A Proof of Main Theorem

This section presents the detailed proofs of Theorems 4.3 and 4.4 in Section 4.

A.1 Proof of Theorem 4.3

Proof. Let $\mathcal{E} = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \neq f^*(\boldsymbol{x})\}$ be the error region in the image space and $\mathcal{E}_\epsilon = \{\boldsymbol{x} \in \mathcal{X} : \Delta(\boldsymbol{x}, \mathcal{E}) \leq \epsilon\}$ be the ϵ-expansion of $\mathcal{E}$ in metric Δ. By Definition 3.1, we have

$$\text{AdvRisk}_\mu^\epsilon(f) = \mu(\mathcal{E}_\epsilon) = \sum_{i=1}^K p_i \cdot \mu_i(\mathcal{E}_\epsilon) = \sum_{i=1}^K p_i \cdot \text{AdvRisk}_{\mu_i}^\epsilon(f).$$

Since according to Definition 3.3, we have $\text{AdvRisk}_{\mu_i}^\epsilon(f) \geq \text{In-AdvRisk}_{\mu_i}^\epsilon(f)$ for any $i \in [K]$. Thus, it remains to lower bound each term $\text{In-AdvRisk}_{\mu_i}^\epsilon(f)$ individually. For any classifier f, we have

$$\begin{aligned}\text{In-AdvRisk}_{\mu_i}^\epsilon(f) &= \Pr_{\boldsymbol{z}\sim\nu_d}\Big[\exists\, \boldsymbol{z}' \in \mathbb{R}^d, \text{ s.t. } \Delta\big(g_i(\boldsymbol{z}'), g_i(\boldsymbol{z})\big) \leq \epsilon \text{ and } f\big(g_i(\boldsymbol{z}')\big) \neq f^*\big(g_i(\boldsymbol{z}')\big)\Big] \\ &\geq \underbrace{\Pr_{\boldsymbol{z}\sim\nu_d}\Big[\exists\, \boldsymbol{z}' \in \mathcal{B}\big(\boldsymbol{z}, \epsilon/L_i(r)\big), \text{ s.t. } f\big(g_i(\boldsymbol{z}')\big) \neq f^*\big(g_i(\boldsymbol{z}')\big)\Big]}_{I} - \delta \end{aligned} \tag{A.1}$$

where the first inequality is due to $\mu_i = (g_i)_*(\nu_d)$, and the second inequality holds because g_i is $L_i(r)$-locally Lipschitz with probability at least $1-\delta$ and $\mathcal{B}\big(\boldsymbol{z}, \epsilon/L_i(r)\big) \subseteq \mathcal{B}\big(\boldsymbol{z}, r\big)$ for any $\boldsymbol{z} \in \mathbb{R}^d$.

To further bound the term I, we make use of the Gaussian Isoperimetric Inequality as presented in Lemma 4.2. Let $\mathcal{A}_f = \{\boldsymbol{z} \in \mathbb{R}^d : f(g_i(\boldsymbol{z})) \neq f^*(g_i(\boldsymbol{z}))\}$ be the corresponding error region in the latent space. By Lemma 4.2, we have

$$I \geq \Phi\bigg(\Phi^{-1}\big(\nu_d(\mathcal{A}_f)\big) + \frac{\epsilon}{L_i(r)}\bigg) = \Phi\bigg(\Phi^{-1}\big(\text{Risk}_{\mu_i}(f)\big) + \frac{\epsilon}{L_i(r)}\bigg). \tag{A.2}$$

Finally, plugging (A.2) into (A.1), we complete the proof. □

A.2 Proof of Theorem 4.4

Proof. According to Definition 3.2 and Theorem 4.3, for any $f \in \mathcal{F}_\alpha$, we have

$$\begin{aligned}\text{Rob}_\mu^\epsilon(\mathcal{F}_\alpha) &\leq 1 + \delta - \sum_{i=1}^K p_i \cdot \Phi\bigg(\Phi^{-1}\big(\text{Risk}_{\mu_i}(f)\big) + \frac{\epsilon}{L_i(r)}\bigg) \\ &\leq 1 + \delta - \sum_{i=1}^K p_i \cdot \Phi\bigg(\Phi^{-1}\big(\text{Risk}_{\mu_i}(f)\big) + \frac{\epsilon}{L_{\max}(r)}\bigg), \end{aligned} \tag{A.3}$$

where the last inequality holds because $\Phi(\cdot)$ is monotonically increasing. For any $f \in \mathcal{F}_\alpha$, let $\mathcal{E} = \{\boldsymbol{x} \in \mathcal{X} : f(\boldsymbol{x}) \neq f^*(\boldsymbol{x})\}$ be the error region and $\alpha_i = \mu_i(\mathcal{E})$ be the measure of $\mathcal{E}$ under the i-th conditional distribution.

Thus, to obtain an upper bound on $\text{Rob}_\mu^\epsilon(\mathcal{F}_\alpha)$ using (A.3), it remains to solve the following optimization problem:

$$\underset{\alpha_1,\ldots,\alpha_K\in[0,1]}{\text{minimize}} \sum_{i=1}^K p_i \cdot \Phi\bigg(\Phi^{-1}(\alpha_i) + \frac{\epsilon}{L_{\max}(r)}\bigg) \quad \text{subject to } \sum_{i=1}^K p_i\alpha_i \geq \alpha. \tag{A.4}$$

Note that for classifier in $\widetilde{\mathcal{F}}_\alpha$, by definition, we can simply replace $\alpha_i = \alpha$ in (A.4), which proves the upper bound on $\text{Rob}_\mu^\epsilon(\widetilde{\mathcal{F}}_\alpha)$.

Next, we are going to show that the optimal value of (A.4) is achieved, only if there exists a class $i' \in [K]$ such that $\alpha_{i'} = \alpha/p_{i'}$ and $\alpha_i = 0$ for any $i \neq i'$. Consider the simplest case where $K = 2$. Note that $\Phi(\cdot)$ and $\Phi^{-1}(\cdot)$

are both monotonically increasing functions, which implies that $\sum_{i=1}^{K} p_i \alpha_i = \alpha$ holds when optimum achieved, thus the optimization problem for $K = 2$ can be formulated as follows

$$\min_{\alpha_1,\alpha_2\in[0,1]} \; p_1 \cdot \Phi\left(\Phi^{-1}(\alpha_1) + \frac{\epsilon}{L_{\max}(r)}\right) + p_2 \cdot \Phi\left(\Phi^{-1}(\alpha_2) + \frac{\epsilon}{L_{\max}(r)}\right) \quad \text{s.t. } p_1\alpha_1 + p_2\alpha_2 = \alpha. \tag{A.5}$$

Suppose $\alpha_1 \geq \alpha_2$ holds for the initial setting. Now consider another setting where $\alpha_1' > \alpha_1$, $\alpha_2' < \alpha_2$. Let $s_1 = \Phi^{-1}(\alpha_1') - \Phi^{-1}(\alpha_1)$ and $s_2 = \Phi^{-1}(\alpha_2) - \Phi^{-1}(\alpha_2')$. According to the equality constraint of the optimization problem (A.5), we have

$$p_1 \cdot \int_{\Phi^{-1}(\alpha_1)}^{\Phi^{-1}(\alpha_1)+s_1} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-x^2/2}\, dx = p_2 \cdot \int_{\Phi^{-1}(\alpha_2)-s_2}^{\Phi^{-1}(\alpha_2)} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-x^2/2}\, dx. \tag{A.6}$$

Let $\eta = \epsilon / L_{\max}(r)$ for simplicity. By simple algebra, we have

$$\begin{aligned}
p_1 \cdot \int_{\Phi^{-1}(\alpha_1)+\eta}^{\Phi^{-1}(\alpha_1)+s_1+\eta} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-x^2/2}\, dx &= p_1 \cdot \int_{\Phi^{-1}(\alpha_1)}^{\Phi^{-1}(\alpha_1)+s_1} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-u^2/2-\eta\cdot u-\eta^2/2}\, du \\
&< p_1 \cdot \exp^{-\eta\cdot\Phi^{-1}(\alpha_1)-\eta^2/2} \cdot \int_{\Phi^{-1}(\alpha_1)}^{\Phi^{-1}(\alpha_1)+s_1} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-u^2/2}\, du \\
&\leq p_2 \cdot \exp^{-\eta\cdot\Phi^{-1}(\alpha_2)-\eta^2/2} \cdot \int_{\Phi^{-1}(\alpha_2)-s_2}^{\Phi^{-1}(\alpha_2)} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-u^2/2}\, du \\
&< p_2 \cdot \int_{\Phi^{-1}(\alpha_2)-s_2+\eta}^{\Phi^{-1}(\alpha_2)+\eta} \frac{1}{\sqrt{2\pi}} \cdot \exp^{-x^2/2}\, dx,
\end{aligned}$$

where the first inequality holds because $\exp^{-\eta\cdot u} < \exp^{-\eta\cdot\Phi^{-1}(\alpha_1)}$ for any $u > \Phi^{-1}(\alpha_1)$, the second inequality follows from (A.6) and the fact that $\Phi^{-1}(\alpha_1) \geq \Phi^{-1}(\alpha_2)$, and the last inequality holds because $\exp^{-\eta\cdot\Phi^{-1}(\alpha_2)} < \exp^{-\eta\cdot u}$ for any $u < \Phi^{-1}(\alpha_2)$. Therefore, the optimal value of (A.5) will be achieved when $\alpha_1 = 0$ or $\alpha_2 = 0$. For general setting with $K > 2$, since $\alpha_1, \ldots, \alpha_K$ are independent in the objective, we can fix $\alpha_3, \ldots, \alpha_K$ and optimize α_1 and α_2 first, then deal with α_i incrementally using the same technique. □

B Experimental Details

This section provides additional details for our experiments.

B.1 Network Architectures and Hyper-parameter Settings

For the certified robust defense (LP-Certify), we adopt the the same four-layer neural network architecture as implemented in Wong et al. (2018), with two convolutional layers and two fully connected layers, and use the an Adam optimizer with learning rate 0.001 and batch size 50 for training the robust classifier. In particular, the adversarial loss function is based on the robust certificate under ℓ_2 proposed in Wong et al. (2018).

For training attack-based robust models (Adv-Train and TRADES), we use a seven-layer CNN architecture which contains four convolution layers and three fully connected layers. We use a SGD optimizer to minimize the attack-based adversarial loss with learning rate 0.05 on MNIST and learning rate 0.01 on ImageNet10. Table 4 summarizes all the hyper-parameters we used for training the robust models (β is an additional parameter specifically used in TRADES).

For evaluating the unconstrained adversarial robustness, we implemented PGD attack with ℓ_2 metric. Table 5 shows all the hyper-parameters we used for robustness evaluation.

B.2 Strategies for Estimating In-distribution Adversarial Robustness

Initialization of $\boldsymbol{z}$: For MNIST data, we design an initialization strategy for $\boldsymbol{z}$ in order to make sure the perturbation term $\|G(\boldsymbol{z}, y) - \boldsymbol{x}\|_2$ can be efficiently optimized. To be more specific, starting from random noise,

Table 4: Hyper-parameters used for training robust models.

Para.	Generated MNIST			ImageNet10	
	LP-Certified	Adv Training	TRADES	Adv Training	TRADES
ϵ (in ℓ_2)	2.0	3.0	3.0	3.0	3.0
optimizer	ADAM	SGD	SGD	SGD	SGD
learning rate	0.001	0.05	0.05	0.01	0.01
#epochs	60	100	100	100	100
attack step size	-	0.5	0.5	0.5	0.5
#attack steps	-	40	40	10	10
β	-	-	6.0	-	6.0

Table 5: Hyper-parameters used for evaluating the model robustness via PGD attack.

Para.	Generated MNIST			ImageNet10		
	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 3.0$	$\epsilon = 1.0$	$\epsilon = 2.0$	$\epsilon = 3.0$
attack step size	0.1	0.3	0.5	0.1	0.3	0.5
#attack steps	100	100	100	100	100	100

we first solve another optimization problem:

$$\boldsymbol{z}_{\text{init}} = \operatorname*{argmin}_{\boldsymbol{z}} \|G(\boldsymbol{z}, y) - \boldsymbol{x}\|_2.$$

By setting $\boldsymbol{z}_{\text{init}}$ as our initial point, we minimize the initial perturbation distance. Here $\boldsymbol{z}$ can start from any random initial point as we will then optimize the generated image under ℓ_2 distance.

For ImageNet10 data, even applying the above optimization procedure doesn't result in an initial $\boldsymbol{z}$ such that $\|G(\boldsymbol{z}, y) - \boldsymbol{x}\|_2 \leq \epsilon$ when ϵ is small. Therefore, we use another strategy by recording the $\boldsymbol{z}^*$ when generating the test sample $\mathbf{x}$, i.e., $G(\boldsymbol{z}^*, y) = \mathbf{x}$. And we adopt $\boldsymbol{z}^*$ as the initial point for $\boldsymbol{z}$ in solving (5.2). This makes sure that the whole optimization procedure could at least find one point satisfying the perturbation constraint[2].

The choice of λ: Inspired by Carlini and Wagner (2017), we also adopt binary search strategy for finding better regularization parameter λ. Specifically, we set initial $\lambda = 1.0$ and if we successfully find an adversarial example, we lower the value of λ via binary search. Otherwise, we raise the value of λ. For each batch of examples, we perform 5 times binary search in order to find qualified in-distribution adversarial examples.

Hyper-parameters: We use Adam optimizer with learning rate 0.01 for finding in-distribution adversarial examples. We set maximum iterations for each λ binary search as 10000.

[2]We didn't use $\boldsymbol{z}^*$ as the initialization for MNIST data as our empirical study shows that the optimization-based initialization achieves better performances on MNIST.