Provably Robust Cost-Sensitive Learning via Randomized Smoothing

Yuan Xin [1] Michael Backes [1] Xiao Zhang [1]

Abstract

We focus on learning adversarially robust classifiers under cost-sensitive scenarios, where the potential harm of different classwise adversarial transformations is encoded in a cost matrix. Existing methods either are empirical that cannot certify robustness or suffer from inherent scalability issues. In this work, we study whether randomized smoothing, a scalable robustness certification framework, can be leveraged to certify cost-sensitive robustness. We first show how to extend the vanilla certification pipeline to provide rigorous guarantees for cost-sensitive robustness. However, when adapting the standard randomized smoothing method to train for cost-sensitive robustness, we observe that the naive reweighting scheme does not achieve a desirable performance due to the indirect optimization of the base classifier. Inspired by this observation, we propose a more direct training method with fine-grained certified radius optimization schemes designed for different data subgroups. Experiments on image benchmarks demonstrate that our method significantly improves certified cost-sensitive robustness without sacrificing overall accuracy.

1. Introduction

Recent studies have revealed that deep learning models are highly vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014). To defend against such attacks, various defensive mechanisms have been proposed, primarily falling into two categories: *empirical defenses* (Goodfellow et al., 2014; Carlini & Wagner, 2016; Kurakin et al., 2016; Madry et al., 2017; Carlini & Wagner, 2017) and *certified methods* (Raghunathan et al., 2018; Wong & Kolter, 2018; Gowal et al., 2018; Cohen et al., 2019; Lecuyer et al., 2019; Jia et al., 2019; Li et al., 2019). In particular, certified methods can produce a certificate for the model prediction to remain unchanged within certain neighborhood of any input and train models to be provably robust for some specific norm-bounded adversarial perturbations.

Most existing adversarial defenses aim to improve the classifier's overall robustness, which assumes the same penalty on all kinds of adversarial misclassifications. For real-world applications, however, it is likely that some specific misclassifications are more important than others (Domingos, 1999; Elkan, 2001). For instance, misclassifying a malignant tumor as benign in the application of medical diagnosis is much more detrimental to a patient than the reverse. Therefore, instead of solely focusing on enhancing overall robustness, the development of defenses should also account for the difference in costs induced by different adversarial examples. In line with existing works on cost-sensitive robust learning (Asif et al., 2015; Zhang & Evans, 2018; Domingos, 1999; Chen et al., 2021), our objective is to develop models that are robust to cost-sensitive adversarial misclassifications, while maintaining the standard overall classification accuracy. However, existing defenses are either hindered by their foundational reliance on heuristics, which often fall short of providing a robustness guarantee (Domingos, 1999; Asif et al., 2015; Chen et al., 2021), or suffer from inherent scalability issues (Zhang & Evans, 2018). Detailed discussions of related works are provided in Appendix A.

To achieve the best of both worlds, we propose to train certified cost-sensitive robust classifiers using randomized smoothing (Cao & Gong, 2017; Liu et al., 2018; Cohen et al., 2019), a new certification technique that has attracted a lot of attention due to its simplicity and scalability. However, we discover that a straightforward reweighting scheme, typically employed for cost-sensitive learning, does not adapt well when training a smoothed classifier, due to the indirect optimization of the base classifier and the non-optimal tradeoff between sensitive and non-sensitive examples. Therefore, we take the distinctive properties of different data subgroups into account and design an advanced certified cost-sensitive robust training method based on MACER (Zhai et al., 2020) to directly optimize the certified radius with respect to the smoothed classifier.

Contributions. We are the first to adapt the randomized smoothing framework to certify and train for cost-sensitive

[1] CISPA Helmholtz Center for Information Security, Saarbrücken, Saarland, Germany. Correspondence to: Yuan Xin <yuan.xin@cispa.de>, Xiao Zhang <xiao.zhang@cispa.de>.

2^{nd} *AdvML Frontiers workshop at* 40^{th} *International Conference on Machine Learning*, Honolulu, Hawaii, USA. PMLR 202, 2023.

robustness. In particular, for any given binary-valued cost matrix, we introduce the notion of *cost-sensitive certified radius* (Definition 3.1), which captures the maximum allowable ℓ_2 perturbation with respect to the smoothed classifier for each input from a cost-sensitive seed class. We show that the proposed definition subsumes the standard notion of certified radius typically used in randomized smoothing, thus is more tailored for certifying cost-sensitive robustness (Section 3.1). Built upon the definition of cost-sensitive certified radius, we propose a practical certification algorithm based on Monte Carlo samples (Algorithm 1). To train for smoothed classifiers, we attempt to adapt the reweighting method which commonly used in vanilla cost-sensitive learning, but observe a big accuracy drop for non-sensitive seed examples (Section 4.1). Instead, we resort to MACER (Zhai et al., 2020), a more direct training method and design a more fine-grained certified robust training method by considering the distinction between different data subgroups (Section 4.2). Experiments on image benchmarks illusrtrate the superiority of our method, compared with baseline randomized smoothing methods, in attaining high cost-sensitive robustness for both seed-wise and pairwise cost matrices while keeping overall accuracy (Section 5).

2. Preliminaries

In this section, we briefly introduce randomized smoothing and the problem setting for cost-sensitive robustness.

Randomized Smoothing. Randomized smoothing (RS) is a scalable, probabilistic certification framework proposed in Cohen et al. (2019) for certifying model robustness. This framework leverages the set of smoothed classifiers, which first augment normal inputs with randomly sampled Gaussian noise, then pass the noisy inputs through a base classifier f_θ and aggregate their predictions using majority voting.

Definition 2.1. Let $\mathcal{X}$ be the input space and $[m] := \{1, 2, \ldots, m\}$ be the set of class labels. For any base classifier f_θ and $\sigma > 0$, the *smoothed classifier* is defined as:

$$g_\theta(\boldsymbol{x}) = \underset{j\in[m]}{\arg\max}\, \mathbb{P}_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}[f_\theta(\boldsymbol{x}+\boldsymbol{\delta}) = j],\ \forall \boldsymbol{x}\in\mathcal{X}.$$

To simplify notations, let $h_\theta : \mathcal{X} \to [0, 1]^m$ be the mapping from the input space $\mathcal{X}$ to the prediction probabilities of g_θ:

$$[h_\theta(\boldsymbol{x})]_j = \mathbb{P}_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}[f_\theta(\boldsymbol{x}+\boldsymbol{\delta}) = j],\ \forall j\in[m].$$

The following theorem characterizes the maximum allowable ℓ_2-perturbation radius for any input $\boldsymbol{x} \in \mathcal{X}$ such that the prediction of g_θ remains the same within the radius.

Theorem 2.2 (Cohen et al. (2019))**.** *Let $\boldsymbol{x} \in \mathcal{X}$ and y be the ground-truth class of $\boldsymbol{x}$. If g_θ classifies $\boldsymbol{x}$ correctly:*

$$\mathbb{P}_{\boldsymbol{\delta}}(f_\theta(\boldsymbol{x}+\boldsymbol{\delta}) = y) > \max_{j\neq y}\mathbb{P}_{\boldsymbol{\delta}}(f_\theta(\boldsymbol{x}+\boldsymbol{\delta}) = j),$$

where $\boldsymbol{\delta}$ is sampled i.i.d. from $\mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, then g_θ is provably robust at $\boldsymbol{x}$ with certified radius in ℓ_2-norm given by:

$$R(\boldsymbol{x}) = \frac{\sigma}{2}\big[\Phi^{-1}\big([h_\theta(\boldsymbol{x})]_y\big) - \Phi^{-1}\big(\max_{j\neq y}\,[h_\theta(\boldsymbol{x})]_j\big)\big], \quad (1)$$

where Φ is the Gaussian CDF and Φ^{-1} denotes its inverse.

Cost-Sensitive Robustness. We consider robust classification tasks for cost-sensitive scenarios, where the goal is to learn a classifier with both high overall accuracy and cost-sensitive robustness. Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space and $[m]$ be the label space. Suppose $C \in \{0, 1\}^{m\times m}$ is a cost matrix that encodes the potential harm of different classwise adversarial transformations. Note that we only consider binary-valued cost matrices in this work, where extending our method and results to more general real-valued cost matrices (Zhang & Evans, 2018) is straightforward. In particular, $C_{jj'} = 1$ means misclassifications from seed class $j \in [m]$ to target class $j' \in [m]$ will bring a cost, whereas $C_{jj'} = 0$ suggests no incentive for an attacker to trick the model to misclassify inputs from class j to class j'. Therefore, we aim to reduce the number of adversarial misclassifications that induce a cost defined by C.

Moreover, we introduce the following notations for the ease of presentation. For any seed class $j \in [m]$, we let $\Omega_j = \{j' \in [m] : c_{jj'} = 1\}$ be the set of cost-sensitive target classes. If Ω_j is an empty set, all the examples from seed class j is non-sensitive. Otherwise, we call any class j with $|\Omega_j| \geq 1$ a *sensitive seed class*. Correspondingly, given a dataset $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[n]}$, we define the set of cost-sensitive examples as $\mathcal{S}^s = \{(\boldsymbol{x}, y) \in \mathcal{S} : |\Omega_y| \geq 1\}$, while the remaining examples are all non-sensitive. In particular, we study the following two categories of cost matrices:

1. *Seed-wise cost matrix*: for any $(\boldsymbol{x}, y) \in \mathcal{S}^s$, $\Omega_y = \{j \in [m] : j \neq y\}$, meaning that all possible classwise adversarial transformations will incur a cost.

2. *Pairwise cost matrix*: for any $(\boldsymbol{x}, y) \in \mathcal{S}^s$, the cost-sensitive target class set Ω_y is a proper subset of $[m]$, and misclassifying to any target class in $[m]\backslash\Omega_y$ is acceptable. It is worth noting that $[m]\backslash\Omega_y$ may include target classes other than the ground-truth class y.

Note that the commonly-used definition of *overall robustness* can be regarded as a special case of seed-wise cost matrix, where the non-diagonal entries of C are all 1.

3. Certifying Cost-Sensitive Robustness

In this section, we illustrate how to certify cost-sensitive robustness using randomized smoothing. We first introduce the formal definition of cost-sensitive certified radius then discuss its connection to the standard certified radius (Section 3.1). Finally, based on the proposed definition, we design a practical certification algorithm using finite samples (Section 3.2) and lay out the metrics for measuring certified cost-sensitive robustness and overall accuracy (Section 3.3).

3.1. Cost-Sensitive Certified Radius

Recall that for any example $(\boldsymbol{x}, y) \in \mathcal{S}^s$, only misclassifying $\boldsymbol{x}$ to a target class in Ω_y incurs a cost, whereas misclassifications to any class from $[m]\backslash\Omega_y$ is tolerable. Below, we formally define *cost-sensitive certified radius*, which adapts the standard certified radius to cost-sensitive scenarios:

Definition 3.1 (Cost-Sensitive Certified Radius)**.** Consider the same setting as in Theorem 2.2. Let C be an $m \times m$ cost matrix. For any example $(\boldsymbol{x}, y)$ where $y \in [m]$ is a sensitive seed class, the *cost-sensitive certified radius* at $(\boldsymbol{x}, y)$ with respect to C is defined as:

$$R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta) = \frac{\sigma}{2}\Big[\Phi^{-1}\big(\max_{j\in[m]}\,[h_\theta(\boldsymbol{x})]_j\big) - \Phi^{-1}\big(\max_{j\in\Omega_y}\,[h_\theta(\boldsymbol{x})]_j\big)\Big] \quad (2)$$

where $\Omega_j = \{j^{'} \in [m] : c_{jj'} = 1\}$ for any $j \in [m]$ and Φ^{-1} denotes the inverse CDF of standard Gaussian $\mathcal{N}(0, 1)$.

Similarly, we can extend the proof of Theorem 2.2 to the cost-sensitive settings to produce a robustness certificate using the above definition of cost-sensitive certified radius.

Theorem 3.2. *Consider the same setting as in Definition 3.1. For any example $(\boldsymbol{x}, y)$, if the predicted class of the smoothed classifier g_θ at $\boldsymbol{x}$ does not incur a cost:*

$$\max_{j\in[m]}[h_\theta(\boldsymbol{x})]_j \geq \max_{j\in\Omega_y}\,[h_\theta(\boldsymbol{x})]_j, \quad (3)$$

then g_θ is provably robust at $\boldsymbol{x}$ with certified radius given by $R_{c\text{-}s}(\boldsymbol{x}; \Omega_y, h_\theta)$ measured in ℓ_2-norm.

Theorem 3.2 can be applied to certify cost-sensitive robustness for any binary-valued cost matrix setting. According to the definition of $h_\theta(\boldsymbol{x})$, the term on the left hand size $\max_{j\in[m]}[h_\theta(\boldsymbol{x})]_j$ denotes the maximum predicted probability across all classes, while $\max_{j\in\Omega_y}\,[h_\theta(\boldsymbol{x})]_j$ denotes the maximum predicted probability across all sensitive target classes within Ω_y. Therefore, if the condition specified by Equation 3 holds, the predicted class of g_θ will fall out of Ω_y which is acceptable for cost-sensitive robustness.

The following theorem, proven in Appendix B, characterizes the connection between cost-sensitive certified radius and the standard notion for different cost matrix scenarios.

Theorem 3.3. *For any seed-wise cost matrix, our cost-sensitive certified radius equals to Cohen et al. (2019)'s standard certified radius, i.e, $R_{c\text{-}s}(\boldsymbol{x}; \Omega_y, h_\theta) = R(\boldsymbol{x})$. For pairwise cost-sensitive scenarios, if the prediction $g_\theta(\boldsymbol{x})$ does not incur a cost, then $R_{c\text{-}s}(\boldsymbol{x}; \Omega_y, h_\theta) \geq R(\boldsymbol{x})$.*

Theorem 3.3 suggests that using $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$ always yields a higher certified cost-sensitive robustness for any binary cost matrix. In particular, for pairwise cost matrices, there will be a larger benefit in choosing $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$.

3.2. Practical Certification Algorithm

According to the construction of h_θ, it requires access to an infinite number of Gaussian samples to compute the cost-sensitive certified radius. However, it is computationally infeasible in practice to obtain the true value of $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$ even for a single example $(\boldsymbol{x}, y)$. In addition, as discussed in Theorem 3.3, the key difference between $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$ and the standard certified radius $R(\boldsymbol{x})$ lies in the pairwise cost matrix scenarios, which also necessitates a new certification process. In this section, we put forward a new Monte Carlo algorithm for certifying cost-sensitive robustness by adapting Cohen et al. (2019)'s, which can be applied to any binary cost matrix scenario.

Algorithm 1 describes the pseudocode of the proposed certification algorithm. In particular, it provides two different ways to compute probabilistic bounds on cost-sensitive certified radius. R_1 is computed using a lower $1 - \alpha$ confidence bound on p_A, while R_2 is computed using both a lower $1 - \alpha/2$ confidence bound of p_A and an upper $1 - \alpha/2$ confidence bound of p_B. According to Theorem 1 and Proposition 2 in (Cohen et al., 2019), we can show by union bounds that with probability at least $1 - \alpha$ over the randomness of Gaussian samples, the returned output $\max(R_1, R_2)$ by Algorithm 1 is guaranteed to be a certified radius for any given cost matrix. In other words, the prediction of the smoothed classifier g_θ at $(\boldsymbol{x}, y)$ will not incur any undesirable cost for any ℓ_2 perturbations within radius $\max(R_1, R_2)$. Details of the sampling scheme used in Algorithm 1 and more discussions are provided in Appendix C.

We remark that the certification algorithm for overall robustness proposed in Cohen et al. (2019) only considers the first approach to compute certified radius. This is because the first approach always yields a larger lower confidence bound on p_A by choosing $1 - \alpha$ thus a larger certified radius, under condition that p_B is close to $1 - p_A$ which typically holds for seed-wise cost matrices. For certain pairwise scenarios, however, it is possible that $R_2 > R_1$ for some inputs, since the maximum class probabilities for computing p_B with respect to Ω_y could be very small, which may not contain the second-highest probability class, especially when the number of cost-sensitive target classes $|\Omega_y|$ is small. To ensure that we always produce the largest possible certified radius for any scenario, Algorithm 1 selects the larger value of R_1 and R_2 to be its output for any cost-sensitive example.

Algorithm 1 Certification for Cost-Sensitive Robustness

```
function CERTIFY($f, \sigma, \boldsymbol{x}, n_0, n, \alpha, \Omega_y$)
$counts_0 \leftarrow$ SAMPLEUNDERNOISE($f, \boldsymbol{x}, n_0, \sigma$)
$\hat{c}_A \leftarrow$ top index in $counts_0$
$\hat{c}_B \leftarrow$ top index in $counts_0[\Omega_y]$
$counts \leftarrow$ SAMPLEUNDERNOISE($f, \boldsymbol{x}, n, \sigma$)
$\underline{p_A} \leftarrow$ LOWERCONFBND($counts[\hat{c}_A], n, 1-\alpha$)
$R_1 = \sigma\Phi^{-1}(\underline{p_A})$
$\underline{p_A} \leftarrow$ LOWERCONFBND($counts[\hat{c}_A], n, 1-\alpha/2$)
$\overline{p_B} \leftarrow$ UPPERCONFBND($counts[\hat{c}_B], n, 1-\alpha/2$)
$R_2 = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$
if $\max(R_1, R_2) > 0$ then
    return prediction $\hat{c}_A$ and $\max(R_1, R_2)$
else
    return ABSTAIN
end if
```

3.3. Evaluation Metrics

Recall that the goal of robust cost-sensitive learning is to produce a classifier that is both robust to cost-sensitive adversarial misclassifications, while maintaining overall accuracy at the same time. Based on the definition of cost-sensitive certified radius introduced in Section 3.1 and Algorithm 1, we can now define the evaluation metrics for measuring a classifier's certified performance in cost-sensitive scenarios.

For any binary cost matrix, we define the *certified cost-sensitive robustness* of a smoothed classifier g_θ over dataset $\mathcal{S} = \{\boldsymbol{x}_i, y_i\}_{i\in[n]}$ as the ratio of cost-sensitive examples that are provably robust against ℓ_2 perturbations with ϵ:

$$\mathrm{Rob}_{\text{c-s}}(g_\theta) = \frac{1}{|\mathcal{S}^s|}\sum_{(\boldsymbol{x},y)\in\mathcal{S}^s} \mathbb{1}\big\{R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta) > \epsilon\big\}. \quad (4)$$

In practice, the term $R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta)$ in Equation 4 will be replaced by the output $\max(R_1, R_2)$ of Algorithm 1, leading to an empirical estimate $\widehat{\mathrm{Rob}}_{\text{c-s}}(g_\theta)$. As explained in Section 3.2, $\max(R_1, R_2)$ is guaranteed to be a probabilistic lower bound of $R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta)$ with $1-\alpha$ confidence level. Therefore, g_θ will be provably robust with certified cost-sensitive robustness $\widehat{\mathrm{Rob}}_{\text{c-s}}(g_\theta)$ against ℓ_2 perturbations with ϵ.

The *overall accuracy* of g_θ is defined as the fraction of correctly classified samples with respect to the whole dataset:

$$\mathrm{Acc}(g_\theta) = \frac{1}{|\mathcal{S}|}\sum_{(x,y)\in\mathcal{S}} \mathbb{1}\big\{R(\boldsymbol{x}) > 0\big\}. \quad (5)$$

We follow the standard procedure of randomized smoothing (Cohen et al., 2019) to estimate $\mathrm{Acc}(g_\theta)$ in our experiments.

4. Training for Cost-Sensitive Robustness

A popular training scheme for cost-sensitive learning is reweighting (Elkan, 2001), which assigns larger weights to cost-sensitive inputs during model training. Thus, a natural question is whether the reweighting scheme can be incorporated in randomized smoothing to train for cost-sensitive robustness. In this section, we first study the effectiveness of reweighting methods, where we empirically observe a non-optimal trade-off between the performance of sensitive and non-sensitive examples. Motivated by this observation, we propose to leverage the design insight of MACER (Zhai et al., 2020) and develop a training method to maximize certified cost-sensitive robustness while minimizing the impact on non-sensitive data (Section 4.2).

4.1. Reweighting Methods Sacrifice Overall Accuracy

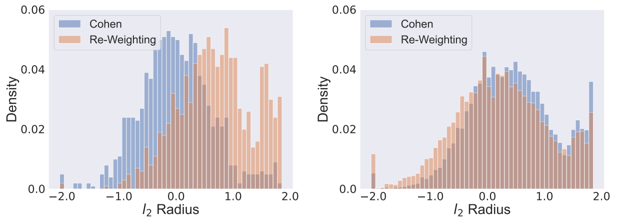
We consider the base classifier training method introduced in Cohen et al. (2019), which proposes to inject Gaussian noise to all inputs during the training process of f_θ. Given a binary cost matrix, let $\mathcal{D}_s$ be the distribution of all sensitive examples which incur costs if misclassified and let $\mathcal{D}_n$ represent the distribution of the remaining normal examples. Intuitively, the training pipeline of randomized smoothing can be adapted to cost-sensitive settings using a simple reweighting scheme by increasing the weights assigned to the loss function of sensitive examples. More concretely, the total training objective function is defined as follows:

$$\min_{\theta\in\Theta} \mathbb{E}_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbf{I})}\Big[\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_n}\, \mathcal{L}_{CE}(f_\theta(\boldsymbol{x}+\boldsymbol{\delta}),y) + \alpha\cdot\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_s}\, \mathcal{L}_{CE}(f_\theta(\boldsymbol{x}+\boldsymbol{\delta}),y)\Big], \quad (6)$$

where Θ denotes the set of model parameters, $\mathcal{L}_{CE}$ represents the cross-entropy loss, and $\alpha \geq 1$ is a trade-off parameter which controls the performance between sensitive and non-sensitive examples. When $\alpha = 1$, the above objective function is equivalent to the training loss used in standard randomized smoothing (Cohen et al., 2019). However, we note that such adaptation of reweighting (Equation 6) can only works for seed-wise cost matrices, which can not be directly applied for pairwise cost matrix scenarios.

Visualizations. We further study the performance of the *naive reweighting* scheme defined by Equation 6, where we empirically observe that naive reweighting sacrifices overall accuracy if we target for improved cost-sensitive robustness. Figure 1 visualizes the distributions of certified radius for both sensitive and non-sensitive classes with respect to the smoothed classifier learned by naive reweighting. We also tune the trade-off parameter α to maximize the performance of the final produced smoothed classifier. Here, the certified radius is estimated using Equation 2 with empirical Gaussian samples on the testing dataset. Note that negative certified radius indicates an incorrect classification, whereas positive radius means the prediction of g_θ is correct.

Compared with standard randomized smoothing, naive



(a) Sensitive Class (b) Normal Class

Figure 1. Density plots of certified radius regarding g_θ learned by Cohen et al. (2019)'s and naive reweighting methods with $\sigma = 0.5$ for different set of examples: (a) sensitive class (b) normal class. The sensitive seed class is selected as the CIFAR-10 class "cat".

reweighting increases the certified radius for sensitive class, suggesting an improvement in cost-sensitive robustness. However, the normal class's radius distribution shifts to the left for the naive reweighting method. In particular, for the cost matrix scenario considered in Figure 1, naive reweighting improves certified cost-sensitive robustness to a large extent from 19% to 58% compared with Cohen et al. (2019)'s, but the overall accuracy degrades from 65% to 62%. This observation illustrates that the naive adaptation of a reweighting scheme tends to result in an undesirable sacrifice of overall clean accuracy, likely due to the fact that reweighting is only indirectly applied to the smoothed classifier g_θ through optimization of the base classifier f_θ.

4.2. Our Method

Motivated by the observation discussed in Section 4.1, we propose a more direct optimization scheme based on the proposed notion of certified cost-sensitive radius, which leverages a similar insight of MACER (Zhai et al., 2020).

To simplify notations, we first introduce a general class of margin-based losses. Given $l \leq u$ denoting the thresholding parameters, we define the following class of margin losses: for any $r \in \mathbb{R}$ representing the certified radius,

$$\mathcal{L}_M\big(r; l, u\big) = \max\{u - r, 0\} \cdot \mathbb{1}(l \leq r \leq u).$$

Here, the indicator function selects data points whose certified radius falls within the range of $[l, u]$.

For seed-wise cost matrices, the training objective of method is defined as:

$$\begin{aligned} &\min_{\theta\in\Theta} \quad I_1 + \lambda \cdot I_2 + \lambda \cdot I_3, \qquad (7)\\ &\text{where} \quad I_1 = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\, \mathcal{L}_{CE}(h_\theta(\boldsymbol{x}), y),\\ &\qquad\quad I_2 = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_n}\, \mathcal{L}_M\big(R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta); 0, \gamma_1\big),\\ &\qquad\quad I_3 = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}_s}\, \mathcal{L}_M\big(R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta); -\gamma_2, \gamma_2\big), \end{aligned}$$

where $\lambda, \gamma_1, \gamma_2 > 0$ are hyperparameters, $\mathcal{D}$ is the underlying data distribution, and $\mathcal{D}_s, \mathcal{D}_n$ denote the distributions of cost-sensitive and the normal examples, respectively.

Equation 7 consists of three terms: I_1 represents the cross-entropy loss with respect to h_θ over $\mathcal{D}$, which controls the overall accuracy; I_2 and I_3 control the robustness with a shared trade-off parameter λ. The range of the interval $[l, r]$ represents which data subpopulation we want to optimize. A larger thresholding parameter such as γ_1 and γ_2 lead to a higher data coverage, whereas the range with a smaller threshold includes fewer data points. We set $\gamma_2 > \gamma_1$ to have a wider adjustment range for sensitive seed examples. As shown in Wang et al. (2020), optimizing misclassified samples can help adversarial robustness, thus, we intend to include sensitive seed examples with a negative radius in $[-\gamma_2, 0)$ in the design of I_3 for a better performance.

For pairwise cost matrices, we replace the term I_2 by:

$$I_2' = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\, \mathcal{L}_M\big(R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta); 0, \gamma_1\big).$$

In seed-wise cost matrix optimization, I_2 only includes normal seed examples, while in pairwise cases, we also include all the sensitive seed examples. This difference mainly lies in that for pairwise scenarios, there is a mismatch between the cost-sensitive certified radius $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$ and the standard certified radius $R(\boldsymbol{x})$ due to the difference between the cost-sensitive target set Ω_y and the ground-truth label y. Therefore, only considering normal seed examples would not be sufficient to ensure a desirable accuracy performance on sensitive distribution $\mathcal{D}_s$ under pairwise settings.

Intuitively speaking, by imposing different threshold restrictions $[l, u]$ on the certified radius of sensitive seed classes and normal seed classes, the optimization process can prioritize making adjustments to data subpopulations of specific classes rather than considering all data points belonging to those classes. This is also a key advantage of our method over the naive reweighting method. As will be shown in our experiments, such fine-grained optimization enables our method to improve certified cost-sensitive robustness to a large extent without sacrificing overall accuracy.

5. Experiments

We evaluate the performance of our method on two image benchmark datasets: CIFAR-10 (Krizhevsky et al., 2009) and MNIST (LeCun et al., 1998). For CIFAR-10, we use the same ResNet (He et al., 2016) architecture as employed in Cohen et al. (2019). Specifically, we choose ResNet-56 network, since it attains comparable performance to ResNet-110 with a shorter computation time. Following existing works (Zhai et al., 2020; Shafahi et al., 2019), we choose the commonly-used LeNet (LeCun et al., 2015) for MNIST.

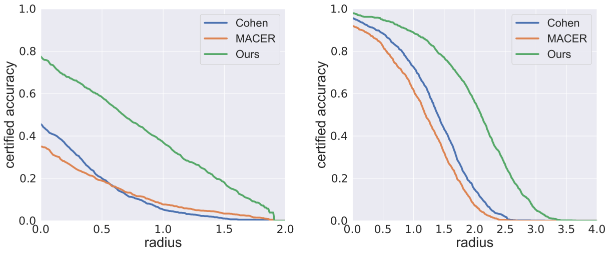
Baselines. We primarily compare our method with two baseline randomized smoothing methods: *Cohen* (Cohen et al.,

Table 1. Certification results of different randomized smoothing based training methods for various seed-wise cost matrices. The noise level σ is 0.5 for CIFAR-10 and 1.0 for MNIST. $\textbf{Acc}$ stands for overall accuracy and $\textbf{Rob}_{\text{c-s}}$ is certified cost-sensitive robustness with $\epsilon = 0.5$. Our results are highlighted in bold.

Task	Type	Method	Acc	$\textbf{Rob}_{\text{c-s}}$
CIFAR	Vuln (3)	Cohen	0.654	0.193
		MACER	0.647	0.189
		Ours	$\mathbf{0.661}$	$\mathbf{0.583}$
	Rand (4)	Cohen	0.654	0.358
		MACER	0.647	0.438
		Ours	$\mathbf{0.654}$	$\mathbf{0.724}$
	Multi (2, 4)	Cohen	0.654	0.253
		MACER	0.647	0.233
		Ours	$\mathbf{0.654}$	$\mathbf{0.455}$
MNIST	Rand (4)	Cohen	0.964	0.867
		MACER	0.940	0.807
		Ours	$\mathbf{0.949}$	$\mathbf{0.930}$
	Multi (4, 7)	Cohen	0.964	0.838
		MACER	0.940	0.837
		Ours	$\mathbf{0.973}$	$\mathbf{0.912}$

2019) and *MACER* (Zhai et al., 2020). We select Cohen for comparisons with standard randomized smoothing and MACER for comparing with methods that optimize for certified radius. Both of these baselines are optimized for overall robustness, so we expect our method by design can largely improve the cost-sensitive robustness. In addition, our experiments are mainly conducted for two categories of cost matrices: seed-wise and pairwise, as their corresponding certification and training procedures are different.

Experimental Details. For both CIFAR-10 and MNIST, we follow Zhai et al. (2020)'s training setting with a total of 440 training epochs and the same learning rate decay schedule (an initial learning rate of 0.01 and decay factor of 0.1 every 200 epochs). The main difference between our method and MACER is the parameter choice of λ and γ. λ trades off the overall accuracy and certified robustness. For CIFAR10 dataset we find $\lambda = 3$ works best for cost-sensitive scenarios whereas MACER selects $\lambda = 4$. For MNIST dataset, we follow the setting in MACER where $\lambda = 16$. In addition, MACER uses $\gamma = 8$ to enhance all classes' robustness, whereas we set $\gamma_1 = 16$ for sensitive classes and $\gamma_2 = 4$ for normal classes, since such choices lead to the optimal result. Our method's performance for other combinations of hyperparameters is discussed in Appendix E. In addition, we compare our method with the convex relaxation-based method of Zhang & Evans (2018) in Appendix D.



(a) CIFAR-10 (b) MNIST

Figure 2. Certified accuracy curves under seed-wise cost matrix settings. Here, we choose class "cat" as the single sensitive seed class for CIFAR-10 and digit 4 as the sensitive seed for MNIST.

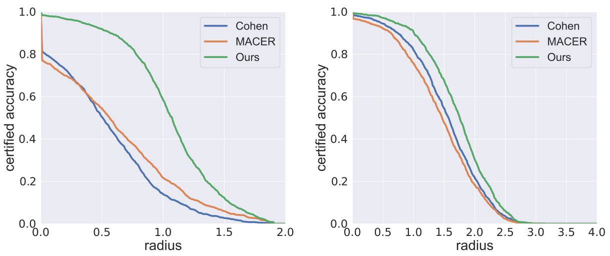
Seed-wise Cost Matrix. Table 1 reports the performance in terms of overall accuracy and certified cost-sensitive robustness of our method and the two baselines with respect to different seed-wise cost matrices. In particular, we consider three types of seed-wise cost matrices in our experiments: (1) *Vuln:* a "vulnerable" sensitive seed class, where we choose the class "cat" (label 3) for CIFAR-10, since standard trained classifiers achieve the highest misclassification rates for such class; (2) *Rand:* a randomly-selected sensitive seed class from all available classes, where we report the performance on the fourth class "deer" (label 4) for CIFAR-10 and digit 4 for MNIST; (3) *Multi:* multiple sensitive seed classes, where "bird" (label 2) and "deer" (label 4) are considered as the sensitive seed classes for CIFAR-10, while we choose digits 4 and 7 as sensitive for MNIST. We observe in Table 1 that our cost-sensitive robust training method achieves a significant improvement in terms of certified cost-sensitive robustness compared with baselines. In addition, our method achieves comparable or slightly improved overall accuracy performance for all cost matrix scenarios.

Moreover, we further compare the certified accuracy curves of cost-sensitive examples with varying radius for the aforementioned methods in Figure 2. For CIFAR-10, it is evident that our method consistently outperforms the baseline in terms of certified cost-sensitive accuracy across various radius. More specifically, when the radius equals zero, our method's cost-sensitive accuracy peaks at 76.9%, a significant improvement over Cohen's 45.5% and MACER's 35%. When the radius is 0.5, the results correspond to certified cost-sensitive robustness presented in Table 1. Similar trends are observed for MNIST, and we note that once the radius exceeds 2.5, the performance of the baselines drops to zero, whereas our method still achieves non-trivial certified accuracy, indicating its robustness to larger perturbations.

Pairwise Cost Matrix. It is worth noting that the certification Algorithm 1 and our training method proposed in Section 4.2 for pairwise cost matrices are different from the seed-wise scenarios. Similar to the previous setting,

Table 2. Certification results for pairwise cost matrices. The noise level σ is 0.5 for CIFAR-10 and 1.0 for MNIST. $\mathbf{Acc}$ stands for overall accuracy and $\mathbf{Rob}_{\text{c-s}}$ refers to certified cost-sensitive robustness with $\epsilon = 0.5$. Our results are highlighted in bold.

Task	Type	Methods	Acc	$\mathbf{Rob}_{\text{c-s}}$
CIFAR	Rand $(3 \to 5)$	Cohen	0.654	0.504
		MACER	0.647	0.543
		Ours	**0.673**	**0.924**
	Multi $(3 \to 2, 4, 5)$	Cohen	0.654	0.336
		MACER	0.647	0.385
		Ours	**0.643**	**0.822**
MNIST	Rand $(4 \to 9)$	Cohen	0.964	0.934
		MACER	0.940	0.908
		Ours	**0.954**	**0.971**
	Multi $(4 \to 3, 5, 9)$	Cohen	0.964	0.924
		MACER	0.940	0.895
		Ours	**0.950**	**0.957**



(a) CIFAR-10 (b) MNIST

Figure 3. Certified accuracy curves for cost-sensitive examples under pairwise cost-sensitive scenarios. Here, we select pairwise transformation $3 \to 5$ as sensitive for CIFAR-10, while the right figure is for MNIST with sensitive transformation $4 \to 9$.

we consider two types of pairwise cost matrices: (1) *Rand:* a randomly-selected sensitive pairwise transformation; (2) *Multi:* a single sensitive seed class with multiple sensitive target classes. Table 2 compares the performance of our method with baselines for the aforementioned pairwise cost matrices on CIFAR-10 and MNIST datasets, while Figure 3 depicts the certified accuracy curves for cost-sensitive examples with varying radius for different methods.

Compared with our results for seed-wise cost sensitive settings, our method achieves a larger improvement in cost-sensitive robustness for pariwise cost matrices. This larger improvement can be attributed to the design of our method that optimizes the cost-sensitive certified radius $R_{\text{c-s}}(\boldsymbol{x}; \Omega_y, h_\theta)$, which is more tailored for pairwise cost-sensitive scenarios. This also confirms the superiority of our certification algorithm for certifying cost-sensitive robustness over the standard one. We notice that for certain pairwise cost matrices, there is a slight drop in overall clean accuracy. Given the significant improvement in cost-sensitive performance, such small decrease in overall performance is negligable, which can be tolerated for typical applications.

6. Conclusion

We developed a generic randomized smoothing framework to certify and train for cost-sensitive robustness. At the core of our framework is a new notion of cost-sensitive certified radius, which is applicable to any binary cost matrix. Built upon fine-grained thresholding techniques for optimizing the certified radius with respect to different subpopulations, our method significantly improves the certified robustness performance for cost-sensitive transformations. Compared with naive reweighting approaches, our method achieves a much more desirable trade-off between overall accuracy and certified cost-sensitive robustness. Experiments on image benchmarks demonstrate the superior performance of our approach compared to various baselines. Our work opens up new possibilities for building certified robust models based on randomized smoothing for cost-sensitive applications.

Acknowledgement

We would like to thank anonymous reviewers for their insightful and constructive reviews for us to improve our work. We want to also thank Dingfan Chen for her help with proofreading the manuscript and providing valuable feedback.

References

Asif, K., Xing, W., Behpour, S., and Ziebart, B. D. Adversarial cost-sensitive classification. In *UAI*, pp. 92–101, 2015.

Cao, X. and Gong, N. Z. Mitigating evasion attacks to deep neural networks via region-based classification. In *Proceedings of the 33rd Annual Computer Security Applications Conference*, pp. 278–287, 2017.

Carlini, N. and Wagner, D. Defensive distillation is not robust to adversarial examples. *arXiv preprint arXiv:1607.04311*, 2016.

Carlini, N. and Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pp. 3–14, 2017.

Chen, Y., Wang, S., Jiang, W., Cidon, A., and Jana, S. Cost-aware robust tree ensembles for security applications. In *USENIX Security Symposium*, pp. 2291–2308, 2021.

Cohen, J., Rosenfeld, E., and Kolter, Z. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pp. 1310–1320. PMLR, 2019.

Domingos, P. Metacost: A general method for making classifiers cost-sensitive. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 155–164, 1999.

Elkan, C. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, pp. 973–978. Lawrence Erlbaum Associates Ltd, 2001.

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

Gowal, S., Dvijotham, K., Stanforth, R., Bunel, R., Qin, C., Uesato, J., Arandjelovic, R., Mann, T., and Kohli, P. On the effectiveness of interval bound propagation for training verifiably robust models. *arXiv preprint arXiv:1810.12715*, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Jia, J., Cao, X., Wang, B., and Gong, N. Z. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. *arXiv preprint arXiv:1912.09899*, 2019.

Khan, S. H., Hayat, M., Bennamoun, M., Sohel, F. A., and Togneri, R. Cost-sensitive learning of deep feature representations from imbalanced data. *IEEE transactions on neural networks and learning systems*, 29(8):3573–3587, 2017.

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

Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

LeCun, Y. et al. Lenet-5, convolutional neural networks. *URL: http://yann. lecun. com/exdb/lenet*, 20(5):14, 2015.

Lecuyer, M., Atlidakis, V., Geambasu, R., Hsu, D., and Jana, S. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Li, B., Chen, C., Wang, W., and Carin, L. Certified adversarial robustness with additive noise. *Advances in neural information processing systems*, 32, 2019.

Liu, X., Cheng, M., Zhang, H., and Hsieh, C.-J. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 369–385, 2018.

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

Raghunathan, A., Steinhardt, J., and Liang, P. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Bys4ob-Rb.

Shafahi, A., Najibi, M., Ghiasi, M. A., Xu, Z., Dickerson, J., Studer, C., Davis, L. S., Taylor, G., and Goldstein, T. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

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

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

Wong, E. and Kolter, Z. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International conference on machine learning*, pp. 5286–5295. PMLR, 2018.

Wong, E., Schmidt, F., Metzen, J. H., and Kolter, J. Z. Scaling provable adversarial defenses. *Advances in Neural Information Processing Systems*, 31, 2018.

Zhai, R., Dan, C., He, D., Zhang, H., Gong, B., Ravikumar, P., Hsieh, C.-J., and Wang, L. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

Zhang, X. and Evans, D. Cost-sensitive robustness against adversarial examples. *arXiv preprint arXiv:1810.09225*, 2018.

A. Related Work

Randomized smoothing (Cohen et al., 2019) proposes to first convert any base neural network into a smoothed classifier by injecting spherical Gaussian noises to inputs followed by majority voting, then provides a robust certificate that can guarantee the prediction of the resulting smoothed classifier is constant within some ℓ_2-norm ball for any given input. Compared with other robustness certification methods, the biggest advantage of the randomized smoothing framework is its scalability to large neural networks and large-scale datasets such as classification task for ImageNet. In particular, Cohen et al. (2019) provided a tight robustness guarantee for randomized smoothing with ℓ_2 perturbations. Later, SmoothADV (Shafahi et al., 2019) improved the proposed training method in Cohen et al. (2019) by designing an adaptive attack on the smoothed classifier using adversarial training and first-order approximations. In addition, MACER (Zhai et al., 2020) developed a more direct way which directly optimizes the smoothed classifiers' certified radius with respect to correctly-classified samples using margin based loss and achieves better robustness and accuracy trade-off than previous methods.

Cost-sensitive learning deals with the situation where different misclassifications will induce different costs (Domingos, 1999; Elkan, 2001). For example, misclassifying a malicious tumor to benign (Khan et al., 2017) will bring more harmful consequences to the patient than the reverse. In adversarial training, it's also valuable to make the classifier adapt to the cost-sensitive setting so that adversarial transformations with high costs will be less likely to happen. Most of the cost-sensitive robust training methods are either could only be employed on linear classifiers or are empirical training methods without any robust certification (Khan et al., 2017; Chen et al., 2021). Zhang & Evans (2018) proposed a method to train cost-sensitive certifiable classifiers using certified methods based on convex optimization, however, it can not scale to large neural network or large datasets such as ImageNet. Our work combines randomized smoothing and cost-sensitive learning to provide more scalable classifiers with good certifiable robustness guarantees under cost-sensitive scenarios.

B. Comparisons with Standard Certified Radius

The main distinction between our *cost-sensitive certified radius* $R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta)$ and Cohen et al. (2019)'s standard certified radius $R(\boldsymbol{x})$ lies in the pairwise cases. In the following, we provide the detailed proof of Theorem 3.3.

Proof of Theorem 3.3. To characterize the connection, we introduce an *intermediate cost-sensitive certified radius* $R(\boldsymbol{x};\Omega_y)$:

$$\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)=\frac{\sigma}{2}\cdot\left[\Phi^{-1}\big(\max_{j\in\mathcal{Y}\backslash\Omega_y}[h_\theta(\boldsymbol{x})]_j\big)-\Phi^{-1}\big(\max_{j\in\Omega_y}[h_\theta(\boldsymbol{x})]_j\big)\right].$$

Note that $R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta)$ and $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)$ are equivalent to each other under our evaluation metrics (Equation 4 and Equation 5). Thus, the relationship between $R_{\text{c-s}}(\boldsymbol{x};\Omega_y,h_\theta)$ and $R(\boldsymbol{x})$ translates to that between $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)$ and $R(\boldsymbol{x})$. For seed-wise cases, $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)$ and $R(\boldsymbol{x})$ are equivalent as the set of target classes in both definitions are identical, while for pairwise cases, their relationship can be explained according to the predicted results, note the prerequisite in the definition of $R(\boldsymbol{x})$ guarantees that $\boldsymbol{x}$ is correctly classified.

To be more specific, we have the following observations:

1. For seed-wise cost matrices, $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)>\epsilon\Leftrightarrow R(\boldsymbol{x})>\epsilon$. Since $\Omega_y=\{j|j\neq y,j\in[m]\}$ and $[m]\backslash\Omega_y=\{y\}$, the two probability terms are fully matched for both radius, so $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)$ degenerates to $R(\boldsymbol{x})$.

2. For pairwise cost matrices, $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)\geq R(\boldsymbol{x})$ for the second term in $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)$ is a relaxed version of $R(\boldsymbol{x})$, as $\Omega_y\in\{j\neq y,j\in[m]\}$, so $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)\geq R(\boldsymbol{x})$. If the prediction is ground-truth label y (which falls in $[m]\backslash\Omega_y$), then $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)\geq R(\boldsymbol{x})$. If the prediction is not the ground-truth label but is cost-sensitively correct, then $R(\boldsymbol{x})<0$ but $\tilde{R}(\boldsymbol{x};\Omega_y,h_\theta)>0$. If the prediction incurs a cost and falls in Ω_y, then all the values of three certified radius are smaller than 0, thus they are all the same.

Therefore, we complete the proof of Theorem 3.3. □

C. Details of Algorithm 1

In this section, we provide further details and discussions of Algorithm 1 for certifying cost-sensitive robustness presented in Section 3.2. We follow the same sampling procedure of Cohen et al. (Cohen et al., 2019). To be more specific, the sampling

Table 3. Comparison results of our method with convex relaxation based method (Zhang & Evans, 2018) for ℓ_2 perturbations on CIFAR-10, where a single pairwise cost-sensitive transformation ($3 \rightarrow 5$) is considered.

Method	ℓ_2 perturbations	Cost-sensitive robustness	Overall accuracy
Zhang & Evans (2018)	$\epsilon = 0.25$	0.944	0.480
Ours	$\epsilon = 0.5$	0.924	0.673

function SAMPLEUNDERNOISE(f, x, n, σ) is defined as:

1. Draw n i.i.d. samples of Gaussian noises $\boldsymbol{\delta}_1 \dots \boldsymbol{\delta}_n \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$.
2. Obtain the predictions $f(\boldsymbol{x} + \boldsymbol{\delta}_1), \dots, f(\boldsymbol{x} + \boldsymbol{\delta}_n)$ with base classifier f on noisy images.
3. Return the counts for each class, where the count for class c is $\sum_{i \in [n]} \mathbb{1}[f(\boldsymbol{x} + \boldsymbol{\delta}_i) = c]$.

We have two certification methods for pairwise cost matrices: Either Condition 1 or Condition 2 is sufficient to achieve our goal. We have two potential approaches: 1) similar to standard randomized smoothing, using one lower bound for seed class; 2) compute both lower and upper bounds.

1. Use the original randomized smoothing certification with one $\underline{p_A}$ (with respect to all classes $[m]$).
2. Compute both lower bound $\underline{p_A}$ (with respect to all classes $[m]$) and upper bound $\overline{p_B}$ (with respect to Ω_y), but the significance level needs to be set as $\alpha/2$ instead of α.

D. Comparisons with Zhang & Evans (2018)

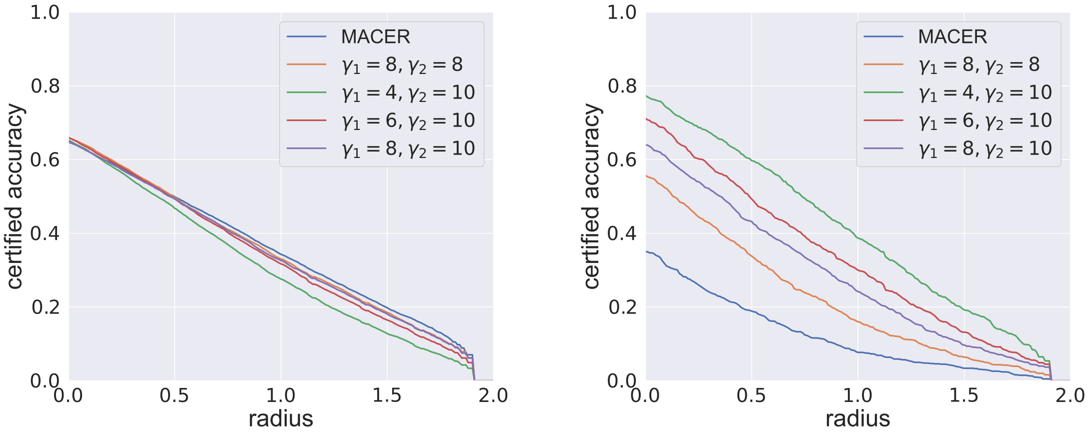
Zhang & Evans (2018) proposed a method to certify cost-sensitive robustness of any classifier based on convex relaxation (Wong & Kolter, 2018), which provides a robustness guarantee for a given input via minimizing the worst-case loss within the relaxed convex outer polytype. Also, Zhang & Evans (2018) developed a training method for training provably cost-senstive robust classifiers. In particular, their method incorporates different types of cost matrices into the convex optimization process to train cost-sensitive robust classifiers.

However, the initial work of Wong & Kolter (2018) only focuses on ℓ_∞-norm bounded perturbations and does not consider perturbations in ℓ_2-norm. As a result, the proposed method in Zhang & Evans (2018) also did not address the cost-sensitive robustness for ℓ_2 perturbations. We note that in a follow-up work of Wong et al. (2018), they extend the developed certification techinques to ℓ_2 perturbations. For fair comparisons with our method, we further extend the cost-senstive robust learning method of Zhang & Evans (2018) to handle ℓ_2-norm perturbations using the method of Wong et al. (2018). We report their comparisons in Table 3, the certified cost-sensitive robustness for the convex-relaxation method is computed as the *cost-sensitive robust error* defined in Zhang & Evans (2018), which represents the fraction of test samples that are guaranteed to be robust to certain ℓ_2 perturbations.

Table 3 shows that our method achieves much higher overall accuracy even against larger ℓ_2 perturbations, suggesting a better cost-sensitive robustness and overall accuracy trade-off. Also, we find in our implementation that convex relaxation-based methods is not applicable to large ℓ_2 perturbations (e.g., $\epsilon = 0.5$), due to memory issues. We also remark that randomized smoothing techniques proposed in existing works (Cohen et al., 2019; Li et al., 2019; Jia et al., 2019) primarily focus on defending against ℓ_2-perturbations. As a result, our methods excel in achieving good cost-sensitive performance under ℓ_2-norm bounded perturbations. There are limitations when it comes to certifying cost-sensitive robustness using our method under other types of perturbations, such as perturbations with ℓ_1-norm, ℓ_∞-norm and even beyond ℓ_p-norm.

E. Hyperparameter Tuning

In this section, we study the effect of hyperparameters γ_1 and γ_2 used in our method proposed in Section 4.2 on the two evaluation metrics, certified overall accuracy and cost-sensitive robustness. Note that our goal is to improve cost-sensitive robustness without sacrificing overall accuracy, where γ_1 controls the margin of normal classes and γ_2 controls the margin



(a) Overall

(b) Cost-Sensitive

Figure 4. Visualizations of our method with $\gamma_1 \in \{4, 6, 8\}$ and fixed $\gamma_2 = 10$ with comparisons to baseline methods with: (a) overall performance and (b) cost-sensitive performance. Here, we consider a single cost-sensitive seed class "Cat" for the cost matrix.

of sensitive classes. In particular, we report the parameter tuning results on CIFAR-10. Here, the cost matrix is selected as a seed-wise cost matrix with a sensitive seed class "cat". We choose the specific "cat" class only for the purpose of illustration, as we observe similar trends in our experiments for other cost matrices, similar to the results shown in Table 1.

In addition, we consider two comparison baselines:

1. MACER (Zhai et al., 2020) with $\gamma = 8$, restricting only on correctly classified examples.
2. Our method with $\gamma_1 = 8$ and $\gamma_2 = 8$, the only difference with MACER is that our method contains misclassified examples for sensitive classes.

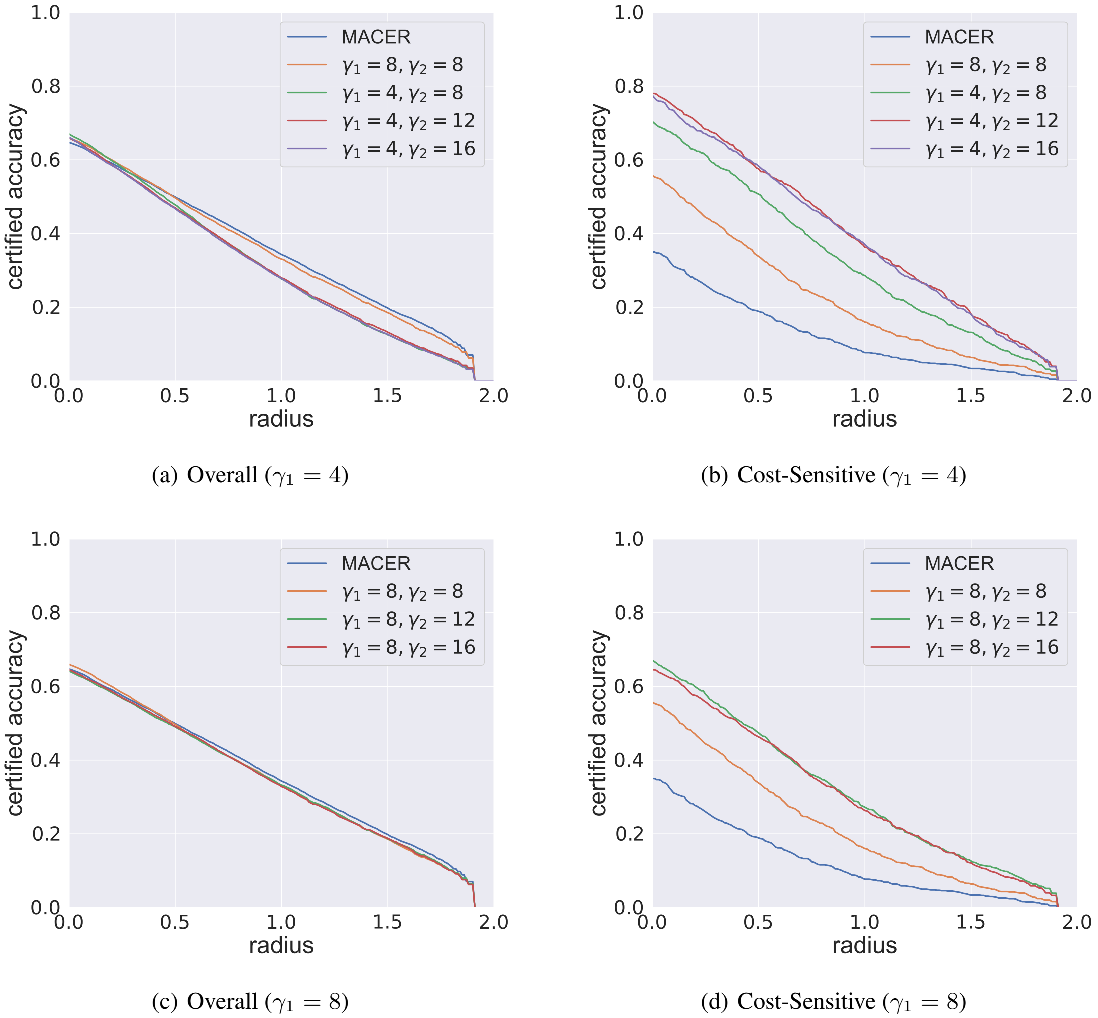
Below, we show the effect of γ_1 and γ_2 on the performance of our method, respectively.

Effect of γ_1. Note that γ_1 is used to restrict the certified radius with respect to normal data points. Figure 4 illustrates the influence of varying $\gamma_1 \in \{4, 6, 8\}$ and fixed $\gamma_2 = 10$ for our method, with comparisons to the two baselines, on both overall accuracy and cost-sensitive robustness.

For the original implementation of MACER, γ is selected as 8 for the best overall performance. Although it achieves good overall robustness, it does not work for cost-sensitive settings, which suggests the possibility of a trade-off space, where different classes can be balanced to achieve our desired goal of cost-sensitive robustness. The second baseline is our method with $\gamma_1 = 8$ and $\gamma_2 = 8$. By incorporating misclassified samples for sensitive seed class, the cost-sensitive performance substantially improvemes. This results shows the significance of including misclassified sensitive samples during the optimization process of the certified radius.

Moreover, we can observe from Figure 4(b) that as we reduce the value of γ_1, the robustness performance of the cost-sensitive seed class increases. This again confirms that by limiting the certified radius of normal classes to a small threshold in our method, the model can prioritize sensitive classes and enhance cost-sensitive robustness.

Effect of γ_2. Figure 5 illustrates the influence of varying $\gamma_2 \in \{8, 12, 16\}$ with fixed $\gamma_1 = 4$ or fixed $\gamma_1 = 8$ for our method, with comparisons to the two baselines, on both overall accuracy and cost-sensitive robustness. Moreover, we can observe from Figure 5(b) and Figure 5(d) that as we increase the value of γ_2, the robustness performance of the cost-sensitive seed class increases. This confirms that by optimizing the certified radius of sensitive classes to a large threshold in our method, the model can focus more on sensitive classes and enhance cost-sensitive robustness. Additionally, there is a slight increase in the overall certified accuracy. This can be attributed to the fact that the overall accuracy takes into account both the accuracy of sensitive samples and normal samples. As the certified accuracy of sensitive samples increases, it dominates the overall accuracy and leads to its overall improvement.



(a) Overall ($\gamma_1 = 4$)

(b) Cost-Sensitive ($\gamma_1 = 4$)

(c) Overall ($\gamma_1 = 8$)

(d) Cost-Sensitive ($\gamma_1 = 8$)

Figure 5. Visualizations of our method for two groups comparisons to baseline methods in terms of (a)(c) overall performance and (b)(d) cost-sensitive performance. The first with $\gamma_2 \in \{8, 12, 16\}$ and fixed $\gamma_1 = 4$, the second with $\gamma_2 \in \{8, 12, 16\}$ and fixed $\gamma_1 = 8$. The cost matrix is set as the matrix representing a single cost-sensitive seed class “Cat”.

Table 4 demonstrates the impact of different combinations of hyperparameters of (γ_1, γ_2) on both the overall accuracy and cost-sensitive performance. The choice of γ_1 and γ_2 is crucial and requires careful consideration. For γ_2, setting a value that is too small can greatly undermine the overall accuracy, even though it may improve cost-sensitive robustness. This is because the performance of normal classes deteriorates, resulting in a degradation of overall performance. On the other hand, if the value is too large such as $\gamma_2 = 8$, it may have a negative impact on cost-sensitive performance.

Regarding γ_1, it is evident that increasing its value while keeping γ_2 fixed leads to a significant improvement in cost-sensitive robustness. It is worth noting that even though the cost-sensitive seed class represents only a single seed, accounting for only 10% of the total classes, enhancing its robustness has a positive effect on overall accuracy as well. For instance, let’s compare the combination $(\gamma_1 = 8, \gamma_2 = 4)$ to $(\gamma_1 = 8, \gamma_2 = 8)$. We observe that the former, which exhibits better cost-sensitive robustness, outperforms the latter in terms of both overall accuracy and cost-sensitive robustness. It achieves an approximate improvement of 1.52% in overall accuracy and a significant improvement of approximately 50% in cost-sensitive robustness.

This finding highlights the effectiveness of our sub-population-based methods. It demonstrates that by fine-tuning the optimization thresholds for the certified radius of sensitive classes and normal classes separately, we can achieve a better trade-off between overall accuracy and cost-sensitive robustness.

Table 4. Performance of our method for different parameter combinations of γ_1 and γ_2. Here, the cost-sensitive scenario is captured by the seed-wise cost matrix with a single sensitive seed class "Cat" for CIFAR-10.

Method	sensitive	normal	Overall accuracy	Cost-Sensitive robustness
MACER	-	-	0.647	0.189
Ours	8	8	0.660	0.338
Ours	8	2	0.654	0.633
	10	2	0.634	0.687
	12	2	0.637	0.691
	16	2	0.630	0.705
Ours	8	4	0.670	0.507
	10	4	0.653	0.597
	12	4	0.659	0.576
	16	4	0.661	0.583
Ours	8	6	0.673	0.396
	10	6	0.660	0.493
	12	6	0.655	0.544
	16	6	0.649	0.552
Ours	8	8	0.660	0.338
	10	8	0.650	0.432
	12	8	0.641	0.474
	16	8	0.645	0.463