where $g(y) = (y+5)\%10$ for CIFAR-10 (Krizhevsky et al., 2009) and ImageNet (Krizhevsky et al., 2017) protected classes, and $g(y) = (y+50)\%100$ for CIFAR-100. For the ImageNet subset, we train f_{a} and f_{p} with the first 100 classes in ImageNet-1K, but only protect samples in 10 significantly different classes, which are African chameleon, black grouse, electric ray, hammerhead, hen, house finch, king snake, ostrich, tailed frog, and wolf spider. This establishes the class-wise data protection setting, and the reported accuracy is calculated on the testing samples in these 10 classes. We train a ResNet18 for $N = 120$ epochs as f_{p} following (Huang et al., 2020a; Fowl et al., 2021b). 15 equidistant intermediate checkpoints (epoch 8, 16, ..., 120) are adopted with $M = 15, T = 30$ if not otherwise stated. Experiments are conducted on an NVIDIA Tesla A100 GPU but could be run on GPUs with 4GB+ memory, because we store checkpoints on hardware instead of memory.

The data protection methods are assessed by training models with 5 architectures, which are ResNet18 (He et al., 2016), SENet18 (Hu et al., 2018), VGG16 (Simonyan & Zisserman, 2015), DenseNet121 (Huang et al., 2017), and GoogLeNet (Szegedy et al., 2015). They are implemented from Pytorch vision (Paszke et al., 2019). We train appropriator DNNs f_{a} for 120 epochs by an SGD optimizer with an initial learning rate of 0.1, which is divided by 10 in epochs 75 and 90. The momentum item in training is 0.9 and the weight decay is 5e-4. In this setting, DNNs trained on clean data could get great accuracy, *i.e.*, 95% / 75% / 78% for CIFAR-10 / CIFAR-100 / ImageNet subset. Training configurations are the same for f_{a} and f_{p}. In the ablation study, we denote the pure self-ensemble (3) as SEP, SEP with feature alignment (4) as SEP-FA, and SEP-FA with variance reduction as SEP-FA-VR (Alg. 1). In other experiments, "ours" stands for our final method, *i.e.*, SEP-FA-VR. We put the confusion matrix of f_{a} in Appendix C to provide class-wise analysis.

4.2 PROTECTIVE PERFORMANCE

We demonstrate the amazing protective performance of our method on 3 datasets, 5 architectures, 7 training strategies, and 7 data protection methods. In table 1, our method surpasses existing state-of-the-art baselines by a large margin, reducing the testing accuracy of CIFAR-10 model to 4.73%, which is far below the random guess accuracy of 10%. This indicates that we successfully inject incorrect-class features that are more sensitive than semantic ones in view of DNNs.

Table 1: ResNet18 testing accuracy trained on CIFAR-10 secured by different methods.

Method ($\epsilon = 8/255$)	Accuracy (↓)
None	94.56
Random Noise	90.52
TensorClog (Shen et al., 2019)	84.24
Gradient Alignment (Fowl et al., 2021a)	53.67
DeepConfuse (Feng et al., 2019)	31.10
Unlearnable Examples (Huang et al., 2020a)	19.85
Robust Unlearnable Examples (Fu et al., 2021)	13.05
Adversarial Poison (Fowl et al., 2021b)	6.25
Ours	**4.73**

We further study the efficacy of our method on different datasets, architectures, and bounds. In Table 2, we set an extremely small bound of $\varepsilon = 2/255$, even for high-resolution ImageNet samples. The perturbations would be invisible even under meticulous inspection, seeing Fig. 1 and Appendix A. However, the appropriator can only reach no more than 30% accuracy in most cases. When we increase the bound to the common choice $\varepsilon = 8/255$, no model can have over 6% accuracy, and the protected ImageNet samples even lead to an extreme 0.00% accuracy. In this case, the data is effectively protected from unauthorized use for training.

Table 2: Model testing accuracy trained on our secured datasets.

ϵ	Dataset	ResNet18	SENet18	VGG16	DenseNet121	GoogLeNet
$2/255$	CIFAR-10	14.68	15.93	23.66	15.02	17.99
	CIFAR-100	21.16	19.48	66.73	21.85	27.49
	ImageNet subset	8.80	10.70	27.00	9.40	30.80
$8/255$	CIFAR-10	4.73	2.47	5.61	3.76	3.44
	CIFAR-100	2.65	2.73	3.15	2.43	2.63
	ImageNet subset	0.00	0.40	0.60	0.20	0.00

Table 3: Effectiveness of data protection methods under difference training strategies.

Training Strategy ($\epsilon = 8/255$)	AdvPoison	Ours
None	6.25	**4.73** (-1.52)
Adv Train (Madry et al., 2018)	83.01	**82.51** (-0.50)
Gaussian Smoothing	11.94	**9.95** (-1.99)
Adding Random Noise	6.55	**2.89** (-3.66)
Mixup (Zhang et al., 2018)	15.86	**10.06** (-5.80)
Cutmix (Yun et al., 2019)	10.09	**4.47** (-5.62)
Cutout (DeVries & Taylor, 2017)	8.11	**5.51** (-2.60)
DPSGD (Abadi et al., 2016)	24.61	**4.60** (-20.01)

The above results are tested without special training strategies. Here we study the performance of data protection methods under adversarial training, image processing (Gaussian smoothing with kernel size 5, adding random noise with variance $8/255$), data augmentations (mixup, cutmix, cutout), and

privacy protection optimization (DPSGD with hyper-parameters in (Fowl et al., 2021b)). As in Table 3, most strategies cannot train a good DNN, and we could also decrease the accuracy of AdvPoison in this setting. The only useful strategy to recover accuracy is adversarial training (AT), which is consistent with the results in existing research (Huang et al., 2020a; Fowl et al., 2021b). The reason is that AT always searches for the worst-case perturbations, which may cancel out the protective perturbations. However, due to the several-fold extra cost and the low accuracy when trained on clean data (Zhang et al., 2019), AT is not practical to erase protective perturbations (Fowl et al., 2021b).

4.3 Ablation Study

We study the performance improvements from SEP, FA, and VR separately along with the best baseline AdvPoison (Fowl et al., 2021b), seeing Table 1. We first control the overall computation the same for all experiments. Then we vary the number of sub-models n to see its effect on our method.

In Table 4, we maintain our methods to have comparative computation with AdvPoison, which trains the protecting DNN for 40 epochs and craft perturbations by 250 steps for 8 times (we modify it to 4). In SEP and SEP-FA, we train for $N = 120$ epochs and use $n = 30$ checkpoints to update samples for $T = 30$ times, aligning the computation with AdvPoison. In SEP-FA-VR with inner update $M = 15$, we alter the number of checkpoints $n = 15$ so that the overall computation is the same, which is also the default setting for all experiments as in Sec. 4.1. In the conventional ensemble, 30 DNNs with 5 architectures mentioned in Sec. 4.1 are trained with 6 different random seeds.

Table 4: Ablation study on SEP (Self-Ensemble Protection), FA (Feature Alignment), VR (Variance Reduction). Lower accuracy stands for better protection performance.

Method ($\epsilon = 2/255$)	Train (Epoch)	Crafting (Step)	Accuracy of DNNs Trained on Secured CIFAR-10 (↓) ResNet18	SENet18	VGG16	DenseNet121	GoogLeNet
AdvPoison	40	1000	41.35	40.54	52.22	43.28	48.04
Ensemble	30×120	30×30	16.32	16.91	29.19	16.74	20.34
SEP	120	30×30	19.92	17.81	28.12	18.07	22.48
SEP-FA	120	30×30	16.91	17.01	26.82	16.88	21.15
SEP-FA-VR (ours)	120	30×(15+15)	**14.68**	**15.93**	**23.66**	**15.02**	**17.99**

As shown in Table 4, SEP is able to more than halve the accuracy of AdvPoison within the same computation budget, indicating that knowledge of multiple models is much more important than additional update steps. Results also illustrate that the superiority of SEP would be more significant when the perturbations are small, seeing Table 1. In comparison with the conventional ensemble, which requires 30× training computation, SEP performs just a little worse. Moreover, equipped with FA, which consumes no additional calculation, efficient SEP-FA could be as effective as the conventional ensemble. With VR, SEP-FA-VR is stably better, and reduces the accuracy from 45.10% to 17.47% on average.

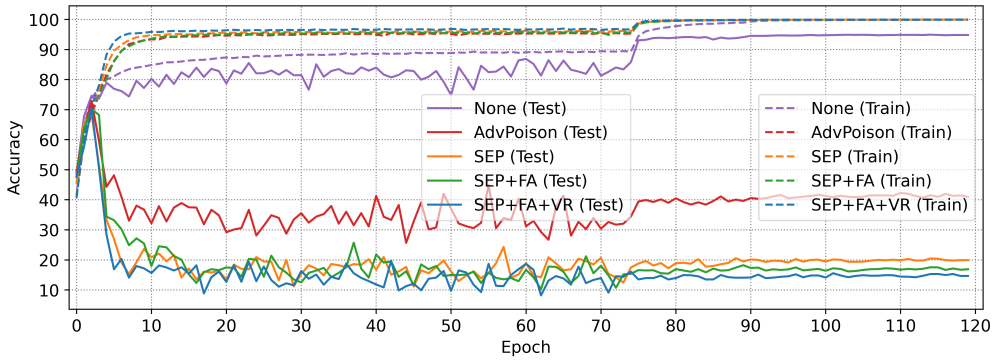


Figure 3: The accuracy trend of training a CIFAR-10 ResNet18 using different data protection methods ($\varepsilon = 2/255$, None means no protection). All methods have high training accuracy v.s. low testing accuracy, but SEP, equipped with FA and VR, reduces model's generalization most effectively.

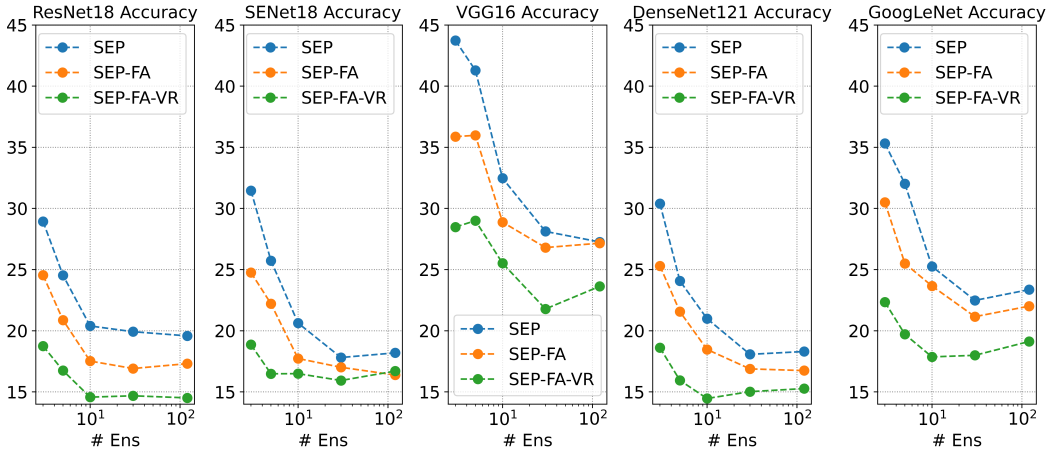


Figure 4: Ablation study on FA and VR with a different number of sub-models in self-ensemble. Results are produced on CIFAR-10 models with the bound of protective perturbations $\varepsilon = 2/255$.

We illustrate the training process of Table 4 experiments in Fig. 3. Compared to the training on clean data (purple line), data protection methods accelerate the model's convergence on training data, but the DNN's testing accuracy would suddenly drop at the initial stage of training. After the learning rate decay at epoch 75, the protection performance of different methods could be clearly observed. SEP accounts for the majority of performance improvements, and FA and VR could also further decrease the accuracy, making our method finally outperform the conventional inefficient ensemble.

We also vary the number of intermediate checkpoints n used in self-ensemble to perform ablation study under different computation budgets. We set $n = 3, 5, 10, 30, 120$ without changing other hyper-parameters in self-ensemble, and plot the results in Fig. 4. Similar results could be seen, *i.e.*, FA, and VR could stably contribute to the performance. We also discover that although n is increasing exponentially, the resulting performance increase would not be that significant as n is large. Most prominently, raising n from 30 to 120 is not bound to yield better results, meaning that the performance would saturate around $n = 30$, and it would be unnecessary to use all checkpoints.

5 Discussion and Conclusion

In this paper, we overcome the inefficiency of ensemble by proposing a self-ensemble method to reuse the intermediate training checkpoints in the data protection problem. We challenge the common belief on checkpoint similarity by a surprising discovery that the gradients of checkpoints during a single training are close to orthogonal. In this regard, checkpoints are sufficiently diverse to serve as good data protectors, and the results support this point: our $\varepsilon = 8/255$ perturbations could poison all CIFAR-10, CIFAR-100, ImageNet models to have an accuracy of less than 6%.

To further boost the performance of self-ensemble protection (SEP), we develop a novel feature alignment (FA) loss inspired by the neural collapse theory. FA forces a sample to have the exact representative high-dimensional mean feature of target-class samples, so that incorrect features are robustly injected into training samples. With a variance reduction optimization, our method exceeds current baselines by a large margin, reducing the appropriator model's accuracy from 45.10% (best-known results) to 17.47% on average.

Our work reveals the possibility of protecting data by extremely small perturbations, *e.g.*, $\varepsilon = 2/255$, without a huge computation budget. The key is to aggregate the knowledge of multiple models, even if they come from a single training process. Our method could also serve as a potential benchmark to evaluate the DNN's learning process, *e.g.*, how to prevent DNNs from learning non-robust features (shortcuts) instead of semantic ones. The code to reproduce our results has been attached in the Supplementary Material.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pp. 308–318, 2016.

Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines. In *International Conference on Machine Learning (ICML)*, pp. 1467–1474, 2012.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (S&P)*, pp. 39–57. IEEE, 2017.

Sizhe Chen, Zhengbao He, Chengjin Sun, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset DAmageNet. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 2188–2197, 2022.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9185–9193, 2018.

Minghong Fang, Neil Zhenqiang Gong, and Jia Liu. Influence function based data poisoning attacks to top-n recommender systems. In *Proceedings of The Web Conference (WWW)*, pp. 3019–3025, 2020.

Ji Feng, Qi-Zhi Cai, and Zhi-Hua Zhou. Learning to confuse: generating training time adversarial data with auto-encoder. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 11971–11981, 2019.

Liam Fowl, Ping-yeh Chiang, Micah Goldblum, Jonas Geiping, Arpit Bansal, Wojtek Czaja, and Tom Goldstein. Preventing unauthorized use of proprietary data: Poisoning for secure dataset release. *arXiv preprint arXiv:2103.02683*, 2021a.

Liam H Fowl, Micah Goldblum, Ping-yeh Chiang, Jonas Geiping, Wojciech Czaja, and Tom Goldstein. Adversarial examples make strong poisons. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 30339–30351, 2021b.

Shaopeng Fu, Fengxiang He, Yang Liu, Li Shen, and Dacheng Tao. Robust unlearnable examples: Protecting data privacy against adversarial learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Jonas Geiping, Liam H Fowl, W Ronny Huang, Wojciech Czaja, Gavin Taylor, Michael Moeller, and Tom Goldstein. Witches' brew: Industrial scale data poisoning via gradient matching. In *International Conference on Learning Representations (ICLR)*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7132–7141, 2018.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017.

Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations (ICLR)*, 2020a.

W Ronny Huang, Jonas Geiping, Liam Fowl, Gavin Taylor, and Tom Goldstein. Metapoison: Practical general-purpose clean-label data poisoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 12080–12091, 2020b.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. pp. 125–136, 2019.

Nathan Inkawhich, Kevin Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *International Conference on Learning Representations (ICLR)*, 2019a.

Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7066–7074, 2019b.

Nathan Inkawhich, Kevin Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20791–20801, 2020.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 315–323, 2013.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pp. 1885–1894, 2017.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM (CACM)*, pp. 84–90, 2017.

Tao Li, Lei Tan, Zhehao Huang, Qinghua Tao, Yipeng Liu, and Xiaolin Huang. Low dimensional trajectory hypothesis is true: Dnns can be trained in tiny subspaces. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, pp. 1–1, 2022.

Jiabin Liu, Zhiquan Qi, Bo Wang, YingJie Tian, and Yong Shi. Self-llp: Self-supervised learning from label proportions with self-ensemble. *Pattern Recognition*, pp. 108767, 2022.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations (ICLR)*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision (ECCV)*, pp. 181–196, 2018.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning (ICML)*, pp. 4970–4979, 2019.

Vardan Papyan, XY Han, and David L Donoho. Prevalence of neural collapse during the terminal phase of deep learning training. In *Proceedings of the National Academy of Sciences (PNAS)*, pp. 24652–24663, 2020.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 8024–8035, 2019.

Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. In *International Conference on Learning Representations (ICLR)*, 2016.

Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. In *Advances in neural information processing systems (NeurIPS)*, pp. 6106–6116, 2018.

Juncheng Shen, Xiaolei Zhu, and De Ma. Tensorclog: An imperceptible poisoning attack on deep neural network applications. In *IEEE Access*, pp. 41498–41506, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning (ICML)*, pp. 9155–9166, 2020.

Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 843–852, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.

Hongjun Wang and Yisen Wang. Self-ensemble adversarial training for improved robustness. In *International Conference on Learning Representations (ICLR)*, 2021.

Yifeng Xiong, Jiadong Lin, Min Zhang, John E Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14983–14992, 2022.

Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks. *arXiv preprint arXiv:1703.01340*, 2017.

Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: diversifying vulnerabilities for enhanced robust generation of ensembles. *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 5505–5515, 2020.

Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin Rubinstein, Ce Zhang, and Bo Li. TRS: Transferability reduced ensemble via promoting gradient diversity and model smoothness. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 17642–17655, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6023–6032, 2019.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Painless adversarial training using maximal principle. In *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 227–238, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018.

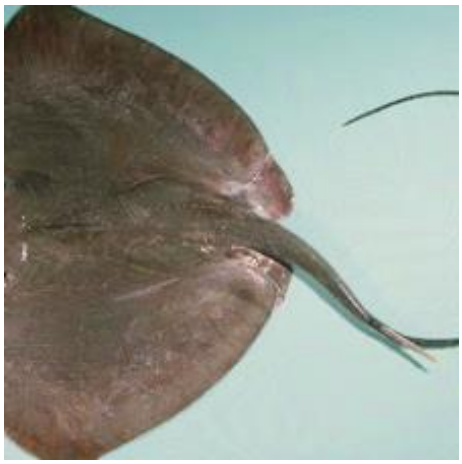
Yuan-Xing Zhao, Yan-Ming Zhang, Ming Song, and Cheng-Lin Liu. Multi-view semi-supervised 3d whole brain segmentation with a self-ensemble network. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pp. 256–265, 2019.

A PROTECTED IMAGENET SAMPLES

African chameleon

black grouse

electric ray



hammerhead

hen

house finch

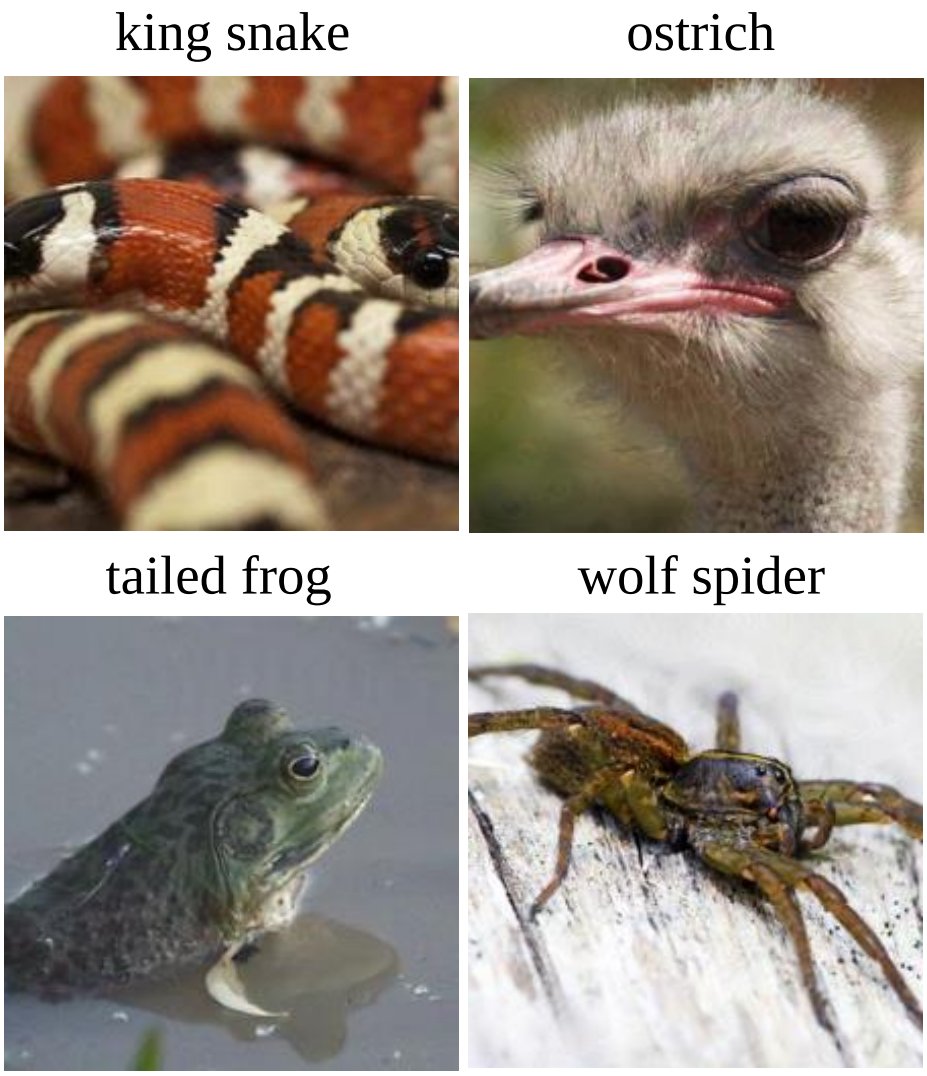


Figure 5: 10-class protected ImageNet samples by our method with $\varepsilon = 2/255$. We select the images with clean backgrounds, but the perturbations are still invisible in such cases. However, it would significantly decrease the model's accuracy to below 30%.

B GRADIENT ANALYSIS OF CIFAR-100 AND IMAGENET CHECKPOINTS

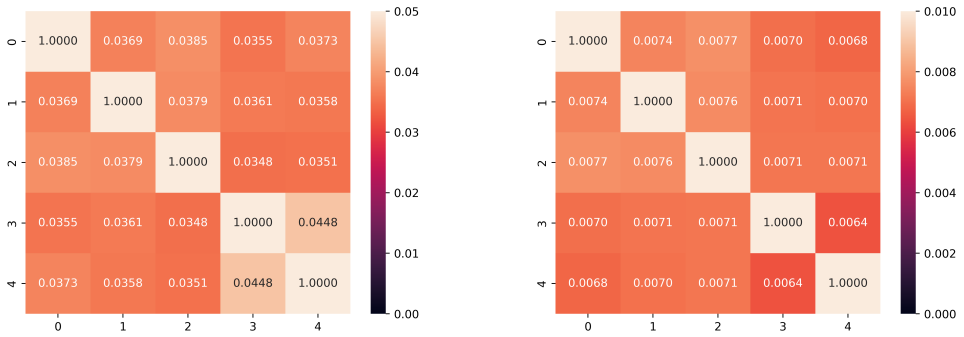


Figure 6: The average absolute cosine value of cross-model gradients for CIFAR-100 (left) and ImageNet (right) checkpoints calculated by 1K samples. The gradients stay nearly orthogonal, indicating good diversity of sub-models in self-ensemble.

C CONFUSION MATRIX OF THE APPROPRIATOR DNN

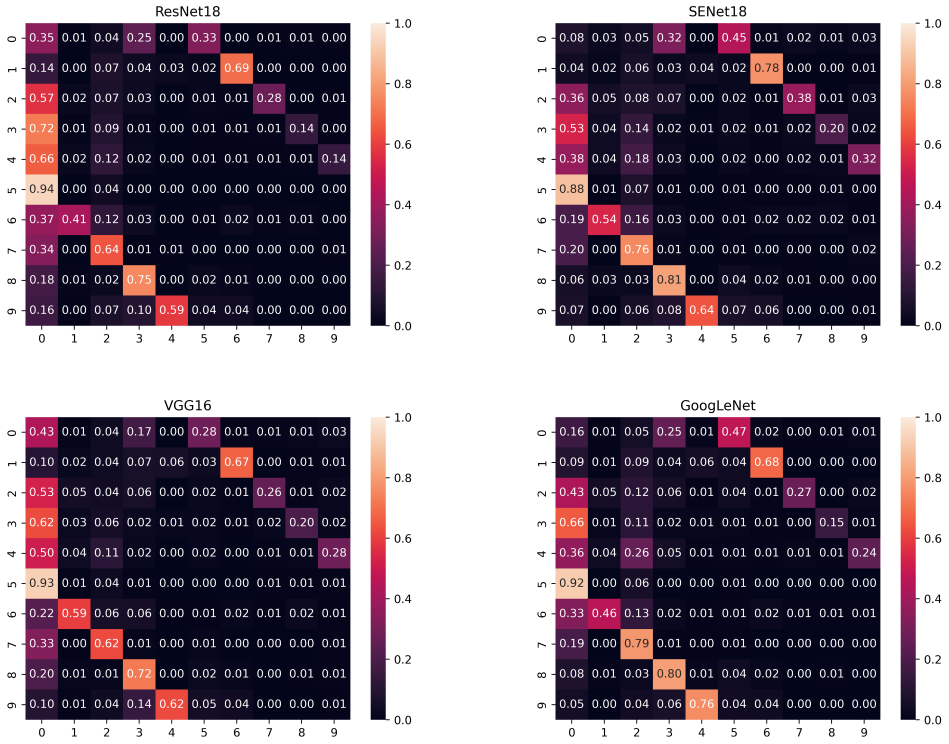


Figure 7: The confusion matrices of DNNs trained on $\varepsilon = 8/255$ CIFAR-10 perturbations by our method. As also shown in (Fowl et al., 2021b), the model tends to predict clean samples to the target class $g(y) = (y + 5)\%10$. We also discover that DNNs prefer to classify them to a fixed class, which is class 0 here, but it may vary during the training. We normalize by the number of samples here.