SteeringTTA: Guiding Diffusion Trajectories for Robust Test-Time-Adaptation

Jihyun Yu [1] **Yoojin Oh** [1] **Wonho Bae** [2] **Mingyu Kim** [2 *] **Junhyug Noh** [1 *]

Abstract

Test-time adaptation (TTA) aims to correct performance degradation of deep models under distribution shifts by updating models or inputs using unlabeled test data. Input-only diffusion-based TTA methods improve robustness for classification to corruptions but rely on gradient guidance, limiting exploration and generalization across distortion types. We propose *SteeringTTA*, an inference-only framework that adapts Feynman-Kac steering to guide diffusion-based input adaptation for classification with rewards driven by pseudo-label. SteeringTTA maintains multiple particle trajectories, steered by a combination of cumulative top-K probabilities and an entropy schedule, to balance exploration and confidence. On ImageNet-C, SteeringTTA consistently outperforms the baseline without any model updates or source data.

1. Introduction

Deep networks often suffer drastic performance drops under distribution shifts at test time. Test-time adaptation (TTA) methods – either updating model weights or refining inputs on-the-fly – can help, but existing diffusion-based input adaptation methods still fail when corruptions lie in the low-frequency band. For instance, Diffusion-Driven Adaptation (DDA) (Gao et al., 2023) preserves low-frequency structure via ILVR (Iterative Latent Variable Refinement) (Choi et al., 2021). While effective against high-frequency noise, it can amplify low-frequency corruptions (*e.g.*, frost), leading to semantic distortion and inaccurate classification.

Figure 1 shows a frost-corrupted image of a *European fire salamander*. A ResNet-50 classifier (trained on clean images) classifies it as *black & gold garden spider*, with *European fire salamander* in its Top-5 predictions – an understandable confusion. After DDA's low-frequency denoising, however, the amplified frost pattern leads to a wildly incorrect Top-1 prediction *spider web*, and *European fire salamander* falls out of the Top-25. This example highlights that uniformly restoring all low-frequency components can eliminate semantic cues.

To address this, we ask: how can we restore images without amplifying signals introduced by corruption, while still sharpening the classifier's plausible confusion set? We answer by *steering* diffusion sampling via Sequential Monte Carlo (SMC) and Feynman-Kac potentials (Singhal et al., 2025), guided by a pseudo-label reward that (1) initially preserves the original confusion group and (2) gradually focuses on the correct class.

In this work, we propose **SteeringTTA**, the first diffusion-based TTA framework to harness SMC steering. SteeringTTA runs entirely at inference, maintains multiple particle trajectories, and uses a dynamic reward to guide resampling and denoising. As Figure 1 demonstrates, our method recovers an image that the classifier correctly labels as *European fire salamander*, even under severe low-frequency corruption.

Our contributions are summarized as follows:

- We pinpoint the failure of DDA-style adaptation for low-frequency corruption, and motivate targeted steering.
- We introduce *SteeringTTA* that leverages Feynman-Kac steering in diffusion restoration to selectively suppress corruption frequencies while preserving object semantics.
- We design a scheduling for pseudo-label reward that first maintains the classifier's confusion set and then anneals entropy to converge to the true label.
- We demonstrate on ImageNet-C that SteeringTTA outperforms DDA, reliably recovering correct classes under challenging low-frequency corruptions.

2. Related Work

Test-Time Adaptation. Methods either update model parameters (*e.g.*, entropy minimization (Wang et al., 2020)) or refine inputs via self-supervision. Parameter updates risk hyperparameter sensitivity and collapse, while input-only approaches may not directly optimize accuracy.

[*]Corresponding authors [1]Department of Artificial Intelligence, Ewha Womans University, Seoul, Republic of Korea [2]Department of Computer Science, University of British Columbia, Vancouver, Canada. Correspondence to: Mingyu Kim <mgyu.kim@ubc.ca>, Junhyug Noh <junhyug@ewha.ac.kr>.

Presented at Workshop on Test-Time Adaptation: Putting Updates to the Test in 42nd International Conference on Machine Learning, Vancouver, Canada. 2025. Copyright 2025 by the author(s).

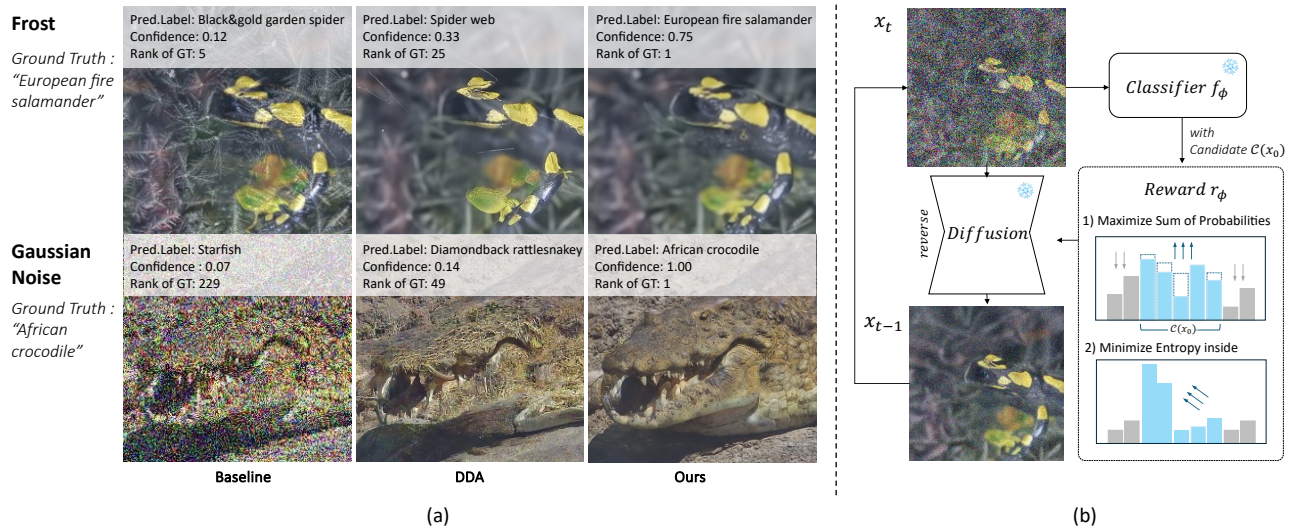

Figure 1: **(a) Our motivation:** from left to right, the original corrupted image, the DDA-denoised result, and our method's reconstruction, with corresponding ResNet-50 predictions. **(b) Pipeline overview:** at each diffusion step the image is updated, the classifier output is used to compute a reward, and the update is steered to maximize this reward.

Diffusion-Based Input Adaptation. Diffusion purifiers restore corrupted inputs: DiffPure (Nie et al., 2022) for adversarial examples, DDA (Gao et al., 2023) via low-frequency denoising, and GDA (Tsai et al., 2024) with gradient-based style/semantic constraints. These enhance robustness but rely on differentiable guidance.

SMC and FK Steering. Sequential Monte Carlo steers diffusion by evolving multiple particles and resampling with arbitrary potentials (Del Moral et al., 2006). Feynman–Kac steering extends this to non-differentiable rewards without retraining (Singhal et al., 2025).

3. Our Approach

We introduce *SteeringTTA*, a test-time adaptation method that steers a pretrained diffusion model toward classification objectives using FK potentials. After first reviewing diffusion models and SMC, we describe how FK steering is adapted for input restoration and detail our design choices – resampling, proposal kernel, weighting – and our pseudo-label reward.

3.1. Preliminaries

Diffusion Models. A diffusion model defines a forward noising process with variance β_t:

$$q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \mathcal{N}\big(\sqrt{1-\beta_t}\,\mathbf{x}_{t-1},\, \beta_t\,\mathbf{I}\big), \quad t = 1, \ldots, T,$$

and trains a denoiser ϵ_θ to approximate the reverse process:

$$p_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t).$$

Starting from $\mathbf{x}_T \sim \mathcal{N}(0, \mathbf{I})$, repeated denoising yields a sample $\mathbf{x}_0$ close to the training distribution (Ho et al., 2020).

Sequential Monte Carlo (SMC). SMC approximates a sequence of target distributions $\{\pi_t\}_{t=0}^{T}$ by maintaining K weighted particles $(\mathbf{x}_t^{(k)}, w_t^{(k)})$. At each reverse step:

1. **Resampling:** normalize $\{w_t^{(k)}\}$, resample the particles accordingly, and begin the next proposal step with uniform weights.

2. **Proposal:** sample backward each particle from defined proposal distribution m_t

$$\mathbf{x}_{t-1}^{(k)} \sim m_t(\mathbf{x}_{t-1} \mid \mathbf{x}_t^{(k)}).$$

3. **Weighting:** compute importance weights to correct the discrepancy between m_t and π_{t-1}

$$w_{t-1}^{(k)} = w_t^{(k)} \frac{\pi_{t-1}(\mathbf{x}_{t-1}^{(k)})\, m_t(\mathbf{x}_t^{(k)} \mid \mathbf{x}_{t-1}^{(k)})}{\pi_t(\mathbf{x}_t^{(k)})\, m_t(\mathbf{x}_{t-1}^{(k)} \mid \mathbf{x}_t^{(k)})}.$$

As $K \to \infty$, SMC recovers exact samples from π_0.

Feynman-Kac (FK) Steering. To bias diffusion toward high-reward outputs, define a tilted target distribution:

$$p_{\text{target}}(\mathbf{x}_0) \;\propto\; p_\theta(\mathbf{x}_0) \cdot \exp\big(\lambda \cdot r(\mathbf{x}_0)\big), \tag{1}$$

with reward $r(\mathbf{x}_0)$ and scale $\lambda > 0$. FK steering introduces potentials G_t which tilt the distribution as

$$p_{\text{FK}}(\mathbf{x}_{0:T}) \;\propto\; p_\theta(\mathbf{x}_{0:T}) \prod_{t=0}^{T-1} G_t\big(\mathbf{x}_{t:T}\big), \tag{2}$$

ensuring the marginal of $\mathbf{x}_0$ follows p_{target} by setting $\prod_{t=0}^{T-1} G_t(\mathbf{x}_{t:T}) = \exp\big(\lambda \cdot r(\mathbf{x}_0)\big)$. In practice, we estimate $\mathbf{x}_0$ with intermediate states of reverse diffusion processes via Tweedie's formula (Chung et al., 2022; Efron, 2011):

$$\hat{\mathbf{x}}_t = \frac{\mathbf{x}_t - \sqrt{1-\alpha_t}\,\epsilon_\theta(\mathbf{x}_t)}{\sqrt{\alpha_t}}, \tag{3}$$

where $\alpha_t = \prod_{s=1}^{t}(1-\beta_s)$, and evaluate $r(\hat{\mathbf{x}}_t)$ to compute G_t. No backpropagation through r is required, allowing non-differentiable and classification-aware rewards to steer sampling processes.

3.2. Generalized FK steering for TTA

Given a corrupted input $\mathbf{x}_0$, we first perform N forward noising steps to reach $\mathbf{x}_N$. We then run N reverse steps, tracking K parallel particles. At each timestep t:

1. **Resampling.** Following Singhal et al. (2025), we monitor particle degeneracy via the *effective sample size* $\text{ESS}_t = \left[\sum_{i=1}^{K}(\hat{G}_t^i)^2\right]^{-1}$, where $\hat{G}_t^i$ are the normalized FK potentials. If $\text{ESS}_t < K/2$, we trigger multinomial resampling to preserve diversity without extra computation.

2. **Proposal.** FK-Steering is compatible with any proposal kernel $\tau(\mathbf{x}_{t-1}|\mathbf{x}_t)$, *e.g.*, the unbiased reverse kernel or gradient-guided variants. Unlike typical generative tasks where we generate i.i.d samples of the data distribution, we need to preserve the semantics of the given image for the following classification. To this end, we employ a low-pass filter that maintains the overall semantics of an image (Gao et al., 2023; Raman et al., 2023).

3. **Weighting.** Among potentials that satisfy Eq. (2), we employ the *difference* potential following (Singhal et al., 2025; Wu et al., 2023),

$$G_t(\mathbf{x}_{t-1}, \mathbf{x}_t) = \exp(\lambda \cdot (r_\phi(\hat{\mathbf{x}}_{t-1}) - r_\phi(\hat{\mathbf{x}}_t))) \tag{4}$$

which directly rewards particles whose predicted clean image $\hat{\mathbf{x}}_{t-1}$ improves the classifier's objective.

We maintain K particles, computing rewards on intermediate $\hat{\mathbf{x}}_t^{(k)}$, and iteratively steering and pruning them. At $t = 0$, the particle with the highest reward is chosen as the adapted image; please refer to Algorithm 1 for details.

3.3. Test-time Reward

Candidate set construction. As ground-truth labels are unavailable at test time, we define an adaptive candidate set,

$$\mathcal{C}(\mathbf{x}_0) = \big\{\, y : \sum_{y' \in \text{desc}(y)} p(y' \mid \mathbf{x}_0) \geq P\% \big\}, \tag{5}$$

accumulating classes in descending order of predicted probability on the corrupted input until their total mass exceeds $P\%$. This preserves the classifier's initial confusion group.

Algorithm 1 SteeringTTA

Input: Corrupted image $\mathbf{x}_0$, Diffusion model $p_\theta(\mathbf{x}_{0:T})$, Classifier f_ϕ, Potentials G_t, Proposal $\tau(\mathbf{x}_t \mid \mathbf{x}_{t+1})$, Reward function $r_\phi(\cdot)$, Number of particles K
N: diffusion range
ϕ_D: low-pass filter of scale D
$\hat{\mathbf{x}}_t$: predicted clean image at timestep t
$\mathcal{C}$: set built with $f_\phi(\mathbf{x}_0)$.
Sample $\mathbf{x}_N^i \sim q(\mathbf{x}_N \mid \mathbf{x}_0)$ ▷ forward pass
Define $r_\phi(\cdot) \leftarrow r_\phi(\cdot, t, \mathcal{C})$ ▷ Eq. 6
Initial weights, $G_N^i = \exp(\lambda \cdot r_\phi(\hat{\mathbf{x}}_N^i))$ for $i \in [K]$
for $t = N-1$ **to** 0 **do**
  Resample:
  Sample indices $a_t^i \sim \text{Multinomial}(\mathbf{x}_t^i, G_t^i)$
  and set $\mathbf{x}_t^i = \mathbf{x}_t^{a_t^i}$ for $i \in [K]$
  Propose: ▷ low-pass filtering
  $\mathbf{x}_{t-1}^i, \hat{\mathbf{x}}_{t-1}^i \sim \tau_\theta(\mathbf{x}_{t-1} \mid \mathbf{x}_t)$
  $\mathbf{x}_{t-1}^i \leftarrow \mathbf{x}_{t-1}^i - w\nabla_{\mathbf{x}_t} \left\|\phi_D(\mathbf{x}_0) - \phi_D(\hat{\mathbf{x}}_{t-1}^i)\right\|_2$
  Weight:
  G_{t-1}^i for $i \in [K]$ using:

$$G_{t-1}^i = \frac{p_\theta(\mathbf{x}_{t-1}^i \mid \mathbf{x}_t^i)}{\tau(\mathbf{x}_{t-1}^i \mid \mathbf{x}_t^i)} G_{t-1}(\mathbf{x}_N^i, \ldots, \mathbf{x}_{t-1}^i)$$

end for
Output: $\mathbf{x}_0^g \leftarrow \arg\max_{i \in [K]} r_\phi(\hat{\mathbf{x}}_0^i)$

Reward formulation. Our steering reward at step t is

$$r_\phi(\hat{\mathbf{x}}_t, t, \mathcal{C}) = (1-\alpha(t)) \log \sum_{y \in \mathcal{C}(\mathbf{x}_0)} p(y \mid \hat{\mathbf{x}}_t) - \alpha(t) H\big(\mathcal{C}(\mathbf{x}_0)\big), \tag{6}$$

with an entropy on the classes in the adaptive candidate set,

$$H\big(\mathcal{C}(\mathbf{x}_0)\big) = -\sum_{y \in \mathcal{C}(\mathbf{x}_0)} p(y \mid \hat{\mathbf{x}}_t) \log p(y \mid \hat{\mathbf{x}}_t). \tag{7}$$

The *log-sum* term encourages boosting total probability over the plausible labels by steering it to be close to 1, preventing jumps to unrelated classes; the *entropy* term discourages the predicted probability to be an uniform distribution, pushing the sampler to commit to a single label.

Annealing schedule. We linearly anneal $\alpha(t)$ from 0 at $t = N$ (favoring exploration via log-sum) to 1 at $t = 0$ (favoring exploitation via entropy minimization). Early iterations thus maintain the original confusion set, while later ones refine focus on the correct class.

4. Experiments

4.1. Experimental Settings

Dataset. We evaluate on ImageNet-C (Hendrycks & Dietterich, 2019), which applies fifteen corruption types at five

Table 1: Average top-1 accuracy (%) on severity 5 ImageNet-C. "Baseline" denotes corrupted images evaluated without any adaptation; other methods are described in the main text. See Appendix A for corruption abbreviation definitions.

Method	Blur				Digital				Noise			Weather				Avg.
	Def.	Glass	Mot.	Zoom	Contr.	Elast.	JPEG	Pixel	Gauss.	Impl.	Shot	Bright	Fog	Frost	Snow	
Baseline (w.o. Adaptation)	12.00	5.80	12.80	21.20	**3.00**	14.60	43.20	30.40	5.20	6.00	8.20	54.80	**21.40**	20.80	16.80	18.41
Diffpure (Nie et al., 2022)	2.00	6.20	5.60	9.80	0.40	18.60	43.20	24.60	5.00	4.40	6.00	42.60	1.60	12.80	10.00	12.85
DDA (Gao et al., 2023)	**12.40**	10.20	13.00	23.40	2.80	**34.20**	50.20	47.80	50.40	49.60	51.40	**55.00**	18.60	27.60	19.40	31.07
Grad-DDA	**12.40**	**11.00**	**13.20**	**24.40**	2.40	32.20	**50.80**	**51.40**	49.00	49.20	50.20	53.60	16.40	28.20	20.60	31.00
SteeringTTA (Ours)	12.20	**11.00**	12.80	23.80	2.40	33.80	49.00	49.80	**52.60**	**50.60**	**53.40**	52.80	18.40	**29.40**	**21.60**	**31.57**
SteeringTTA (GT)	12.40	10.60	13.40	26.00	2.20	34.60	50.20	51.80	57.40	57.40	58.60	53.80	18.20	30.80	24.40	33.45

severity levels to the ImageNet validation set. Following the small-subset protocol, we randomly sample one image per class for 100 classes, forming five disjoint splits (1,500 images per split at severity 5). We report mean of Top-1 accuracy across these splits.

Models. For input restoration, we use a pretrained unconditional diffusion model (256×256) trained on ImageNet-1K (Dhariwal & Nichol, 2021). The classifier is fixed with the ResNet-50 (He et al., 2016) trained on ImageNet. During TTA, the classifier provides pseudo-labels for steering and serves as the evaluator – its weights remain frozen, preventing information leakage from the target data.

Baselines. We compare SteeringTTA against:

- **Diffpure** (Nie et al., 2022): Diffusion based adversarial purification that simply adds a small amount of noise and then reverses until $t = 0$.
- **DDA** (Gao et al., 2023): Diffusion-Driven Adaptation with default hyperparameters.
- **Gradient-guided DDA (Grad-DDA)**: an ablation that steers DDA's reverse process via our pseudo-label reward using gradient ascent.
- **SteeringTTA (GT)**: SteeringTTA using the ground-truth log-likelihood $\log p(y|x)$ as the reward; upper-bound of improvement from SteeringTTA.

Implementation Details. We use $K = 4$ particles and $N = 50$ reverse steps, and follow the resampling schedule of Singhal et al. (2025): resample whenever $\text{ESS} < K/2$ at every 5 steps. Our test-time reward combines the cumulative Top-K probability using adaptive threshold $P = 70\%$, with an entropy term, and employs a linear annealing schedule $\alpha(t) : 0 \to 1$. We set the reward scale $\lambda = 1$. We use an ensemble for the final predictions (Gao et al., 2023), taking $\text{argmax}_y \ \frac{1}{2}\big(p(y|\mathbf{x}_0) + p(y|\mathbf{x}_0^g)\big)$, the most probable class predicted from the original $\mathbf{x}_0$ and adapted image $\mathbf{x}_0^g$. All hyperparameters were fixed a priori (no per-image tuning); each experiment was run once on a NVIDIA A100 GPU.

4.2. Experimental Results

Overall Performance. Table 1 presents average Top-1 accuracy on severity 5 ImageNet-C. SteeringTTA outperforms DDA by 0.51%, while Grad-DDA shows no significant gain.

Table 2: Performance gain *vs*. DDA per corruption category.

Category	DDA	Grad-DDA	Ours	GT
Blur	14.75	+0.50	+0.20	+0.85
Digital	33.75	+0.45	+0.00	+0.95
Noise	50.47	−1.00	+1.73	+7.33
Weather	30.15	−0.45	+0.40	+1.65
Average	31.07	−0.07	+0.51	+2.39

We hypothesize that this is due to Grad-DDA's higher sensitivity to noise in the guidance. Although both methods use the same guidance, Grad-DDA applies it by injecting gradients directly into the pixel space along a single trajectory, which can amplify label noise into high-frequency artifacts D.2. In contrast, SteeringTTA utilizes guidance through resampling and weighting, avoiding direct pixel-level updates. GT-guided SteeringTTA further improves by 2.39%, indicating the gap to the ideal upper-bound. These results confirm that FK steering with pseudo-label rewards is more effective than gradient guidance alone for robust TTA.

Category-wise gains. Table 2 reports relative improvements over DDA by corruption type. Gradient-guided adaptation shows mixed results; it improves blur and digital but degrades noise and weather. GT-guided steering delivers large gains (up to $+7.3\%$), particularly for noise. Our SteeringTTA yields consistent, moderate improvements except for digital; 0.51% gain on average.

5. Conclusion

We present *SteeringTTA*, a novel test-time adaptation framework that steers a pretrained diffusion model using Feynman-Kac potentials. By integrating a pseudo-label-driven reward into the reverse diffusion process, SteeringTTA directly optimizes classification accuracy – unlike prior methods that rely on surrogate objectives or gradient-only guidance. Our multi-particle sampling explores diverse hypotheses and selects high-reward paths to prevent collapse. On ImageNet-C, SteeringTTA outperforms DDA by 0.51% top-1 accuracy, validating the effectiveness of reward-based steering. Future work includes adaptive resampling, richer reward designs, and applications to other domains and real-world shifts.

References

Cardoso, G., Idrissi, Y. J. E., Corff, S. L., and Moulines, E. Monte carlo guided diffusion for bayesian linear inverse problems. *arXiv preprint arXiv:2308.07983*, 2023.

Choi, J., Kim, S., Jeong, Y., Gwon, Y., and Yoon, S. Ilvr: Conditioning method for denoising diffusion probabilistic models. *arXiv preprint arXiv:2108.02938*, 2021.

Chung, H., Kim, J., Mccann, M. T., Klasky, M. L., and Ye, J. C. Diffusion posterior sampling for general noisy inverse problems. *arXiv preprint arXiv:2209.14687*, 2022.

Del Moral, P., Doucet, A., and Jasra, A. Sequential monte carlo samplers. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 68(3):411–436, 2006.

Dhariwal, P. and Nichol, A. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

Efron, B. Tweedie's formula and selection bias. *Journal of the American Statistical Association*, 106(496):1602–1614, 2011.

Gao, J., Zhang, J., Liu, X., Darrell, T., Shelhamer, E., and Wang, D. Back to the source: Diffusion-driven adaptation to test-time corruption. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11786–11796, 2023.

Guo, J., Zhao, J., Ge, C., Du, C., Ni, Z., Song, S., Shi, H., and Huang, G. Everything to the synthetic: Diffusion-driven test-time adaptation via synthetic-domain alignment. *arXiv*, 2024.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *arXiv preprint arxiv:2006.11239*, 2020.

Kim, S., Kim, M., and Park, D. Test-time alignment of diffusion models without reward over-optimization. In *The Thirteenth International Conference on Learning Representations*, 2025.

Li, X., Zhao, Y., Wang, C., Scalia, G., Eraslan, G., Nair, S., Biancalani, T., Regev, A., Levine, S., and Uehara, M. Derivative-free guidance in continuous and discrete diffusion models with soft value-based decoding. *arXiv preprint arXiv:2408.08252*, 2024.

Nie, W., Guo, B., Huang, Y., Xiao, C., Vahdat, A., and Anandkumar, A. Diffusion models for adversarial purification. *arXiv preprint arXiv:2205.07460*, 2022.

Prabhudesai, M., Ke, T.-W., Li, A., Pathak, D., and Fragkiadaki, K. Diffusion-tta: Test-time adaptation of discriminative models via generative feedback. *Advances in Neural Information Processing Systems*, 36:17567–17583, 2023.

Raman, M., Shah, R., Kannan, A., and Chawla, P. Turn down the noise: Leveraging diffusion models for test-time adaptation via pseudo-label ensembling. In *NeurIPS 2023 Workshop on Distribution Shifts: New Frontiers with Foundation Models*, 2023.

Singhal, R., Horvitz, Z., Teehan, R., Ren, M., Yu, Z., McKeown, K., and Ranganath, R. A general framework for inference-time scaling and steering of diffusion models. *arXiv preprint arXiv:2501.06848*, 2025.

Tsai, Y.-Y., Chen, F.-C., Chen, A. Y., Yang, J., Su, C.-C., Sun, M., and Kuo, C.-H. Gda: Generalized diffusion for robust test-time adaptation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 23242–23251, 2024.

Wang, D., Shelhamer, E., Liu, S., Olshausen, B., and Darrell, T. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*, 2020.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36:31372–31403, 2023.

A. Corruption Abbreviations

This section provides the full names of the corruption-type abbreviations used in Table 1, to help interpret the per-category results.

Table A.1: Definitions of corruption abbreviations.

Abbrev.	Corruption
Def.	Defocus blur
Glass	Glass blur
Mot.	Motion blur
Zoom	Zoom blur
Contr.	Contrast
Elast.	Elastic transform
JPEG	JPEG compression
Pixel	Pixelate
Gauss.	Gaussian noise
Impl.	Impulse noise
Shot	Shot noise
Bright	Brightness change
Fog	Fog
Frost	Frost
Snow	Snow

B. Ablation Studies

B.1. Effects of hyperparamters

Reward Coefficient λ. The coefficient λ rescales the reward in Eq. (1) which is then exponentiated by the difference-potential of Eq. (4) to obtain particle weights. With the candidate–set threshold fixed at $P = 70\%$, the reward values approximately lie in $[-0.25, 0]$; the corresponding potential is therefore upper-bounded by 0.25, so larger λ values exponentially attenuate the weights of low-reward particles at very early steps. Table B.2 lists Top-1 accuracy on ImageNet-C for several λ values, confirming that the default setting $\lambda = 1$ consistently outperforms $\lambda = 5$ across all variants on average.

Adaptive Threshold P. Table B.2 contrasts $P \in \{50\%, 70\%\}$ showing $P = 70$ consistently outperforms which means that in $P = 50$, the ground-truth (GT) label is frequently excluded – expected because corrupted inputs place the GT label deep in the posterior tail.

Table B.2: Ablation on varying λ and adaptive threshold P.

Settings		Category				Avg.
λ	P	Blur	Digital	Noise	Weather	
1	50%	**15.30**	33.20	52.07	29.95	31.33
5	50%	14.80	**33.95**	50.93	29.05	30.93
1	70%	14.95	33.75	**52.20**	**30.55**	**31.57**
5	70%	15.15	32.45	50.07	29.45	30.56

B.2. Robustness Across ImageNet-C Subsets

We randomly sample the ImageNet-C into five disjoint splits, each comprising 100 classes across all 15 corruption types. As Table B.3 shows, our method attains the best Top-1 accuracy on every split, demonstrating that its superiority is robust and not merely the result of a fortuitous partition.

Table B.3: Comparison of results on 5 ImageNet-C subsets.

Method	Subset					Avg.
	1	2	3	4	5	
Baseline	18.60	15.67	19.47	20.47	17.87	18.41
Diffpure	13.87	10.60	13.40	14.40	12.00	12.85
DDA	32.27	28.33	32.27	33.27	29.20	31.07
Grad-DDA	32.47	27.53	32.20	32.93	29.87	31.00
Ours	**33.20**	**28.80**	**32.33**	**33.47**	**30.07**	**31.57**
GT	35.20	30.40	34.33	35.20	32.13	33.45

B.3. Benefit of Post-ensemble Aggregation

Table B.4 compares the effect of the ensemble strategy for DDA, Grad-DDA, SteeringTTA (Ours) and SteeringTTA (GT). The ensemble strategy improves the performance by about 1–2% on average. The improvement with the ensemble for different corruptions is generally not significant except for weather corruption; the increase for weather corruption is up to $+8.8\%$.

Table B.4: Effect of ensemble for the final predictions.

Method	Blur	Digital	Noise	Weather	Avg.
DDA	14.05	33.10	51.80	22.65	28.97
+ Ensemble	14.75	33.75	50.47	30.15	31.07
Grad-DDA	14.50	35.10	50.80	23.65	29.69
+ Ensemble	15.25	34.20	49.47	29.70	31.00
Ours	14.25	33.85	52.87	21.75	29.20
+ Ensemble	14.95	33.75	52.20	30.55	31.57
GT	14.80	35.80	59.20	26.05	32.28
+ Ensemble	15.60	34.70	57.80	31.80	33.45

B.4. Impact of Classifier Guidance Scale for Grad-DDA

Table B.5 provides an ablation study with varying guidance scale $s = \{1, 10\}$ and threshold $P = \{50, 70\}$. The results show that $s = 1$ is significantly better than $s = 10$. However, the gap between $P = 50$ and $P = 70$ is marginal in general.

Table B.5: Ablation on classifier scale s and threshold P for Grad-DDA.

Settings		Category				Avg.
s	P	Blur	Digital	Noise	Weather	
1	50%	**15.25**	**34.40**	48.33	**30.40**	**31.01**
10	50%	9.55	22.45	10.93	23.00	16.85
1	70%	**15.25**	34.20	**49.47**	29.70	31.00
10	70%	9.50	24.35	13.60	23.05	17.89

B.5. Computational Costs

We compare per-image latency at the `p_sample_loop` kernel to isolate adaptation cost. On a single NVIDIA A100 and the brightness corruption, Grad-DDA takes 8.0 seconds, whereas SteeringTTA takes 13.2 seconds. This gap stems from SteeringTTA's multiple particles: it simulates K particles per image, which multiplies neural nets function evaluations (NFEs) for the backbone UNet and requires doubled overall runtime than the single-particle Grad-DDA in the case of $K = 4$.

C. Detailed Related Works

C.1. Diffusion Models for Test-Time Adaptation

Test-time adaptation (TTA) using diffusion models has emerged as a promising research direction to improve robustness of a discriminative model *e.g.*, image classifier, under distribution shifts. Recent works can be categorized into two branches based on what is adapted at test time: (1) joint adaptation that adapts both inputs and model parameters using diffusion-based feedback, and (2) input-only adaptation that refines test inputs via a diffusion model keeping the discriminative model fixed.

Joint Adaptation. The most common way to improve robustness at test time is to update both the input and model weights simultaneously. Raman et al. (2023) applies pseudo-label ensembling to refine the classifier as it transforms each test image (Raman et al., 2023). Similarly, Diffusion-TTA (Prabhudesai et al., 2023) ties the classifier and diffusion model in a feedback loop: the classifier conditions the reverse diffusion process, while diffusion outputs guide small weight updates. SDA (Guo et al., 2024) uses a diffusion model to translate target images into a synthetic domain that mimics the source, then fine-tunes the classifier on this synthetic data so the model itself adapts. These joint strategies may yield higher performance gains than input-only approaches, allowing a discriminative model to actively adapt to generated samples. However, they may be more sensitive to hyperparameters and introduce additional computational overhead.

Input-only Adaptation. Diffpure (Nie et al., 2022) purifies adversarial attacks by using a small diffusion timestep. Diffpure suggests that only adding a small amount of noise and solving the reverse stochastic differential equation in diffusion could effectively wash out adversarial perturbation. However, Diffpure showed performance degradation when it applied to test time adaptation (Tsai et al., 2024; Gao et al., 2023). DDA (Gao et al., 2023) adapts test images only via a diffusion model trained on a source domain. It aligns corrupted images to the source domain by denoising them with low-pass filtering. However, as DDA preserves low-frequency information only using ILVR (Choi et al., 2021), without considering following classification tasks, it often fails to recover images for correct classification. GDA (Tsai et al., 2024) also attempts to denoise corrupted test images using a diffusion model, but it incorporates additional style and semantic constraints in the reverse process. In addition, it minimizes marginal entropy during the reverse process considering downstream classification tasks. However, as it is computed only on top-1 pseudo-labels, it may not be reliable when class predictions are wrong. Both DDA and GDA rely on gradient-based diffusion process, which limits the guidance to differentiable objectives.

C.2. Sequential Monte Carlo for Diffusion Models

Some recent works in diffusion models employ Sequential Monte Carlo (SMC) for more flexible sampling in the reverse process. In contrast to gradient-based guidance, with SMC, particles (or samples) evolve through diffusion processes; they are reweighted and resampled according to an user-defined criterion. A practical trigger for resampling is the *effective sample size* (ESS):

$$\mathrm{ESS}_t = \Big[\sum_{i=1}^{K} (\hat{G}_t^i)^2\Big]^{-1},$$

where $\hat{G}_t^i$ are the normalized potentials. When $\mathrm{ESS}_t < 0.5K$, resampling prevents weight collapse and maintains particle diversity (Singhal et al., 2025; Wu et al., 2023).

By biasing the sampling distribution toward higher-potential regions, SMC can incorporate non-differentiable rewards without retraining or backpropagating through the diffusion model. This property is particularly appealing for test-time adaptation, where reward functions may be implicit or non-differentiable.

Wu et al. (2023) introduces a twisted diffusion sampler offering asymptotically exact conditional generation via SMC, outperforming naive conditional heuristics on tasks like image inpainting. Kim et al. (2025) proposes test-time Diffusion Alignment as Sampling(DAS), which uses SMC to maximize a reward that reflects alignment to a given goal, avoiding reward over-optimization issues common in RL fine-tuning and reward under-optimization issues in gradient-guidance. Singhal et al. (2025) presents a comprehensive Feynman-Kac (FK) steering framework that formalizes diffusion models with SMC, allowing arbitrary, possibly non-differentiable rewards to modify the generative trajectory without updating model parameters. Their method shows strong performance on text-to-image tasks, often rivaling specialized fine-tuned models, all via particle-based sampling. Various works also explore SMC to address domain gaps in designing biological sequence (Li et al., 2024), text generation (Singhal et al., 2025) and inverse problems (Cardoso et al., 2023) highlighting the versatility of particle filtering strategies for diffusion models.

D. Qualitative Results

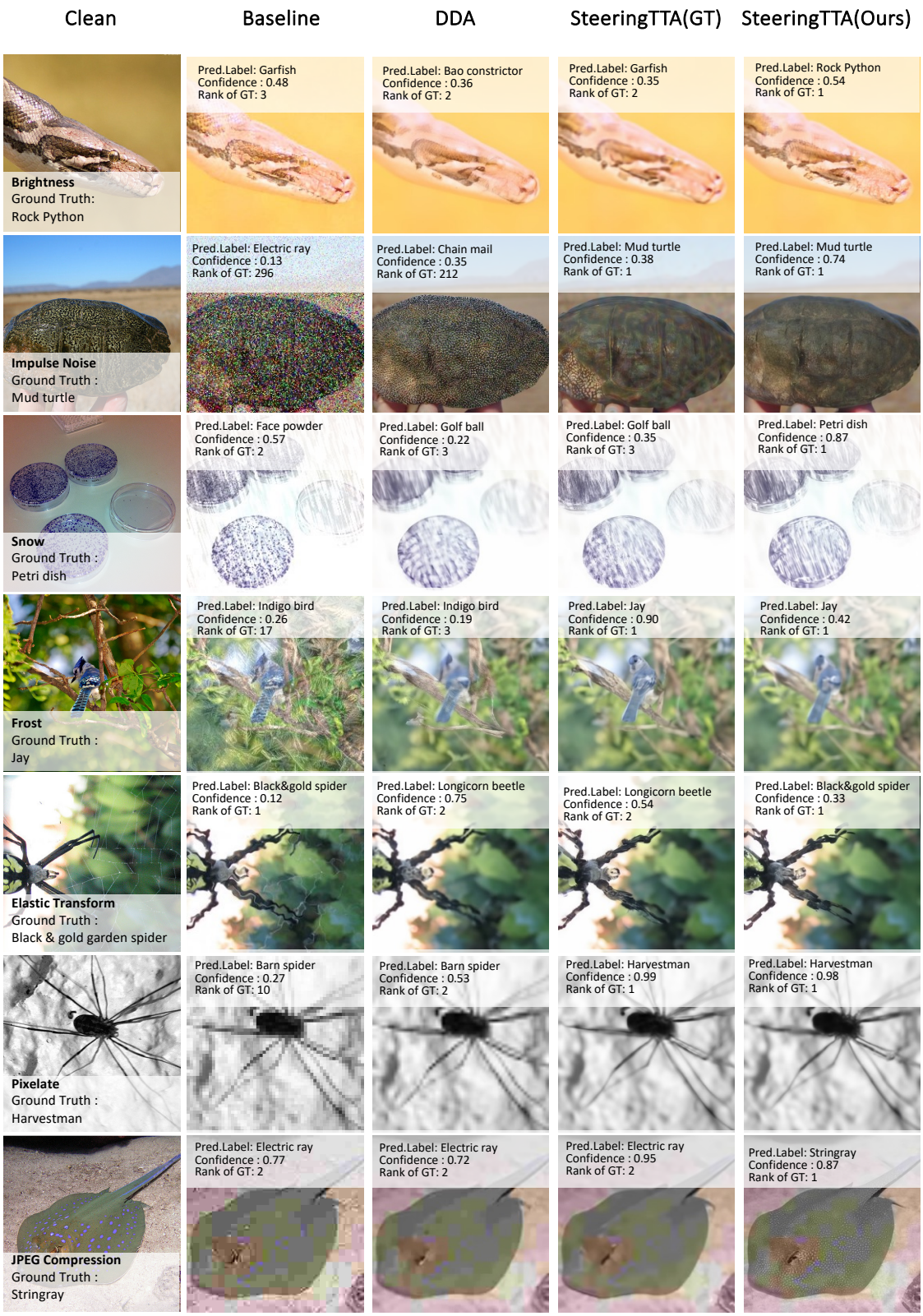

Figure D.1: Qualitative results comparing the original corrupted image, DDA, GT-based SteeringTTA and ours.

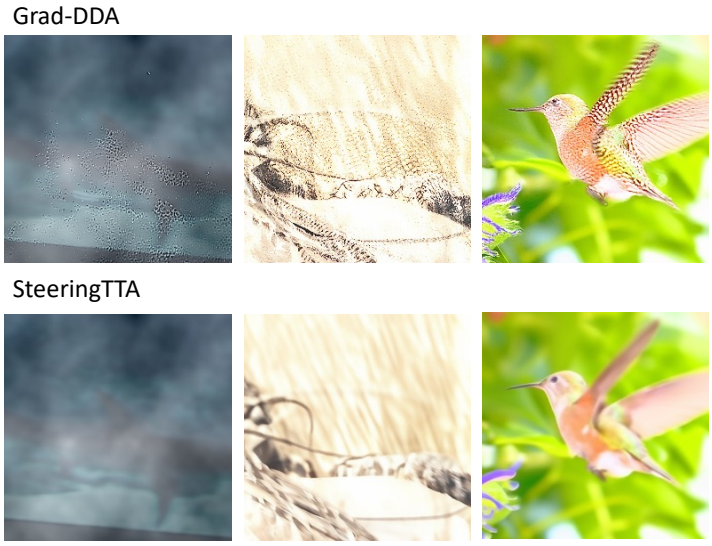

Figure D.2: From top to bottom, the adapted image with Grad-DDA ($scale = 10$) and our method's with same rewards. Some unreliable high-frequency artifacts appear in Grad-DDA which are not in ours.