Appendix

A1. Additional Experimental Results on ImageNet

To further verify the scalability of NAPPure, we conducted a large-scale experiment on ImageNet dataset. We sample 512 samples for evaluation. For the diffusion model in the adversarial purification method, we adopted the pretrained unconditional diffusion model provided by Karras et al. (2022), and for the classifier, we followed the ResNet-50 framework used by Nie et al. (2022). As shown in Tab. 4, we performed experiments using four types of attacks, with the detailed configurations of these attack types as follows:

- Conv: A convolution-based blur attack using a 15×15 uniform kernel ε_0, with attack parameters constrained by $\|\varepsilon - \varepsilon_0\|_\infty \leq 0.025$.
- Patch: The patch-based occlusion attack. The patch is fixed at the center of the image with a fixed size 50x50.
- Flow: The flow-field based distortion attack. Parameters are limited by $\|\varepsilon\|_\infty \leq 1.2$. To ensure natural-looking of the distortion, we apply Gaussian smoothing with standard deviation 1.5 onto the parameters, before the flow-field transformation. The kernel size is 29x29.
- Add: The traditional adversarial attack with additive perturbations. Parameters are limited by $\|\varepsilon\|_\infty \leq 4/255$.

Result. As shown in Table 4, our NAPPure method outperforms DiffPure method by 8.19%. This indicates that our method is also effective on large-scale datasets.

A2. More details of the experiments

Table 5 summarizes the detailed parameter settings used in our evaluations on the GTSRB and CIFAR-10 datasets.

Specifically, this table outlines the number of iterations, as well as the values of regularization parameters λ_1 (controlling the perturbation prior loss) and λ_2 (governing the image reconstruction loss), for each combination of dataset, attack type (Additive, Blur, Flow, Patch), and defense method (NAPPure and NAPPure-joint). The variations in settings across different scenarios (e.g., fewer iterations for Additive attacks on CIFAR-10 compared to non-additive attacks) reflect the need to adapt to the distinct characteristics of each perturbation type and dataset.

A3. Computational Cost Analysis

Purification efficiency holds significant importance for real-world deployment scenarios. To delve into this, we conducted an analysis of the trade-off between the number of purification iterations and model robustness, using the GTSRB dataset under patch attacks as the test case. The detailed results are presented in Table 1.

The findings reveal that NAPPure reaches a near-optimal performance level within 200 iterations, achieving a robust accuracy of 72.26%. This is merely 1.96% lower than the 74.22% robust accuracy obtained after 500 iterations. However, when the number of iterations is extended to 1000, a noticeable performance degradation occurs, with the robust accuracy dropping to 60.74%. This decline is likely attributed to the over-optimization of perturbation parameters during the extended purification process.

Iterations	Robust Acc
100	63.48%
200	72.26%
500	74.22%
1000	60.74%

Table 1. The robust accuracy under different numbers of purification iterations (GTSRB, patch attack).

Auxiliary Model	Robust Acc	Clean Acc
3-layer CNN	74.22%	93.55%
small-scale ResNet	71.29%	93.16%

Table 2. Impact of auxiliary model architecture on NAPPure performance (GTSRB, patch attack).

Attack Type	Attack Parameter	Robust Acc
Patch Attack	5×5	85.16%
	7×7	74.22%
	9×9	67.97%
Blur Attack	3×3	91.80%
	5×5	86.91%

Table 3. Generalization of NAPPure to varying attack parameters (GTSRB).

A4. The robustness verification of the NAPPure auxiliary model for architectural changes

The auxiliary model in NAPPure (used for non-differentiable perturbations like patch occlusion) is designed as an image-to-image generative network. To validate its robustness to architectural variations, we compared two architectures: a lightweight 3-layer CNN and a deeper small-scale ResNet.

Table 2 shows that replacing the 3-layer CNN with small-scale ResNet results in a minor drop in robust accuracy (74.22% → 71.29%, a 2.93% difference), while clean accuracy remains stable. This insensitivity to architecture arises because the auxiliary model focuses on reconstructing perturbed images rather than discriminative tasks, making it less vulnerable to architectural changes. Importantly, both configurations outperform baselines (e.g., DiffPure's

Defense	Conv		Patch		Flow		Add		Avg	
Method	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob	Acc	Rob
None	75.78	11.33	75.78	7.81	75.78	0	75.78	0	75.78	4.79
DiffPure*	69.92	20.83	69.92	42.97	69.92	7.81	69.92	46.88	69.92	29.62
LM*	67.97	12.11	67.97	6.25	67.97	17.97	67.97	59.38	67.97	23.93
NAPPure	69.11	**21.48**	65.26	**48.05**	68.35	**21.48**	69.33	**60.16**	68.01	**37.79**

Table 4. Clean accuracy (Acc %) and robust accuracy (Rob %) of different methods against adversarial attacks with different types of perturbations on ImgNet dataset. Methods marked with * share identical implementation across attack types.

Dataset	**Attack Type**	**Defense Method**	**Iterations**	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$
GTSRB	Additive	NAPPure	100	0.1	3
	Blur	NAPPure	500	0.001	3
	Flow	NAPPure	500	0.01	1
	Patch	NAPPure	500	0.01	5
	-	NAPPure-joint	500	0.001	3
CIFAR-10	Additive	NAPPure	20	0.1	5
	Blur	NAPPure	500	0.001	5
	Flow	NAPPure	500	0.01	1
	Patch	NAPPure	500	0.01	5
	-	NAPPure-joint	100	0.01	5
ImageNet	Additive	NAPPure	10	0.1	3
	Blur	NAPPure	100	0.01	5
	Flow	NAPPure	100	0.01	1
	Patch	NAPPure	100	0.01	10

Table 5. Detailed parameter settings for NAPPure and NAPPure-joint under different attacks on GTSRB, CIFAR-10 and ImageNet datasets

46.29% robust accuracy for patch attacks), confirming the reliability of NAPPure's design.

A5. The generalization ability of NAPPure under different attack parameters

A key advantage of NAPPure is its ability to maintain robustness under varying attack parameters, even when the attack parameters differ from those used in defense configuration. We evaluate this generalization capability for two representative non-additive attack types: patch occlusion and convolution-based blur.

For patch attacks, we test NAPPure with a fixed defense model (configured for general patch occlusion) against varying attack patch sizes. NAPPure achieves robust accuracies of 85.16%, 74.22%, and 67.97% for attack patch sizes of 5×5, 7×7, and 9×9, respectively. All results outperform baseline methods (e.g., DiffPure and LM) under the same settings. This is because NAPPure features an adaptive learning mechanism for patch sizes, endowing it with the ability to adapt to different attack scenarios. Such adaptability ensures its effectiveness even when attack patch sizes vary.

For convolution-based blur attacks, we use a defense model with a fixed 5×5 kernel and evaluate against attacks with different kernel sizes. As shown in Table 3, NAPPure achieves 91.80% robust accuracy against 3×3 attack kernels and 86.91% against 5×5 attack kernels. These results confirm that NAPPure remains effective as long as the attack kernel size does not exceed the defense kernel size, validating its generalization to varying convolution parameters.