Theoretical Understanding of Diffusion-Based Adversarial Purification

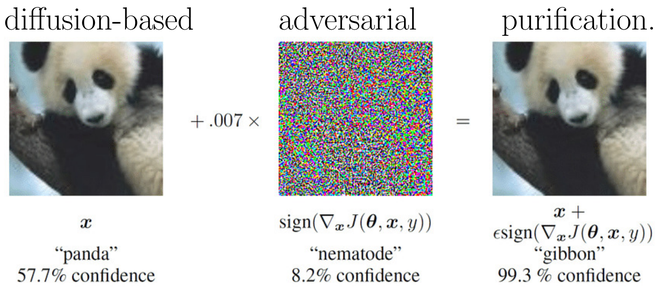
Dinesh Karthik M Bedartha Goswami
Machine Learning For Weather and Climate Lab, Data Science Department, IISER Pune

Problem Statement

Adversarial Examples pose a critical threat to deep neural networks:

- Small imperceptible perturbations $\|\delta\|_\infty \leq \epsilon$ cause misclassification
- Adversarial training is expensive and fails on unseen attacks
- Current defenses lack rigorous theoretical justification

Our Goal: Formulate the first rigorous theoretical framework for diffusion-based adversarial purification.



Notation & Definitions

- D: data distribution over $\mathcal{X} \times \mathcal{Y}$
- $S = \{(x_i, y_i)\}_{i=1}^n$: training set
- h_S: classifier; f_θ: feature map
- $\ell(\cdot, \cdot)$: loss function (0-1 or cross-entropy)
- $g(x; \xi)$: fixed diffusion purifier, $\xi \sim \mathcal{N}(0, \sigma^2 I)$
- L_f, L_ℓ: Lipschitz constants
- c_g: contraction coefficient; $\mathcal{M}$: data manifold

Statistical Learning Theory

- **Population Risk:** $R(f) = \mathbb{E}_{(x,y)\sim D}[\ell(f(x), y)]$
- **Empirical Risk:** $\hat{R}_S(f) = \frac{1}{n}\sum_{i=1}^n \ell(f(x_i), y_i)$
- **Uniform Stability:** Algorithm A is β_n-stable if:
$$\sup_{S,i,(x,y)} \left|\ell(h_S(x), y) - \ell(h_{S\setminus i \cup (x',y')}(x), y)\right| \leq \beta_n$$
- **Stability $\Rightarrow$ Generalization:** $|R(f) - \hat{R}_S(f)| = O\left(\sqrt{(\beta_n + 1/n)\log(1/\delta)}\right)$ w.p. $1 - \delta$

FGSM Attack

Fast Gradient Sign Method.
Construction: Maximize loss under L_∞ constraint:

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}\left(\nabla_x \ell(f(x), y)\right)$$

First-Order Approximation:

$$\ell(f(x+\delta), y) \approx \ell(f(x), y) + \nabla_x \ell(f(x), y)^\top \delta$$

Optimal perturbation:

$$\delta^* = \arg\max_{\|\delta\|_\infty \leq \epsilon} \nabla_x \ell(f(x), y)^\top \delta = \epsilon \cdot \text{sign}\left(\nabla_x \ell(f(x), y)\right)$$

Background: Diffusion Models (DDPM)

Forward Process (add noise):

$$q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}x_{t-1}, \beta_t I)$$

Reverse Process (denoise):

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}\left(\mu_\theta(x_t, t), \Sigma_\theta(x_t, t)\right)$$

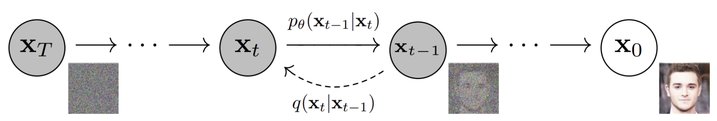
with mean:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}}\left(x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(x_t, t)\right)$$

Training objective (score matching):

$$\mathcal{L}_{\text{simple}} = \mathbb{E}_{x_0,t,\epsilon}\left\|\epsilon - \epsilon_\theta(x_t, t)\right\|_2^2$$

Sampling:
$x_T \sim \mathcal{N}(0, I) \rightarrow$ iteratively denoise $\rightarrow x_0$



DiffPure Method

Adversarial Purification via Diffusion
Given adversarial example $x_{\text{adv}} = x + \delta$:

1. Add noise: $x_T = x_{\text{adv}} + \tilde{\epsilon}$, $\tilde{\epsilon} \sim \mathcal{N}(0, \sigma_T^2 I)$
2. Reverse diffusion: $x_{t-1} \sim p_\theta(x_{t-1} \mid x_t)$ for $t = T, \ldots, 1$
3. Output: x_0^{pur} (likely on/near manifold $\mathcal{M}$)

Key mechanism: Adversarial perturbations appear as high-frequency noise; reverse diffusion trained on clean data removes them.

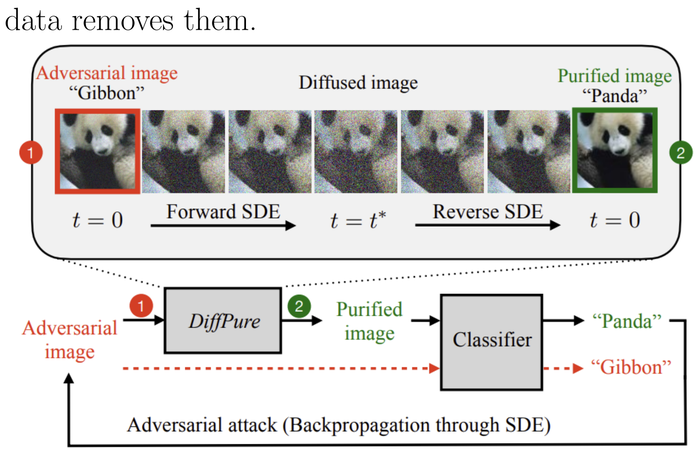


Figure 1: DiffPure adversarial purification pipeline.

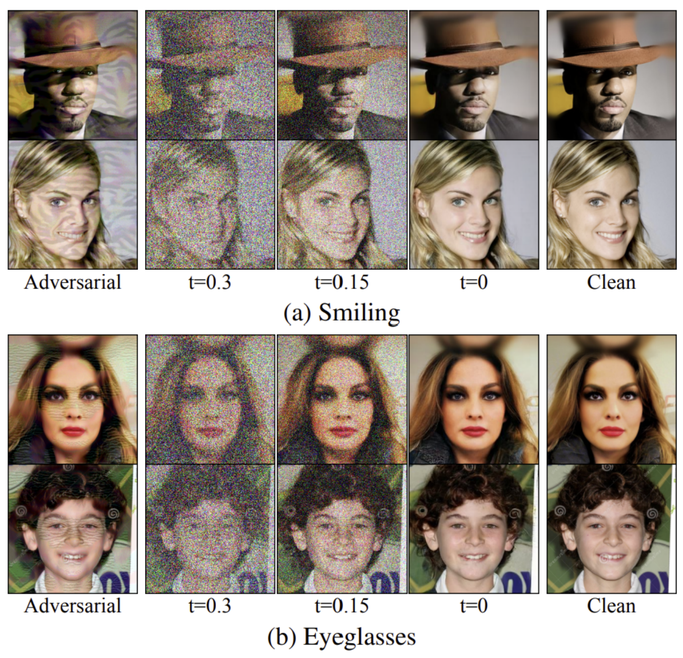


Figure 2: Diffusion-based purification recovers clean images from adversarial inputs. The reverse diffusion progressively removes adversarial perturbations while preserving semantic content.

Purified Classifier

Definition: Let h_S be classifier trained on clean data S. Define:

$$\boxed{h_S^{\text{pur}}(x; \xi) := h_S(g(x; \xi))}$$

where g is fixed diffusion purifier, $\xi \sim \mathcal{N}(0, \sigma^2 I)$ sampled at each inference.
Risks. The *purified risk* measures the expected loss of the stochastic classifier:

$$R_{\text{pur}}(h_S) \;:=\; \mathbb{E}_{(x,y)\sim D}\,\mathbb{E}_\xi\big[\,\ell\big(h_S(g(x;\xi)),\, y\big)\big]\,.$$

The *adversarial purified risk* evaluates the worst-case loss over all ℓ_∞ perturbations of size at most ϵ, while still averaging over purification noise:

$$R_{\text{adv}}^{\text{pur}}(h_S; \epsilon) \;:=\; \mathbb{E}_{(x,y)\sim D}\left[\sup_{\|\delta\|_\infty \leq \epsilon} \mathbb{E}_\xi\big[\,\ell\big(h_S(g(x+\delta;\xi)),\, y\big)\big]\right].$$

These quantities capture the clean and adversarial performance of the classifier when inputs are preprocessed by the diffusion-based purifier.

Core Assumptions

(A1) Bounded Loss: $0 \leq \ell(u, y) \leq B$
(A2) Lipschitz Continuity: (standard, enforceable)

$$\|f_\theta(x) - f_\theta(x')\| \leq L_f\|x - x'\|$$

$$|\ell(u, y) - \ell(u', y)| \leq L_\ell\|u - u'\|_2$$

(A3) Local Contraction:
For $x \in \mathcal{M}_\epsilon := \{x' : \text{dist}(x', \mathcal{M}) \leq \epsilon\}$ and $\|\delta\|_2 \leq \epsilon$:

$$\mathbb{E}_\xi\|g(x+\delta;\xi) - g(x;\xi)\|_2 \leq c_g\|\delta\|_2, \quad c_g \in (0, 1)$$

Critical: If $c_g \geq 1$, robustness bounds become vacuous. Must verify empirically.

Open Theoretical Questions

Q1. Stability After Purification.
Does the stochastic purifier preserve the algorithm's uniform stability?

$$\sup_{(x,y)} \mathbb{E}_\xi[|\ell(h_S(g(x;\xi)), y) - \ell(h_{S'}(g(x;\xi)), y)|] \overset{?}{\leq} \beta_n.$$

In other words, does purification keep the same stability parameter?

Q2. Generalization Bound.
Can purified risk be upper bounded by empirical risk plus stability and a purification-shift term?

$$R_{\text{pur}}(h_S) \overset{?}{\leq} \hat{R}_S(h_S) + O\Big(\beta_n + \Delta_g + \tfrac{1}{\sqrt{n}}\Big)$$

$$\Delta_g := \mathbb{E}\|g(x;\xi) - x\|_2.$$

Q3. Adversarial Robustness.

Does purification yield a Lipschitz-type robustness guarantee?

$$R_{\text{adv}}^{\text{pur}}(h_S; \epsilon) \overset{?}{\leq} R_{\text{pur}}(h_S) + L_\ell\, L_f\, c_g\, \epsilon.$$

All questions assume (A1)–(A3): Lipschitz loss, Lipschitz model, and bounded purifier drift.

Empirical Validation Plan and Furture Work

Setup: ResNet-50 (CIFAR-10), Stable Diffusion v1.5
Measure:

- $\Delta_g = \mathbb{E}_x[\|g(x;\xi) - x\|_2]$ (purifier shift)
- $c_g = \mathbb{E}[\|g(x+\delta;\xi) - g(x;\xi)\|/\|\delta\|]$ (contraction ratio)
- L_f (classifier Lipschitz via spectral norm)

Attacks: PGD, AutoAttack, DiffHammer
Metrics: Clean accuracy, purified accuracy, robust accuracy, inference time
Future Work:

- Prove Q1–Q3 rigorously with full lemmas
- Validate Assumption (A3) empirically (critical)

References

[1] Nie, W., et al. (2022). Diffusion Models for Adversarial Purification. *ICML*.
[2] Ho, J., et al. (2020). Denoising Diffusion Probabilistic Models. *NeurIPS*.
[3] Bousquet, O., & Elisseeff, A. (2002). Stability and Generalization. *JMLR*, 46, 499–526.
[4] Szegedy, C., et al. (2014). Intriguing Properties of Neural Networks. *ICLR*.

Email: mulumudi.dineshkarthik@students.iiserpune.ac.in