

Perturb-and-Revise: Flexible 3D Editing with Generative Trajectories

Susung Hong¹ Johanna Karras¹ Ricardo Martin-Brualla² Ira Kemelmacher-Shlizerman¹

¹University of Washington ²Google Research



Figure 1. **Perturb-and-Revise** takes a source NeRF and an edit prompt as input and produces the edited result through: (1) versatile initialization via parameter perturbation, (2) multi-view consistent score distillation, and (3) refinement with the identity-preserving gradient.

Abstract

The fields of 3D reconstruction and text-based 3D editing have advanced significantly with the evolution of text-based diffusion models. While existing 3D editing methods excel at modifying color, texture, and style, they struggle with extensive geometric or appearance changes, thus limiting their applications. We propose **Perturb-and-Revise**, which makes possible a variety of NeRF editing. First, we **perturb** the NeRF parameters with random initializations to create a versatile initialization. We automatically determine the perturbation magnitude through analysis of the local loss landscape. Then, we **revise** the edited NeRF via generative trajectories. Combined with the generative process, we impose identity-preserving gradients to refine the edited NeRF. Extensive experiments demonstrate that **Perturb-and-Revise** facilitates flexible, effective, and consistent editing of color, appearance, and geometry in 3D. For 360° results, please visit our project page: <https://susunghong.github.io/Perturb-and-Revise>.

1. Introduction

Neural Radiance Fields (NeRFs) [36] have revolutionized the creation of high-quality 3D scenes, marking a significant advancement in 3D reconstruction technology. Beyond

their initial applications, NeRFs have enabled flexible generation of 3D content from models trained solely on image-text pairs [18, 42] through score distillation [25, 42, 60].

However, editing 3D content—an essential aspect of refinement and customization—remains a time-consuming and labor-intensive process across various industries, including animation, manufacturing, design, and gaming. This challenge is particularly pronounced with NeRFs, where color and density attributes are intricately encoded within their parameters. Consequently, there is an ongoing need for more intuitive and universally accessible tools for editing NeRFs, leveraging user-friendly interfaces such as text prompts for broader applicability and faster production.

Fortunately, recent innovations in text-based 3D editing techniques have emerged, utilizing state-of-the-art diffusion models to transform and shape 3D scenes using natural language prompts. For instance, leveraging diffusion models [2, 45], recent works such as Instruct-NeRF2NeRF [8] and Posterior Distillation [25] propose methods that use text prompts to modify 3D scenes. While these methods excel in altering the color, texture, or style of an object, their limitations are distinct: they struggle with edits involving significant geometric or appearance changes and reliable, consistent updates, which considerably limit their applications.

To overcome these limitations, we propose a 3D object editing framework called **Perturb-and-Revise (PnR)**,

which leverages existing 3D generation methods for editing in a flexible and natural way. Fig. 2 shows the overall process. Drawing on the notion that NeRF parameters optimized with score distillation can be considered as a particle [62] or data point [6], we propose a novel method that leverages perturbation at the parameter level to perform text-based edits requiring various changes. Specifically, we construct versatile NeRF initialization by interpolating the source NeRF and a random NeRF initialization. Intuitively, perturbation in the parameter space helps the particle escape local minima and facilitates its following of the natural trajectory of the generative ODE towards the distribution of the desired edit, allowing for a wide range of challenging edits including changing the pose and introducing new objects.

Additionally, to determine the amount of perturbation needed to escape the basin of attraction without costly searching algorithms, we propose an algorithm based on the loss landscape near the source parameters, by simulating a few score distillation steps ahead of the parameter perturbation and optimization. After the early step of our framework, to achieve more similarity to the original object and refine the quality of the result, we employ the Identity-Preserving Gradient (IPG) to increase the fidelity of the edited NeRF to the source NeRF. As a result, the output closely resembles the source while still maintaining fidelity to the intended edits.

We extensively evaluate our method on 3D fashion objects, as well as general objects in Objaverse [4], on a variety of appearance- and geometry-based edits. Furthermore, we demonstrate that our method achieves state-of-the-art results on various 3D editing baselines.

2. Related Works

Diffusion models. Diffusion models [9, 51], which are closely associated with score-based models [21, 52, 53], have recently shown remarkable sample quality in image synthesis. Latent Diffusion Models (LDM) [45], which perform the diffusion forward and backward process in the latent space [7], have demonstrated their efficacy and generation quality. This trend has been scaled up by training diffusion models on large-scale text-image datasets [47] for text-conditional generation. The prosperity of text-to-image diffusion models has turned attention to leveraging their knowledge for complex content creation, such as 3D scenes and videos [1, 10, 12, 14, 15, 23, 27, 35, 42, 48, 60, 62, 64]. Recently, many adapting methods of text-to-image diffusion models have been proposed to achieve either generation with additional conditions [2, 11, 13, 20, 37, 66, 68] or to maintain consistency needed for videos [14, 23, 43, 64] or 3D scenes [29, 49].

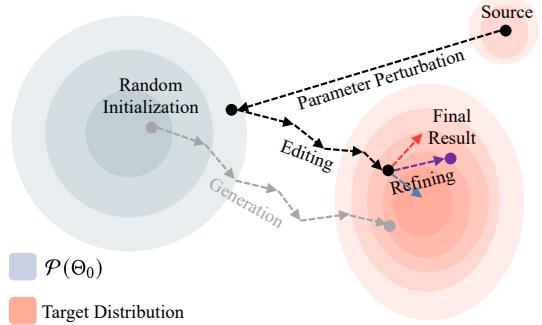


Figure 2. Conceptual figure. The target distribution in the figures represents the conditional distribution of NeRF parameters relative to the edit prompt. First, parameter perturbation enables the parameters to escape from local minima and follow a natural generative path. Subsequently, during the refining process, the tug-of-war between two vectors, $\lambda_d \nabla_{\theta} d(\theta_\tau, \theta_{src})$ (the red arrow) and $d\theta_\tau$ (the blue arrow), pushes the actual parameters into a region that is closer to either the source parameters or the high-density region specified by the edit prompt.

Score distillation. Score distillation [42, 60, 62], a recent trend in synthesizing underlying parameterizations, mostly in 3D scene generation, uses only score-based models [45, 46] operating on 2D data. This approach has revolutionized text-to-3D generation by introducing an elegant method that does not require intense training on a 3D dataset. It enables the rich and flexible knowledge of 2D text-to-image models to be transferred to 3D generation [12, 27, 35, 60, 62]. However, due to bias in 2D models, a multi-view consistency problem has arisen, necessitating further analysis and generalization of the original method [12, 29, 49]. Another limitation of score distillation is its time-consuming optimization, with the controllability and editing capability remaining underexplored.

Text-based NeRF editing. In recent years, neural radiance fields (NeRF) [36] have emerged as a groundbreaking approach for generating photorealistic novel views of a scene captured in photographs, and have been extended in many follow-up works [22, 33, 38, 41]. In response to these advancements, several techniques for modifying NeRF have been proposed. Traditional methodologies for editing NeRF include the alteration of materials and lighting [39, 54, 57], as well as the spatial manipulation of objects via bounding box frameworks [40, 65]. Additionally, there has been exploration in the stylization of NeRFs, including Edit-NeRF [30], Clip-NeRF [58], and NeRF-Art [59], as well as in distilling 2D features into radiance fields, such as in Distilled Feature Fields [24] and Neural Feature Fusion Fields [56], which enables nuanced, guided edits based on language or images. Crucially, Instruct-NeRF2NeRF [8] focuses on user-friendly, language-based editing commands, using an instruction-driven, 2D image-conditioned diffusion model [2] for more intuitive and context-aware 3D edit-

ing. However, even in a trend using 2D language-based models such as CLIP [44] or text-to-image diffusion models [2, 45, 46], these methodologies are unable to control large geometrical changes, such as changing the pose or introducing a massive object. Posterior Distillation [25], instead of naively matching the noise, proposes a way to use a loss that matches the stochastic latents [16, 63] of source and target images. Still, this method falls into issues such as slower convergence and large computational costs.

3. Background

Diffusion models are a type of generative model that iteratively restores an image from Gaussian noise. The training objective of diffusion models involves training a neural network to estimate the score for a noised image y given the noise level σ via denoising score matching [17]. Following the preconditioning convention [21], a diffusion model, parameterized by ϕ , can be interpreted as a denoiser $D_\phi(y; \sigma)$, which minimizes the weighted L2 loss across different σ values in data samples.

In the context of text-based 3D scene generation, Score Distillation Sampling (SDS) [42] updates the 3D scene representation given a text caption c using the following rule:

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\text{SDS}}(\phi, c, x = g(\theta, \psi)) &= \\ -\mathbb{E}_{\sigma \sim \Sigma, n \sim \mathcal{N}(0, \sigma^2 I)} \left[\omega(\sigma) (D_\phi(x + n; \sigma, c) - x) \frac{\partial x}{\partial \theta} \right]. \end{aligned} \quad (1)$$

In this equation, $g(\cdot)$ is the differentiable renderer, ψ is the random camera pose, Σ is a predefined distribution from which the noise level for the denoiser is sampled, and $\omega(\cdot)$ denotes the weighting function.

SDS can be viewed as a form of particle-based variational inference [3, 5, 28, 32, 61, 62]. In this context, the update rule can be expressed as a generative ODE over an optimization step τ , derived via Wasserstein gradient flow [3]:

$$\frac{d\theta_\tau}{d\tau} = -\mathbb{E}_{\sigma \sim \Sigma, n \sim \mathcal{N}(0, \sigma^2 I)} \left[\omega(\sigma) (D_\phi(x + n; \sigma, c) - x) \frac{\partial x}{\partial \theta_\tau} \right]. \quad (2)$$

For further details, please see the supplementary material.

As background, multi-view diffusion models [49], denoted as $\{D_\phi^M(g(\theta, \psi_i) + n; \sigma, c, \psi_i)\}_{i=1}^N$, consistently generate multi-view images simultaneously. Here, N denotes the total number of multi-views generated simultaneously, and ψ_i denotes the i -th viewpoint. Note that for $N = 1$, these are ordinary diffusion models. For $N > 1$, they are implemented by adapting the attention module to attend to different frames [49] and by training on 3D assets [4].

4. Method

In text-based editing scenarios, we assume that a user provides a text prompt, c_{edit} , and a source 3D object represented

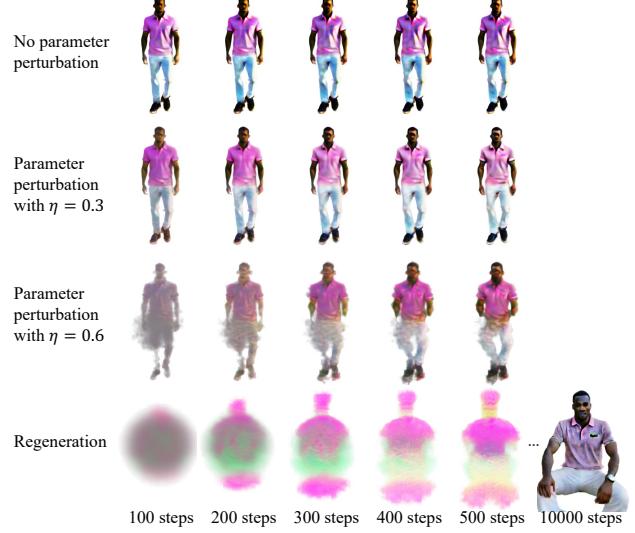


Figure 3. Effect of parameter perturbation. In this example, we aim to make a NeRF model of a standing person sit down using the word “sitting.” The scene converges quickly even with large perturbations ($\eta = 0.6$), while complete regeneration yields blurry rendering results given the same number of optimization steps.

by the parameterization θ_{src} . The aim is to modify the 3D object into a new form $\theta^* = \mathcal{E}(\theta_{\text{src}}, c_{\text{edit}})$, ensuring that the transformation aligns with the text prompt and retains a resemblance to the original 3D object.

At a high level, we leverage the generative ODE of 3D objects through novel parameter perturbation while preserving proximity to the source object. In Sec. 4.1, we introduce perturbation in parameter space that enables the parameters to escape local minima and converge to natural generative trajectories. In Sec. 4.2, we propose a novel algorithm to determine the degree of perturbation by analyzing the local loss landscape. Finally, in Sec. 4.3, we propose balancing fidelity between the text prompt and the source object to place the final output in the desired region.

4.1. Parameter Perturbation for Flexible Object Editing

Given the generative ODE in Eq. 2, which minimizes the energy functional between distributions, a fully-optimized NeRF of a 3D object is considered to exhibit relatively low energy. This state corresponds to the particle residing in a local minimum, which remains stable despite alterations in the text prompts, as it is still capable of producing realistic renderings. As exemplified in the first row of Fig. 3, naively using θ_{src} as initialization and making independent predictions on 2D noisy images with diffusion models are insufficient to handle the consistent changes in overall appearance and geometry that occur in a 3D object.

Interpreting neural fields as a particle [62] or a data point [6], we propose parameter-level perturbation for an

editing task, which facilitates changes in θ_{src} by enabling the editing particle ODE to hijack the natural, coarse-to-fine generative process in Eq. 2 that the parameters are likely to follow. In other words, since adding more noise to the parameters is equivalent to undoing more of the optimization process, this approach enables the parameters to be flexible, allowing for significant changes.

Specifically, let $\mathcal{P}(\Theta_0)$ denote the probability density function over Θ_0 , which is a set of random parameterizations. Further, assume that we have a fully optimized 3D scene, θ_{src} . Then, we obtain the perturbed version of θ_{src} by using linear interpolation:

$$\theta_{\text{perturbed}} = \text{Lerp}(\theta_{\text{src}}, \theta_0, \eta) \quad \text{where } \theta_0 \sim \mathcal{P}(\Theta_0). \quad (3)$$

where $\eta \in [0, 1]$ denotes the perturbation amount and $\text{Lerp}(\theta_{\text{src}}, \theta_0, \eta) = (1 - \eta) \cdot \theta_{\text{src}} + \eta \cdot \theta_0$. Note that $\eta = 1$ indicates $\theta_{\text{perturbed}}$ is the random parameters. Theoretically, the interpolation makes the distribution of perturbed particles similar to the initializing distribution of NeRF, making it versatile. See the supplementary material for details.

Subsequently, we run the parameter ODE for textual editing, starting from $\theta_{\text{perturbed}}$, using a new prompt c_{edit} . However, we observe that adopting ordinary single-view diffusion models (when $N = 1$) poses ambiguity when an edit introduces asymmetry or adds a new object. Therefore, we execute multi-view consistent updates as follows:

$$\frac{d\theta_\tau}{d\tau} = -\frac{1}{N} \sum_{i=1}^N \mathbb{E}_{\sigma, n} \left[\left(D_\phi^M(z_i + n; \sigma, c_{\text{edit}}, \psi_i) - z_i \right) \frac{\partial z_i}{\partial \theta_\tau} \right], \quad (4)$$

where we define $z_i := g(\theta_\tau, \psi_i)$ and omit the weighting factor for brevity.

The conceptual illustration of parameter perturbation can be found in Fig. 2. This approach essentially reverts the optimization process by interpolating between the initial state and the fully optimized state. Consequently, the particles are relocated to a less-optimized state characterized by a more moderate energy landscape. This repositioning allows the particle ODE (Eq. 4) to intervene in the generative process with a new text prompt c_{edit} , guiding the parameter trajectory. As shown in Fig. 3, increasing the noise level in the parameters inversely correlates with the degree of reversion of the optimization, allowing for more radical modifications in the model outputs.

4.2. Determining η by Analyzing Loss Landscape

In Sec. 4.1, we see that η controls parameter versatility by determining how similar the distribution of parameters is to the distribution of random NeRF initializations. The required degree of versatility depends on both the type of edits and the current source NeRF parameters. However, finding an optimal η through standard mechanisms, such as grid or

random search, is computationally expensive. Therefore, we propose a novel algorithm that explores the basin of attraction surrounding local minima to determine η by analyzing the loss landscape.

Specifically, we leverage the loss function as a proxy to measure the depth and volume of the basin of attraction. Interestingly, for some examples, we observe that the loss function even increases during optimization steps, supporting our claim in Sec. 4.1 that the parameters are in a low-energy state. To address this, we simulate several optimization steps with c_{edit} and calculate the total loss decrease prior to parameter perturbation. For algorithmic stability, we compute the difference between the averages of the first few steps and the last few steps. We then determine $\eta(\theta_{\text{src}}, c_{\text{edit}})$ based on the source parameters θ_{src} and the text prompt c_{edit} using an inverted exponential decay function. Subsequently, we apply parameter perturbation in Eq. 3 with $\eta = \eta(\theta_{\text{src}}, c_{\text{edit}})$. The complete algorithms for our parameter perturbation and η selection are provided in the supplementary material.

4.3. Identity-Preserving Gradient (IPG)

While our parameter perturbation in Sec. 4.1 successfully enables flexible changes, there are still some estimation errors or biases arising from the diffusion model during this phase. On the other hand, imposing constraints that make it similar to the source object (e.g., L2 distance) conflicts with the generative ODE, making the optimization challenging. To counteract the estimation error and to circumvent the conflict, we introduce the Identity-Preserving Gradient (IPG) term that is added at later steps to the editing gradient presented in Sec. 4.1.

Initially, we assume that the ideal parameters are closer to the source parameters and the high-density region of the target distribution. Then, as conceptually shown in Fig. 2, we instigate a tug-of-war between two vectors: one symbolizes the velocity consistently pointing towards the high-density region as dictated by the edit prompt, and the other symbolizes the velocity towards θ_{src} . Specifically, the velocity in Eq. 4 propels the particle towards the region with a high likelihood given the text c_{edit} , while the other gradient represents the pull towards the original NeRF parameterization θ_{src} . This effectively corrects the shift caused by the error in the editing process, ensuring it resembles the original while maintaining the intended edits. To this end, we extend the optimization process and compute IPGs during these additional steps.

Formally, we combine the result of Eq. 4 with the IPG to define a refinement step as follows:

$$d\theta_\tau^{\text{refine}} = d\theta_\tau + \lambda_d \nabla_\theta d(\theta_\tau, \theta_{\text{src}}), \quad (5)$$

where $d(\cdot, \cdot)$ is a similarity metric. In practice, we observe that the combination of L_1 and perceptual loss [19],



Figure 4. Baseline comparisons for a wide range of fashion object editing, including color, pattern, shape, pose, and object edits. We compare our method with Score Distillation Sampling (SDS) [42], Posterior Distillation Sampling (PDS) [25], and Instruct-NeRF2NeRF [8]. For SDS and PDS, we use MVDream [49] as the backbone for fair comparison. SDS significantly alters the appearance and texture of the source objects but is unable to handle edits that require extensive geometric changes (3rd, 4th, and 5th rows). PDS is not capable of making significant edits and cannot deviate far from local minima. While Instruct-NeRF2NeRF changes the texture of objects as desired, it cannot address geometric changes. In contrast, our method is capable of various types of edits, including those involving large geometric changes.

which is also the preferred choice in NeRF [36] training, is more robust and less susceptible to noise than using L_2 loss. Thus, given a random camera pose ψ , we use a combination of L1 distance and perceptual similarity between the rendered images $g(\theta_\tau, \psi)$ and $g(\theta_{\text{src}}, \psi)$, weighted by λ_{L1} and λ_p , respectively.

4.4. Timestep Annealing

In text-based 3D generation, some recent works have revealed that using timestep annealing, which adjusts the noise level added to the 2D rendered images according to the global optimization step, effectively boosts the quality of the results [49, 62].

In our text-based 3D editing case, while we use a smaller noise level at the start, we also identify the annealing method as crucial for maintaining the original quality of the

3D object while significantly reducing blurry textures. To this end, we regard Σ as a function that depends on the editing step τ , $\Sigma(\tau)$, from which the noise level σ is sampled.

Specifically, for $\tau \leq T$, where T is the final step of the schedule, $\Sigma(\tau)$ is gradually annealed from $\mathcal{U}(\sigma_{\min}^0, \sigma_{\max}^0)$ to $\mathcal{U}(\sigma_{\min}^T, \sigma_{\max}^T)$. This approach initially facilitates low-frequency edits, followed by fine-grained ones. We provide detailed information in the supplementary material.

5. Experiments

As baselines, we adopt two distillation-based methods: Score Distillation Sampling (SDS) [42] and Posterior Distillation Sampling (PDS) [25]. We use the exact weighting factors for the losses as originally proposed in [25, 42]. For fair comparison, we use MVDream [49] as their backbones. Additionally, we use Instruct-NeRF2NeRF [8] as our base-

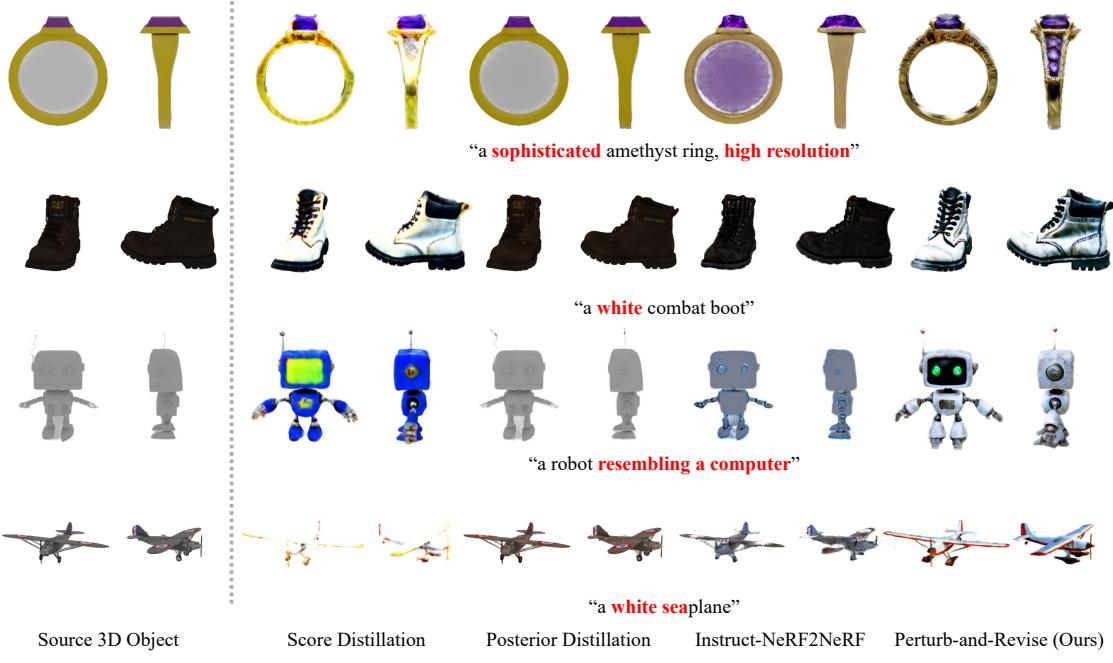


Figure 5. Baseline comparisons of editing various general 3D objects from the Objaverse dataset [4].

Metric	Score Distillation [42]	Posterior Distillation [25]	Instruct-NeRF2NeRF [8]	Perturb-and-Revise (Ours)
CLIP-Dir-Sim _{ViT-B/32} ↑	0.0480	0.0287	<u>0.0583</u>	0.0594
CLIP-Dir-Sim _{ViT-B/16} ↑	0.0418	0.0304	0.0549	<u>0.0534</u>
CLIP-Dir-Sim _{ViT-L/14} ↑	0.0415	0.0264	<u>0.0539</u>	0.0567
CLIP-Dir-Sim _{averaged} ↑	0.0438	0.0285	<u>0.0557</u>	0.0565
LPIPS _{VGG} ↓	0.1273	0.0337	0.1065	<u>0.1060</u>
LPIPS _{Alex} ↓	0.1533	0.0215	0.1112	<u>0.1034</u>

Table 1. Comparison of different methods for fashion object editing. The best, second-best, and worst values are highlighted in **bold**, underlined, and gray, respectively. Values represent averages across all edit types and prompts. Our approach achieves a better trade-off between faithfulness to edit prompts and preservation of source 3D objects. While PDS exhibits lower LPIPS, it mostly generates edited objects that are nearly identical to the source, as evidenced by CLIP directional similarity.

line for the iterative dataset update strategy.

We use several backbones for each metric and note them in the subscripts, while averaged refers to the average value across all backbones.

5.1. 3D Object Editing

Fashion object editing. In our fashion object editing experiment, we use MVDream [49] to generate a synthetic dataset of 3D fashion objects and perform edits using our framework. Specifically, we categorize the fashion object edits into color, pattern, shape, pose, and object edits, and synthesize a total of 150 editing examples, with 30 for each type of edit. In this experiment, our framework takes as input a source object in NeRF with the desired edits to its characteristics, such as color, pattern, shape, pose, and added objects. Note that, unlike some previous work [25], we do not require a description of the original object.

The qualitative results, as shown in Fig. 3 and Fig. 4, demonstrate the ability to apply versatile and various types

of edits, respectively. In Fig. 4, we apply five types of edits: color, pattern, shape, pose, and object additions. We illustrate that our framework supports a wide range of edits, including those requiring pose changes and introducing new objects, thereby corroborating its capability for significant geometric modifications, which have rarely been addressed in previous literature [8, 25].

General object editing. We propose a general object editing capability for our method. For this experiment, we edited objects from Objaverse [4]. First, we optimized InstantNGP [38] with 200 rendered images and camera transforms extracted from an object. Subsequently, we applied our editing method as planned.

We display the qualitative results of our method in Fig. 5. In addition to textural changes, our method successfully enables large density or structural changes. In Fig. 6, we also demonstrate our method’s ability to perform creative edits anchored by the source 3D object. Here, to demonstrate



Figure 6. Comparisons with Instruct-NeRF2NeRF (dataset update) [8] and MVDream (regeneration) [49] in editing Objaverse objects.

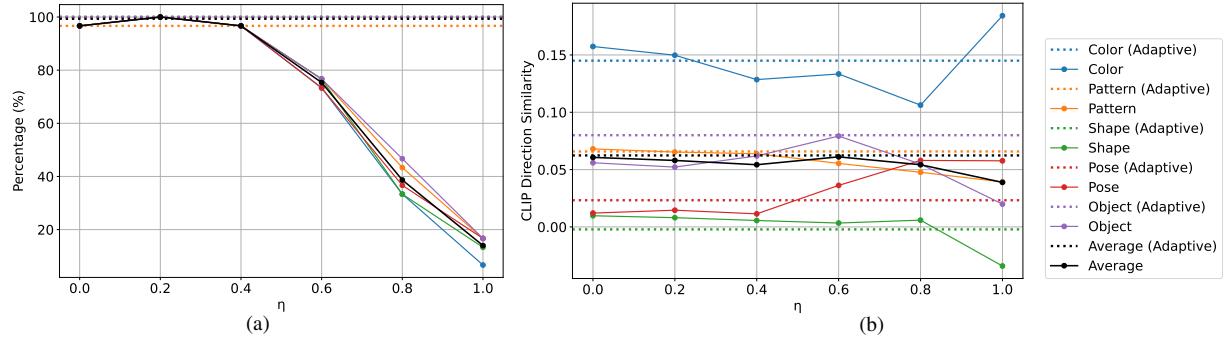


Figure 7. Ablation study on the selection of η . (a) and (b) show the CLIP direction similarity and the percentage of successful experiments (without errors) for different η values, respectively. When averaged across all types of edits, our adaptive method achieves near-maximum performance on these metrics compared to all fixed η values.

Method	CLIP-Dir-Sim↑	CLIP-Dir-Con↑	LPIPS↓
w/o Refinement	0.0624	0.7572	0.1147
w/ Refinement	0.0565	0.7642	0.1047

Table 2. Ablation study on the IPG steps. We present average values across all edit types, prompts, and evaluation backbones. Our approach yields a substantial reduction in LPIPS while balancing faithfulness to the source object and edit prompt.

flexibility, we use two prompts with different animal motifs. We can see that regenerating the object with MVDream produces results that appear completely different from the source 3D object. In contrast, our method successfully anchors the original shape while drastically changing it to align with the edit prompt.

5.2. Comparison with Baselines

We compare our method against baselines to demonstrate our clear advantage in complex edits, as shown in Figs. 4 and 5. For the comparison, we display editing results using SDS with MVDream, Instruct-NeRF2NeRF, and PDS. Additionally, we compare our strategy with dataset update and regeneration strategies in Fig. 6.

Different from other methods, SDS exhibits blurry textures and lacks the capability to perform complex edits.

While it can handle simple changes, it significantly alters the appearance and texture of the source objects and is unable to handle edits that require extensive geometric changes as seen in the 3rd, 4th, and 5th rows of Fig. 4. While Instruct-NeRF2NeRF successfully achieves simple color edits or symmetric edits, we observe that even when the editing process converges, it is not able to address geometric changes in the objects, such as a change in the pose. PDS, while maintaining similarity to the source object, is not capable of making significant edits and does not deviate far from the local minima. Our combined method, though adopting parameter ODE of score distillation, achieves state-of-the-art results for editing color, appearance, and geometry in a flexible and consistent manner. As shown in Figs. 4, 5, and 6, our method is capable of various types of edits, including those involving large geometric changes.

We have to consider both the similarity to the source object and faithfulness to the edit prompt. To this end, we evaluate these aspects using CLIP directional similarity [8] across ViT-B/32, ViT-B/16, and ViT-L/14 [44] for measuring faithfulness to the edit prompts and LPIPS [67] across VGG [50] and AlexNet [26] for evaluating the similarity between original and edited objects. The results are shown in Table 1. We can see that our method excels at both metrics

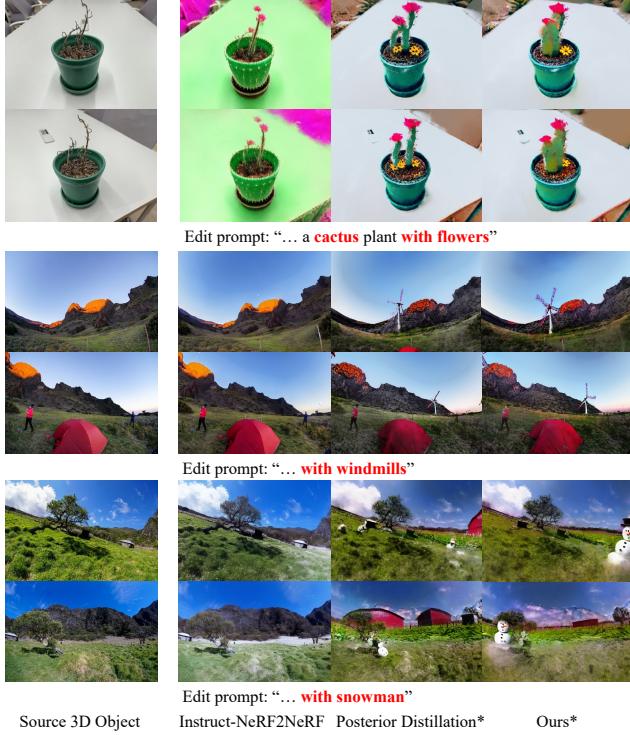


Figure 8. Real scene editing results. An asterisk (*) denotes the reduced optimization schedule used to improve computational efficiency. We used identical steps for both posterior distillation sampling and our method.

and achieves a better trade-off than other baselines. This, along with the quantitative evaluation, corroborates the effectiveness of our approach.

Computational efficiency. Our approach requires approximately 7 minutes with IG and 4 minutes without, to produce meaningful results. This is faster than the 13 minutes needed for Instruct-NeRF2NeRF. We attribute this to Instruct-NeRF2NeRF requiring updates to the entire dataset, while our method of parameter perturbation and timestep annealing benignly affects the optimization process of an object (Sec. 4.1). Note that completely regenerating an object requires around 26 minutes.

5.3. Real Scene Editing

Our parameter perturbation approach described in Sec. 4.1 can be readily extended to real scene editing scenarios [8, 25]. We present real scene editing results in Fig. 8, building upon Posterior Distillation [25]. Using the same number of iterations, we can better modify the scene geometry through parameter perturbation.

5.4. Ablation Study

We demonstrate the effectiveness and robustness of our design choices, particularly regarding parameter perturbation and identity-preserving gradients. We refer readers to the supplementary material for additional analyses.



Figure 9. Failure cases.

Parameter perturbation. In Fig. 3, we display the rendered images based on the amount of parameter perturbation and optimization steps. This demonstrates that parameter perturbation facilitates easy alteration of the source object’s structure while keeping crucial parts unchanged. Additionally, we present a quantitative study of η selection in Table 7, where we categorize edits into five categories: color, pattern, shape, pose, and object edits. We observe that pose changes and object additions generally require a larger degree of versatility and thus need a larger value of η to achieve maximum performance. Furthermore, we demonstrate that, when averaged across all types of edits, the adaptive η selection approach outperforms strategies using fixed $\eta \in \{0.0, 0.2, 0.4, 0.6, 0.8\}$ while maintaining high experimental success rates across various edit categories and significantly reducing computational costs compared to grid search. This indicates that our method dynamically selects an almost-optimal η value while remaining robust in terms of optimization. We present additional metrics in the supplementary materials.

Identity-preserving gradients. We demonstrate the effect of identity-preserving gradients. In Table 2, we present diverse metrics. Our results show that the refinement steps with IPG achieve a significant decrease in LPIPS while decreasing CLIP directional similarity, due to their trade-off relationship. In addition, we observe an overall increase in CLIP directional consistency [8].

6. Conclusion

Limitations. Although we address limitations of previous work, our method inherits some limitations from pretrained diffusion models [45, 49], such as color biases and saturation artifacts (Fig. 9). Additionally, while our method can handle pose and object changes, it rarely accommodates changes to the entire layout.

Conclusion. We present Perturb-and-Revise (PnR), a framework for text-guided 3D object editing. Through the introduction of adaptive parameter perturbation and identity-preserving gradients, our method enables extensive geometric and appearance changes that adhere to the text prompt while maintaining fidelity to source objects. Our experiments corroborate that PnR achieves state-of-the-art results across diverse editing tasks without requiring model retraining or multiple input images.

References

- [1] Andreas Blattmann, Robin Rombach, Huan Ling, Tim Dockhorn, Seung Wook Kim, Sanja Fidler, and Karsten Kreis. Align your latents: High-resolution video synthesis with latent diffusion models. *CVPR*, 2023. 2
- [2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. *CVPR*, 2023. 1, 2, 3
- [3] Changyou Chen, Ruiyi Zhang, Wenlin Wang, Bai Li, and Liqun Chen. A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*, 2018. 3, 12
- [4] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 2, 3, 6
- [5] Hanze Dong, Xi Wang, Yong Lin, and Tong Zhang. Particle-based variational inference with preconditioned functional gradient flow. *arXiv preprint arXiv:2211.13954*, 2022. 3, 12
- [6] Emilien Dupont, Hyunjik Kim, SM Eslami, Danilo Rezende, and Dan Rosenbaum. From data to functa: Your data point is a function and you should treat it like one. *ICML*, 2022. 2, 3
- [7] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, pages 12873–12883, 2021. 2
- [8] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. *arXiv preprint arXiv:2303.12789*, 2023. 1, 2, 5, 6, 7, 8, 15, 16, 17
- [9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 2
- [10] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2
- [11] Susung Hong. Smoothed energy guidance: Guiding diffusion models with reduced energy curvature of attention. *arXiv preprint arXiv:2408.00760*, 2024. 2
- [12] Susung Hong, Donghoon Ahn, and Seungryong Kim. Debiasing scores and prompts of 2d diffusion for view-consistent text-to-3d generation. *Advances in Neural Information Processing Systems*, 36:11970–11987, 2023. 2, 15
- [13] Susung Hong, Gyuseong Lee, Wooseok Jang, and Seungryong Kim. Improving sample quality of diffusion models using self-attention guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7462–7471, 2023. 2
- [14] Susung Hong, Junyoung Seo, Heeseong Shin, Sunghwan Hong, and Seungryong Kim. Direct2v: Large language models are frame-level directors for zero-shot text-to-video generation. *arXiv preprint arXiv:2305.14330*, 2023. 2
- [15] Hanzhuo Huang, Yufan Feng, Cheng Shi, Lan Xu, Jingyi Yu, and Sibei Yang. Free-bloom: Zero-shot text-to-video generator with llm director and ldm animator. *arXiv preprint arXiv:2309.14494*, 2023. 2
- [16] Inbar Huberman-Spiegelglas, Vladimir Kulikov, and Tomer Michaeli. An edit friendly ddpm noise space: Inversion and manipulations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12469–12478, 2024. 3
- [17] Aapo Hyvärinen and Peter Dayan. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(4), 2005. 3
- [18] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, pages 867–876, 2022. 1
- [19] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 4
- [20] Johanna Karras, Aleksander Holynski, Ting-Chun Wang, and Ira Kemelmacher-Shlizerman. Dreampose: Fashion image-to-video synthesis via stable diffusion. *arXiv preprint arXiv:2304.06025*, 2023. 2
- [21] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 2022. 2, 3
- [22] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics (ToG)*, 42(4):1–14, 2023. 2
- [23] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. *arXiv preprint arXiv:2303.13439*, 2023. 2
- [24] Sosuke Kobayashi, Eiichi Matsumoto, and Vincent Sitzmann. Decomposing nerf for editing via feature field distillation. *Advances in Neural Information Processing Systems*, 35:23311–23330, 2022. 2
- [25] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13352–13361, 2024. 1, 3, 5, 6, 8, 15, 16, 17
- [26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 7
- [27] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 300–309, 2023. 2
- [28] Qiang Liu and Dilin Wang. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016. 3

- [29] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. *arXiv preprint arXiv:2303.11328*, 2023. 2
- [30] Steven Liu, Xiuming Zhang, Zhoutong Zhang, Richard Zhang, Jun-Yan Zhu, and Bryan Russell. Editing conditional radiance fields. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 5773–5783, 2021. 2
- [31] Ying-Tian Liu, Yuan-Chen Guo, Vikram Voleti, Ruizhi Shao, Chia-Hao Chen, Guan Luo, Zixin Zou, Chen Wang, Christian Laforte, Yan-Pei Cao, et al. Threestudio: A modular framework for diffusion-guided 3d generation. *ICCV*, 2023. 14
- [32] Chang Liua and Jun Zhub. Geometry in sampling methods: A review on manifold mcmc and particle-based variational inference methods. *Advancements in Bayesian Methods and Implementations*, 47:239, 2022. 3, 12
- [33] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7210–7219, 2021. 2
- [34] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jianjun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *ICLR*, 2022. 18
- [35] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2
- [36] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1, 2, 5
- [37] Chong Mou, Xintao Wang, Liangbin Xie, Jian Zhang, Zhongang Qi, Ying Shan, and Xiaohu Qie. T2i-adapter: Learning adapters to dig out more controllable ability for text-to-image diffusion models. *arXiv preprint arXiv:2302.08453*, 2023. 2
- [38] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Transactions on Graphics (ToG)*, 41(4):1–15, 2022. 2, 6, 16
- [39] Jacob Munkberg, Jon Hasselgren, Tianchang Shen, Jun Gao, Wenzheng Chen, Alex Evans, Thomas Müller, and Sanja Fidler. Extracting triangular 3d models, materials, and lighting from images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8280–8290, 2022. 2
- [40] Julian Ost, Fahim Mannan, Nils Thuerey, Julian Knodt, and Felix Heide. Neural scene graphs for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2856–2865, 2021. 2
- [41] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2
- [42] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988*, 2022. 1, 2, 3, 5, 6, 12, 16, 17
- [43] Chenyang Qi, Xiaodong Cun, Yong Zhang, Chenyang Lei, Xintao Wang, Ying Shan, and Qifeng Chen. Fatezero: Fusing attentions for zero-shot text-based video editing. *arXiv preprint arXiv:2303.09535*, 2023. 2
- [44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021. 3, 7
- [45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 3, 8, 16
- [46] 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, 3
- [47] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2
- [48] Junyoung Seo, Susung Hong, Wooseok Jang, Inès Hyeonsu Kim, Minseop Kwak, Doyup Lee, and Seungryong Kim. Retrieval-augmented score distillation for text-to-3d generation. *arXiv preprint arXiv:2402.02972*, 2024. 2
- [49] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2, 3, 5, 6, 7, 8, 16
- [50] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 7
- [51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 2
- [52] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *NeurIPS*, 32, 2019. 2
- [53] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020. 2
- [54] Pratul P Srinivasan, Boyang Deng, Xiuming Zhang, Matthew Tancik, Ben Mildenhall, and Jonathan T Barron. Nerv: Neural reflectance and visibility fields for relighting and view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7495–7504, 2021. 2

- [55] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, et al. Nerfstudio: A modular framework for neural radiance field development. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–12, 2023. [16](#)
- [56] Vadim Tschernezki, Iro Laina, Diane Larlus, and Andrea Vedaldi. Neural feature fusion fields: 3d distillation of self-supervised 2d image representations. In *2022 International Conference on 3D Vision (3DV)*, pages 443–453. IEEE, 2022. [2](#)
- [57] Dor Verbin, Peter Hedman, Ben Mildenhall, Todd Zickler, Jonathan T Barron, and Pratul P Srinivasan. Ref-nerf: Structured view-dependent appearance for neural radiance fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5481–5490. IEEE, 2022. [2](#)
- [58] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3835–3844, 2022. [2](#)
- [59] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2023. [2](#)
- [60] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12619–12629, 2023. [1, 2](#)
- [61] Ziyu Wang, Tongzheng Ren, Jun Zhu, and Bo Zhang. Function space particle optimization for bayesian neural networks. *arXiv preprint arXiv:1902.09754*, 2019. [3, 12](#)
- [62] Zhengyi Wang, Cheng Lu, Yikai Wang, Fan Bao, Chongxuan Li, Hang Su, and Jun Zhu. Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. *Advances in Neural Information Processing Systems*, 36, 2024. [2, 3, 5, 12](#)
- [63] Chen Henry Wu and Fernando De la Torre. A latent space of stochastic diffusion models for zero-shot image editing and guidance. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7378–7387, 2023. [3](#)
- [64] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *arXiv preprint arXiv:2212.11565*, 2022. [2](#)
- [65] Hong-Xing Yu, Leonidas J Guibas, and Jiajun Wu. Unsupervised discovery of object radiance fields. *arXiv preprint arXiv:2107.07905*, 2021. [2](#)
- [66] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. [2](#)
- [67] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. [7](#)
- [68] Luyang Zhu, Dawei Yang, Tyler Zhu, Fitsum Reda, William Chan, Chitwan Saharia, Mohammad Norouzi, and Ira Kemelmacher-Shlizerman. Tryondiffusion: A tale of two unets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4606–4615, 2023. [2](#)

Perturb-and-Revise: Flexible 3D Editing with Generative Trajectories

Supplementary Material

A. Score Distillation as Particle-Based Variational Inference

Our parameter perturbation and identity gradients build upon the mathematical intuition of the variational score distillation (VSD) approach [62], an extension of score distillation sampling (SDS) [42]. In this context, the parameters of NeRF during distillation are treated as particles.

VSD minimizes the KL divergence between a variational distribution $q^\gamma(x|c)$, which is implicitly modeled by γ , and the target distribution $p_\phi(x|c)$, which is implicitly modeled by the diffusion model ϕ . Incorporating timesteps and camera poses, the objective is formulated as follows:

$$\gamma^* := \arg \min_{\gamma} \mathbb{E}_{t,\psi} \left[\frac{\sigma_t}{\alpha_t} w(t) D_{\text{KL}}(q_t^\gamma(x_t|c,t) \| p_\phi(x_t|c,t)) \right] \quad (6)$$

where $\frac{\sigma_t}{\alpha_t}$ and $w(t)$ are diffusion-related weighting factors, and $q_t^\gamma(x_t|c,t)$ and $p_\phi(x_t|c,t)$ represent the distributions of noisy images to be modeled by diffusion models.

To minimize this objective, VSD employs particle-based variational inference based on Wasserstein gradient flow, as detailed in [3, 5, 32, 61]. Specifically, the Wasserstein gradient flow satisfies:

$$\frac{\partial \gamma_\tau}{\partial \tau} = \nabla \cdot (\gamma_\tau \nabla (\frac{\partial E}{\partial \gamma_\tau}(\gamma_\tau))) \quad (7)$$

In our case, the energy functional E is defined as follows:

$$E(\gamma) := \mathbb{E}_{t,\psi} \left[\frac{\sigma_t}{\alpha_t} w(t) D_{\text{KL}}(q_t^\gamma(x_t|c,t) \| p_\phi(x_t|c,t)) \right] \quad (8)$$

In the particle-based variational inference, particles represent samples from the variational distribution. A set of M particles $\{\theta^{(i)}\}_{i=1}^M \sim \gamma$ is iteratively updated following the velocity of particles [3]: $\frac{d\theta_\tau}{d\tau} = \nabla(\frac{\partial E}{\partial \gamma_\tau}(\gamma_\tau))$. With the energy function in Eq. 8, the particles follow the ordinary differential equation (ODE):

$$\frac{d\theta_\tau}{d\tau} = -\mathbb{E}_{t,\epsilon,\psi} \left[w(t) \left(-\sigma_t \nabla_{x_t} \log p_\phi(x_t|c,t) - (-\sigma_t \nabla_{x_t} \log q_t^{\gamma_\tau}(x_t|c,t)) \frac{\partial g(\theta_\tau, \psi)}{\partial \theta_\tau} \right) \right] \quad (9)$$

where τ denotes the ODE time, constrained to $\tau \geq 0$, and γ_τ progressively evolves toward the optimal distribution γ^* as $\tau \rightarrow \infty$. In this VSD framework, the gradient of the SDS loss is a specific instance of the equation [62], where a single particle represents the entire distribution.

B. Resulting Distribution from Parameter Interpolation (Sec. 4.1)

Here, we show that interpolating parameters with $\eta \in [0, 1]$ results in a versatile sampling distribution that interpolates between a point mass at θ_{src} and the initial distribution. Given a source parameter θ_{src} and an initial distribution $\mathcal{P}(\Theta_0)$ with bounded variance σ^2 , we define the parameter perturbation as:

$$\theta_{\text{perturbed}} = (1 - \eta)\theta_{\text{src}} + \eta\theta_0, \quad \theta_0 \sim \mathcal{P}(\Theta_0), \quad \eta \in [0, 1] \quad (10)$$

Using the change of variables formula with transformation $T(\theta_0) = (1 - \eta)\theta_{\text{src}} + \eta\theta_0$ and its inverse $T^{-1}(\theta_{\text{perturbed}}) = (\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}})/\eta$:

$$p(\theta_{\text{perturbed}}) = \mathcal{P}(\Theta_0)(T^{-1}(\theta_{\text{perturbed}})) \cdot |\det(J_{T^{-1}})| \quad (11)$$

Since the Jacobian matrix is $J_{T^{-1}} = \frac{1}{\eta}I_d$, where I_d is the d -dimensional identity matrix, we have:

$$p(\theta_{\text{perturbed}}) = \frac{1}{\eta^d} \mathcal{P}(\Theta_0) \left(\frac{\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}}}{\eta} \right) \quad (12)$$

Here, η controls the degree of interpolation through both a scale factor $\frac{1}{\eta^d}$ and the argument $(\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}})/\eta$ of $\mathcal{P}(\Theta_0)$.

For $\eta \rightarrow 1$, both terms approach simple limits:

$$\lim_{\eta \rightarrow 1} p(\theta_{\text{perturbed}}) = \lim_{\eta \rightarrow 1} \frac{1}{\eta^d} \mathcal{P}(\Theta_0) \left(\frac{\theta_{\text{perturbed}} - (1 - \eta)\theta_{\text{src}}}{\eta} \right) \quad (13)$$

$$= \mathcal{P}(\Theta_0)(\theta_{\text{perturbed}}) \quad (14)$$

For $\eta \rightarrow 0$, we consider the distribution of $\theta_{\text{perturbed}}$. By Chebyshev's inequality, for any $\varepsilon > 0$:

$$P(|\theta_{\text{perturbed}} - \mathbb{E}[\theta_{\text{perturbed}}]| \geq \varepsilon) \leq \frac{\eta^2 \sigma^2}{\varepsilon^2} \rightarrow 0 \quad \text{as } \eta \rightarrow 0 \quad (15)$$

Moreover, since $\mathbb{E}[\theta_{\text{perturbed}}] \rightarrow \theta_{\text{src}}$ as $\eta \rightarrow 0$:

$$P(|\theta_{\text{perturbed}} - \theta_{\text{src}}| \geq \varepsilon) \rightarrow 0 \quad \text{as } \eta \rightarrow 0 \quad (16)$$

This proves convergence in probability to θ_{src} . The $\frac{1}{\eta^d}$ factor ensures that the total probability remains 1, while the concentration around θ_{src} becomes arbitrarily tight as $\eta \rightarrow 0$, characterizing convergence to:

$$\lim_{\eta \rightarrow 0} p(\theta_{\text{perturbed}}) = \delta(\theta_{\text{perturbed}} - \theta_{\text{src}}) \quad (17)$$

Thus, we have shown that the interpolation of parameters results in an interpolation between two extremes: a point mass at θ_{src} and the initial distribution, and the parameter η controls the degree of interpolation, i.e., the versatility.

Algorithm 1: Parameter Perturbation

```
Function ParameterPerturbation( $\eta$ ):
     $\theta_{\text{new}} \leftarrow$  Initialize new geometry instance
    for  $(\theta_c, \theta_n, \theta_i)$  in zip( $\theta_{\text{current}}$ ,  $\theta_{\text{new}}$ ,  $\theta_{\text{init}}$ ) do
        |  $\theta_c \leftarrow (1 - \eta)\theta_i + \eta\theta_n$                                 // Parameter interpolation
    end
    Free memory and clear cache
```

Algorithm 2: Parameter Perturbation with Adaptive η Selection

Input: Empty loss history list \mathcal{L} , minimum loss decrease Δ_{\min} , maximum parameter perturbation η_{\max}

Input: Initial NeRF parameters θ_{init}

Function TrainingStep:

```
if  $|\mathcal{L}| = 50$  then
    |  $\Delta\mathcal{L} \leftarrow$  LossDecrease( $\mathcal{L}$ )
    |  $\eta \leftarrow$  DetermineEta( $\Delta\mathcal{L}, \Delta_{\min}, \eta_{\max}$ )
    | ParameterPerturbation( $\eta$ )
end
Proceed with training step
 $\mathcal{L} \leftarrow \mathcal{L} \oplus \{\text{Current step training loss}\}$ 
```

Function LossDecrease(\mathcal{L}):

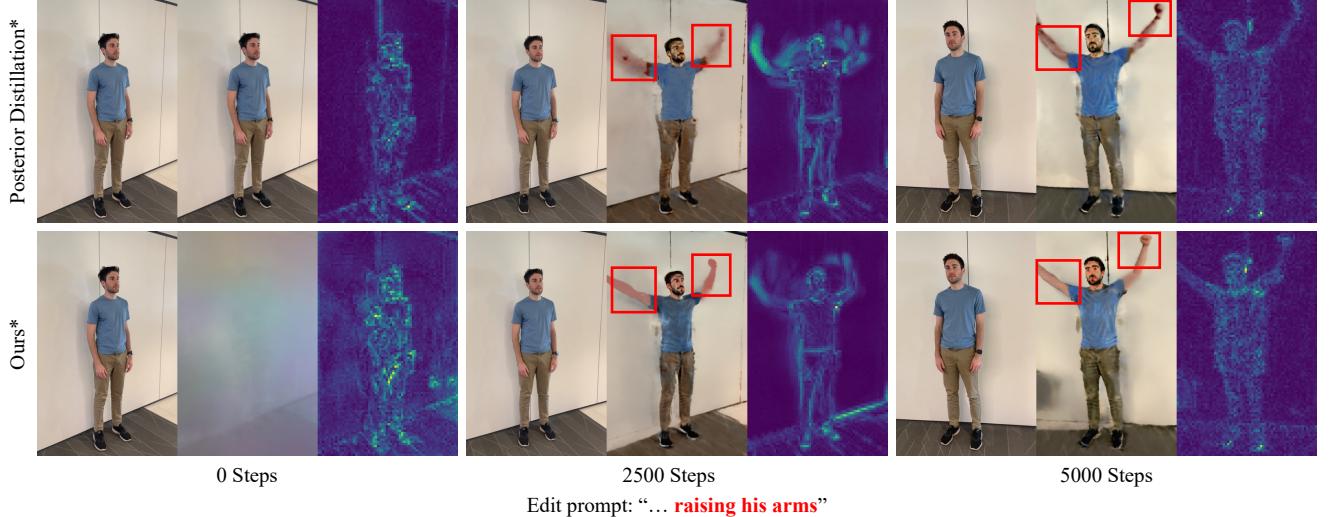
```
 $\mathcal{L}_{\text{final}} \leftarrow \frac{1}{10} \sum_{i=|\mathcal{L}|-10}^{|\mathcal{L}|} \mathcal{L}_i$                                 // Average of last 10 losses
 $\mathcal{L}_{\text{init}} \leftarrow \frac{1}{10} \sum_{i=1}^{10} \mathcal{L}_i$                                 // Average of first 10 losses
return  $\mathcal{L}_{\text{final}} - \mathcal{L}_{\text{init}}$ 
```

Function DetermineEta($\Delta\mathcal{L}, \Delta_{\min}, \eta_{\max}$):

```
return max(0,  $\eta_{\max}(1 - 2^{-(\Delta\mathcal{L} + \Delta_{\min})/\Delta_{\min}})$ )
```

C. Algorithms for Parameter Perturbation and Adaptive η Selection

We present the complete algorithms for parameter perturbation in Alg. 1 and the adaptive η selection method in Alg. 2. The underlying intuition for the adaptive η selection algorithm is that there exists a minimum loss decrease Δ_{\min} required for parameter perturbation and a maximum parameter perturbation η_{\max} that can be applied without resulting in complete regeneration of the object. Here, we have two parameters to control, Δ_{\min} and η_{\max} . Δ_{\min} is set to 1000 based on observations that it achieves near-optimal CLIP directional similarity and CLIP directional consistency, as shown in Table 3. η_{\max} is set to 0.6 based on the finding that the percentage of successful experiments drops significantly when η exceeds 0.6, as shown in the main paper. We also include the corresponding parts of our framework’s code, implemented in threestudio [31], in the supplementary material.



Edit prompt: "... **raising his arms**"



Edit prompt: "... a **cactus** plant **with flowers**"

Figure 10. Original scene, edited scene, and image-level gradients are shown at 0, 2500, and 5000 optimization steps. We can see that the density forms earlier and changes drastically even when the perturbation is large and barely has any structure.

D. Additional Analyses

Intermediate results from real scene editing. In Fig. 10, we show intermediate results from a real scene editing experiment. In this experiment, we aim to make the person raise the arms. Despite the initialization having little 3D structure, it converges faster with the same number of optimization steps. We can see that the density near the raised arms quickly converges with parameter perturbation, while the original PDS [25] generates blurry results. This demonstrates the effectiveness of our parameter perturbation approach in various editing scenarios.

Additional ablation study. We present additional ablation study results on Δ_{\min} . Additionally, we examine the effects of different values for λ_{L1} and λ_p in Table 4. Our results demonstrate that our method is relatively robust to these parameters, with our chosen values achieving a near-optimal balance across metrics. In addition, in Fig. 14, we display additional visualizations for the selection of η . A-LPIPS [12] is a metric for view consistency between adjacent frames, and CLIP directional consistency [8] is a metric that computes how much the editing directions differ across frames. Considering that we showed in the main paper that using fixed values of $\eta \geq 0.6$ had a higher likelihood of causing errors, our method outperforms approaches using fixed values of $\eta < 0.6$ in both metrics while maintaining lower error rates.

Additional comparisons. In Figs. 12 and 13, we showcase additional comparisons with the baseline methods.

Comparisons in 360° views. We present qualitative comparisons with 360° views on our project page.

Method	CLIP-Dir-Sim _{averaged} ↑	CLIP-Dir-Con _{averaged} ↑	LPIPS _{averaged} ↓
$\Delta_{\min} = 500$	0.061	0.757	0.112
$\Delta_{\min} = 1000$	0.062	0.757	0.115
$\Delta_{\min} = 2000$	0.060	0.754	0.111

Table 3. Experiment controlling Δ_{\min} .

Method	CLIP-Dir-Sim _{averaged} ↑	CLIP-Dir-Con _{averaged} ↑	LPIPS _{averaged} ↓
$\lambda_{L1} = 10000, \lambda_p = 100$	0.057	0.777	0.115
$\lambda_{L1} = 30000, \lambda_p = 300$	0.057	0.764	0.105
$\lambda_{L1} = 50000, \lambda_p = 500$	0.051	0.752	0.091

Table 4. Experiment controlling λ_{L1} and λ_p .

Effect of IPG. In Fig. 11, we demonstrate refinement outcomes through IPG and the generative ODE. A notable IPG attribute is its preservation of areas in the 3D object not explicitly specified for modification within the editing prompt.

E. Implementation Details

Optimization steps. For all perturbation values, we perform 1.5k editing steps, significantly fewer than the 10k steps required for regeneration [49]. We set a resolution milestone in both fashion and general object editing at which the rendering resolution changes for efficacy to half the number of editing steps. We perform 1k additional refinement steps, making the total runtime similar to 1.5k steps of PDS and thus highly efficient.

Identity-preserving gradients. For the identity-preserving gradients in Sec. 4.3, we adopt a combination of perceptual and L1 losses, finding this more stable and less fragile to noise than using only L1 or L2 loss. Specifically, we choose $\lambda_{L1} = 300.0$ and $\lambda_p = 30000.0$, with an annealed schedule, i.e., we linearly decrease them to 0 until the halfway point of the steps. We present an ablation study on the scales of λ_p and λ_{L1} in Table 4.

Timestep annealing. In the original Score Distillation [42], Instruct-NeRF2NeRF [8], and Posterior Distillation [25] papers, a fixed schedule, $\Sigma := \mathcal{U}(0.02, 0.98)$, is utilized. Contrary to this fixed schedule, and considering that our editing purpose does not inherently start from random parameters, we adopt a schedule in which $\Sigma(0) = \mathcal{U}(0.75, 0.75)$, a range to be decreased to (0.02, 0.4) by the time 80% of the total editing steps are reached.

NeRF representation. Technically, the parameter perturbation method can be applied to arbitrary representations whose parameters are initialized from a distribution and optimized. For computational efficiency while maintaining high quality of 3D objects, we choose InstantNGP [38] as our NeRF implementation.

Real scene editing. We show in the main paper that our parameter perturbation approach can be readily extended to real scene editing scenarios [8]. Using Nerfstudio’s implementation of Nerfacto [55] as the representation, we integrate Instruct-NeRF2NeRF [8] without modifications. For this experiment, we build upon the distillation method proposed in PDS [25], use Stable Diffusion v1-5 [45] as the backbone, and set $\eta = 0.6$. We reduce the timesteps by half and omit the selection and refinement steps for both PDS and our method to manage computational complexity and ensure a fair comparison of the parameter perturbation approach. Even with these shortened iteration steps, our parameter perturbation approach enables extensive geometric editing of the scene.

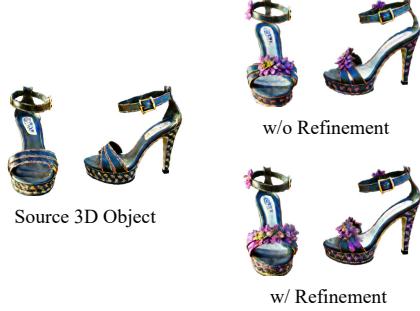


Figure 11. Effects of IPG. IPG restores changed regions that were not explicitly mentioned in the edit prompt during the editing process, for example, the support part of the strap.



Figure 12. Additional comparisons of fashion object editing with Score Distillation [42], Posterior Distillation [25], Instruct-NeRF2NeRF [8], and Perturb-and-Revise (ours).

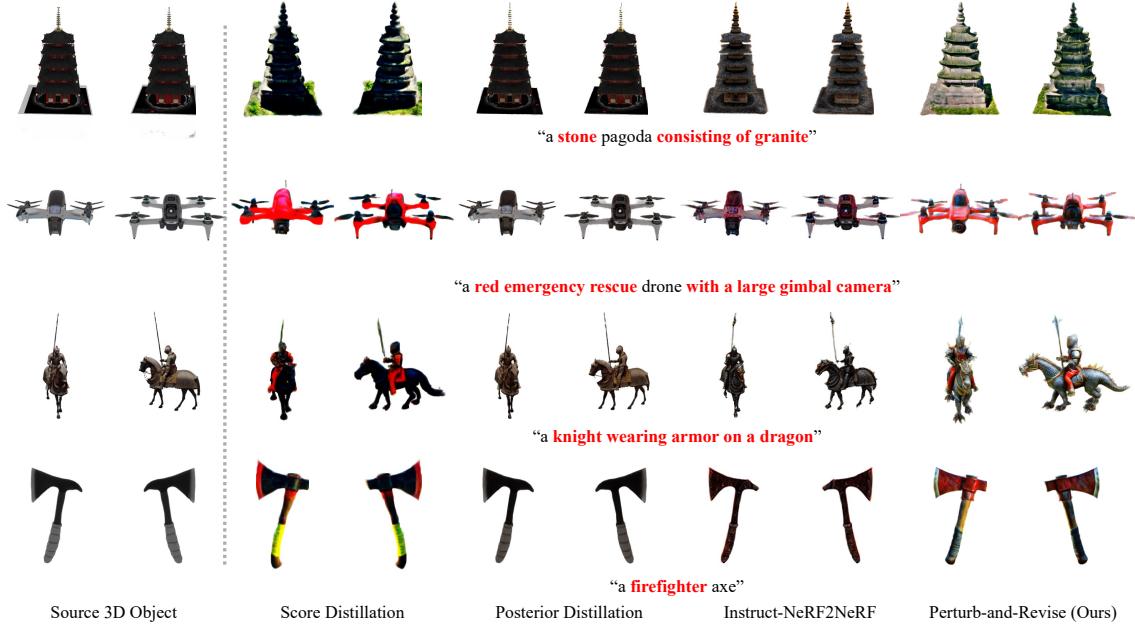


Figure 13. Additional comparisons of general object editing with Score Distillation [42], Posterior Distillation [25], Instruct-NeRF2NeRF [8], and Perturb-and-Revise (ours).

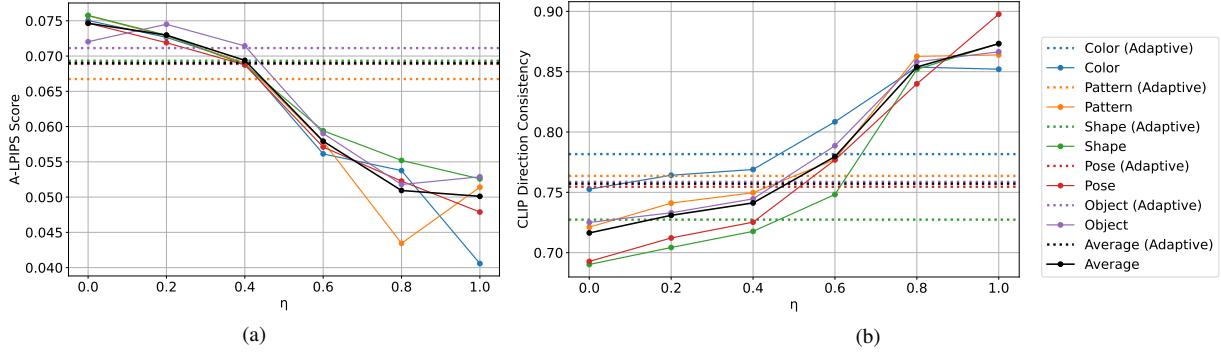


Figure 14. Additional visualizations for the selection of η . (a) and (b) show the A-LPIPS (lower is better) and CLIP directional consistency (higher is better) for different η values, respectively.

F. Discussion and Future Work

Perturb-and-Revise is a training-free editing method that is fast and effective, opening up new possibilities. The main point of the paper is that parameter perturbation is of prime importance in achieving this. Regarding this approach, one can draw an analogy with SDEdit [34], which injects Gaussian noise for image editing and is commonly adopted in many image editing pipelines. Indeed, PnR demonstrates that similar yet general principles can be applied in parameter space for 3D editing.

While PnR currently focuses on static 3D scenes, future research in this direction could extend the principles of parameter perturbation to 4D neural fields representing dynamic scenes. This extension would enable powerful video editing applications, such as modifying the motion of objects or characters while preserving their appearance and physical plausibility.