NeuralRemaster: Phase-Preserving Diffusion for Structure-Aligned Generation

Yu Zeng[1] Charles Ochoa[1] Mingyuan Zhou[2] Vishal M. Patel[3] Vitor Guizilini[1] Rowan McAllister[1]
[1]Toyota Research Institute [2]University of Texas, Austin [3]Johns Hopkins University

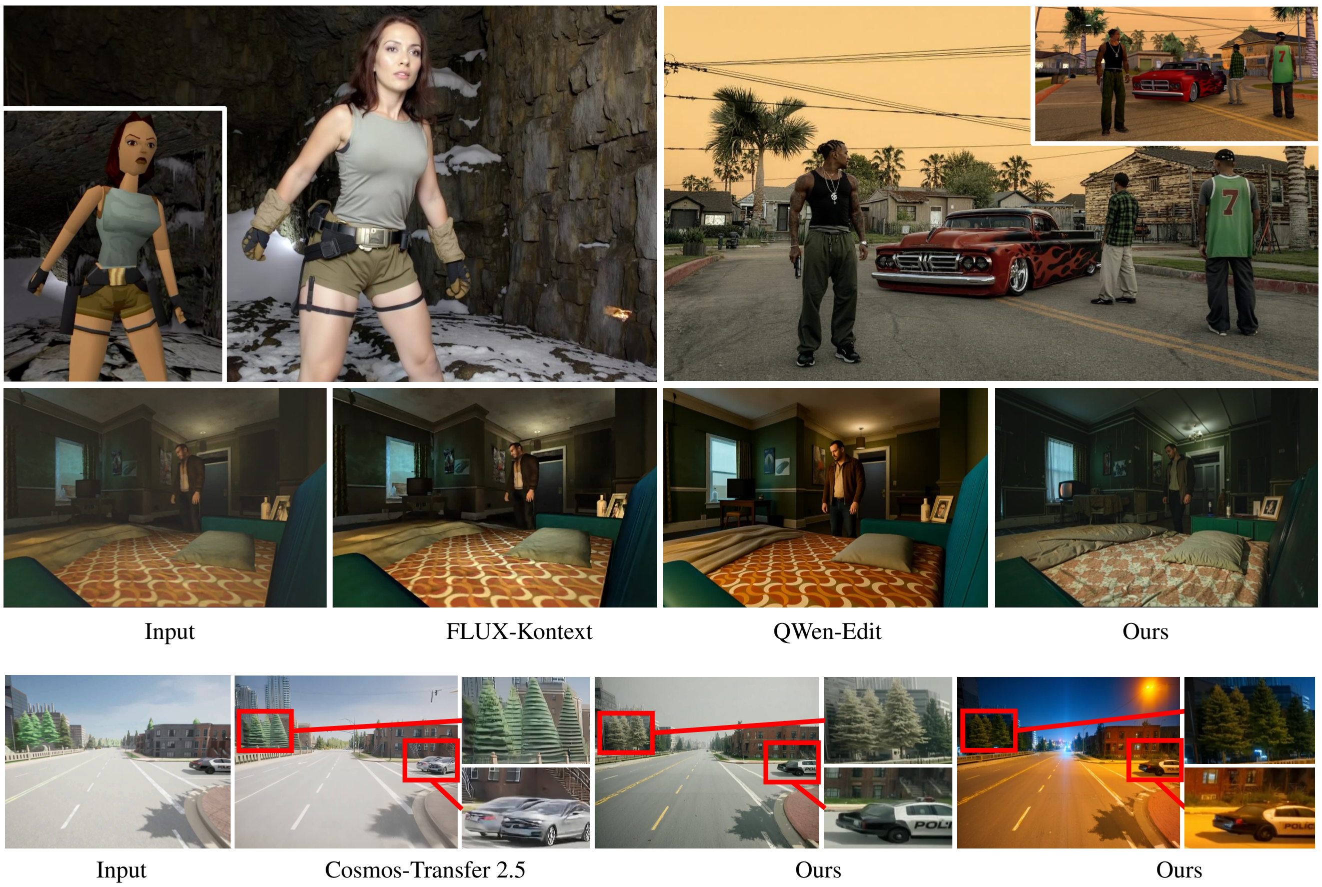


Figure 1. We present Phase-Preserving Diffusion (ϕ-PD), a model-agnostic reformulation of the diffusion process that preserves an image's phase while randomizing its magnitude, enabling structure-aligned generation with **no architectural changes or additional parameters**.

Abstract

Standard diffusion corrupts data using Gaussian noise whose Fourier coefficients have random magnitudes and random phases. While effective for unconditional or text-to-image generation, corrupting phase components destroys spatial structure, making it ill-suited for tasks requiring geometric consistency, such as re-rendering, simulation enhancement, and image-to-image translation. We introduce Phase-Preserving Diffusion (ϕ-PD), a model-agnostic reformulation of the diffusion process that preserves input phase while randomizing magnitude, enabling structure-aligned generation without architectural changes or additional parameters. We further propose Frequency-Selective Structured (FSS) noise, which provides continuous control over structural rigidity via a single frequency-cutoff parameter. ϕ-PD adds no inference-time cost and is compatible with any diffusion model for images or videos. Across photorealistic and stylized re-rendering, as well as sim-to-real enhancement for driving planners, ϕ-PD produces controllable, spatially aligned results. When applied to the CARLA simulator, ϕ-PD improves CARLA-to-Waymo planner performance by 50%. The method is complementary to existing conditioning approaches and broadly applicable to image-to-image and video-to-video generation. Videos, additional examples, and code are available on our project page.

1. Introduction

Recent advances in diffusion models have revolutionized image generation, achieving high-fidelity results for unconditional or text-conditioned synthesis. Yet many practical applications do not require generating a scene from scratch. Instead, they operate within an image-to-image setting where the spatial layout, such as object boundaries, geometry and scene structures, should remain fixed while the appearance is modified. Examples include neural rendering, stylization, and sim-to-real transfer for autonomous driving or robotics simulation. We refer to this broad class of problems as **structure-aligned generation**.

Although these tasks are conceptually easier than generating an image from scratch, existing solutions are unnecessarily complex. Methods such as ControlNet [42], T2I-Adapter [21], and related variants attach auxiliary encoders, or adapter branches to inject structural input into the model. While effective, this introduces additional parameters and computational cost, paradoxically making structure-aligned generation harder than it should be.

We argue that this inefficiency stems not from the network architecture, but from the diffusion process itself. The forward diffusion process injects Gaussian noise, which destroys both the magnitude and phase components in the frequency domain. Classical signal processing [23, 30, 37], however, tells us that phase encodes structure while magnitude encodes texture. Destroying the phase means destroying the very spatial coherence that structure-aligned generation depends on, forcing the model to reconstruct structure from scratch.

Motivated by this insight, we propose Phase-Preserving Diffusion (ϕ-PD). Instead of corrupting data with Gaussian noise, ϕ-PD constructs *structured noise* whose magnitude matches that of Gaussian noise while preserving the input phase. This naturally maintains spatial alignment throughout sampling (Figure 1) with **no architectural modification, no extra parameters** (Figure 2), and is compatible with any DDPM or flow-matching model for images or videos.

To provide controllable levels of structural rigidity, we further introduce **Frequency-Selective Structured (FSS) noise**, which interpolates between input phase and pure Gaussian noise via a single cutoff parameter (Figure 4). This allows us to control the trade-off between strict alignment and creative flexibility.

We evaluate ϕ-PD across photorealistic re-rendering, stylized re-rendering and simulation enhancement for embodied-AI agents. ϕ-PD consistently maintains geometry alignment while producing high-quality visual outputs, outperforming prior methods across both quantitative and qualitative metrics. When used to enhance CARLA simulations, ϕ-PD improves planner transfer to the Waymo Open Dataset by 49%, substantially narrowing the sim-to-real gap. In summary, our contributions include:

- **Phase-preserving diffusion process:** A diffusion process that preserves phase while randomizing magnitude in frequency domain, maintaining spatial structure without architectural changes.
- **Frequency-selective structured noise:** A single-parameter mechanism that enables continuous control over structural alignment rigidity.
- **Unified and efficient framework** applicable to both images and videos, compatible with DDPMs and flow-matching, and requires no inference-time overhead.

2. Related Work

Diffusion Models. Diffusion models have become a dominant paradigm for generative modeling, capable of representing complex data distributions with remarkable fidelity [11, 16]. They progressively corrupt data into Gaussian noise through a forward diffusion process, then learn to invert this process via iterative denoising. This framework has demonstrated state-of-the-art performance across diverse domains, including image, video, and audio generation [3, 4, 12, 17, 25, 28, 31], as well as reinforcement learning [14, 26, 36] and robotics [1, 5, 35].

Frequency-Domain Manipulation for Diffusion. Recent work has explored frequency domain operations for diffusion models. [7] argues that diffusion models of images perform approximate autoregression in the frequency domain. [40] shows that modifying the UNet frequency domain features significantly improves the generating quality for image or video generation. FreeDiff [38] introduces a fine-tuning free approach for image editing that employs progressive frequency truncation to refine the guidance of diffusion models. [27] proposed to use a frequency-dependent moving average during sampling. [2] proposes a training-free approach for image inpainting that optimizes the initial seed noise in the spectral domain. [9] proposes to generate optical illusion images using phase interpolation based on DDIM inversion. While these approaches modify the diffusion sampling process to achieve the desired behavior, ϕ-PD introduces minimal changes to existing diffusion frameworks and preserves the original sampling dynamics.

Structure-Aligned Generation with Diffusion. Most existing methods achieve structure-aligned generation by modifying the network architecture and introducing additional adaptation components. ControlNet [42] copies the entire U-Net encoder into a trainable encoder branch, which adds significant computation overhead. T2I-Adapter [21] reduces computation overhead using a lightweight adapter module but sacrifices control precision. Uni-ControlNet [43] enables simultaneous utilization of multiple local controls by training two adapters. OmniControl [32] integrates image conditions into Diffusion Transformer (DiT) architectures with only 0.1% additional parameters by re-using

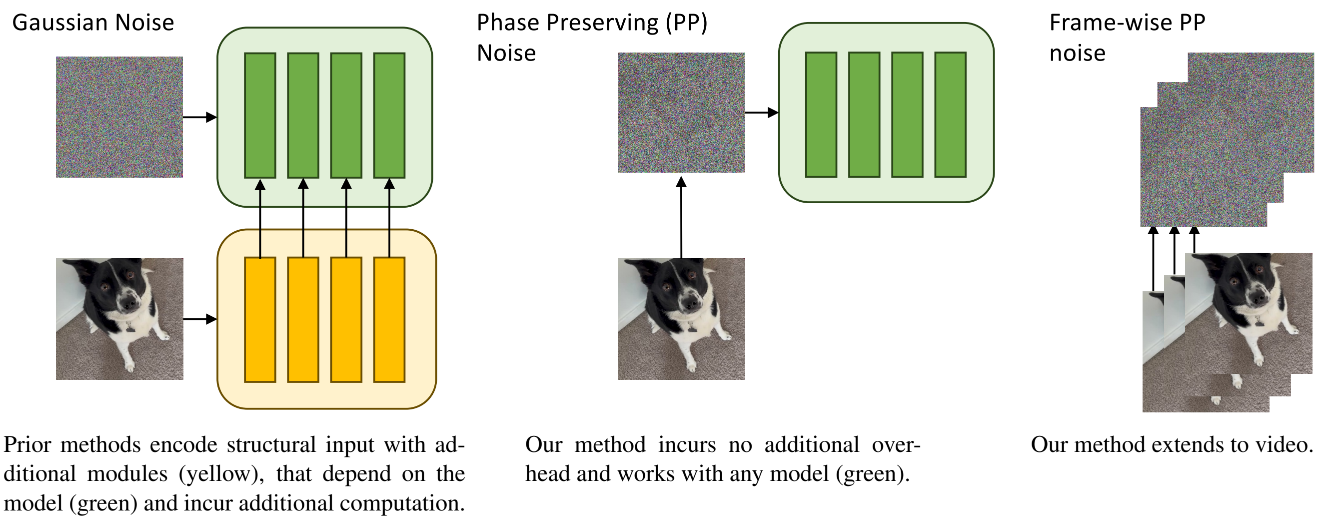


Figure 2. Unlike prior approaches that modify architectures and add overhead, ϕ-PD preserves structure via phase consistency, remaining lightweight and model-agnostic, reflecting that image-conditioned generation should be simpler, not harder.

the VAE and transformer blocks of the base model. ControlNeXt [24] uses a lightweight convolutional module to inject control signals, and directly finetune selective parameters of the base model to reduce training costs and latency increase. SCEdit [15] proposes an efficient finetuning framework that edits skip connections using a lightweight module. NanoControl [13] aims to achieve efficient control with a LoRA-style control module. The above methods all rely on an additional module to incorporate the control signal, though some are more lightweight than others. CosmosTransfer [22] achieves multi-modal control by combining multiple ControlNet branches, demonstrating promising applications on physical tasks; however, multiple branches introduce significant computation overhead. In contrast, ϕ-PD does not introduce any computation overhead or additional parameters while enabling universal spatial control.

Training-Free Guidance Methods. Recently, several training-free methods have been developed. [41] introduced FreeDoM, which leverages off-the-shelf pre-trained networks to construct time-independent energy functions that guide generation. [6] proposed ZestGuide for zero-shot spatial layout conditioning, utilizing implicit segmentation maps extracted from cross-attention layers to align generation with input masks. [20] presented FreeControl, a training-free approach that enforces structure guidance with the base model feature extracted from the control signal. Although these methods avoid training cost, they introduce additional overhead at test time, either an external model, DDIM inversion, or multiple inferences of the base model. In contrast, ϕ-PD can achieve training-free spatial control without any additional inference time overhead.

3. Method

3.1. Frequency Domain Fundamentals

In the frequency domain, any image $I(x, y)$ can be represented through the 2D Fourier transform:

$$\begin{aligned} F(u, v) &= \mathcal{F}\{I(x, y)\} \\ &= \sum_{x=0}^{W-1} \sum_{y=0}^{H-1} I(x, y) e^{-2\pi j(ux/W + vy/H)}, \end{aligned} \tag{1}$$

where $F(u, v)$ is a complex-valued function that can be decomposed into magnitude and phase components:

$$F(u, v) = |F(u, v)| \cdot e^{j\phi(u,v)} = A(u, v) \cdot e^{j\phi(u,v)}. \tag{2}$$

Here, $A(u, v) = |F(u, v)|$ is the magnitude spectrum and $\phi(u, v)$ the phase spectrum. The inverse Fourier transform uses magnitude and phase to reconstruct the original image:

$$\begin{aligned} I(x, y) &= \mathcal{F}^{-1}\{F(u, v)\} \\ &= \sum_{u=0}^{W-1} \sum_{v=0}^{H-1} F(u, v) e^{2\pi j(ux/W + vy/H)}. \end{aligned} \tag{3}$$

Phase-Magnitude Separation in Signal Processing. Foundational work by Oppenheim *et al.* [23] shows that phase primarily determines spatial structure, while magnitude largely controls texture statistics. Mixing phase and magnitude from different images produces reconstructions whose spatial layout follows the source of the phase, not magnitude (see Figure 3). This observation motivates our approach: if diffusion destroys phase, it destroys spatial geometry; if we preserve phase, we preserve structure.

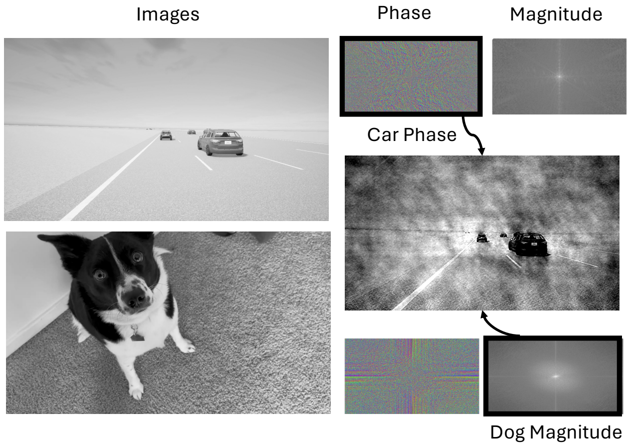


Figure 3. Mixing phase and magnitude from two images. The mixture keeps the structure of the image where the phase is taken.

3.2. Phase-Preserving Diffusion

Standard diffusion corrupts data using Gaussian noise whose Fourier coefficients have random magnitudes and random phases. As a result, even early diffusion steps erase spatial alignment. We propose a simple alternative: *preserve the input image's phase and randomize the magnitude*, by using structured noise that shares the input phase.

Structured Noise Construction. Given an input image I, we compute its Fourier transform:

$$F_I = A_I \cdot e^{j\phi_I}. \tag{4}$$

We construct **phase-preserving noise** by pairing the input image phase with a random magnitude:

$$F_{\hat{\epsilon}} = A_\epsilon \cdot e^{j\phi_I}, \tag{5}$$

and invert it:

$$\hat{\epsilon} = \mathcal{F}^{-1}\{F_{\hat{\epsilon}}\}, \tag{6}$$

where the random magnitude A_ϵ can be either from the Fourier transforms of Gaussian noise:

$$A_\epsilon = |\mathcal{F}\{\epsilon\}|, \quad \epsilon \sim \mathcal{N}(0,1), \tag{7}$$

or sampled from Rayleigh distribution directly [10]:

$$A_\epsilon = \sqrt{-2\ln U}, \qquad U \sim \text{Uniform}(0,1). \tag{8}$$

This structured noise is used in place of Gaussian noise in forward diffusion for training. It injects randomness while maintaining the phase of the input. At test time, we achieve structure-aligned generation by starting sampling from structured noise constructed with input image phase.

Frequency Selective Structured (FSS) Noise. In practice, we often want to control to what extent we keep the structure from the input image. Some tasks require strict structure preservation, while others benefit from partial freedom to reinterpret the scene. To provide this control, we introduce Frequency Selective Structured (FSS) noise, which only keep the image within a radius r and use the phase from the noise for the remainder. We define a smooth frequency mask $M(u,v)$ based on the cutoff radius r:

$$M(u,v) = \begin{cases} 1 & \text{if } \sqrt{u^2+v^2} \le r \\ \exp\left(-\frac{(\sqrt{u^2+v^2}-r)^2}{2\sigma^2}\right) & \text{if } \sqrt{u^2+v^2} > r \end{cases} \tag{9}$$

where σ controls the smoothness of the transition.

The FSS noise $\hat{\epsilon}$ is the combination of image phase and noise phase using the mask:

$$F_{\hat{\epsilon}} = A_\epsilon \cdot e^{j\phi_I \odot M + j\phi_\epsilon \odot (1-M)}, \tag{10}$$

where $\odot$ represents element-wise multiplication. We can sample phase ϕ_ϵ as the Fourier transform of Gaussian noise:

$$\begin{aligned} \epsilon &\sim \mathcal{N}(0,1), \\ \phi_\epsilon &= \arg(\mathcal{F}\{\epsilon\}) \sim \text{Uniform}(-\pi,\pi). \end{aligned} \tag{11}$$

When the mask is all zero, we take the phase from Gaussian noise for all frequencies, then this FFS noise becomes Gaussian noise and ϕ-PD becomes standard diffusion. Figure 4 visualizes FSS noise with different cutoff radius r. We can see that the noise becomes increasingly more structured with increasing r. Figure 5 shows images generated from the same input with different cutoff radius where the generated image aligns more tightly to the input with larger r.

3.3. Training Objective

ϕ-PD does not depend on model architecture or diffusion formulation. In the experiment section, we demonstrate integration both DDPM and Flow Matching, without modifying their architectures of loss functions.

The flow matching objective learns a vector field that transports the structured noise distribution to the target image distribution. During training, given a target image I and a structured noise $\hat{\epsilon}$, and a timestep $t \in [0,1]$, an intermediate image x_t is obtained using a linear combination between I and $\hat{\epsilon}$ following Rectified Flows [19]:

$$x_t = t\,\hat{\epsilon} + (1-t)\,I. \tag{12}$$

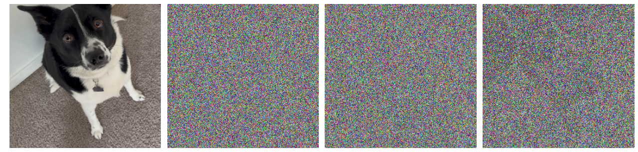


Figure 4. Frequency Selective Structured (FSS) Noise with increasing cutoff radius r.

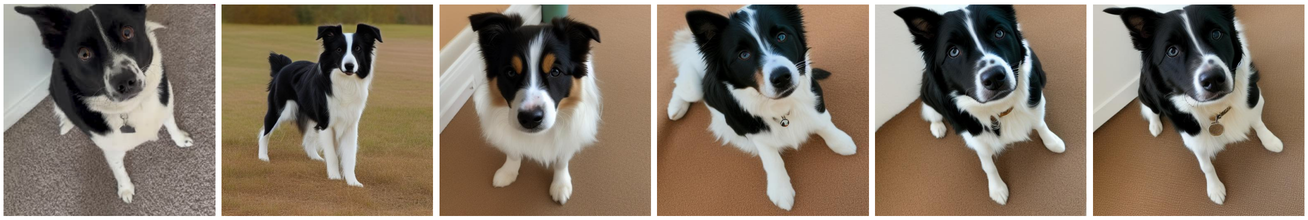


Figure 5. Image generated with the same noise and different cutoff radius r. Results are based on SD1.5.

The ground-truth velocity is

$$v_t = \frac{dx_t}{dt} = \hat{\epsilon} - I. \tag{13}$$

With this ground-truth, we can then train the model by minimizing the mean squared error between the model output and the ground-truth:

$$\mathcal{L} = \mathbb{E}_{I,\hat{\epsilon},t} \| u(x_t, t; \theta) - v_t \|_2^2. \tag{14}$$

In the Fourier domain, the velocity becomes

$$F_{v_t} = (A_{\hat{\epsilon}} - A_I) e^{j\phi_I} \tag{15}$$

which has the same phase ϕ_I as the image; therefore, the trained model always generates images with the same phase as the input image, remaining structurally aligned by construction.

In DDPMs [11], data x_0 is gradually corrupted by Gaussian noise:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(\sqrt{1-\beta_t}\, x_{t-1}, \beta_t I). \tag{16}$$

The model learns the reverse process by predicting the added noise ϵ at each step using the loss

$$\mathcal{L}_{\text{DDPM}} = \mathbb{E}_{x_0,\epsilon,t} \left[\| \epsilon - \epsilon_\theta(x_t, t) \|_2^2 \right], \tag{17}$$

where $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\epsilon$ and $\bar{\alpha}_t = \prod_{s=1}^{t}(1-\beta_s)$.

In our formulation, we replace the Gaussian noise ϵ with structured noise $\hat{\epsilon}$ that preserves the input phase:

$$x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \hat{\epsilon}. \tag{18}$$

The training objective remains identical,

$$\mathcal{L}_{\phi\text{-PD}} = \mathbb{E}_{x_0,\hat{\epsilon},t} \left[\| \hat{\epsilon} - \epsilon_\theta(x_t, t) \|_2^2 \right]. \tag{19}$$

The phase-preserving noise biases denoising toward structure-consistent trajectories, maintaining spatial alignment without altering the DDPM training objective.

3.4. Extension to Videos

ϕ-PD can be directly repurposed for video generation by constructing phase-preserving noise frame-by-frame. For a video $\{I_1, I_2, ..., I_T\}$, we construct structured noise for each frame, and concatenate them along the time dimension. We found that the best video generation results are when we first apply an image-based ϕ-PD on the first frame, and then use the first-frame-conditioned video ϕ-PD to generate the rest of the video. Similar to image generation, for video generation, our method requires no architectural changes either and simply supplies structured noise for each frame.

4. Experiments

We evaluate ϕ-PD across three settings: photorealistic re-rendering, stylized re-rendering, and simulation enhancement for autonomous driving, comparing against state-of-the-art methods. Our goals are to (1) assess spatial alignment under appearance changes, (2) measure visual realism, and (3) quantify the impact on downstream embodied-AI tasks.

To demonstrate its broad applicability, we implement ϕ-PD on three representative diffusion models: SD 1.5, FLUX-dev, and WAN 2.2 14B, which vary in size, formulation, and modality, covering both image and video generation.

4.1. Implementation Details

4.1.1 Datasets

UnrealCV[1] is a open-source tool that includes multiple assets. We created a diverse test set consisting of 5,000 images across all available assets, for a total of around 200 scenes. Figure 6 shows examples from this test set. This dataset covers a diverse range of scenes, including outdoor and indoor, city and natural etc, with geometry diversity while lacking photorealism. This dataset evaluates photorealistic enhancement and structure preservation.
ImageNetR is a test set proposed by [34], including 29 images of various objects and styles. While the original dataset

[1] https://github.com/unrealcv/unrealcv

Figure 6. Results on UnrealCV compared to FLUX-Kontext and QWenEdit.

provides prompts, these are generic image editing prompts. Since our work primarily focuses on re-rendering, we keep the editing prompts with style hints in the original dataset and added additional style prompts, resulting in a total of 8 prompts for each image. This benchmark assesses stylized re-rendering and structure preservation.

CARLA is an open-source driving simulator [8]. We collect 5.5 hours of driving videos from *CARLA Town 4* using the simulator's default autopilot. We then split the videos into 25 second clips and annotate a caption for each clip. For simulation enhancement, we use these captions combined with the style hint "A photorealistic video of driving". We evaluate the effectiveness of sim-to-real transfer by testing the CARLA-trained planner on Waymo's WOD-E2E [39] validation set.

4.1.2 Model architecture

We integrate ϕ-PD into:

- SD 1.5 (image DDPM)
- FLUX-dev (image flow matching)
- Wan2.2-14B (video flow matching)

We either fully finetune or LoRA-finetune each model using phase-preserving noise; no architectural changes are introduced. Notably, this finetuning is highly efficient: adapting the Wan2.2-14B video model with LoRA required only a single GPU while still yielding high-quality results, further demonstrating the lightweight nature of ϕ-PD. Please refer to the Appendix for additional implementation details and ablation studies.

4.1.3 Evaluation Metrics

For **photorealistic re-rendering**, we use our UnrealCV dataset and apply different methods to make images photorealistic. We use the style prompt "A high-quality picture" combined with the caption extracted from the images using ChatGPT.

Visual quality. We define an appearance score (AS) that measures how successful re-rendering is by looking at the ratio of CLIP similarity between positive and negative prompts:

$$AS = \frac{x^\top t_p}{x^\top t_n} \tag{20}$$

where x indicates the CLIP embedding of the re-rendered image; t_n the CLIP embedding of the negative prompt; and t_p the CLIP embedding of the positive prompt. For the positive prompt we use "Photo, camera captured, picture, photorealistic", and for the negative prompt we use "Game, render, cartoon, unreal".

Structural alignment. Besides visual quality, successful re-rendering also requires preserving the structure from the original image. To evaluate the structural alignment, we compute the error between the depth map of the original image and the generated image, using metrics SSIM (Struc-

Table 1. Quantitative evaluation results for photorealistic re-rendering on UnreadlCV.

Model	Input images	ControlNet-Tile	SDEdit	Ours
AS	0.9485	0.9733	0.9782	1.0008
SSIM	-	0.8781	0.8883	0.8982
ABSREL	-	0.5936	0.4938	0.4569

Table 2. Quantitative evaluation for stylized re-rendering.

Model	ControlNet-Tile	SDEdit	PNP	Ours
AS	1.3167	1.4243	1.4726	1.4709
SSIM	0.8831	0.7638	0.8498	0.8502
ABSREL	0.6684	1.0336	0.8194	0.7949

tural Similarity Index Measure) and ABSREL (Absolute Relative error).

For **stylized re-rendering**, we evaluate in a similar way as for photorealistic re-rendering. We compute the appearance score with the target style prompt as positive and the original style hint of the original image as negative. We also use the error between depth maps to evaluate structural alignment.

For **simulation enhancement**, we train an end2end planner for each compared method on the corresponding re-rendered image. We then test the planner's trajectories on Waymo's WOD-E2E validation set using distance-based metrics from the demonstrated driving, including Average Displacement Error (ADE) and Final Displacement Error (FDE).

4.2. Results

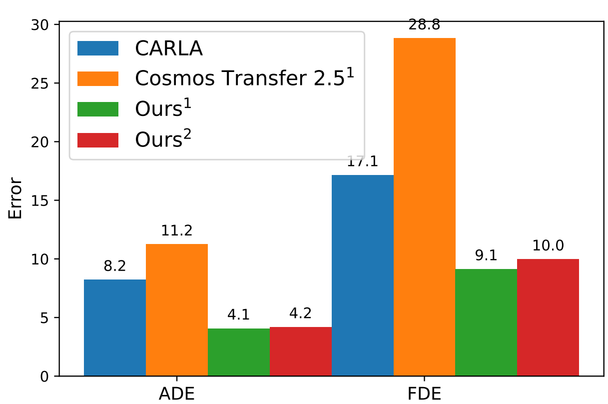


Figure 7. Planner error on Waymo validation set. Lower is better. [1]zero-shot, [2]finetuned on Waymo training set videos.

Photorealistic Re-Rendering. Quantitative results on UnrealCV re-rendering are summarized in Table 1, where all methods are implemented using the SD 1.5-based models for fair quantitative comparison. Qualitative examples are shown in Figure 6, which compares our Flux-based model against stronger recent models such as FLUX-Kontext, and Qwen-Edit. Across both settings, all methods improve photorealism-as reflected by higher AS scores than the input images while ϕ-PD achieves the highest photorealism and superior structure alignment. We observe that QWen-Edit produces visually high-quality results but often fails to maintain structural alignment with the input image; for example, in the first four cases, it enlarges the main subjects significantly. FLUX-Kontext aligns better with the input structure but provides only limited improvement in visual quality. Our method achieves both high visual fidelity and consistent structural alignment across frames.

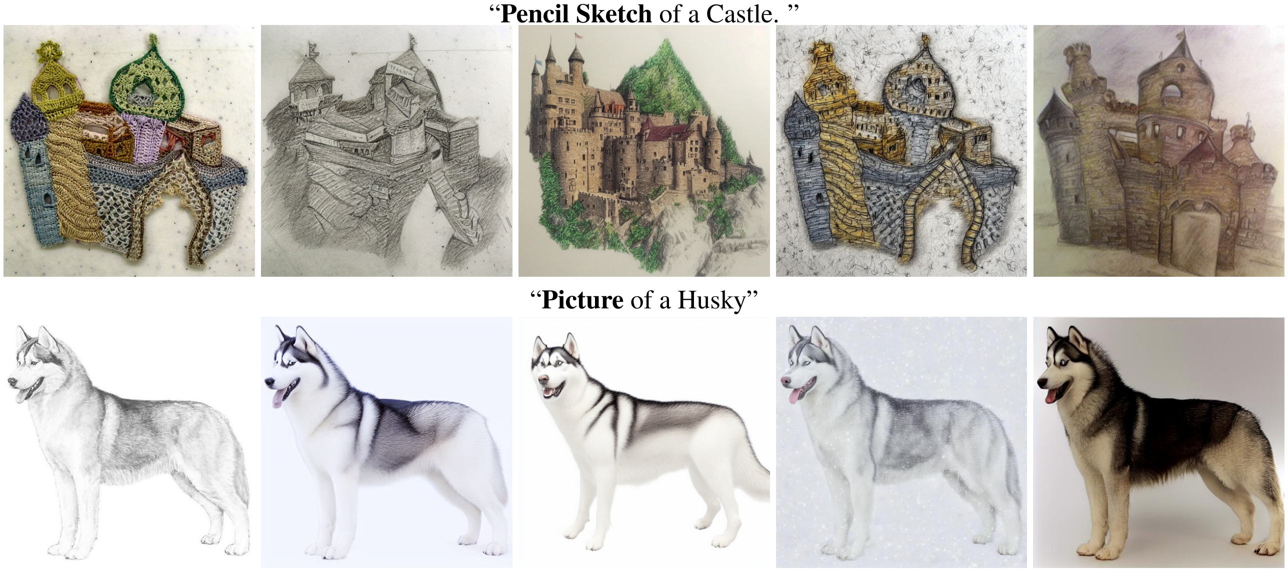
Stylized Re-rendering. Results of stylized re-rendering are presented in Table 2, with representative examples shown in Figure 8. All models are based on SD 1.5. This task evaluates the model's ability to alter appearance while preserving scene structure. As shown, ϕ-PD produces visually coherent stylizations that maintain object boundaries and spatial consistency, while prior methods often distort geometry or introduce texture misalignment. Quantitatively, ϕ-PD achieves a better trade-off between style strength and structural fidelity.

Simulation Enhancement. For this experiment, we generate 5.5 hours of demonstration driving videos from CARLA using its autopilot. Then we train an end2end planner on the re-rendered CARLA videos from each method using a ResNet backbone with a GRU to take temporal input and an MLP head to output a trajectory $\in \mathcal{R}^{16\times 2}$ (4s predictions at 4Hz in XY space). As baseline, we also present the results from a purely CARLA-trained model. Open-loop imitation driving results are given in Figure 7. ϕ-PD boosts planner generalization by 50% in zero-shot setting, demonstrating that structure-preserving appearance enhancement significantly reduces the sim-to-real gap. Video examples in Figure 9 show that ϕ-PD maintains road boundaries, vehicle shapes, and spatial layout consistently across frames, whereas the compared method produces distorted trees and multi-object artifacts.

5. Conclusion

We introduced Phase-Preserving Diffusion (ϕ-PD), a simple yet effective reformulation of the diffusion process that replaces Gaussian noise with structured noise that preserves image phase while randomizing the magnitude in the frequency domain. This simple change retains spatial alignment throughout sampling without modifying the architecture, altering training objectives, or introducing inference-time overhead. We also introduced Frequency-Selective

“**Pencil Sketch** of a Castle. ”



“**Picture** of a Husky”

Input PnP SDEdit ControlNet Ours

Figure 8. Results on stylized rerendering.

Figure 9. Video re-rendering results for simulation enhancement. From top to bottom are: input, Cosmos-Transfer2.5 [22], ours.

Structured (FSS) noise, which provides continuous control over structural alignment rigidity through a single frequency cutoff parameter, making it broadly applicable to different applications. Across photorealistic re-rendering, stylized re-rendering, and simulation enhancement for driving, ϕ-PD demonstrates strong spatial fidelity and visual realism. When applied to CARLA re-rendering, ϕ-PD significantly improves zero-shot planner transfer to the Waymo dataset, narrowing the sim-to-real gap.

Limitation. ϕ-PD assumes image-like inputs; modalities such as depth or normals may require a lightweight prior to produce an initial image representation.

Future work. ϕ-PD is orthogonal to existing conditioning or adapter methods and can be integrated with them for enhanced control. Future work includes extending ϕ-PD to tasks such as deblurring, relighting, super-resolution, and general image restoration.

References

[1] Anurag Ajay, Yilun Du, Abhi Gupta, Joshua Tenenbaum, Tommi Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? *arXiv preprint arXiv:2211.15657*, 2022. 2

[2] Seungyeon Baek, Erqun Dong, Shadan Namazifard, Mark J Matthews, and Kwang Moo Yi. Sonic: Spectral optimization of noise for inpainting with consistency. *arXiv preprint arXiv:2511.19985*, 2025. 2

[3] Yogesh Balaji, Seungjun Nah, Xun Huang, Arash Vahdat, Jiaming Song, Qinsheng Zhang, Karsten Kreis, Miika Aittala, Timo Aila, Samuli Laine, et al. eDiff-I: Text-to-image diffusion models with an ensemble of expert denoisers. *arXiv preprint arXiv:2211.01324*, 2022. 2

[4] Haoxin Chen, Menghan Xia, Yingqing He, Yong Zhang, Xiaodong Cun, Shaoshu Yang, Jinbo Xing, Yaofang Liu, Qifeng Chen, Xintao Wang, et al. VideoCrafter1: Open diffusion models for high-quality video generation. *arXiv preprint arXiv:2310.19512*, 2023. 2

[5] Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. *arXiv preprint arXiv:2303.04137*, 2023. 2

[6] Guillaume Couairon, Marlène Careil, Matthieu Cord, Stéphane Lathuilière, and Jakob Verbeek. Zero-shot spatial layout conditioning for text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2174–2183, October 2023. 3

[7] Sander Dieleman. Diffusion is spectral autoregression. urlhttps://sander.ai/2024/09/02/spectral-autoregression.html, September 2024. Accessed: 7 Dec 2025. 2

[8] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator, 2017. 6

[9] Xiang Gao, Shuai Yang, and Jiaying Liu. PTDiffusion: Free lunch for generating optical illusion hidden pictures with phase-transferred diffusion model. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 18240–18249, 2025. 2

[10] Joseph W. Goodman. *Statistical Optics*. Wiley, 2 edition, 2015. Section 2.9.3. 4

[11] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2, 5

[12] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *Advances in Neural Information Processing Systems*, 35:8633–8646, 2022. 2

[13] Huang et al. NanoControl: A lightweight framework for precise and efficient control in diffusion transformer. *arXiv preprint arXiv:2508.10424*, 2024. 3

[14] Michael Janner, Yilun Du, Joshua B Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. *arXiv preprint arXiv:2205.09991*, 2022. 2

[15] Zeyinzi Jiang, Chaojie Mao, Yulin Pan, Zhen Han, and Jingfeng Zhang. SCEdit: Efficient and controllable image diffusion generation via skip connection editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8995–9004, June 2024. 3

[16] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2

[17] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. DiffWave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020. 2

[18] Black Forest Labs. Flux. `https://github.com/black-forest-labs/flux`, 2024. 11

[19] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *arXiv preprint arXiv:2209.03003*, 2022. 4

[20] Sicheng Mo, Fangzhou Mu, Kuan Heng Lin, Yanli Liu, Bochen Guan, Yin Li, and Bolei Zhou. FreeControl: Training-free spatial control of any text-to-image diffusion model with any condition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, June 2024. 3

[21] Chong Mou, Xintao Wang, Liangbin Xie, Yanze Wu, Jian Zhang, Zhongang Qi, and Ying Shan. T2I-Adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. In *AAAI*, volume 38, pages 4296–4304, 2024. 2

[22] NVIDIA. Cosmos-Transfer1: Conditional world generation with adaptive multimodal control. *ArXiv*, abs/2503.14492, 2025. 3, 8

[23] Alan V. Oppenheim and Jae S. Lim. The importance of phase in signals. *Proceedings of the IEEE*, 69(5):529–541, 1981. 2, 3

[24] Bohao Peng, Jian Wang, Yuechen Zhang, Wenbo Li, Ming-Chang Yang, and Jiaya Jia. ControlNeXt: Powerful and efficient control for image and video generation. *arXiv preprint arXiv:2408.06070*, 2024. 3

[25] Vadim Popov, Ivan Vovk, Vladimir Gogoryan, Tasnima Sadekova, and Mikhail Kudinov. Grad-TTS: A diffusion probabilistic model for text-to-speech. In *International Conference on Machine Learning*, pages 8599–8608. PMLR, 2021. 2

[26] Michael Psenka, Alejandro Escontrela, Pieter Abbeel, and Yi Ma. Learning a diffusion model policy from rewards via q-score matching. *arXiv preprint arXiv:2312.11752*, 2023. 2

[27] Yurui Qian, Qi Cai, Yingwei Pan, Yehao Li, Ting Yao, Qibin Sun, and Tao Mei. Boosting diffusion models with moving

average sampling in frequency domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8911–8920, 2024. 2

[28] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[29] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, June 2022. 11

[30] Daniel L. Ruderman and William Bialek. The statistics of natural images. *Network: Computation in Neural Systems*, 5(4):517–548, 1994. 2

[31] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, S Sara Mahdavi, Rapha Gontijo Lopes, et al. Photorealistic text-to-image diffusion models with deep language understanding. *arXiv preprint arXiv:2205.11487*, 2022. 2

[32] Zhenxiong Tan et al. OminiControl2: Efficient conditioning for diffusion transformers. *arXiv preprint arXiv:2503.08280*, 2025. 2

[33] Alibaba Team Wan. Wan: Open and advanced large-scale video generative models. *arXiv preprint arXiv:2503.20314*, 2025. 11

[34] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1921–1930, June 2023. 5

[35] Julen Urain, Niklas Funk, Jan Peters, and Georgia Chalvatzaki. SE(3)-DiffusionFields: Learning smooth cost functions for joint grasp and motion optimization through diffusion. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5923–5930. IEEE, 2023. 2

[36] Zhendong Wang, Jonathan J Hunt, and Mingyuan Zhou. Diffusion policies as an expressive policy class for offline reinforcement learning. *arXiv preprint arXiv:2208.06193*, 2022. 2

[37] Zhou Wang and Eero P. Simoncelli. Translation insensitive image similarity in complex wavelet domain. *IEEE Transactions on Image Processing*, 14(4):466–479, 2005. 2

[38] Wei Wu, Qingnan Fan, Shuai Qin, Hong Gu, Ruoyu Zhao, and Antoni B Chan. Freediff: Progressive frequency truncation for image editing with diffusion models. In *European Conference on Computer Vision*, pages 194–209. Springer, 2024. 2

[39] Runsheng Xu, Hubert Lin, Wonseok Jeon, Hao Feng, Yuliang Zou, Liting Sun, John Gorman, Ekaterina Tolstaya, Sarah Tang, Brandyn White, Ben Sapp, Mingxing Tan, Jyh-Jing Hwang, and Dragomir Anguelov. WOD-E2E: Waymo open dataset for end-to-end driving in challenging long-tail scenarios, 2025. 6

[40] Cuihong Yu, Cheng Han, Chao Zhang, Yuewei Wang, Qihang Hu, Yin Yan, Moran Zhan, Meng Li, and Guangjin Bi. DMFFT: improving the generation quality of diffusion models using fast fourier transform. *Scientific Reports*, 15, March 2025. 2

[41] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. FreeDoM: Training-free energy-guided conditional diffusion model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23174–23184, October 2023. 3

[42] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, pages 3836–3847, 2023. 2

[43] Shihao Zhao, Dongdong Chen, Yen-Chun Chen, Jianmin Bao, Shaozhe Hao, Lu Yuan, and Kwan-Yee K. Wong. Uni-ControlNet: All-in-one control to text-to-image diffusion models. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023. 2

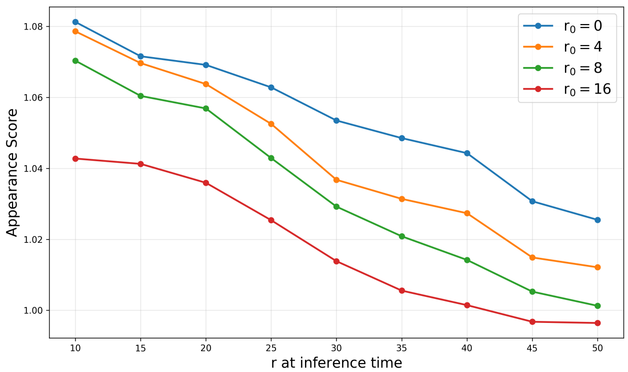


Figure 10. Visual realism measured by Appearance Score.

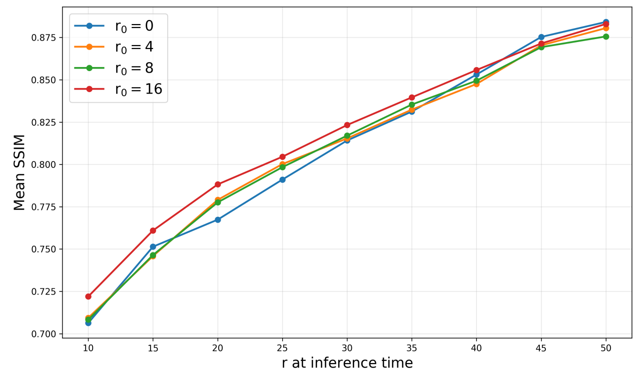


Figure 11. Structural alignment measured by depth SSIM.

A. Additional Implementation Details

We implemented ϕ-PD based on three models: SD 1.5 [29], FLUX-dev [18] and Wan2.2-14b [33]. We use the implementation of these models from *DiffSynthStudio*[2] This section describes the additional implementation details for the experiment with each model.

A.1. Training and Inference Details

We start from the officially released checkpoints of SD 1.5 [29], FLUX-dev [18], and Wan2.2-14B [33], and finetune each model with phase-preserving noise. For SD 1.5, we experiment with both full finetuning and LoRA finetuning, while for FLUX-dev and Wan2.2-14B we use LoRA finetuning due to computational constraints. At inference time, for Wan2.2-14B we adopt the 4-step LoRA from LightX2V[3] and apply it directly on top of our finetuned LoRA weights to accelerate sampling.

We LoRA finetune Wan2.2-14B for 1,200 iterations and FLUX-dev for 10,000 iterations, while SD 1.5 is fully finetuned for 140,000 iterations. Each training run takes approximately 48 hours on an NVIDIA A100 GPU.

For each training iteration, we sample a cutoff radius r from an exponential distribution and add a constant offset r_0 to ensure a minimum amount of phase information is always preserved:

$$r = r_0 + r', \quad r' \sim \mathrm{Exp}(\lambda), \tag{21}$$

where $\lambda > 0$ is the rate parameter of the exponential distribution and $r_0 > 0$ controls the minimum cutoff. In our experiments, we set $\lambda = 0.1$ empirically. We set the transition bandwidth parameter $\sigma = 2$, which controls the smoothness of the frequency mask $M(u, v)$ around the cutoff radius r.

B. Ablation Studies

We ablate the choices of r_0, the minimal cutoff radius at training time, and the inference-time cutoff radius r in this section. All ablation experiments are conducted using SD 1.5 with LoRA finetuning and evaluated on $1,000$ randomly selected samples from the UnrealCV test set. Figure 11 shows the Appearance Score (Sec. 4.1.3), which measures the photorealism of re-rendered images. Figure 11 shows the depth consistency measured by SSIM, reflecting how well the re-rendered image aligns structurally with the original input.

From both figures, we observe that increasing the cutoff radius r at inference time improves structural alignment at the cost of reduced photorealism. The minimal cutoff threshold r_0 during training also affects performance across different inference-time radii r. A higher r_0 during training leads to better performance with higher r during inference, making it suitable for re-rendering images of decent quality where the model primarily refines details. Conversely, a lower r_0 favors scenarios with smaller inference-time r, performing better on low-quality inputs that require larger visual changes to achieve photorealism. In our experiments, we set $r_0 = 4$ for a balanced trade-off between structure preservation and photorealism.

[2] https://github.com/modelscope/DiffSynth-Studio
[3] https://huggingface.co/lightx2v/Wan2.2-Lightning