A PROOFS

A.1 PROOFS AND REMARKS FOR IMPLICIT LABEL NOISE

Total variation distance.

For two discrete probability distributions $p(y)$ and $p(y')$ where $y, y' \in \mathcal{Y}$, the total variation distance between them can be equally defined as

$$\begin{aligned}\|p(y) - p(y')\|_{\text{TV}} &= \max_J \left| \sum_{j \in J} p(y = j) - \sum_{j \in J} p(y' = j) \right| \\ &= \frac{1}{2} \sum_j |p(y = j) - p(y' = j|)|\end{aligned}$$

Proof of Theorem 3.1. This theorem can be easily proven by the *coupling inequality*.

Proof of Theorem 3.2.

Proof. For simplicity, we consider the adversarial perturbation generated by FGSM. Other adversarial perturbation can be viewed as a Taylor series of such perturbation.

$$\delta^* \approx -\varepsilon \frac{\nabla_x \ \max_j p(y^* = j|x)}{\|\nabla_x \ \max_j p(y^* = j|x)\|}, \tag{12}$$

Now let $j^* = \arg\max \ p(y = j|x)$. Recall $f^*(x) = p(y^*|x)$, $f^*(x_\delta) = p(y^*_\delta|x_\delta)$ and $p(y_\delta|x_\delta) = p(y^*|x)$. We have

$$\begin{aligned}\min p(y_\delta \neq y^*_\delta|x_\delta) &= \|p(y^*|x) - p(y^*_\delta|x_\delta)\|_{\text{TV}} \\ &= \frac{1}{2} \sum_j |p(y^* = j|x) - p(y^*_\delta = j|x_\delta)| \quad \boxed{\text{TV distance}} \\ &\geq \frac{1}{2} |p(y^* = j^*|x) - p(y^*_\delta = j^*|x_\delta)| \\ &= \frac{1}{2} |f^*(x)_{j^*} - f^*(x + \delta)_{j^*}| \\ &= \frac{1}{2} [f^*(x)_{j^*} - f^*(x + \delta)_{j^*}] \quad \boxed{\text{Adversarial perturbation}} \\ &= \frac{1}{2} \left[-\nabla_x f^*(x)_{j^*} \cdot \delta - \frac{1}{2} \delta^T \nabla^2 f^*(z)_{j^*} \delta \right] \\ &\geq \frac{1}{2} \left[-\nabla_x f^*(x)_{j^*} \cdot \delta - \frac{M}{2} \|\delta\|_2^2 \right] \quad \boxed{\text{Local convexity}} \\ &\geq \frac{1}{2} \left[\varepsilon \frac{\|\nabla_x f^*(x)_{j^*}\|_2^2}{\|\nabla_x f^*(x)_{j^*}\|} - \frac{M}{2} \|\delta\|_2^2 \right] \\ &\geq \frac{1}{2} \left[\varepsilon \|\nabla_x f^*(x)_{j^*}\|_2 - \frac{M}{2} \|\delta\|_2^2 \right] \\ &\geq \frac{1}{2} \left[\varepsilon \|\nabla_x f^*(x)_{j^*}\|_\infty - \frac{M}{2} \|\delta\|_2^2 \right],\end{aligned}$$

where the last two inequalities leverage the fact that $\|\cdot\|_\infty \leq \|\cdot\|_2$.

With the assumption

$$\|\nabla_x \ f^*(x)_j\| \propto \begin{cases} 1 - f^*(x)_j, & f^*(x)_j \to 1 \\ f^*(x)_j, & f^*(x)_j \to 0, \end{cases}$$

we then have

$$\underline{\min}\, p(\hat{y} \neq y | x + \delta) \propto \varepsilon(1 - p(y = j^* | x)),$$

where $\underline{\min}$ means the lower bound of the minimum label noise.

□

Sanity check of Assumption 7. Here we check if Assumption 7 is reasonable by studying a Gaussian mixture model (GMM).

The conditional distribution in a Gaussian mixture model can be formulated as

$$p(y = j|x) \equiv f^*(x)_j = \frac{\psi_j \mathcal{N}_x(\mu_j, \sigma_j)}{\sum_l \psi_l \mathcal{N}_x(\mu_l, \sigma_l)} \equiv \frac{\psi_j g_j(x)}{\sum_l \psi_l g_l(x)},$$

where

$$g_l(x) = \frac{1}{\sqrt{\det(2\pi\sigma_l)}} \exp\left[-\frac{1}{2}(x - \mu_l)^T \sigma_l^{-1}(x - \mu_l)\right].$$

The gradient can be derived as

$$\begin{aligned}
\nabla_x f^*(x)_j &= \frac{\psi_j \nabla g_j(x)}{\sum_l \psi_l g_l(x)} - \psi_j g_j(x) \frac{\sum_l \psi_l \nabla g_l(x)}{\left[\sum_l \psi_l g_l(x)\right]^2} \\
&= \frac{-\psi_j g_j(x) \sigma_j^{-1}(x - \mu_j)}{\sum_l \psi_l g_l(x)} + \psi_j g_j(x) \frac{\sum_l \psi_l g_l(x) \sigma_l^{-1}(x - \mu_l)}{\left[\sum_l \psi_l g_l(x)\right]^2} \\
&= \frac{\psi_j g_j(x)}{\sum_l \psi_l g_l(x)} \left[-\sigma_j^{-1}(x - \mu_j) + \frac{\sum_l \sigma_l^{-1}(x - \mu_l)\psi_l g_l(x)}{\sum_l \psi_l g_l(x)}\right] \\
&= f^*(x)_j \left[-\sigma_j^{-1}(x - \mu_j) + \sum_l f^*(x)_l \sigma_l^{-1}(x - \mu_l)\right]
\end{aligned}$$

When $x - \mu_j \to 0$, $f^*(x)_j \to 1$ and $f^*(x)_l \to 0$, for $l \neq j$,

$$\nabla_x f^*(x)_j \approx f^*(x)_j (f^*(x)_j - 1)\sigma_j^{-1}(x - \mu_j)$$

Therefore

$$\|\nabla_x f^*(x)_j\| \propto 1 - f^*(x)_j$$

A.2 Proofs for mitigating double descent

Proof of Theorem 4.1.

Proof. Let $j^* = \text{argmax}\ p(y_\delta = j|x_\delta)$ and thus $p(y_\delta = j^*|x_\delta) \in [1/c, 1]$. Let $g(T) := f(x_\delta; \theta, T)_{j^*}$, which is a continuous function defined on $[0, \infty]$. The condition $j^* = \arg\max_j f(x_\delta; \theta, T)_j$ ensures that $g(T) \in [1/c, 1]$, where c is the number of classes. By the intermediate value theorem, there exists T^*, such that $g(T^*) = p(y_\delta = j^*|x_\delta)$.

Let $T = T^*$, we have

$$\begin{aligned}
\|f(x_\delta; \theta, T) - p(y_\delta^*|x_\delta)\|_{TV} &= \frac{1}{2}\sum_j |f(x_\delta; \theta, T)_j - p(y_\delta^* = j|x_\delta)| \\
&= \frac{1}{2}\sum_{j, j\neq j^*} |f(x_\delta; \theta, T)_j - p(y_\delta^* = j|x_\delta)| \\
&\leq \frac{1}{2}\left[\sum_{j, j\neq j^*} f(x_\delta; \theta, T)_j + \sum_{j, j\neq j^*} p(y_\delta^* = j|x_\delta)\right] \\
&= 1 - p(y_\delta = j^*|x_\delta),
\end{aligned}$$

where the inequality holds by the triangle inequality.

Meanwhile, we have

$$\begin{aligned}\|p(\tilde{y}_\delta|x_\delta) - p(y^*_\delta|x_\delta)\|_{TV} &= \|p(y|x) - p(y^*_\delta|x_\delta)\|_{TV} \\ &= \|\mathbb{1}(y) - p(y^*_\delta|x_\delta)\|_{TV} \\ &= \frac{1}{2}\left[1 - p(y^*_\delta = y|x_\delta) + \sum_{j,j\neq\hat{y}} p(y^*_\delta = y|x_\delta)\right] \\ &= 1 - p(y^*_\delta = y|x_\delta) \\ &\geq 1 - p(y^*_\delta = j^*|x_\delta).\end{aligned}$$

Therefore, it can seen that for $T = T^*$,

$$\|f(x_\delta;\theta,T) - p(y^*_\delta|x_\delta)\|_{TV} \leq \|p(\tilde{y}_\delta|x_\delta) - p(y^*_\delta|x_\delta)\|_{TV}.$$

□

Proof of Theorem 4.2.

Lemma A.1. *Let x_δ be an example incorrectly classified by a classifier f in terms of the true label distribution $p(y^*_\delta = j|x_\delta)$, namely*

$$\arg\max_j \ f(x_\delta;\theta,T)_j \neq j^*,$$

where $j^* = \arg\max_j \ p(y^*_\delta = j|x_\delta)$. Assume $p(y^*_\delta = j^*|x_\delta) \geq 1/2$, then

$$f(x_\delta;\theta,T)_{j^*} \leq p(y^*_\delta = j^*|x_\delta).$$

Proof. We prove it by contradiction. Assume $f(x_\delta;\theta,T)_{j^*} > p(y^*_\delta = j^*|x_\delta)$, we have $f(x_\delta;\theta,T)_{j^*} > p(y^*_\delta = j^*|x_\delta) \geq 1/2$. Therefore,

$$f(x_\delta;\theta,T)_j \leq \sum_{j,j\neq j^*} f(x_\delta;\theta,T)_j = 1 - f(x_\delta;\theta,T)_{j^*} < 1/2, \ \forall j \neq j^*,$$

which means $f(x_\delta;\theta,T)_j < f(x_\delta;\theta,T)_{j^*}, \ \forall j \neq j^*$. This leads to $j^* = \arg\max_j f(x_\delta;\theta,T)_j$, which contradicts our condition. □

Now we prove Theorem 4.2

Proof. First let $p(y_\delta|x_\delta) = p(y|x) \approx \mathbb{1}(y)$. This is our assumption. But the approx here would be a problem, we need exactly one-hot.

Let $j^* = \arg\max_j p(y^*_\delta = j|x_\delta)$. By Lemma A.1 we have $f(x_\delta;\theta,T)_{j^*} \leq p(y^*_\delta = j^*|x_\delta) \leq 1$. Then there exists $\lambda^* > 0$, such that $\lambda^* \cdot f(x_\delta;\theta,T)_{j^*} + (1-\lambda^*) = p(y^*_\delta = j^*|x_\delta)$ by the intermediate value theorem.

Let $\lambda = \lambda^*$, we have

$$
\begin{aligned}
&2\left[\|\lambda \cdot f(x_\delta;\theta,T) + (1-\lambda)\cdot p(\tilde{y}_\delta|x_\delta) - p(y_\delta^*|x_\delta)\|_{TV} - \|f(x_\delta;\theta,T) - p(y_\delta^*|x_\delta)\|_{TV}\right] \\
=&2\left[\|\lambda \cdot f(x_\delta;\theta,T) + (1-\lambda)\cdot \mathbb{1}(y) - p(y_\delta^*|x_\delta)\|_{TV} - \|f(x_\delta;\theta,T) - p(y_\delta^*|x_\delta)\|_{TV}\right] \\
=&\sum_j |\lambda \cdot f(x_\delta;\theta,T)_j + (1-\lambda)\cdot 1(j=y) - p(y_\delta^*=j|x_\delta)| - \sum_j |f(x_\delta;\theta,T)_j - p(y_\delta^*=j|x_\delta)| \\
=&\sum_j |\lambda \cdot f(x_\delta;\theta,T)_j + (1-\lambda)\cdot 1(j=y^*) - p(y_\delta^*=j|x_\delta)| - \sum_j |f(x_\delta;\theta,T)_j - p(y_\delta^*=j|x_\delta)| \\
=&\sum_{j,j\neq j^*} |\lambda \cdot f(x_\delta;\theta,T)_j - p(y_\delta^*=j|x_\delta)| - \sum_{j,j\neq j^*} |f(x_\delta;\theta,T)_j - p(y_\delta^*=j|x_\delta)| - |f(x_\delta;\theta,T)_{j^*} - p(y_\delta^*=j^*|x_\delta)| \\
\leq&\sum_{j,j\neq j^*} |\lambda \cdot f(x_\delta;\theta,T)_j - f(x_\delta;\theta,T)_j| - |f(x_\delta;\theta,T)_{j^*} - p(y_\delta^*=j^*|x_\delta)| \\
=&\sum_{j,j\neq j^*} [f(x_\delta;\theta,T)_j - \lambda \cdot f(x_\delta;\theta,T)_j] - [p(y_\delta^*=j^*|x_\delta) - f(x_\delta;\theta,T)_{j^*}] \\
=&\sum_{j,j\neq j^*} [f(x_\delta;\theta,T)_j - \lambda \cdot f(x_\delta;\theta,T)_j] - [\lambda \cdot f(x_\delta;\theta,T)_{j^*} + (1-\lambda) - f(x_\delta;\theta,T)_{j^*}] \\
=&\sum_j f(x_\delta;\theta,T)_j - \lambda \sum_j f(x_\delta;\theta,T)_j - (1-\lambda) \\
=&0.
\end{aligned}
$$

□

B More empirical analyses

B.1 Epoch-wise double descent is ubiquitous in adversarial training

In this section, we conduct extensive experiments with different optimizers, sample sizes, model architectures, and learning rate schedulers to verify the connection between robust overfitting and epoch-wise double descent. The default experiment settings are listed in Appendix F.2 in detail.

Optimizer. Similar to the setting employed in Nakkiran et al. (2020), we conduct the adversarial training using both the Adam optimizer and SGD. As already shown in Figure 1, double descent can be observed for both optimizers, although Adam may be inferior compared to SGD.

Sample size. We randomly sample a desired number of examples without replacement from the original training set in CIFAR-10. As shown in Figure 6, for both optimizers, increasing sample size will shrink the area under the double descent curve, but will not significantly distort its shape. Similar observation is also made for double descent curve under standard training (Nakkiran et al., 2020).

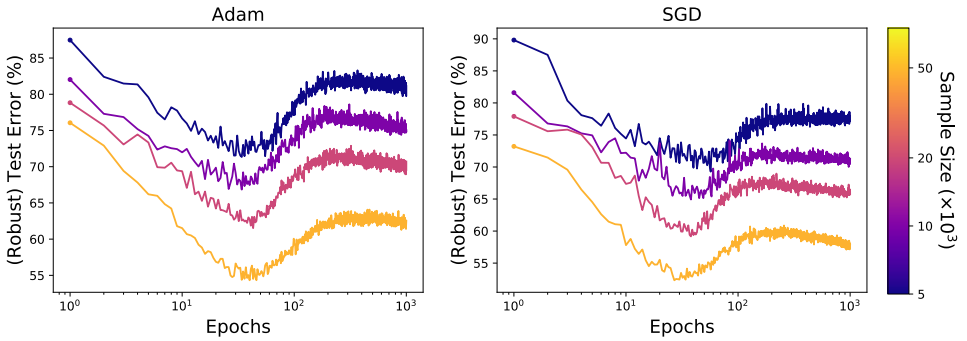
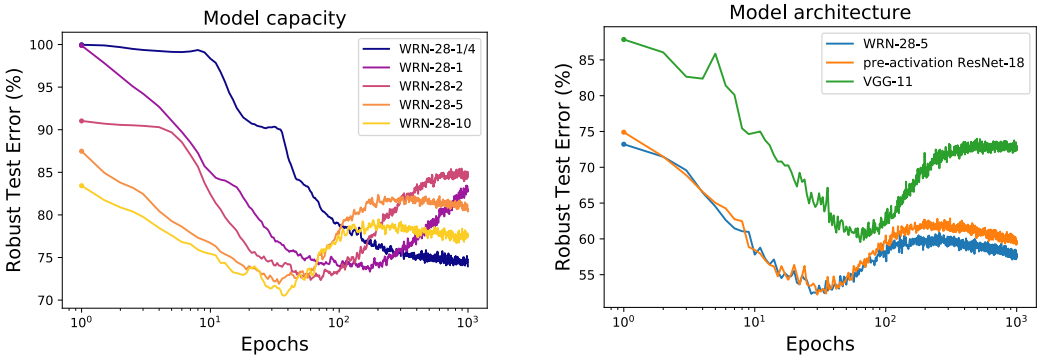


Figure 6: Varying sample size will shrink the area under the epoch-wise double descent curve, but will not significantly distort its shape.

Model capacity. We modulate the capacity of the deep model by varying the widening factor of the Wide ResNet. To extend the lower limit of the capacity, we allow the widening factor to be less

than 1, in which case the number of channels in each residual block is scaled similarly but rounded. The number of channels in the first convolutional layer will be reduced accordingly to ensure the width monotonically increasing through the forward propagation. To accelerate the training with an extremely large model, we randomly sample a training set of size 5000 and employ the Adam optimizer, since the sample size will not significantly distort the shape of the double descent as shown above. Figure 7a shows that the double descent will gradually become more complete as the model capacity increases and the model translates from under-parameterized to over-parameterized regime (Nakkiran et al., 2020).



(a) Similar to Figure 1 in the introduction, but include more widening factors to show the gradual transition of the model from under-parameterized to over-parameterized regime. The training curves are smoothed by a window of 5.

(b) Different model architectures will affect the double descent curve. In particular, VGG-11 will have the second descent delayed due to its inferior performance compared to residual architectures.

Figure 7: Effect of model on the epoch-wise double descent curve

Model architecture. We also experiment on model architectures other than Wide ResNet, including pre-activation ResNet-18 (He et al., 2016) and VGG-11 (Simonyan & Zisserman, 2015). We select these configurations to ensure approximately comparable model capacities[2]. As shown in Figure 7, different model architectures may produce slightly different double descent curves. The second descent of VGG-11 in particular will be delayed due to its inferior performance compared to residual architectures.

Learning rate scheduler. A specific learning rate scheduler may shape the robust overfitting differently as suggested by Rice et al. (2020). We consider the following learning rate schedulers in our experiments.

- **Piecewise decay**: The initial learning rate rate is set as 0.1 and is decayed by a factor of 10 at the 100th and 500th epochs within a total of 1000 epochs.
- **Cyclic**: This scheduler was initially proposed by Smith (2017) and has been popular in adversarial training. We set the maximum learning rate to be 0.2, and the learning rate will linearly increase from 0 to 0.2 for the initial 400 epochs and decrease to 0 for the later 600 epochs.
- **Cosine**: This scheduler was initially proposed by Loshchilov & Hutter (2017). The learning rate starts at 0.1 and gradually decrease to 0 following a cosine function for a total of 1000 epochs.

Experiments on various learning rate schedulers show the second descent can be widely observed except the piecewise decay, where the appearance of second descent might be delayed due to extremely small learning rate in the late stage of training. This further demonstrates the connection between robust overfitting and epoch-wise double descent.

B.2 Dependence of double descent in adversarial training

In this section, we show that the double descent in adversarial training is conditioned on the data quality. As models are trained on adversarial examples, data properties in adversarial training are

[2]WRN-28-5, pre-activation ResNet-18 and VGG-11 have 9.13×10^6, 11.17×10^6 and 9.23×10^6 parameters, respectively.

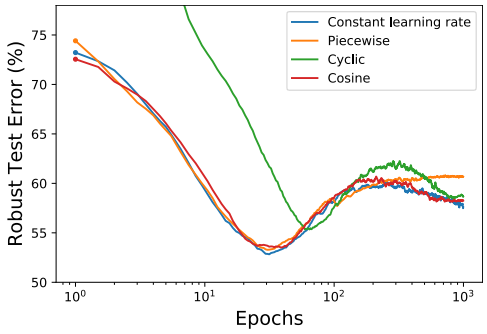


Figure 8: The effect of the learning rate scheduler on the epoch-wise double descent curve in adversarial training. Modulating the model capacity can produce training curves with diverse behaviors. Different model architectures may produce slightly different double descent curves. The training curve is smoothed by moving average with a window of 5.

concerned with the quality of the clean examples, and the adversary employed to generate the perturbation, which further depends on the perturbation radius and the number of attack iterations[3]. We show that the double descent in adversarial training has a strong dependence on the data quality and the perturbation radius, but almost no dependence on the number of attack iterations.

Perturbation radius. Overfitting has been shown to dominate in adversarial training, but rarely appear in standard training (Rice et al., 2020). This suggests the overfitting, or more generally double descent, is conditioned on the perturbation radius in adversarial training, since standard training and standard accuracy can be viewed as adversarial training and robust accuracy with zero perturbation radius, respectively. By modulating the perturbation radius, we show that such correlation is gradual.

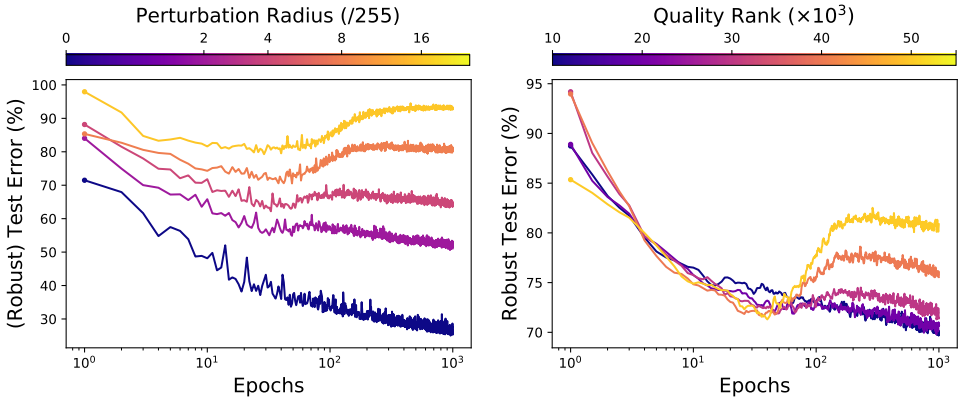


Figure 9: (Left) Dependence of double descent on the perturbation radius. $\varepsilon = 0/255$ indicates the standard training where no double descent occurs. (Right) Dependence of double descent on the data quality. The curves are smoothed by a window of 5 epochs to reduce overlapping.

As shown in Figure 9, the double descent emerges and exacerbates as the perturbation radius increases, indicating a strong correlation between perturbation radius and double descent in adversarial training. In particular, when the perturbation radius is around $4/255$, a second minimum can be observed which is equivalently good or even better than the first minimum. This again validates the strong connection between robust overfitting and double descent, and suggests longer training can still be helpful for adversarial training.

Data quality. Previous works have shown that some datasets such as ImageNet (Deng et al., 2009) may produce significantly stronger robust overfitting (Rice et al., 2020). It has also been observed that the low-quality data in the same dataset causes the robust overfitting (Dong et al., 2021a). This

[3]The best step size in terms of the model performance can often be determined from the combination of perturbation radius and attack iterations (Madry et al., 2018)

implies that the double descent in adversarial training hinges on the quality of the data. We measure the data quality using the predictive probability of a classifier (see Appendix B.2 for details), and sample training sets with different levels of data quality controlled by a threshold of the quality-based rank. Figure 9 shows that as the quality of the training set degrades, the double descent gradually emerges and exacerbates, indicating a strong correlation between the data quality and the double descent in adversarial training. One may again note that when the quality of the training data is relatively high, a second minimum can be observed which is equivalently good as the first minimum.

The number of attack iterations. We have shown that the double descent in adversarial training strongly depends on the perturbation radius. In this section we conduct experiments to show whether it also depends on the strength of the adversary.

In Figure 10, we fix the perturbation radius as $4/255$ where the double descent is relatively complete and vary the number of attack iterations of the PGD attack employed in the inner maximization. One may find that as long as the model capacity is reasonably large, the number of attack iterations will not significantly affect the double descent curve, both for epoch-wise one and model-wise one. From the analysis of implicit label noise, this is easy to understand as more attack iterations will not reduce the probability corresponding to the true label much more—it is widely observed more iterations in PGD attack only marginally increase the attack successful rate. Consequently, the distribution mismatch between the true label distribution and the assigned label distribution that induces the implicit label noise will not expand significantly.

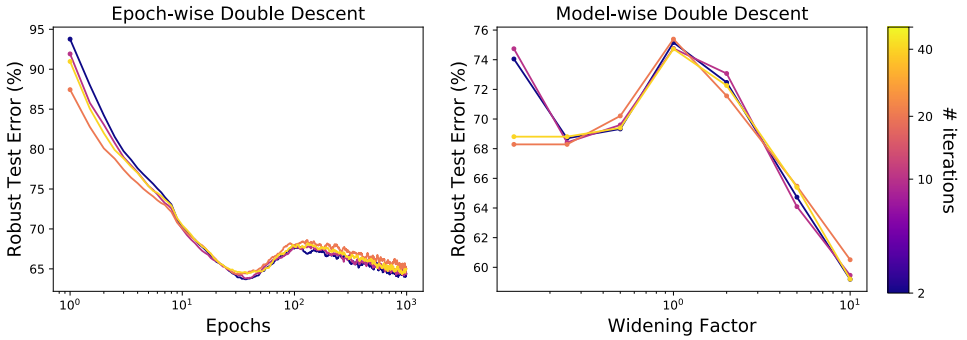


Figure 10: Dependence of double descent on the number of attack iterations. As more iterations are employed in the inner maximization, neither epoch-wise nor model-wise double descent changes significantly except for an extremely small model (WRN-28-1/8). For the model-wise double descent, only the test error at the best checkpoint is shown to avoid overlapped curves since the last checkpoint achieves similar error as the best checkpoint in this training setting.

B.3 Gaussian noise

One may expect adversarial training and adversarial augmentation (see Section 3.5) produce double descent simply by perturbing the inputs, thus increasing the variance. To verify the implicit label noise induced by a mismatch between the true label distribution and the assigned label distribution is essential to produce double descent, we select one type of common corruption, Gaussian noise, to perturb the image inputs. We first briefly show that Gaussian noise does not cause a distribution mismatch.

We follow the notation introduced in Section 3.1, where $\delta \in \mathbb{R}^d$ now refers to a Gaussian perturbation. Under a 1-st order approximation of $p(y|x)$, the mismatch between the true label distribution and label distribution can be derived as (see proof of Theorem 3.2 in Appendix A.1)

$$\|p(y^*_\delta|x_\delta) - p(\tilde{y}_\delta|x_\delta)\|_{\text{TV}} \equiv \|p(y^*_\delta|x_\delta) - p(y|x)\|_{\text{TV}} = \frac{1}{2}\sum_j \left|\nabla_x f^*(x)_j \cdot \delta + \frac{1}{2}\delta^T \nabla^2 f^*(z)_j \delta\right| \tag{13}$$

where $\delta \sim \mathcal{N}(0, \sigma \cdot I_d)$. Since the image inputs span a low-dimension subspace $\mathcal{X}$ and the dimensionality d is large (3072 for CIFAR-10), it is highly likely that $\delta \perp \mathcal{X}$, which means $\nabla_x f^*(x)_j \cdot \delta = 0$ and $\delta^T \nabla^2 f^*(z)_j \delta = 0$ for all j. One can also empirically verify that a Gaussian perturbation is almost always orthogonal to the difference between any two images, while an adversarial perturbation is not.

We now experiment on dataset perturbed by Gaussian noise and verify our intuition. Similar to adversarial augmentation, we apply Gaussian noise to the training set only once, and then conduct standard training on the perturbed training set. Aligned with the setting of adversarial augmentation (See Appendix F.3), we employ Adam to train a WRN-28-5 on randomly selected 5000 examples for 1000 epochs. As shown in Figure 11, Gaussian noise with a perturbation radius as high as $80/255$, which will reduce the accuracy of a classifier to the same level as an adversarial attack will, does not produce significant double descent. We note that similar observation has been made on common curruption benchmark CIFAR-10-C (Hendrycks & Dietterich, 2019) in a previous work (Yang et al., 2020b).

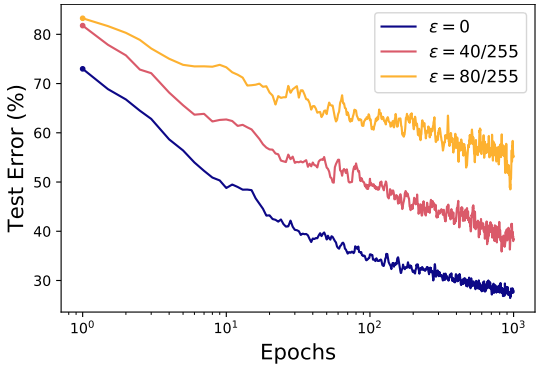


Figure 11: Even with extremely high Gaussian noise corrupting the training set, no significant double descent can be observed. This shows the input perturbation is not essential to produce double descent.

C More experiment results

C.1 Adversarial training methods, neural architectures and evaluation metrics

In this section we conduct extensive experiments with different adversarial training methods, neural architectures and robustness evaluation metrics to verify the effectiveness of our method.

Method	Setting	T	λ	Robust Acc. (%)			Standard Acc. (%)		
				Best	Last	Diff.	Best	Last	Diff.
TRADES	AT	-	-	48.50	45.53	2.97	82.79	82.68	0.11
	KD-AT	2	0.5	48.74	47.52	1.22	82.30	**83.03**	-0.73
	KD-AT-Auto	1.12^*	0.82^*	**48.75**	**48.39**	**0.36**	**82.44**	82.80	**-0.36**
FGSM	AT	-	-	41.96	35.39	6.57	85.91	87.20	-1.29
	KD-AT	2	0.5	42.82	41.61	1.21	86.69	**87.93**	-1.24
	KD-AT-Auto	2.18^*	0.78^*	**44.11**	**43.75**	**0.36**	**87.38**	87.66	**-0.28**

Table 2: Performance of our method with different adversarial training methods.

Architecture	Setting	T	λ	Robust Acc. (%)			Standard Acc. (%)		
				Best	Last	Diff.	Best	Last	Diff.
VGG-19	AT	-	-	42.21	39.12	3.09	73.95	80.45	-6.50
	KD-AT	2	0.5	43.59	42.69	0.90	74.30	**77.80**	-3.50
	KD-AT-Auto	1.28^*	0.79^*	**44.27**	**44.24**	**0.03**	**76.41**	76.79	**-0.38**
WRN-28-5	AT	-	-	49.85	42.89	6.96	84.82	85.87	-1.05
	KD-AT	2	0.5	51.08	48.40	2.68	85.36	**86.88**	-1.52
	KD-AT-Auto	1.6^*	0.82^*	**51.47**	**51.10**	**0.37**	**86.05**	86.24	**-0.19**
WRN-34-10	AT	-	-	52.29	46.04	6.25	86.57	86.75	-0.18
	KD-AT	2	0.5	53.11	50.97	2.14	86.41	**88.06**	-1.65
	KD-AT-Auto	1.6^*	0.83^*	**54.17**	**53.71**	**0.46**	**87.69**	88.01	**-0.32**

Table 3: Performance of our method with different neural architectures.

Attacks	Setting	T	λ	Robust Acc. (%)		
				Best	Last	Diff.
PGD-1000	AT	-	-	50.64	43.00	7.64
	KD-AT	2	0.5	51.79	48.43	3.36
	KD-AT-Auto	1.47^*	0.8^*	**52.05**	**51.71**	**0.34**
Square Attack	AT	-	-	53.47	48.90	4.57
	KD-AT	2	0.5	54.39	52.92	1.47
	KD-AT-Auto	1.28^*	0.79^*	**55.23**	**55.17**	**0.06**
RayS	AT	-	-	55.76	51.63	4.13
	KD-AT	2	0.5	56.59	55.50	1.09
	KD-AT-Auto	1.6^*	0.82^*	**57.74**	**57.54**	**0.20**

Table 4: Performance of our method under different adversarial attacks. PGD-1000 refers to PGD attack with 1000 attack iterations, with step size fixed as $2/255$ as recommended by Croce & Hein (2020).

C.2 Combined with additional techniques

Here, we show that combined with the additional techniques proposed in (Chen et al., 2021), our method can achieve better performance.

We note that our proposed method is essentially the baseline knowledge distillation for adversarial training with a robustly trained self-teacher, equipped with an algorithm that automatically finds its optimal hyperparameters (i.e. the temperature T and the interpolation ratio λ). Stochastic Weight Averaging (SWA) and additional standard teachers employed in (Chen et al., 2021) are orthogonal contributions. KD-AT-Auto can certainly be combined with SWA and KD-Std to achieve better performance. As shown in Table 5, on CIFAR-10, KD-AT + KD-Std + SWA (Chen et al., 2021) can already reduce the overfitting gap (difference between the best and last robust accuracy) to almost 0, while KD-AT-Auto + KD-Std + SWA maintains an overfitting gap close to 0. Interestingly, on the SVHN dataset (Netzer et al., 2011), where KD-AT + KD-Std + SWA still produces a high overfitting gap (also see Appendix A1.3 in (Chen et al., 2021)), KD-AT-Auto + KD-Std + SWA can further push this gap to close to 0.

Here, the interpolation ratio of the standard teacher is fixed as 0.2 and the SWA starts at the first learning rate decay for all experiments. We employ PGD-AT (Madry et al., 2018) as the base adversarial training method and conduct experiments with a pre-activation ResNet-18. The robust accuracy is evaluated with AutoAttack. Other experiment details are in line with Appendix F.1.

Furthermore, we note that (Chen et al., 2021) shows SWA and KD-Std are essential components to mitigate robust overfitting on top of KD-AT, while we show that KD-AT itself can mitigate robust overfitting by proper parameter tuning. We are thus able to separate these components and allow a more flexible selection of hyperparameters in diverse training scenarios without fear of overfitting. In particular, although (Chen et al., 2021) suggests SWA starting at the first learning rate decay (exactly when the overfitting starts) mitigates robust overfitting, the effectiveness of SWA on mitigating

Dataset	Setting	T	λ	Robust Acc. (%) Best	Robust Acc. (%) Last	Robust Acc. (%) Diff.	Standard Acc. (%) Best	Standard Acc. (%) Last	Standard Acc. (%) Diff.
CIFAR-10	AT	-	-	47.35	41.42	5.93	82.67	84.91	-2.24
	KD-AT + KD-Std + SWA	2	0.5	49.98	49.89	0.09	**85.06**	**85.52**	-0.46
	KD-AT-Auto + KD-Std + SWA	1.47*	0.8*	**50.03**	**50.05**	**-0.02**	84.69	84.91	**-0.22**
SVHN	AT	-	-	47.83	39.77	8.06	90.18	91.11	-0.93
	KD-AT + KD-Std + SWA	2	0.5	47.88	46.46	1.42	**91.59**	**91.76**	**-0.17**
	KD-AT-Auto + KD-Std + SWA	1.53*	0.83*	**50.58**	**50.09**	**0.49**	90.54	90.76	-0.22

Table 5: Performance of our method combined with SWA and additional standard teacher on different datasets.

overfitting may strongly depend on its hyper-parameter selection including s_0, i.e., the starting epoch and τ, i.e., the decay rate[4], which is also mentioned in recent work (Rebuffi et al., 2021). We also did some additional experiments on CIFAR-10 following the SWA setting in (Rebuffi et al., 2021) to demonstrate the wide applicability of our method. As shown by Table 6, when changing the hyperparameters of SWA, KD-AT + KD-Std + SWA cannot consistently mitigate robust overfitting, while KD-AT-Auto + KD-Std + SWA can maintain an overfitting gap close to 0 and achieve better robustness as well.

Setting	s_0	τ	Robust Acc. (%) Best	Robust Acc. (%) Last	Robust Acc. (%) Diff.	Standard Acc. (%) Best	Standard Acc. (%) Last	Standard Acc. (%) Diff.
KD-AT + KD-Std + SWA	80	0.999	49.00	48.04	0.96	84.04	**86.11**	-2.07
KD-AT-Auto + KD-Std + SWA	80	0.999	**49.35**	**49.25**	**0.1**	**85.38**	85.91	**-0.37**
KD-AT + KD-Std + SWA	0	0.999	49.01	48.01	1.0	83.78	**86.20**	-2.42
KD-AT-Auto + KD-Std + SWA	0	0.999	**49.32**	**49.25**	**0.07**	**84.78**	85.48	**-0.7**

Table 6: Performance of our method combined with SWA with different hyper-parameters

D STUDY ON A SYNTHETIC DATASET WITH KNOWN TRUE LABEL DISTRIBUTION

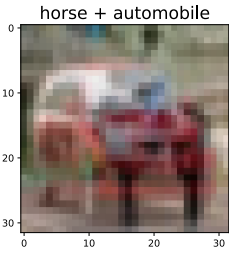


Figure 12: Example mixup augmentation.

Synthetic Dataset. Since the true label distribution is typically unknown for adversarial examples in real-world datasets, we simulate the mechanism of implicit label noise in adversarial training from a feature learning perspective. Specifically, we adapt *mixup* (Zhang et al., 2018) for data augmentation on CIFAR-10. For every example x in the training set, we randomly select another example x' in a different class and linearly interpolate them by a ratio ρ, namely $x := \rho x + (1-\rho)x'$, which essentially perturbs x with features from other classes. Therefore, the true label distribution is arguably $y \sim \rho \cdot \mathbb{1}(y) + (1-\rho) \cdot \mathbb{1}(y')$. Unlike mixup, we intentionally set the assigned label as $\hat{y} \sim \mathbb{1}(y)$, thus deliberately create a mismatch between the true label distribution and the assigned label distribution. We refer this strategy as *mixup augmentation* and only perform it once before the training. In this way, the true label distribution of every example in the synthetic dataset is fixed.

Concentration of optimal temperature and interpolation ratio of individual examples. In Section 4.1 we have shown that in terms of individual examples, the rectified model probability can provably reduce the distribution mismatch between the assigned label distribution and true label distribution of the adversarial example. However, since the true label distribution is unknown in realistic scenarios, it is not possible to directly follow Theorems 4.1 and 4.2 and calculate the optimal set of hyper-parameters for each example in the training set. The best we can do is to employ a

[4]SWA can be implemented using an exponential moving average θ' of the model parameters θ with a decay rate τ, namely $\theta' \leftarrow \tau \cdot \theta' + (1-\tau) \cdot \theta$ at each training step (Rebuffi et al., 2021).

validation set and determine a universal set of hyper-parameters based on the NLL loss, which expects all training examples to share similar optimal temperatures and interpolation ratios. Here, based on the synthetic dataset where a true label distribution is known, we empirically verify this assumption is reasonable.

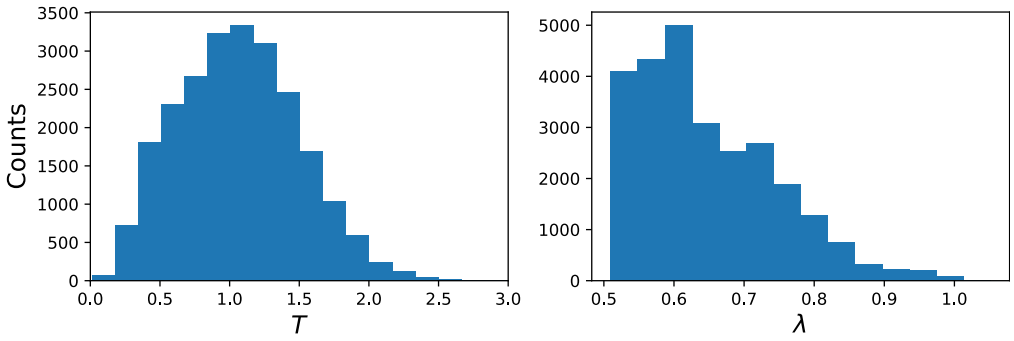


Figure 13: The histograms of optimal temperature (left) and interpolation ratio (right) of individual examples.

In Figure 13 left, we solve the optimal temperature for each correctly classified training example based on Theorem 4.1 with the interpolation ratio fixed as 1.0. One can find that the individual optimal temperatures mostly concentrate between 0.5 and 1.5. In Figure 13 right, we solve the optimal interpolation ratio for each incorrectly classified training example based on Theorem 4.2 with the temperature fixed as 1.0 . One can find that the individual optimal interpolation ratio mostly concentrate between 0.5 and 0.7.

E TOOLKIT

E.1 ESTIMATION OF THE DATA QUALITY

We use the predicative probabilities of classifiers trained on CIFAR-10 to score its training data. Similar strategy is employed in previous works to select high-quality unlabeled data to improve adversarial robustness (Uesato et al., 2019; Carmon et al., 2019; Gowal et al., 2020). Slightly deviating from these works focusing on out-of-distribution data, we use adversarially trained instead of regularly trained models to measure the quality of in-distribution data, since under standard training almost all training examples will be overfitted and gain overwhelmingly high confidence. Specifically, we adversarially train a pre-activation ResNet-18 with PGD and select the model at the best checkpoint in terms of the robustness. The quality of an example is estimated by the model probability corresponding to the true label without adversarial perturbation and random data augmentation (flipping and clipping). We repeat this process 10 times with random initialization to obtain a relatively accurate estimation.

E.2 DETERMINE THE OPTIMAL HYPER-PARAMETERS

We employ a model overfitted on the training set to generate approximate traditional adversarial label of the adversarial example in the validation set. Such overfitted model is typically the model at the final checkpoint when conducting regular adversarial training for sufficient epochs. Mathematically, our final method to determine the optimal temperature and interpolation ratio in rectified model probability can be described as

$$T, \lambda = \arg\min_{T,\lambda} \mathbb{E}_{(x_\delta, y_\delta)\sim\mathcal{D}_{\text{val}}} \ell\left(\lambda \cdot f(x_\delta; \theta, T) + (1-\lambda) \cdot f(x_\delta; \theta^s, T), y_\delta\right), \tag{14}$$

where $f(x_\delta; \theta^s, T)$ denotes the predictive probability of a surrogate model scaled by temperature on x_δ.

F EXPERIMENTAL DETAILS

F.1 SETTINGS FOR MAIN EXPERIMENT RESULTS

Dataset. We include experiment results on CIFAR-10, CIFAR-100, Tiny-ImageNet and SVHN.

Training setting. We employ SGD as the optimizer. The batch size is fixed to 128. The momentum and weight decay are set to 0.9 and 0.0005 respectively. Other settings are listed as follows.

- CIFAR-10/CIFAR-100: we conduct the adversarial training for 160 epochs, with the learning rate starting at 0.1 and reduced by a factor of 10 at the 80 and 120 epochs.
- Tiny-ImageNet: we conduct the adversarial training for 80 epochs, with the learning rate starting at 0.1 and reduced by a factor of 10 at the 40 and 60 epochs.
- SVHN: we conduct the adversarial training for 80 epochs, with the learning rate starting at 0.01 (as suggested by (Chen et al., 2021)) and reduced by a factor of 10 at the 40 and 60 epochs.

Adversary setting. We conduct adversarial training with ℓ_∞ norm-bounded perturbations. We employ adversarial training methods including PGD-AT, TRADES and FGSM. We set the perturbation radius to be $8/255$. For PGD-AT and TRADES, the step size is $2/255$ and the number of attack iterations is 10.

Robustness evaluation. We consider the robustness against ℓ_∞ norm-bounded adversarial attack with perturbation radius $8/255$. We employ AutoAttack for reliable evaluation. We also include the evaluation results again PGD-1000, Square Attack and RayS.

Neural architectures. We include experiments results on pre-activation ResNet-18, WRN-28-5, WRN-34-10 and VGG-19.

Hardware. We conduct experiments on NVIDIA Quadro RTX A6000.

F.2 SETTINGS FOR ANALYZING DOUBLE DESCENT IN ADVERSARIAL TRAINING

Dataset. We conduct experiments on the CIFAR-10 dataset, without additional data.

Training setting. We conduct the adversarial training for 1000 epochs unless otherwise noted. By default we use the Adam optimizer with the learning rate fixed as 0.0001, since it requires minimal hyper-parameter tuning. For SGD the learning rate starts at 0.1, and will not be changed unless otherwise noted. The batch size will be fixed to 128, and the momentum will be set as 0.9 wherever necessary. No regularization such as weight decay is used. These settings are mostly aligned with the empirical analyse of double descent under standard training (Nakkiran et al., 2020).

Sample size. To reduce the computation load demanded by exponential training epochs In individual cases, we reduce the size of the training set by randomly sampled a subset of size 5000 from the original training set without replacement. This will linearly shift the double descent curve but will not significant distort its shape as shown in Appendix B.1. We adopt this setting for extensive experiments such as the analyses of the dependence of double descent on perturbation radius, data quality and the number of attack iterations. Note that in the experiment associated with data quality, we randomly sampled the training subset from those examples with quality lower than a threshold. The sampled subset is restricted to class-balanced.

Adversary setting. We conduct adversarial training with ℓ_∞ norm-bounded perturbations. We employ standard PGD training with the perturbation radius set to $8/255$ unless otherwise noted. The number of attack iterations is fixed as 10, and the perturbation step size is fixed as $2/255$.

Robustness evaluation. We consider the robustness against ℓ_∞ norm-bounded adversarial attack with perturbation radius $8/255$. We use PGD attack with 10 attack iterations and step size set to $2/255$.

Neural architecture. By default we experiment on Wide ResNet (Zagoruyko & Komodakis, 2016) with depth 28 and widening factor 5 (WRN-28-5) to speed up training.

Hardware. We conduct experiments on NVIDIA Quadro RTX A6000.

F.3 SETTINGS FOR ADVERSARIAL AUGMENTATION

We generate adversarial examples with PGD attack on the model obtained at the best checkpoint through a pratical adversarial training (see Appendix F.1 for details). The number of attack iterations is fixed as 10 and the step size fixed as $2/255$. The adversarial examples of all training examples along with their labels are then grouped into a new training set, where we conduct standard training for 1000 epochs. Other settings are same as those listed in Appendix F.2 except no adversary is employed.