

InstructHumans: Editing Animated 3D Human Textures with Instructions

Jiayin Zhu¹, Linlin Yang², and Angela Yao¹

¹ National University of Singapore
zhujiayin@u.nus.edu, ayao@comp.nus.edu.sg
² Communication University of China
lyang@cuc.edu.cn

Abstract. We present InstructHumans, a novel framework for instruction-driven 3D human texture editing. Existing text-based editing methods use Score Distillation Sampling (SDS) to distill guidance from generative models. This work shows that naively using such scores is harmful to editing as they destroy consistency with the source avatar. Instead, we propose an alternate SDS for Editing (SDS-E) that selectively incorporates subterms of SDS across diffusion timesteps. We further enhance SDS-E with spatial smoothness regularization and gradient-based view-point sampling to achieve high-quality edits with sharp and high-fidelity detailing. InstructHumans significantly outperforms existing 3D editing methods, consistent with the initial avatar while faithful to the textual instructions. Project page: <https://jyzhu.top/instruct-humans>.

Keywords: 3D Human Texture Editing · Text-guided Editing

1 Introduction

3D human avatars have broad applications in gaming, virtual reality (VR), and augmented reality (AR) and there is a growing demand for intuitive methods to create, customize and edit such avatars. With the recent development of large vision-language models [25] and diffusion-based generative models [2, 26], natural language has become a control signal to generate [10, 19] and edit human avatars [5, 21]. This work presents the first method for text-guided *editing* of animatable 3D human avatars. Animatable avatars can be re-posed and offer control over the 3D human pose, though this adds challenges in aligning texture edits with a parametric animation or pose model. Previous works on text-guided 3D avatars are either not animatable [5, 21] or not editable [10, 19].

A straightforward way of editing [7] is to alter the texture for a certain range of mesh vertices. However, this method is complex and cumbersome, as it involves manual texture definition and vertex selection. To make editing more intuitive, we adopt the approach of InstructPix2Pix [2] that allows users to specify editing with natural language. InstructPix2Pix performs 2D image editing through a diffusion model given image and text instruction as conditions.



Fig. 1: Our approach provides animatable and editable 3D avatars. The outputs are rich in detail while remaining consistent with the source avatar and editing instructions.

Score Distillation Sampling (SDS) [24] leverages the predicted noise of a 2D diffusion model to guide the update direction for 3D models. Approach-wise, SDS is straightforward and works well for 3D generation [17, 19, 20, 24, 27, 33] and has been extended to 3D editing of scenes [15] and human avatars [21]. Yet the resulting edited textures of the avatars tend to be blurry and compromise characteristics that should remain constant, such as facial identity, or clothing. These poor-quality results are partially addressed by fine-tuning a personalized diffusion model [21], but leave much room for improvement.

Rather than post-processing, we study what we believe to be the cause of the poor-quality edits - the SDS guidance signal. Unlike generation tasks starting from a noise base, editing tasks already have a pre-defined source. For editing, it is necessary to preserve certain aspects of the source - in our case, the 3D geometry, and any unaffected facial or clothing texture sources that are not specified by the edit. This dichotomy of “preservation” versus “change” presents an inherent conflict with the guidance direction.

To further investigate, we break down SDS into individual terms. The baseline-shift term leads to structure formation in the early stage of denoising and is counterproductive to editing, as it causes shifts away from the original structures. Other terms, like the condition-divergence term and full-condition term, are beneficial only at the appropriate optimization stages. Naively applied at all stages, which is the case in standard SDS, these terms have conflicting and counter-productive effects which result in poor-quality edits. Our findings lead us to design the customized SDS-Editing (SDS-E) to distill guidance dedicated towards 3D editing. Specifically, SDS-E introduces a temporal staging to selectively apply the SDS terms with non-increasing sampled timesteps during optimization, allowing control over the terms’ impact on the editing guidance.

We also investigate the spatial distribution of the distilled guidance and make two innovations that improve the efficiency and quality of the edits. First, we propose a gradient-aware view sampling strategy to dynamically allocate more camera viewpoints based on the need for guidance. This sampling strategy directs the editing focus toward desired regions and speeds up the overall editing convergence. Secondly, we propose a smoothness regularizer to improve spatial coherence and mitigate spotting and other artifacts in the resulting textures.

Summarizing our contributions: (1) We perform an in-depth analysis of SDS for the 3D editing context and reveal the changing roles of the different SDS

terms over the denoising process; (2) We introduce SDS for Editing with non-increasing timestep sampling and selective temporal staging of the SDS terms to distill effective editing guidance; (3) A gradient-aware camera view sampling that improves editing efficiency and specificity; and (4) a smoothness regularizer that enhances the quality of the edited textures by encouraging information exchange in local latent codes. Experiments verify that our framework for instruction-driven editing of 3D human avatars is highly effective. It is efficient and flexible, and can produce faithful edits while maintaining consistency with the original avatar. The resulting edited avatars are high-fidelity and adhere to the textual instructions, all while remaining animatable.

2 Related Works

Text-guided 3D Editing. Traditional 3D editing procedures typically require explicit visual guidance, such as 3D cages [14] and segmentation masks [29]. Utilizing the power of 2D vision-language models [2, 26, 28], recent works intuitively optimize 3D objects via text guidance. One line of works adopts CLIP-based similarity as losses to guide the 3D editing [1, 10, 12, 13]. The outputs however, are unrealistic and often require additional fine-tuning such as GANs. On the other hand, SDS [24] succeeds in distilling information from 2D diffusion models. The use of the predicted noise to guide the update direction of 2D models is straightforward yet practical. Meanwhile, it is also efficient in not requiring back-propagation through large diffusion models. As such, SDS has been adopted by many recent text-to-3D generation works to generate scenes and human avatars [18, 19, 24, 31, 34]. Some works propose to improve SDS. DreamTime [11] proposes a non-increasing timestep sampling strategy. CSD [36] proposes using only classifier guidance in SDS. From a similar perspective to decompose SDS into subterms by considering classifier-free guidance, SSD [30] analyzes the subterms’ roles and proposes to add an adaptive estimator to reduce the variance.

Different from generation, editing optimizes an original source that is already given, seeking consistency with both the original source and the editing changes. DDS [6] tackles this difference in 2D image editing by additionally estimating the score of the original image-text pair. Several works also adopt SDS into 3D editing [15, 21, 35, 37], but all of them use SDS naively in its full form.

3D Human Editing. Recent methodologies for avatar manipulation face various limitations. Some focus narrowly on specific areas, like the head [1, 12, 21] or upper body [37], constraining their scope. Others, such as Instruct-NeRF2NeRF [5] and NeRF-Art [32], utilize implicit NeRFs for human representation, which lacks flexibility in control. A different approach involves explicit mesh representations, as seen in Text2Mesh [22] and Chupa [16], which trade off accuracy and diversity for explicit topology. While there are efforts towards generating drivable 3D humans [3, 10, 17, 19], the domain of editing these representations remains underexplored. EditableHumans [7] leverages a parametric human mesh model that encodes local texture and geometry features at mesh vertices, merging the

explicit structure of meshes with the representational and editing versatility of NeRFs. Building on this hybrid model, our work introduces text-driven editing for animatable humans, offering intuitive control and flexible representations.

3 Score Distillation Sampling for Editing (SDS-E)

3.1 Preliminaries

InstructPix2Pix (IP2P) [2] is a text-driven image-editing diffusion model. IP2P edits source image I according to text-instructions y by iteratively reducing estimated noise $\hat{\epsilon}_\phi$ from a noisy latent representation of the image $z = \mathcal{E}(I)$. To that end, IP2P optimizes the following objective:

$$\mathcal{L}_{\text{IP2P}}(z, t, y, I) = w(t) \|\hat{\epsilon}_\phi(z_t, t, y, I) - \epsilon\|_2^2. \quad (1)$$

Above, $t \in T$ denotes a uniform randomly sampled timestep, $\epsilon \sim \mathcal{N}(0, \mathbf{I})$ is the ground truth Gaussian noise and $w(t)$ is a weighting function depending on t . z_t is the noisy latent at timestep t , and is generated through an iterative forward diffusion process: $z_t = \sqrt{\alpha_t}z + \sqrt{1 - \alpha_t}\epsilon$, where the coefficient α_t represents a predefined noise schedule.

Classifier-free Guidance (CFG) [9] allows for adjustable adherence to the specified conditions through hyperparameter tuning. For a two-condition model like IP2P with conditions I and y , CFG is expressed as a conditional probability based on I and y , with hyperparameters ω_t and ω_I , respectively:

$$\begin{aligned} \hat{\epsilon}_\phi^{CFG}(z_t, t, y, I) = & \hat{\epsilon}_\phi(z_t, t, \emptyset, \emptyset) + \omega_I \cdot (\hat{\epsilon}_\phi(z_t, t, \emptyset, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, \emptyset)) \\ & + \omega_t \cdot (\hat{\epsilon}_\phi(z_t, t, y, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, I)). \end{aligned} \quad (2)$$

Score Distillation Sampling (SDS) [24] leverages pre-trained 2D image diffusion models to facilitate 3D generation. By applying the normal denoising process of IP2P to a rendered image from a 3D model, SDS can be used to distill editing guidance from the diffusion model into a 3D model. Specifically, SDS assumes that the diffusion model’s noise ϵ correlates with the score function (the gradient of the log-density) of the perturbed data distribution [8]:

$$\hat{\epsilon}_\phi = -\sigma_t \nabla_{z_t} \log p(z_t; t, y, I), \quad \text{where } \sigma_t = \sqrt{1 - \alpha_t}. \quad (3)$$

This assumption means SDS directs updates towards the data distribution $p(z_t)$ ’s high-density regions. Applied to a 3D model parameterized by Θ , the gradient can be given as:

$$\nabla_\Theta \mathcal{L}_{\text{SDS}}(\phi, z) = [w(t)(\hat{\epsilon}_\phi^{CFG}(z_t; t, y, I) - \epsilon) \frac{\partial z_t}{\partial \Theta}], \quad (4)$$

where $\hat{\epsilon}_\phi^{CFG}(z_t; t, y, I)$ is the IP2P model’s noise estimation guided by CFG.

3.2 Score Distillation Sampling for Editing

Timestep Sampling. Standard SDS uses uniformly random sampling for timesteps t . Previous analysis shows that large timesteps are crucial for forming coarse features, while middle³ and small timesteps are geared towards detailing [4]. In the context of 3D editing, we are already provided with a source 3D representation, so large timesteps serve little value. In fact, having them may even break up the initial structure and deviate from the original. As such, we opt to fully remove large timesteps from the sampling.

Previously, [11] proposed a non-increasing timestep sampling strategy which they showed to be more informative for updating 3D neural fields. We refer to the Supplementary for detailed equations; the sampling strategy enforces a monotonically decreasing envelope function to ensure that sampled timesteps are non-increasing. We observe that using this sampling strategy for our 3D human editing is more effective, as the successively smaller timesteps facilitate the escape of intermediate modes and promote convergence towards the optimal edited mode.

Decomposition of SDS. Our analysis begins with decomposing SDS for a two-condition diffusion model. This process aims to distinguish the editing directions influenced by the conditions and those influenced by the baseline (unconditioned) noise model. We substitute Eq. 2 into Eq. 4 and split it into two parts as follows:

$$\begin{aligned} \hat{\epsilon}_\phi^{CFG}(z_t; t, y, I) - \epsilon &= (\omega_I - 1) \cdot \underbrace{(\hat{\epsilon}_\phi(z_t, t, \emptyset, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, \emptyset))}_{m_1} \\ &+ \underbrace{\omega_t \cdot (\hat{\epsilon}_\phi(z_t, t, y, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, I)) + \hat{\epsilon}_\phi(z_t, t, \emptyset, I) - \epsilon}_{m_2}. \end{aligned} \quad (5)$$

The first part, m_1 , weighted by $\omega_I - 1$, is a *baseline-shift term*. m_1 quantifies the divergence induced by the image condition I , since it measures the shift from a baseline (unconditioned) noise model to a conditionally influenced model. Note this term measures shift from the image condition I only, and does not account for the text instruction condition. The second part, m_2 , is a *condition-integration term*, as it integrates the condition of the text instruction y and helps align the generated output with the specified conditions of the image I and text y .

Since there is both a text instruction condition y and an image condition I in m_2 , it can be further re-arranged into a form analogous to Eq. 5:

$$m_2 = (\omega_t - 1) \cdot \underbrace{(\hat{\epsilon}_\phi(z_t, t, y, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, I))}_{m_3} + \underbrace{\hat{\epsilon}_\phi(z_t, t, y, I) - \epsilon}_{m_4}. \quad (6)$$

³ Timestep sizing is relative. To facilitate our discussion, we separate “small”, “middle” and “large” timesteps, though our “small” and “middle” correspond to the “small” timesteps of [30], as they only make a distinction between “small” and “large”.

The term m_3 , weighted by $\omega_t - 1$, is a *condition-divergence term* that measures the adjustment needed when shifting from a base image condition to integrate the text condition y . Meanwhile, m_4 is the *full-condition term*, as it captures the model’s output with full consideration of the conditions.

While our decomposition of SDS is for a two-condition model, it has broader applicability as similar principles can be applied to models with a different number of conditions. For example, concurrent work [30] can be regarded as a simplified case of our decomposition for a single condition model.

Analysis of the Baseline-Shift Term m_1 . Based on Eq. 3, the baseline-shift term can be represented by the score function as follows:

$$m_1 = \hat{\epsilon}_\phi(z_t, t, \emptyset, I) - \hat{\epsilon}_\phi(z_t, t, \emptyset, \emptyset) = -\sigma_t (\nabla_{z_t} \log p_\phi(z_t; t, I) - \nabla_{z_t} \log p_\phi(z_t; t)). \quad (7)$$

Following the analysis of [30], the term m_1 causes shifts away from natural image distributions at small (and middle) timesteps. Specifically, when $t \rightarrow 0$, the distributions $p_\phi(z_t; t, I) \rightarrow p_\phi(z; t, I)$ and $p_\phi(z_t; t) \rightarrow p_\phi(z; t)$. As such, the target of this term can be regarded as maximizing:

$$\frac{p_\phi(z_t; t, I)}{p_\phi(z_t; t)} \rightarrow \frac{p_\phi(z; t, I)}{p_\phi(z; t)} = \frac{p_\phi(z; I)}{p_\phi(z)}. \quad (8)$$

An underlying assumption is that a latent z with a high conditional probability, *i.e.*, sourced from a mode of $p(z; I)$, corresponds inherently to a plausible image sourced from $p(z)$. Yet maximizing the term $\frac{p_\phi(z; I)}{p_\phi(z)}$ also requires minimizing $p_\phi(z)$, making it difficult to effectively target any mode of $p_\phi(z; I)$ and leads to shifts away from natural image distributions. Empirically, this results in saturated artifacts [30, 36]. As such, we follow [30]’s recommendation to omit this term for small (and middle) timesteps.

Analysis of the Condition-Divergence Term m_3 . Similar to m_1 , m_3 is characterized by two directional influences: one towards the two conditional mode $p_\phi(z_t; t, y, I)$ and one away from the image conditional mode $p_\phi(z_t; t, I)$. Again, in the limit that $t \rightarrow 0$, the condition-divergence term will then maximize

$$\frac{p_\phi(z_t; t, y, I)}{p_\phi(z_t; t, I)} \rightarrow \frac{p_\phi(z; t, y, I)}{p_\phi(z; t, I)} = \frac{p_\phi(z; y, I)}{p_\phi(z; I)}. \quad (9)$$

Analogous to m_1 , we presume that a mode of $p(z; y, I)$ should also be a mode of $p(z; I)$. Yet Eq. 9 cannot effectively drive the editing process towards a maximum of $p_\phi(z; y, I)$ where $p_\phi(z; I)$ is also high, as the latter appears in the denominator and gets minimized during the optimization process. This discourages convergence to any significant mode of the distribution $p_\phi(z; I)$, thereby distancing the result from the original image.

As such, we should remove the m_3 term at small timesteps. However, empirical evidence suggests that m_3 significantly contributes to the guide towards the

Table 1: Impact of SDS terms at different timesteps. The shading indicates utility; red and green denote harmful and helpful respectively, while yellow denotes mixed effects.

Timestep Sizing	m_1	m_3	m_4
Large (> 800)	Counterproductive		
Middle (150-800)	Saturation	Counterbalance	Intermediate trap
Small (< 150)		Distant from image	Two condition

text-conditioned mode, improving alignment with instructions. This presents a nuanced trade-off between editing faithfulness and image fidelity. Therefore, the decision to retain or exclude m_3 at small timesteps is left as an open consideration, reflecting a balance between these two aspects. In principle, m_3 should also be removed for middle timesteps; however, this would leave only the m_4 term for guidance, which is problematic in its own right. We further elaborate in the analysis on m_4 .

Analysis of the Full-Condition Term m_4 . The full-condition term m_4 can be viewed as a guide towards a two-condition mode $p_\phi(z_t; t, y, I)$. It is augmented by a factor $-\epsilon$ that counterbalances the variance introduced by the noise without altering the targeted mode:

$$m_4 = \hat{\epsilon}_\phi(z_t, t, y, I) - \epsilon = -\sigma_t \nabla_{z_t} \log p_\phi(z_t; t, y, I) - \epsilon. \quad (10)$$

Applying m_4 alone may lead the model to be trapped in intermediate modes. In particular, in large or middle timesteps, the denoising is incomplete, so the peak of a joint probabilistic density with multiple modes is likely higher than that of any individual desired mode [30]. This issue diminishes in smaller timesteps, when the probabilistic density of the desired mode becomes higher and can dominate the direction for the update. Yet in a uniformly random sampling strategy for timesteps, as in standard SDS, revisiting large or middle timesteps allows this issue to persist and disrupts the convergence to any desired mode. This is the root cause for over-smoothing by SDS [24, 30].

As such, we can either remove m_4 at middle timesteps and use only m_3 , or combine m_3 and m_4 (*i.e.* keep the full m_2 term). Empirically, the latter yields better results. Using m_3 alone causes the output to shift too far towards the text-conditioned mode and is problematic to optimize in its own right, as we analyzed previously. Using the two together allows m_4 to facilitate a balance of the text and image conditions while allowing m_3 to provide a counterbalance for breaking free of intermediate modes. Combining the m_3 and m_4 terms with non-increasing timestep sampling [11] produces the best results.

SDS-E: Score Distillation Sampling for Editing. Our analysis of the three SDS terms based on timestep size is summarized in Table 1. Based on these findings, we present a customized SDS for editing (SDS-E), where we selectively apply the terms at distinct timestep sizes.

For each sampled timestep t , SDS-E is defined as:

$$\mathcal{L}_{\text{SDS-E}} = \omega_t \cdot (\hat{\epsilon}_\phi(\mathbf{x}_t, t, y, I) - \hat{\epsilon}_\phi(\mathbf{x}_t, t, \emptyset, I)) + \hat{\epsilon}_\phi(\mathbf{x}_t, t, \emptyset, I) - \epsilon. \quad (11)$$

We also consider an alternative where the condition-divergence term m_3 is excluded at small timesteps:

$$\mathcal{L}'_{\text{SDS-E}} = \begin{cases} \mathcal{L}_{\text{SDS-E}} & \text{if } t > M \\ \hat{\epsilon}_\phi(\mathbf{x}_t, t, y, I) - \epsilon & \text{if } t \leq M, \end{cases} \quad (12)$$

where M is the threshold between small and middle timesteps. We set M empirically to 150; we also limit medium timesteps to 800 to exclude larger timesteps.

4 InstructHumans Editing Pipeline

Hybrid 3D Human Representation. We adopt the hybrid 3D human representation proposed by EditableHumans [7]. It associates an explicit 3D human mesh model, SMPL-X [23], with an implicit NeRF. Each mesh vertex from SMPL-X is linked with local geometry and texture latent codes. For a specific 3D avatar, it stores trainable latent codes Θ , obtained by barycentric interpolation of three local features accessed via vertex indices. EditableHumans also contains a pre-trained NeRF or implicit network, which outputs RGB color c and SDF value s for any queried global coordinate x_g . Specifically, the implicit network Φ is provided a local coordinate $x_l = \mathcal{M}(x_g)$ that is transformed from the global coordinate $x_g = (x, y, z)$ and a local normal vector \vec{n} . Conditioned on the latent codes Θ , the implicit network provides the following:

$$\Phi(\Theta, x_l, \vec{n}) = (c(x_g), s(x_g)). \quad (13)$$

The global to local coordinate transformation is performed by finding the nearest triangle on the body mesh of an input query point x_g , and transforming the position into local triangle coordinates $x_l = (u, v, d)$, where d is the distance. \vec{n} is calculated as the direction of the closest point on the mesh towards the global position to provide auxiliary information about the position. This coordinate transformation ensures the NeRF can only access local features and prevents it from memorizing any global information, and therefore provides the ability for disentangling local features for further editing.

Editing Pipeline. Fig. 2 visualizes the entire pipeline. Starting with an input human subject j and pre-trained latent codes Θ_j at mesh vertices, we optimize the latent codes to modify the human texture. In each iteration, we render an image I_v from a sampled camera view v using a conditional NeRF Φ . Image I_v is provided to IP2P for editing, conditioned on the instruction y and an original image rendered from the same view. We use our proposed SDS-E to distill editing gradients from IP2P (Sec. 3.2). The gradients, together with a

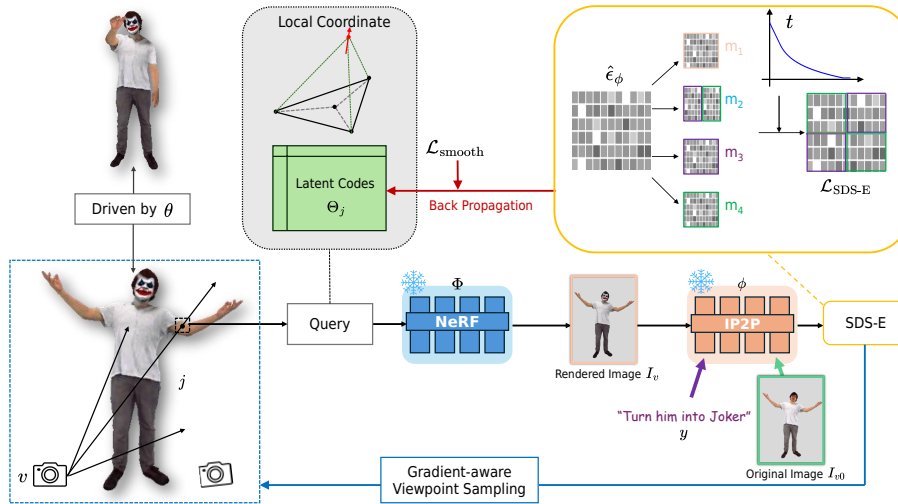


Fig. 2: Instruction-driven 3D human editing pipeline. Our pipeline optimizes a specific human subject’s texture based on textual instructions. Images rendered through a conditional NeRF are edited by IP2P, with SDS-E used to distill the editing gradients and update the texture latent codes. The editing is enhanced by gradient-aware viewpoint sampling and a smoothness regularizer. The edited avatar is easily drivable by altering pose parameters.

smoothness regularizer $\mathcal{L}_{\text{smooth}}$ (see Eq. 14), are backpropagated for optimizing the latent codes. Gradient-aware viewpoint sampling dynamically adjusts the camera views based on the gradients. The edited human is easily drivable by changing the SMPL-X pose parameter θ .

Laplacian Smoothness Regularization. EditableHumans [7] keeps texture latent codes at each vertex independent. This provides animation flexibility, as it disentangles the pose from the texture. For editing, this leads to the independent estimation of gradients for each local latent code. Empirically, we observe the independence assumption leads to uneven textures with spotting artifacts (see Fig. 6). We speculate that the learned latent space is not sufficiently smooth and therefore sensitive to minor perturbations. To mitigate the sensitivity and artifacts, we introduce a smoothness regularizer to encourage the coordination of the editing direction on local latent codes. Inspired by Laplacian loss used in 3D reconstruction to encourage smoothness in the vertex deformations, we propose a Laplacian latent smoothness:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{N} \sum_i^N \|(\vec{\mathcal{L}} \vec{\Delta F})_i\|^2, \quad (14)$$

where N is number of vertices, \vec{L} is the Laplacian matrix encoding the connectivity between vertices, and $\vec{\Delta F}$ is the matrix of delta latent codes, with each row representing the delta vector in latent space before and after one iteration. This regularizer penalizes large discrepancies in the delta latent codes between neighboring vertices, thereby encouraging local smoothness for resulting textures. The overall gradient is calculated as:

$$\nabla_{\theta}(w_1 \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{SDS-E}}). \quad (15)$$

Gradient-Aware Viewpoint Sampling. Another challenge we observed is that editing strengths are unevenly distributed in different body regions given different textual instructions. For example, “Put the person in a suit” primarily targets the clothing area, while “Turn him into Joker” emphasizes the facial features. Such uneven distribution undermines the efficiency of a uniform random viewpoint sampling due to misallocating editing efforts towards areas that do not require significant modification.

To address this issue, we introduce a simple yet effective viewpoint sampling technique utilizing the concept of editing strength. Thanks to the hybrid representation, we can use the average gradient magnitude anchored at a region of vertices to represent the editing strength of that region, and prioritize regions according to editing strengths. First, we split the 10,475 mesh vertices from SMPL-X into 5 regions based on their source: the face, the back of the head, the front body, the back of the body, and the arms. We then conduct one editing iteration with a batch of $|V|$ views uniformly sampled, and calculate the average gradient magnitude w_r across the region r :

$$w_r = \frac{1}{|V|} \frac{1}{|S_r|} \sum_{v \in V} \sum_{i \in S_r} \|\nabla(i)\|, \quad (16)$$

where V denotes the set of sampled views, and S_r represents the set of vertices within region r . Using w_r as the weight, we adjust the sampling of camera views for each region based on their normalized weights:

$$C(r) = \frac{w_r}{\sum_{r \in R} w_r} |V|, \quad (17)$$

where $C(r)$ is the total number of views sampled for region r .

This method effectively redistributes the number of camera views per region. Implementing this technique allows us to cap the number of sampled views at a predefined limit, *e.g.*, 1000, and significantly reduce the time required for rendering. Moreover, it also accelerates the convergence rate, leading to a reduction in the overall number of editing iterations needed. In addition, it improves the editing specificity on the desired regions, facilitating editing quality.

5 Experiments

Our goal is to edit drivable 3D human textures based on text instructions. Despite being the first to tackle this specific challenge, we ensure a robust evaluation

by conducting both qualitative (Sec. 5.1) and quantitative (Sec. 5.2) comparisons with baseline text-based editing methods for general scenes, including Instruct-NeRF2NeRF [5] and SDS [24]. To validate the effectiveness of our proposed methods, we provide a comprehensive ablation study (Sec. 5.3).

Implementation Details. We train 1000 steps and sample 50 camera views at each step, with rendering resolution 400×400 . The pre-trained NeRF and initial human latent codes are from [7]. The camera views in the body regions, including the front body, back body, and arms, are distributed at a circle with a radius 2 centered at the origin, and those in the head regions, including the face and back of the head, are set on a circle with a radius 0.35 centered at the avatar’s head. Camera positions are added with random offsets of 1%. The smoothness loss weight is set at $w_1 = 300$. The gradient-aware viewpoint sampling is conducted once at the first editing iteration. For CFG setting, $\omega_I = 1.5$ and $\omega_t = 7.5$. The editing process takes around 8 hours on an NVIDIA A5000 GPU.

5.1 Qualitative Experiment

We provide a qualitative comparison with Instruct-NeRF2NeRF (IN2N) [5] and SDS [24] on human subjects from the CustomHuman Dataset [7]. This comparison leverages editing instructions from IN2N, augmented with more challenging ones, such as "Turn him into wearing traditional Japanese kimono". The focus on 3D generation in [24] differs from our 3D editing task, precluding a direct comparison with SDS in its original framework. Hence, for a fair comparison, we utilize our reimplementation, substituting SDS-E with SDS within our editing pipeline. As illustrated in Fig. 3, our method surpasses existing techniques by delivering high-quality textures that not only adhere closely to the editing instructions but also faithfully retain the original human characteristics. More qualitative results showcasing the superiority of our approach can be observed in Fig. 4. Furthermore, a key distinction of our approach is the ability to produce animatable humans, in contrast to the non-animatable implicit NeRF representations generated by IN2N.

Application demo: Animation. We present a human animation demo in Fig. 1 (right) in which the edited human is driven by arbitrary SMPL-X poses.

5.2 Quantitative Experiment

Metrics. Following IN2N [5], we measure the CLIP text-image directional similarity (CLIP-Direc \uparrow) to evaluate how well the editing direction aligns with the textual instructions. To evaluate structural and semantic fidelity to the original avatar, we also evaluate the CLIP image similarity (CLIP-Img \uparrow) between the rendered images of the edited and original avatars. Note that high CLIP-Img scores may not solely indicate successful editing; they could also suggest editing failure. Thus, examining both metrics in conjunction allows us to discern a balance between maintaining consistency with the original image and achieving the intended editing outcomes.



Fig. 3: Qualitative comparison with IN2N [5] and SDS [24]. Our method excels in texture quality, adherence to the original avatars, and conformity to editing instructions, while IN2N struggles to maintain adherence to the original and produce high-quality textures, and SDS outputs textures with spotting and saturation.

Quantitative comparison with SOTA. In alignment with IN2N [5], our quantitative analysis covers 10 total edits across 2 unique subjects. Metrics comparisons are shown in Tab. 2 and Fig. 5. Our method outperforms IN2N and SDS in both terms of CLIP-Direc and CLIP-Img scores, achieving 0.162 and 0.838, respectively, demonstrating the superior ability to balance between the two objectives – preserving the original image’s essence while closely following the editing directives.

User Study. Since editing tasks lack definitive ground truths and are hence subjective, we conducted a user study in addition to metrics evaluations, through online Amazon Turk. Participants evaluated 10 pairs of editing comparisons considering overall quality, adherence to editing instructions, and fidelity to the original images. The survey reached 192 participants, garnering 800 responses in total. According to the results summarized in Tab. 3, our approach surpasses competing methods across all metrics, securing a 56.6% preference for overall quality, 46.1% for text consistency, and 51.3% for image consistency.

5.3 Ablation Study

We conduct ablation studies to evaluate the impact of key components in our 3D human editing pipeline. Fig. 6 (a) demonstrates the necessity of the smoothness regularizer, as its absence results in uneven textures with unrealistic spots, notably on the face. Omitting gradient-aware viewpoint sampling leads to an undesired shift in the edited face due to an imprecise editing focus. Moreover, excluding this component results in $5\times$ runtime.



Fig. 4: Qualitative visualization of our results. We illustrate 4 edits on each of 4 human subjects and obtain stable and high-quality editing results.

To validate the effectiveness of SDS-E, Fig. 6 (b) explores various design choices. Without SDS-E (*i.e.*, using standard SDS) significantly damages the original clothing and facial features and produces saturation. Omitting non-increasing timestep sampling adversely affects the convergence of clothing details, a consequence of intermediate traps detailed in Sec. 3.2. An alternative approach that excludes term m_4 during middle timesteps leads to deviations from desired image guidance. Comparing our default $\mathcal{L}_{\text{SDS-E}}$ with its alternative, $\mathcal{L}'_{\text{SDS-E}}$, the former achieves a balance between editing instructions and image adherence, while the latter preserves greater consistency with the original image, as observed in facial features. Therefore, we recommend a selective application of both loss functions, tailored to the specific editing contexts.

Method	CLIP-Direc \uparrow	CLIP-Img \uparrow
IN2N [5]	0.117	0.700
SDS [24]	0.151	0.827
Ours	0.162	0.838

Table 2: Quantitative Comparison. Our method is superior in both CLIP-Direc and CLIP-Img scores.

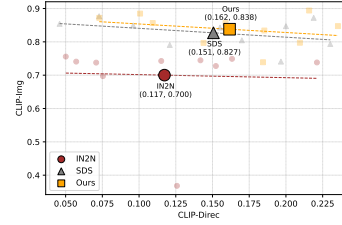


Fig. 5: We plot the comparison of our method against SOTA, showing a balancing between both metrics.

Table 3: User study results of 800 responses from 192 unique users. Ours is significantly more preferred considering all three metrics. We omit the "uncertain" choices.

Method	Quality	Text Consistency	Image Consistency
Instruct-NeRF2NeRF [5]	27.6%	34.6%	20.0%
SDS [24]	15.1%	18.5%	27.0%
InstructHumans (Ours)	56.6%	46.1%	51.3%

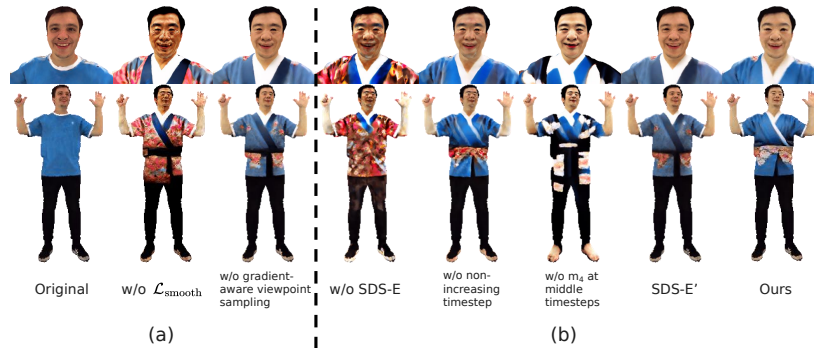


Fig. 6: Ablation studies.

6 Conclusion

This work presents a method for 3D human texture editing guided by textual instructions, which combines intuitive editing capabilities with the flexibility to animate the modified avatar. By analyzing and adapting SDS, we propose SDS for Editing (SDS-E), to distill faithful and high-fidelity editing guidance from the 2D diffusion model. Enhancements including Laplacian smoothness regularization and gradient-aware viewpoint sampling further augment the efficiency and effectiveness of our editing pipeline. Experiments affirm our method’s superior editing performance relative to existing text-based 3D editing approaches.

References

1. Aneja, S., Thies, J., Dai, A., Nießner, M.: Clipface: Text-guided editing of textured 3d morphable models. In: ACM SIGGRAPH 2023 Conference Proceedings. pp. 1–11 (2023) [3](#)
2. Brooks, T., Holynski, A., Efros, A.A.: Instructpix2pix: Learning to follow image editing instructions (Jan 2023). <https://doi.org/10.48550/arXiv.2211.09800>, <http://arxiv.org/abs/2211.09800>, arXiv:2211.09800 [cs] [1](#), [3](#), [4](#), [19](#), [21](#)
3. Cao, Y., Cao, Y.P., Han, K., Shan, Y., Wong, K.Y.K.: Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models. arXiv preprint arXiv:2304.00916 (2023) [3](#)
4. Choi, J., Lee, J., Shin, C., Kim, S., Kim, H., Yoon, S.: Perception prioritized training of diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11472–11481 (2022) [5](#)
5. Haque, A., Tancik, M., Efros, A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023) [1](#), [3](#), [11](#), [12](#), [14](#), [21](#)
6. Hertz, A., Aberman, K., Cohen-Or, D.: Delta denoising score. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 2328–2337 (October 2023) [3](#)
7. Ho, H.I., Xue, L., Song, J., Hilliges, O.: Learning locally editable virtual humans. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21024–21035 (2023) [1](#), [3](#), [8](#), [9](#), [11](#), [21](#)
8. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020) [4](#)
9. Ho, J., Salimans, T.: Classifier-free diffusion guidance (arXiv:2207.12598) (2022) [4](#)
10. Hong, F., Zhang, M., Pan, L., Cai, Z., Yang, L., Liu, Z.: Avatarclip: Zero-shot text-driven generation and animation of 3d avatars. *ACM Transactions on Graphics (TOG)* **41**(4), 1–19 (2022) [1](#), [3](#), [21](#)
11. Huang, Y., Wang, J., Shi, Y., Qi, X., Zha, Z.J., Zhang, L.: DreamTime: An Improved Optimization Strategy for Text-to-3D Content Creation (Jun 2023). <https://doi.org/10.48550/arXiv.2306.12422>, <http://arxiv.org/abs/2306.12422>, arXiv:2306.12422 [cs] [3](#), [5](#), [7](#), [19](#)
12. Hwang, S., Hyung, J., Kim, D., Kim, M., Choo, J.: Faceclipnerf: Text-driven 3d face manipulation using deformable neural radiance fields. In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 3446–3456. IEEE Computer Society, Los Alamitos, CA, USA (oct 2023). <https://doi.org/10.1109/ICCV51070.2023.00321>, <https://doi.ieeecomputersociety.org/10.1109/ICCV51070.2023.00321> [3](#)
13. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 867–876 (2022) [3](#)
14. Jambon, C., Kerbl, B., Kopanas, G., Diolatzis, S., Drettakis, G., Leimkühler, T.: Nerfshop: Interactive editing of neural radiance fields. *Proceedings of the ACM on Computer Graphics and Interactive Techniques* **6**(1) (2023) [3](#)
15. Kamata, H., Sakuma, Y., Hayakawa, A., Ishii, M., Narihira, T.: Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion (arXiv:2303.15780) (2023). <https://doi.org/10.48550/arXiv.2303.15780>, <http://arxiv.org/abs/2303.15780> [2](#), [3](#)

16. Kim, B., Kwon, P., Lee, K., Lee, M., Han, S., Kim, D., Joo, H.: Chupa: Carving 3d clothed humans from skinned shape priors using 2d diffusion probabilistic models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 15965–15976 (October 2023) **3**
17. Kolotouros, N., Alldieck, T., Zanfir, A., Bazavan, E., Fieraru, M., Sminchisescu, C.: Dreamhuman: Animatable 3d avatars from text. *Advances in Neural Information Processing Systems* **36** (2024) **2, 3**
18. Li, W., Chen, R., Chen, X., Tan, P.: SweetDreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d (arXiv:2310.02596) (2023). <https://doi.org/10.48550/arXiv.2310.02596>, <http://arxiv.org/abs/2310.02596> **3**
19. Liao, T., Yi, H., Xiu, Y., Tang, J., Huang, Y., Thies, J., Black, M.J.: TADA! Text to Animatable Digital Avatars. In: International Conference on 3D Vision (3DV) (2024) **1, 2, 3, 21**
20. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 300–309 (2023) **2**
21. Mendiratta, M., Pan, X., Elgharib, M., Teotia, K., R, M.B., Tewari, A., Golyanik, V., Kortylewski, A., Theobalt, C.: Avatarstudio: Text-driven editing of 3d dynamic human head avatars. *ACM Trans. Graph.* **42**(6) (dec 2023). <https://doi.org/10.1145/3618368>, <https://doi.org/10.1145/3618368> **1, 2, 3, 21**
22. Michel, O., Bar-On, R., Liu, R., Benaïm, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13492–13502 (June 2022) **3**
23. Pavlakos, G., Choutas, V., Ghorbani, N., Bolkart, T., Osman, A.A.A., Tzionas, D., Black, M.J.: Expressive body capture: 3D hands, face, and body from a single image. In: Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 10975–10985 (2019) **8**
24. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFusion: Text-to-3d using 2d diffusion (arXiv:2209.14988) (2022). <https://doi.org/10.48550/arXiv.2209.14988>, <http://arxiv.org/abs/2209.14988> **2, 3, 4, 7, 11, 12, 14**
25. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) **1**
26. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. arXiv preprint arXiv:2112.10752 (2021) **1, 3**
27. Ruiz, N., Li, Y., Jampani, V., Pritch, Y., Rubinstein, M., Aberman, K.: Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 22500–22510 (June 2023) **2**
28. Saharia, C., Chan, W., Saxena, S., Li, L., Whang, J., Denton, E.L., Ghasemipour, K., Gontijo Lopes, R., Karagol Ayan, B., Salimans, T., et al.: Photorealistic text-to-image diffusion models with deep language understanding. *Advances in Neural Information Processing Systems* **35**, 36479–36494 (2022) **3**
29. Sun, J., Wang, X., Shi, Y., Wang, L., Wang, J., Liu, Y.: Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis. *ACM Transactions on Graphics (TOG)* **41**(6), 1–10 (2022). <https://doi.org/10.1145/3550454.3555506> **3**

30. Tang, B., Wang, J., Wu, Z., Zhang, L.: Stable score distillation for high-quality 3d generation (arXiv:2312.09305) (2023) [3](#), [5](#), [6](#), [7](#)
31. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: DreamGaussian: Generative gaussian splatting for efficient 3d content creation (arXiv:2309.16653) (2023). <https://doi.org/10.48550/arXiv.2309.16653>, <http://arxiv.org/abs/2309.16653> [3](#)
32. Wang, C., Jiang, R., Chai, M., He, M., Chen, D., Liao, J.: Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics* (2023) [3](#), [21](#)
33. Wang, Z., Lu, C., Wang, Y., Bao, F., LI, C., Su, H., Zhu, J.: ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation. In: Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., Levine, S. (eds.) *Advances in Neural Information Processing Systems*. vol. 36, pp. 8406–8441. Curran Associates, Inc. (2023), https://proceedings.neurips.cc/paper_files/paper/2023/file/1a87980b9853e84dfb295855b425c262-Paper-Conference.pdf [2](#)
34. Yi, H., Zheng, Z., Xu, X., Chua, T.s.: Progressive text-to-3d generation for automatic 3d prototyping (arXiv:2309.14600) (2023). <https://doi.org/10.48550/arXiv.2309.14600>, <http://arxiv.org/abs/2309.14600> [3](#)
35. Yu, J., Zhu, H., Jiang, L., Loy, C.C., Cai, W., Wu, W.: PaintHuman: Towards high-fidelity text-to-3d human texturing via denoised score distillation (arXiv:2310.09458) (2023), <http://arxiv.org/abs/2310.09458> [3](#)
36. Yu, X., Guo, Y.C., Li, Y., Liang, D., Zhang, S.H., Qi, X.: Text-to-3D with Classifier Score Distillation (Oct 2023), <https://arxiv.org/abs/2310.19415v2> [3](#), [6](#)
37. Zhang, H., Feng, Y., Kulits, P., Wen, Y., Thies, J., Black, M.J.: Teca: Text-guided generation and editing of compositional 3d avatars. *arXiv* (arXiv:2309.07125) (2023) [3](#)

InstructHumans: Editing Animated 3D Human Textures with Instructions

Appendix

A Evaluating SDS Components Through a Toy Example

To assess the individual behaviors of the SDS components m_1 , m_2 , m_3 , and m_4 outlined in Sec. 3.2, we introduce a toy example depicted in Fig. A1. This experiment optimizes $z = \theta \in \mathbb{R}^2$, in a 2D space $p(z)$ that simulates the distribution of images. Without losing generality, we consider a mixture of Gaussian distributions $p(z) = 0.1\mathcal{N}([0, 0]^\top, 0.1\mathbf{I}) + 0.15\mathcal{N}([3, 1]^\top, 0.1\mathbf{I}) + 0.15\mathcal{N}([0.5, 1]^\top, 0.1\mathbf{I}) + 0.3\mathcal{N}([1.5, 1.4]^\top, 0.05\mathbf{I}) + 0.3\mathcal{N}([1.5, 0.4]^\top, 0.05\mathbf{I})$. Each Gaussian component signifies a mode influenced by specific conditions. Specifically, the component at $[0, 0]^\top$ represents the unconditional mode; the one at $[0.5, 1]^\top$ is the image conditional mode; $[3, 1]^\top$ corresponds to the text instruction conditional mode; and the components at $[1.5, 1.4]^\top$ and $[1.5, 0.4]^\top$ are two-conditional modes. We simulate the 3D editing process as guiding θ towards the two-conditional modes as closely as possible, using estimators m_1 , m_2 , m_3 , and m_4 adopted from Eq. 5 and Eq. 6, with their trajectories recorded.

Fig. A1 illustrates three distinct learning phases: The left plot shows the initial phase, where θ is initiated at the image conditional mode $[0.5, 1]^\top$ to emulate the onset of 3D editing from a plausible human figure. The middle plot showcases the density map of $P(z_t; y, I)$ at a middle timestep $t = 300$, starting θ at $[1.3, 1]^\top$ —positioned in the middle of two target modes to highlight the intermediate trapping issue discussed in Sec. 3.2. In the right plot, we present the density map of $P(z_t; y, I)$ at a small timestep $t = 100$, with θ starting at $[1.7, 0.7]^\top$, representing the latter stages of learning. Observations reveal that m_1 invariably leads θ in a counterproductive direction, as diverging from the unconditional mode fails to facilitate convergence to the target modes. In the phase with a middle timestep, m_4 suffers from trapping into an intermediate mode at the peak of the joint probabilistic density of two nearby dual-conditional modes, whereas m_3 demonstrates the capacity to extricate θ from this trapping point. This trait is similarly observed with m_2 , as it combines m_3 with m_4 . In the late phase with a small timestep, however, m_3 propels θ away from the desired high-density region, a consequence of distancing from the image conditional mode. While in this phase, both m_4 and m_2 steer θ towards the target. The effects of all the terms are coherently aligned with our analyses summarized in Tab. 1.

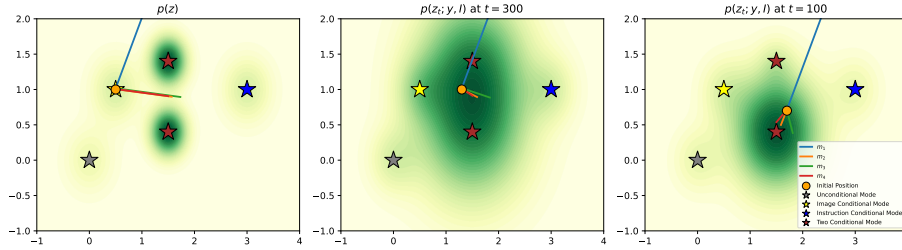


Fig. A1: Visualizing the impact of SDS components m_1 , m_2 , m_3 , and m_4 in a 2D toy example. Each component, serving as an estimator, guides the optimization of $z = \theta \in \mathbb{R}^2$ within a Gaussian mixture model representing $p(z)$. The objective is to guide θ towards the two-conditional modes (red stars). **Left:** In an early phase, θ is initiated at the image conditional mode (yellow star). The trajectory indicates that m_1 is counterproductive. **Center:** At middle timesteps, m_4 faces entrapment by an intermediate mode, while m_3 (and by extension, m_2) facilitates escape. **Right:** In small timesteps, as θ nears the target mode, m_4 and m_2 drives towards the denser region, while m_3 guides a deviated direction due to distancing from the image conditional mode. Detailed experimental settings are in Appendix A, and the detailed analysis of the terms is in Sec. 3.2.

B Non-Increasing Timestep Sampling

The non-increasing timestep sampling function \mathcal{S} as defined in DreamTime [11], is given by:

$$\mathcal{S}(t_i) = \arg \min_{t^*} \left| \sum_{t=t^*}^T p(t) - \frac{i}{N} \right|, \quad (\text{A1})$$

where T represents the total noise levels in the pre-trained IP2P model, N denotes the number of editing iterations, and $p(t)$ denotes the normalized time prior, calculated as $w^*(t) / \sum_{t=1}^T w^*(t)$. Here, $w^*(t)$ is a non-increasing weight function defined as:

$$w^*(t) = \begin{cases} e^{-(t-a_1)^2/2b_1^2} & \text{if } t > a_1 \\ 1 & \text{if } a_2 \leq t \leq a_1 \\ e^{-(t-a_2)^2/2b_2^2} & \text{if } t < a_2, \end{cases} \quad (\text{A2})$$

with hyperparameters $\{a_1, a_2, b_1, b_2\}$ modulating the decrease of timestep t . In IP2P [2], the timestep range is set to $[0, 1000]$.

C Impact of Gradient-Aware Viewpoint Sampling

We compare our proposed gradient-aware viewpoint sampling with a baseline uniform viewpoint sampling. Our superior editing quality is depicted in Fig. A2. The calculated weights w_r (see Eq. 16) and numbers of sampled views $\mathcal{C}(r)$ (see

Eq. 17) derived from our gradient-aware sampling are detailed in Tab. A1. Our sampling strategy successfully assigns weights to specific regions based on the editing instructions, ensuring that the editing focus is distributed strategically and efficiently.

Table A1: Weights w_r and number of sampled views $\mathcal{C}(r)$ for each region r derived from our gradient-aware sampling. The total number of sampled views $|V|$ is 50000. For the "kimono" instruction, body regions are assigned relatively higher weights and more sampled views, which aligns with the instruction's semantical targets, *i.e.*, clothing. In contrast, the head regions gain relatively more sampled views for the "clown" instruction, which necessitates more intensive editing of the head.

Instruction	Metric	Region				
		Face	Back of the head	Front body	Back of the body	Arms
"kimono"	Weight	0.04	0.08	0.47	0.26	0.15
	View Number	2000	4000	23500	13000	7500
"clown"	Weight	0.07	0.20	0.30	0.31	0.12
	View Number	3500	10000	15000	15500	6000

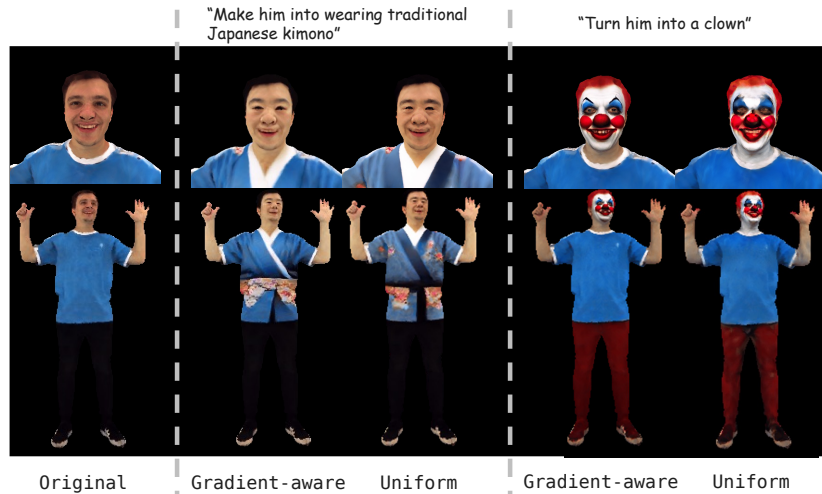


Fig. A2: Our proposed gradient-aware viewpoint sampling produces more efficient and specific editing by selectively focusing on desired regions per instruction, while the baseline uniform viewpoint sampling results in inaccurate and blurry results.

D Framework Comparison with Existing Works

We compare the capability of our work with existing works in Tab. A2. Ours is the first to enable editing full-body animatable humans with textual instructions.

Table A2: Our method is designed to edit full-body humans that are animatable given textual editing instructions. The other existing works can not fulfill all these functions. “Editing” indicates the method is designed to maintain consistency with the original personality.

Method	Text-guided Animatable Full-body Editing			
EditableHuman [7]	✗	✓	✓	✓
Instruct-NeRF2NeRF [5]	✓	✗	✓	✓
TADA [19]	✓	✓	✓	✗
AvatarCLIP [10]	✓	✓	✓	✗
AvatarStudio [21]	✓	✗	✗	✓
Ours	✓	✓	✓	✓

E Limitations and Potential Social Impacts

Our framework employs IP2P [2] for guiding texture edits on 3D human avatars. IP2P is pre-trained on non-human-specific datasets; therefore, its generalization to human editing might be limited, potentially affecting the accuracy and vividness of our edits. A viable approach to address this could be integrating other diffusion models fine-tuned for human avatar editing. In addition, our framework builds upon a hybrid human representation [7], which exhibits constraints in capturing very high-frequency details and may produce artifacts at joint areas under extreme poses. As suggested by [7], adopting a mesh template with more vertices and training with sufficiently large datasets are potential solutions. Our framework can easily incorporate improvements made in [7]. Finally, a common challenge in 3D editing is the disentanglement of geometric and textural changes given only 2D guidance. We focus on texture editing while maintaining the original geometry to help preserve the avatars’ individual identities. Although some existing works [5, 32] support concurrent geometric and textural editing, they achieve that in a rough manner, ignoring the texture-geometry ambiguity, therefore producing artifacts, such as cloudy geometric noises and broken geometric structures (see IN2N’s results in Fig. 5). Therefore, solving such ambiguity remains a challenge we aim to tackle in future research.

Concerning potential social impacts, the ability to edit human avatars raises issues related to copyright and privacy violations, and the dissemination of misleading or harmful content. These risks stem from applying pre-trained models without adequate safeguards against inappropriate use. By making our code publicly available, we hope to foster future research for detecting and combating inauthentic content involving 3D human editing.