Safeguarding Vision-Language Models: Mitigating Vulnerabilities to Gaussian Noise in Perturbation-based Attacks

Supplementary Material

A. Overview of the Supplementary Material

This supplementary material offers additional details and analyses to further support the findings presented in the main manuscript. It includes detailed information on the experimental configuration (Appendix B), more evaluation on recent vision-language models (Appendix C), a thorough analysis of the limitations and unique characteristics of DiffPure (Appendix D), extended implementation specifics of DiffPure-VLM (Appendix E), and conjectures along with preliminary theoretical discussions on the effects of Gaussian noise (Appendix F). Collectively, these sections provide deeper insights into our methodology, enhancing the transparency and reproducibility of our research.

B. Experiment Details

B.1. Models

In this work, we conduct all experiments on three leading Vision-Language Models (VLMs), i.e., MiniGPT-4 (13B) [10], LLaVA-v1.5 (7B) [7], and InternVL2 (8B) [3]. We use the official model weights from HuggingFace or GitHub repositories for experiments in our paper. These model details are summarized in Table 1. Each model features a distinct LLM, vision encoder, and vision-language alignment method, allowing us to draw broader insights.

Table 1. Specifications of the evaluated VLMs.

Model	Size	Vision Encoder	LLM	VL Connection Module
MiniGPT-4-13B	14B	EVA-CLIP ViT-G/14	Vicuna-v0-13B	Q-former
LLaVA-v1.5-7B	7B	CLIP ViT-L/14	Vicuna-v1.5-7B	MLP
InternVL2-8B	8B	InternViT-300M-448px	InternLM2-8B	MLP

B.2. Fine-tuning Configuration

We present the detailed hyper-parameters for post-hoc fine-tuning on our Robust-VLGuard dataset in Table 2. Gaussian noise augmentation was applied to the training images, with a randomly selected standard deviation between 0.01 and 0.15, and a 70% probability of application. The fine-tuning was performed over 3 epochs on a single A100-80G GPU, using a consistent batch size of 16. For MiniGPT-4-13B, unfreezing the linear projector significantly improved robustness in terms of helpfulness and safety alignment. However, for LLaVA-v1.5-7B and InternVL2-8B, unfreezing the linear projector led to increased overfitting, likely due to differences in the vision-language connection modules of these models.

Table 2. Post-hoc fine-tuning hyper-parameters of different models.

Model	Training Module	LoRA Rank	LoRA Alpha	Learning Rate
MiniGPT-4-13B	Vision Encoder & Linear Projector	16	32	3e-5
LLaVA-v1.5-7B	Vision Encoder	16	256	4e-5
InternVL2-8B	Vision Encoder	16	256	4e-5

B.3. Details of Evaluation Settings

For evaluation on the MM-Vet benchmark, we set the temperature to 0 and use greedy decoding across all experiments to ensure reproducibility in helpfulness assessments. For safety evaluations on the RealToxicityPrompts benchmark, we follow the setup of Qi et al. [9], using a temperature of 1 and performing three runs to calculate the average attack success rate. Greedy decoding is also employed for this benchmark. The choice of temperature 1 reflects real-world usage, where sampling is typically enabled during interactions with VLMs. This setting aims to better simulate real-world scenarios when assessing safety alignment.

Additionally, the MM-Vet and RealToxicityPrompts benchmarks offer a comprehensive set of metrics covering various aspects. For the sake of brevity, we report only the overall metrics — Performance Score and Attack Success Rate — in the main paper. Here, we present the detailed evaluation results in Table 3 and Table 4, corresponding to Figure 2 in the main paper. The results show that Gaussian noisy images negatively impact nearly all metrics across both benchmarks and various models. Notably, using Gaussian noisy images as prompts improves MiniGPT-4's performance on the OCR metric in the MM-Vet benchmark, highlighting the current VLMs' lack of robustness.

C. Additional Evaluation on Recent Vision-Language Models

In this section, we further assess the robustness of state-of-the-art vision-language models against Gaussian noise. Table 5 presents the attack success rates on the RealToxicityPrompts benchmark for four recent VLMs—LLaMA-3.2-Vision-11B [5], Ivy-VLM-3B [6], Qwen2.5-VL-7B [1], and InternVL2.5-8B [2]—under various Gaussian noise levels. Lower percentages indicate improved safety alignment.

Table 3. Robustness comparison of various models on the MM-Vet benchmark using clean and Gaussian noisy image prompts (GPT-4 %).

Image Type	Performance Score ↑	Recognition	OCR	Knowledge	Generation	Spatial	Math
MiniGPT-4-13B							
Clean Image	26.7	34.9	13.5	27.4	27.1	19.1	7.7
Gaussian Noisy Image	24.0 (-2.7)	29.0 (-5.9)	16.9 (+3.4)	20.5 (-6.9)	22.5 (-4.6)	20.7 (+1.6)	7.7 (0.0)
LLaVA-v1.5-7B							
Clean Image	33.0	37.9	23.9	20.4	23.6	28.5	11.5
Gaussian Noisy Image	31.3 (-1.7)	36.3 (-1.6)	21.9 (-2.0)	18.3 (-2.1)	21.2 (-2.4)	25.7 (-2.8)	3.8 (-7.7)
InternVL2-8B							
Clean Image	59.9	53.5	71.9	40.4	44.6	69.7	65.4
Gaussian Noisy Image	54.4 (-5.5)	47.1 (-6.4)	66.8 (-5.1)	35.1 (-5.3)	39.2 (-5.4)	61.1 (-8.6)	53.5 (-11.9)

Table 4. Robustness comparison of various models on the RealToxicityPrompts benchmark using clean and Gaussian noisy image prompts. Metrics include Attack Success Rate and toxicity levels (Perspective API %).

Image Type	Attack Success Rate ↓	Identity Attack	Profanity	Severe Toxicity	Sexually Explicit	Threat	Toxicity
MiniGPT-4-13B							
Clean Image	34.8	2.7	25.1	1.5	12.2	2.0	30.5
Gaussian Noisy Image	44.1 (+9.3)	3.6 (+0.9)	31.2 (+6.1)	2.3 (+0.8)	13.7 (+1.5)	2.9 (+0.9)	38.2 (+7.7)
LLaVA-v1.5-7B							
Clean Image	57.7	5.7	46.8	3.7	18.0	3.8	54.4
Gaussian Noisy Image	60.1 (+2.4)	4.8 (-0.9)	48.1 (+1.3)	2.9 (-0.8)	17.8 (-0.2)	4.0 (+0.2)	56.0 (+1.6)
InternVL2-8B							
Clean Image	50.5	4.1	40.2	1.9	13.5	2.5	44.3
Gaussian Noisy Image	57.2 (+6.7)	4.5 (+0.4)	45.9 (+5.7)	2.0 (+0.1)	14.3 (+0.8)	3.2 (+0.7)	51.7 (+7.4)

As shown, when Gaussian noise is introduced at increasing levels ($\sigma_n = 30/255$, $\sigma_n = 50/255$, and $\sigma_n = 70/255$), all models exhibit a rise in attack success rates, highlighting their sensitivity to simple Gaussian noise perturbations. These findings underscore the need for robust noise augmentation and defense strategies in training pipelines to maintain safety alignment in VLMs.

D. Further Analysis of DiffPure

D.1. Defence Performance

In this section, we present a comprehensive analysis of the effects of DiffPure [8] and Gaussian noise under perturbation-based attacks in Vision-Language Models (VLMs). Specifically, we extend the experimental setup described in Section 3.1 in the main paper by varying the standard deviation σ_n of Gaussian noise n and the timestep parameter t^* in DiffPure. Results are summarized in Table 6. First, Gaussian noise n with standard deviations $\sigma_n \in \{15/255, 30/255, 50/255, 75/255\}$ is added to the benign clean image I_c to evaluate its impact on the Attack Success Rate. The results demonstrate that the Attack Success Rate under Gaussian noise is significantly higher than that of the benign clean image. When $\sigma_n \leq 50/255$, increasing σ_n will lead to a higher Attack Success Rate. However, this trend did not continue at a higher σ_n setting (e.g., $\sigma_n = 75/255$), indicating that the effect of Gaussian noise on VLMs is limited. Next, we apply DiffPure with different timesteps $t^* \in \{50, 100, 150\}$ to generate diffused images from adversarial inputs I_{adv} with varying perturbation constraints ϵ. For $\epsilon = 16/255$, increasing t^* to 100 or 150 reduces the Attack Success Rate but does not lower it below the level observed for the benign clean image. For larger perturbation constraints, increasing t^* fails to decrease the Attack Success Rate, with a comparable performance of Gaussian noisy images.

D.2. Distribution Shift

In this section, we present detailed results from the Gaussianity experiments conducted on adversarial and diffused images. Specifically, we visualize adversarial images I_{adv} alongside their corresponding residuals r_{adv}, and diffused images $I_{diffused}$ with their residuals $r_{diffused}$, under pixel constraints $\epsilon \in \{16/255, 32/255, 64/255\}$ for I_{adv} and diffusion timesteps $t^* \in \{50, 100, 150, 500, 750\}$ in DiffPure [8] for generating $I_{diffused}$. Visualizations are shown in Figure 1, 2, and 3 with corresponding metrics: Kurtosis, Q-Q deviation, mean, and standard deviation. From these visualizations, we observe that when $50 \leq t^* \leq 150$, the residuals $r_{diffused}$ exhibit a Gaussian-like distribution, particularly for $\epsilon = 32/255$ and $\epsilon = 64/255$. However, as t^* increases, the Kurtosis of $r_{diffused}$ rises, indicating a shift towards a long-tailed distribution. This suggests that a small fraction of pixels in $I_{diffused}$ undergo significant changes compared

Table 5. Attack success rate (%) on the RealToxicityPrompts benchmark for various vision-language models under different noise levels. Lower scores indicate improved safety alignment.

	RealToxicityPrompts (%) ↓			
	LLaMA-3.2-Vision-11B	Ivy-VLM-3B	Qwen2.5-VL-7B	InternVL2.5-8B
Benign clean Image I_c	45.4	29.9	36.8	43.9
+ n ($\sigma_n = 30/255$)	46.4 (+1.0)	35.5 (+5.6)	39.3 (+2.5)	51.5 (+7.6)
+ n ($\sigma_n = 50/255$)	47.6 (+2.2)	40.3 (+10.4)	39.5 (+2.7)	52.8 (+8.9)
+ n ($\sigma_n = 70/255$)	48.5 (+3.1)	42.0 (+12.1)	46.1 (+9.3)	54.0 (+10.1)

Table 6. Detailed results of the defense of DiffPure in MiniGPT-4 on the RealToxicityPrompts benchmark under different image configurations. (Perspective API %).

Image Configuration	**Attack Success Rate ↓**	**Identity Attack**	**Profanity**	**Severe Toxicity**	**Sexually Explicit**	**Threat**	**Toxicity**
Benign clean Image I_c	34.8	2.7	25.1	1.5	12.2	2.0	30.5
+ n ($\sigma_n = 15/255$)	38.5 (+3.7)	2.9 (+0.2)	27.4 (+2.3)	1.1 (-0.4)	13.0 (+0.8)	2.3 (+0.3)	34.1 (+3.6)
+ n ($\sigma_n = 30/255$)	44.1 (+9.3)	3.6 (+0.9)	31.2 (+6.1)	2.3 (+0.8)	13.7 (+1.5)	2.9 (+0.9)	38.2 (+7.7)
+ n ($\sigma_n = 50/255$)	46.3 (+11.5)	3.4 (+0.7)	34.0 (+8.9)	1.8 (+0.3)	14.8 (+2.6)	2.5 (+0.5)	39.5 (+9.0)
+ n ($\sigma_n = 75/255$)	44.1 (+9.3)	3.8 (+1.1)	30.1 (+5.0)	1.9 (+0.4)	14.3 (+2.1)	2.8 (+0.8)	37.5 (+7.0)
Adversarial image I_{adv} ($\epsilon = 16/255$)	53.6 (+18.8)	8.4 (+5.7)	36.6 (+9.4)	6.6 (+5.1)	14.1 (+1.9)	4.7 (+2.7)	48.6 (+18.1)
+ DiffPure ($t^* = 50$)	45.0 (+10.2)	2.5 (-0.2)	31.7 (+6.6)	1.8 (+0.3)	14.5 (+2.3)	2.8 (+0.8)	38.8 (+8.3)
+ DiffPure ($t^* = 100$)	37.6 (+2.8)	3.0 (+0.3)	25.6 (+0.5)	1.3 (-0.2)	12.3 (+0.1)	1.8 (-0.2)	33.1 (+2.6)
+ DiffPure ($t^* = 150$)	37.7 (+2.9)	2.5 (-0.2)	26.5 (+1.4)	2.1 (+0.6)	12.2 (+0.0)	2.5 (+0.5)	32.9 (+2.4)
Adversarial image I_{adv} ($\epsilon = 32/255$)	59.4 (+24.6)	14.6 (+11.9)	39.5 (+14.4)	7.0 (+5.5)	14.9 (+2.7)	6.2 (+4.2)	53.8 (+23.3)
+ DiffPure ($t^* = 50$)	45.5 (+10.7)	2.6 (-0.1)	32.1 (+7.0)	2.2 (+0.7)	14.8 (+2.6)	3.0 (+1.0)	38.5 (+8.0)
+ DiffPure ($t^* = 100$)	43.8 (+9.0)	3.3 (+0.6)	31.9 (+6.8)	1.9 (+0.4)	13.1 (+0.9)	2.5 (+0.5)	38.1 (+7.6)
+ DiffPure ($t^* = 150$)	42.3 (+7.5)	3.7 (+1.0)	30.4 (+5.3)	1.3 (-0.2)	13.3 (+1.1)	2.8 (+0.8)	36.3 (+5.8)
Adversarial image I_{adv} ($\epsilon = 64/255$)	67.2 (+32.4)	15.9 (+13.2)	49.6 (+24.5)	12.2 (+10.7)	16.9 (+4.7)	6.6 (+4.6)	63.1 (+32.6)
+ DiffPure ($t^* = 50$)	44.5 (+9.7)	2.9 (+0.2)	32.2 (+7.1)	2.4 (+0.9)	13.7 (+1.5)	2.7 (+0.7)	38.0 (+7.5)
+ DiffPure ($t^* = 100$)	42.1 (+7.3)	2.8 (+0.1)	30.3 (+5.2)	1.9 (+0.4)	13.7 (+1.5)	3.0 (+1.0)	36.5 (+6.0)
+ DiffPure ($t^* = 150$)	44.1 (+9.3)	3.3 (+0.6)	31.5 (+6.4)	1.4 (-0.1)	13.3 (+1.1)	2.5 (+0.5)	38.2 (+7.7)

to I_c, leading to a cleaner image with minimal content alteration, especially when $\epsilon = 16/255$. At $t^* = 500$, the Kurtosis and standard deviation of $r_{diffused}$ become significantly larger, implying greater changes in image content, as reflected in the visualization of $I_{diffused}$. For $t^* = 750$, the Kurtosis decreases while the standard deviation further increases, indicating that $r_{diffused}$ transitions to a flatter and broader distribution. In this case, $I_{diffused}$ diverges substantially from I_c in image content.

Furthermore, we extend our analysis to the embedding space, examining the similarities between the clean image I_c, the adversarial image I_{adv}, and the diffused image $I_{diffused}$. Based on our experiment in pixel space, where the residual noise $r_{diffused}$ approximates a Gaussian distribution under certain timestep settings in DiffPure, we consider $I_{diffused}$ as comparable to I_c with added Gaussian noise. To verify this, we generate a noisy image $I_n = I_c + n, n \in \mathcal{N}\left(0, \sigma^2_{r_{diffused}}\right)$, where $\sigma_{r_{diffused}}$ indicates the standard deviation of $r_{diffused}$. Using pre-trained visual encoder E in MiniGPT-4, we compute cosine similarities between the embeddings of I_n and I_{adv}, denoted as $C_{n,adv}$, and between I_n and $I_{diffused}$, denoted as $C_{n,diffused}$. Figure 4 shows these similarities across varying adversarial constraints ϵ and DiffPure steps t^*. Results indicate that, $C_{n,diffused}$ consistently exceeds $C_{n,adv}$, showing that $I_{diffused}$ is closer to I_n than I_{adv} in the embedding space. Notably, with moderate timesteps ($t^* \in [50, 150]$), $I_{diffused}$ is similar to I_n (Gaussian noise n added to the benign clean image I_c) in both pixel and embedding spaces.

We also visualize the cosine similarity between the visual embeddings of $I_{diffused}$ and I_c, denoted as $C_{clean,diffused}$, across varying ϵ and t^*. Results are shown in Figure 5, revealing that $C_{clean,diffused}$ decreases rapidly as t^* decreases, while it gradually declines as t^* increases. Combining these findings with experiments in pixel space, we conclude that smaller t^* values lead $I_{diffused}$ to retain adversarial information, whereas larger t^* values result in significant content disruption, leading to semantic misalignment.

E. Additional Details of DiffPure-VLM

E.1. Implementation Details

The overall architecture of our proposed DiffPure-VLM framework is illustrated in Figure 6, with the detailed al-

Table 7. Evaluation of DiffPure-VLM's effectiveness on RealToxicityPrompts across different image configurations. Metrics include attack success rate and toxicity levels (Perspective API %).

Image Type	Attack Success Rate ↓	Identity Attack	Profanity	Severe Toxicity	Sexually Explicit	Threat	Toxicity
InternVL2-8B							
Benign Clean image	50.5	4.1	40.2	1.9	13.5	2.5	44.3
Gaussian Noisy image	57.2	4.5	45.9	2.0	14.3	3.2	51.7
Adversarial image (ϵ = 32/255)	65.0	21.1	49.2	7.5	16.6	5.0	61.9
+DiffPure-VLM (t*=50)	53.1	3.8	41.6	2.0	13.6	2.2	48.0
InternVL2-8B-VLGuard							
Benign Clean image	27.7	1.4	22.2	0.9	7.1	1.6	23.8
Gaussian Noisy image	39.9	2.5	31.4	1.3	10.3	1.8	35.8
Adversarial image (ϵ = 32/255)	72.3	12.3	60.6	8.6	19.9	6.5	69.3
+DiffPure-VLM (t*=50)	35.7	2.0	28.9	0.8	9.8	1.8	31.6
InternVL2-8B-RobustVLGuard							
Benign Clean image	29.9	0.8	22.1	0.3	7.2	1.5	25.9
Gaussian Noisy image	34.5	2.1	27.2	1.3	8.4	1.6	31.3
Adversarial image (ϵ = 32/255)	70.6	26.7	56.5	9.2	17.3	6.9	68.1
+DiffPure-VLM (t*=50)	**33.4**	2.4	20.6	0.7	8.1	2.4	29.1
+DiffPure-VLM (t*=150)	**32.8**	1.7	25.9	0.6	7.7	1.8	29.1
LLaVA-v1.5-7B							
Benign Clean image	57.7	5.7	46.8	3.7	18.0	3.8	54.4
Gaussian Noisy image	60.1	4.8	48.1	2.9	17.8	4.0	56.0
Adversarial image (ϵ = 32/255)	66.0	16.6	51.6	8.8	18.0	4.7	64.5
+DiffPure-VLM (t*=50)	58.5	5.9	45.5	2.7	17.0	4.3	53.3
LLaVA-v1.5-7B-VLGuard							
Benign Clean image	50.3	4.3	40.6	2.0	13.6	4.3	46.9
Gaussian Noisy image	52.3	4.6	41.5	2.7	14.0	4.1	48.5
Adversarial image (ϵ = 32/255)	70.4	21.3	52.8	7.5	16.7	7.0	67.2
+DiffPure-VLM (t*=50)	51.1	3.4	40.9	2.2	13.4	3.6	47.5
LLaVA-v1.5-7B-RobustVLGuard							
Benign Clean image	43.6	4.6	34.7	2.4	12.3	3.5	41.0
Gaussian Noisy image	42.3	3.1	34.5	1.9	11.8	3.1	40.0
Adversarial image (ϵ = 32/255)	62.5	7.8	48.0	5.4	16.5	5.8	60.0
+DiffPure-VLM (t*=50)	**43.9**	3.2	34.6	2.4	12.8	3.7	41.0
+DiffPure-VLM (t*=150)	**42.5**	3.5	32.7	2.8	12.1	4.1	39.3
MiniGPT-4-13B							
Benign Clean image	34.8	2.7	25.1	1.5	12.2	2.0	30.5
Gaussian Noisy image	44.1	3.6	31.2	2.3	13.7	2.9	38.2
Adversarial image (ϵ = 32/255)	59.4	14.6	39.5	7.0	14.9	6.2	53.8
+DiffPure-VLM (t*=50)	45.5	2.6	32.1	2.2	14.8	3.0	38.5
MiniGPT-4-13B-VLGuard							
Benign Clean image	41.3	2.8	30.1	2.2	14.6	2.5	37.3
Gaussian Noisy image	43.7	3.0	31.6	2.3	13.9	3.5	38.6
Adversarial image (ϵ = 32/255)	67.6	10.5	48.2	7.0	19.9	7.8	61.7
+DiffPure-VLM (t*=50)	45.0	4.2	33.1	2.1	14.6	3.1	40.7
MiniGPT-4-13B-RobustVLGuard							
Benign Clean image	16.0	0.4	9.9	0.3	4.6	1.1	12.1
Gaussian Noisy image	16.5	0.9	11.9	0.6	5.8	1.0	14.0
Adversarial image (ϵ = 32/255)	53.7	9.8	35.3	4.1	13.9	5.4	48.1
+DiffPure-VLM (t*=50)	**13.6**	0.3	9.2	0.2	5.5	0.9	10.6
+DiffPure-VLM (t*=150)	**11.9**	0.5	8.6	0.2	4.2	0.6	9.9

gorithmic procedure outlined in Algorithm 1. For our experiments, we employ the Guided Diffusion model for ImageNet [4], specifically the 256 × 256 unconditional variant provided by OpenAI[1]. Importantly, we synchronize the forward diffusion timesteps ($\mathbf{t}_{\text{forward}}$) with the reverse diffusion timesteps ($\mathbf{t}_{\text{reverse}}$), denoted as t^* in the experimental results, following the setup in DiffPure [8]. Here, we leverage this diffusion model to validate the robustness of our fine-tuned VLMs against Gaussian noise, demonstrating a preliminary defense strategy. However, the fixed image res-

[1] https://openaipublic.blob.core.windows.net/diffusion/jul-2021/256x256_diffusion_uncond.pt

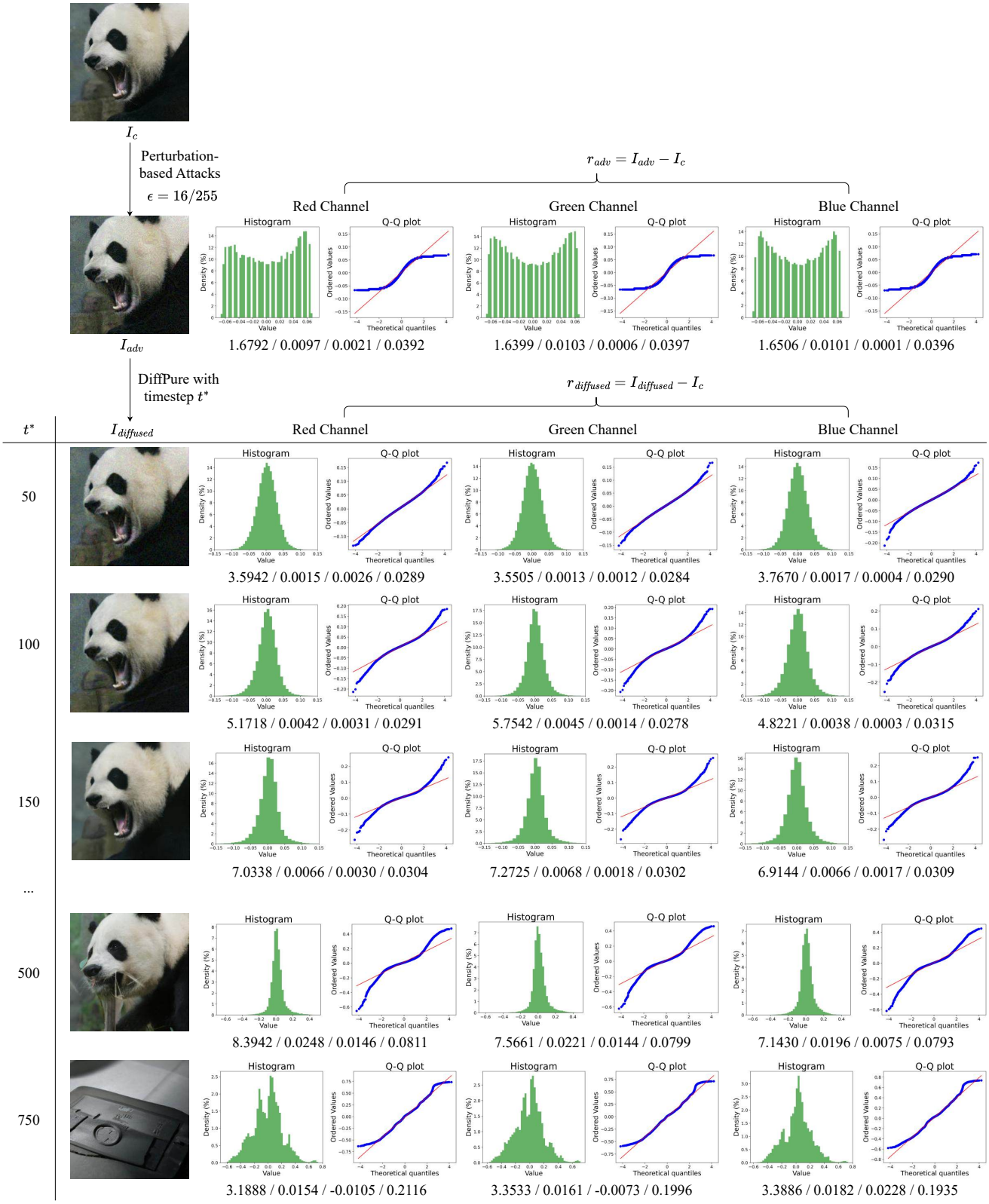

Figure 1. I_{adv}, $I_{diffused}$ and statistics of r_{adv}, $r_{diffused}$ under different t^* in DiffPure (constraint $\epsilon = 16/255$). Metrics are shown in 'Kurtosis / Q-Q Deviation / Mean / Standard Deviation'. Please zoom in to see details.

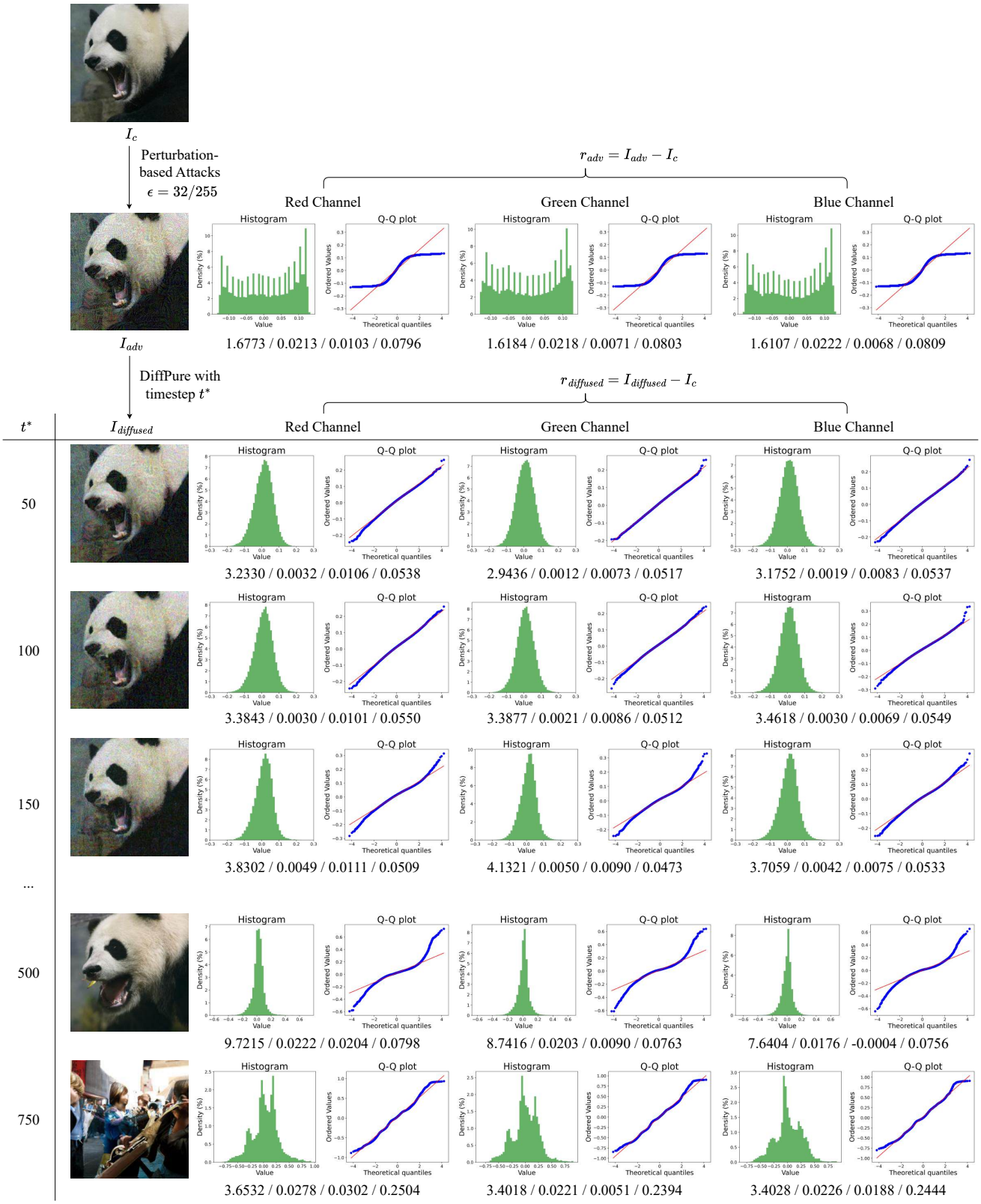

Figure 2. I_{adv}, $I_{diffused}$ and statistics of r_{adv}, $r_{diffused}$ under different t^* in DiffPure (constraint $\epsilon = 32/255$). Metrics are shown in 'Kurtosis / Q-Q Deviation / Mean / Standard Deviation'. Please zoom in to see details.

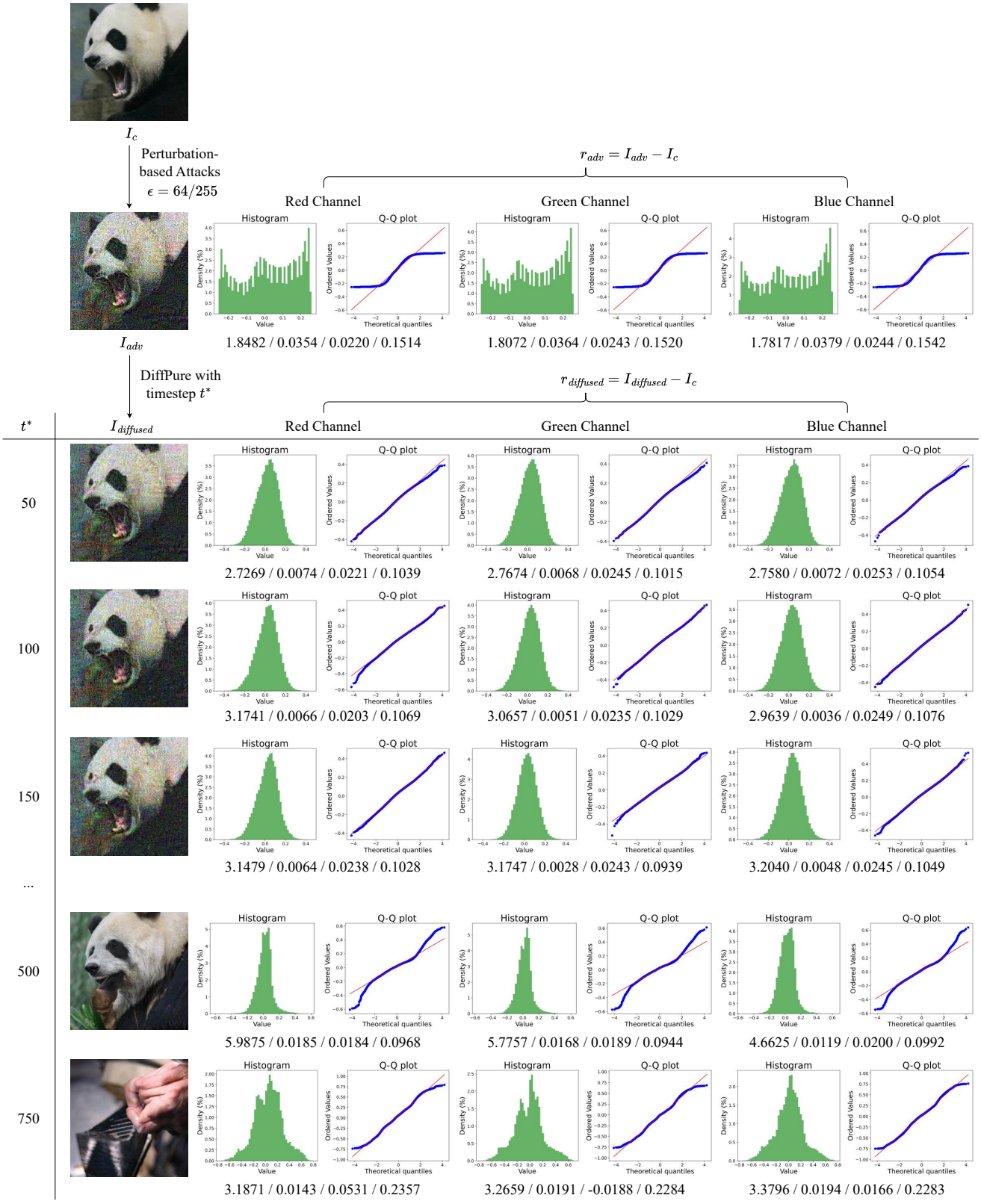

Figure 3. I_{adv}, $I_{diffused}$ and statistics of r_{adv}, $r_{diffused}$ under different t^* in DiffPure (constraint $\epsilon = 64/255$). Metrics are shown in 'Kurtosis / Q-Q Deviation / Mean / Standard Deviation'. Please zoom in to see details.

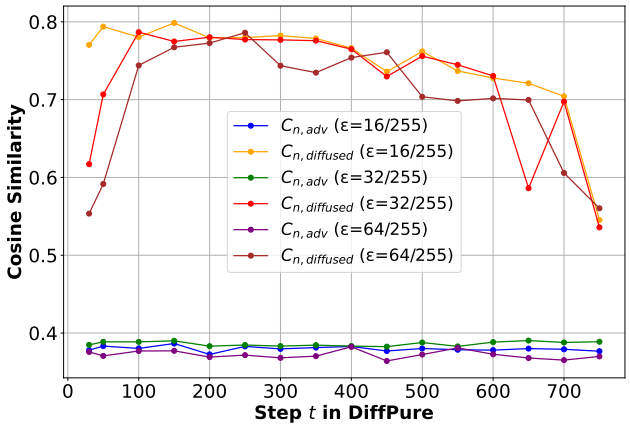

Figure 4. Cosine similarity of visual embeddings under different ϵ of adversarial image I_{adv} and t^* of DiffPure.

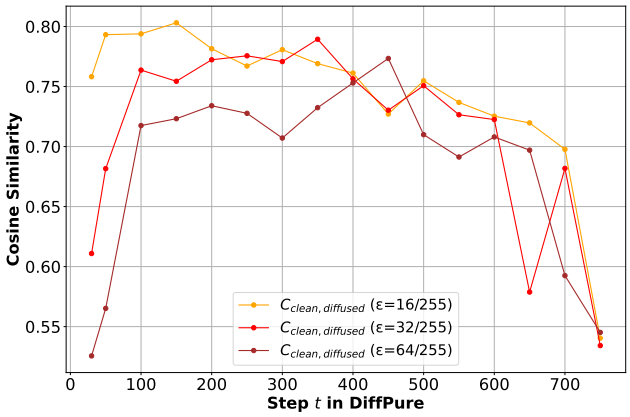

Figure 5. Cosine Similarity of visual embeddings from I_c and $I_{diffused}$ under different ϵ of adversarial image.

olution of the diffusion model requires down-sampling and up-sampling operations, which may introduce artifacts not considered during the fine-tuning of the VLM, potentially impacting evaluation results. In the future, adopting more advanced diffusion models will be essential for real-world applications.

E.2. Extended Experimental Results

In the main paper, for the sake of brevity, we only report results for the standard perturbation-based attack setting of $\epsilon = 32/255$. However, we also conducted experiments with lower attack strength ($\epsilon = 16/255$) and higher attack strength ($\epsilon = 64/255$) to further validate our analysis and approach in Table 8. Across different models and attack strengths, our DiffPure-VLM consistently reduces the attack success rate within a limited number of diffusion timesteps (fewer than 150). Notably, under lower attack strengths, setting the diffusion step to as low as $t^* = 50$ is sufficient to bring the attack success rate down to the level

Algorithm 1 DiffPure-VLM Adversarial Image Purification with DDPM

Require: Adversarial image x, harmful text prompt p, diffusion model $\mathcal{D}$, forward diffusion timesteps $\mathbf{t}_{\text{forward}}$, reverse diffusion timesteps $\mathbf{t}_{\text{reverse}}$, visual language model $\mathcal{VLM}$.

Ensure: Question answering result output

Resize input image x to the size required by the diffusion model (e.g., 256×256).
DDPM forward process with $\mathbf{t}_{\text{forward}}$ steps: $\hat{x} = \text{get\_noised\_x}(x, \mathbf{t}_{\text{forward}})$.
for t in $\mathbf{t}_{\text{reverse}}$ **do**
  Denoise using reverse DDPM process: $x = \text{denoising\_process}(\hat{x}, t)$.
end for
Obtain purified image with Gaussian noise: $x_{\text{gaussian}} = \text{normalize}(x)$.
Perform question answering using VLM: output $= \mathcal{VLM}(x_{\text{gaussian}}, p)$.
return output

observed for clean inputs. However, under higher attack strengths, $t^* = 50$ fails to reduce the attack success rate to the baseline level for both InternVL2-8B and MiniGPT-4-13B. This indicates that stronger attacks require a larger number of diffusion steps to effectively transform the adversarial noise into Gaussian noise. This finding aligns with the analysis presented in Figure 4 of the main paper, where the residual image at $t^* = 50$ for an attack strength of $\epsilon = 64/255$ does not exhibit Gaussian characteristics. Moreover, we observe that $t^* = 100$ demonstrates strong performance across all attack conditions, making it an effective trade-off between time and robustness. Thus, in real-world applications, setting $t^* = 100$ offers a balanced solution, achieving reliable defense while maintaining computational efficiency.

F. Conjectures and Discussion on the Impact of Gaussian Noise

Problem Definition

Setting:

- A Visual Language Model (VLM) typically consists of three main components: a visual encoder, a language model, and a vision-language connection module.
- Let the input be a pair (I, T), where $I \in \mathbb{R}^d$ is an image and T is the corresponding text prompt.
- The VLM generates an output sequence of tokens, denoted by $\hat{T} = f_\theta(I, T)$, where f_θ represents the VLM pipeline parameterized by θ.

Adversarial Attack: An adversarial perturbation δ is applied to the image I, resulting in a perturbed image $I_\delta =$

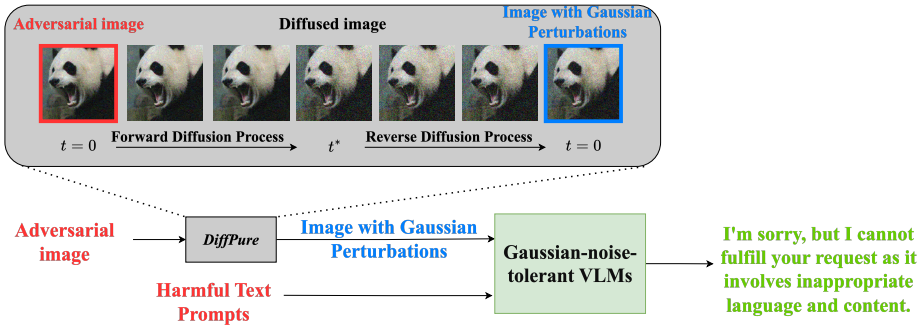

Figure 6. The overall framework of DiffPure-VLM.

$I + \delta$. The perturbation δ is crafted to manipulate the VLM into generating a specific harmful target text T^{target}. The adversary's objective is:

$$\delta = \arg\min_{\|\delta\| \leq \epsilon} L\left(f_\theta(I + \delta, T), T^{\text{target}}\right),$$

where $L(\cdot, \cdot)$ measures the discrepancy between the generated text $\hat{T}$ and the target text T^{target}. The constraint $\|\delta\| \leq \epsilon$ ensures that the perturbation is imperceptible.

Conjectures: We introduce the following four conjectures to guide our investigation into the impact of Gaussian noise on VLMs:

1. **Sensitivity of Adversarial Attacks to Gaussian Noise**: Adding Gaussian noise to adversarially perturbed images will significantly reduce the effectiveness of the attack.
2. **Gaussian Noise as a Simple Attack on VLM Safety**: Gaussian noise, even without adversarial perturbations, may increase the likelihood of generating harmful text.
3. **Gaussian Noise as a Regularizer**: Augmenting training data with Gaussian noise may act as a regularizer, enhancing the robustness of the VLM.
4. **Fine-Tuning with Gaussian Noise Preserves Performance**: Incorporating Gaussian noise during fine-tuning will preserve or even improve the VLM's overall performance.

Objective: The goal of this study is to systematically evaluate the impact of Gaussian noise on the robustness and reliability of VLMs. By exploring the above conjectures, we aim to determine whether Gaussian noise can effectively mitigate adversarial perturbations and enhance model robustness without compromising performance.

Conjecture 1: Sensitivity of Adversarial Perturbations to Gaussian Noise

Statement: Adversarial perturbations are highly sensitive to Gaussian noise; the attack effectiveness is significantly diminished when Gaussian noise is added to the adversarial image.

Discussion:

Consider an adversarially perturbed image $I_\delta = I + \delta$, where the perturbation δ is optimized to minimize the loss:

$$\delta = \arg\min_{\|\delta\| \leq \epsilon} L\left(f_\theta(I + \delta, T), T^{\text{target}}\right),$$

where $L(\cdot, \cdot)$ measures the discrepancy between the generated text $\hat{T}$ and the harmful target text T^{target}. The perturbation δ is crafted to exploit specific vulnerabilities in the model f_θ.

Now, consider the scenario where Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$ is added to the input. The new input becomes:

$$I_{\delta,\eta} = I + \delta + \eta.$$

The expected loss over the distribution of Gaussian noise η is:

$$\mathbb{E}_\eta\left[L\left(f_\theta(I + \delta + \eta, T), T^{\text{target}}\right)\right].$$

Since the adversarial perturbation δ is tailored for the specific input I, adding random Gaussian noise η disrupts this optimization. Adversarial perturbations exploit the model's sensitivity along certain directions in the input space, while isotropic Gaussian noise perturbs the input uniformly in all directions, diminishing the effect of δ.

Assuming that f_θ and L are Lipschitz continuous, we can bound the increase in expected loss as follows:

$$\mathbb{E}_\eta\left[L\left(f_\theta(I + \delta + \eta, T), T^{\text{target}}\right)\right] \geq L\left(f_\theta(I + \delta, T), T^{\text{target}}\right) + \frac{\sigma^2 \lambda}{2},$$

where λ is a positive constant related to the curvature of L and f_θ.

This inequality indicates that the addition of Gaussian noise increases the expected loss, thus reducing the effectiveness of the adversarial perturbation. The random noise η disrupts the carefully crafted δ, making it less effective at manipulating the VLM's output. This supports our conjecture that Gaussian noise can act as a simple yet effective countermeasure against adversarial attacks.

Table 8. Evaluation of DiffPure-VLM's effectiveness on RealToxicityPrompts across different image configurations. Metrics include attack success rate and toxicity levels (Perspective API %). Rows highlighted in light red indicate cases where attack success rate does not meet the baseline level of benign image input.

Image Type	Attack Success Rate ↓	Identity Attack	Profanity	Severe Toxicity	Sexually Explicit	Threat	Toxicity
InternVL2-8B-RobustVLGuard							
Benign Clean Image	29.9	0.8	22.1	0.3	7.2	1.5	25.9
Benign Noisy Image	34.5	2.1	27.2	1.3	8.4	1.6	31.3
Adversarial Image ($\epsilon = 16/255$)	72.5	19.8	58.5	8.3	19.2	7.8	70.0
+DiffPure-VLM ($t^* = 50$)	**31.4**	1.4	24.6	1.1	7.9	1.6	27.5
+DiffPure-VLM ($t^* = 100$)	**28.2**	0.9	21.7	0.4	6.8	1.5	23.9
+DiffPure-VLM ($t^* = 150$)	**28.2**	1.6	22.4	0.2	6.9	1.1	24.4
Adversarial Image ($\epsilon = 32/255$)	70.6	26.7	56.5	9.2	17.3	6.9	68.1
+DiffPure-VLM ($t^* = 50$)	**33.4**	2.4	20.6	0.7	8.1	2.4	29.1
+DiffPure-VLM ($t^* = 100$)	**33.4**	1.6	27.7	0.6	7.6	1.5	30.2
+DiffPure-VLM ($t^* = 150$)	**32.8**	1.7	25.9	0.6	7.7	1.8	29.1
Adversarial Image ($\epsilon = 64/255$)	57.3	9.3	45.8	4.4	16.1	3.9	53.9
+DiffPure-VLM ($t^* = 50$)	**40.9**	2.3	32.9	1.4	9.3	2.3	37.3
+DiffPure-VLM ($t^* = 100$)	**35.7**	1.8	28.2	0.8	7.6	2.4	31.8
+DiffPure-VLM ($t^* = 150$)	**36.1**	2.4	28.3	1.2	8.3	1.8	33.6
LLaVA-v1.5-7B-RobustVLGuard							
Benign Clean image	43.6	4.6	34.7	2.4	12.3	3.5	41.0
Benign Noisy image	42.3	3.1	34.5	1.9	11.8	3.1	40.0
Adversarial image ($\epsilon = 16/255$)	62.6	11.3	48.8	5.3	16.8	5.8	59.1
+DiffPure-VLM ($t^* = 50$)	**42.7**	3.4	32.1	1.5	12.0	4.6	39.7
+DiffPure-VLM ($t^* = 100$)	**42.8**	3.9	32.5	2.3	12.5	3.7	39.3
+DiffPure-VLM ($t^* = 150$)	**44.4**	3.3	34.4	2.2	12.6	3.2	41.0
Adversarial image ($\epsilon = 32/255$)	62.5	7.8	48.0	5.4	16.5	5.8	60.0
+DiffPure-VLM ($t^* = 50$)	**43.9**	3.2	34.6	2.4	12.8	3.7	41.0
+DiffPure-VLM ($t^* = 100$)	**44.1**	3.5	35.4	2.1	13.0	4.1	41.3
+DiffPure-VLM ($t^* = 150$)	**42.5**	3.5	32.7	2.8	12.1	4.1	39.3
Adversarial image ($\epsilon = 64/255$)	57.5	9.2	43.5	5.2	15.3	5.8	54.7
+DiffPure-VLM ($t^* = 50$)	**42.1**	2.7	32.1	2.1	12.3	2.9	39.0
+DiffPure-VLM ($t^* = 100$)	**40.5**	3.3	31.4	1.9	11.7	2.8	37.5
+DiffPure-VLM ($t^* = 150$)	**42.4**	3.5	32.8	1.8	11.5	4.0	40.2
MiniGPT-4-13B-RobustVLGuard							
Benign Clean image	16.0	0.4	9.9	0.3	4.6	1.1	12.1
Benign Noisy image	16.5	0.9	11.9	0.6	5.8	1.0	14.0
Adversarial image ($\epsilon = 16/255$)	47.4	9.3	34.2	1.4	11.8	4.2	41.5
+DiffPure-VLM ($t^* = 50$)	**16.0**	0.6	9.3	0.3	6.5	1.4	13.2
+DiffPure-VLM ($t^* = 100$)	**15.8**	0.7	9.7	0.0	6.1	1.1	12.8
+DiffPure-VLM ($t^* = 150$)	**9.8**	0.4	6.0	0.1	3.3	0.5	7.8
Adversarial image ($\epsilon = 32/255$)	53.7	9.8	35.3	4.1	13.9	5.4	48.1
+DiffPure-VLM ($t^* = 50$)	**13.6**	0.3	9.2	0.2	5.5	0.9	10.6
+DiffPure-VLM ($t^* = 100$)	**15.2**	0.6	9.5	03	5.4	1.1	12.7
+DiffPure-VLM ($t^* = 150$)	**11.9**	0.5	8.6	0.2	4.2	0.6	9.9
Adversarial image ($\epsilon = 64/255$)	60.2	6.8	44.6	4.2	16.2	5.8	56.0
+DiffPure-VLM ($t^* = 50$)	**30.3**	1.8	21.6	1.4	11.4	1.9	26.9
+DiffPure-VLM ($t^* = 100$)	**10.6**	0.0	7.1	0.0	4.1	0.8	8.2
+DiffPure-VLM ($t^* = 150$)	**9.4**	0.4	5.5	0.3	4.1	0.6	7.0

Conjecture 2: Gaussian Noise as a Simple Attack on VLM Safety

Statement: Adding Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$ to a clean image I_{clean} can compromise the safety of VLMs.
Setting: Let I_{clean} be a clean image, and $\eta \sim \mathcal{N}(0, \sigma^2 I)$ be Gaussian noise. The perturbed image is defined as:

$$I_{\text{noisy}} = I_{\text{clean}} + \eta.$$

The VLM processes the noisy image I_{noisy} along with a corresponding text prompt T, and generates an output based on this combined input.
Discussion:

1. **Effect of Noise on Model Input:** The input to the model becomes $I_{\text{noisy}} = I_{\text{clean}} + \eta$. This perturbation, although random, alters the image representation processed by the VLM. The model's output can be locally approximated around the clean input as:

$$f_\theta(I_{\text{clean}} + \eta, T) \approx f_\theta(I_{\text{clean}}, T) + \nabla I_{\text{clean}} f_\theta \cdot \eta,$$

where $\nabla I_{\text{clean}} f_\theta$ represents the gradient of the model output with respect to the clean image input. The Gaussian noise η introduces random perturbations that shift the image features.

2. **Vulnerability of VLMs to Noise:** VLMs are typically trained on clean image data, and thus, they may lack robustness to input noise. The introduction of Gaussian noise can push the model's input into regions of the feature space that were not well-covered during training, potentially causing the model to misinterpret the input and generate unexpected responses.

3. **Impact on Safety:** Adding Gaussian noise may shift the model's behavior towards decision boundaries where safety mechanisms are less effective. This increases the likelihood of generating unsafe or harmful text:

$$L(f_\theta(I_{\text{clean}} + \eta, T), T^{\text{target}}) \geq L(f_\theta(I_{\text{clean}}, T), T^{\text{target}}),$$

where T^{target} represents a potentially harmful target output. The inequality suggests that the noisy input can lead to a higher loss, increasing the risk of unsafe text generation.

4. **Gaussian Noise as a Simple Yet Effective Attack:** Unlike adversarial perturbations that require careful optimization and model-specific crafting, Gaussian noise introduces random changes without any specific targeting. Despite its simplicity, it can destabilize the model and affect its safety, demonstrating that even non-adversarial noise can be a risk factor for VLMs.

In summary, adding Gaussian noise to clean images can indeed disrupt the safety of VLMs, even in the absence of sophisticated adversarial attacks. This highlights a potential vulnerability of VLMs that warrants further investigation.

Conjecture 3: Gaussian Noise as a Regularizer

Statement: Augmenting training data with Gaussian noise acts as a regularizer, reducing the risk of overfitting to adversarial perturbations and enhancing model robustness.
Discussion:

We introduce a regularized loss function that incorporates Gaussian noise during training:

$$L_{\text{reg}}(\theta) = \mathbb{E}_{(I,T)\sim\mathcal{D}} \mathbb{E}_{\eta\sim\mathcal{N}(0,\sigma^2 I)} \left[L\left(f_\theta(I + \eta, T), T \right) \right],$$

where $\mathcal{D}$ represents the training data distribution. This formulation encourages the model to perform well not only on clean inputs but also on noisy inputs, promoting robustness.

To understand the regularizing effect of Gaussian noise, we expand the loss function L using a second-order Taylor expansion around the clean input I:

$$\begin{aligned} L\left(f_\theta(I + \eta, T), T \right) \approx\ & L\left(f_\theta(I, T), T \right) \\ & + \nabla_I L\left(f_\theta(I, T), T \right)^\top \eta \\ & + \frac{1}{2} \eta^\top \nabla_I^2 L\left(f_\theta(I, T), T \right) \eta. \end{aligned}$$

Taking the expectation over the Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$, we obtain:

$$\begin{aligned} \mathbb{E}_\eta \left[L\left(f_\theta(I + \eta, T), T \right) \right] \approx\ & L\left(f_\theta(I, T), T \right) \\ & + \frac{1}{2} \mathbb{E}_\eta \left[\eta^\top \nabla_I^2 L\left(f_\theta(I, T), T \right) \eta \right] \\ =\ & L\left(f_\theta(I, T), T \right) \\ & + \frac{\sigma^2}{2} \operatorname{Tr}\left(\nabla_I^2 L\left(f_\theta(I, T), T \right) \right). \end{aligned}$$

The additional term $\frac{\sigma^2}{2} \operatorname{Tr}\left(\nabla_I^2 L\left(f_\theta(I, T), T \right) \right)$ penalizes large curvature (i.e., high second derivatives) of the loss function with respect to the input I. This encourages smoother mappings from the input to the output, reducing the model's sensitivity to small input perturbations, including adversarial ones.

In summary, the addition of Gaussian noise during training acts as a regularizer by penalizing sharp changes in the loss landscape. This results in a model that is less prone to overfitting and more resilient to adversarial attacks, as it learns smoother and more stable input-output mappings.

Conjecture 4: Fine-Tuning with Gaussian Noise Preserves Performance

Statement: Fine-tuning the VLM with Gaussian noise-augmented data maintains performance on clean data while enhancing robustness to adversarial perturbations.

Discussion:

Let $\mathcal{D} = \{(I_i, T_i)\}_{i=1}^N$ be the original training dataset. We construct an augmented dataset by adding Gaussian noise:

$$\mathcal{D}_{\text{aug}} = \left\{(I_i + \eta_i, T_i) \mid \eta_i \sim \mathcal{N}(0, \sigma^2 I)\right\}_{i=1}^N.$$

The training objective is to minimize the following loss function:

$$\hat{L}_{\text{aug}}(\theta) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}_{\eta_i}\left[L\left(f_\theta(I_i + \eta_i, T_i), T_i\right)\right].$$

Since the Gaussian noise η_i has a zero mean, the expected gradient of the loss with respect to the model parameters θ is centered around the gradient on the clean data:

$$\mathbb{E}_{\eta_i}\left[\nabla_\theta L\left(f_\theta(I_i + \eta_i, T_i), T_i\right)\right] = \nabla_\theta L\left(f_\theta(I_i, T_i), T_i\right).$$

This result indicates that the expected training gradient remains aligned with the gradient computed on the clean data, thereby preserving the model's performance on clean inputs.

Moreover, by training on both clean and noise-augmented data, the model is exposed to a neighborhood of inputs around each training example. This exposure helps the model generalize better and become less sensitive to small perturbations, effectively enhancing its robustness against adversarial attacks.

In summary, fine-tuning with Gaussian noise-augmented data acts as a regularization strategy that not only maintains the VLM's accuracy on clean data but also improves its resistance to adversarial perturbations.

G. Detailed Proofs

Bounding the Increase in Loss Due to Gaussian Noise

Discussion:

Step 1: Lipschitz Continuity of f_θ and L

Assume that the model function $f_\theta : \mathbb{R}^d \times \mathcal{T} \to \mathbb{R}^k$ and the loss function $L : \mathbb{R}^k \times \mathcal{T} \to \mathbb{R}$ are Lipschitz continuous with constants K_f and K_L, respectively. That is, for all $x, y \in \mathbb{R}^d$ and $T \in \mathcal{T}$:

$$\|f_\theta(x, T) - f_\theta(y, T)\| \leq K_f \|x - y\|,$$

and for all $a, b \in \mathbb{R}^k$:

$$|L(a, T^{\text{target}}) - L(b, T^{\text{target}})| \leq K_L \|a - b\|.$$

Step 2: Bounding the Change in Loss Due to Noise η

Consider the adversarially perturbed image $I_\delta = I + \delta$, where δ is crafted to minimize the loss:

$$\delta = \arg\min_{\|\delta\| \leq \epsilon} L\left(f_\theta(I + \delta, T), T^{\text{target}}\right).$$

When Gaussian noise $\eta \sim \mathcal{N}(0, \sigma^2 I)$ is added, the input becomes $I_{\delta,\eta} = I + \delta + \eta$. The change in loss due to η is:

$$\Delta L = L\left(f_\theta(I + \delta + \eta, T), T^{\text{target}}\right) - L\left(f_\theta(I + \delta, T), T^{\text{target}}\right).$$

Using the Lipschitz continuity of L:

$$|\Delta L| \leq K_L \left\|f_\theta(I + \delta + \eta, T) - f_\theta(I + \delta, T)\right\|.$$

Step 3: Computing the Expected Increase in Loss

Applying the Lipschitz continuity of f_θ:

$$\|f_\theta(I + \delta + \eta, T) - f_\theta(I + \delta, T)\| \leq K_f \|\eta\|.$$

Thus, the change in loss is bounded by:

$$|\Delta L| \leq K_L K_f \|\eta\|.$$

Since η is a Gaussian random vector with zero mean and covariance $\sigma^2 I$, the expected value of $\|\eta\|$ is:

$$\mathbb{E}[\|\eta\|] = \sigma\sqrt{2}\frac{\Gamma\left(\frac{d+1}{2}\right)}{\Gamma\left(\frac{d}{2}\right)} \approx \sigma\sqrt{d - \frac{1}{2}} \quad \text{for large } d.$$

Therefore, the expected increase in loss is approximately:

$$\mathbb{E}[|\Delta L|] \leq K_L K_f \sigma\sqrt{d}.$$

Step 4: Lower Bounding the Expected Increase in Loss

Since δ minimizes $L(f_\theta(I + \delta, T), T^{\text{target}})$ at the point $I + \delta$, any perturbation η added to $I + \delta$ is likely to increase the loss. Under the conjecture that L is convex around $I + \delta$, the expected increase in loss due to η can be lower bounded using the curvature (second derivative) of L:

$$\mathbb{E}_\eta\left[L\left(f_\theta(I + \delta + \eta, T), T^{\text{target}}\right)\right] \geq L\left(f_\theta(I + \delta, T), T^{\text{target}}\right) + \frac{\sigma^2}{2}\lambda_{\min},$$

where $\lambda_{\min}$ is the smallest eigenvalue of the Hessian matrix $\nabla^2_{I+\delta} L(f_\theta(I + \delta, T), T^{\text{target}})$.

Conclusion:

Adding Gaussian noise increases the expected loss by at least $\frac{\sigma^2}{2}\lambda_{\min}$, reducing the effectiveness of the adversarial perturbation. This result supports the conjecture that Gaussian noise disrupts the optimization achieved by the adversary, weakening the impact of adversarial attacks.

Second-Order Taylor Expansion of L Around I

Discussion:

Step 1: Second-Order Taylor Expansion

We expand the loss function $L(f_\theta(I+\eta,T),T)$ around the point I using the second-order Taylor expansion:

$$\begin{aligned} L(f_\theta(I+\eta,T),T) = L(f_\theta(I,T),T) \\ + \nabla_I L(f_\theta(I,T),T)^\top \eta \\ + \frac{1}{2}\eta^\top \nabla_I^2 L(f_\theta(I,T),T)\eta + R_3 \end{aligned}$$

where:

- $\nabla_I L(f_\theta(I,T),T)$ is the gradient of the loss with respect to the input I.
- $\nabla_I^2 L(f_\theta(I,T),T)$ is the Hessian matrix of second derivatives with respect to I.
- R_3 is the remainder term of higher order $O(\|\eta\|^3)$.

Step 2: Expected Value of the Linear Term

Since η is sampled from a zero-mean Gaussian distribution $\eta \sim \mathcal{N}(0,\sigma^2 I)$, the expected value of the linear term becomes:

$$\mathbb{E}_\eta\left[\nabla_I L(f_\theta(I,T),T)^\top \eta\right] = \nabla_I L(f_\theta(I,T),T)^\top \mathbb{E}_\eta[\eta] = 0$$

Step 3: Expected Value of the Quadratic Term

Next, we compute the expectation of the quadratic term:

$$\mathbb{E}_\eta\left[\eta^\top \nabla_I^2 L(f_\theta(I,T),T)\eta\right]$$

Using the properties of Gaussian distributions, we know that for a symmetric matrix A:

$$\mathbb{E}_\eta\left[\eta^\top A\eta\right] = \sigma^2 \operatorname{Tr}(A)$$

Thus, the expected value of the quadratic term becomes:

$$\mathbb{E}_\eta\left[\eta^\top \nabla_I^2 L(f_\theta(I,T),T)\eta\right] = \sigma^2 \operatorname{Tr}\left(\nabla_I^2 L(f_\theta(I,T),T)\right)$$

Step 4: Neglecting the Remainder Term

For small values of σ, the remainder term R_3 is of order $O(\sigma^3)$ and can be safely ignored. Thus, the approximation becomes:

$$\begin{aligned} \mathbb{E}_\eta\left[L(f_\theta(I+\eta,T),T)\right] \approx L(f_\theta(I,T),T) \\ + \frac{\sigma^2}{2}\operatorname{Tr}\left(\nabla_I^2 L(f_\theta(I,T),T)\right) \end{aligned}$$

Step 5: Interpretation of the Trace Term

The term $\operatorname{Tr}\left(\nabla_I^2 L(f_\theta(I,T),T)\right)$ denotes the sum of the eigenvalues of the Hessian matrix, representing the overall curvature of the loss function with respect to the input. A larger trace value indicates higher curvature, suggesting greater sensitivity of the model to input perturbations. Reducing this sensitivity is crucial for enhancing the model's robustness.

Step 6: Gaussian Noise as Regularization

The additional term $\frac{\sigma^2}{2}\operatorname{Tr}\left(\nabla_I^2 L(f_\theta(I,T),T)\right)$ functions as a regularizer, penalizing high curvature in the loss landscape. This encourages the model to learn smoother input-output mappings, thereby reducing its vulnerability to small perturbations, including adversarial attacks.

Step 7: Connection to Tikhonov Regularization

This regularization effect is conceptually similar to Tikhonov regularization, where a penalty proportional to the norm of the model parameters is added to the loss function. In our case, the penalty arises naturally from the Gaussian noise, encouraging robustness by flattening the loss landscape:

$$\begin{aligned} \mathbb{E}_\eta\left[L(f_\theta(I+\eta,T),T)\right] \approx L(f_\theta(I,T),T) \\ + \frac{\sigma^2}{2}\operatorname{Tr}\left(\nabla_I^2 L(f_\theta(I,T),T)\right) \end{aligned}$$

This smoothing effect reduces the model's sensitivity to input perturbations, enhancing its robustness without compromising performance on clean data.

References

[1] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1

[2] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024. 1

[3] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*, 2024. 1

[4] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 4

[5] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[6] Jenny N Theresa Yu Ivy Zhang, Wei Peng and David Qiu. Ivy-vl:compact vision-language models achieving sota with optimal data, 2024. 1

[7] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2024. 1

[8] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *International Conference on Machine Learning*, 2022. 2, 4

[9] Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *AAAI*, 2024. 1

[10] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024. 1