Manipulation of noise in generative modeling

Supervisor: Luca Eyring, Akata Lab

Student: Artem Serebriakov

Introduction

Common issues of Text-to-Image models:

- Incorrect text rendering;
- Poor attribute binding;
- Image inaccuracies.

Approaches to address these issues:

- Employ enhanced language encoders and larger diffusion models;
- Fine-tune T2I models.

Human preference reward models

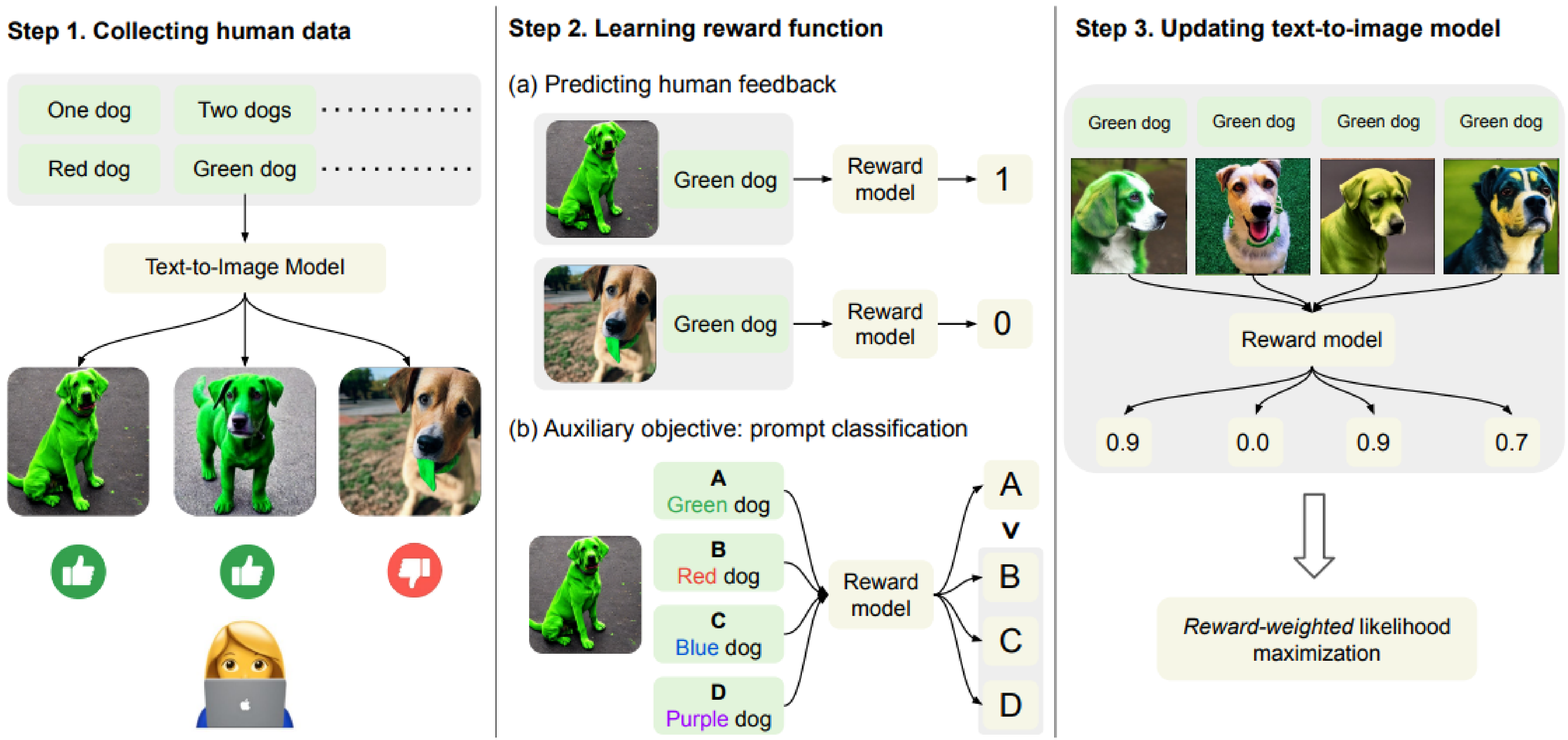
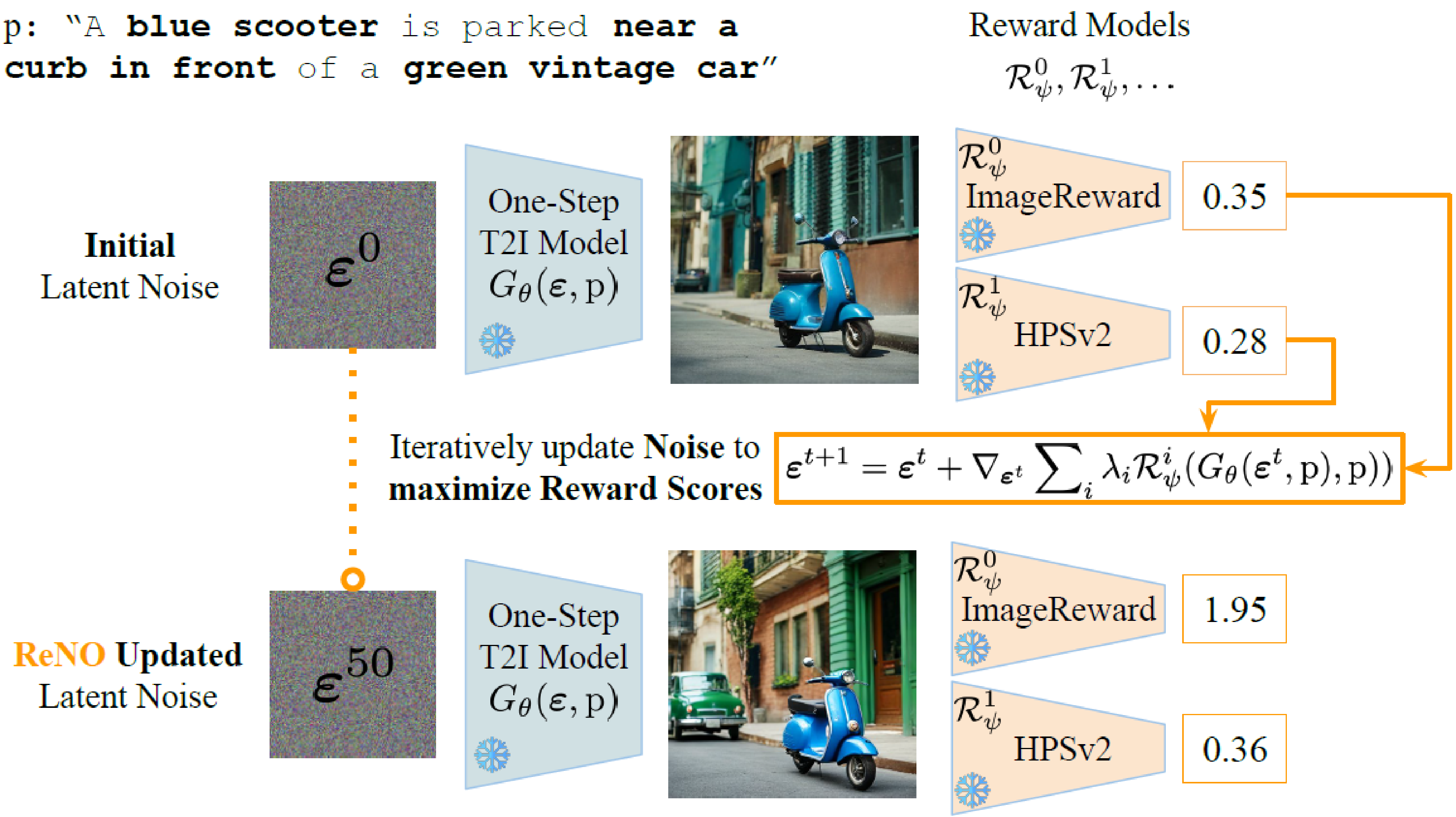


Figure 1. The steps in our fine-tuning method. (1) Multiple images sampled from the text-to-image model using the same text prompt, followed by collection of (binary) human feedback. (2) A reward function is learned from human assessments to predict image-text alignment. We also utilize an auxiliary objective called prompt classification, which identifies the original text prompt within a set of *perturbed* text prompts. (3) We update the text-to-image model via reward-weighted likelihood maximization.

Motivation for ReNO

- Can we improve T2I models at inference time without any fine-tuning?
- Does the initial noise have a huge effect on generations?
- How can we modify the initial noise to get desirable outputs?

ReNO: Enhancing One-step Text-to-Image Models through Reward-based Noise Optimization



Background: One-Step Diffusion Models

$$\mathbf{x}_t = \alpha_t \mathbf{x}_0 + \sigma_t \varepsilon, \tag{1}$$

where $G_\theta(\varepsilon, \mathrm{p}) = \mathrm{x}_0$, Text-to-Image generative model $G_\theta(\varepsilon, \mathrm{p})$, noise vector $\varepsilon \sim \mathcal{N}(0, \mathbf{I})$, prompt p, parameters θ, α_t is a decreasing and σ_t is an increasing function of t.

Forward SDE: $\mathrm{d}\mathbf{x}_t = \mathbf{u}(\mathbf{x}_t, t)\, \mathrm{d}t + g(t)\, \mathrm{d}\mathbf{w}_t$

where $\mathbf{u}_t(x_t, t)$ — drift, $\mathbf{w}_t$ — Wiener process, $g(t)$ — diffusion schedule.

Reverse-time SDE: $\mathrm{d}\mathbf{x}_t = [\mathbf{u}(\mathbf{x}_t, t) - g(t)^2 \mathbf{s}(\mathbf{x}_t, t)]\, \mathrm{d}t + g(t)\, \mathrm{d}\bar{\mathbf{w}}_t, \quad (2)$

where $\mathrm{x}_T = \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \mathrm{x}_0 \sim p_0(\mathrm{x})$, $\mathrm{s}(\mathrm{x}, t) = \nabla \log p_t(\mathrm{x})$ is the score function.

Denoising loss: $\mathcal{L}_{\mathrm{s}}(\theta) = \mathbb{E}_{\mathbf{x}_0 \sim p(\mathbf{x}_0), \boldsymbol{\varepsilon} \sim \mathcal{N}(0, \mathbf{I}), t \sim \mathcal{U}(0, T)} [\|\sigma_t \mathbf{s}_\theta(\mathbf{x}_t, t) + \varepsilon\|^2]. \quad (3)$

where parameterized score $\mathrm{s}_\theta(\mathrm{x}_t, t)$.

Distillation

- Adversarial Diffusion Distillation (ADD) combines score distillation with an adversarial loss and is employed to train SD-Turbo based on SD 2.1 as a teacher and SDXL-Turbo based on SDXL;
- Diffusion Matching Distillation (DMD) additionally leverages a distributional loss based on an approximated KL divergence and is applied for PixArt-α DMD;
- Trajectory Segmented Consistency Distillation (TSCD) introduces a progressive segment-wise consistency distillation loss to train HyperSDXL [65] with reward fine-tuning.

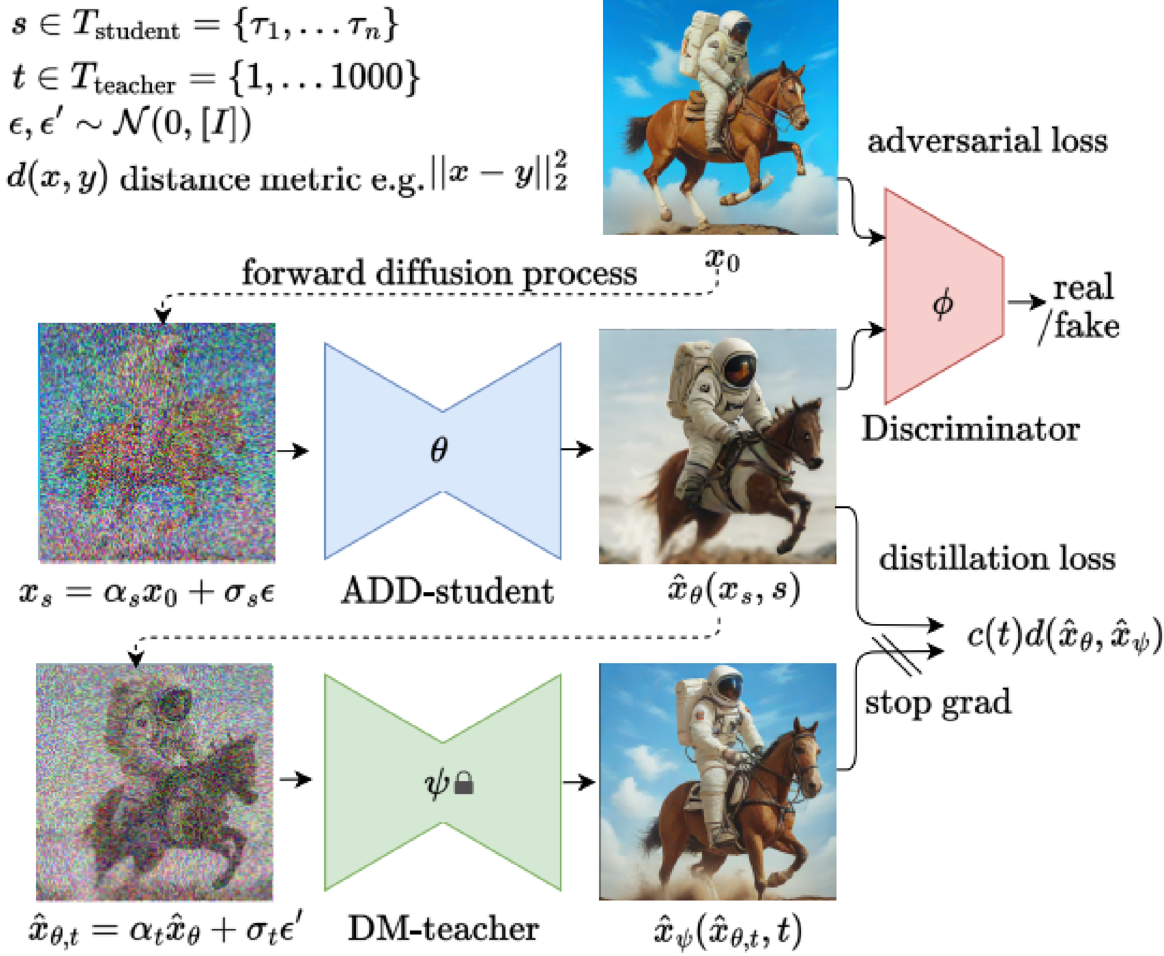


Figure 2. **Adversarial Diffusion Distillation.** The ADD-student is trained as a denoiser that receives diffused input images x_s and outputs samples $\hat{x}_\theta(x_s, s)$ and optimizes two objectives: a) adversarial loss: the model aims to fool a discriminator which is trained to distinguish the generated samples $\hat{x}_\theta$ from real images x_0. b) distillation loss: the model is trained to match the denoised targets $\hat{x}_\psi$ of a frozen DM teacher.

Initial Noise Optimization and Regularization

Optimization problem: $\varepsilon^\star = \arg\max_{\varepsilon} \mathcal{C}(G_\theta(\varepsilon, \mathrm{p})).$ (4)

where $\mathcal{C} : \mathbb{R}^{H\times W\times c} \to \mathbb{R}$ — criterion function, prompt p , noise ε.

Regularized criterion function: $\mathcal{C}(\mathrm{x}_0, \varepsilon) = \tilde{\mathcal{C}}(\mathrm{x}_0) + K(\varepsilon)$,

where $K(\varepsilon)$ — regularization term.

$$\tilde{\mathcal{C}}(\mathrm{x}_0) = \sum_{i,j} \mathrm{x}_0^{i,j,c} - \mathrm{x}_0^{i,j,\bar{c}_1} - \mathrm{x}_0^{i,j,\bar{c}_2}, \quad (5)$$

where $\mathrm{x}_0^{i,j,c}$ — channel c of the pixel at (i, j).

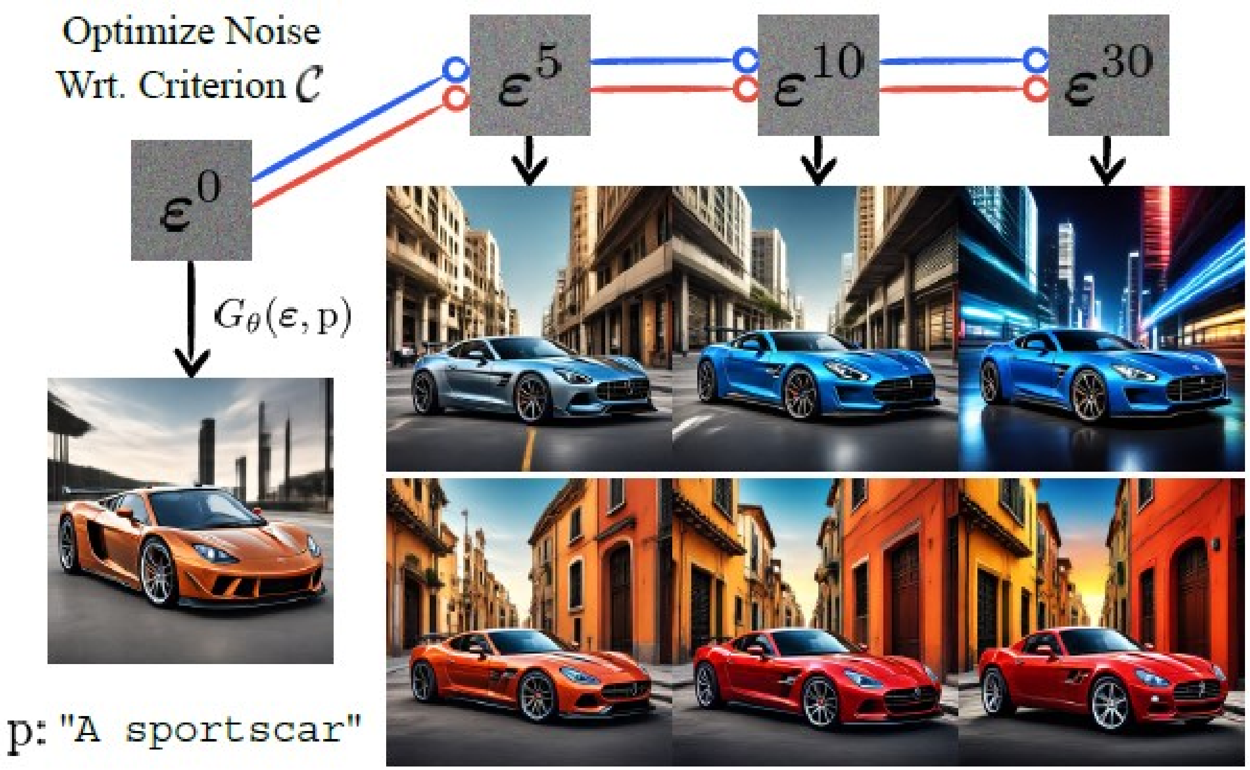


Figure 3: Initial noise optimization for one-step G_θ HyperSDXL with two color channel criterions (5).

ReNO Reward Criterion

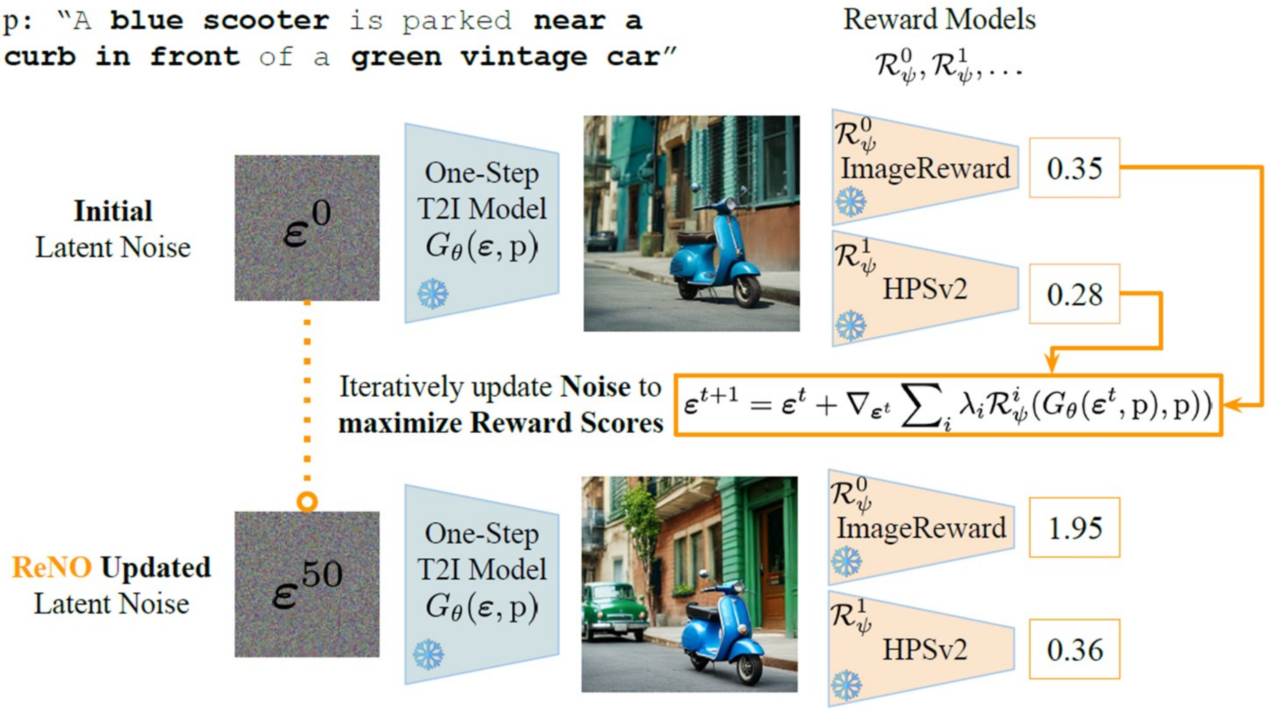
Reward-based criterion function for Noise Optimization (ReNO):

$$\tilde{C}(\mathrm{x}_0, \mathrm{p}) = \sum_{i}^{n} \lambda_i \mathcal{R}_\psi^i(\mathrm{x}_0, \mathrm{p}), \tag{6}$$

where pre-trained reward models $\mathcal{R}_\psi^0, \ldots \mathcal{R}_\psi^n$, λ_i denotes the weighting for reward model $\mathcal{R}_\psi^i$.

Gradient ascent: $$\varepsilon^{t+1} = \varepsilon^t + \eta \nabla_{\varepsilon^t}[K(\varepsilon^t) + \sum_{i}^{n} \lambda_i \mathcal{R}_\psi^i(G_\theta(\varepsilon^t, \mathrm{p}), \mathrm{p})], \tag{7}$$

where η is the learning rate.



Algorithm 1 ReNO

Input: p (prompt), G_θ (One-Step T2I Model), $\mathcal{R}_\psi^{0,1\ldots n}$ (Reward Functions), $\lambda_{0,1\ldots n}$ (Reward Weights), m (# Optimization Steps), η (Learning Rate), λ_{reg} (Regularization Strength)

Initialize $v_{-1} = 0.0$, $\varepsilon^0 = \mathcal{N}(0, \mathbf{I})$, $R^\star = -\inf$.

for $t = 0$ **to** m **do**

  Generate image $\mathrm{x}_0^t = G_\theta(\varepsilon^t, p)$

  Compute reward-based criterion $R^t = \sum_i^n \lambda_i \mathcal{R}_\psi^i(\mathrm{x}_0^t, \mathrm{p})$

  $\mathrm{grad}_t = \nabla_{\varepsilon^t}[\lambda_{\text{reg}} K(\varepsilon^t) + R^t]$

  $\mathrm{grad}_t = \mathrm{GradNormClip}(\mathrm{grad}_t, 0.1)$

  $v_t = 0.9 \cdot v_{t-1} + \eta \cdot \mathrm{grad}_t$

  $\varepsilon^{t+1} = \varepsilon^t + v_t$

  if $R^t > R^\star$ **then**

    $\mathrm{x}_0^\star = \mathrm{x}_0^t$, $R^\star = R^t$

  end if

end for

return $\mathrm{x}_0^\star$

Experimental Setup

One-Step T2I Models			
SD-Turbo	SDXL-Turbo	PixArt-α DMD	HyperSDXL

Benchmarks		
T2I-CompBench	GenEval	Parti-Prompts

Reward Models			
PickScore	HPSv2	ImageReward	CLIPScore

Effect of Reward Models

Table 1: SD-Turbo evaluated on the attribute binding categories of T2I-CompBench and the LAION aesthetic score predictor [74] for different reward models.

Reward	**Attribute Binding**			**Aesthetic** ↑
	Color ↑	**Shape** ↑	**Texture** ↑	
SD-Turbo	0.5513	0.4448	0.5690	5.647
+ CLIPScore	0.6625	0.5501	0.6621	5.475
+ HPSv2	0.6443	0.5451	0.6859	5.752
+ ImageReward	0.7720	0.6104	0.7334	5.611
+ PickScore	0.6341	0.5069	0.6242	5.711
+ All	0.7830	0.6244	0.7466	5.704

Quantitative Results

Table 2: **Quantitative Results on T2I-CompBench**. ReNO combined with (1) PixArt-α DMD [11, 12, 91], (2) SD-Turbo [72], (3) SDXL-Turbo [72], HyperSD [65] demonstrates superior compositional generation ability in both attribute binding, object relationships, and complex compositions. The best value is bolded, and the second-best value is underlined. Multi-step results taken from [12, 20].

Model	Attribute Binding			Object Relationship		Complex↑
	Color ↑	Shape↑	Texture↑	Spatial↑	Non-Spatial↑	
SD v1.4	0.38	0.36	0.42	0.12	0.31	0.31
SD v2.1	0.51	0.42	0.49	0.13	0.31	0.34
SDXL	0.64	0.54	0.56	0.20	0.31	0.41
PixArt-α	0.69	0.56	0.70	0.21	**0.32**	0.41
DALL-E 2	0.57	0.55	0.64	0.13	0.30	0.37
DALL-E 3	**0.81**	**0.68**	**0.81**	-	-	-
(1) PixArt-α DMD	0.38	0.34	0.47	0.19	0.30	0.36
(1) + ReNO (Ours)	0.64	0.57	0.72	0.25	0.31	0.46
(2) SD-Turbo	0.55	0.44	0.57	0.17	0.31	0.41
(2) + ReNO (Ours)	0.78	0.62	0.75	0.22	**0.32**	**0.48**
(3) SDXL-Turbo	0.61	0.44	0.60	0.24	0.31	0.43
(3) + ReNO (Ours)	0.78	0.60	0.74	**0.26**	0.31	0.47
(4) HyperSDXL	0.65	0.50	0.65	0.25	0.31	0.46
(4) + ReNO (Ours)	<u>0.79</u>	<u>0.63</u>	<u>0.77</u>	**0.26**	0.31	**0.48**

Table 3: **Quantitative Results on GenEval**. ReNO combined with (1) PixArt-α DMD [11, 12, 91], (2) SD-Turbo [72], (3) SDXL-Turbo [72], HyperSDXL [65] improves results across all categories. The best value is bolded, and the second-best value is underlined. Multi-step results taken from [20].

Model	Mean ↑	Single↑	Two↑	Counting↑	Colors↑	Position↑	Color Attribution↑
SD v2.1	0.50	0.98	0.51	0.44	0.85	0.07	0.17
SDXL	0.55	0.98	0.74	0.39	0.85	0.15	0.23
IF-XL	0.61	0.97	0.74	0.66	0.81	0.13	0.35
PixArt-α	0.48	0.98	0.50	0.44	0.80	0.08	0.07
DALL-E 2	0.52	0.94	0.66	0.49	0.77	0.10	0.19
DALL-E 3	<u>0.67</u>	0.96	<u>0.87</u>	0.47	0.83	**0.43**	**0.45**
SD3 (8B)	**0.68**	0.98	0.84	<u>0.66</u>	0.74	<u>0.40</u>	<u>0.43</u>
(1) PixArt-α DMD	0.45	0.95	0.38	0.46	0.76	0.05	0.09
(1) + ReNO (Ours)	0.59	0.98	0.72	0.58	0.85	0.15	0.27
(2) SD-Turbo	0.49	0.99	0.51	0.38	0.85	0.07	0.14
(2) + ReNO (Ours)	0.62	**1.00**	0.82	0.60	0.88	0.12	0.33
(3) SDXL-Turbo	0.54	**1.00**	0.66	0.45	0.84	0.09	0.20
(3) + ReNO (Ours)	0.65	**1.00**	0.84	**0.68**	<u>0.90</u>	0.13	0.35
(4) HyperSDXL	0.56	**1.00**	0.76	0.43	0.87	0.10	0.21
(4) + ReNO (Ours)	0.65	**1.00**	**0.90**	0.56	**0.91**	0.17	0.33

User Study Results and Computational Cost of ReNO

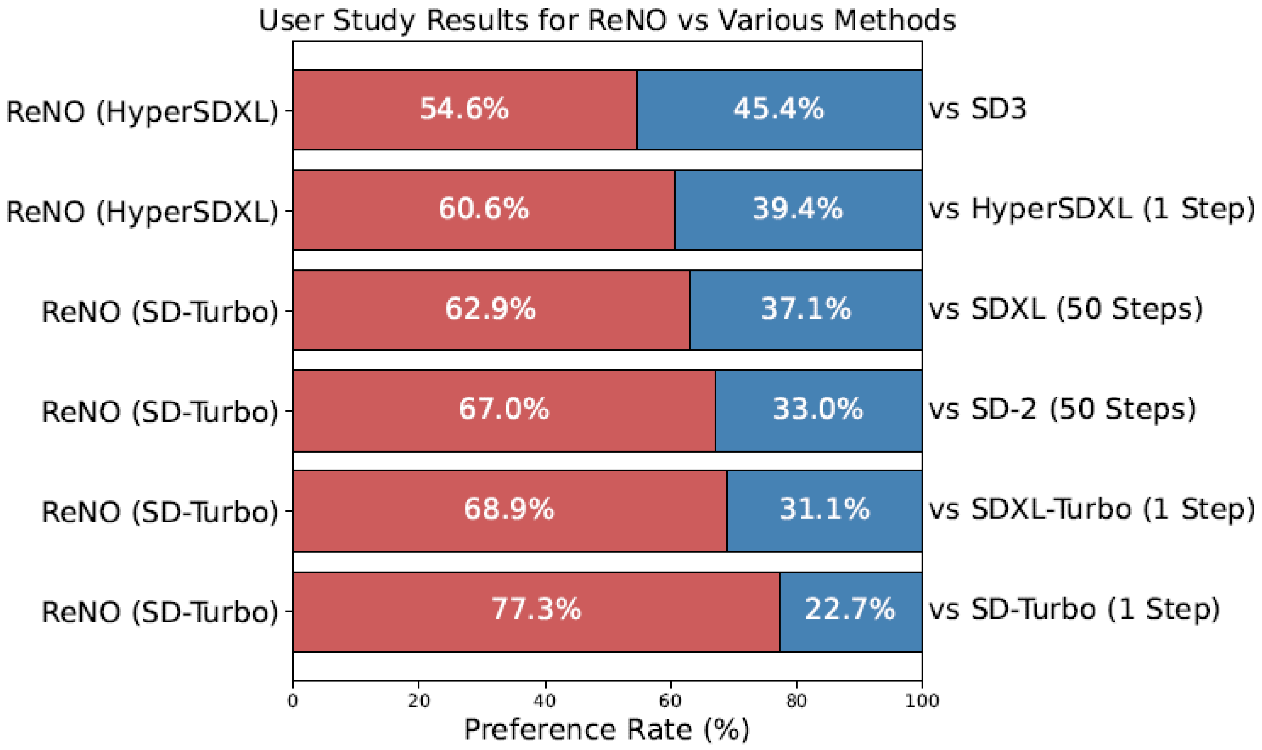


Figure 4: User Study Results for ReNO

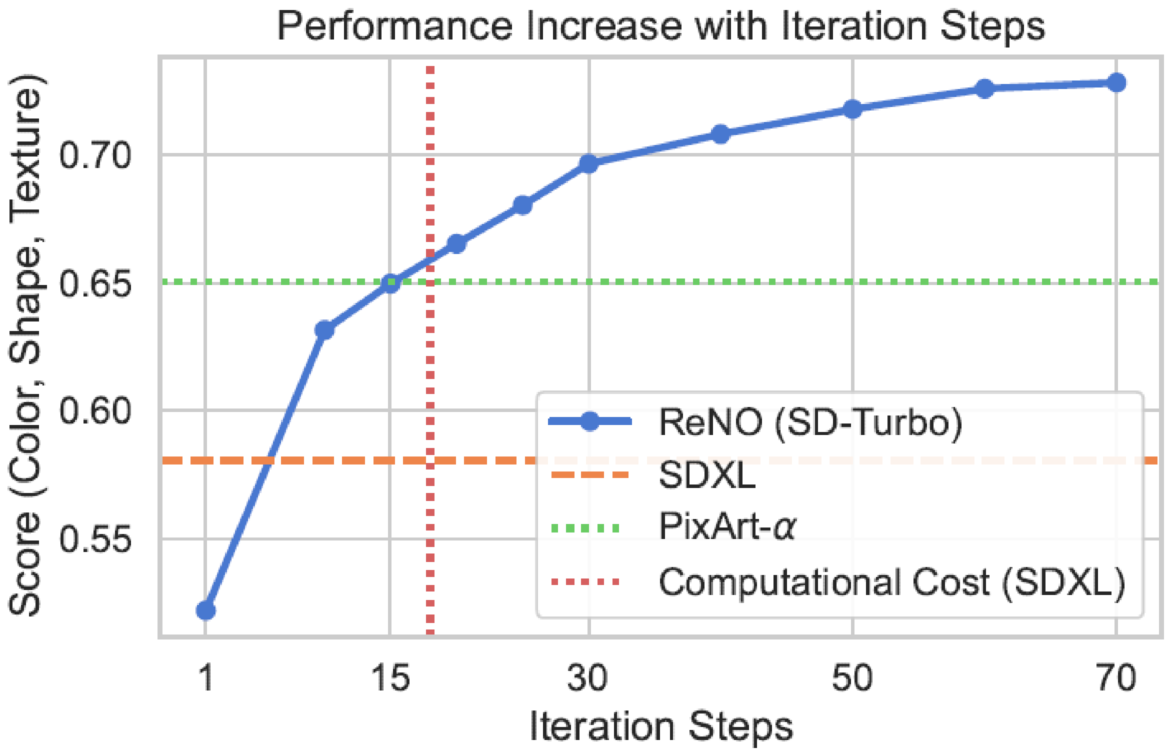


Figure 5: Attribute binding results on T2I-CompBench with varying number of iterations.

Limitations

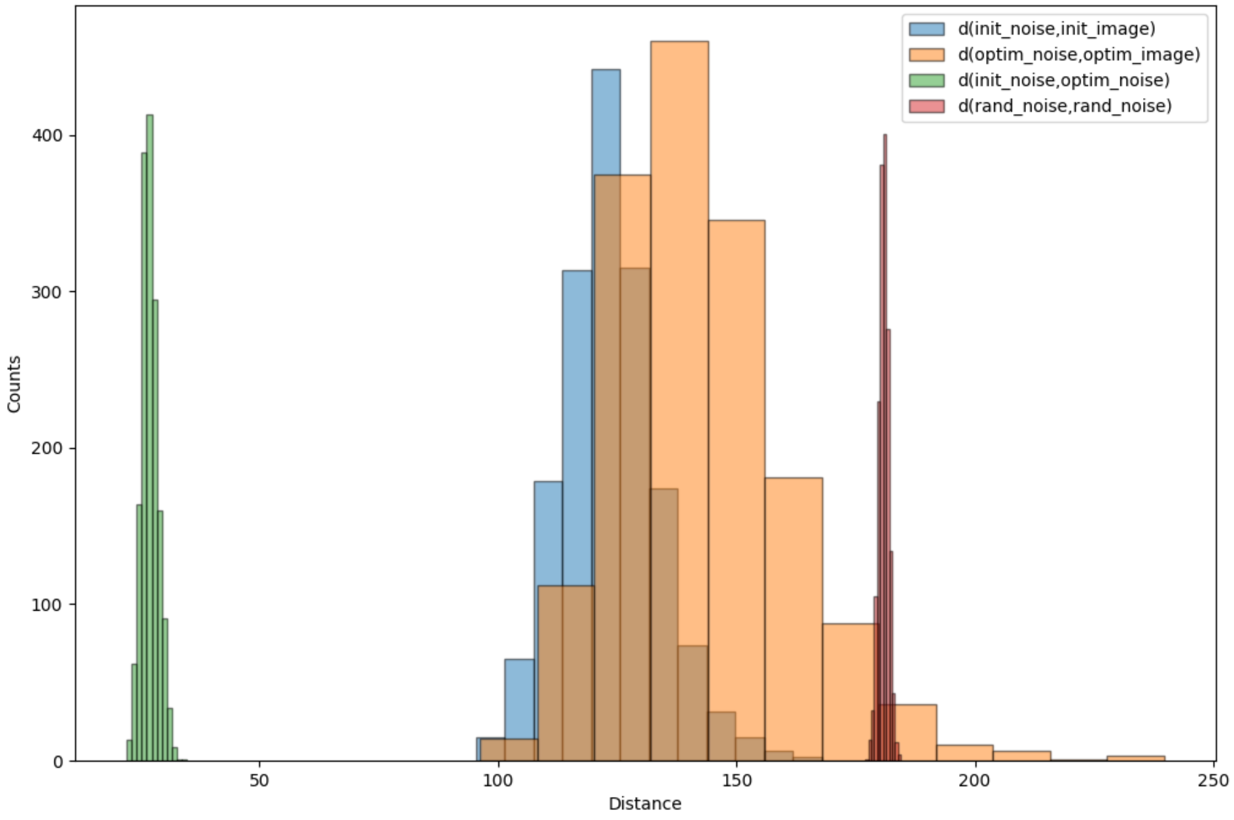
- Despite using different image generation models of varying architectures and sizes, they converge to similar performance on both T2I-Compbench and GenEval;
- The amount of needed GPU VRAM is significantly higher when using ReNO;
- ReNO is designed for distilled diffusion models.

Further analysis

- Goal: understand how the optimization process changes the initial noise;
- Metrics: L2 distance, coherence function;
- Parti-Prompts dataset (a set of diverse 1.6k prompts).

Analysis of the distances between noises and images

- Mean L2 distance between initial and optimized noises: 27.0122;
- Mean L2 distance between two random noises: 180.7667;
- Mean L2 distance between initial noise and generated image: 123.3377;
- Mean L2 distance between optimized noise and generated image: 142.3693;
- Mean L2 distance between random noise and random image: 159.0550.

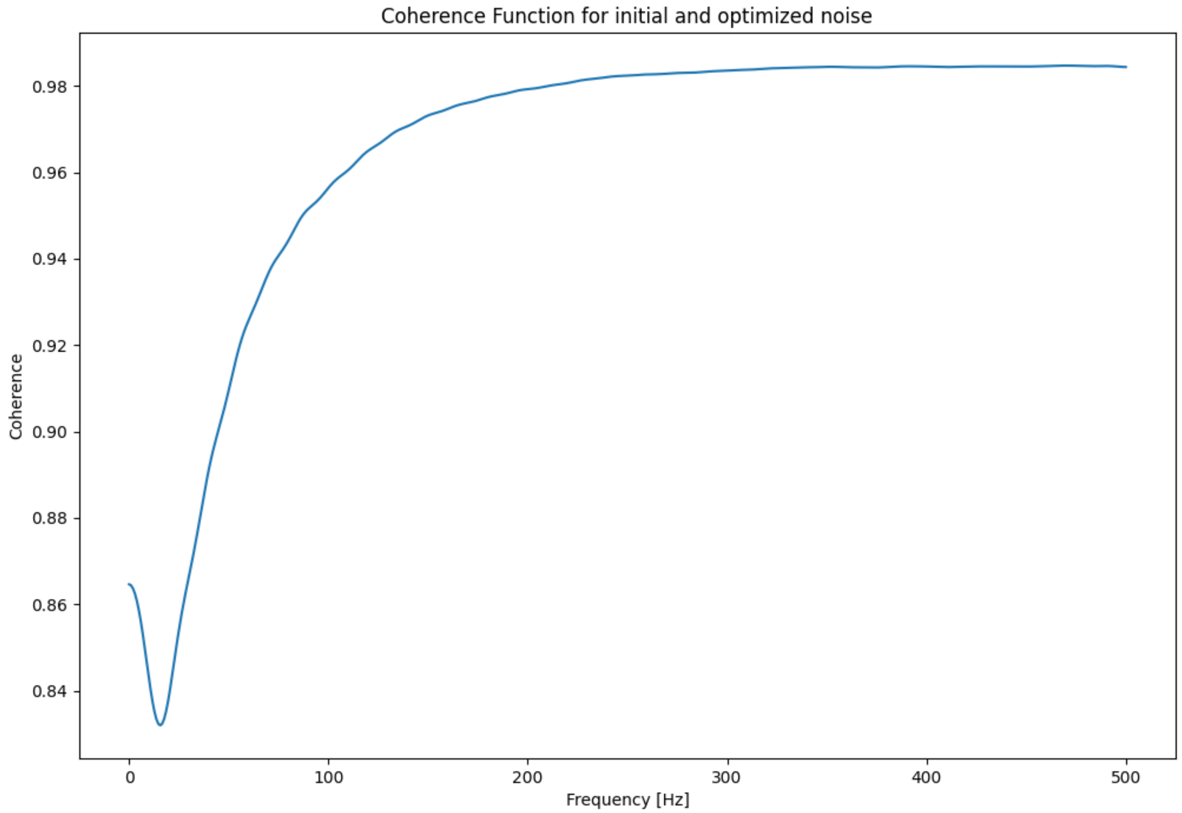


Conclusion:

- Mean L2 distance between the initial and optimized noises is much less than distance between random noises;

- Mean L2 distance between the initial noise and image is smaller than the distance between the optimized ones.

Coherence analysis of the initial and optimized noises

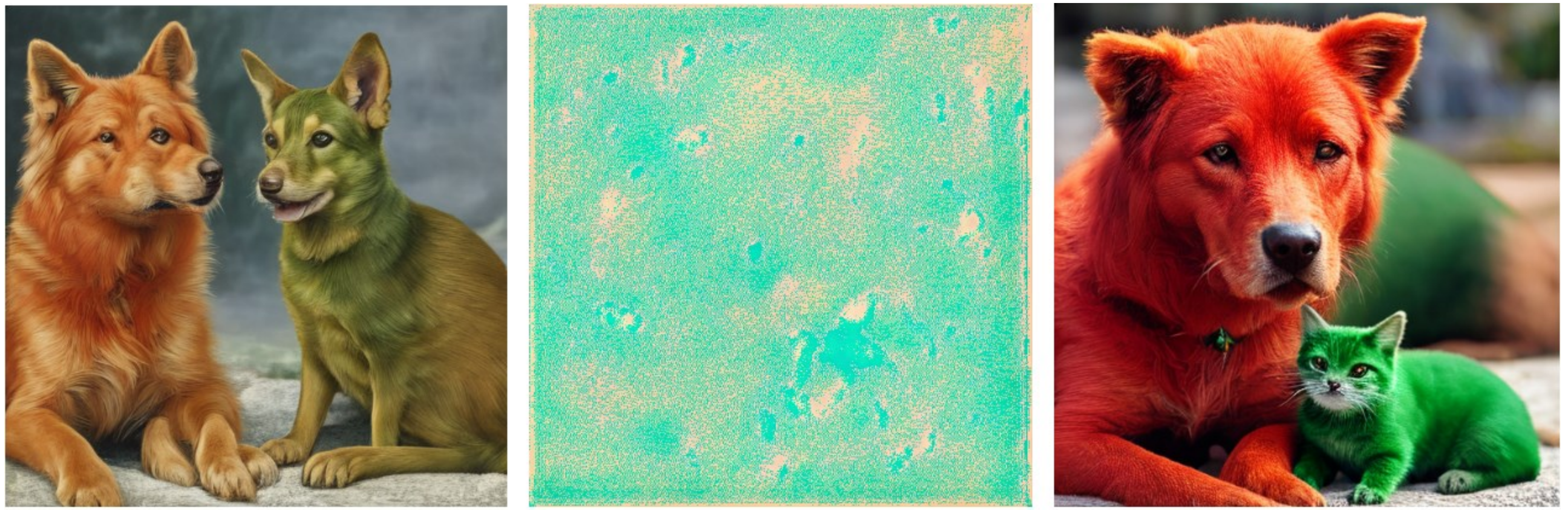


- Coherence function measures the frequency-dependent correlation between two noises;
- Coherence values range from 0 (no correlation) to 1 (perfect correlation) and are computed across different frequencies;
- Low frequencies represent the "smooth" or slowly varying parts of the signal;
- High frequencies correspond to the fast-changing or detailed parts of the signal;

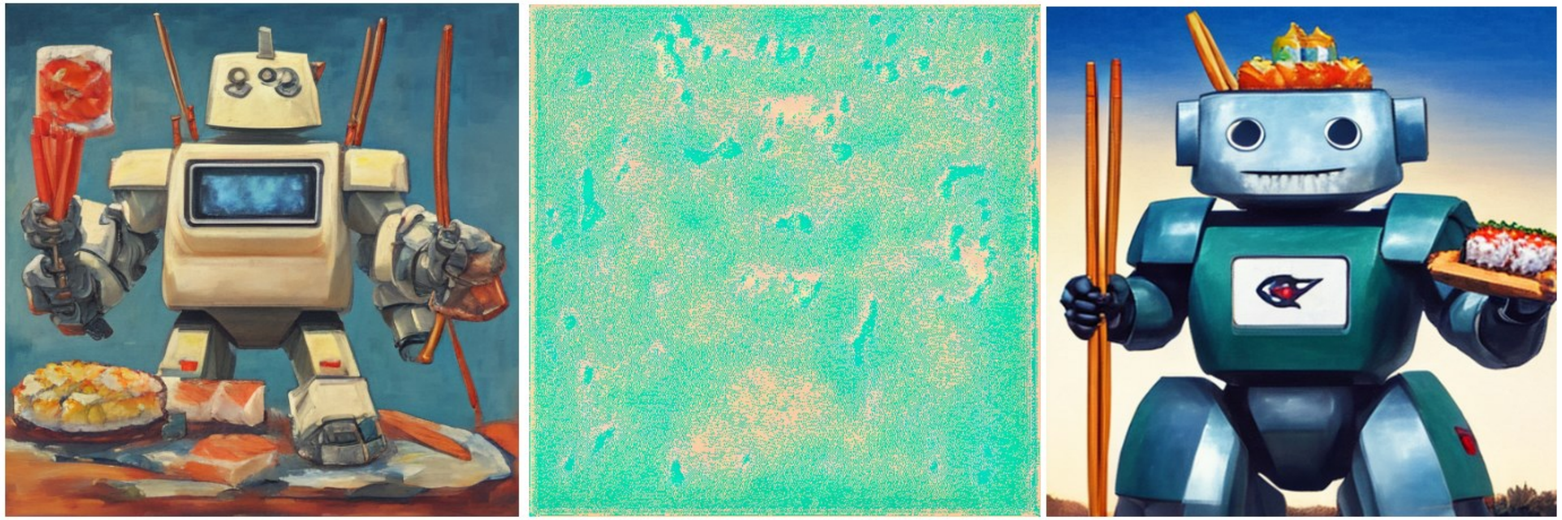
Conclusion: ReNO primarily modifies the slower-varying components while preserving the rapid fluctuations.

Visualization of the difference between the initial and optimized noises

Prompt: "A red dog and a green cat"



Prompt: "Oil painting of a giant robot made of sushi, holding chopsticks."



Conclusion: No visual difference between the initial and optimized noises.

Findings

- Mean L2 distance between the initial and optimized noises is much less than distance between random noises;
- Mean L2 distance between the initial noise and image is smaller than the distance between the optimized ones;
- Coherence function is lower at low frequencies and it is increasing at higher frequencies;
- No visual difference between the initial and optimized noises.

Thank you for your attention!