Training Deep Neural Networks with Virtual Smoothing Classes

Anonymous submission

Abstract

Learning with softmax cross-entropy on one-hot labels often leads to overconfidence on the correct class. While label smoothing regulates this overconfidence by redistributing α confidence from the correct class to other incorrect classes, it compromises the representation in the logits about the similarity between samples of different classes and may hurt calibration if a larger α is required for high accuracy. To overcome these limitations, we propose a Virtual Smoothing label that redistributes certain confidence from the correct class to additional Virtual Smoothing (VS) classes to regularize overconfidence. In VS labels, the VS class nodes act as adversaries to the original class nodes, enforcing regularization by clustering samples across all classes. The zero confidence of each incorrect class also allows the incorrect logits to be different from each other without erasing information about sample similarities. The prediction probability can still approach 1 when applying softmax to the logits of the original real classes, which avoids harming but consistently improves calibration. Experiments show that VS labels consistently improve accuracy and calibration while providing better logits for improved knowledge distillation. Additionally, VS labels exhibit effectiveness in improving adversarial training, robust distillation, and out-of-distribution detection.

1 Introduction

Deep Neural Networks (DNNs) have shown impressive performance in various tasks. Training DNN classifiers with one-hot labels is a widely adopted norm in the classification task. One-hot labels enable DNNs to extract class-specific information (Yang et al. 2021) and learn knowledge in the logits about the similarity between samples of different classes (Müller, Kornblith, and Hinton 2019), from input samples. The well-known knowledge distillation (Hinton et al. 2015) further uses soft labels from teacher models to improve student model performance. However, training with one-hot labels easily cause overconfidence on the correct class, as minimizing the cross-entropy loss encourages the correct class node to extract any information unavailable in the samples of other classes as its predictive information.

It is widely recognized that DNNs are susceptible to yielding confidently error predictions for adversarial samples with imperceptible perturbations (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014). Among various defenses, adversarial training (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017), which treats adversarial samples as a form of data regularization, effectively enhances the robustness of DNNs (Athalye, Carlini, and Wagner 2018). Its successful variants (Zhang et al. 2019; Carmon et al. 2019; Goldblum et al. 2020) introduce more regularization, e.g., weight perturbation (Wu, Xia, and Wang 2020), to further improve robustness. On the other hand, DNNs also tend to be overconfident on unseen Out-Of-Distribution (OOD) samples (Hendrycks and Gimpel 2016), and semi-supervised training using diverse auxiliary outliers (Hendrycks, Mazeika, and Dietterich 2018; Mohseni et al. 2020) significantly improves and achieves state-of-the-art (SOTA) performance in OOD detection. These methods all improve DNNs by introducing increased regularization. However, they rely on one-hot labels and fail to consider the regularization from a label perspective.

As a label considering regularization, Label Smoothing (LS) (Szegedy et al. 2016) effectively improve DNN performance by redistributing some confidence from the correct class to other incorrect classes:

$$y^{LS} = (1-\alpha)\cdot y + \alpha/K \tag{1}$$

where K is the number of sample classes, and $\alpha \in [0,1]$ controls the smoothness. However, due to the non-zero supervisory signals applied on incorrect classes, LS hurts the representation in the logits about sample similarities, which further impairs the downstream knowledge distillation (Müller, Kornblith, and Hinton 2019). Moreover, as shown in our later experiments (Sec 4.1), LS may lead to a trade-off between accuracy and calibration (Guo et al. 2017) when a model requires a larger α to achieve higher accuracy. With the model's accuracy increases (up to 1), the prediction confidence of the correct class decreases (up to 1-α), which is detrimental to calibration.

In this work, we propose a Virtual Smoothing (VS) label to help regularize overconfidence but eliminate the limitations in LS. Similar to the philosophy of GANs (Goodfellow et al. 2020) and adversarial training, VS labels use additional virtual classes (named VS classes) as adversaries to the original real classes to uniformly regulate all input samples, as shown in Fig 1. By assigning the VS classes the same confidence across samples from different classes, these VS classes *compete with* each correct class to collect information that is unspecific to any correct class, thereby

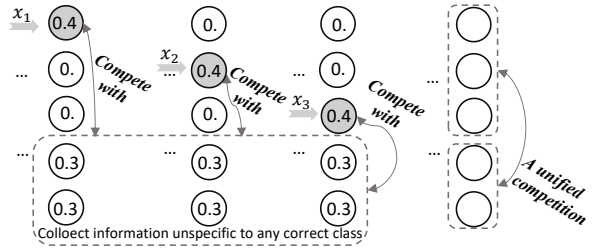


Figure 1: An intuitive example of VS labels. While the first three real classes classify input samples (by extracting class-specific information) for classification, the last two VS classes act as their adversaries to cluster input samples (by collecting class-unspecific information) for regularization.

regularizing overconfidence. Meanwhile, the confidence of each incorrect class in VS labels remains zero, as in one-hot labels, allowing the logits of incorrect classes to differ without erasing sample similarity information. In the inference stage, by applying softmax only to the logits of the original real classes, the upper bound of the prediction probability on the correct class is still 1, aligning with that of the accuracy.

Experiments show that VS labels enable the model to achieve better accuracy and calibration while providing better logits for downstream knowledge distillation, which are not achievable with one-hot and LS labels. In adversarial settings, VS labels also outperform one-hot and LS labels in terms of robustness and downstream robust knowledge distillation. Additionally, VS labels outperform one-hot and LS labels in out-of-distribution detection. Our contributions are:

- We propose a Virtual Smoothing (VS) label, designed to regularize overconfidence in the correct class by clustering all input samples using additional VS classes.
- We present a mathematical analysis of VS labels and demonstrate that VS classes avoid erasing information in the logits regarding sample similarities and introducing a trade-off between accuracy and calibration while regularizing overconfidence (on the correct classes).
- Through extensive experiments, we demonstrate that VS labels simultaneously enhance accuracy, calibration, and downstream knowledge distillation. In additional scenarios, VS labels further improve adversarial training and downstream robust distillation, as well as out-of-distribution detection.

2 Related Work

We discuss standard training and adversarial training but include adversarial attacks, robust evaluation and out-of-distribution detection in Sec A in the Appendix.

2.1 Standard Training

Training DNN classifiers with cross-entropy (CE) loss on one-hot labelled data has been the norm for the classification task for many years (He et al. 2019; Chen et al. 2018). One-hot labels enable DNN classifiers to extract class-specific information from input samples. Let $y_i \in \{0,1\}^K$ be the one-hot label for the i-th sample and $f_\theta(x_i) \in [0,1]^K$ (abbreviated as f_i) be the predicted probabilities of the DNN classifier f_θ on the input x_i. Then, the CE loss $\ell(f,y)$ is: $-\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{K} y_{ij}\log(f_{ij}) = -\frac{1}{N}\sum_{i=0}^{N}\log(f_{ik})$, where f_{ik} represents the predicted probability of the correct class k of x_i. One-hot labels also enable DNN classifiers to learn the dark knowledge about input samples, e.g., information in the logits about the similarities between samples of different classes. The well-known model compression method, knowledge distillation (Hinton et al. 2015), exploits the dark knowledge in the logits of large models to teach small models to achieve better performance.

2.2 Adversarial Training

DNNs are vulnerable to adversarial attacks (Szegedy et al. 2013; Goodfellow, Shlens, and Szegedy 2014). Across various defenses (Papernot et al. 2016; Bai et al. 2019; Ma et al. 2018; Tramèr et al. 2017; Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017), Adversarial Training (AT) (Goodfellow, Shlens, and Szegedy 2014; Madry et al. 2017) is one of the most effective methods to defending against adversarial attacks (Athalye, Carlini, and Wagner 2018). It treats adversarial samples as a data augmentation:

$$\arg\min_{\theta} \frac{1}{N}\sum_{i=1}^{N} \max_{||\delta||_p \leq \epsilon} \ell(f_\theta(x_i+\delta_i), y_i) \tag{2}$$

where δ_i is the adversarial perturbation bounded by an L_p norm within the ϵ-ball of the clean sample x_i and $\ell(\cdot)$ represents the adversarial loss, e.g., the CE loss. In the inner maximization, a K-step Projected Gradient Descent (PGD-K) Attack (Goodfellow, Shlens, and Szegedy 2014; Kurakin, Goodfellow, and Bengio 2017) is used to approximately search for the optimal perturbation. AT inspires a group of variants to further improve robustness, e.g., TRADES (Zhang et al. 2019), semi-supervised RST (Carmon et al. 2019), Adversarial Weight Perturbation (AWP) (Wu, Xia, and Wang 2020) and Adversarial Robust Distillation (ARD) (Goldblum et al. 2020). We put more details of them in Sec A.3 in the Appendix.

3 Proposed Approach

We present the construction method and analysis of VS labels in Sec 3.1 and Sec 3.2, respectively.

3.1 Virtual Smoothing (VS) Label

Suppose we add V VS class nodes to the last layer of a K-way classifier, the classifier becomes a new (K+V)-way classifier f_θ whose first K-way, $f_{\theta[1:K]}$, corresponds to the original classifier. Given a K-dimension one-hot label y, its (K+V)-dimension VS label $\widetilde{y}$ is:

$$(\widetilde{y})_j : \begin{cases} (1\text{-}\alpha)\cdot y, \ where\ 1 \leq j \leq K \\ \alpha/V, \ where\ K{+}1 \leq j \leq K{+}V \end{cases} \tag{3}$$

where $(\widetilde{y})_j$ is the j-th element of $\widetilde{y}$, $\alpha \in [0,1]$ is the total confidence of all VS classes. Taking all VS classes as a super-virtual class for easy understanding. While the original real classes classify input samples for prediction, the super-virtual class with a confidence of α cluster all input

samples for regularization. In the inference phase, we consider the index of the maximum logit within the first K real classes as the predicted result. Applying softmax to the first K logits ensures that the upper bound of the prediction probability on the correct class remains 1, independent of α, avoiding the issue in LS where the upper bounds of prediction confidence (1-α) and accuracy (1) are inconsistent.

Assigning the VS classes a uniform distribution (with a sum of α) encourages them to collect information unspecific to any correct class as unnecessary predictive information for regularization. A larger α encourages the VS classes (to compete with each correct classes) to collect more information as class-unspecific information to regularize overconfidence, as analyzed in the next Sec 3.2. Note that the VS class differs from the *reject* class (Vernekar et al. 2019; Mohseni et al. 2020) in that it is designed for regularization and its confidence assigned is independent of the ground-truth of the input sample. However, when the total confidence of the VS classes is set to 1, VS labels represent information unspecific to any sample in the training set.

3.2 Analysis

The Cross-Entropy (CE) loss on the VS label is:

$$-[(1-\alpha)\log(f_{\hat{y}}) + \sum_{i=1,i\neq\hat{y}}^{K} 0\cdot\log(f_i) + \sum_{j=K+1}^{K+V}\frac{\alpha}{V}\log(f_j)] \tag{4}$$

where $\hat{y}$ is the correct class, the second term is 0 can be eliminated, and the third is the regularization term (called VS regularization term) only related to VS classes.

Following (Müller, Kornblith, and Hinton 2019), we write the prediction of a DNN as a function $f_k = \frac{e^{p^T w_k}}{\sum_{l=1}^{L} e^{p^T w_l}}$, where f_k denotes the prediction probability of the k-th class, p denotes the activation of the penultimate layer, and w_k denotes the weight parameter of last layer corresponding to the k-th class. The logit $p^T w_k$ of the k-th class can be viewed as a measure of the Euclidean distance between the penultimate layer activation p and the template w_k. This is because $||p-w_k||^2 = p^Tp - 2p^Tw_k + w_k^Tw_k$, where p^Tp is factored out after calculating softmax outputs and w^Tw remains constant across classes. Therefore, we can rewrite f_k to:

$$f_k \Rightarrow \frac{e^{-||p-w_k||^2}}{\sum_{l=1}^{L} e^{-||p-w_l||^2}} \tag{5}$$

Replacing $f_{\hat{y}}$ and f_j in Eq (4) by the rightmost side of Eq (5) respectively, we find that minimizing Eq (4) is equivalent to minimizing (1-α)$||p-w_{\hat{y}}||^2$ and $\sum_{j=K+1}^{K+V}\frac{\alpha}{V}||p-w_j||^2$. This encourages the penultimate activation p to be close to the template $w_{\hat{y}}$ of the correct class (with weight 1-α) and equally distant to the template w_j of each VS class (with weight α/V). We call this as the competition between the correct class and the VS classes, as depicted in Fig 1.

During training, each correct class template competes with the same VS class templates $\{w_{K+1}, ..., w_{K+V}\}$, forcing the penultimate layer activation p to always be equally distant to each VS class template with the weight of α/V (regardless of the ground-truth class of the input sample), collecting class-unspecific information $\{p^Tw_{K+1}, ..., p^Tw_{K+V}\}$ to regularize overconfidence on all correct classes. Obviously, a larger α encourages the penultimate layer activation p to move more toward the VS templates, and a larger V encourages p to be equidistant from more VS templates. This increases the difficulty of fitting and is beneficial for achieving better generalization if the DNN is powerful enough. Moreover, the confidence of all incorrect classes remains 0, which does not force the penultimate layer activation to be close to any incorrect class template (due to $0\cdot\log(f_i) = 0$), saving information in the logits about similarities between samples of different classes.

Conversely, minimizing the CE loss on the LS label, i.e., $-[(1-\alpha)\log(f_{\hat{y}}) + \sum_{i=1}^{K}\frac{\alpha}{K}\log(f_i)]$, encourages the penultimate layer activation p to be close to the template $w_{\hat{y}}$ of the correct class and equidistant to the template w_i of each incorrect class. This erases the similarity information between samples of different classes, which is one of the most significant difference between LS and VS. Tab 1 summarizes the comparison between VS and LS regularization, where VS simultaneously improve model accuracy and calibration while avoiding damage to the representation in the logits about the similarity between samples of different classes. In contrast, LS not only harms the similarity information but may also impair calibration (especially when a larger α is required to achieve better performance).

Table 1: Comparison between LS regularization and VS regularization. $\surd$, $\neq$ and $\times$ represent 'favorable to', 'possibly unfavorable to' and 'unfavorable to', respectively. 'Similarity' refers to the representation in the logits regarding the similarity between samples from different classes.

labels	Accuracy	Similarity	Calibration
LS	$\surd$	$\times$	$\neq$
VS	$\surd$	$\surd$	$\surd$

4 Experiments

We evaluate the impact of VS labels on model accuracy, calibration, and knowledge distillation (Sec 4.1), followed by assessing their effects on robustness and robust distillation in adversarial settings (Sec 4.2), as well as their effects on out-of-distribution detection (Sec A.4).

4.1 Standard Training

Experimental Settings On SVHN (Netzer et al. 2011), CIFAR10, and CIFAR100 (Krizhevsky, Hinton et al. 2009), we train ResNet-18 (He et al. 2016) and ResNeXt-29 (2x64d) (Xie et al. 2017) for 200 epochs using SGD optimizer with momentum 0.9, weight decay 0.0001, batch size 128 and an initial Learning Rate (LR) 0.1 divided by 10 at the 100-th and 150-th epochs. Standard data augmentation includes random crop and random horizontal flip are adopted. On Tiny-ImageNet-200[1], we select ResNet-18 and ResNeXt-50 (32x4d) and use the same settings for

[1]https://tiny-imagenet.herokuapp.com/

training. On ImageNet (Russakovsky et al. 2015), we train ResNet-18 and ResNeXt-50 (32x4d) for 120 epochs with similar settings but set batch size to 256 and divide the LR by 10 at the 60-th, 90-th and 110-th epochs. Besides, we train Transformer architectures T2T-ViT-14 and T2T-ViT-24 (Yuan et al. 2021), which can be trained from scratch more easily than the original ViT (Dosovitskiy et al. 2020), on ImageNet. We use the AdamW scheduler with an initial LR of 0.001, batch sizes 256 (512), weight decay 0.05 (0.065) for T2T-ViT-14 (T2T-ViT-24). Additional data augmentation including MixUp and RandAugment are applied. Note that ViTs typically require more training epochs and larger batch sizes to achieve better performance. However, we use the above settings to save computational cost due to the numerous experiments required for different αs.

On all datasets, we set the number of VS classes to the number of the original real classes (V=K). The confidence α will be detailed in the subsequent subsections.

Table 2: Test accuracy (%). ResNeXt refers to ResNeXt-29 for SVHN and CIFAR but ResNeXt-50 for Tiny-ImageNet-200 (Tiny-200), and 'imp.' is the obtained improvement over baseline one-hot.

Model	Label	SVHN	CIFAR10	CIFAR100	Tiny-200
ResNet-18	one-hot	95.58	94.56	75.43	64.10
	LS [α]	95.74 [*0.3*]	94.99 [*0.5*]	77.56 [*0.4*]	65.00 [*0.5*]
	imp.	+ 0.16	+ 0.43	+ 2.13	+ 0.9
	VS [α]	**96.02** [*0.6*]	**95.30** [*0.5*]	**78.05** [*0.8*]	**65.86** [*0.8*]
	imp.	**+ 0.67**	**+ 0.74**	**+ 2.62**	**+ 1.76**
ResNeXt	one-hot	96.12	93.86	76.54	64.43
	LS [α]	96.51 [*0.1*]	94.88 [*0.4*]	78.18 [*0.9*]	65.77 [*0.9*]
	imp.	+ 0.39	+ 1.02	+ 1.64	+ 1.34
	VS [α]	**96.79** [*0.7*]	**95.27** [*0.8*]	**79.84** [*0.9*]	**66.19** [*0.8*]
	imp.	**+ 0.67**	**+ 1.41**	**+ 3.3**	**+ 1.76**

Table 3: Test accuracy (%) on Image-Net.

Model	one-hot	LS [α]	imp.	VS [α]	imp.
ResNet-18	70.63	70.69 [*0.05*]	+ 0.06	**70.74** [*0.05*]	+ **0.11**
ResNeXt-50	77.57	78.31 [*0.3*]	+ 0.74	**78.39** [*0.3*]	+ **0.82**
T2T-ViT-14	78.67	79.01 [*0.3*]	+ 0.35	**79.08** [*0.4*]	+ **0.42**
T2T-ViT-24	78.68	79.45 [*0.7*]	+ 0.77	**79.70** [*0.8*]	+ **1.02**

Accuracy We search for the optimal α within [0, 1] for all models, using a step size of 0.1, except for ResNet-18 ImageNet, which uses a step size of 0.05. Tab 2 shows the test accuracy in standard training on SVHN, CIFAR10, CIFAR100, and Tiny-ImageNet-200 (Tiny-200), with the optimal α also reported. The accuracy improvement ('imp.') achieved by VS labels over baseline one-hot labels is more pronounced than that of LS labels, especially on CIFAR100 and Tiny-200. For example, while LS labels achieve a 1.64% improvement on ResNeXt-29 for CIFAR100, VS labels achieve a 3.3% improvement. On ImageNet, VS labels again outperform LS labels across ResNet and ViT models, as shown in Tab 3. Furthermore, we observe that when fixing the dataset, a larger model usually require a larger α for higher accuracy. For instance, on CIFAR10, ResNeXt-29 needs a larger α compared to ResNet-18. Similarly, simpler datasets often necessitate a higher α for the same model. This supports our intuition that if a model is sufficiently powerful relative to the dataset, using a higher α to impose a larger penalty can enhance performance.

However, model accuracy is not our only concern. The effects on model calibration and learned representations in logits for downstream knowledge distillation are also significant. We will report these results after the visualization.

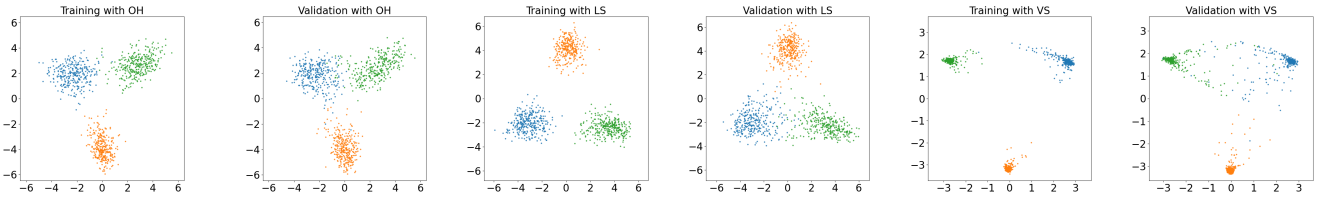
Visualization Following (Müller, Kornblith, and Hinton 2019), we visualize the penultimate layer activations. Specifically, we pick samples from three classes and find an orthogonal basis of the plane that crosses the templates of these three classes, and then project the activations of the penultimate layer onto this plane. For models trained on VS labels, we only consider its first K-way. The selected model is ResNet-18 on CIFAR10, where α is set to 0.5 for LS and VS labels for a fair comparison.

Fig 2 shows the visualization of the penultimate activations of training and validation samples, where the selected three classes are 'airplane' (blue), 'automobile' (orange) and 'bird' (green) respectively. Our observations are as follows: (1) The clusters corresponding to these three classes in LS and VS (columns 3-6) exhibit greater separability compared to the One-Hot (OH) (columns 1-2). (2) The clusters on LS labels are more closely arranged in the shape of an equilateral triangle, which diminishes the similarity information between different sample classes. In contrast, the clusters for the semantically similar classes 'airplane' and 'bird' in the OH and VS labels are positioned closer together, preserving the similarity information between classes. These visualizations illustrate that, while LS labels make the classification boundary more separable at the cost of erasing sample similarities, VS labels also make the classification boundary more separable but avoid erasing sample similarities. More visualization can be found in Sec F in the Appendix.

Model Calibration To better analyze the impact of different α values on calibration, we not only report the Expected Calibration Error (ECE) of the model in Sec 4.1 at its peak accuracy but also provide ECEs under varying α values. Tab 4 shows the results, where the ECE corresponding to the α yielding the highest accuracy enclosed in brackets '[]'. Instances where results deviate unfavorably from the baseline One-Hot (OH) label are highlighted in red. The selected models for CIFAR and ImageNet are ResNet-18 and ResNeXt-50, respectively. We observe that LS labels significantly impair calibration when a larger α is required for higher accuracy, since the predicted probability (1-α) over the correct class in LS-trained models becomes smaller than 1 as α increases. Conversely, regardless of the α value, VS labels consistently benefit calibration.

Note that normalizing LS model predictions to the range [0, 1] cannot rectify the adverse impact of LS on calibration, and incorporating temperature scaling (TS) enhance calibration but VS labels consistently achieve better calibration. We put these results in Sec B in the Appendix.

Knowledge Distillation We employ models trained on One-Hot (OH), LS and VS labels in Sec 4.1 as teachers to study their effect on Knowledge Distillation (KD). The interpolation parameter γ and temperature parameter τ are 1 and 30 respectively, following (Goldblum et al. 2020). Other training settings follow Sec 4.1.



(a) (b) (c) (d) (e) (f)

Figure 2: Visualization for the penultimate layer activation of ResNet-18 CIFAR10. The selected three classes are 'airplane' (blue), 'automobile' (orange) and 'bird' (green) respectively. We observe that (1) the clusters of these three classes on the LS and VS labels in columns 3-6 appear to be more separable, (2) the clusters of these three classes on the LS labels are organized more closely to a equilateral triangle (erase similarities between samples from different classes), whereas the clusters of 'airplane' (blue) and 'bird' (green) on the One-Hot (OH) and VS labels are closer to each other (save similarities between samples from different classes).

Table 4: ECE of models with different labels. ECEs for the α with highest accuracy are enclosed in brackets '[]', and ECEs worse than the baseline One-Hot (OH) label are marked red. We observe that VS consistently benefits calibration, whereas LS may impair calibration when it needs a larger α to achieve high accuracy.

	Labels	0.05	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CIFAR10	LS	0.0559	0.0884	0.1721	0.2529	[0.3466]	0.4207	0.5034	0.5937	0.6744	0.7557
(OH: 0.0306)	VS	0.0166	0.0172	0.0180	0.0167	0.0172	0.0209	0.0206	0.0166	[0.0154]	0.0098
CIFAR100	LS	0.1381	0.2006	0.2960	0.2432	0.4109	0.4671	0.5263	0.5977	0.6490	[0.6947]
(OH: 0.0942)	VS	0.0377	0.0245	0.0165	0.0168	0.0223	0.0238	0.0277	0.0146	0.0267	[0.0302]
ImageNet	LS	-	0.0601	0.0873	[0.1056]	0.1275	-	-	-	-	-
(OH: 0.1054)	VS	-	0.0485	0.04153	[0.0374]	0.0356	-	-	-	-	-

Table 5: Results of knowledge distillation. 'plain' denotes models trained on one-hot labels without teachers.

Dataset	Teacher → Student	plain	one-hot	LS	VS
SVHN	ResNet-18 → MobileNet-V2	95.94	96.27	96.57	**96.62**
CIFAR10	ResNet-18 → MobileNet-V2	94.54	94.87	94.61	**95.21**
CIFAR100	ResNet-18 → MobileNet-V2	75.39	76.47	77.15	**78.61**
Tiny-ImageNet-200	ResNeXt-50 → ResNet-18	64.10	65.32	54.17	**66.37**

Tab 5 shows the results of KD, where 'plain' denotes students trained on OH labels without teachers. Overall, VS teachers consistently enable students to achieve higher accuracy compared to OH and LS teachers. In contrast, while LS teachers perform better than OH teachers (as shown in Tab 2), they may lead to poorer student performance, as seen with MobileNet-V2 on CIFAR10 and Tiny-ImageNet-200. These results demonstrate that VS labels provide superior logits, enhancing downstream knowledge distillation while yielding higher accuracy and better calibration.

4.2 Adversarial Settings

In this section, we investigate the impact of VS labels on AT (and its variants) and robust distillation. We put the details of incorporating VS labels into ATs in Sec C in the Appendix.

Experimental Settings. Following the mainstream setting, we use the L_∞ threat model with a perturbation radius ϵ of 0.031 ($\approx$ 8/255). We train WRN-34-10 on CIFAR10 and CIFAR100 for AT, TRADES, and AWP, and WRN-28-10 on CIFAR10 with 500K unlabeled data for RST. AT and TRADES are trained for 160 epochs using SGD with momentum 0.9, weight decay 5e-4, batch size 128, and an initial LR 0.1 divided by 10 at the 150th and 155th epochs to prevent robust overfitting (Rice, Wong, and Kolter 2020). AWP follows its original paper's settings, with the LR reduced at the 150th and 180th epochs (200 epochs in total). RST is trained for 300 epochs with a batch size of 256, reducing the learning rate at the 150th and 225th epochs. The training attack is PGD-10 with a step-size of 0.00784 ($\approx$ 2/255). We increase α from 0 to 0.9 in 0.1 increments to find the optimal initial α, then fine-tune by 0.05 increments. For the evaluation, we use 20-step PGD, CW, and Auto-Attack. We also modify the number of targets in Auto-Attack from 9 to K+V-1 (denoted as Auto-Attack$^+$) to avoid overestimation by targeting the VS class.

Table 6: Test robustness (%) on WRN-34-10 CIFAR10. VS labels improve robustness of all defenses.

Defense	Para. β, α	Clean (↑)	PGD -20 (↑)	CW (↑)	AA (↑)	AA$^+$ (↑)
AT	-, 0	**87.42**	54.87	54.49	51.88	51.88
AT$_{VS}$	-, 0.9	87.37	**56.67**	**56.69**	**54.72**	**53.16**
TRADES	6, 0	86.26	57.21	55.24	54.01	54.01
TRADES$_{VS}$	12, 0.6	86.26	**58.34**	**57.43**	**57.11**	**54.80**
AWP	6, 0	85.65	59.82	57.61	56.20	56.20
AWP$_{VS}$	12, 0.8	**86.30**	**61.96**	**61.93**	**60.68**	**57.34**
RST	6, 0	89.49	62.92	60.88	59.57	59.57
RST$_{VS}$	10, 0.7	**89.62**	**63.63**	**63.78**	**62.24**	**59.94**

Robustness Following the training suggesting from (He et al. 2019; Gowal et al. 2020), we retrain AT, TRADES, AWP (with TRADES header), and RST to build strong baselines. For TRADES$_{VS}$, AWP$_{VS}$ and RST$_{VS}$, we increase the weight of the robust regularization term (i.e., β in Equation (7) in the Appendix) to re-balance the trade-off between accuracy and robustness since the CE loss over VS labels

Table 7: Test robustness (%) on WRN-34-10 CIFAR100. AT_{VS} obtains higher robustness than more costly TRADES.

Defense	Para. β, α	Clean (↑)	PGD -20 (↑)	CW (↑)	AA (↑)	AA^+ (↑)
AT	-, 0	**62.85**	32.41	30.81	28.10	28.13
AT_{VS}	-, 0.75	62.50	**32.07**	**31.56**	**29.70**	**29.00**
TRADES	6, 0	62.84	32.39	29.89	28.82	28.64
$TRADES_{VS}$	18, 0.6	**63.26**	**34.22**	**30.53**	**30.07**	**29.19**
AWP	6, 0	62.59	34.63	30.67	29.66	29.63
AWP_{VS}	18, 0.8	**62.95**	**36.58**	**31.82**	**31.67**	**30.71**

tends to be larger than one-hot labels[2].

Tab 6 and Tab 7 show the test robustness on WRN-34-10 over CIFAR10 and CIFAR100. In general, VS labels consistently improve the worst model robustness of all defenses under Auto-Attack$^+$ (AA^+), while maintaining similar or even slightly higher clean accuracy. For example, on CIFAR10, VS labels improve AT's AA^+ robustness from 51.88% to 53.16%. On more complex CIFAR100, AT_{VS} even achieves higher robustness to TRADES using about it's half the training cost[3]. These results demonstrate that VS regularization is effective in improving robustness.

Further, an interesting result in Tab 6 and Tab 7 is that the robustness of VS models under standard Auto-Attack (AA) is significantly higher than other methods. We emphasize that this is not regular gradient obfuscation (Athalye, Carlini, and Wagner 2018), but that the *robustified* competition between the correct class and VS classes effectively defends targeted attacks, e.g., targeted DLR(Croce and Hein 2020b), from minimizing confidence over VS classes and maximizing confidence over incorrect classes. These results mean that multi-targeted attacks have to consume more computational cost to decrease the robustness of VS models. More analysis can be found in Sec D in the Appendix.

Visualization of Extracted Input Features We calculate the loss gradients w.r.t., input pixels to visualize extracted input features (Tsipras et al. 2019). Specifically, we calculate $\nabla_x f_\theta(x)_k$ and $\nabla_x \sum_{v=K+1}^{K+V} f_\theta(x)_v$ to visualize input features extracted by the correct class (VS.correct) and the VS classes (VS.vs) respectively. The selected model is TRADES on WRN-34-10 (α=0.6) over CIFAR10.

As depicted in Fig 3, the information extracted by the correct classes in both the VS (VS.correct) and LS (LS.correct) models adequately identifies the input samples. In the VS model, an optimal scenario occurs where most irrelevant features, such as green leaves and red watermarks unrelated to the true labels, are excluded from the correct class but captured by the VS classes (VS.vs). In contrast, the features extracted by the correct class and the incorrect class (LS.incorrect) in the LS model are difficult to distinguish.

Comparison to Label Smoothing Existing work (Pang et al. 2020) shows that LS improves standard AT to some extent. We compare VS and LS labels on AT in Tab 8, where

[2]Some recent work has further improved the record on RobustBench, e.g., (Wang et al. 2023), which uses data generated by the diffusion model. However, we do not consider these works as they do not introduce new loss heads and our conclusions apply to them.

[3]TRADES's KL consumes double the training cost of AT.

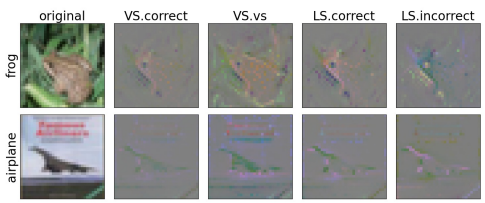


Figure 3: Visualization of extracted input features. 'VS.correct' and 'VS.vs' ('LS.correct' and 'LS.incorrect') represent features extracted by the correct class and VS (incorrect) classes in the VS (LS) model. An ideal case in the VS model is that most green leaves and red watermarks are better excluded from correct classes, which is more in line with human perception.

Table 8: Comparison to Label Smoothing on AT.

Dataset	Label	Clean	PGD	CW	AA^+ (imp.)
CIFAR10	one-hot	**87.42**	54.87	54.49	51.88
	LS (α=0.7)	87.16	55.92	54.70	52.79 (+ 0.91)
	VS (α=0.9)	**87.37**	**56.67**	**56.69**	**53.16** (+ **1.28**)
CIFAR100	one-hot	**62.85**	32.41	30.81	28.13
	LS (α=0.6)	62.21	**33.89**	30.86	28.68 (+ 0.55)
	VS (α=0.75)	**62.50**	32.07	**31.56**	**29.00** (+ **0.87**)

Table 9: Robustness (%) of teacher ResNet-18.

	Clean / Auto-Attack$^+$		
Dataset	one-hot	LS (α)	VS (α)
CIFAR10	84.24 / 48.87	84.38 / 49.66 (0.6)	**84.40** / **50.03** (0.85)
CIFAR100	58.08 / 25.77	58.90 / 26.50 (0.8)	**59.28** / **26.75** (0.8)

Table 10: Robust performance (%) of students.

		Clean / Auto-Attack$^+$			
Dataset	Student	plain	one-hot	LS	VS
CIFAR10	MobileNet-V2	82.59 / 47.24	**82.70** / 48.77	82.07 / 49.37	82.33 / **49.79**
CIFAR100	MobileNet-V2	**57.15** / 25.48	55.72 / 26.37	52.79 / 25.83	56.29 / **26.68**
CIFAR10	ResNet-18	84.24 / 48.87	**84.81** / 49.25	84.23 / 49.69	84.56 / **49.90**
CIFAR100	ResNet-18	58.08 / 25.77	59.45 / 26.76	58.08 / 27.39	**59.22** / **27.52**

the model is WRN-34-10. We see that VS labels again outperform LS labels in improving robustness. Next, we further study the effects of their learned robust representations in the logits on downstream adversarial robust distillation.

Adversarial Robust Distillation We study the effect of using robust models trained on different labels on the robustness of students. The teacher of ResNet-18 is AT WRN-34-10 trained in Sec 4.2; the teacher of MobileNet-V2 is AT ResNet-18 trained using settings in Sec 4.2. Tab 9 shows the performance of teacher AT ResNet-18, where VS labels again outperform one-hot and LS labels. We train students for 200 epochs and decay LR at the 150-th and 180-th epochs but set the weight decay to 0.0002 for MobileNet-V2. The interpolation parameter γ and temperature parameter τ are both set to 1 following (Goldblum et al. 2020). Here, we also consider VS models' first K-way as teachers.

As shown in Tab 10, VS teachers teach students to achieve higher robustness than one-hot and LS teachers. Like KD in Sec 4.1, LS teachers achieve higher robustness than one-hot teachers but teach a worse MobileNet-V2 on CIFAR100, hurting ARD. These results prove the better regularization

Table 11: OOD detection (%).

D^{in}	Method	Acc	AUC	FPR-95	D^{in}	Method	Acc	AUC	FPR-95
CIFAR10	OE	94.89	98.93	3.44	CIFAR100	OE	77.92	91.42	34.44
	OE + LS	95.42	99.10	2.53		OE + LS	**78.16**	91.37	36.78
	OE+VS$_1$	95.12	99.20	2.48		OE+VS$_1$	76.00	**92.84**	**30.24**
	OE+VS$_2$	**95.44**	**99.27**	**2.16**		OE+VS$_2$	77.90	92.77	30.77
	SSL	94.43	99.03	2.97		SSL	**75.63**	91.14	37.82
	SSL + LS	94.21	99.23	3.18		SSL + LS	75.39	91.90	36.29
	SSL+VS	**94.74**	**99.39**	**2.10**		SSL+VS	75.58	**93.00**	**33.22**

Table 12: Effect of varying V on the accuracy.

	Model [α]	0	3	5	10	15	20
CF10	ResNet-18 [0.5]	94.56	94.95	95.03	**95.3**	95.29	95.03
	ResNeXt-29 [0.8]	93.86	94.39	94.83	95.27	95.27	**95.49**
	Model [α]	0	10	50	100	150	200
CF100	ResNet-18 [0.8]	75.43	76.79	77.29	**78.05**	77.59	77.53
	ResNeXt-29 [0.9]	76.54	76.44	79.74	79.84	79.48	**80.01**

effect of VS labels without hurting the representation in the logits for the downstream ARD.

4.3 Out-Of-Distribution (OOD) Detection

In OOD detection, semi-supervised OE (Hendrycks, Mazeika, and Dietterich 2018) and SSL (Mohseni et al. 2020) achieve and maintain leading performance by using diverse, real-word OOD samples. We apply VS labels to them to study the effect of VS regularization on OOD detection[4]. For OE with a uniform distribution, we consider two options for generating pseudo-labels for OOD samples: (1) set the confidence of each real class and each VS class to $(1-\alpha)/K$ and α/V respectively (denoted as VS$_1$); (2) set the confidence of each real class to 0 but each VS class to $1/V$ (denoted as VS$_2$). For SSL using multiple reject classes, we treat its reject classes as ordinary classes and add V VS classes and use Eq (3) to generate pseudo-labels. We set a small α, i.e., 0.05, for all experiments since OE and SSL have regularized the overconfidence from the perspective of auxiliary OOD data.

Tab 11 shows the results, where the number of reject classes in SSL is K. The test OOD dataset is a mixture of six test sets: Textures (Cimpoi et al. 2014), Places365 (Zhou et al. 2017), iSUN (Xu et al. 2015), LSUN (crop), LSUN (resize) (Yu et al. 2015), and SVHN (Netzer et al. 2011). AUC is the area under the ROC curve (higher is better). FPR-95 is the false recognition rate for ID samples when 95% of OOD samples are correctly identified (lower is better). Note that significantly enhancing these two state-of-the-art methods poses a challenge, considering their good-enough performance in OOD detection (Augustin, Meinke, and Hein 2020). VS labels achieve better detection performance compared to one-hot and LS labels, demonstrating their superior regularization effect on OOD data.

4.4 Ablation Studies

In this section, we study how different confidences and numbers of VS classes affect accuracy, with further analysis on the robustness in Sec E in the Appendix.

[4]We replaced their originally used 80 million Tiny Images with 300K random images (https://github.com/hendrycks/outlier-exposure) due to their containing offensive elements.

Table 13: Effect of varying the confidence of VS classes on the accuracy. '0' denotes both V and α are 0.

Data	Model	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
CF10	ResNet-18	94.56	94.69	94.91	95.21	95.20	**95.3**	95.12	95.20	95.27	94.98
	ResNeXt-29	93.86	94.44	94.59	94.76	95.02	95.19	94.88	95.16	**95.27**	94.91
CF100	ResNet-18	75.43	76.29	76.64	76.94	77.04	77.06	76.99	77.67	**78.05**	77.51
	ResNeXt-29	76.54	76.59	77.06	77.86	77.74	77.23	77.72	78.58	79.06	**79.84**

Varying the Number of VS Classes We fix the confidence of VS classes and vary the number (V) of VS classes to observe its effect on model accuracy in standard training, with the results presented in Tab 12. For ResNet-18, the optimal Vs are 10 and 100 for CIFAR10 (CF10) and CIFAR100 (CF100), respectively, while for ResNeXt-29, they are 20 and 200. Notably, setting V to 10 and 100 for CIFAR10 and CIFAR100 on ResNeXt-29 achieves performance very close to that of setting V to 20 and 200. These results support our analysis in Sec 3.2, which suggests that a larger V encourages the penultimate layer activation to be equidistant from more VS class templates and enhances performance (if the DNN is sufficiently powerful). In practice, setting V to the number of original real classes is usually adequate.

Varying the Confidence of VS Classes We fix V to K and vary the confidence α to test its effect on accuracy in standard training. Tab 13 shows the results. On the same dataset, the larger ResNeXt-29 requires a larger α than ResNet-18 to achieve peak accuracy. Furthermore, both ResNet-18 and ResNeXt-29 on the more complex CIFAR100 require a larger α to achieve higher accuracy than CIFAR10. These suggest that a more complex dataset requires a higher α if the model is powerful enough. Note that ResNet-18 in Tab 3 achieves peak accuracy with a much smaller α compared to Tab 13, because ResNet-18 is sufficiently powerful for CIFAR100 but not for ImageNet.

In Sec E.1 and Sec E.2 of the Appendix, we observe similar trends in adversarial training. Based on these, we summarize the guidelines for tuning V and α: (1) set V close to the number of original real classes, (2) assign a larger α if the model is powerful enough for the dataset. Besides, we conduct studies on the confidence distribution over VS classes in Sec E.3 in the Appendix, which shows that assigning VS classes a uniform distribution yields the best performance.

5 Conclusion

Training on one-hot labels easily leads to overconfidence. While label smoothing regularizes the overconfidence from the perspective of labels, it hurts representations in the logits about sample similarities and may hurt calibration especially when a larger α is required for optimal performance. To address these limitations, we propose Virtual Smoothing (VS) labels, which introduce additional VS classes as adversaries to the original classes, clustering all input samples for regularization. Experiments demonstrate that VS labels can simultaneously improve accuracy and calibration while providing better logits for improved knowledge distillation. Additionally, VS labels prove to be more effective in enhancing adversarial training and downstream robust distillation, and out-of-distribution detection.

References

Alayrac, J.-B.; Uesato, J.; Huang, P.-S.; Fawzi, A.; Stanforth, R.; and Kohli, P. 2019. Are labels required for improving adversarial robustness? In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 12214–12223.

Andriushchenko, M.; Croce, F.; Flammarion, N.; and Hein, M. 2020. Square attack: A query-efficient black-box adversarial attack via random search. In *Proceedings of European Conference on Computer Vision (ECCV)*, 484–501.

Athalye, A.; Carlini, N.; and Wagner, D. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of International Conference on Machine Learning (ICML)*, 274–283.

Augustin, M.; Meinke, A.; and Hein, M. 2020. Adversarial robustness on in-and out-distribution improves explainability. In *European Conference on Computer Vision (ECCV)*, 228–245. Springer.

Bai, Y.; Feng, Y.; Wang, Y.; Dai, T.; Xia, S.-T.; and Jiang, Y. 2019. Hilbert-based generative defense for adversarial examples. In *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, 4784–4793.

Carlini, N.; and Wagner, D. 2017. Towards evaluating the robustness of neural networks. In *Proceedings of Symposium on Security and Privacy (SP)*, 39–57.

Carmon, Y.; Raghunathan, A.; Schmidt, L.; Duchi, J. C.; and Liang, P. S. 2019. Unlabeled data improves adversarial robustness. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 11192–11203.

Chen, H.-Y.; Wang, P.-H.; Liu, C.-H.; Chang, S.-C.; Pan, J.-Y.; Chen, Y.-T.; Wei, W.; and Juan, D.-C. 2018. Complement objective training. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Cimpoi, M.; Maji, S.; Kokkinos, I.; Mohamed, S.; and Vedaldi, A. 2014. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3606–3613.

Croce, F.; Andriushchenko, M.; Sehwag, V.; Debenedetti, E.; Flammarion, N.; Chiang, M.; Mittal, P.; and Hein, M. 2020. RobustBench: A standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*.

Croce, F.; and Hein, M. 2020a. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *Proceedings of International Conference on Machine Learning (ICML)*, 2196–2205.

Croce, F.; and Hein, M. 2020b. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *Proceedings of International Conference on Machine Learning (ICML)*, 2206–2216.

Dong, Y.; Fu, Q.-A.; Yang, X.; Pang, T.; Su, H.; Xiao, Z.; and Zhu, J. 2020. Benchmarking adversarial robustness on image classification. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 321–331.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Feinman, R.; Curtin, R. R.; Shintre, S.; and Gardner, A. B. 2017. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*.

Goldblum, M.; Fowl, L.; Feizi, S.; and Goldstein, T. 2020. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 3996–4003.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2020. Generative adversarial networks. *Communications of the ACM*, 63(11): 139–144.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Gowal, S.; Qin, C.; Uesato, J.; Mann, T.; and Kohli, P. 2020. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*.

Gowal, S.; Uesato, J.; Qin, C.; Huang, P.-S.; Mann, T.; and Kohli, P. 2019. An alternative surrogate loss for pgd-based adversarial testing. *arXiv preprint arXiv:1910.09338*.

Guo, C.; Pleiss, G.; Sun, Y.; and Weinberger, K. Q. 2017. On calibration of modern neural networks. In *International conference on machine learning*, 1321–1330. PMLR.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, T.; Zhang, Z.; Zhang, H.; Zhang, Z.; Xie, J.; and Li, M. 2019. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 558–567.

Hendrycks, D.; and Gimpel, K. 2016. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*.

Hendrycks, D.; Mazeika, M.; and Dietterich, T. 2018. Deep anomaly detection with outlier exposure. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Hinton, G.; Vinyals, O.; Dean, J.; et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Kurakin, A.; Goodfellow, I.; and Bengio, S. 2017. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*.

Lee, K.; Lee, K.; Lee, H.; and Shin, J. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. *Proceedings of Advances in Neural Information Processing Systems (NeuIPS)*, 31.

Ma, X.; Li, B.; Wang, Y.; Erfani, S. M.; Wijewickrema, S.; Schoenebeck, G.; Song, D.; Houle, M. E.; and Bailey, J. 2018. Characterizing adversarial subspaces using local intrinsic dimensionality. *arXiv preprint arXiv:1801.02613*.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*.

Mohseni, S.; Pitale, M.; Yadawa, J.; and Wang, Z. 2020. Self-supervised learning for generalizable out-of-distribution detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, 5216–5223.

Müller, R.; Kornblith, S.; and Hinton, G. E. 2019. When does label smoothing help? *Proceedings of Advances in Neural Information Processing Systems (NeuIPS)*, 32.

Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

Pang, T.; Yang, X.; Dong, Y.; Su, H.; and Zhu, J. 2020. Bag of tricks for adversarial training. *arXiv preprint arXiv:2010.00467*.

Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; and Swami, A. 2016. Distillation as a defense to adversarial perturbations against deep neural networks. In *Proceedings of Symposium on Security and Privacy (SP)*, 582–597.

Rice, L.; Wong, E.; and Kolter, Z. 2020. Overfitting in adversarially robust deep learning. In *Proceedings of International Conference on Machine Learning (ICML)*, 8093–8104.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. 2015. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3): 211–252.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of IEEE/CVF Computer Vision and Pattern Recognition (CVPR)*, 2818–2826.

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2013. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*.

Tramer, F.; Carlini, N.; Brendel, W.; and Madry, A. 2020. On adaptive attacks to adversarial example defenses. *arXiv preprint arXiv:2002.08347*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Tsipras, D.; Santurkar, S.; Engstrom, L.; Turner, A.; and Madry, A. 2019. Robustness may be at odds with accuracy. In *Proceedings of International Conference on Learning Representations (ICLR)*.

Vernekar, S.; Gaurav, A.; Denouden, T.; Phan, B.; Abdelzad, V.; Salay, R.; and Czarnecki, K. 2019. Analysis of confident-classifiers for out-of-distribution detection. *arXiv preprint arXiv:1904.12220*.

Wang, Z.; Pang, T.; Du, C.; Lin, M.; Liu, W.; and Yan, S. 2023. Better diffusion models further improve adversarial training. In *International Conference on Machine Learning*, 36246–36263. PMLR.

Wu, D.; Xia, S.-T.; and Wang, Y. 2020. Adversarial weight perturbation helps robust generalization. In *Proceedings of Neural Information Processing Systems (NeurIPS)*, 2958–2969.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 1492–1500.

Xu, P.; Ehinger, K. A.; Zhang, Y.; Finkelstein, A.; Kulkarni, S. R.; and Xiao, J. 2015. Turkergaze: Crowdsourcing saliency with webcam based eye tracking. *arXiv preprint arXiv:1504.06755*.

Yang, K.; Zhou, T.; Zhang, Y.; Tian, X.; and Tao, D. 2021. Class-disentanglement and applications in adversarial detection and defense. *Proceedings of Advances in Neural Information Processing Systems (NeuIPS)*, 34: 16051–16063.

Yu, F.; Seff, A.; Zhang, Y.; Song, S.; Funkhouser, T.; and Xiao, J. 2015. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*.

Yuan, L.; Chen, Y.; Wang, T.; Yu, W.; Shi, Y.; Jiang, Z.-H.; Tay, F. E.; Feng, J.; and Yan, S. 2021. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision*, 558–567.

Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; and Jordan, M. 2019. Theoretically principled trade-off between robustness and accuracy. In *Proceedings of International Conference on Machine Learning (ICML)*, 7472–7482.

Zhou, B.; Lapedriza, A.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6): 1452–1464.

Reproducibility Checklist

1. This paper:
   (a) Includes a conceptual outline and/or pseudocode description of AI methods introduced. **yes**
   (b) Clearly delineates statements that are opinions, hypothesis, and speculation from objective facts and results. **yes**
   (c) Provides well marked pedagogical references for less-familiare readers to gain background necessary to replicate the paper. **yes**
2. Does this paper make theoretical contributions? **yes**
   (a) All assumptions and restrictions are stated clearly and formally. **yes**
   (b) All novel claims are stated formally. **yes**
   (c) Proofs of all novel claims are included. **yes**
   (d) Proof sketches or intuitions are given for complex and/or novel results. **yes**
   (e) Appropriate citations to theoretical tools used are given. **yes**
   (f) All theoretical claims are demonstrated empirically to hold. **yes**

(g) All experimental code used to eliminate or disprove claims is included. **yes**

3. Does this paper rely on one or more datasets? **yes**

(a) A motivation is given for why the experiments are conducted on the selected datasets. **yes**

(b) All novel datasets introduced in this paper are included in a data appendix. **yes**

(c) All novel datasets introduced in this paper will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **yes**

(d) All datasets drawn from the existing literature (potentially including authors' own previously published work) are accompanied by appropriate citations. **yes**

(e) All datasets drawn from the existing literature (potentially including authors' own previously published work) are publicly available. **yes**

(f) All datasets that are not publicly available are described in detail, with explanation why publicly available alternatives are not scientifically satisficing. **NA**

4. Does this paper include computational experiments? **yes**

(a) Any code required for pre-processing data is included in the appendix. **yes**

(b) All source code required for conducting and analyzing the experiments is included in a code appendix. **yes**

(c) All source code required for conducting and analyzing the experiments will be made publicly available upon publication of the paper with a license that allows free usage for research purposes. **partial**

(d) All source code implementing new methods have comments detailing the implementation, with references to the paper where each step comes from. **yes**

(e) If an algorithm depends on randomness, then the method used for setting seeds is described in a way sufficient to allow replication of results. **yes**

(f) This paper specifies the computing infrastructure used for running experiments (hardware and software), including GPU/CPU models; amount of memory; operating system; names and versions of relevant software libraries and frameworks. **partial**

(g) This paper formally describes evaluation metrics used and explains the motivation for choosing these metrics. **yes**

(h) This paper states the number of algorithm runs used to compute each reported result. **yes**

(i) Analysis of experiments goes beyond single-dimensional summaries of performance (e.g., average; median) to include measures of variation, confidence, or other distributional information. **no**

(j) The significance of any improvement or decrease in performance is judged using appropriate statistical tests (e.g., Wilcoxon signed-rank). **partial**

(k) This paper lists all final (hyper-)parameters used for each model/algorithm in the paper's experiments. **yes**

(l) This paper states the number and range of values tried per (hyper-) parameter during development of the paper, along with the criterion used for selecting the final parameter setting. **yes**