A Proofs

A.1 Preliminaries

Proof in Assumption 3.1. Here we prove that if there no label error in the clean dataset, then $P(\tilde{Y}|x) = P(Y|x)$.

Proof. First, we note that

$$P(\tilde{Y} = j'|x) = \sum_j P(\tilde{Y} = j'|Y = j, x)P(Y = j|x).$$

Since $P(E = 1|Y = j, x) = 0$ we have,

$$\begin{aligned} P(\tilde{Y} = j'|Y = j, x) &= \sum_e P(\tilde{Y} = j'|E = e, Y = j, x)P(E = e|Y = j, x) \\ &= P(\tilde{Y} = j'|E = 0, Y = j, x) \\ &= \begin{cases} 1, & \text{if } j' = j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned}$$

Therefore we have for all j',

$$P(\tilde{Y} = j'|x) = P(Y = j'|x).$$

□

Total Variation distance for discrete probability distributions. 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

$$\begin{aligned} \|P(Y) - P(Y')\|_{\text{TV}} &= \sup_{J \in \mathcal{A}} |P(Y \in J) - P(Y' \in J)| \\ &= \sup_{J \in \mathcal{A}} \left| \sum_{j \in J} P(Y = j) - \sum_{j \in J} P(Y' = j) \right| \\ &= \frac{1}{2} \sum_j |P(Y = j) - P(Y' = j|)| \end{aligned}$$

A.2 Proofs in Section 4.1

Proof of Lemma 4.1.

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

$$\delta = -\varepsilon \frac{\nabla f(x)_y}{\|\nabla f(x)_y\|}, \tag{11}$$

First, we bound the distribution mismatch by gradient norm.

$$\begin{aligned}
\|P(Y|x) - P(Y'|x')\|_{\text{TV}} &= \frac{1}{2}\sum_j |P(Y=j|x) - P(Y'=j|x')| \quad \boxed{\text{TV distance}} \\
&\geq \frac{1}{2}|P(Y=y|x) - P(Y'=y|x')| \\
&= \frac{1}{2}|f(x)_y - f(x')_y| \\
&= \frac{1}{2}\left[-\nabla f(x)_y \cdot \delta - \frac{1}{2}\delta^T \nabla^2 f(z)_y \delta\right] \\
&\geq \frac{1}{2}\left[-\nabla f(x)_y \cdot \delta - \frac{\sigma_M}{2}\|\delta\|_2^2\right] \quad \boxed{\text{Bounded Hessian}} \\
&\geq \frac{1}{2}\left[\varepsilon\frac{\|\nabla f(x)_y\|_2^2}{\|\nabla f(x)_y\|} - \frac{\sigma_M}{2}\varepsilon^2\frac{\|\nabla f(x)_y\|_2^2}{\|\nabla f(x)_y\|^2}\right].
\end{aligned}$$

Now if $\|\cdot\| = \|\cdot\|_2$, we have

$$\|P(Y|x) - P(Y'|x')\|_{\text{TV}} \geq \frac{1}{2}\left[\varepsilon\|\nabla f(x)_y\|_2 - \frac{\sigma_M}{2}\varepsilon^2\right]. \tag{12}$$

If $\|\cdot\| = \|\cdot\|_\infty$, we can utilize the fact that $\|\cdot\|_\infty \leq \|\cdot\|_2 \leq \sqrt{d}\|\cdot\|_\infty$, thus

$$\|P(Y|x) - P(Y'|x')\|_{\text{TV}} \geq \frac{1}{2}\left[\varepsilon\|\nabla f(x)_y\|_\infty - \frac{\sigma_M}{2}\varepsilon^2\sqrt{d}\right]. \tag{13}$$

Second, we bound the gradient norm by the L-local Lipschitzness assumption. Let x^* be a closest input that achieves the local maximum on the predicted probability at y, namely $x^* = \arg\min_{z\in X, f(z)_y=1}\|x-z\|$. Because x^* is the local maximum and f is continuously differentiable, $\nabla f(x^*)_y = 0$, thus

$$\nabla f(x)_y = \nabla f(x^*)_y + \nabla^2 f(z)_y(x-x^*) = \nabla^2 f(z)_y(x-x^*).$$

Therefore we have

$$\begin{aligned}
\|\nabla f(x)_y\| &= \|\nabla^2 f(z)_y(x-x^*)\| \\
&\geq \sigma_m\|x-x^*\| \\
&\geq \sigma_m\frac{|f(x^*)_y - f(x)_y|}{L} \\
&= \frac{\sigma_m}{L}(1 - f(x)_y).
\end{aligned}$$

Plug this into Equation (12) or Equation (13) we then obtain the desired result. □

Proof of Lemma 4.2.

Proof. First, we show that the expectation of the label error is lower bounded by the mismatch between the true label distribution and the assigned label distribution.

$$\begin{aligned}
\|P(\tilde{Y}|x) - P(Y|x)\|_{TV} &= \frac{1}{2}\sum_j |P(\tilde{Y}=j|x) - P(Y=j|x)| \\
&= \frac{1}{2}\sum_j |P(\tilde{Y}=j, Y=j|x) + P(\tilde{Y}=j, Y\neq j|x) \\
&\quad - P(Y=j, \tilde{Y}=j|x) - P(Y=j, \tilde{Y}\neq j|x)| \\
&= \frac{1}{2}\sum_j |P(\tilde{Y}=j, Y\neq j|x) - P(Y=j, \tilde{Y}\neq j|x)| \\
&\leq \frac{1}{2}\sum_j P(\tilde{Y}=j, Y\neq j|x) + P(Y=j, \tilde{Y}\neq j|x) \\
&= P(Y'\neq Y|x) \\
&= P(E=1|x)
\end{aligned} \tag{14}$$

Second, given a sampled training set $\mathcal{D} = \{(x_i, \tilde{y}_i)\}_{i\in[N]}$, the empirical measure of label error E should converge to its expectation almost surely, namely

$$\lim_{N\to\infty} p_e(\mathcal{D}) = \lim_{N\to\infty} \frac{1}{N} \sum_{i\in[N]} e_i = \mathbb{E}[E] = P(E = 1).$$

Using standard concentration inequality such as Hoeffding's inequality we have, with probability $1 - \delta$,

$$|p_e(\mathcal{D}) - P(E = 1)| \leq \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}.$$

This implies

$$p_e(\mathcal{D}) \geq P(E = 1) - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}.$$

Since $P(E = 1) = \mathbb{E}_x P(E = 1|x)$, we have, with probability $1 - \delta$,

$$p_e(\mathcal{D}) \geq \mathbb{E}_x \|P(\tilde{Y}|x) - P(Y|x)\|_{\text{TV}} - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}.$$

which means $p_e(\mathcal{D}) > 0$ as long as N is large.

□

Proof of Theorem 4.3.

Proof. First, by the fact that $P(\tilde{Y}'|x') = P(\tilde{Y}|x)$ and $P(\tilde{Y}|x) = P(Y|x)$ we have $P(\tilde{Y}'|x') = P(Y|x)$.

Therefore, apply Lemma 4.2 to an adversarially augmented training set we have with probability $1 - \delta$,

$$\begin{aligned} p_e(\mathcal{D}') &\geq \mathbb{E}_x \|P(\tilde{Y}'|x') - P(Y'|x')\|_{\text{TV}} - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}} \\ &\geq \mathbb{E}_x \|P(Y|x) - P(Y'|x')\|_{\text{TV}} - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}. \end{aligned}$$

Further, apply Lemma 4.1 and the definition of data quality, we have with probability $1 - \delta$,

$$\begin{aligned} p_e(\mathcal{D}') &\geq \frac{\varepsilon}{2}(1 - \mathbb{E}_x f(x)_y)\frac{\sigma_m}{L} - \frac{\varepsilon^2}{4}\sigma_M - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}} \\ &\geq \frac{\varepsilon}{2}(1 - q(\mathcal{D}))\frac{\sigma_m}{L} - \frac{\varepsilon^2}{4}\sigma_M - \sqrt{\frac{1}{2N} \log \frac{2}{\delta}}. \end{aligned}$$

□

A.3 Proofs in Section 4.2

Proof of Lemma 4.4.

Proof. Let $\mathcal{D} = (x, y)$ be the adversarially augmented training set. Let $\mathcal{S} = \{x : (x, y) \in \mathcal{D}\}$ be the collection of all training inputs. First we note that the set of all training inputs can be grouped into several subsets such that the inputs in each subset possess similar true label distribution. More formally, Let $\mathcal{C} = \{\bar{x}_j\}_{j=1}^{N_{\rho\varepsilon}}$ be an $\rho\varepsilon$-external covering of $\mathcal{S}$ with minimum cardinality, namely $\mathcal{S} \subseteq \bigcup_{x\in\mathcal{C}} \{x' \mid \|x' - x\| \leq \rho\varepsilon\}$, where we refer $\bar{x}_j$ as the covering center, and $N_{\rho\varepsilon}$ is the covering number.

Let $\{\mathcal{S}_j\}_{j=1}^{N_{\rho\varepsilon}}$ be any disjoint partition of $\mathcal{S}$ such that $\mathcal{S}_j \subseteq \{x' | \|x' - \bar{x}_j\| \leq \rho\varepsilon\}$. We show that $\mathcal{S}_j$ attains a property that the true label distribution of any input x in this subset will not be too

far from the sample mean of one-hot labels $\bar{\mathbf{y}}_j = |\mathcal{S}_j|^{-1} \sum_{x \in \mathcal{S}_j} \mathbf{1}_y$ in this subset. Specifically, let $p(x) = P(Y|x)$. We have with probability $1 - \delta$,

$$\|p(x) - \bar{\mathbf{y}}_j\|_1 \leq \sqrt{\frac{2K}{|\mathcal{S}_j|} \log \frac{2}{\delta}} + 2L\rho\varepsilon. \tag{15}$$

To prove this property we first present two lemmas.

Lemma A.1 (Lipschitz constraint of the true label distribution)**.** *Let $\mathcal{S}_j$ be a subset constructed above and $\bar{\mathbf{y}}_j = |\mathcal{S}_j|^{-1} \sum_{x \in \mathcal{S}_j} \mathbf{1}_y$. Then for any $x \in \mathcal{S}_j$ we have,*

$$\|p(x) - \mathbb{E}[\bar{\mathbf{y}}_j]\|_1 \leq 2L\rho\varepsilon. \tag{16}$$

Proof. First, since $x \in \mathcal{S}_j$, we have $\|x - \bar{x}_j\|_1 \leq \rho\varepsilon$, which implies $\|p(x) - p(\bar{x}_j)\|_1 \leq L\rho\varepsilon$ by the locally Lipschitz continuity of p. Then for any $x, x' \in \mathcal{S}_j$, we will have $\|p(x) - p(x')\| \leq 2L\rho\varepsilon$ by the triangle inequality. Let $N_{\mathcal{S}} = |\mathcal{S}_j|$. Therefore,

$$\left\| p(x) - \frac{1}{N_{\mathcal{S}}} \sum_{x \in \mathcal{S}_j} p(x) \right\|_1 \leq 2 \frac{N_{\mathcal{S}} - 1}{N_{\mathcal{S}}} L\rho\varepsilon \leq 2L\rho\varepsilon. \tag{17}$$

Further, the linearity of the expectation implies

$$\mathbb{E}[\bar{\mathbf{y}}] = {N_{\mathcal{S}}}^{-1} \sum_{x \in \mathcal{S}_j} \mathbb{E}[\mathbf{1}_{y(x)}] = {N_{\mathcal{S}}}^{-1} \sum_{x \in \mathcal{S}_j} p(x). \tag{18}$$

Therefore $\|p(x) - \mathbb{E}[\bar{\mathbf{y}}_j]\| \leq 2L\rho\varepsilon$. □

Lemma A.2 (Concentration inequality of the sample mean)**.** *Let $\mathcal{S}$ be a set of x with cardinality N. Let $\bar{\mathbf{y}} = N^{-1} \sum_{x \in \mathcal{S}} \mathbf{1}_y$ be the sample mean. Then for any p-norm $\|\cdot\|$ and any $\varepsilon > 0$, we have with probability $1 - \delta$,*

$$\|\bar{\mathbf{y}} - \mathbb{E}[\bar{\mathbf{y}}]\|_1 \leq \sqrt{\frac{2K}{N} \log \frac{2}{\delta}} \tag{19}$$

Proof. Note that $\bar{\mathbf{y}}$ obeys a multinomial distribution, *i.e.* $\bar{\mathbf{y}} \sim N^{-1} multinomial(N, \mathbb{E}[\bar{\mathbf{y}}])$. This lemma is thus the classic result on the concentration properties of multinomial distribution based on ℓ_1 norm Weissman et al. (2003); Qian et al. (2020). □

One can see that Lemma A.1 bounds the difference between true label distribution of individual inputs and the mean true label distribution, while Lemma A.2 bounds the difference between the sample mean and the mean true label distribution. Therefore the difference between the true label distribution and the sample mean is also bounded, since by the triangle inequality we have with probability $1 - \delta$,

$$\begin{aligned} \|p(x) - \bar{\mathbf{y}}\| &\leq \|p(x) - \mathbb{E}[\bar{\mathbf{y}}]\| + \|\bar{\mathbf{y}} - \mathbb{E}[\bar{\mathbf{y}}]\| \\ &\leq \sqrt{\frac{2K}{N} \log \frac{2}{\delta}} + 2L\rho\varepsilon. \end{aligned} \tag{20}$$

We now show that given the locally Lipschitz constraint established in each disjoint partition we constructed above, the prediction given by the empirical risk minimizer will be close to the sample mean. As an example, we focus on the negative log-likelihood loss, namely $\ell(f_\theta(x), y) = -\mathbf{1}_y \cdot \log f_\theta(x)$. Other loss functions that are subject to the proper scoring rule can be investigated in a similar manner. First, we regroup the sum in the empirical risk based on the partition constructed above, namely

$$\hat{R}(f_\theta, \mathcal{S}) = \frac{1}{N_{\rho\varepsilon}} \sum_{j=1}^{N_{\rho\varepsilon}} \hat{R}(f_\theta, \mathcal{S}_j), \tag{21}$$

where $\hat{R}(f_\theta, \mathcal{S}_j) = -|\mathcal{S}_j|^{-1} \sum_{i=1}^{|\mathcal{S}_j|} \mathbf{1}_{y_i} \cdot \log f_\theta(x_i)$ is the empirical risk in each partition. Since we are only concerned with the existence of a desired minimizer of the empirical risk, we can view f_θ as

able to achieve any labeling of the training inputs that suffices the local Lipschitz constraint. Thus the empirical risk minimization is equivalent to the minimization of the empirical risk in each partition. The problem can thus be defined as, for each $j = 1, \cdots, N_{\rho\varepsilon}$,

$$\begin{aligned} &\min_{f_\theta} \hat{R}(f_\theta, \mathcal{S}_j) \\ &s.t.\ \ \|f_\theta(x) - f_\theta(\bar{x}_j)\|_1 \leq L\rho\varepsilon,\ \forall\, x \in \mathcal{S}_j, \end{aligned} \tag{22}$$

where the constraint is imposed by the locally-Lipschitz continuity of f_θ. By the following lemma, we show that the minimizer of such problem is achieved only if $f_\theta(\bar{x}_j)$ is close to the sample mean.

Lemma A.3. *Let $\bar{\mathbf{y}} = |\mathcal{S}_j|^{-1} \sum_{x \in \mathcal{S}_j} \mathbf{1}_y$. The minimum of the problem (22) is achieved only if $f_\theta(\bar{x}_j) = \bar{\mathbf{y}}_\mathbf{j}(1 + KL_\theta\rho\varepsilon) - L_\theta\rho\varepsilon$.*

Proof. We note that since the loss function we choose is strongly convex, to minimize the empirical risk, the prediction of any input x must be as close to the one-hot labeling as possible. Therefore the problem (22) can be formulated into a vector minimization where we can employ Karush–Kuhn–Tucker (KKT) theorem to find the necessary conditions of the minimizer.

Let $\mathbf{p}_i := f_\theta(x_i)$ and $\tilde{\varepsilon} = L\rho\varepsilon$ for simplicity. We rephrase the problem (22) as

$$\begin{aligned} &\min_{\{\mathbf{p}_i\}_{i=1}^N} -\frac{1}{N}\sum_i \mathbf{1}_{y_i} \cdot \log \mathbf{p}_i \\ &s.t.\ \ \|\mathbf{p}_i - \mathbf{p}\|_1 \leq \tilde{\varepsilon},\ \sum_k \mathbf{p}_i^k = 1,\ \sum_k \mathbf{p}^k = 1, \mathbf{p}_i^k \geq 0, \mathbf{p}^k \geq 0. \end{aligned} \tag{23}$$

Case I. We first discuss the case when $\mathbf{p}^k + \tilde{\varepsilon} < 1$ for all k. First, we observe that for any $\mathbf{p}$, the minimum of the above problem is achieved only if $\mathbf{p}_i^{y_i} = \mathbf{p}^{y_i} + \tilde{\varepsilon}$. Because by contradiction, if $\mathbf{p}_i^{y_i} < \mathbf{p}^{y_i} + \tilde{\varepsilon}$, we will have $-\log \mathbf{p}_i^{y_i} > -\log(\mathbf{p}^{y_i} + \tilde{\varepsilon})$, and $\mathbf{p}^{y_i} + \tilde{\varepsilon}$ belongs to the feasible set, which means $\mathbf{p}_i^{y_i}$ does not attain the minimum.

The above problem can then be rephrased as

$$\min_{\mathbf{p}} -\frac{1}{N}\sum_i \log(\mathbf{p}^{y_i} + \tilde{\varepsilon}),\ \ s.t.\ \ \sum_k \mathbf{p}^k = 1, \mathbf{p}^k \geq 0, \tag{24}$$

where we have neglected the condition associated with $\mathbf{p}_i^{k \neq y_i}$, since they do not contribute to the objective, they can be chosen arbitrarily as long as the constraints are sufficed, and clearly the constraints are underdetermined.

Let $N_k = \sum_i 1(y_i = k)$, we have $\sum_i \log(\mathbf{p}^{y_i} + \tilde{\varepsilon}) = \sum_k N_k \log(\mathbf{p}^k + \tilde{\varepsilon})$. Therefore the above problem is equivalent to

$$\min_{\mathbf{p}} -\sum_k \bar{\mathbf{y}}^k \log(\mathbf{p}^k + \tilde{\varepsilon}),\ \ s.t.\ \ \sum_k \mathbf{p}^k = 1, \mathbf{p}^k \geq 0, \tag{25}$$

where $\bar{\mathbf{y}} \equiv [N_1/N, \cdots, N_k/N]^T$ is equal to the sample mean $N^{-1}\sum_i \mathbf{1}_{y_i}$.

To solve the strongly convex minimization problem (25) it is easy to employ KKT conditions to show that

$$\mathbf{p} = \bar{\mathbf{y}}(1 + K\tilde{\varepsilon}) - \tilde{\varepsilon}.$$

Case II. We now discuss the case when $\hat{\mathbf{p}}$ is the minimizer of (23) and there exists k' such that $\hat{\mathbf{p}}^{k'} + \tilde{\varepsilon} \geq 1$. And $\hat{\mathbf{p}} \neq p$, where $p = \bar{\mathbf{y}}(1 + K\tilde{\varepsilon}) - \tilde{\varepsilon}$ is the form of the minimizer in the previous case.

Considering a non-trivial case $\mathbf{p}^{*k'} < 1 - \tilde{\varepsilon}$. Otherwise the true label distribution is already close to the one-hot labeling, which is the minimizer of the empirical risk. Therefore by $\sum_{k \neq k'} p^k > \tilde{\varepsilon}$ we have the condition

$$\sum_{k \neq k'} \bar{\mathbf{y}}^k > \frac{K\tilde{\varepsilon}}{1 + K\tilde{\varepsilon}} \tag{26}$$

Now considering the minimization objective $R(p) = -N^{-1}\sum_i \mathbf{1}_{y_i} \cdot \log \mathbf{p}_i$. For all i with $y_i = k'$, we must have $\mathbf{p}_i^{y_i} = 1$, otherwise the optimal cannot be attained by contradiction. Then the minimization problem can be rephrased as

$$\min \sum_{k \neq k'} \bar{\mathbf{y}}^k \log(\hat{\mathbf{p}} + \tilde{\varepsilon}),\ s.t. \sum_{k' \neq k} \hat{\mathbf{p}}^k \geq \tilde{\varepsilon}, \hat{\mathbf{p}}^k \geq 0, \tag{27}$$

where the first constraint is imposed by $\hat{\mathbf{p}}^{k'} \geq 1 - \tilde{\varepsilon}$.

Employ KKT conditions similarly we can have $\hat{\mathbf{p}}^k = \bar{\mathbf{y}}^k/\lambda - \tilde{\varepsilon}$ where λ is a constant. By checking the constraint we can derive $\lambda \geq \sum_k \bar{\mathbf{y}}^k/(K\tilde{\varepsilon})$.

However, the minimization objective

$$\min_{\lambda} - \sum_{k \neq k'} \bar{\mathbf{y}}^k \log \frac{\bar{\mathbf{y}}^k}{\lambda},$$

requires λ to be minimized. Therefore $\lambda = \sum_{k \neq k'} \bar{\mathbf{y}}^k/(K\tilde{\varepsilon})$, which implies

$$\hat{\mathbf{p}}^k = K\tilde{\varepsilon}\frac{\bar{\mathbf{y}}^k}{\sum_{k \neq k'} \bar{\mathbf{y}}^k} - \tilde{\varepsilon}. \tag{28}$$

Now since $\hat{\mathbf{p}} = \arg\min_p R(p)$ and $\hat{\mathbf{p}} \neq p$, we must have $R(\hat{\mathbf{p}}) < R(p)$. This means

$$-\sum_{k \neq k'} \bar{\mathbf{y}}^k \log \frac{K\tilde{\varepsilon}\bar{\mathbf{y}}^k}{\sum_{k \neq k'} \bar{\mathbf{y}}^k} < -\sum_{k \neq k'} \bar{\mathbf{y}}^k \log[\bar{\mathbf{y}}^k(1 + K\tilde{\varepsilon})], \tag{29}$$

which is reduced to

$$\sum_{k \neq k'} \bar{\mathbf{y}}^k < \frac{K\tilde{\varepsilon}}{1 + K\tilde{\varepsilon}} \tag{30}$$

But this is contradict to our assumption. □

We are now be able to bound the difference between the predictions of the training inputs produced by the empirical risk minimizer and the sample mean in each $\mathcal{S}_j$. To see that we have for each $x \in \mathcal{S}_j$.

$$\begin{aligned} \|f_\theta(x) - \bar{\mathbf{y}}_j\|_1 &\leq \|f_\theta(x) - f_\theta(\bar{x}_j)\|_1 + \|f_\theta(\bar{x}_j) - \bar{\mathbf{y}}_j\|_1 \\ &\leq L\rho\varepsilon(1 + K\|\bar{\mathbf{y}}_j - K^{-1}\mathbf{1}\|_1) \\ &\leq L\rho\varepsilon(1 + K\|\mathbf{1}_{(\cdot)} - K^{-1}\mathbf{1}\|_1). \\ &= L\rho\varepsilon\left(3 - \frac{2}{K}\right) \end{aligned} \tag{31}$$

By Equation (15) we then have for any $x \in \mathcal{S}_j$, with probability $1 - \delta$,

$$\|f_\theta(x) - p(x)\|_1 \leq \sqrt{\frac{2K}{|\mathcal{S}_j|}\log\frac{2}{\delta}} + L_\theta\rho\varepsilon\left(3 - \frac{2}{K}\right) + 2L\rho\varepsilon, \tag{32}$$

which means the difference between the predictions and the true label distribution is also bounded.

Step III: Show the disjoint partition is non-trivial. In (32), we have managed to bound the difference between the predictions yielded by an empirical risk minimizer and the true label distribution based on the cardinality of the subset $|\mathcal{S}_j|$, namely the number of inputs in j-partition. However $|\mathcal{S}_j|$ is critical to the bound here as if $|\mathcal{S}_j| = 1$, then (32) becomes a trivial bound. Here we show $|\mathcal{S}_j|$ is non-negligible based on simple combinatorics.

Lemma A.4. *Let $\{\mathcal{S}_j\}_{j=1}^{N_{\rho\varepsilon}}$ be a disjoint partition of the entire training set $\mathcal{S}$. Denote $\mathcal{S}(x)$ as the partition that includes x. Let $N(x) = |\mathcal{S}(x)|$ and $N = |\mathcal{S}|$. Then for any $\kappa \geq 1$,*

$$\left|\left\{x \mid N(x) \geq \frac{N}{\kappa N_{\rho\varepsilon}}\right\}\right| \geq \left(1 - \frac{1}{\kappa} + \frac{1}{\kappa N_{\rho\varepsilon}}\right) N. \tag{33}$$

Proof. We note that the problem is to show the minimum number of x such that $N(x) \geq N/(\kappa N_{\rho\varepsilon})$. This is equivalent to find the maximum number of x such that $N(x) \leq N/(\kappa N_{\rho\varepsilon})$. Since we only have $N_{\rho\varepsilon}$ subsets, the maximum can be attained only if for $N_{\rho\varepsilon}-1$ subsets $\mathcal{S}$, $|\mathcal{S}| = N/(\kappa N_{\rho\varepsilon})$. Otherwise, if for any one of these subsets $|\mathcal{S}| < N/(\kappa N_{\rho\varepsilon})$, then it is always feasible to let $|\mathcal{S}| = N/(\kappa N_{\rho\varepsilon})$ and the maximum increases. Similarly, if the number of such subsets is less than $N_{\rho\varepsilon} - 1$, then it is always feasible to let another subset subject to $|\mathcal{S}| = N/(\kappa N_{\rho\varepsilon})$ and the maximum increases. We can then conclude that at most $N(N_{\rho\varepsilon} - 1)/(\kappa N_{\rho\varepsilon})$ inputs can have the property $N(x) \leq N/(\kappa N_{\rho\varepsilon})$.

□

The above lemma basically implies when partitioning N inputs into $N_{\rho\varepsilon}$ subsets, a large fraction of the inputs will be assigned to a subset with cardinality at least $N/(\kappa N_{\rho\varepsilon})$. Here $N_{\rho\varepsilon}$ is the covering number and is bounded above based on the property of the covering in the Euclidean space. Apply Lemma A.4 to (32), and use the fact that $\|\cdot\|_{\text{TV}} = \|\cdot\|_1/2$ for category distributions, we then arrive at Lemma 4.4.

□

Proof of Theorem 4.5.

Proof. First, we show that adversarial perturbation generated by a realistic classifier can change its predictive distribution. Considering adversarial perturbation based on FGSM and cross-entropy loss, namely $x' = x - \varepsilon\|\nabla\ f_\theta(x)_y\|^{-1}\nabla\ f_\theta(x)_y$, we can obtain a result similar to Lemma 4.1.

Lemma A.5. *Assume $f_\theta(x)_y$ is L_θ-locally Lipschitz around x with bounded Hessian. Let $\sigma_m = \inf_{z\in\mathcal{B}_\varepsilon(x)} \sigma_{\min}(\nabla^2 f_\theta(z)_y) > 0$ and $\sigma_M = \sup_{z\in\mathcal{B}_\varepsilon(x)} \sigma_{\max}(\nabla^2 f_\theta(z)_y) > 0$. Here $\sigma_{\min}$ and $\sigma_{\max}$ denote the minimum and maximum eigenvalues of the Hessian, respectively. We then have*

$$\|f_\theta(x) - f_\theta(x')\|_{TV} \geq \frac{\varepsilon}{2}(1 - f_\theta(x)_y)\frac{\sigma_m}{L_\theta} - \frac{\varepsilon^2}{4}\sigma_M, \tag{34}$$

Second, We prove that the true label distribution will be distorted by the adversarial perturbation generated by a realistic classifier. This is guaranteed if the predictive distribution of a realistic classifier can approximate the true label distribution. Specifically, by utilizing Lemma A.5 and Lemma 4.4, we have with probability $1 - 2\delta$,

$$\begin{aligned}
&\|P(Y|x) - P(Y'|x')\|_{\text{TV}} \\
\geq&\|f_\theta(x) - f_\theta(x')\|_{\text{TV}} - (\|f_\theta(x) - P(Y|x)\|_{\text{TV}} + \|f_\theta(x') - P(Y'|x')\|_{\text{TV}}) \\
\geq&\frac{\varepsilon}{2}(1 - f_\theta(x)_y)\frac{\sigma_m}{L_\theta} - \frac{\varepsilon^2}{4}\sigma_M - \sqrt{\frac{2\kappa N_{\rho\varepsilon} K}{N}\log\frac{2}{\delta}} - \left(\left(\frac{3}{2} - \frac{1}{K}\right)L_\theta + L\right)2\rho\varepsilon \\
=&\varepsilon\left[(1 - f_\theta(x)_y)\frac{\sigma_m}{2L_\theta} - 2\rho\left(\left(\frac{3}{2} - \frac{1}{K}\right)L_\theta + L\right)\right] - \varepsilon^2\frac{\sigma_M}{4} - \sqrt{\frac{2\kappa N_{\rho\varepsilon} K}{N}\log\frac{2}{\delta}}.
\end{aligned} \tag{35}$$

Finally, we show that such distribution mismatch induces label noise in the adversarially augmented training set. Similar to the proof for the true classifier, by the common labeling practice of adversarial examples we have $P(\tilde{Y}'|x') = P(\tilde{Y}|x) = P(Y|x)$. By utilizing Lemma 4.2 [1] we then have with probability $1 - 3\delta$,

$$p_e(\mathcal{D}') \geq \varepsilon\left[(1 - \mathbb{E}_x f_\theta(x)_y)\frac{\sigma_m}{2L_\theta} - 2\rho\left(\left(\frac{3}{2} - \frac{1}{K}\right)L_\theta + L\right)\right] - \varepsilon^2\frac{\sigma_M}{4} - \xi\sqrt{\frac{1}{2N}\log\frac{2}{\delta}}, \tag{36}$$

where $\xi = 1 + \sqrt{4\kappa N_{\rho\varepsilon} K}$.

□

[1] Note this is a result only associated with the training set, thus is not dependent on the specific classifier.

A.4 Proofs in Section 5

Proof of Theorem 5.1.

Proof. Let $j^* = \text{argmax } P(Y' = j|x')$ and thus $P(Y' = j^*|x') \in [1/c, 1]$. Let $g(T) := f(x'; \theta, T)_{j^*}$, which is a continuous function defined on $[0, \infty]$. The condition $j^* = \arg\max_j f(x'; \theta, T)_j$ 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 $g(T^*) = P(Y' = j^*|x')$.

Let $T = T^*$, we have

$$\begin{aligned}
\|f(x'; \theta, T) - P(Y'|x')\|_{TV} &= \frac{1}{2} \sum_j |f(x'; \theta, T)_j - P(Y' = j|x')| \\
&= \frac{1}{2} \sum_{j, j \neq j^*} |f(x'; \theta, T)_j - P(Y' = j|x')| \\
&\leq \frac{1}{2} \left[\sum_{j, j \neq j^*} f(x'; \theta, T)_j + \sum_{j, j \neq j^*} P(Y' = j|x') \right] \\
&= 1 - P(Y' = j^*|x'),
\end{aligned}$$

where the inequality holds by the triangle inequality.

Meanwhile, we have

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

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

$$\|f(x'; \theta, T) - P(Y'|x')\|_{TV} \leq \|P(\tilde{Y}'|x') - P(Y'|x')\|_{TV}.$$

□

Proof of Theorem 5.2.

Lemma A.6. *Let x' be an example incorrectly classified by a classifier f in terms of the true label distribution $P(Y' = j|x')$, namely*

$$\arg\max_j \ f(x'; \theta, T)_j \neq j^*,$$

where $j^* = \arg\max_j \ P(Y' = j|x')$. Assume $P(Y' = j^*|x') \geq 1/2$, then

$$f(x'; \theta, T)_{j^*} \leq P(Y' = j^*|x').$$

Proof. We prove it by contradiction. Assume $f(x'; \theta, T)_{j^*} > P(Y' = j^*|x')$, we have $f(x'; \theta, T)_{j^*} > P(Y' = j^*|x') \geq 1/2$. Therefore,

$$f(x'; \theta, T)_j \leq \sum_{j, j \neq j^*} f(x'; \theta, T)_j = 1 - f(x'; \theta, T)_{j^*} < 1/2, \ \forall j \neq j^*,$$

which means $f(x'; \theta, T)_j < f(x'; \theta, T)_{j^*}, \ \forall j \neq j^*$. This leads to $j^* = \arg\max_j f(x'; \theta, T)_j$, which contradicts our condition. □

Now we prove Theorem 5.2

Proof. First let $P(Y'|x') = P(y|x) \approx \mathbb{1}(y)$. Let $j^* = \arg\max_j P(Y' = j|x')$. By Lemma A.6 we have $f(x';\theta,T)_{j^*} \leq P(Y'^* = j^*|x') \leq 1$. Then there exists $\lambda^* > 0$, such that $\lambda^* \cdot f(x';\theta,T)_{j^*} + (1-\lambda^*) = P(Y' = j^*|x')$ by the intermediate value theorem.

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

$$
\begin{aligned}
&2\left[\|\lambda \cdot f(x';\theta,T) + (1-\lambda)\cdot P(\tilde{Y}'|x') - P(Y'|x')\|_{TV} - \|f(x';\theta,T) - P(Y'|x')\|_{TV}\right] \\
=&2\left[\|\lambda \cdot f(x';\theta,T) + (1-\lambda)\cdot \mathbb{1}(y) - P(Y'|x')\|_{TV} - \|f(x';\theta,T) - P(Y'|x')\|_{TV}\right] \\
=&\sum_j |\lambda \cdot f(x';\theta,T)_j + (1-\lambda)\cdot 1(j=y) - P(Y'=j|x')| - \sum_j |f(x';\theta,T)_j - P(Y'=j|x')| \\
=&\sum_j |\lambda \cdot f(x';\theta,T)_j + (1-\lambda)\cdot 1(j=Y) - P(Y'=j|x')| - \sum_j |f(x';\theta,T)_j - P(Y'=j|x')| \\
=&\sum_{j,j\neq j^*} |\lambda \cdot f(x';\theta,T)_j - P(Y'=j|x')| - \sum_{j,j\neq j^*} |f(x';\theta,T)_j - P(Y'=j|x')| - |f(x';\theta,T)_{j^*} - P(Y'=j^*|x')| \\
\leq&\sum_{j,j\neq j^*} |\lambda \cdot f(x';\theta,T)_j - f(x';\theta,T)_j| - |f(x';\theta,T)_{j^*} - P(Y'=j^*|x')| \\
=&\sum_{j,j\neq j^*} [f(x';\theta,T)_j - \lambda \cdot f(x';\theta,T)_j] - [P(Y'=j^*|x') - f(x';\theta,T)_{j^*}] \\
=&\sum_{j,j\neq j^*} [f(x';\theta,T)_j - \lambda \cdot f(x';\theta,T)_j] - [\lambda \cdot f(x';\theta,T)_{j^*} + (1-\lambda) - f(x';\theta,T)_{j^*}] \\
=&\sum_j f(x';\theta,T)_j - \lambda \sum_j f(x';\theta,T)_j - (1-\lambda) \\
=&0.
\end{aligned}
$$

□

B Limitations

We note that alternative labeling of adversarial examples proposed in this paper is based on the fact that the predictive distribution of a classifier trained with empirical risk minimization can approximate the true label distribution of training examples. However, such approximation may not be accurate especially if the classifier is not carefully regularized during training. Post-training confidence calibration techniques such as temperature scaling and interpolation can only improve the approximation in terms of the entire training set, but cannot improve it in a sample-wise manner. How to learn the true label distribution of adversarial training examples during adversarial training more accurately remains an open problem.

Also, such alternative labeling also requires to train another independent classifier beforehand, which induces additional training cost.

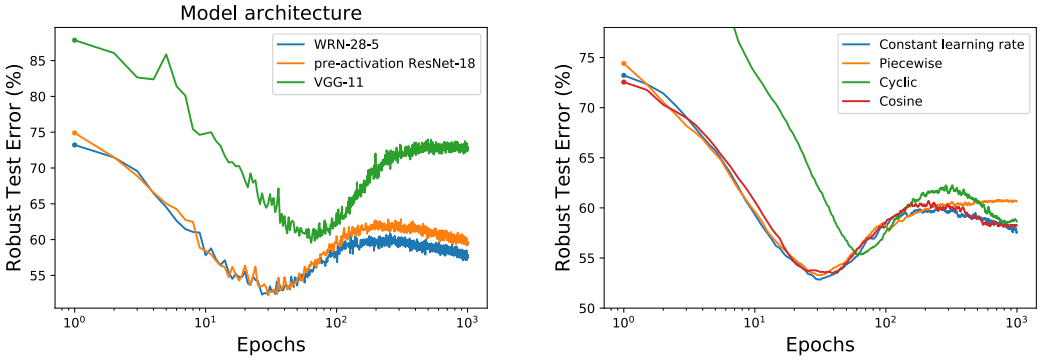
C More empirical analyses

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

In this section, we conduct extensive experiments with different 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 G.2 in detail.

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 such case, the number of channels in each residual block is scaled similarly but rounded, and the number of channels in the first convolutional layer will be reduced accordingly to ensure the width monotonically increasing through the forward propagation.

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



(a) Epoch-wise double descent curves in adversarial training with various model architectures.

(b) Epoch-wise double descent curves in adversarial training with various learning rate schedulers. The curves are smoothed by a moving average with a window of 5 to avoid overlapping.

Figure 8: Effect of model on the epoch-wise double descent curve

these configurations to ensure comparable model capacities[2]. As shown in Figure 8, 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.

D More experiment results

D.1 Training longer

As shown in Figure 9, We show that our method can maintain the robust test accuracy with more training epochs. Here, we follow the settings in Figure 7 except we train for additional epochs up to 400 epochs for each dataset.

D.2 Adversarial training methods, neural architectures and evaluation metrics

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

[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.

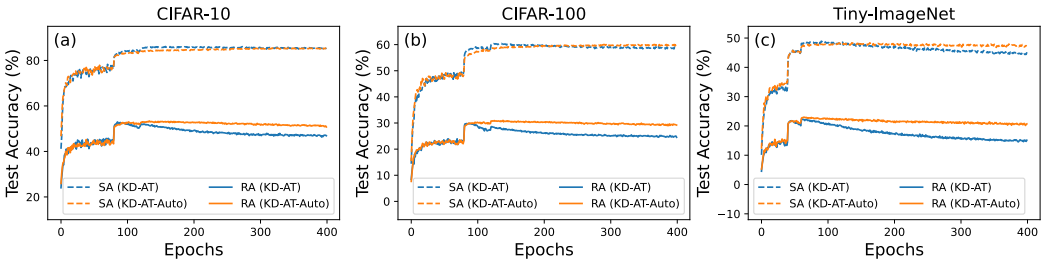


Figure 9: Our method can maintain robust test accuracy for more training epochs.

Table 2: Performance of our method with different neural architectures.

Architecture	Setting	T	λ	Robust Acc. (%) Best	Robust Acc. (%) Last	Robust Acc. (%) Diff.	Standard Acc. (%) Best	Standard Acc. (%) Last	Standard Acc. (%) 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 adversarial training methods.

Method	Setting	T	λ	Robust Acc. (%) Best	Robust Acc. (%) Last	Robust Acc. (%) Diff.	Standard Acc. (%) Best	Standard Acc. (%) Last	Standard Acc. (%) 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 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).

Attacks	Setting	T	λ	Robust Acc. (%) Best	Robust Acc. (%) Last	Robust Acc. (%) 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**

D.3 Combined with additional orthogonal techniques

We note that motivated from our theoretical analyses, our proposed method (KD-AT-Auto) is essentially the baseline knowledge distillation for adversarial training (KD-AT) 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 (KD-Std) 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. It is thus hard to see any further reduction by combining our method. To this end, we introduce an extra dataset SVHN (Netzer et al., 2011). As shown in Table 5, on SVHN, KD-AT + KD-Std + SWA still produces a high overfitting gap (also see Appendix A1.3 in (Chen et al., 2021)), whereas by combining with our algorithm to automatically find the optimal hyper-parameters (KD-AT-Auto + KD-Std + SWA), the overfitting gap can be further reduced to almost 0. This demonstrates the effectiveness and wide applicability of our principle-guided method on mitigating robust overfitting.

Table 5: Performance of our method combined with SWA and an additional standard teacher.

Dataset	Setting	T	λ	Robust Acc. (%)			Standard Acc. (%)		
				Best	Last	Diff.	Best	Last	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

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 G.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 overfitting may strongly depend on its hyper-parameter selection including s_0, i.e., the starting epoch and τ, i.e., the decay rate[3], 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.

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

Setting	s_0	τ	Robust Acc. (%)			Standard Acc. (%)		
			Best	Last	Diff.	Best	Last	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**

E Study on a synthetic dataset with known true label distribution

[3]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).

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.

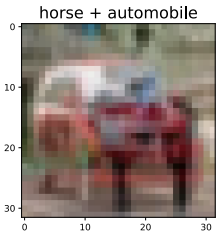


Figure 10: Sample image by mixup augmentation.

Concentration of optimal temperature and interpolation ratio of individual examples. In Section 5.1 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 5.1 and 5.2 and calculate the optimal set of hyper-parameters for each example in the training set. The best we can do is to employ a 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.

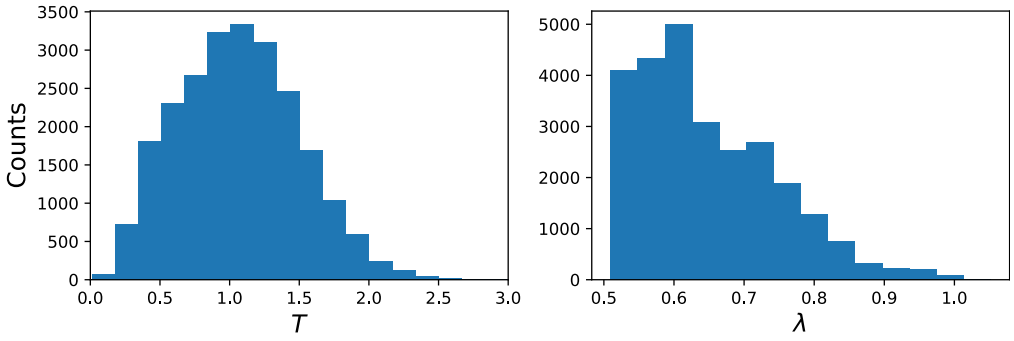


Figure 11: The histograms of optimal temperature (left) and interpolation ratio (right) of individual examples.

In Figure 11 left, we solve the optimal temperature for each correctly classified training example based on Theorem 5.1 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 11 right, we solve the optimal interpolation ratio for each incorrectly classified training example based on Theorem 5.2 with the temperature fixed as 1.0 . One can find that the individual optimal interpolation ratio mostly concentrate between 0.5 and 0.7.

F Method details

F.1 Determine the optimal hyper-parameters

One may note that Equation (9) cannot be directly optimized since the traditional adversarial label is only defined on the example in the training set and cannot be simply generalized to the validation set. A reasonable solution is using the nearest neighbour classifier to find the closest traditional adversarial label for every example in the validation set. However, to speed up the optimization, we propose to employ the classifier overfitted by the traditional adversarial labels on the training set as an surrogate, which works well in practice. Specially, 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

$$T, \lambda = \arg\min_{T,\lambda} \mathbb{E}_{(x',y')\sim\mathcal{D}'_{\text{val}}} \ell\left(\lambda \cdot f_\theta(x';T) + (1-\lambda)\cdot f_{\theta_s}(x';T), y'\right), \tag{37}$$

where $f_{\theta_s}(x';T)$ denotes the temperature-scaled predictive probability of a surrogate model on x'. Here the validation set is constructed by applying adversarial perturbation generated by f_θ to the clean validation set. For adversarial perturbation we utilize PGD attack with 10 iterations, the perturbation radius as $8/255$ and the step size as $2/255$. Note that Such process incurs almost no additional computation as we simply obtain the logits of a surrogate classifier.

G Experimental details

G.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.

G.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 SGD as the optimizer with a fixed learning rate 0.1. When we experiment on a subset (see below) we use the Adam optimizer to improve training stability, where the learning rate is fixed as 0.0001. 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 an exponential number of training epochs, we reduce the size of the training set by randomly sampled a subset of size 5000 from the original training set without replacement. We adopt this setting for extensive experiments for analyzing the dependence of epoch-wise double descent on the perturbation radius and data quality (i.e. Figure 5)..

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.

G.3 Estimation of the data quality

In this section we elaborate on the calculation of data quality for analyzing the dependence on label noise in adversarial training.

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.

G.4 Settings for standard training on fixed augmented training sets

G.4.1 General settings for both adversarial augmentation and Gaussian augmentation

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

Training setting. We conduct the standard training for 1000 epochs. We use Adam as the optimizer with a fixed learning rate 0.0001 to improve training stability with a small training set (see below). 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.

Sample size. To reduce the computation load demanded by an exponential number of training epochs, we reduce the size of the training set by randomly sampled a subset of size 5000 from the original training set without replacement.

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

Hardware. We conduct experiments on NVIDIA Quadro RTX A6000.

G.4.2 Construction of the training set

Adversarial augmentation. We first obtain a robust model by conduct PGD training with pre-activation ResNet-18 on CIFAR-10. We use early stopping to obtain the most robust model on a validation set. The specific settings are aligned with Section G.1.

Using this model, we then generate adversarial examples with PGD attack on the 5000 examples randomly sampled from CIFAR-10 training set. The number of attack iterations is fixed as 10 and the step size is fixed as $2/255$. The adversarial examples along with their original labels are then grouped into a training set for adversarial augmentation experiments.

Gaussian augmentation. We apply Gaussian noise to the 5000 examples randomly sampled from CIFAR-10 training set. The perturbed examples along with their original labels are then grouped into a training set for Gaussian augmentation experiments.