ETHICS STATEMENT

Our work can positively impact the society by improving the robustness and security of AI systems. We have not involved human subjects or data set releases; instead, we carefully follow the provided licenses of existing data and models for developing and evaluating our method.

REPRODUCIBILITY STATEMENT

For theoretical analysis, all necessary assumptions are listed in B.1 and the complete proofs are included in B.2. The experimental setting and datasets are provided in section 5. The pseudo-code for DensePure is in C.1 and the fast sampling procedures are provided in C.2.

REFERENCES

Brian DO Anderson. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Hangbo Bao, Li Dong, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

Nicholas Carlini, Florian Tramer, J Zico Kolter, et al. (certified!!) adversarial robustness for free! *arXiv preprint arXiv:2206.10550*, 2022.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1310–1320. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/cohen19c.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 2019.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020.

Mitch Hill, Jonathan Craig Mitchell, and Song-Chun Zhu. Stochastic security: Adversarial defense using long-run dynamics of energy-based models. In *International Conference on Learning Representations*, 2021.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020. URL https://arxiv.org/abs/2006.11239.

Miklós Z Horváth, Mark Niklas Müller, Marc Fischer, and Martin Vechev. Boosting randomized smoothing with variance reduced classifiers. *arXiv preprint arXiv:2106.06946*, 2021.

Jongheon Jeong and Jinwoo Shin. Consistency regularization for certified robustness of smoothed classifiers. *Advances in Neural Information Processing Systems*, 33:10558–10570, 2020.

Jongheon Jeong, Sejun Park, Minkyu Kim, Heung-Chang Lee, Do-Guk Kim, and Jinwoo Shin. Smoothmix: Training confidence-calibrated smoothed classifiers for certified robustness. *Advances in Neural Information Processing Systems*, 34:30153–30168, 2021.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Mathias Lecuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *2019 IEEE Symposium on Security and Privacy (SP)*, pp. 656–672. IEEE, 2019.

Kyungmin Lee. Provable defense by denoised smoothing with learned score function. In *ICLR Workshop on Security and Safety in Machine Learning Systems*, 2021.

Matthew Mirman, Timon Gehr, and Martin Vechev. Differentiable abstract interpretation for provably robust neural networks. In *International Conference on Machine Learning*, pp. 3578–3586. PMLR, 2018.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pp. 8162–8171. PMLR, 2021.

Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning (ICML)*, 2022.

Bernt Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *International Conference on Learning Representations*, 2018a.

Aditi Raghunathan, Jacob Steinhardt, and Percy S Liang. Semidefinite relaxations for certifying robustness to adversarial examples. In *NeurIPS*, 2018b.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. *Advances in Neural Information Processing Systems*, 32, 2019a.

Hadi Salman, Greg Yang, Huan Zhang, Cho-Jui Hsieh, and Pengchuan Zhang. A convex relaxation barrier to tight robustness verification of neural networks. *Advances in Neural Information Processing Systems*, 32:9835–9846, 2019b.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33:21945–21957, 2020.

Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-gan: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. In *International Conference on Learning Representations*, 2018.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. *Advances in Neural Information Processing Systems*, 34:1415–1428, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Jiachen Sun, Weili Nie, Zhiding Yu, Z Morley Mao, and Chaowei Xiao. Pointdp: Diffusion-driven purification against adversarial attacks on 3d point cloud recognition. *arXiv preprint arXiv:2208.09801*, 2022.

Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Shiqi Wang, Huan Zhang, Kaidi Xu, Xue Lin, Suman Jana, Cho-Jui Hsieh, and J Zico Kolter. Beta-crown: Efficient bound propagation with per-neuron split constraints for neural network robustness verification. *Advances in Neural Information Processing Systems*, 34:29909–29921, 2021.

Quanlin Wu, Hang Ye, and Yuntian Gu. Guided diffusion model for adversarial purification from random noise. *arXiv preprint arXiv:2206.10875*, 2022.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Runtian Zhai, Chen Dan, Di He, Huan Zhang, Boqing Gong, Pradeep Ravikumar, Cho-Jui Hsieh, and Liwei Wang. Macer: Attack-free and scalable robust training via maximizing certified radius. *arXiv preprint arXiv:2001.02378*, 2020.

Huan Zhang, Tsui-Wei Weng, Pin-Yu Chen, Cho-Jui Hsieh, and Luca Daniel. Efficient neural network robustness certification with general activation functions. In *NeurIPS*, 2018.

APPENDIX

Here is the appendix.

A NOTATIONS

p	data distribution
$\mathbb{P}(A)$	probability of event A
$\mathcal{C}^k$	set of functions with continuous k-th derivatives
$\boldsymbol{w}(t)$	standard Wiener Process
$\overline{\boldsymbol{w}}(t)$	reverse-time standard Wiener Process
$h(\boldsymbol{x}, t)$	drift coefficient in SDE
$g(t)$	diffusion coefficient in SDE
α_t	scaling coefficient at time t
σ_t^2	variance of added Gaussian noise at time t
$\{\mathbf{x}_t\}_{t\in[0,1]}$	diffusion process generated by SDE
$\{\hat{\mathbf{x}}_t\}_{t\in[0,1]}$	reverse process generated by reverse-SDE
p_t	distribution of $\mathbf{x}_t$ and $\hat{\mathbf{x}}_t$
$\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$	diffusion process generated by DDPM
$\{\beta_i\}_{i=1}^N$	pre-defined noise scales in DDPM
$\boldsymbol{\epsilon}_a$	adversarial attack
$\boldsymbol{x}_a$	adversarial sample
$\boldsymbol{x}_{a,t}$	scaled adversarial sample
$f(\cdot)$	classifier
$g(\cdot)$	smoothed classifier
$\mathbb{P}\left(\hat{\mathbf{x}}_0 = \boldsymbol{x} \middle\vert \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}\right)$	density of conditional distribution generated by reverse-SDE based on $\boldsymbol{x}_{a,t}$
$\mathcal{P}(\boldsymbol{x}_a; t)$	purification model with highest density point
$\mathcal{G}(\boldsymbol{x}_0)$	data region with the same label as $\boldsymbol{x}_0$
$\mathcal{D}_{\mathcal{P}}^f(\mathcal{G}(\boldsymbol{x}_0); t)$	robust region for $\mathcal{G}(\boldsymbol{x}_0)$ associated with base classifier f and purification model $\mathcal{P}$
$r_{\mathcal{P}}^f(\boldsymbol{x}_0; t)$	robust radius for the point associated with base classifier f and purification model $\mathcal{P}$
$\mathcal{D}_{sub}(\boldsymbol{x}_0; t)$	convex robust sub-region
$\boldsymbol{s}_\theta(\boldsymbol{x}, t)$	score function
$\{\mathbf{x}_t^\theta\}_{t\in[0,1]}$	reverse process generated by score-based diffusion model
$\mathbb{P}\left(\mathbf{x}_0^\theta = \boldsymbol{x} \middle\vert \mathbf{x}_t^\theta = \boldsymbol{x}_{a,t}\right)$	density of conditional distribution generated by score-based diffusion model based on $\boldsymbol{x}_{a,t}$
$\lambda(\tau)$	weighting scheme of training loss for score-based diffusion model
$\mathcal{J}_{\mathrm{SM}}(\theta, t; \lambda(\cdot))$	truncated training loss for score-based diffusion model
$\boldsymbol{\mu}_t, \boldsymbol{\nu}_t$	path measure for $\{\hat{\mathbf{x}}_\tau\}_{\tau\in[0,t]}$ and $\{\mathbf{x}_\tau^\theta\}_{\tau\in[0,t]}$ respectively

B MORE DETAILS ABOUT THEORETICAL ANALYSIS

B.1 ASSUMPTIONS

(i) The data distribution $p \in \mathcal{C}^2$ and $\mathbb{E}_{\boldsymbol{x}\sim p}[||\boldsymbol{x}||_2^2] < \infty$.

(ii) $\forall t \in [0,T] : h(\cdot,t) \in \mathcal{C}^1, \exists C > 0, \forall \boldsymbol{x} \in \mathbb{R}^n, t \in [0,T] : ||h(\boldsymbol{x},t)||_2 \leqslant C\,(1 + ||\boldsymbol{x}||_2)$.

(iii) $\exists C > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n : ||h(\boldsymbol{x},t) - h(\boldsymbol{y},t)||_2 \leqslant C\|\boldsymbol{x} - \boldsymbol{y}\|_2$.

(iv) $g \in \mathcal{C}$ and $\forall t \in [0,T], |g(t)| > 0$.

(v) $\forall t \in [0,T] : \boldsymbol{s}_\theta(\cdot,t) \in \mathcal{C}^1, \exists C > 0, \forall \boldsymbol{x} \in \mathbb{R}^n, t \in [0,T] : ||\boldsymbol{s}_\theta(\boldsymbol{x},t)||_2 \leqslant C\,(1 + ||\boldsymbol{x}||_2)$.

(vi) $\exists C > 0, \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n : ||\boldsymbol{s}_\theta(\boldsymbol{x},t) - \boldsymbol{s}_\theta(\boldsymbol{y},t)||_2 \leqslant C\|\boldsymbol{x} - \boldsymbol{y}\|_2$.

B.2 THEOREMS AND PROOFS

Theorem 3.1. *Under conditions B.1, solving equation reverse-SDE starting from time t and point $\boldsymbol{x}_{a,t} = \sqrt{\alpha_t}\boldsymbol{x}_a$ will generate a reversed random variable $\hat{\mathbf{x}}_0$ with conditional distribution*

$$\mathbb{P}\left(\hat{\mathbf{x}}_0 = \boldsymbol{x} | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}\right) \propto p(\boldsymbol{x}) \cdot \frac{1}{\sqrt{\left(2\pi\sigma_t^2\right)^n}} e^{\frac{-||\boldsymbol{x}-\boldsymbol{x}_a||_2^2}{2\sigma_t^2}}$$

where $\sigma_t^2 = \frac{1-\alpha_t}{\alpha_t}$ is the variance of the Gaussian noise added at timestamp t in the diffusion process SDE.

Proof. Under the assumption, we know $\{\mathbf{x}_t\}_{t\in[0,1]}$ and $\{\hat{\mathbf{x}}_t\}_{t\in[0,1]}$ follow the same distribution, which means

$$\begin{aligned}
\mathbb{P}\left(\hat{\mathbf{x}}_0 = \boldsymbol{x} | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}\right) &= \frac{\mathbb{P}(\hat{\mathbf{x}}_0 = \boldsymbol{x}, \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t})}{\mathbb{P}(\hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t})} \\
&= \frac{\mathbb{P}(\mathbf{x}_0 = \boldsymbol{x}, \mathbf{x}_t = \boldsymbol{x}_{a,t})}{\mathbb{P}(\mathbf{x}_t = \boldsymbol{x}_{a,t})} \\
&= \mathbb{P}\left(\mathbf{x}_0 = \boldsymbol{x}\right) \frac{\mathbb{P}(\mathbf{x}_t = \boldsymbol{x}_{a,t} | \mathbf{x}_0 = \boldsymbol{x})}{\mathbb{P}(\mathbf{x}_t = \boldsymbol{x}_{a,t})} \\
&\propto \mathbb{P}\left(\mathbf{x}_0 = \boldsymbol{x}\right) \frac{1}{\sqrt{\left(2\pi\sigma_t^2\right)^n}} e^{\frac{-||\boldsymbol{x}-\boldsymbol{x}_a||_2^2}{2\sigma_t^2}} \\
&= p(\boldsymbol{x}) \cdot \frac{1}{\sqrt{\left(2\pi\sigma_t^2\right)^n}} e^{\frac{-||\boldsymbol{x}-\boldsymbol{x}_a||_2^2}{2\sigma_t^2}}
\end{aligned}$$

where the third equation is due to the chain rule of probability and the last equation is a result of the diffusion process. □

Theorem 3.3. *Under conditions B.1 and classifier f, let $\boldsymbol{x}_0$ be the sample with ground-truth label and $\boldsymbol{x}_a$ be the adversarial sample, then (i) the purified sample $\mathcal{P}(\boldsymbol{x}_a; t)$ will have the ground-truth label if $\boldsymbol{x}_a$ falls into the following convex set,*

$$\mathcal{D}_{sub}\left(\boldsymbol{x}_0; t\right) := \bigcap_{\left\{\boldsymbol{x}_0' : f(\boldsymbol{x}_0') \neq f(\boldsymbol{x}_0)\right\}} \left\{ \boldsymbol{x}_a : (\boldsymbol{x}_a - \boldsymbol{x}_0)^\top (\boldsymbol{x}_0' - \boldsymbol{x}_0) < \sigma_t^2 \log\left(\frac{p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_0')}\right) + \frac{||\boldsymbol{x}_0' - \boldsymbol{x}_0||_2^2}{2} \right\},$$

and further, (ii) the purified sample $\mathcal{P}(\boldsymbol{x}_a; t)$ will have the ground-truth label if and only if $\boldsymbol{x}_a$ falls into the following set, $\mathcal{D}\left(\mathcal{G}(\boldsymbol{x}_0); t\right) := \bigcup_{\tilde{\boldsymbol{x}}_0 : f(\tilde{\boldsymbol{x}}_0) = f(\boldsymbol{x}_0)} \mathcal{D}_{sub}\left(\tilde{\boldsymbol{x}}_0; t\right)$. In other words, $\mathcal{D}\left(\mathcal{G}(\boldsymbol{x}_0); t\right)$ is the robust region for data region $\mathcal{G}(\boldsymbol{x}_0)$ under $\mathcal{P}(\cdot; t)$ and f.

Proof. We start with part (i).

The main idea is to prove that a point $\boldsymbol{x}_0'$ such that $f(\boldsymbol{x}_0') \neq f(\boldsymbol{x}_0)$ should have lower density than $\boldsymbol{x}_0$ in the conditional distribution in Theorem 3.1 so that $\mathcal{P}(\boldsymbol{x}_a; t)$ cannot be $\boldsymbol{x}_0'$. In other words, we should have

$$\mathbb{P}\left(\hat{\mathbf{x}}_0 = \boldsymbol{x}_0 | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}\right) > \mathbb{P}\left(\hat{\mathbf{x}}_0 = \boldsymbol{x}_0' \mid \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}\right).$$

By Theorem 3.1, this is equivalent to

$$\begin{aligned}
& p(\boldsymbol{x}_0) \cdot \frac{1}{\sqrt{(2\pi\sigma_t^2)^n}} e^{\frac{-||\boldsymbol{x}_0 - \boldsymbol{x}_a||_2^2}{2\sigma_t^2}} > p(\boldsymbol{x}_0') \cdot \frac{1}{\sqrt{(2\pi\sigma_t^2)^n}} e^{\frac{-||\boldsymbol{x}_0' - \boldsymbol{x}_a||_2^2}{2\sigma_t^2}} \\
\Leftrightarrow\ & \log\left(\frac{p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_0')}\right) > \frac{1}{2\sigma_t^2}\left(||\boldsymbol{x}_0 - \boldsymbol{x}_a||_2^2 - ||\boldsymbol{x}_0' - \boldsymbol{x}_a||_2^2\right) \\
\Leftrightarrow\ & \log\left(\frac{p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_0')}\right) > \frac{1}{2\sigma_t^2}\left(||\boldsymbol{x}_0 - \boldsymbol{x}_a||_2^2 - ||\boldsymbol{x}_0' - \boldsymbol{x}_0 + \boldsymbol{x}_0 - \boldsymbol{x}_a||_2^2\right) \\
\Leftrightarrow\ & \log\left(\frac{p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_0')}\right) > \frac{1}{2\sigma_t^2}\left(2(\boldsymbol{x}_a - \boldsymbol{x}_0)^\top(\boldsymbol{x}_0' - \boldsymbol{x}_0) - \|\boldsymbol{x}_0' - \boldsymbol{x}_0\|_2^2\right).
\end{aligned}$$

Re-organizing the above inequality, we obtain

$$(\boldsymbol{x}_a - \boldsymbol{x}_0)^\top(\boldsymbol{x}_0' - \boldsymbol{x}_0) < \sigma_t^2 \log\left(\frac{p(\boldsymbol{x}_0)}{p(\boldsymbol{x}_0')}\right) + \frac{1}{2}||\boldsymbol{x}_0' - \boldsymbol{x}_0||_2^2.$$

Note that the order of $\boldsymbol{x}_a$ is at most one in every term of the above inequality, so the inequality actually defines a half-space in $\mathbb{R}^n$ for every $(\boldsymbol{x}_0, \boldsymbol{x}_0')$ pair. Further, we have to satisfy the inequality for every $\boldsymbol{x}_0'$ such that $f(\boldsymbol{x}_0') \neq f(\boldsymbol{x}_0)$, therefore, by intersecting over all such half-spaces, we obtain a convex $\mathcal{D}_{\text{sub}}(\boldsymbol{x}_0; t)$.

Then we prove part (ii).

On the one hand, if $\boldsymbol{x}_a \in \mathcal{D}(\mathcal{G}(\boldsymbol{x}_0); t)$, then there exists one $\tilde{\boldsymbol{x}}_0$ such that $f(\tilde{\boldsymbol{x}}_0) = f(\boldsymbol{x}_0)$ and $\boldsymbol{x}_a \in \mathcal{D}_{\text{sub}}(\tilde{\boldsymbol{x}}_0; t)$. By part (i), $\tilde{\boldsymbol{x}}_0$ has higher probability than all other points with different labels from $\boldsymbol{x}_0$ in the conditional distribution $\mathbb{P}(\hat{\mathbf{x}}_0 = \boldsymbol{x} | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t})$ characterized by Theorem 3.1. Therefore, $\mathcal{P}(\boldsymbol{x}_a; t)$ should have the same label as $\boldsymbol{x}_0$. On the other hand, if $\boldsymbol{x}_a \notin \mathcal{D}(\mathcal{G}(\boldsymbol{x}_0); t)$, then there is a point $\tilde{\boldsymbol{x}}_1$ with different label from $\boldsymbol{x}_0$ such that for any $\tilde{\boldsymbol{x}}_0$ with the same label as $\boldsymbol{x}_0$, $\mathbb{P}(\hat{\mathbf{x}}_0 = \tilde{\boldsymbol{x}}_1 | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}) > \mathbb{P}(\hat{\mathbf{x}}_0 = \tilde{\boldsymbol{x}}_0 | \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t})$. In other words, $\mathcal{P}(\boldsymbol{x}_a; t)$ would have different label from $\boldsymbol{x}_0$. □

Theorem 3.4. *Under score-based diffusion model Song et al. (2021b) and conditions B.1, we can bound*

$$D_{KL}(\mathbb{P}(\hat{\mathbf{x}}_0 = \boldsymbol{x} \mid \hat{\mathbf{x}}_t = \boldsymbol{x}_{a,t}) \| \mathbb{P}(\mathbf{x}_0^\theta = \boldsymbol{x} \mid \mathbf{x}_t^\theta = \boldsymbol{x}_{a,t})) = \mathcal{J}_{\mathrm{SM}}(\theta, t; \lambda(\cdot))$$

where $\{\hat{\boldsymbol{x}}_\tau\}_{\tau\in[0,t]}$ *and* $\{\boldsymbol{x}_\tau^\theta\}_{\tau\in[0,t]}$ *are stochastic processes generated by reverse-SDE and score-based diffusion model respectively,*

$$\mathcal{J}_{\mathrm{SM}}(\theta, t; \lambda(\cdot)) := \frac{1}{2}\int_0^t \mathbb{E}_{p_\tau(\mathbf{x})}\left[\lambda(\tau)\left\|\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x}) - \boldsymbol{s}_\theta(\mathbf{x}, \tau)\right\|_2^2\right] \mathrm{d}\tau,$$

$\boldsymbol{s}_\theta(\mathbf{x}, \tau)$ *is the score function to approximate* $\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})$*, and* $\lambda : \mathbb{R} \to \mathbb{R}$ *is any weighting scheme used in the training score-based diffusion models.*

Proof. Similar to proof of (Song et al., 2021a, Theorem 1), let $\boldsymbol{\mu}_t$ and $\boldsymbol{\nu}_t$ be the path measure for reverse processes $\{\hat{\mathbf{x}}_\tau\}_{\tau\in[0,t]}$ and $\{\mathbf{x}_\tau^\theta\}_{\tau\in[0,t]}$ respectively based on the scaled adversarial sample $\boldsymbol{x}_{a,t}$. Under conditions B.1, the KL-divergence can be computed via the Girsanov theorem Oksendal

(2013):

$$
\begin{aligned}
& D_{\mathrm{KL}}\left(\mathbb{P}(\hat{\mathbf{x}}_0=\boldsymbol{x} \mid \hat{\mathbf{x}}_t=\boldsymbol{x}_{a,t}) \| \mathbb{P}(\mathbf{x}_0^\theta=\boldsymbol{x} \mid \mathbf{x}_t^\theta=\boldsymbol{x}_{a,t})\right) \\
= & -\mathbb{E}_{\boldsymbol{\mu}_t}\left[\log \frac{d \boldsymbol{\nu}_t}{d \boldsymbol{\mu}_t}\right] \\
\overset{(i)}{=} & \mathbb{E}_{\boldsymbol{\mu}_t}\left[\int_0^t g(\tau)\left(\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})-\boldsymbol{s}_\theta(\mathbf{x}, \tau)\right) \mathrm{d} \overline{\mathbf{w}}_\tau+\frac{1}{2} \int_0^t g(\tau)^2\left\|\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})-\boldsymbol{s}_\theta(\mathbf{x}, \tau)\right\|_2^2 \mathrm{~d} \tau\right] \\
= & \mathbb{E}_{\boldsymbol{\mu}_t}\left[\frac{1}{2} \int_0^t g(\tau)^2\left\|\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})-s_\theta(\mathbf{x}, \tau)\right\|_2^2 \mathrm{~d} \tau\right] \\
= & \frac{1}{2} \int_0^\tau \mathbb{E}_{p_\tau(\mathbf{x})}\left[g(\tau)^2\left\|\nabla_{\mathbf{x}} \log p_\tau(\mathbf{x})-s_\theta(\mathbf{x}, \tau)\right\|_2^2\right] \mathrm{d} \tau \\
= & \mathcal{J}_{\mathrm{SM}}\left(\theta, t ; g(\cdot)^2\right)
\end{aligned}
$$

where (i) is due to Girsanov Theorem and (ii) is due to the martingale property of Itô integrals. □

C MORE DETAILS ABOUT DENSEPURE

C.1 PSEUDO-CODE

We provide the pseudo code of DensePure in Algo. 1 and Alg. 2

Algorithm 1 DensePure pseudo-code with the highest density point

1: Initialization: choose off-the-shelf diffusion model and classifier f, choose $\psi = t$,
2: Input sample $\boldsymbol{x}_a = \boldsymbol{x}_0 + \boldsymbol{\epsilon}_a$
3: Compute $\hat{\boldsymbol{x}}_0 = \mathcal{P}(\boldsymbol{x}_a; \psi)$
4: $\hat{y} = f(\hat{\boldsymbol{x}}_0)$

Algorithm 2 DensePure pseudo-code with majority vote

1: Initialization: choose off-the-shelf diffusion model and classifier f, choose σ
2: Compute $\overline{\alpha}_n = \frac{1}{1+\sigma^2}$, $n = \arg\min_s \left\{ \left| \overline{\alpha}_s - \frac{1}{1+\sigma^2} \right| \mid s \in \{1, 2, \cdots, N\} \right\}$
3: Generate input sample $\boldsymbol{x}_{\mathrm{rs}} = \boldsymbol{x}_0 + \boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$
4: Choose schedule S^b, get $\hat{\boldsymbol{x}}_0^i \leftarrow \mathbf{rev}(\sqrt{\overline{\alpha}_n}\boldsymbol{x}_{\mathrm{rs}})_i, i = 1, 2, \ldots, K$ with Fast Sampling
5: $\hat{y} = \mathbf{MV}(\{f(\hat{\boldsymbol{x}}_0^1), \ldots, f(\hat{\boldsymbol{x}}_0^K)\}) = \arg\max_c \sum_{i=1}^K \mathbf{1}\{f(\hat{\boldsymbol{x}}_0^i) = c\}$

C.2 DETAILS ABOUT FAST SAMPLING

Applying single-step operation n times is a time-consuming process. In order to reduce the time complexity, we follow the method used in (Nichol & Dhariwal, 2021) and sample a subsequence S^b with b values (i.e., $S^b = \underbrace{\{n, \lfloor n - \frac{n}{b} \rfloor, \cdots, 1\}}_{b}$, where S_j^b is the j-th element in S^b and $S_j^b = \lfloor n - \frac{jn}{b} \rfloor, \forall j < b$ and $S_b^b = 1$) from the original schedule S (i.e., $S = \underbrace{\{n, n-1, \cdots, 1\}}_{n}$, where $S_j = j$ is the j-th element in S).

Within this context, we adapt the original $\overline{\alpha}$ schedule $\overline{\alpha}^S = \{\overline{\alpha}_1, \cdots, \overline{\alpha}_i, \cdots, \overline{\alpha}_n\}$ used for single-step to the new schedule $\overline{\alpha}^{S^b} = \{\overline{\alpha}_{S_1^b}, \cdots, \overline{\alpha}_{S_j^b}, \cdots, \overline{\alpha}_{S_b^b}\}$ (i.e., $\overline{\alpha}_i^{S^b} = \overline{\alpha}_{S_i^b} = \overline{\alpha}_{S_{\lfloor n - \frac{in}{b} \rfloor}}$ is the i-th element in $\overline{\alpha}^{S^b}$). We calculate the corresponding $\beta^{S^b} = \{\beta_1^{S^b}, \beta_2^{S^b}, \cdots, \beta_i^{S^b}, \cdots, \beta_b^{S^b}\}$ and $\widetilde{\beta}^{S^b} = \{\widetilde{\beta}_1^{S^b}, \widetilde{\beta}_2^{S^b}, \cdots, \widetilde{\beta}_i^{S^b}, \cdots, \widetilde{\beta}_b^{S^b}\}$ schedules, where $\beta_{S_i^b} = \beta_i^{S^b} = 1 - \frac{\overline{\alpha}_i^{S^b}}{\overline{\alpha}_{i-1}^{S^b}}$, $\widetilde{\beta}_{S_i^b} = \widetilde{\beta}_i^{S^b} = \frac{1 - \overline{\alpha}_{i-1}^{S^b}}{1 - \overline{\alpha}_i^{S^b}} \beta_{S_i^b}$. With these new schedules, we can use b times reverse steps to calculate

Methods	Noise	Certified Accuracy at $\epsilon(\%)$ 0.0	0.25	0.5	0.75	1.0
Carlini (Carlini et al., 2022)	$\sigma = 0.25$	**88.0**	73.8	56.2	41.6	0.0
	$\sigma = 0.5$	74.2	62.0	50.4	40.2	31.0
	$\sigma = 1.0$	49.4	41.4	34.2	27.8	21.8
Ours	$\sigma = 0.25$	87.6(-0.4)	**76.6(+2.8)**	**64.6(+8.4)**	**50.4(+8.8)**	0.0(+0.0)
	$\sigma = 0.5$	73.6(-0.6)	65.4(+3.4)	55.6(+5.2)	46.0(+5.8)	**37.4(+6.4)**
	$\sigma = 1.0$	55.0(+5.6)	47.8(+6.4)	40.8(+6.6)	33.0(+5.2)	28.2(+6.4)

Table A: Certified accuracy compared with Carlini et al. (2022) for CIFAR-10 at all σ. The numbers in the bracket are the difference of certified accuracy between two methods. Our diffusion model and classifier are the same as Carlini et al. (2022).

$\hat{\boldsymbol{x}}_0 = \underbrace{\textbf{Reverse}(\cdots \textbf{Reverse}(\textbf{Reverse}(\boldsymbol{x}_n; S_b^b); S_{b-1}^b); \cdots; 1)}_{b}$. Since $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_{S_i^b}, S_i^b)$ is parameterized as a range between β^{S^b} and $\widetilde{\beta}^{S^b}$, it will automatically be rescaled. Thus, $\hat{\boldsymbol{x}}_{S_{i-1}^b} = \textbf{Reverse}(\hat{\boldsymbol{x}}_{S_i^b}; S_i^b)$ is equivalent to sample $\boldsymbol{x}_{S_{i-1}^b}$ from $\mathcal{N}(\boldsymbol{x}_{S_{i-1}^b}; \boldsymbol{\mu}_{\boldsymbol{\theta}}(\boldsymbol{x}_{S_i^b}, S_i^b), \boldsymbol{\Sigma}_{\boldsymbol{\theta}}(\boldsymbol{x}_{S_i^b}, S_i^b))$.

D More Experimental details and Results

D.1 Implementation details

We select three different noise levels $\sigma \in \{0.25, 0.5, 1.0\}$ for certification. For the parameters of DensePure , The sampling numbers when computing the certified radius are $n = 100000$ for CIFAR-10 and $n = 10000$ for ImageNet. We evaluate the certified robustness on 500 samples subset of CIFAR-10 testset and 100 samples subset of ImageNet validation set. we set $K = 40$ and b = 10 except the results in ablation study. The details about the baselines are in the appendix.

D.2 Baselines.

We select randomized smoothing based methods including PixelDP (Lecuyer et al., 2019), RS (Cohen et al., 2019), SmoothAdv (Salman et al., 2019a), Consistency (Jeong & Shin, 2020), MACER (Zhai et al., 2020), Boosting (Horváth et al., 2021) , SmoothMix (Jeong et al., 2021), Denoised (Salman et al., 2020), Lee (Lee, 2021), Carlini (Carlini et al., 2022) as our baselines. Among them, PixelDP, RS, SmoothAdv, Consistency, MACER, and SmoothMix require training a smooth classifier for a better certification performance while the others do not. Salman et al. and Lee use the off-the-shelf classifier but without using the diffusion model. The most similar one compared with us is Carlini et al., which also uses both the off-the-shelf diffusion model and classifier. The above two settings mainly refer to Carlini et al. (2022), which makes us easier to compared with their results.

D.3 Main Results for Certified Accuracy

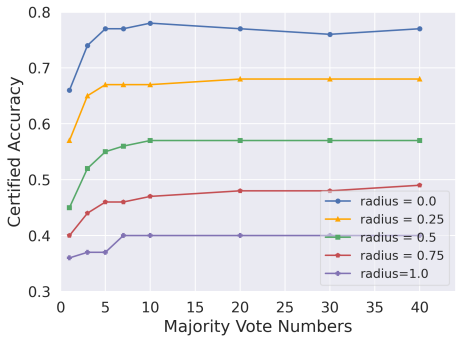
We compare with Carlini et al. (2022) in a more fine-grained version. We provide results of certified accuracy at different ϵ in Table A for CIFAR-10 and Table B for ImageNet. We include the accuracy difference between ours and Carlini et al. (2022) in the bracket in Tables. We can observe from the tables that the certified accuracy of our method outperforms Carlini et al. (2022) except $\epsilon = 0$ at $\sigma = 0.25, 0.5$ for CIFAR-10.

D.4 Experiments for Voting Samples

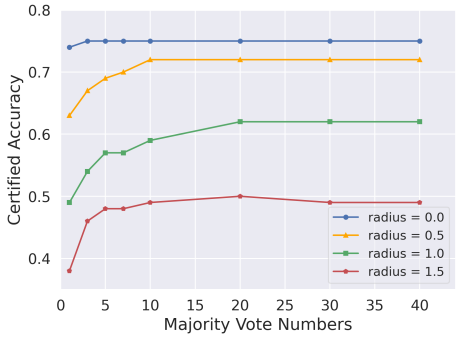
Here we provide more experiments with $\sigma \in \{0.5, 1.0\}$ and $b = 10$ for different voting samples K in Figure A and Figure B. The results for CIFAR-10 is in Figure G. We can draw the same conclusion mentioned in the main context .

Methods	Noise	Certified Accuracy at ϵ(%) 0.0	0.5	1.0	1.5	2.0	3.0
Carlini (Carlini et al., 2022)	$\sigma = 0.25$	77.0	71.0	0.0	0.0	0.0	0.0
	$\sigma = 0.5$	74.0	67.0	54.0	46.0	0.0	0.0
	$\sigma = 1.0$	59.0	53.0	49.0	38.0	29.0	22.0
Ours	$\sigma = 0.25$	**80.0(+3.0)**	**76.0(+5.0)**	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)
	$\sigma = 0.5$	75.0(+1.0)	72.0(+5.0)	**62.0(+8.0)**	**49.0(+3.0)**	0.0(+0.0)	0.0(+0.0)
	$\sigma = 1.0$	61.0(+2.0)	57.0(+4.0)	53.0(+4.0)	**49.0(+11.0)**	**37.0(+8.0)**	**26.0(+4.0)**

Table B: Certified accuracy compared with Carlini et al. (2022) for ImageNet at all σ. The numbers in the bracket are the difference of certified accuracy between two methods. Our diffusion model and classifier are the same as Carlini et al. (2022).



CIFAR=10



ImageNet

Figure A: Certified accuracy among different vote numbers with different radius. Each line in the figure represents the certified accuracy among different vote numbers K with Gaussian noise $\sigma = 0.50$.

D.5 Experiments for Fast Sampling Steps

We also implement additional experiments with $b \in \{1, 2, 10\}$ at $\sigma = 0.5, 1.0$. The results are shown in Figure C and Figure D. The results for CIFAR-10 are in Figure G. We draw the same conclusion as mentioned in the main context.

D.6 Experiments for Different Architectures

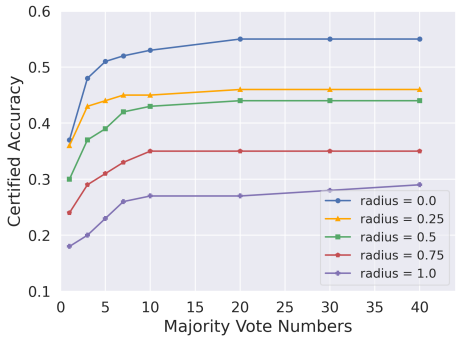
We try different model architectures of ImageNet including Wide ResNet-50-2 and ResNet 152 with $b = 2$ and $K = 10$. The results are shown in Figure F. we find that our method outperforms (Carlini et al., 2022) for all σ among different classifiers.

D.7 Experiments for randomized smoothing without diffusion model

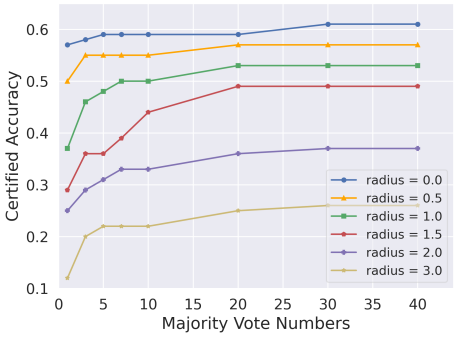
To explore randomized smoothing without diffusion model, we directly remove the diffusion model from our pipeline and conduct additional experiments.

First, we remove the diffusion model and perform randomized smoothing only on the pretrained classifier we used in DensePure (i.e., ViT-B/16 for CIFAR-10 and BEiT for ImageNet). The results are shown in Table C and Table D. The number in the bracket is calculated by the robust accuracy of pretrained classifier - the robust accuracy of DensePure. We can conclude from the table that without the help of diffusion models, neither ViT nor BEiT could reach high certified accuracy.

Second, we conduct additional experiments to fairly compare with randomized smoothing without diffusion models under majority vote settings. Specifically, we activate droppath in BEiT at the inference stage to support majority votes. The other settings are the same as DensePure. The results are shown in Table E. The number in the bracket is calculated by the robust accuracy of BeiT with majority votes - the robust accuracy of DensePure. We find that simply performing majority votes on the BeiT classifier will not result in higher certified robustness.

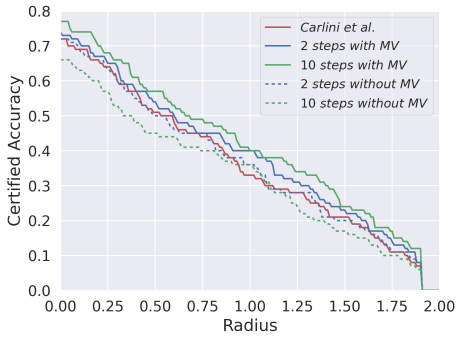


CIFAR=10

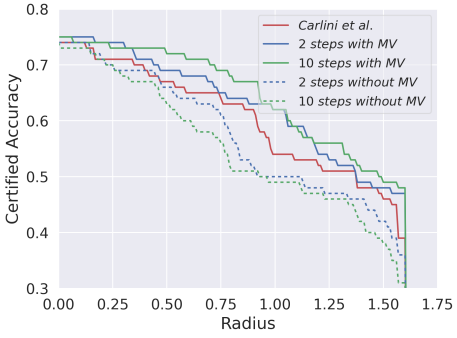


ImageNet

Figure B: Certified accuracy among different vote numbers with different radius. Each line in the figure represents the certified accuracy among different vote numbers K with Gaussian noise $\sigma = 1.00$.



CIFAR=10



ImageNet

Figure C: Certified accuracy with different fast sampling steps b. Each line in the figure shows the certified accuracy among different L_2 adversarial perturbation bound with Gaussian noise $\sigma = 0.50$.

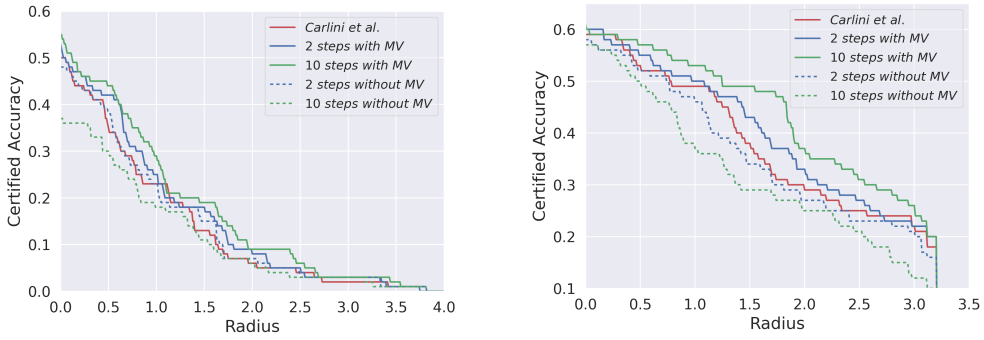
Third, to compare with randomized smoothing without diffusion model, we also evaluate certified accuracy with Gaussian augmentation-trained ViT models on CIFAR-10. The results shown in the table F prove that DensePure can still achieve higher certified accuracy than randomized smoothing on even Gaussian augmented models without diffusion models. The numbers in the bracket are the difference between the robust accuracy of Gaussian augmentation randomized smoothing and DensePure.

D.8 Experiments for K-consensus Aggregation

In K-Consensus Aggregation, if the classification results of the K consecutive reversed samples are the same, an early stop will be triggered. Here We calculate certified robustness for 100 subsamples of CIFAR-10 and ImageNet with 2 sampling steps, a maximum 10 majority votes and consensus threshold k=3. Results are shown in Table G and Table H. The column of "Avg MV" in the tables means the average of the actual number of majority votes required for our algorithm. For instance, if the predicted labels of the first 3 reversed samples are the same, the actual majority vote numbers will be 3. The numbers in the bracket are the difference between certified accuracy w/o K-Consensus Aggregation.

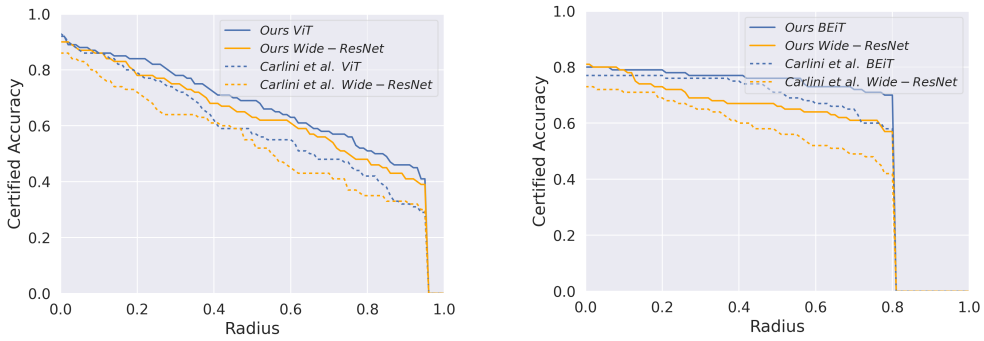
D.9 Experiments for Certified accuracy with less sampling steps and vote numbers

We also conduct additional experiments with 2 sampling steps and 5 majority votes. The results are shown in Table I. We find that our method still achieves better results than the existing method.



CIFAR=10 ImageNet

Figure D: Certified accuracy with different fast sampling steps b. Each line in the figure shows the certified accuracy among different L_2 adversarial perturbation bound with Gaussian noise $\sigma = 1.00$.

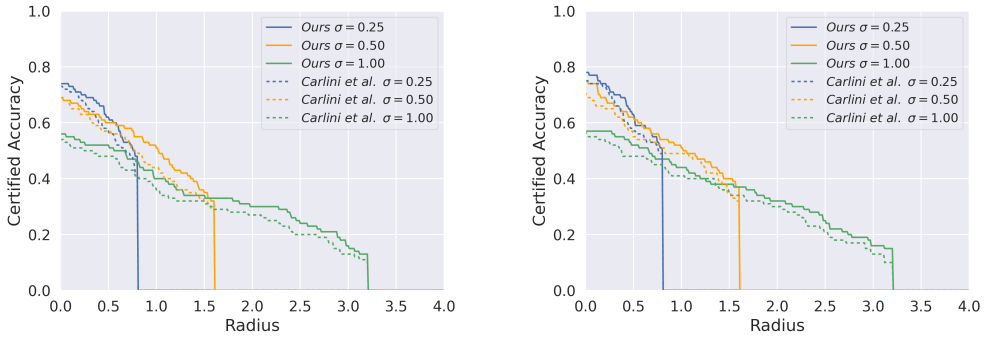


CIFAR=10 ImageNet

Figure E: Certified accuracy with different architectures. Each line in the figure shows the certified accuracy among different L_2 adversarial perturbation bound with Gaussian noise $\sigma = 0.25$.

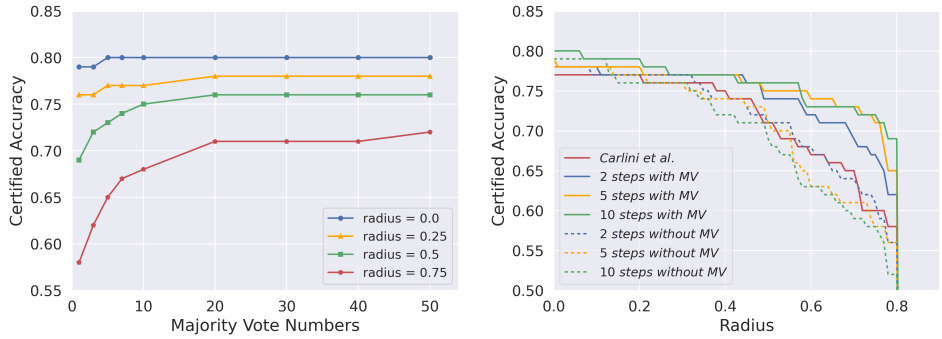
D.10 Experiments for DensePure 500 test sampling number results on ImageNet

We increase the ImageNet test sampling number from 100 to 500 and update the experiment results in Table J and Table K.



Wide ResNet-50-2 ResNet152

Figure F: Certified accuracy of ImageNet for different architectures. The lines represent the certified accuracy with different L_2 perturbation bound with different Gaussian noise $\sigma \in \{0.25, 0.50, 1.00\}$.



ImageNet ImageNet

Figure G: Ablation study. The left image shows the certified accuracy among different vote numbers with different radius $\epsilon \in \{0.0, 0.25, 0.5, 0.75\}$. Each line in the figure represents the certified accuracy of our method among different vote numbers K with Gaussian noise $\sigma = 0.25$. The right image shows the certified accuracy with different fast sampling steps b. Each line in the figure shows the certified accuracy among different L_2 adversarial perturbation bound.

	Certified Accuracy at ϵ(%)				
Noise	0.0	0.25	0.5	0.75	1.0
$\sigma = 0.25$	20.8(-66.8)	7.4(-69.2)	1.8(-62.8)	0.2(-50.2)	0.0(+0.0)
$\sigma = 0.5$	11.6(-62.0)	6.6(-58.8)	3.8(-51.8)	1.2(-44.8)	0.2(-37.2)
$\sigma = 1.0$	10.6(-44.4)	10.6(-37.4)	9.4(-31.4)	9.4(-23.6)	9.4(-18.8)

Table C: Certified accuracy of randomized smoothing on pretrained classifier ViT-B/16 at all σ for CIFAR-10

	Certified Accuracy at ϵ(%)					
Noise	0.0	0.5	1.0	1.5	2.0	3.0
$\sigma = 0.25$	73.2(-10.8)	55.8(-22.0)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)
$\sigma = 0.5$	7.8(-72.4)	4.6(-71.0)	3.2(-63.8)	1.0(-53.6)	0.0(+0.0)	0.0(+0.0)
$\sigma = 1.0$	0.0(-67.8)	0.0(-61.4)	0.0(-55.6)	0.0(-50.0)	0.0(-42.2)	0.0(-25.8)

Table D: Certified accuracy of randomized smoothing on pretrained classifier BEiT at all σ for ImageNet

	Certified Accuracy at $\epsilon(\%)$					
Noise	0.0	0.5	1.0	1.5	2.0	3.0
$\sigma = 0.25$	73.8(-10.2)	58.0(-19.8)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)
$\sigma = 0.5$	9.0(-71.2)	7.0(-68.6)	4.0(-63.0)	2.0(-52.6)	0.0(+0.0)	0.0(+0.0)
$\sigma = 1.0$	0.0(-67.8)	0.0(-61.4)	0.0(-55.6)	0.0(-50.0)	0.0(-42.2)	0.0(-25.8)

Table E: Certified accuracy of randomized smoothing on droppatch activated BEiT with 10 majority votes at all σ for ImageNet

	Certified Accuracy at $\epsilon(\%)$				
Noise	0.0	0.25	0.5	0.75	1.0
$\sigma = 0.25$	88.2(+0.6)	71.4(-5.2)	53.2(-11.4)	35.2(-15.2)	0.0(+0.0)
$\sigma = 0.5$	69.8(-3.8)	60.0(-5.4)	48.4(-7.2)	37.2(-8.8)	27.2(-10.2)
$\sigma = 1.0$	49.0(-6.0)	41.8(-6.0)	34.0(-6.8)	27.0(-6.0)	22.0(-6.2)

Table F: Certified accuracy of randomized smoothing on Gaussian augmentation-trained ViT at all σ on CIFAR-10

	Certified Accuracy at $\epsilon(\%)$					
Noise	0.0	0.25	0.5	0.75	1.0	Avg MV
$\sigma = 0.25$	92(+0.0)	77(+0.0)	60(+0.0)	48(-1.0)	0(+0.0)	3.84
$\sigma = 0.5$	74(+0.0)	65(+0.0)	53(-1.0)	45(+0.0)	40(+0.0)	4.43
$\sigma = 1.0$	53(+0.0)	46(+0.0)	42(+0.0)	31(+0.0)	25(+0.0)	5.49

Table G: Certified accuracy and average majority votes with 2 sample steps and $k = 3$ consensus threshold at all σ for CIFAR-10.

	Certified Accuracy at $\epsilon(\%)$						
Noise	0.0	0.5	1.0	1.5	2.0	3.0	Avg MV
$\sigma = 0.25$	78(+0.0)	74(+0.0)	0(+0.0)	0(+0.0)	0(+0.0)	0(+0.0)	3.34
$\sigma = 0.5$	75(+0.0)	69(+0.0)	61(+0.0)	47(+0.0)	0(+0.0)	0(+0.0)	3.89
$\sigma = 1.0$	60(+0.0)	54(+0.0)	50(+0.0)	41(+0.0)	32(+0.0)	23(+0.0)	5.23

Table H: Certified accuracy and average majority votes with 2 sample steps and $k = 3$ consensus threshold at all σ for ImageNet.

	Certified Accuracy at $\epsilon(\%)$										
	CIFAR-10					ImageNet					
Noise	0.0	0.25	0.5	0.75	1.0	0.0	0.5	1.0	1.5	2.0	3.0
$\sigma = 0.25$	87.6	74.8	59.2	44.6	0.0	78	74	0	0	0	0
$\sigma = 0.50$	73.2	62.6	52.6	41.8	34.0	75	69	58	47	0	0
$\sigma = 1.00$	53.4	44.0	35.8	30.2	24.4	60	54	49	39	30	22

Table I: Certified accuracy with 2 sampling steps and 5 vote numbers at all σ for both CIFAR-10 and ImageNet

		Certified Accuracy at ϵ(%)								
		CIFAR-10				ImageNet				
Method	Off-the-shelf	0.25	0.5	0.75	1.0	0.5	1.0	1.5	2.0	3.0
PixelDP (Lecuyer et al., 2019)	✗	$^{(71.0)}$22.0	$^{(44.0)}$2.0	-	-	$^{(33.0)}$16.0	-	-	-	-
RS (Cohen et al., 2019)	✗	$^{(75.0)}$61.0	$^{(75.0)}$43.0	$^{(65.0)}$32.0	$^{(65.0)}$23.0	$^{(67.0)}$49.0	$^{(57.0)}$37.0	$^{(57.0)}$29.0	$^{(44.0)}$19.0	$^{(44.0)}$12.0
SmoothAdv (Salman et al., 2019a)	✗	$^{(82.0)}$68.0	$^{(76.0)}$54.0	$^{(68.0)}$41.0	$^{(64.0)}$32.0	$^{(63.0)}$54.0	$^{(56.0)}$42.0	$^{(56.0)}$34.0	$^{(41.0)}$26.0	$^{(41.0)}$18.0
Consistency (Jeong & Shin, 2020)	✗	$^{(77.8)}$68.8	$^{(75.8)}$58.1	$^{(72.9)}$48.5	$^{(52.3)}$37.8	$^{(55.0)}$50.0	$^{(55.0)}$44.0	$^{(55.0)}$34.0	$^{(41.0)}$24.0	$^{(41.0)}$17.0
MACER (Zhai et al., 2020)	✗	$^{(81.0)}$71.0	$^{(81.0)}$59.0	$^{(66.0)}$46.0	$^{(66.0)}$38.0	$^{(68.0)}$57.0	$^{(64.0)}$43.0	$^{(64.0)}$31.0	$^{(48.0)}$25.0	$^{(48.0)}$14.0
Boosting (Horváth et al., 2021)	✗	$^{(83.4)}$70.6	$^{(76.8)}$60.4	$^{(71.6)}$**52.4**	$^{(73.0)}$**38.8**	$^{(65.6)}$57.0	$^{(57.0)}$44.6	$^{(57.0)}$38.4	$^{(44.6)}$28.6	$^{(38.6)}$21.2
SmoothMix (Jeong et al., 2021)	✓	$^{(77.1)}$67.9	$^{(77.1)}$57.9	$^{(74.2)}$47.7	$^{(61.8)}$37.2	$^{(55.0)}$50.0	$^{(55.0)}$43.0	$^{(55.0)}$38.0	$^{(40.0)}$26.0	$^{(40.0)}$17.0
Denoised (Salman et al., 2020)	✓	$^{(72.0)}$56.0	$^{(62.0)}$41.0	$^{(62.0)}$28.0	$^{(44.0)}$19.0	$^{(60.0)}$33.0	$^{(38.0)}$14.0	$^{(38.0)}$6.0	-	-
Lee (Lee, 2021)	✓	60.0	42.0	28.0	19.0	41.0	24.0	11.0	-	-
Carlini (Carlini et al., 2022)	✓	$^{(88.0)}$73.8	$^{(88.0)}$56.2	$^{(88.0)}$41.6	$^{(74.2)}$31.0	$^{(82.0)}$74.0	$^{(77.2.0)}$59.8	$^{(77.2)}$47.0	$^{(64.6)}$31.0	$^{(64.6)}$19.0
Ours	✓	$^{(87.6)}$**76.6**	$^{(87.6)}$**64.6**	$^{(87.6)}$50.4	$^{(73.6)}$37.4	$^{(84.0)}$**77.8**	$^{(80.2)}$**67.0**	$^{(80.2)}$**54.6**	$^{(67.8)}$**42.2**	$^{(67.8)}$**25.8**

Table J: Certified accuracy compared with existing works. The certified accuracy at $\epsilon = 0$ for each model is in the parentheses. The certified accuracy for each cell is from the respective papers except Carlini et al. (2022). Our diffusion model and classifier are the same as Carlini et al. (2022), where the off-the-shelf classifier uses ViT-based architectures trained on a large dataset (ImageNet-22k).

		Certified Accuracy at ϵ(%)					
Methods	Noise	0.0	0.5	1.0	1.5	2.0	3.0
Carlini (Carlini et al., 2022)	$\sigma = 0.25$	82.0	74.0	0.0	0.0	0.0	0.0
	$\sigma = 0.5$	77.2	71.8	59.8	47.0	0.0	0.0
	$\sigma = 1.0$	64.6	57.8	49.2	40.6	31.0	19.0
Ours	$\sigma = 0.25$	**84.0(+2.0)**	**77.8(+3.8)**	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)	0.0(+0.0)
	$\sigma = 0.5$	80.2(+3.0)	75.6(+3.8)	**67.0(+7.2)**	**54.6(+7.6)**	0.0(+0.0)	0.0(+0.0)
	$\sigma = 1.0$	67.8(+3.2)	61.4(+3.6)	55.6(+6.4)	50.0(+9.4)	**42.2(+11.2)**	**25.8(+6.8)**

Table K: Certified accuracy compared with Carlini et al. (2022) for ImageNet at all σ. The numbers in the bracket are the difference of certified accuracy between two methods. Our diffusion model and classifier are the same as Carlini et al. (2022).