A. Training Procedure of CAT-RS

Algorithm 1 Confidence-aware Training for Randomized Smoothing (CAT-RS)

```
Require: training sample $(x, y)$. smoothing factor $\sigma$. number of noise samples $M > 0$. consistency targets $\hat{y} \in \Delta^{K-1}$,
    regularization strength $\lambda > 0$. attack norm $\varepsilon > 0$.
Sample $\delta_1, \cdots, \delta_M \sim \mathcal{N}(0, \sigma^2 I)$
$\hat{p}_f \leftarrow \frac{1}{M} \sum_i \mathbb{1}[f(x + \delta_i) = y]$
Sample $K \sim \mathrm{Bin}(M, \hat{p}_f), K^+ \leftarrow \max(1, K)$
for $i = 1$ to $M$ do
    $L_i \leftarrow \mathbb{CE}(F(x + \delta_i), y)$
    $\delta_i^* \leftarrow \arg\max_{\|\delta_i^* - \delta_i\| \leq \varepsilon} \mathrm{KL}(F(x + \delta_i^*), \hat{y})$
end for
$L_{1:M}^{\pi} \leftarrow \texttt{argsort}(L_{1:M})$
$L^{\texttt{low}}, L^{\texttt{high}} \leftarrow \frac{1}{M}(\sum_{i=1}^{K^+} L_i^{\pi}),\ \max_i \mathrm{KL}(F(x + \delta_i^*), \hat{y})$
$L^{\texttt{CAT-RS}} \leftarrow L^{\texttt{low}} + \lambda \cdot \mathbb{1}[K^+ = M] \cdot L^{\texttt{high}}$
```

B. Related Work

There have been continual attempts to provide a certificate on robustness of deep neural networks against adversarial attacks [9,10,27,40,44,48], and correspondingly to further improve the robustness with respect to those certification protocols [2, 6, 7]. *Randomized smoothing* [5] has attracted a particular attention among them, due to its scalability to large datasets, *e.g.*, ImageNet [31], and its flexibility to various applications [8, 30, 34, 38, 42] or other threat models [17, 22, 25, 32, 45, 47]. A more extensive survey on certified robustness can be found in [24].

This work aims to improve adversarial robustness of randomized smoothing, along a line of research on designing training schemes specialized for smoothed classifiers [15,16,33,46]. Specifically, we focus on the relationship between confidence and robustness of smoothed classifiers, a property rarely investigated previously but few [15, 19]: *e.g.*, [19] extends randomized smoothing to also provide certificates on confidences, and [15] exploits over-confident adversarial examples to improve smoothed classifiers. We leverage the property to overcome challenges in estimating sample-wise robustness, and to develop a data-dependent adversarial training which has been also challenging even for empirical robustness [39, 52].

Comparison to SmoothAdv. The idea of incorporating an adversarial search for the robustness of smoothed classifiers has been also considered in previous works [15, 33]: *e.g.*, [33] have proposed *SmoothAdv* that applies adversarial training [26] to a "soft" approximation of $\hat{f}$ given f and M noise samples:

$$x^* = \underset{||x'-x||_2 \leq \epsilon}{\arg\max} \left(-\log \left(\frac{1}{M} \sum_i F_y(x' + \delta_i) \right) \right). \tag{9}$$

Our method is different from the previous approaches in which part of the inputs is adversarially optimized: *i.e.*, we directly optimize the noise samples δ_i's instead of x, with no need to assume a soft relaxation of $\hat{f}$. This is due to our unique motivation of finding the worst-case Gaussian noise, and our experiments in Section 4 further support the effectiveness.

C. Experimental Details

We follow the training setup considered in most of the previous works to compare the performance of the smoothed classifiers [5, 15, 16, 46]: specifically, we mainly consider LeNet [20], ResNet-110 [11], and ResNet-50 for MNIST/Fashion-MNIST, CIFAR-10/100, and ImageNet, respectively, and consider different scenarios of $\sigma \in \{0.25, 0.5, 1.0\}$ for randomized smoothing. We apply the same σ for both training and evaluation. When training, we use stochastic gradient descent (SGD) optimizer with a momentum of 0.9, and weight decay of 10^{-4}. The learning rate is initialized to 0.01 for MNIST/Fashion-MNIST and 0.1 for CIFAR-10/100, and decreased by a factor of 0.1 for every 50 epochs. For ImageNet, we train ResNet-50 [11] for 90 epochs, with initial learning rate of 0.1 decreased by a factor of 0.1 for every 30 epochs.

C.1. Datasets

MNIST [20] consists of 70,000 gray-scale hand-written digit images of size 28×28, 60,000 for training and 10,000 for testing, where each is labeled to one value between 0 and 9. We do not perform any pre-processing except for normalizing the range of each pixel from 0-255 to 0-1. The dataset can be downloaded at `http://yann.lecun.com/exdb/mnist/`.

Fashion-MNIST [43] consists of 70,000 gray-scale 10-category fashion product images of size 28 × 28, 60,000 for training and 10,000 for testing. Each category is assigned to one value between 0 and 9, where each image is labeled to the value assigned to its category. We do not perform any pre-processing except for normalizing the range of each pixel from 0-255 to 0-1. The dataset can be downloaded at `https://github.com/zalandoresearch/fashion-mnist`.

CIFAR-10/100 [18] consists of 60,000 RGB images of size 32×32, 50,000 for training and 10,000 for testing, where each is labeled to one of 10 and 100 classes, repsectively. We use the standard data-augmentation scheme of random horizontal flip and random translation up to 4 pixels, following the practice of other baselines [5, 15, 16, 33, 46]. We also normalize the images in pixel-wise by the mean and the standard deviation calculated from the training set. The full dataset can be downloaded at `https://www.cs.toronto.edu/~kriz/cifar.html`.

ImageNet [31] consists of 1,281,167 images for training, and 50,000 images for validation. Each of the images are labeled to one of 1,000 classes. We perform 224×224 randomly resized cropping and horizontal flipping for the training images. For test images, we resize the images into 256×256 resolution, followed by 224×224 center cropping. The full dataset can be downloaded at `https://image-net.org/download`.

C.2. Baselines

We compare our method with an extensive list of baseline methods in the literature of training smoothed classifiers: (a) *Gaussian training* [5] simply trains a classifier with Gaussian augmentation (5); (b) *Stability training* [23] adds a cross-entropy term between the logits from clean and noisy images; (c) *SmoothAdv* [33] employs adversarial training for smoothed classifiers (9); (d) *MACER* [46] adds a regularization that aims to maximize a soft approximation of certified radius; (e) *Consistency* [16] regularizes the variance of confidences over Gaussian noise; (f) *SmoothMix* [15] proposes a mixup-based [49] adversarial training for smoothed classifiers. Whenever possible, we use the pre-trained models publicly released by the authors to reproduce the results.

C.3. Evaluation Metrics

We follow the standard evaluation protocol for smoothed classifiers [15,16,33,46]: specifically, [5] has proposed a practical Monte-Carlo-based certification procedure, namely CERTIFY, that returns the prediction of $\hat{f}$ and a lower bound of certified radius, $\mathrm{CR}(f, \sigma, x)$, over the randomness of n samples with probability at least $1 - \alpha$, or abstains the certification. Based on CERTIFY, we consider two major evaluation metrics: (a) the *average certified radius* (ACR) [46]: the average of certified radii on the test dataset $\mathcal{D}_{\texttt{test}}$ while assigning incorrect samples as 0, namely $\mathrm{ACR} := \frac{1}{|\mathcal{D}_{\texttt{test}}|}\sum_{(x,y)\in\mathcal{D}_{\texttt{test}}}[\mathrm{CR}(f, \sigma, x)\cdot \mathbb{1}_{\hat{f}(x)=y}]$, and (b) the *approximate certified test accuracy* at r: the fraction of the test dataset which CERTIFY classifies correctly with the radius larger than r without abstaining. We use $n = 100,000$, $n_0 = 100$, and $\alpha = 0.001$ for CERTIFY, following the previous works [5, 15, 16, 33].

C.4. Implementation Details

Bottom-K Gaussian loss. Although it is well-defined, the basic form of the bottom-K loss given in (6) may not handle the *cold-start* problem on $p_f(x, y)$, *e.g.*, at the early stage of the training where $x + \delta$ has not been adequately exposed to f, so that it is uncertain whether the current $p_f(x, y)$ is optimal: in this case, $L^{\texttt{low}}$ can be minimized with an under-estimated $p_f \approx 0$, potentially with samples those never optimize the cross-entropy losses during training. Nevertheless, we found that a simple workaround of *clamping* K can effectively handle the issue, *i.e.*, by using $K^+ \leftarrow \max(K, 1)$ instead of K: in other words, we always allow the "easiest" noise among the M samples to be fed into f throughout the training.

Worst-case Gaussian loss. In practice, we use the *projected gradient descent* (PGD) [26] to solve the inner maximization in (7): namely, we perform a T-step gradient ascent from each δ_i with step size $2 \cdot \varepsilon / T$ while projecting the perturbations to be in the ℓ_2-ball of size ε. This procedure would find a noise δ^* that maximizes the loss around x, while maintaining the Gaussian-like noise appearance due to the projected search in a small ε-ball. In order to further make sure that the Gaussian likelihood of δ^* is maintained from the original δ, we additionally apply a simple trick of *normalizing* the mean and standard deviation of δ^* to follow those of δ.

C.5. Hyperparameters

Stability training [23] introduces a single hyperparameter γ to control the relative strength of the regularization for the logits under Gaussian augmentation. We fix $\gamma = 2$ for MNIST/Fashion-MNIST experiments. For CIFAR-10/100 experiments, $\gamma = 2$ is used for $\sigma = 0.25, 0.5$, and $\gamma = 1$ is used for $\sigma = 1.0$.

SmoothAdv [33] uses three major hyperparameters to perform the projected gradient descent: namely, the attack radius in terms of ℓ_2-norm ε, the number of PGD steps T, and the number of noises m. In our experiments, we fix $T = 10$. For MNIST/Fashion-MNIST experiments, we fix $\varepsilon = 1.0$ and $m = 4$ as well. In case of CIFAR-10/100, on the other hand, we report the results chosen among the list of "best" configurations for each noise level which are previously searched by [33]: specifically, we report the results of $\varepsilon = 1.0$ and $m = 4$ for $\sigma = 0.25$, and $\varepsilon = 1.0$ and $m = 8$ for $\sigma = 0.5$, and $\varepsilon = 2.0$ and $m = 2$ for $\sigma = 1.0$. When SmoothAdv is used, we adopt the *warm-up* strategy, *i.e.*, we initially set $\varepsilon = 0.0$ and linearly increase to the target value of ε for 10-epochs.

MACER [46] introduces four hyperparameters: the number of noises k, the coefficient for the regularization term λ, the clamping parameter for maximizing the certified radius γ, and the temperature scaling parameter β. For the MNIST experiments, we use $k = 16, \gamma = 8.0, \beta = 16.0$, and $\lambda = 16.0$ when $\sigma = 0.25, 0.5$, following the configurations in [46]. For $\sigma = 1.0$, we had to reduce $\lambda = 6.0$ for a stable training. For the Fashion-MNIST experiments, we follow all the hyperparameters of the MNIST experiments except λ. Due to the stability issue for training, we had to set $\lambda = 8.0$ and $\lambda = 2.0$ for $\sigma = 0.5$ and $\sigma = 1.0$, respectively. For the CIFAR-10/100 experiments, we follow the original configurations used by [46]. We set $k = 16, \gamma = 8.0$, and $\beta = 16.0$. λ is set to be 12.0 and 4.0 for $\sigma = 0.25$ and 0.5, respectively. For $\sigma = 1.0$, the training starts with $\lambda = 0$ until the first learning rate decay and we set $\lambda = 12.0$ thereafter.

Consistency [16] uses two hyperparameters: namely, the coefficient for the consistency term η and the entropy term γ. We report the best results in terms of ACR among those reported by [16] varying η. Following the original practice, we fix $\gamma = 0.5$ throughout our experiments. For MNIST/Fashion-MNIST, we use $\lambda = 10$ for $\sigma = 0.25$ and $\lambda = 5$ for other noises. For the CIFAR-10/100 experiments, we use $\lambda = 20$ for $\sigma = 0.25$ and $\lambda = 10$ for other noises.

SmoothMix [15] introduces four hyperparameters: namely, the mixup coefficient between the original and adversarial sample η, the step size for adversarial attack α, the number of steps for adversarial attack T, and the number of noises m. For the MNIST/Fashion-MNIST experiments, we fix $\eta = 5.0, \alpha = 1.0$, and $m = 4$. $T = 2, 4, 8$ are used for the models with $\sigma = 0.25, 0.5, 1.0$, respectively. For the CIFAR-10/100 experiments, we again report the best result among those reported from [15]: *i.e.*, we fix $\eta = 5.0, m = 2$, and $T = 4$, and use $\alpha = 0.5, 1.0, 2.0$ for $\sigma = 0.25, 0.5, 1.0$, respectively. The "one-step adversary" is used for $\sigma = 0.5, 1.0$ to follow the best configurations reported.

CAT-RS (Ours). We introduce one main hyperparameter: namely, the coefficient λ for the worst-case loss. Although the number of noises M, the number of attack steps T, and the attack radius ε are also can be tuned for a better performance, we fix $M = 4$, $T = 4$, and $\varepsilon = 1.0$ unless otherwise noted. For the MNIST/Fashion-MNIST experiments, we use the fixed configuration of $\lambda = 1.0$. For the CIFAR-10/100 experiments, we use $\lambda = 0.5, 1.0$, and 2.0 for $\sigma = 0.25, 0.5$, and 1.0. For the ImageNet experiments, we use $\lambda = 2.0$. Also, we set $M = 2$ and $T = 1$ to reduce the overall training cost.

For each training sample x, we compute its soft-label $\hat{y}$ for (7) by the *smoothed prediction* of another classifier $\bar{f}$ pre-trained via Gaussian training (5) with a fixed $\sigma_0 = 0.25$: specifically, we obtain a soft-label $\hat{y} \in \mathbb{R}^K$ by computing:

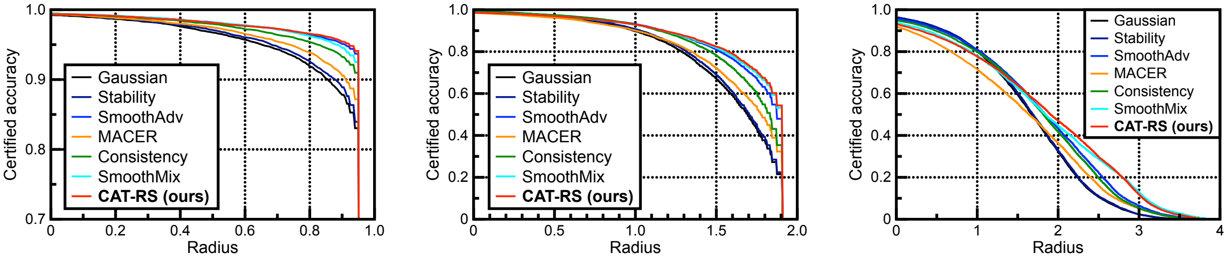
$$\hat{y}_c := \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\bar{f}(x + \delta_i) = c], \tag{10}$$

where $\delta_i \sim \mathcal{N}(0, \sigma_0^2 I)$. In our experiments, we use $N = 10,000$ Gaussian noises for MNIST/Fashion-MNIST and CIFAR-10/100, and $N = 500$ for ImageNet.

D. Results on More Datasets

D.1. MNIST

We compare the certified robustness of the smoothed classifiers trained on MNIST from our method to those from other baselines in Table 3, considering three different smoothing factors $\sigma \in \{0.25, 0.5, 1.0\}$. We also present in Figure 2 the plots of the approximate certified accuracy across varying r. Overall, the results show that CAT-RS clearly surpasses all the other baselines in terms of ACR: *i.e.*, our method could better balance between the clean accuracy and robustness. For $\sigma = 0.25$,



(a) $\sigma = 0.25$

(b) $\sigma = 0.50$

(c) $\sigma = 1.00$

Figure 2. Comparison of approximate certified accuracy for various training methods on MNIST. The sharp drop of certified accuracy in each plot is due to a strict upper bound in radius that CERTIFY can output for a given σ, $N = 100,000$, and $\alpha = 0.001$.

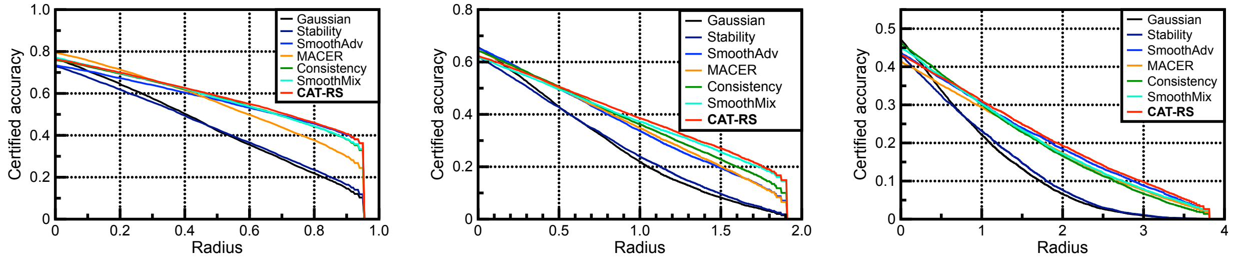
Table 3. Comparison of ACR and approximate certified test accuracy (%) on MNIST. For each column, we set our result bold-faced whenever the value improves the Gaussian baseline. We mark the highest and lowest values of certified accuracy at each radius in blue and red colors, respectively.

σ	Methods	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
0.25	Gaussian [5]	0.910	99.2	98.5	96.7	93.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Stability [23]	0.914	99.3	98.6	97.1	93.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothAdv [33]	0.932	99.4	99.0	98.2	96.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MACER [46]	0.921	99.3	98.7	97.5	94.8	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Consistency [16]	0.928	99.5	98.9	98.0	96.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothMix [15]	0.932	99.4	99.0	98.2	96.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CAT-RS (Ours)	**0.933**	**99.4**	**99.0**	**98.2**	**96.9**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.50	Gaussian [5]	1.557	99.2	98.3	96.8	94.3	89.7	81.9	67.3	43.6	0.0	0.0	0.0
	Stability [23]	1.573	99.2	98.5	97.1	94.8	90.7	83.2	69.2	45.4	0.0	0.0	0.0
	SmoothAdv [33]	1.687	99.0	98.3	97.3	95.8	93.2	88.5	81.1	67.5	0.0	0.0	0.0
	MACER [46]	1.583	98.5	97.5	96.2	93.7	90.0	83.7	72.2	54.0	0.0	0.0	0.0
	Consistency [16]	1.655	99.2	98.6	97.6	95.9	93.0	87.8	78.5	60.5	0.0	0.0	0.0
	SmoothMix [15]	1.694	98.7	98.0	97.0	95.3	92.7	88.5	81.8	70.0	0.0	0.0	0.0
	CAT-RS (Ours)	**1.700**	98.6	98.0	**97.0**	**95.4**	**92.8**	**88.7**	**82.5**	**71.1**	0.0	0.0	0.0
1.00	Gaussian [5]	1.619	96.3	94.4	91.4	86.8	79.8	70.9	59.4	46.2	32.5	19.7	10.9
	Stability [23]	1.636	96.5	94.6	91.6	87.2	80.7	71.7	60.5	47.0	33.4	20.6	11.2
	SmoothAdv [33]	1.779	95.8	93.9	90.6	86.5	80.8	73.7	64.6	53.9	43.3	32.8	22.2
	MACER [46]	1.598	91.6	88.1	83.5	77.7	71.1	63.7	55.7	46.8	38.4	29.2	20.0
	Consistency [16]	1.738	95.0	93.0	89.7	85.4	79.7	72.7	63.6	53.0	41.7	30.8	20.3
	SmoothMix [15]	1.820	93.7	91.6	88.1	83.5	77.9	70.9	62.7	53.8	44.8	36.6	28.9
	CAT-RS (Ours)	**1.831**	93.2	90.5	87.2	83.1	77.6	**71.7**	**64.0**	**55.8**	**47.2**	**39.2**	**30.0**

we notice that some baselines, *i.e.*, SmoothAdv and SmoothMix, already achieve a reasonably saturated level of ACR: even in this trivial task, our method could further push the boundary of robust accuracies. In more challenging cases of $\sigma = 0.5$ and $\sigma = 1.0$, on the other hand, the improvements from CAT-RS in ACR become more evident as σ increases: *e.g.*, at $\sigma = 1.0$, compared to SmoothMix (the best-performing baseline), CAT-RS could improve the certified accuracy at $r = 2.50$ by $28.9\% \to 30.0\%$, resulting in ACR increment by $1.820 \to 1.831$. This means that our proposed CAT-RS can be more effective at challenging tasks, where it is more likely that a given classifier gets a more diverse confidence distribution for the training samples, so that our proposed confidence-aware training can better play its role.

D.2. Fashion-MNIST

In this section, we compare the performance on Fashion-MNIST dataset [43]. Table 4 shows ACR and certified accuracy varying the severity of noise level $\sigma \in \{0.25, 0.50, 1.00\}$. Overall, CAT-RS offers a better trade-off between accuracy and robustness improving ACR compared to the baselines. We highlight that our method is more effective in challenging setting, *e.g.*, $\sigma = 1.0$, where leveraging confidence information is critical. For instance, CAT-RS improves the certified accuracy at



(a) $\sigma = 0.25$

(b) $\sigma = 0.50$

(c) $\sigma = 1.00$

Figure 3. Comparison of approximate certified accuracy for various training methods on CIFAR-10. The sharp drop of certified accuracy in each plot is due to a strict upper bound in radius that CERTIFY can output for a given σ, $N = 100,000$, and $\alpha = 0.001$.

Table 4. Comparison of ACR and approximate certified test accuracy (%) on Fashion-MNIST. For each column, we set our result bold-faced whenever it improves the Gaussian baseline. We set our result underlined if it achieves the highest among the baselines.

σ	Methods	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75	2.00	2.25	2.50
0.25	Gaussian [5]	0.670	89.5	82.0	70.8	57.7	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Stability [23]	0.689	89.2	83.2	73.2	60.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothAdv [33]	0.756	86.2	83.3	79.8	75.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	MACER [46]	0.727	88.1	84.2	77.8	68.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	Consistency [16]	0.744	88.5	84.7	78.8	71.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	SmoothMix [15]	0.745	88.8	84.6	78.9	71.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	CAT-RS (Ours)	**0.757**	86.3	**83.5**	**79.6**	**75.2**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
0.50	Gaussian [5]	1.056	86.2	80.7	73.2	64.8	55.5	45.6	35.0	24.1	0.0	0.0	0.0
	Stability [23]	1.118	85.9	81.6	75.8	68.8	60.2	50.5	39.4	27.6	0.0	0.0	0.0
	SmoothAdv [33]	1.255	83.3	80.2	76.5	71.9	66.7	61.2	54.5	45.9	0.0	0.0	0.0
	MACER [46]	1.183	83.3	80.1	75.9	70.4	64.2	56.7	47.7	36.0	0.0	0.0	0.0
	Consistency [16]	1.212	84.9	81.1	76.4	71.2	65.2	57.8	49.3	39.2	0.0	0.0	0.0
	SmoothMix [15]	1.237	84.4	80.7	76.3	71.2	65.6	58.9	52.4	44.2	0.0	0.0	0.0
	CAT-RS (Ours)	**1.274**	82.5	79.6	**76.2**	**72.4**	**67.8**	**62.5**	**56.7**	**49.0**	0.0	0.0	0.0
1.00	Gaussian [5]	1.316	79.0	74.3	68.6	62.5	56.2	50.0	43.1	36.4	29.2	23.1	17.5
	Stability [23]	1.394	78.1	74.4	70.2	65.5	59.4	53.3	46.4	39.9	32.8	26.2	19.6
	SmoothAdv [33]	1.538	77.0	73.7	69.6	65.5	61.3	56.3	50.9	45.5	39.1	32.6	26.9
	MACER [46]	1.504	74.1	71.2	67.6	63.9	60.2	55.7	50.6	45.5	39.5	33.4	27.4
	Consistency [16]	1.491	75.5	72.4	68.4	64.5	59.8	54.8	49.4	44.0	37.9	31.7	25.7
	SmoothMix [15]	1.534	76.4	72.6	68.3	63.3	58.4	53.7	48.6	43.4	38.4	33.3	28.3
	CAT-RS (Ours)	**1.607**	73.8	71.1	68.0	**64.9**	**61.1**	**57.3**	**52.9**	**48.0**	**43.2**	**37.4**	**31.7**

$r = 2.50$ by $28.3\% \rightarrow 31.7\%$, resulting in the increment of ACR by $1.534 \rightarrow 1.607$. It confirms that confidence-aware training can effectively boost the robustness when smoothed via randomized smoothing.

D.3. CIFAR-100

Table 5 shows the results for $\sigma \in \{0.25, 0.50\}$[8] on CIFAR-100 [18] dataset. Still, CAT-RS achieves the best ACR by boosting the robustness of the smoothed classifier. Especially, CAT-RS improves the certified accuracy over the whole range of radii while keeping the certified accuracy at $r = 0.00$ comparable to other methods. For example, compared to SmoothMix for $\sigma = 0.50$, CAT-RS achieves higher accuracy at $r = 0.00$ by $34.0\% \rightarrow 35.4\%$ as well as at $r = 1.75$ by $8.2\% \rightarrow 9.0\%$, resulting in the ACR improvement by $0.352 \rightarrow 0.372$. This result suggests that our confidence-aware training effectively plays its role.

[8] We omit the results for $\sigma = 1.0$ as all methods achieve low clean accuracy of $\sim 20\%$, which is less meaningful.

Table 5. Comparison of ACR and approximate certified test accuracy (%) on CIFAR-100. For each column, we set our result bold-faced whenever it improves the Gaussian baseline. We set our result underlined if it achieves the highest among the baselines.

σ	Methods	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75
0.25	Gaussian [5]	0.228	48.9	33.7	20.9	12.0	0.0	0.0	0.0	0.0
	Stability [23]	0.159	34.3	23.4	14.5	7.8	0.0	0.0	0.0	0.0
	SmoothAdv [33]	0.298	46.4	38.3	30.4	23.0	0.0	0.0	0.0	0.0
	MACER [46]	0.283	51.1	39.5	28.1	18.1	0.0	0.0	0.0	0.0
	Consistency [16]	0.263	39.3	33.1	26.9	21.0	0.0	0.0	0.0	0.0
	SmoothMix [15]	0.295	49.9	39.5	29.5	20.8	0.0	0.0	0.0	0.0
	CAT-RS (Ours)	**0.312**	48.2	**39.8**	**31.7**	**24.4**	0.0	0.0	0.0	0.0
0.50	Gaussian [5]	0.259	36.5	27.8	20.4	14.7	10.1	6.8	4.2	2.3
	Stability [23]	0.078	8.6	7.2	5.9	4.6	3.7	2.6	1.9	1.2
	SmoothAdv [33]	0.342	36.7	30.5	24.9	19.9	15.8	12.0	9.1	6.3
	MACER [46]	0.314	37.8	29.7	23.4	18.2	14.0	10.3	7.3	4.7
	Consistency [16]	0.275	24.3	21.4	18.5	16.1	13.8	11.7	9.3	7.0
	SmoothMix [15]	0.352	34.0	29.1	24.6	20.3	16.9	13.9	11.0	8.2
	CAT-RS (Ours)	**0.368**	35.8	**30.5**	**25.7**	**21.2**	**17.5**	**14.4**	**11.5**	**8.6**

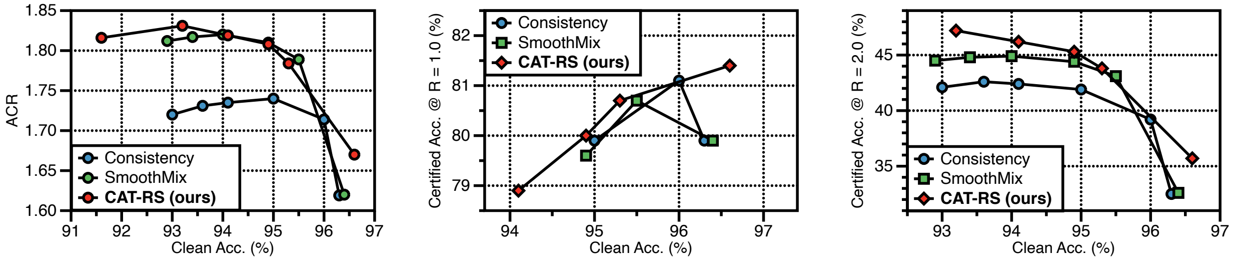
D.4. ImageNet

Table 6. Comparison of ACR and approximate certified test accuracy (%) on ImageNet. For each column, we set our result bold-faced whenever it improves the Gaussian baseline. We set our result underlined if it achieves the highest among the baselines.

Methods	ACR	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5
Gaussian [5]	0.875	44	38	33	26	19	15	12	9
Consistency [16]	0.982	41	37	32	28	24	21	17	14
SmoothAdv [33]	1.040	40	37	34	30	27	25	20	15
SmoothMix [15]	1.047	40	37	34	30	26	24	20	17
CAT-RS (Ours)	**1.071**	**44**	**38**	**35**	**31**	**27**	**24**	**20**	**17**

In this section, we compare the certified robustness of our method on ImageNet [31] dataset for $\sigma = 1.0$. We evaluate the performance on the uniformly-subsampled 500 samples in the ImageNet validation dataset following [5, 15, 16, 33]. We train ResNet-50 [11] for 90 epochs, with the initial learning rate of 0.1 decreased by a factor of 0.1 in every 30 epochs, as well as by a factor of 0.1 for the last 5 epochs. For CAT-RS training, we use $\varepsilon = 1.0$ for the 80 epochs of training, and increase it to $\varepsilon = 2.0$ for the last 10 epochs. Also, to further alleviate the cold-start problem in (6) under many-class ImageNet, we assume $K \sim \mathrm{Bin}(M, \hat{y}_c)$ instead of $K \sim \mathrm{Bin}(M, \hat{p}_f(x, y))$ so that the training can avoid binomial sampling from $\hat{p}_f(x, y) \approx 1/C$ for the early stage of training. The results shown in Table 6 confirm that our method achieves better results in terms of ACR and certified test accuracy compared to the baselines considered, verifying the effectiveness of CAT-RS even in the large-scale dataset.

E. Comparison of Accuracy-Robustness Trade-off



(a) ACR

(b) Certified Acc., $r = 1.0$

(c) Certified Acc., $r = 2.0$

Figure 4. Comparison of the trends between the clean accuracy *vs.* (a) ACR, (b) the certified accuracy at $r = 1.0$, and (c) at $r = 2.0$, that each method exhibits as varying its hyperparameter. We assume MNIST dataset with $\sigma = 1.0$ for this experiment.

Table 7. Comparison of ACR and approximate certified test accuracy on MNIST for varying hyperparameters of three different methods: Consistency, SmoothMix, and CAT-RS (ours). We assume $\sigma = 1.0$ in this experiment. "Gaussian" indicates the baseline Gaussian training. Consistency and SmoothMix degenerates to Gaussian when their hyperparameter is set to 0.

Methods	Setups	ACR	0.00	0.50	1.00	1.50	2.00	2.50
Gaussian	-	1.620	96.4	91.4	79.9	59.6	32.6	10.8
Consistency	$\lambda = 1$	1.714	96.0	91.2	81.1	63.5	39.2	16.2
	$\lambda = 5$	1.740	95.0	89.7	79.9	63.7	41.9	20.0
	$\lambda = 10$	1.735	94.1	88.6	78.5	62.8	42.4	22.1
	$\lambda = 15$	1.731	93.6	87.7	77.8	62.3	42.6	22.9
	$\lambda = 20$	1.720	93.0	86.6	77.1	61.6	42.1	23.4
	$\lambda = 25$	1.226	73.2	64.4	53.9	42.4	27.4	14.5
SmoothMix	$\eta = 1$	1.789	95.5	90.5	80.7	64.1	43.1	24.1
	$\eta = 2$	1.810	94.9	89.7	79.6	63.8	44.4	26.6
	$\eta = 4$	1.820	94.0	88.4	78.3	63.0	44.9	28.7
	$\eta = 8$	1.817	93.4	87.5	77.3	62.4	44.8	29.3
	$\eta = 16$	1.812	92.9	86.7	76.6	61.8	44.5	29.6
CAT-RS (Ours)	$\lambda = 0.00$	1.670	96.6	91.8	81.4	62.4	35.7	12.2
	$\lambda = 0.12$	1.784	95.3	90.2	80.7	64.7	43.8	23.4
	$\lambda = 0.25$	1.808	94.9	89.6	80.0	64.9	45.3	26.0
	$\lambda = 0.50$	1.819	94.1	88.4	78.9	64.6	46.2	28.1
	$\lambda = 1.00$	1.831	93.2	87.2	77.6	64.0	47.2	30.0
	$\lambda = 2.00$	1.816	91.6	85.0	75.7	62.9	48.0	31.5
	$\lambda = 4.00$	1.777	87.2	80.1	71.6	61.7	48.4	33.4

F. Ablation Study

We conduct an ablation study to further analyze individual effectiveness of the design components in our method. Unless otherwise noted, we use ResNet-20 [11] and test it on the uniformly subsampled CIFAR-10 test set of size 1,000.

Effect of λ. In CAT-RS (8), λ controls the relative contribution of $L^{\mathtt{high}}$ over $L^{\mathtt{low}}$. Here, Figure 6(a) shows the impact of λ to the model on varying $\lambda \in \{0.25, 0.5, 1.0, 2.0, 4.0\}$, assuming $\sigma = 0.5$. The results confirm that λ successfully balances the trade-off between robustness and clean accuracy [50]. In addition, Figure 4 in Appendix E verifies that CAT-RS offers more effective trade-off compared to other baseline training methods.

Effect of M. We investigate the effect of the number of noise M. Figure 6(b) illustrates the certified accuracy with varying $M \in \{1, 2, 4, 8\}$. The robustness of the smoothed classifier increases as M increases, sacrificing its clean accuracy. For large M, the classifier can incorporate the information of many Gaussian noises and take advantage of increasing p_f. Therefore,

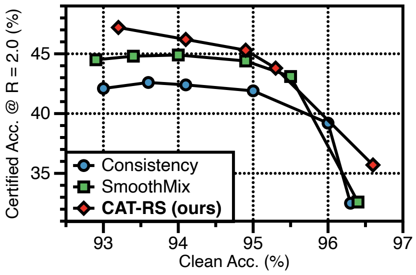
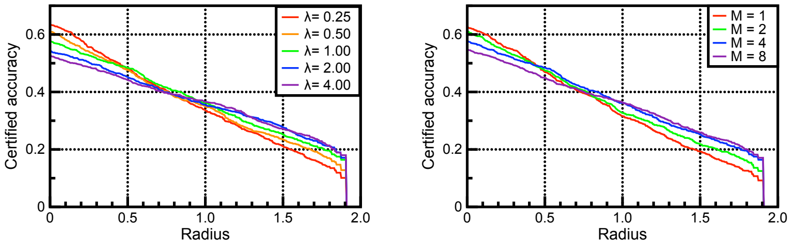


Figure 5. Trade-off between clean *vs.* certified acc. on MNIST ($\sigma = 1.0$) for varying control hyperparameter.



(a) Effect of λ

(b) Effect of M

Figure 6. Comparison of certified accuracy of CAT-RS ablations on CIFAR-10. We use ResNet-20 for ablation study and plot the results at $\sigma = 0.5$. More results for the plots can be found in Table 10 and 11, respectively.

Table 8. Comparison of ACR and approximate certified test accuracy (%) for ablations of CAT-RS. All the models are trained on CIFAR-10 with $\sigma = 0.5$. L^{base} as mark indicates the use of Gaussian training (5). Also, we mark "Mask" column if we apply indicator $\mathbb{1}[K = M]$ to L^{high} in (8).

Method (CIFAR-10)	L^{low}	L^{High}	Mask	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75
L^{base} (Gaussian; (5))	L^{base}	✗	-	0.523	66.2	55.2	42.9	31.0	21.3	14.4	7.9	3.7
(a) L^{low} only	✓	✗	-	0.508	67.0	54.6	41.9	29.7	20.4	13.1	7.6	3.6
(b) L^{high} only	✗	✓	✗	0.685	55.2	48.7	44.0	39.9	34.8	30.7	26.5	20.7
(c) $L^{\text{base}} + \lambda \cdot L^{\text{high}}$	L^{base}	✓	✗	0.694	62.4	54.4	48.1	41.4	34.4	28.1	22.5	17.6
(d) $L^{\text{low}} + \lambda \cdot L^{\text{high}}$	✓	✓	✗	0.706	59.7	54.6	48.2	41.2	35.5	30.1	23.6	18.5
$L^{\text{CAT-RS}}$ **(Ours; (8))**	✓	✓	✓	0.710	57.7	52.7	48.4	41.6	36.2	29.7	25.3	20.6

the smoothed classifier can provide a more robust prediction (3). We fix $M = 4$ for overall experiments as it offers a better trade-off between accuracy and robustness.

Accuracy-robustness trade-off. To further validate that our method can exhibit a better trade-off between accuracy and robustness compared to other methods, we additionally compare the performance trends between clean accuracy and certified accuracy at $r = 2.0$ as we vary a hyperparameter to control the trade-off, *e.g.*, λ (8) in case of our method. We use $\sigma = 1.0$ for this experiment. We choose Consistency [16] and SmoothMix [15] for this comparison, considering that they also offer a single hyperparameter (namely λ and η, respectively) for the balance between accuracy and robustness similar to our method, while both generally achieve good performances among the baselines considered. The results plotted in Figure 5 show that CAT-RS indeed exhibits a higher trade-off frontier compared to both methods, which confirms the effectiveness of our method. More detailed results can be found in Appendix E.

Loss design. Our loss design of $L^{\text{CAT-RS}}$ in (8) combines several important ideas as proposed in Section 3, and here we validate that each of the components has an individual effect in improving the certified robustness. In Table 8, we compare several variants of $L^{\text{CAT-RS}}$, including the followings: (a) training with L^{low} (6) only, (b) L^{high} (7) only, (c) $L^{\text{base}} + \lambda \cdot L^{\text{high}}$, where $L^{\text{base}} := \frac{1}{M}\sum_{i=1}^{M} \mathbb{CE}(F(x+\delta_i), y)$ denotes the standard Gaussian training, and (d) $L^{\text{low}} + \lambda \cdot L^{\text{high}}$. Here, notie that (c) and (d) does not apply the masking condition $\mathbb{1}[K = M]$ to L^{high} (Section 3.3) compared to $L^{\text{CAT-RS}}$.

Overall, we observe that (a) even though ACR of L^{low} is slightly degraded compared to L^{base}, L^{low} can achive a better clean accuracy instead, and (b) when combined with L^{high}, L^{low} achieves a better ACR than $L^{\text{base}} + \lambda \cdot L^{\text{high}}$ from a better balancing between accuracy and robustness; and (c) yet, CAT-RS further improves ACR by applying the masking to L^{high}.

Table 9, on the other hand, considers three variants of L^{high} (7): (a) the outer maximization (7) is replaced by averaging; (b) the label assignment $\hat{y}$ is set by $\hat{F}(x) := \frac{1}{M}\sum_{i=1}^{M} F(x+\delta_i)$, *i.e.*, the averaged prediction over M noise samples; and (c) the label assignment $\hat{y}$ is set by the hard label y. The results show that our form of worst-case loss achieves the best performance in terms of ACR, confirming that both designs of (a) maximizing loss over noise samples, and (b) utilizing soft-labeled $\hat{y}$'s in L^{high} work effectively.

Table 9. Comparison of ACR and approximate certified test accuracy (%) ablations of $L^{\texttt{high}}$ (7). All the models are trained on CIFAR-10 with $\sigma = 0.5$.

Method (CIFAR-10)	ACR	0.00	0.25	0.50	0.75	1.00	1.25	1.50	1.75
(a) $\frac{1}{M}\sum_i \left(\max_{\delta_i^*} \mathrm{KL}(F(x+\delta_i^*), \hat{y})\right)$	0.694	61.2	53.5	46.7	41.0	34.1	29.3	23.6	18.2
(b) $\max_{i,\delta_i^*} \mathrm{KL}(F(x+\delta_i^*), \hat{F}(x))$	0.694	57.2	51.8	46.9	40.7	34.7	30.7	24.4	18.7
(c) $\max_{i,\delta_i^*} \mathrm{KL}(F(x+\delta_i^*), y)$	0.701	56.4	51.5	46.3	39.8	36.0	30.6	25.8	20.9
$\max_{i,\delta_i^*} \mathrm{KL}(F(x+\delta_i^*), \hat{y})$ (**$L^{\texttt{high}}$; Ours**)	0.710	57.7	52.7	48.4	41.6	36.2	29.7	25.3	20.6

Table 10. Comparison of ACR and approximate certified test accuracy (%) for varying λ on CIFAR-10. We assume $\sigma = 0.5$.

CIFAR-10		Certified accuracy (%)							
Setups	ACR	0.0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
$\lambda = 0.25$	0.684	63.4	55.6	48.1	40.4	33.6	27.1	21.2	15.2
$\lambda = 0.50$	0.692	60.9	54.1	47.6	40.2	35.0	27.9	23.5	18.2
$\lambda = 1.00$	0.710	57.7	52.7	48.4	41.6	36.2	29.7	25.3	20.6
$\lambda = 2.00$	0.703	54.2	50.3	45.2	39.9	35.5	31.9	27.8	22.1
$\lambda = 4.00$	0.698	52.6	48.6	44.2	39.7	36.6	32.7	27.2	22.9

Table 11. Comparison of ACR and approximative certified test accuracy (%) for varying M on CIFAR-10. We assume $\sigma = 0.5$.

CIFAR-10		Certified accuracy (%)							
Setups	ACR	0.0	0.25	0.5	0.75	1.0	1.25	1.5	1.75
$M = 1$	0.661	66.2	55.2	42.9	31.0	21.3	14.4	7.9	3.7
$M = 2$	0.684	61.2	54.2	47.5	40.5	32.8	28.1	21.9	17.4
$M = 4$	0.710	57.7	52.7	48.4	41.6	36.2	29.7	25.3	20.6
$M = 8$	0.697	54.7	50.2	45.0	40.1	36.4	31.3	25.9	21.6

G. Statistical Significance of Results

Table 12. Comparison of the mean and standard deviation of ACR on MNIST and CIFAR-10. The results are calculated over 5 runs with different seeds. For each column, we set our result bold-faced if it achieves the highest ACR among the baselines.

Dataset	MNIST			CIFAR-10
ACR	$\sigma = 0.25$	$\sigma = 0.5$	$\sigma = 1.0$	$\sigma = 0.5$
Gaussian [5]	0.9109 ± 0.0003	1.5581 ± 0.0016	1.6184 ± 0.0021	0.5406 ± 0.0109
Stability [23]	0.9152 ± 0.0007	1.5719 ± 0.0028	1.6341 ± 0.0018	0.5254 ± 0.0209
SmoothAdv [33]	0.9322 ± 0.0005	1.6872 ± 0.0007	1.7786 ± 0.0017	0.7009 ± 0.0145
MACER [46]	0.9201 ± 0.0006	1.5899 ± 0.0069	1.5950 ± 0.0051	0.6698 ± 0.0045
Consistency [16]	0.9279 ± 0.0003	1.6549 ± 0.0011	1.7376 ± 0.0017	0.7170 ± 0.0034
SmoothMix [15]	0.9317 ± 0.0002	1.6932 ± 0.0007	1.8185 ± 0.0016	0.7362 ± 0.0063
CAT-RS (Ours)	**0.9329** ± 0.0001	**1.7004** ± 0.0005	**1.8282** ± 0.0018	**0.7525** ± 0.0028

In Table 3 and 1, we compare single-seed results of ACR and approximate certified accuracy following the evaluation scheme of the baselines [5,15,16,23,33,46]. We report a variance analysis of results across 5 different seeds in Table 12.[9] Our major performance metric of ACR shows quite robust performance over multiple runs. It confirms the statistical significance of our improvements.

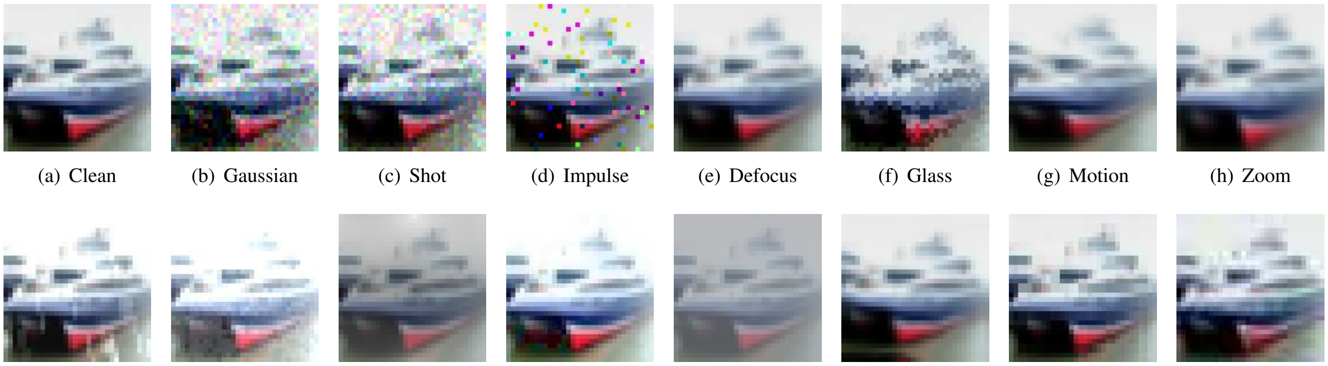
[9] For CIFAR-10, we subsampled test CIFAR-10 of size 2000. There can be discrepancy from the value reported in Table 1 based on the full test set.

H. Detailed Results on CIFAR-10-C

In this section, we report the detailed results on CIFAR-10-C test dataset, *i.e.*, ACR and the certified accuracy for each corruption severity and type. Our method consistently achieves the best performance in terms of mACR and mAcc among the baselines over severities.[10]

Table 13. Comparison of *average certified radius* (ACR) and certified accuracy at $r = 0.0$ on CIFAR-10-C. We report the results for five different corruption severities. We set the best values bold-faced for each column. We set the runner-up values underlined.

	Average Certified Radius						Certifed Test Accuracy (%)					
Severity	1	2	3	4	5	mACR	1	2	3	4	5	mAcc
Gaussian [5]	0.392	0.363	0.342	0.319	0.298	0.343	68.6	66.4	64.7	62.9	59.6	64.4
Stability [23]	0.341	0.319	0.299	0.286	0.267	0.302	67.0	63.1	60.1	58.4	55.0	60.7
SmoothAdv [33]	0.490	0.465	0.449	0.428	0.404	0.447	68.1	65.2	63.7	62.7	58.6	63.7
MACER [46]	0.457	0.431	0.409	0.385	0.364	0.409	73.5	71.5	69.0	66.4	63.5	68.8
Consistency [16]	0.488	0.463	0.442	0.424	0.402	0.444	69.5	67.1	65.4	63.9	62.0	65.6
SmoothMix [15]	0.490	0.466	0.445	0.422	0.405	0.446	72.1	69.5	66.8	66.8	63.3	67.7
CAT-RS (Ours)	**0.521**	**0.493**	**0.476**	**0.458**	**0.430**	**0.475**	**75.3**	**71.6**	**69.8**	**69.4**	**64.4**	**70.1**



(a) Clean (b) Gaussian (c) Shot (d) Impulse (e) Defocus (f) Glass (g) Motion (h) Zoom

(i) Snow (j) Frost (k) Fog (l) Bright (m) Contrast (n) Elastic (o) Pixel (p) JPEG

Figure 7. Images in CIFAR-10-C: (a) is a clean test image in CIFAR-10 dataset, and other images are the corresponding corrupted images contained in CIFAR-10-C. All corrupted images are drawn from severity 3.

[10] The dataset is hosted at `https://zenodo.org/record/2535967#.Yisixi8RpQI`.

Table 14. Comparison of *average certified radius* (ACR) on CIFAR-10-C of severity 1. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	0.419	0.358	0.509	0.479	0.506	0.511	**0.549**
Shot	0.422	0.365	0.512	0.480	0.509	0.514	**0.550**
Impulse	0.417	0.354	0.507	0.477	0.507	0.510	**0.546**
Defocus	0.416	0.360	0.505	0.478	0.506	0.512	**0.544**
Glass	0.377	0.312	0.481	0.451	0.484	0.496	**0.512**
Motion	0.394	0.341	0.483	0.449	0.482	0.497	**0.517**
Zoom	0.367	0.329	0.487	0.442	0.483	0.501	**0.520**
Snow	0.412	0.362	0.516	0.482	0.515	0.510	**0.544**
Frost	0.365	0.359	0.488	0.443	0.487	0.482	**0.511**
Fog	0.360	0.310	0.466	0.436	0.460	0.453	**0.485**
Bright	0.421	0.375	0.517	0.480	0.512	0.514	**0.553**
Contrast	0.332	0.272	0.441	0.403	0.435	0.424	**0.444**
Elastic	0.337	0.299	0.421	0.407	0.422	0.411	**0.446**
Pixel	0.422	0.361	0.509	0.477	0.509	0.514	**0.548**
JPEG	0.420	0.361	0.510	0.476	0.505	0.508	**0.543**
mACR	0.392	0.341	0.490	0.457	0.488	0.490	**0.521**

Table 15. Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C of severity 1. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	70.0	67.0	71.0	72.0	70.0	73.0	**77.0**
Shot	72.0	68.0	70.0	74.0	71.0	74.0	**77.0**
Impulse	69.0	69.0	69.0	75.0	71.0	74.0	**78.0**
Defocus	69.0	68.0	69.0	73.0	69.0	71.0	**77.0**
Glass	67.0	65.0	67.0	72.0	69.0	71.0	**75.0**
Motion	66.0	66.0	68.0	**74.0**	72.0	71.0	72.0
Zoom	68.0	67.0	70.0	74.0	67.0	73.0	**75.0**
Snow	71.0	68.0	68.0	77.0	70.0	74.0	**79.0**
Frost	71.0	66.0	68.0	**76.0**	72.0	72.0	74.0
Fog	68.0	67.0	69.0	72.0	70.0	**74.0**	72.0
Bright	71.0	70.0	67.0	76.0	71.0	75.0	**80.0**
Contrast	66.0	62.0	64.0	**72.0**	67.0	69.0	70.0
Elastic	66.0	64.0	62.0	69.0	62.0	65.0	**70.0**
Pixel	67.0	69.0	69.0	75.0	70.0	73.0	**77.0**
JPEG	68.0	69.0	70.0	71.0	71.0	73.0	**77.0**
mAcc	68.6	67.0	68.1	73.5	69.5	72.1	**75.3**

Table 16. Comparison of *average certified radius* (ACR) on CIFAR-10-C of severity 2. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	0.414	0.356	0.510	0.476	0.506	0.515	**0.546**
Shot	0.419	0.360	0.505	0.477	0.507	0.511	**0.544**
Impulse	0.411	0.345	0.502	0.467	0.498	0.506	**0.538**
Defocus	0.397	0.344	0.494	0.464	0.497	0.506	**0.530**
Glass	0.363	0.303	0.481	0.435	0.485	0.497	**0.514**
Motion	0.372	0.338	0.464	0.440	0.479	0.493	**0.512**
Zoom	0.361	0.325	0.477	0.436	0.474	0.491	**0.514**
Snow	0.361	0.334	0.470	0.444	0.482	0.470	**0.512**
Frost	0.321	0.340	**0.475**	0.421	0.444	0.447	0.465
Fog	0.251	0.200	0.355	0.348	0.349	0.335	**0.359**
Bright	0.413	0.378	0.512	0.472	0.509	0.505	**0.555**
Contrast	0.166	0.136	**0.269**	0.229	0.242	0.233	0.253
Elastic	0.359	0.307	0.453	0.420	0.457	0.464	**0.467**
Pixel	0.417	0.360	0.505	0.468	0.505	0.513	**0.544**
JPEG	0.415	0.355	0.500	0.472	0.504	0.506	**0.536**
mACR	0.363	0.319	0.465	0.431	0.463	0.466	**0.493**

Table 17. Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C of severity 2. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	70.0	65.0	70.0	72.0	68.0	73.0	**76.0**
Shot	70.0	69.0	68.0	74.0	69.0	72.0	**76.0**
Impulse	70.0	63.0	70.0	74.0	71.0	74.0	**75.0**
Defocus	65.0	66.0	68.0	73.0	69.0	70.0	**76.0**
Glass	65.0	61.0	68.0	**74.0**	67.0	70.0	72.0
Motion	69.0	64.0	68.0	74.0	73.0	72.0	**75.0**
Zoom	66.0	66.0	69.0	72.0	67.0	73.0	**75.0**
Snow	69.0	66.0	64.0	74.0	70.0	74.0	**76.0**
Frost	65.0	70.0	67.0	71.0	71.0	**74.0**	69.0
Fog	**65.0**	53.0	55.0	**65.0**	59.0	60.0	58.0
Bright	74.0	69.0	68.0	77.0	73.0	74.0	**79.0**
Contrast	49.0	32.0	42.0	**50.0**	42.0	44.0	43.0
Elastic	64.0	65.0	65.0	**76.0**	69.0	70.0	71.0
Pixel	67.0	69.0	68.0	75.0	69.0	72.0	**78.0**
JPEG	68.0	68.0	68.0	71.0	69.0	70.0	**75.0**
mAcc	66.4	63.1	65.2	71.5	67.1	69.5	**71.6**

Table 18. Comparison of *average certified radius* (ACR) on CIFAR-10-C of severity 3. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	0.414	0.349	0.504	0.477	0.506	0.515	**0.542**
Shot	0.410	0.348	0.505	0.469	0.500	0.506	**0.542**
Impulse	0.397	0.327	0.500	0.454	0.493	0.502	**0.528**
Defocus	0.376	0.330	0.484	0.447	0.485	0.494	**0.514**
Glass	0.355	0.301	0.480	0.433	0.479	0.491	**0.513**
Motion	0.337	0.302	0.455	0.410	0.464	0.472	**0.481**
Zoom	0.347	0.315	0.466	0.422	0.462	0.478	**0.503**
Snow	0.370	0.328	0.462	0.436	0.477	0.458	**0.509**
Frost	0.287	0.276	**0.436**	0.365	0.382	0.381	0.420
Fog	0.173	0.126	0.291	0.249	0.269	0.253	**0.301**
Bright	0.392	0.375	0.504	0.459	0.504	0.490	**0.548**
Contrast	0.113	0.107	**0.205**	0.158	0.175	0.166	0.190
Elastic	0.338	0.298	0.436	0.417	0.435	0.456	**0.465**
Pixel	0.405	0.353	0.500	0.467	0.499	0.507	**0.537**
JPEG	0.413	0.351	0.501	0.473	0.502	0.504	**0.540**
mACR	0.342	0.299	0.449	0.409	0.442	0.445	**0.476**

Table 19. Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C of severity 3. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	72.0	66.0	71.0	73.0	70.0	**76.0**	**76.0**
Shot	69.0	64.0	69.0	73.0	69.0	73.0	**76.0**
Impulse	70.0	60.0	69.0	73.0	71.0	73.0	**74.0**
Defocus	64.0	66.0	69.0	71.0	70.0	71.0	**73.0**
Glass	67.0	63.0	71.0	73.0	69.0	71.0	**74.0**
Motion	65.0	61.0	68.0	**74.0**	71.0	68.0	69.0
Zoom	64.0	65.0	64.0	70.0	68.0	71.0	**76.0**
Snow	70.0	65.0	62.0	73.0	68.0	69.0	**74.0**
Frost	63.0	65.0	60.0	**69.0**	66.0	65.0	66.0
Fog	**56.0**	35.0	46.0	54.0	49.0	48.0	55.0
Bright	72.0	71.0	69.0	75.0	74.0	77.0	**78.0**
Contrast	39.0	22.0	34.0	**40.0**	32.0	29.0	34.0
Elastic	64.0	62.0	68.0	**71.0**	65.0	**71.0**	70.0
Pixel	68.0	70.0	68.0	74.0	69.0	71.0	**76.0**
JPEG	67.0	66.0	68.0	72.0	70.0	69.0	**76.0**
mAcc	64.7	60.1	63.7	69.0	65.4	66.8	**69.8**

Table 20. Comparison of *average certified radius* (ACR) on CIFAR-10-C of severity 4. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	0.402	0.342	0.504	0.468	0.505	0.510	**0.543**
Shot	0.417	0.352	0.500	0.473	0.503	0.507	**0.541**
Impulse	0.376	0.308	0.490	0.442	0.489	0.494	**0.531**
Defocus	0.360	0.320	0.474	0.432	0.477	0.484	**0.503**
Glass	0.313	0.271	0.474	0.386	0.461	0.469	**0.499**
Motion	0.335	0.301	0.451	0.405	0.458	0.461	**0.481**
Zoom	0.337	0.308	0.459	0.410	0.453	0.465	**0.493**
Snow	0.311	0.308	0.414	0.360	0.399	0.369	**0.448**
Frost	0.270	0.282	0.400	0.349	0.362	0.369	**0.405**
Fog	0.125	0.084	0.196	0.186	0.195	0.167	**0.214**
Bright	0.363	0.369	0.486	0.446	0.492	0.473	**0.524**
Contrast	0.071	0.082	0.140	0.107	0.122	0.112	**0.148**
Elastic	0.309	0.263	0.438	0.385	0.446	0.440	**0.469**
Pixel	0.389	0.345	0.498	0.460	0.496	0.509	**0.532**
JPEG	0.412	0.352	0.503	0.465	0.500	0.501	**0.535**
mACR	0.319	0.286	0.428	0.385	0.424	0.422	**0.458**

Table 21. Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C of severity 4. We set the highest values bold-faced for each row. We set the runner-up values underlined.

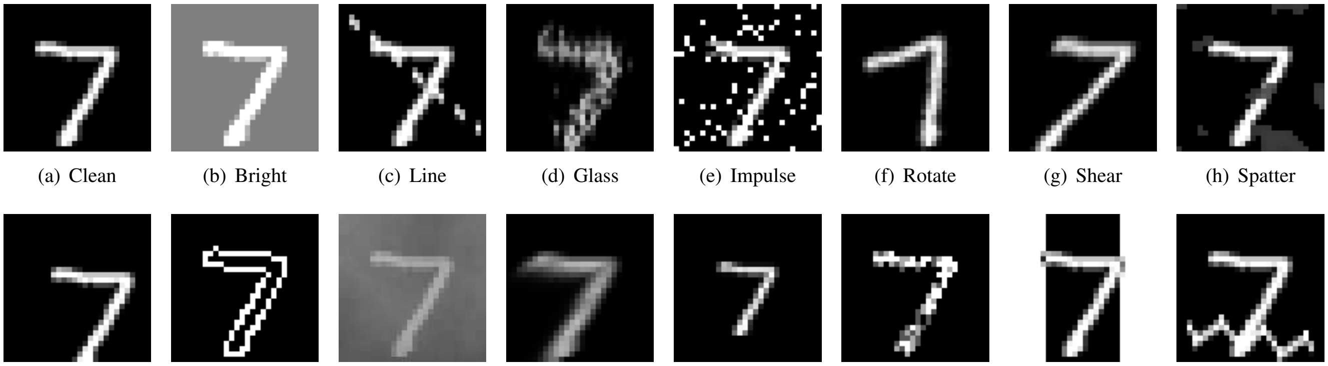
Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	71.0	64.0	68.0	72.0	70.0	72.0	**79.0**
Shot	71.0	65.0	68.0	72.0	70.0	74.0	**77.0**
Impulse	70.0	59.0	69.0	76.0	73.0	73.0	**77.0**
Defocus	64.0	66.0	69.0	71.0	69.0	71.0	**73.0**
Glass	64.0	62.0	70.0	72.0	70.0	**74.0**	73.0
Motion	66.0	61.0	69.0	70.0	70.0	69.0	**72.0**
Zoom	65.0	63.0	64.0	69.0	70.0	70.0	**76.0**
Snow	68.0	66.0	67.0	**71.0**	64.0	68.0	69.0
Frost	69.0	60.0	64.0	64.0	65.0	**74.0**	69.0
Fog	42.0	26.0	40.0	**45.0**	40.0	42.0	**45.0**
Bright	70.0	72.0	69.0	72.0	76.0	73.0	**77.0**
Contrast	25.0	19.0	22.0	**29.0**	21.0	24.0	23.0
Elastic	64.0	62.0	63.0	69.0	65.0	74.0	**77.0**
Pixel	65.0	66.0	70.0	74.0	71.0	72.0	**76.0**
JPEG	69.0	65.0	69.0	70.0	65.0	72.0	**78.0**
mAcc	62.9	58.4	62.7	66.4	63.9	66.8	**69.4**

Table 22. Comparison of *average certified radius* (ACR) on CIFAR-10-C of severity 5. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	0.408	0.335	0.501	0.467	0.500	0.511	**0.540**
Shot	0.403	0.325	0.494	0.458	0.498	0.502	**0.532**
Impulse	0.346	0.275	0.476	0.421	0.471	0.484	**0.505**
Defocus	0.311	0.290	0.445	0.389	0.447	0.449	**0.471**
Glass	0.308	0.269	0.449	0.372	0.451	0.464	**0.488**
Motion	0.321	0.286	0.438	0.382	0.445	0.446	**0.471**
Zoom	0.316	0.296	0.449	0.391	0.437	0.446	**0.475**
Snow	0.277	0.290	0.401	0.363	0.366	0.384	**0.420**
Frost	0.248	0.236	**0.372**	0.309	0.330	0.334	0.369
Fog	0.078	0.046	0.086	0.110	**0.112**	0.100	0.104
Bright	0.301	0.335	0.415	0.400	0.430	0.409	**0.439**
Contrast	0.046	0.058	0.087	0.079	0.093	0.075	**0.103**
Elastic	0.313	0.280	0.458	0.398	0.466	0.462	**0.472**
Pixel	0.386	0.332	0.486	0.453	0.488	0.503	**0.527**
JPEG	0.405	0.350	0.504	0.466	0.500	0.502	**0.530**
mACR	0.298	0.267	0.404	0.364	0.402	0.405	**0.430**

Table 23. Comparison of certified accuracy at $r = 0.0$ (%) on CIFAR-10-C of severity 5. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Gaussian	71.0	61.0	71.0	74.0	71.0	73.0	**76.0**
Shot	68.0	62.0	67.0	71.0	69.0	70.0	**77.0**
Impulse	72.0	57.0	68.0	72.0	66.0	**74.0**	**74.0**
Defocus	62.0	61.0	67.0	68.0	69.0	70.0	**72.0**
Glass	63.0	59.0	67.0	67.0	70.0	**74.0**	70.0
Motion	65.0	60.0	63.0	69.0	68.0	68.0	**70.0**
Zoom	63.0	60.0	61.0	68.0	70.0	70.0	**75.0**
Snow	57.0	58.0	59.0	59.0	**63.0**	61.0	59.0
Frost	60.0	54.0	61.0	65.0	60.0	**66.0**	61.0
Fog	31.0	13.0	17.0	**33.0**	28.0	28.0	27.0
Bright	68.0	71.0	65.0	69.0	**72.0**	70.0	68.0
Contrast	18.0	15.0	12.0	**23.0**	16.0	16.0	19.0
Elastic	64.0	64.0	65.0	70.0	**71.0**	69.0	69.0
Pixel	65.0	64.0	68.0	**74.0**	70.0	71.0	**74.0**
JPEG	67.0	66.0	68.0	70.0	67.0	70.0	**75.0**
mAcc	59.6	55.0	58.6	63.5	62.0	63.3	**64.4**



(a) Clean (b) Bright (c) Line (d) Glass (e) Impulse (f) Rotate (g) Shear (h) Spatter

(i) Translate (j) Edges (k) Fog (l) Motion (m) Scale (n) Shot (o) Stripe (p) Zigzag

Figure 8. Images in MNIST-C test dataset: (a) is a clean test image in MNIST, and other images are the corresponding corrupted images contained in MNIST-C.

I. Results on MNIST-C

We perform the evaluation on MNIST-C [29], 15 replicas of MNIST [20] dataset, where each replica consists of a different type of corruption (*e.g.*, rotate, shear, spatter, etc.). We evaluate the corruption performance of the smoothed classifiers on the full test dataset of MNIST-C after training the base classifiers with MNIST dataset. In this experiment, we use $\sigma = 0.25$. Although the improvement of CAT-RS in MNIST-C is less dramatic than in CIFAR-10-C due to the fact that confidence information is more important in more complex dataset, CAT-RS still achieves higher mACR compared to the baselines considered.[11]

Table 24. Comparison of *average certified radius* (ACR) on MNIST-C. We set the highest values bold-faced for each row. We set the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Bright	0.540	0.599	0.320	**0.606**	0.410	0.316	0.319
Line	0.856	0.865	0.906	0.867	0.885	0.901	**0.910**
Glass	0.655	0.643	0.743	0.670	0.686	0.710	**0.758**
Impulse	0.785	0.800	0.868	0.813	0.828	0.847	**0.876**
Rotate	0.762	0.776	0.833	0.793	0.822	0.831	**0.835**
Shear	0.850	0.857	0.900	0.869	0.891	0.899	**0.902**
Spatter	0.841	0.844	0.895	0.860	0.880	0.892	**0.902**
Translate	0.315	0.332	0.392	0.346	0.388	**0.449**	0.366
Edges	0.354	0.390	0.496	0.430	0.489	0.486	**0.519**
Fog	0.116	0.097	0.108	**0.123**	0.094	0.102	0.112
Motion	0.626	0.610	0.704	0.627	0.675	**0.730**	0.704
Scale	0.637	0.636	0.727	0.666	0.736	**0.766**	0.714
Shot	0.836	0.835	0.902	0.856	0.886	0.894	**0.907**
Stripe	0.532	0.590	0.678	0.700	**0.771**	0.736	0.759
Zigzag	0.726	0.740	0.794	0.746	0.779	0.774	**0.815**
mACR	0.629	0.641	0.684	0.665	0.681	0.689	**0.693**

Table 25. Comparison of certified accuracy at $r = 0.0$ (%) on MNIST-C. We set the highest values bold-faced for each row, and the runner-up values underlined.

Type	Gaussian [5]	Stability [23]	SmoothAdv [33]	MACER [46]	Consistency [16]	SmoothMix [15]	**CAT-RS (Ours)**
Bright	91.6	98.1	68.7	**97.1**	82.0	63.1	64.5
Line	98.5	98.7	**99.1**	98.6	98.9	**99.1**	**99.1**
Glass	96.6	96.6	**97.3**	96.8	96.7	96.6	**97.3**
Impulse	97.9	98.3	**98.9**	98.5	98.7	98.7	**98.9**
Rotate	92.5	93.2	94.4	93.6	94.4	**94.7**	94.1
Shear	97.4	97.9	98.4	98.1	98.3	**98.5**	98.3
Spatter	97.9	98.1	98.8	98.3	98.8	**98.9**	**98.9**
Translate	51.7	52.8	55.6	53.4	56.6	**64.6**	51.4
Edges	72.3	71.9	72.1	**75.1**	73.5	72.2	73.8
Fog	54.7	55.8	35.2	**62.2**	35.0	24.8	35.8
Motion	94.7	94.8	95.9	94.9	96.2	**97.1**	95.1
Scale	94.0	94.3	93.4	94.9	95.8	**96.2**	91.6
Shot	98.6	98.6	99.0	98.8	**99.1**	99.0	99.0
Stripe	76.8	81.7	88.2	89.9	**94.0**	92.5	92.0
Zigzag	90.2	91.9	93.6	91.2	92.9	93.1	**95.2**
mAcc	87.0	88.2	85.9	**89.4**	87.4	85.9	85.7

[11] The dataset is hosted at https://zenodo.org/record/3239543#.YisCti8RpQJ.