InstructHumans: Editing Animated 3D Human Textures with Instructions

Jiayin Zhu, Linlin Yang, Angela Yao, *Member, IEEE*

***Abstract*—We present InstructHumans, a novel framework for instruction-driven animatable 3D human texture editing. Existing text-based 3D editing methods often directly apply Score Distillation Sampling (SDS). SDS, designed for generation tasks, cannot account for the defining requirement of editing – maintaining consistency with the source avatar. This work shows that naively using SDS harms editing, as it may destroy consistency. We propose a modified 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 viewpoint sampling for edits with sharp and high-fidelity detailing. Incorporating SDS-E into a 3D human texture editing framework allows us to outperform existing 3D editing methods. Our avatars faithfully reflect the textual edits while remaining consistent with the original avatars. Project page: https://jyzhu.top/instruct-humans/.**

***Index Terms*—3D Human Texture Editing, Text-guided Editing.**

I. Introduction

WITH the recent development of vision-language [1]–[5], natural language has emerged to become a control signal for generating and editing human avatars [6]–[10]. This work presents a novel method for text-guided *editing of animatable* 3D human avatars. Animatable avatars offer control over the 3D human pose, though this adds challenges in aligning texture edits with an animation or pose model. Previous works are largely either not animatable [8], [9] or not editable [6], [7], [11]. A recent work, TEDRA [12], studies text-based editing of dynamic avatars via subject-specific retraining, whereas we aim to edit generic animatable avatars while preserving identity consistency.

For intuitive handling of 3D avatars, we adapt Score Distillation Sampling (SDS) [13]. SDS leverages the predicted noise of a 2D diffusion model to guide 3D model optimization. SDS has been effective for 3D generation [6], [13]–[17] but directly applying it to editing can lead to blurriness and loss of essential characteristics like facial identity or clothing details. Fig. 2 (a) left shows an example of an avatar edited naively with SDS. It is blurry and wears a different outfit not specified in the editing text. Such drawbacks may be partially fixed by fine-tuning a personalized diffusion model [9] at the cost of extra compute.

We believe the cause of the poor-quality edits is the SDS guidance signal. SDS was originally designed for generation, where the 3D model is randomly initialized. Editing tasks, on the other hand, begin with an existing source avatar (see Fig. 1). It is necessary to preserve certain aspects of the source - in our case, the 3D geometry and any unaffected facial or clothing textures not specified by the edit. This dichotomy of "preservation" versus "change" presents an inherent conflict with the guidance direction.

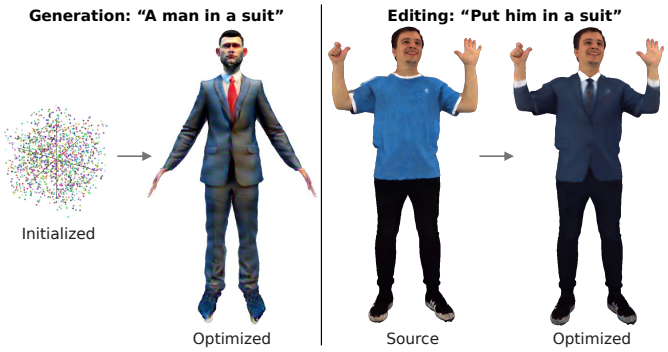

Fig. 1: 3D avatar generation (TADA [6]) vs. editing (Ours).

To further investigate, we break down SDS into individual terms. Previous works [18], [19] analogously showed how SDS terms affect mode-seeking and variance-reduction in the denoising process for generation. Such findings are not applicable for editing since they only consider a single condition - the generating text. The editing scenario has two conditions - input avatar and editing text. We also consider aspects unique to editing - specifically, how to preserve unedited features.

Our decomposition reveals that some terms are critical for structure formation in the early stages of denoising. While meaningful for generation, they are counterproductive to editing as they cause shifts away from the original structures. Similarly, other terms are beneficial only at later optimization stages. If all the terms are applied naively at all stages of denoising, which is the case in standard SDS, the terms will conflict and have counter-productive effects that lead to poor-quality edits. Our findings motivate us to design a customized SDS for editing (SDS-E). SDS-E distills guidance specifically for 3D editing. It introduces a temporal staging that selectively applies the SDS terms, allowing control over the terms' impact on the editing guidance.

To incorporate SDS-E for 3D human editing, we integrate it with a hybrid human representation [20] and form our *InstructHumans* framework. The hybrid representation uses local texture and geometry latent codes fixed to the human mesh vertices. Such a separation allows for localized texture edits while preserving the *animation* capability of edited avatars. For editing guidance, we require a 2D diffusion model conditioned on both the source image and text instruction. We adopt InstructPix2Pix [21]; the only diffusion model currently supporting dual conditioning, widely used in text-

Jiayin Zhu and Angela Yao, National University of Singapore, 117418, Singapore. (email: zhujiayin@u.nus.edu; ayao@comp.nus.edu.sg).
Linlin Yang, Communication University of China, Beijing, 100024, China. (email: lyang@cuc.edu.cn).

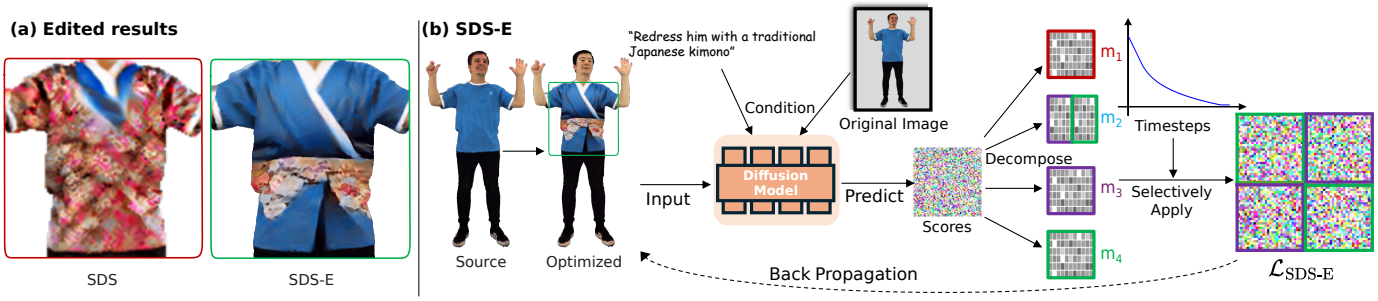

Fig. 2: **(a)** Edited results of SDS [13] vs. SDS-E. SDS results in a blurred avatar, with clothing deviating from the original features. **(b)** SDS-E edits a human avatar by querying the diffusion model conditioned on a text instruction and the original image, decomposes predicted scores into individual terms, and selectively applies them across timesteps. By controlling these terms, SDS-E provides cleaner guidance, leading to high-quality, faithful edits while maintaining consistency in unedited regions of the avatar.

based editing [8], [22]–[24]. Note, however, that SDS-E is general and can extend to other diffusion models that support dual conditioning as they emerge.

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 allocate camera viewpoints based on the need for guidance dynamically. This 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.

To summarize our contributions, we perform (1) an in-depth analysis of SDS for 3D editing and reveal the changing roles of the different SDS terms in the denoising process. Based on our analysis, we introduce (2) SDS for Editing with selective temporal staging of the SDS terms to distill effective editing guidance. We further introduce (3) a gradient-aware camera sampling that improves editing efficiency and specificity and (4) a smoothness regularizer that enhances the texture quality. Our resulting framework is efficient and flexible, yielding high-fidelity and faithful edits while maintaining consistency with the original avatar.

II. Related Works

Text-guided 3D Editing. Traditional 3D editing typically require explicit visual guidance, such as 3D cages [25] and masks [26]. Recent works [21], [27], [28] try to edit 3D objects via text guidance. One line of works adopts CLIP-based similarity to guide the 3D editing [7], [29], [30]. The outputs are, however, unrealistic and often require additional fine-tuning with *e.g*, GANs. Another line of work uses SDS [13] to distill information from 2D diffusion models. Using the predicted noise to guide 3D model updates is practical and efficient. As such, SDS has been adopted by many recent works for both text-to-3D generation [6], [31]–[33] and 3D editing [8], [9], [34]–[37], applied to both scenes and avatars. Classifier-free guidance [38] is typically adopted, and SDS [13] then distills the resulting conditional and unconditional signal pairs into 3D updates. IP2P [21] provides dual conditioning on the source image and text prompt; integrating such dual-conditioned editors within SDS for 3D optimization has seen limited exploration.

Improving SDS. [39] proposes non-increasing timestep sampling; [18] proposes using only classifier guidance. [19] and [40] take similar approaches as us and decompose SDS to improve stability. The findings of [18], [19], [40] improve SDS for generation, but are not applicable for editing. A key limitation is that it does account for the preservation of features of the source avatar during optimization. [34] tackles this difference in 2D image editing by additionally estimating the score of the original image-text pair. Other works [9], [35]–[37] directly apply SDS to 3D editing in its original form, without modifying the individual terms, which leads to suboptimal results.

3D Human Editing. Methods that *generate* controllable 3D humans [6], [7], [14], [41] primarily optimize a generative objective and are not designed to edit an existing personal avatar. Some works, like TADA [6] and HumanNorm [11], can appear "edit-like" by prompt changes, but they remain fundamentally generative, and depend on text-encoder familiarity with subjects, rather than editing a given personal avatar. *Editing*, in contrast, starts from an existing human and seeks text-aligned changes while preserving identity and structure across pose and animation. Several 3D editing approaches apply SDS in its original form [9], [35]–[37], which may be limiting for fidelity and animation consistency. In terms of scope, many works target specific regions such as the head [29], [42], [43] or upper body [37]. By representation, implicit NeRF-based editors, such as Instruct-NeRF2NeRF (IN2N) [8] and NeRF-Art [44] offer multi-view coherence but afford less direct control over topology or large deformations, while explicit mesh approaches [45], [46] provide surface-level edits under fixed topology. Hybrid representations seek to balance these trade-offs. EditableHumans [20] leverages a parametric human mesh model and merges it with the adaptability 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. A concurrent work, TEDRA [12], also examines text-based editing of animatable avatars; our scope differs by focusing on generic animatable avatars without per-subject model personalization.

III. Score Distillation Sampling for Editing

A. Preliminaries

InstructPix2Pix (IP2P) [21] 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:

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

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 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) [38] adjusts the diffusion model's adherence to specified conditions through hyperparameter tuning. For a 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:

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

Score Distillation Sampling (SDS) [13] leverages pre-trained 2D image diffusion models to facilitate 3D generation. By applying the 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 $\boldsymbol{\epsilon}$ correlates with the score function (the gradient of the log-density) of the perturbed data distribution [47]:

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

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 is given as:

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

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

Decomposition of SDS (Generation Setting). For single-condition (text-only) diffusion models used for 3D generation, SSD [19] reorganizes the SDS guidance into two subterms:

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}_\phi^{CFG}(\boldsymbol{x}_t; t, y) - \boldsymbol{\epsilon} = \omega \cdot \underbrace{\left(\hat{\boldsymbol{\epsilon}}_\phi(z_t, t, y) - \hat{\boldsymbol{\epsilon}}_\phi(z_t, t, \emptyset)\right)}_{\text{mode-disengaging}} \\ + \underbrace{\hat{\boldsymbol{\epsilon}}_\phi(z_t, t, y) - \boldsymbol{\epsilon}}_{\text{mode-seeking}}.\end{aligned} \tag{5}$$

From the score view (Eq. 3 with a single condition), the mode-disengaging term compares $\nabla \log p(z_t; t, y)$ against $\nabla \log p(z_t; t)$ and, at small t, maximizes $\frac{p(z;y)}{p(z)}$. This tends to decrease $p(z)$ and leads to saturation. SSD therefore recommends omitting the corresponding contribution at small timesteps; the mode-seeking term, when used alone with uniform t, can trap optimization in intermediate modes, contributing to over-smoothing. We recall these behaviors here as background; our analysis below adapts the decomposition and implications to the dual-conditional (text and image) editing setting.

3D Generation vs. Editing. Advancements in extending 2D diffusion models to 3D using techniques like SDS [13] have led to two main tasks: *3D generation* [7], [13], [15], [19], [36], [40] and *3D editing* [8], [34].

In *3D generation*, the goal is to synthesize a 3D representation Θ_{tgt} from an initial random state Θ_0, guided by a textual prompt y:

$$\Theta_{\text{tgt}} = \arg\min_\Theta \ \mathcal{L}_{\text{gen}}(\Theta, y), \text{ starting from } \Theta = \Theta_0, \tag{6}$$

where $\mathcal{L}_{\text{gen}}(\Theta, y)$ measures how well Θ aligns with y, such as using the SDS loss.

In contrast, *3D editing* transforms an existing 3D representation Θ_{ori} into Θ_{tgt}, based on an editing instruction y, while preserving original content. It utilizes images rendered from Θ_{ori}, denoted as $I = \mathcal{R}(\Theta_{\text{ori}})$, where $\mathcal{R}$ is the rendering function:

$$\Theta_{\text{tgt}} = \arg\min_\Theta \ \mathcal{L}_{\text{edit}}(\Theta, \mathcal{R}(\Theta_{\text{ori}}), y), \text{ starting from } \Theta = \Theta_{\text{ori}}, \tag{7}$$

where $\mathcal{L}_{\text{edit}}(\Theta, \mathcal{R}(\Theta_{\text{ori}}), y)$ measures alignment with y while retaining features of Θ_{ori}. Computing $\mathcal{L}_{\text{edit}}$ requires a diffusion model conditioned on both image and text; we adopt IP2P [21], which supports dual conditioning and is widely used [8], [22]–[24]. Yet, our analysis and method are general and can be extended to other diffusion models that support dual-conditioning if available.

Our work focuses on 3D editing. Despite sharing common aspects like being text-based and SDS decomposition, 3D generation works [7], [13], [15], [19], [36], [40] are not directly comparable due to the differences in the objectives outlined in Eq. 6 vs. Eq. 7. It is also worth noting that some works like TADA [6] and HumanNorm [11] extend their method to an edit-like setting. However, they are fundamentally generative, as they follow the same objective as Eq. 6. They 'edit' by altering keywords in the generation prompts (*e.g*, changing "Messi in a suit" to "Messi in a jacket.") This strategy relies on the text encoder's familiarity with specific subjects (in this case Messi), and does not extend to arbitrary individuals like our work.

B. Score Distillation Sampling for Editing

Timestep Sampling. Standard SDS samples timesteps t uniformly at random, but prior analysis shows that large timesteps are crucial for forming coarse features, while middle[1] and small timesteps are geared towards detailing [48]. In the context of 3D editing, a source 3D representation exists. Large timesteps serve little value and even risk disrupting the original structure, so we opt to fully remove large timesteps from the sampling.

Previously, [39] proposed a non-increasing timestep sampling strategy which they showed to be more informative for

[1]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 [19], as they only make a distinction between "small" and "large".

updating 3D neural fields. 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 Dual-Conditioned SDS. Our analysis begins with decomposing SDS for a dual-conditional diffusion model. This process aims to distinguish the editing directions influenced by the conditions and those influenced by the baseline (unconditioned) noise model. Adapting Eq. 5 to two conditions (y, I) by substituting Eq. 2 into Eq. 4 yields:

$$\begin{aligned}\hat{\boldsymbol{\epsilon}}_\phi^{CFG}(z_t;t,y,I)-\boldsymbol{\epsilon} &= (\omega_I-1)\cdot\underbrace{\left(\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,I)-\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,\emptyset)\right)}_{m_1}\\ &+\underbrace{\omega_t\cdot\left(\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,y,I)-\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,I)\right)+\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,I)-\boldsymbol{\epsilon}}_{m_2}.\end{aligned} \tag{8}$$

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 I only, and does not account for the text instruction. 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 both conditions of I and y.

Since m_2 involves both conditions, it can be further rearranged into a form analogous to Eq. 8:

$$\begin{aligned}m_2 = (\omega_t-1)\cdot&\underbrace{\left(\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,y,I)-\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,I)\right)}_{m_3}\\ &+\underbrace{\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,y,I)-\boldsymbol{\epsilon}}_{m_4}.\end{aligned} \tag{9}$$

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.

Analysis of the Baseline-Shift Term m_1. Replacing the unconditional reference in the SSD [19] argument with the image-conditioned reference yields:

$$\begin{aligned}m_1 &= \hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,I)-\hat{\boldsymbol{\epsilon}}_\phi(z_t,t,\emptyset,\emptyset)\\ &= -\sigma_t(\nabla_{z_t}\log p_\phi(z_t;t,I)-\nabla_{z_t}\log p_\phi(z_t;t)).\end{aligned} \tag{10}$$

The term m_1 causes shifts away from natural image distributions at small (and middle) timesteps. Specifically, when $t \to 0$, the distributions $p_\phi(z_t;t,I) \to p_\phi(z;t,I)$ and $p_\phi(z_t;t) \to p_\phi(z;t)$ lead to the maximization of the following term:

$$\frac{p_\phi(z_t;t,I)}{p_\phi(z_t;t)} \to \frac{p_\phi(z;t,I)}{p_\phi(z;t)} = \frac{p_\phi(z;I)}{p_\phi(z)}. \tag{11}$$

Therefore, as in SSD's single-condition case, we omit m_1 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 toward 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 $t \to 0$, the condition-divergence term maximizes:

$$\frac{p_\phi(z_t;t,y,I)}{p_\phi(z_t;t,I)} \to \frac{p_\phi(z;t,y,I)}{p_\phi(z;t,I)} = \frac{p_\phi(z;y,I)}{p_\phi(z;I)}. \tag{12}$$

TABLE I: 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

Analogous to m_1, we assume that a mode of $p(z;y,I)$ should also be a mode of $p(z;I)$. Yet Eq. 12 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 term sits in the denominator and gets minimized during the optimization process. This discourages convergence to any significant mode of the distribution $p_\phi(z;I)$, and distances the result from the original image.

While m_3 should presumably be removed at *small* timesteps, empirical evidence suggests that it significantly guides the text-conditioned mode and *improves* alignment with instructions. This allows for greater control over the trade-off between editing faithfulness and image fidelity—two competing factors central to editing tasks, which differ from single-objective generation. Therefore, we regard the inclusion of m_3 as a flexible design choice, 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 $-\boldsymbol{\epsilon}$ that counterbalances the variance introduced by the noise without altering the targeted mode:

$$\begin{aligned}m_4 &= \hat{\boldsymbol{\epsilon}}_\phi(z_t,t,y,I)-\boldsymbol{\epsilon}\\ &= -\sigma_t\nabla_{z_t}\log p_\phi(z_t;t,y,I)-\boldsymbol{\epsilon}.\end{aligned} \tag{13}$$

Applying m_4 alone may trap the model in intermediate modes. In particular, for large or middle timesteps, denoising is incomplete, so the peak of a joint probabilistic density with multiple modes is likely higher than that of any individual desired mode [19]. This issue diminishes in smaller timesteps, when the probabilistic density of the desired mode becomes higher, dominating the update direction. Yet in a uniformly random timestep sampling strategy, as in standard SDS, revisiting large or middle timesteps allows this issue to persist and disrupts convergence to any desired mode. This is the root cause for over-smoothing by SDS [13], [19].

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 is better. Using m_3 alone shifts the output 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 m_3 and m_4 with non-increasing timestep sampling [39] 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 Tab. I. Based on these findings, we present a

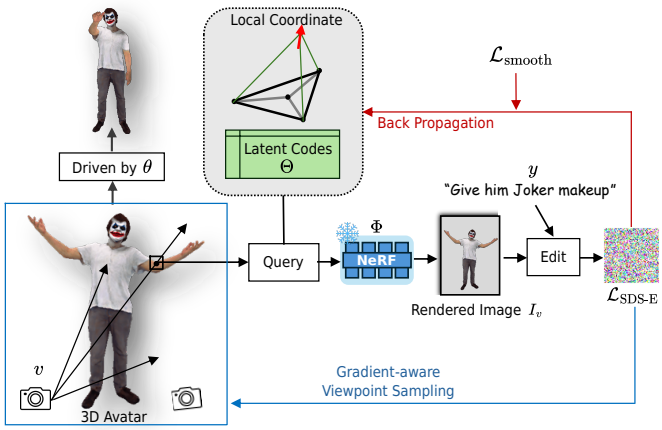

Fig. 3: **Instruction-driven 3D human editing pipeline.** Our pipeline optimizes a specific human subject's texture based on textual instructions. Images rendered through a conditional NeRF are edited by IP2P, with SDS-E used to distill the editing gradients and update the texture latent codes. The editing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

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:

$$\begin{aligned}\mathcal{L}_{\text{SDS-E}} = \omega_t \cdot (\hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, y, I) - \hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \emptyset, I)) \\ + \hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, \emptyset, I) - \boldsymbol{\epsilon}.\end{aligned} \tag{14}$$

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

$$\mathcal{L}'_{\text{SDS-E}} = \begin{cases} \mathcal{L}_{\text{SDS-E}} & \text{if } t > M \\ \hat{\boldsymbol{\epsilon}}_\phi(\boldsymbol{x}_t, t, y, I) - \boldsymbol{\epsilon} & \text{if } t \leq M, \end{cases} \tag{15}$$

where M is the threshold between small and middle timesteps. We empirically set M to 150 and limit middle timesteps to a maximum of 800 to exclude larger timesteps.

IV. InstructHumans Editing Pipeline

Hybrid 3D Human Representation. We adopt the hybrid 3D human representation proposed by EditableHumans [20]. It associates an explicit 3D human mesh model, SMPL-X [49], 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 $\boldsymbol{\Phi}$ is provided with 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 $\overrightarrow{n}$. Conditioned on the latent codes Θ, the implicit network provides the following:

$$\boldsymbol{\Phi}(\Theta, x_l, \overrightarrow{n}) = \Big(c(x_g), s(x_g)\Big). \tag{16}$$

The global to local coordinate transformation finds the nearest triangle on the body mesh of an input query point x_g and transforms the position into local triangle coordinates $x_l = (u, v, d)$, where d is the distance. $\overrightarrow{n}$ is calculated as the direction from the closest point on the mesh to the global position, providing auxiliary positional information. This transformation ensures that the NeRF accesses only local features, prevents it from memorizing global information, and disentangles local features for further editing.

Editing Pipeline (see Fig. 3). Starting with an input human subject with pre-trained latent codes Θ at mesh vertices, we optimize the latent codes to modify the human texture. At each iteration, an image I_v from a sampled camera view v is rendered with a conditional NeRF $\boldsymbol{\Phi}$. 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. III-B). The gradients, together with a smoothness regularizer $\mathcal{L}_{\text{smooth}}$ (see Eq. 17), are backpropagated for optimizing the latent codes. Gradient-aware viewpoint sampling dynamically adjusts the camera views based on the gradients (see Eq. 20). The edited human is easily drivable by changing the SMPL-X pose parameter θ.

Laplacian Smoothness Regularization. SDS-based 3D optimization often suffers from high-frequency noise due to three factors: (1) Randomly sampled camera views provide inconsistent supervision for overlapping 3D regions, leading to multi-view inconsistency [13]. (2) The distilled guidance is inherently noisy due to network instability [40] or architectural limitations [50]. (3) In discrete parameterizations (*e.g*, per-vertex latent codes [20] or 3D Gaussians [51]), SDS gradients update each location independently, amplifying noise and causing texture artifacts (Fig. 13).

We observe that guidance at a 3D location should be consistent across both views and adjacent neighbors. Given the explicit connectivity of the mesh, we introduce Laplacian latent smoothness to enforce spatial coherence in SDS updates. Inspired by Laplacian constraints in surface reconstruction [52], we define:

$$\mathcal{L}_{\text{smooth}} = \frac{1}{N} \sum_{i}^{N} \|(\overrightarrow{L}\overrightarrow{\Delta F})_i\|^2, \tag{17}$$

where N is number of vertices, $\overrightarrow{L}$ is the Laplacian matrix encoding connectivity between vertices, and $\overrightarrow{\Delta F}$ 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 local inconsistencies while preserving global details, improving texture coherence and convergence stability by reducing high-frequent flickering. The overall gradient is:

$$\nabla_\Theta(w_1 \mathcal{L}_{\text{smooth}} + \mathcal{L}_{\text{SDS-E}}). \tag{18}$$

Gradient-Aware Viewpoint Sampling. Another challenge is that edits are distributed unevenly across body regions depending on the text instructions. For example, "Put the person in a suit" targets clothing on the entire body, while "Give him Joker makeup" emphasizes the facial features. Uniform random viewpoint sampling misallocates editing effort. Some generation methods [6], [7], [14], [36] simply prioritize facial views, but this is not suitable for editing since not all prompts require significant modifications to the face. As such, we introduce the concept of editing strength, defined

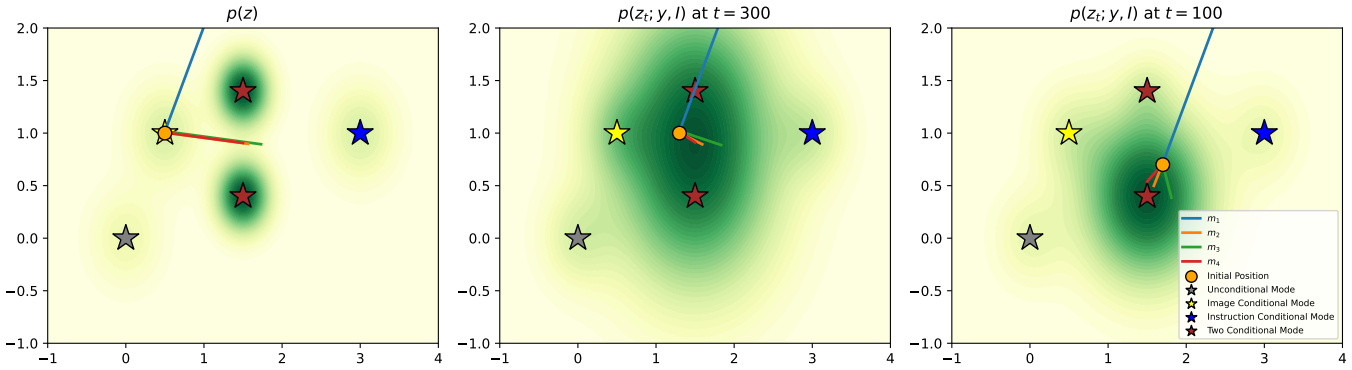

Fig. 4: **Visualizing the impact of SDS components m_1, m_2, m_3, and m_4 in a 2D toy example.** Each component, serving as an estimator, guides the optimization of $z = \theta \in \mathbb{R}^2$ within a Gaussian mixture model representing $p(z)$. The objective is to guide θ towards the two-conditional 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.

as the average gradient magnitude anchored at a region of vertices, and prioritize regions according to editing strength. 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. Note that this division is flexible and can be adapted for different applications. We then conduct one editing iteration with a batch of uniformly sampled $|V|$ views and calculate the average gradient magnitude w_r across the region r:

$$w_r = \frac{1}{|V|}\frac{1}{|S_r|}\sum_{v\in V}\sum_{i\in S_r}\|\nabla(i)\|, \tag{19}$$

where V denotes the set of sampled views, and S_r represents the set of vertices within region r. Using w_r as a normalization weight, we set $\mathcal{C}(r)$ as the the total number of views sampled for region r as follows:

$$\mathcal{C}(r) = \frac{w_r}{\sum_{r\in R} w_r}|V|, \tag{20}$$

to 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. It also accelerates the convergence rate, leading to a reduction in the overall number of editing iterations needed. Moreover, it improves the editing specificity on the desired regions, facilitating editing quality.

Selective Local Editing. As an alternative to gradient-aware viewpoint sampling, we can specify exact regions for editing by leveraging the controllability of 3D human models. Using the same body-region partition as above, a large language model (LLM) assistant maps an instruction to one or more target regions (represented by mesh vertices). SDS-E gradients are then applied only to the latent codes of the selected regions, leaving others unchanged. This option is useful when the intent is localized (*e.g*, accessories or facial attributes), whereas gradient-aware sampling adapts automatically when edits are distributed across multiple regions.

V. Experiments

A. Evaluating SDS Components via a Toy Example

We assess the behavior of the SDS components m_1, m_2, m_3, and m_4 (Sec. III-B) using a 2D toy example. In this experiment, we optimize $z = \theta \in \mathbb{R}^2$ over a Gaussian mixture model defined as $p(z) = 0.1\mathcal{N}([0,0]^\top, 0.1\mathbf{I}) + 0.15\mathcal{N}([3,1]^\top, 0.1\mathbf{I}) + 0.15\mathcal{N}([0.5,1]^\top, 0.1\mathbf{I}) + 0.3\mathcal{N}([1.5,1.4]^\top, 0.05\mathbf{I}) + 0.3\mathcal{N}([1.5,0.4]^\top, 0.05\mathbf{I})$. Here, the mode at $[0,0]^\top$ is unconditional, $[0.5,1]^\top$ is image conditional, $[3,1]^\top$ is text conditional, and $[1.5,1.4]^\top$ along with $[1.5,0.4]^\top$ denote the two-conditional modes. We simulate the 3D editing process by guiding θ toward the two-conditional modes using the estimators formulated in Eqs. 8 and 9. Figure 4 illustrates three learning phases that align with our analyses in Tab. I.

B. Experiment Settings

Comparison Methods. Our goal is to edit animatable 3D human textures based on text instructions. Directly comparable prior work is limited; we adapt related text-based methods for fair comparison, both qualitatively (Sec. V-C) and quantitatively (Sec. V-D).

We compare with IN2N [8], which edits NeRF texture/geometry but is non-animatable in its original setting; we therefore use it in static comparisons. We also compare with SDS-based methods [13], [19], [40]. Their focus on 3D generation differs from our 3D editing task, precluding a direct comparison in the original framework. To isolate the editing objective while ensuring animatability, we instantiate SDS, SSD, and NFSD within EditableHumans [20] and drive all edited avatars with identical pose/motion inputs. Specifically, we substitute SDS-E in our pipeline with SDS, SSD, and NFSD for fair comparison. Lastly, we compare with the avatar generation methods AvatarCLIP [7] and TADA [6]; since generation and editing methods have different objectives and evaluation metrics (Sec. III-A), we adapt our edit prompts into generation prompts and assess the resulting avatars' quality

Fig. 5: **Qualitative comparison with IN2N**. IN2N struggles with content preservation and texture quality.

Fig. 6: **Qualitative comparison with AvatarCLIP and TADA**. Ours achieves superior photorealistic quality.

Fig. 7: **Qualitative visualization of our results.**

relative to the prompts. We also visualize TADA under the same animation drivers for qualitative comparison.

Implementation Details. We optimize for 1000 steps and sample 50 camera views per step at 400×400 resolution. Image conditioning follows IP2P [21]: the source image is encoded by the Stable Diffusion VAE, and the resulting latent is concatenated with the noisy latent along the channel dimension after modifying the UNet's first convolution to accept the additional channels. The pre-trained NeRF and initial human latent codes are identical to EditableHumans [20]. Concretely, the NeRF uses two MLP decoders (SDF and RGB), each with 4 linear layers and 128 hidden units, and the latent is a 32-D per-vertex codebook over 10,475 mesh vertices. The smoothness loss weight is $w_1 = 300$. Gradient-aware viewpoint sampling is computed once at the first editing iteration (negligible overhead, < 0.1 s) and then fixed. With batch size 1, the per-iteration time on an NVIDIA A40 is 25.2 s (total ≈ 7 h for 1000 steps) with peak memory ≈ 13 GB.

C. Qualitative Experiment

We compare with IN2N and SDS-based methods using human subjects from CustomHumans [20]. As shown in Fig. 5 and Fig. 8, our approach outperforms existing methods in producing high-quality textures that better follow editing instructions while retaining human identity. Unlike IN2N, our avatars remain fully animatable. For avatar generation baselines, Fig. 6 shows that our method yields more photorealistic results compared to AvatarCLIP and TADA. Fig. 9 shows animated comparisons, where methods are evaluated under identical motions. More qualitative results are provided in Fig. 7.

Selective Local Editing. We also evaluate the alternative selective local editing. While the primary pipeline already achieves robust localized edits without affecting unselected areas, this option enhances precision when exact regions are specified. As shown in Fig. 10, this method preserves the original clothing and identity cues, offering finer control over local edits compared to running the main pipeline alone, *e.g*, the instruction "Put on a pair of sunglasses," where the head region is edited and the rest of the body remains unchanged; for "Redress him with a traditional Japanese kimono," only the clothes region is edited.

Diversity. Most SDS-based methods exhibit limited diversity due to the inherent constraints of the distillation process [40]. While our work prioritizes editing quality, diversity can also

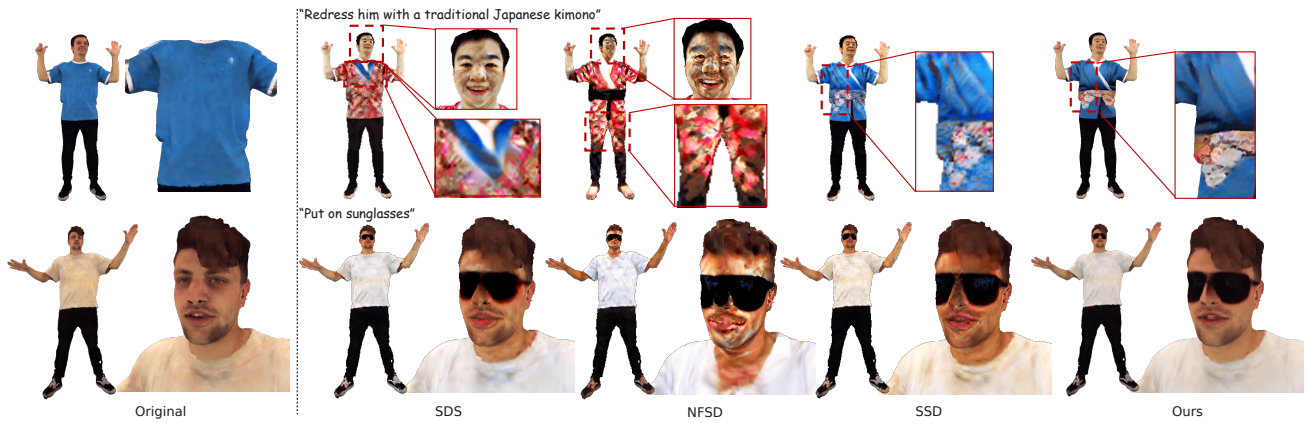

Fig. 8: **Qualitative comparison with SDS, NFSD, and SSD**. Our method excels in both texture quality and adherence to the original avatars and editing instructions, whereas the others produce textures that are spotty, blurry, and over-saturated.

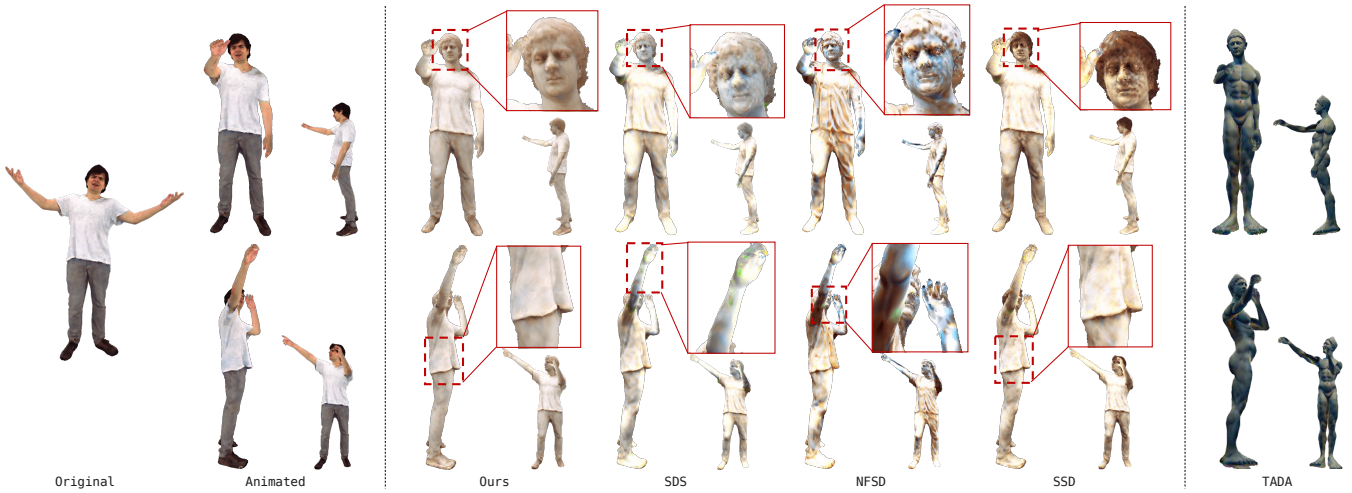

Fig. 9: **Animation comparison** of edited or generated avatars. We show the prompt "A marble statue" (full-body texture case). SDS, NFSD, and SSD are instantiated in the same framework [20] as our method, making all edited avatars animatable; all results are driven by identical motions. These baselines exhibit spotty or over-saturated textures under animation. TADA [6] produces animatable avatars from prompts but does not edit a given source avatar.

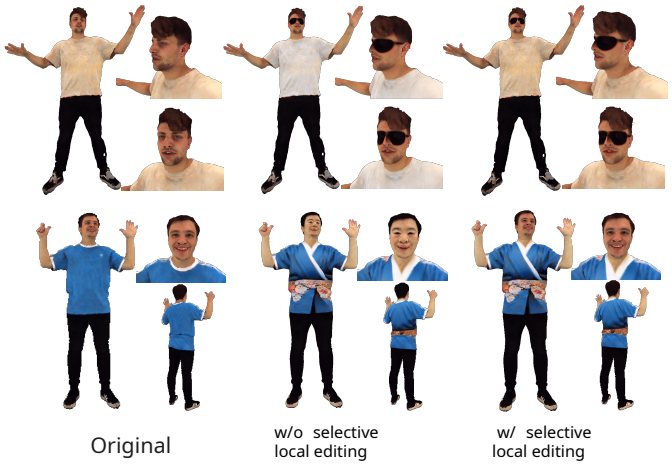

Fig. 10: **Selective local editing**. Examples with head-only ("Put on a pair of sunglasses") and clothes-only ("Redress him with a traditional Japanese kimono") updates. This approach preserves the original clothing, improving upon the results from our primary pipeline.

be improved through a simple filtering strategy that amplifies edits with specific attributes, such as color (see Fig. 11).

Fig. 11: **Diversified results.**

D. Quantitative Experiment

Metrics. Following IN2N, we measure *CLIP text-image directional similarity* (CLIP-Direc↑) for text alignment. To evaluate structural and semantic fidelity to the original avatar, we also evaluate *CLIP image similarity* (CLIP-Img↑) between the

rendered images of the edited and original avatars. Both metrics, in conjunction, balance maintaining consistency with the original image and achieving the intended editing outcomes. We also evaluate LPIPS↓ [53] compared with the original images for texture quality, and CLIP-Score↑ between the result images and text prompts for generation coherence.

Quantitative comparison with SOTA. We follow IN2N's 10 full-body edits and add 3 localized edits (hair, eyes, mouth) for fine-grained evaluation. As shown in Tab. II, we outperform IN2N and existing SDS-based methods across all metrics, achieving the best balance between text adherence, identity preservation, and visual quality. For avatar generation, Tab. III shows our higher CLIP-Score, indicating better semantic alignment than AvatarCLIP and TADA.

TABLE II: **Quantitative comparison with text-based editing methods.** SSD has slightly higher CLIP-Img at the cost of CLIP-Direc. Ours balances the best between adherence to instructions and preservation of original avatar features, while delivering superior texture quality.

	Ours	IN2N	SDS	SSD	NFSD
CLIP-Direc↑	**0.160**	0.117	0.144	0.141	0.149
CLIP-Img↑	**0.863**	0.686	0.852	**0.868**	0.783
LPIPS↓	**0.047**	0.216	0.070	0.057	0.109

TABLE III: **Quantitative comparison with avatar generation methods.** Ours achieves the highest CLIP-Score, showing superior semantic alignment.

	Ours	AvatarCLIP	TADA
CLIP-Score↑	**0.231**	0.223	0.230

TABLE IV: **User study.** Ours is significantly preferred across all three metrics. "No Pref." indicates no preference.

	Ours	NFSD	SDS	SSD	No Pref.
Visual Quality	**57.82%**	10.26%	10.64%	19.42%	1.86%
Image Consistency	**58.14%**	7.50%	9.10%	22.63%	2.63%
Text Consistency	**53.59%**	15.96%	9.23%	19.23%	1.99%

User Study. Editing quality is subjective. We conducted a user study on Mechanical Turk, where 315 participants provided 1560 responses on all editing comparisons, considering overall quality, instruction adherence, and fidelity to the original images. As summarized in Tab. IV, our method is preferred across all metrics.

E. Ablation Studies

Ablation setup. Unless otherwise noted, all ablations are run with both text prompts and image conditioning (the source avatar). We vary one component at a time and keep all other settings fixed (same seed, prompt, and avatar).

Smoothness regularizer & Viewpoint sampling. Fig. 13 (a) shows how removing the regularizer leads to uneven textures and unrealistic spots, especially on the face. Omitting the gradient-aware viewpoint sampling leads to an undesired shift in the edited face due to an imprecise editing focus. Excluding it also increases the runtime 5-fold.

Gradient-Aware Viewpoint Sampling. Figure 14 compares our gradient-aware sampling to a uniform baseline. Our method assigns region-specific weights w_r (see Eq. 19) and view counts $\mathcal{C}(r)$ (see Eq. 20), as detailed in Tab. V. To validate the efficiency of the gradient-aware viewpoint sampling, we investigate its runtime. It is computed once at the first iteration, adding negligible overhead (< 0.1 s) because it reuses gradients already computed in that iteration; the operation is dominated by simple reductions. Rather than the per-step cost, it reduces the overall iteration numbers until convergence, leading to $2\times$ fewer steps in overall runtime. Our sampling strategy successfully assigns weights to specific regions based on the editing instructions, ensuring more precise and efficient editing.

Timestep Division. We empirically determined the thresholds M and L for dividing timesteps into small, medium, and large stages (see Tab. I). Fig. 12 illustrates the impact of these divisions on editing performance.

Other design choices. Fig. 13 (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. III-B. An alternative approach that excludes term m_4 during middle timesteps leads to deviations from desired image guidance. Excluding image conditioning collapses to low-frequency color fill. 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.

TABLE V: **Region weights w_r and view counts $\mathcal{C}(r)$ from our gradient-aware sampling.** The total number of views is $|V| = 50\,000$. For the "kimono" instruction, body regions receive higher weights and more views to match its focus on clothing; for "clown," more views are assigned to the head for intensive editing.

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

F. Applications

InstructHumans and SDS-E enable various applications. Leveraging the explicit controllability of the human representation [20], InstructHumans allows *animation* of edited avatars by adjusting SMPL-X mesh's pose parameters (see Fig. 15). Furthermore, SDS-E can be applied to broader pipelines, such as *editing texture and geometry together* on a 3D Gaussian splatting pipeline [22] (see Fig. 16).

G. Limitations.

Fig. 17 depicts typical failure cases. Our framework employs IP2P [21] to guide texture edits and thus inherits its behavior. A commonly observed issue with IP2P is that, when prompts specify colors, it tends to apply a global color cast that

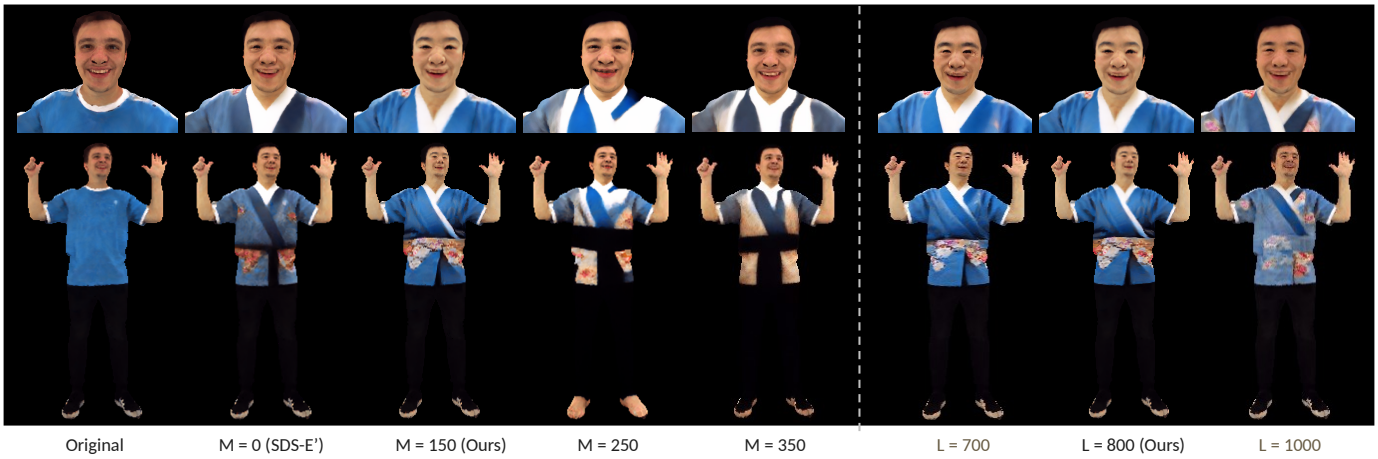

Fig. 12: **Ablation study on timestep division thresholds. Left:** Varying M, the threshold between small and medium timesteps. Larger M values lead to fewer details and possible editing failures. At $M = 150$, results are most faithful to the editing instructions with high texture quality. When $M = 0$ (equivalent to SDS-E′), results maintain similar quality but emphasize coherence with the original avatar. **Right:** Varying L, the threshold between medium and large timesteps. Larger L values can destroy original features (e.g., clothing color), consistent with our analysis in Sec. III-B. Setting $L = 800$ gives the best results, while $L = 700$ causes a slight drop in texture quality, noticeable in areas like the eyes.

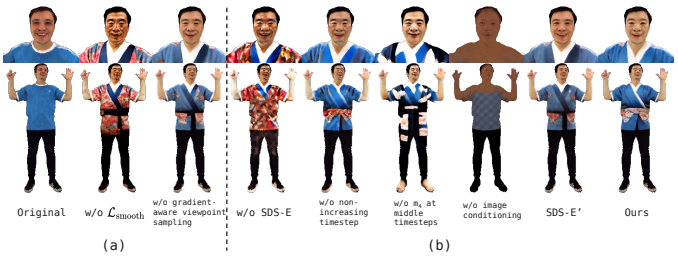

Fig. 13: **Ablation study on key components.**

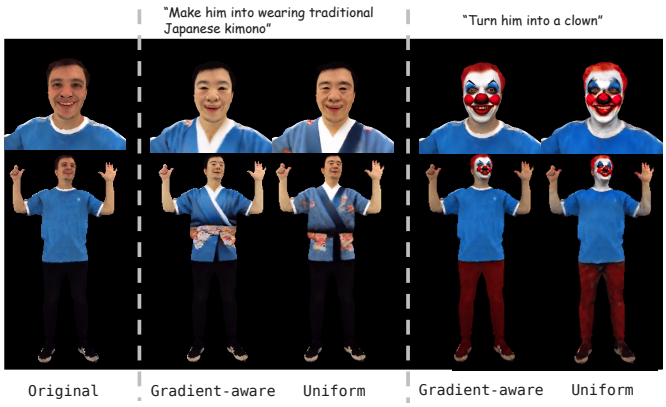

Fig. 14: **Ablation study on gradient-aware viewpoint sampling.** Our approach efficiently focuses editing on desired regions, whereas uniform sampling yields inaccurate and blurry results.

spills beyond the intended region, as illustrated in Fig. 17(a). Our selective local editing mitigates this to some extent, but a more thorough remedy is to adopt stronger 2D editing models as they become available. In addition, our framework builds upon the hybrid human representation of [20], which may produce artifacts at joint areas under extreme poses, as seen in Fig. 17(b). As noted in [20], increasing mesh resolution

Fig. 15: **Animated edited avatars**. The animations demonstrate the seamless integration of editing while preserving natural motion.

and training on larger datasets are potential remedies. Our framework can directly benefit from such improvements.

VI. Conclusion

This work presents a method for 3D human texture editing guided by textual instructions, achieving a balance between intuitive editing capabilities and animation flexibility. 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 latent smoothness 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.

Fig. 16: Applying SDS-E on the GaussianEditor [22] pipeline, which enables simultaneous texture and geometry editing.

Fig. 17: **Failure cases.** (a) Global color leakage when IP2P is prompted with explicit colors. (b) Joint artifacts under extreme poses due to representation limits.

REFERENCES

[1] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *Proc. Int. Conf. Mach. Learn.* PMLR, 2021, pp. 8748–8763. 1

[2] D. Liu, J. Zhu, X. Fang, Z. Xiong, H. Wang, R. Li, and P. Zhou, "Conditional video diffusion network for fine-grained temporal sentence grounding," *IEEE Trans. Multimedia*, vol. 26, pp. 5461–5476, 2024. 1

[3] C. Zhang, W. Yang, X. Li, and H. Han, "Mmginpainting: Multi-modality guided image inpainting based on diffusion models," *IEEE Trans. Multimedia*, vol. 26, pp. 8811–8823, 2024. 1

[4] S. Zhou, D. Guo, J. Li, X. Yang, and M. Wang, "Exploring sparse spatial relation in graph inference for text-based vqa," *IEEE Trans. on Image Process.*, vol. 32, pp. 5060–5074, 2023. 1

[5] K. Liu, F. Xue, D. Guo, P. Sun, S. Qian, and R. Hong, "Multimodal graph contrastive learning for multimedia-based recommendation," *IEEE Trans. Multimedia*, vol. 25, pp. 9343–9355, 2023. 1

[6] T. Liao, H. Yi, Y. Xiu, J. Tang, Y. Huang, J. Thies, and M. J. Black, "TADA! Text to Animatable Digital Avatars," in *Proc. Int. Conf. 3D Vis.*, 2024. 1, 2, 3, 5, 6, 8

[7] F. Hong, M. Zhang, L. Pan, Z. Cai, L. Yang, and Z. Liu, "Avatarclip: Zero-shot text-driven generation and animation of 3d avatars," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 1–19, 2022. 1, 2, 3, 5, 6

[8] A. Haque, M. Tancik, A. Efros, A. Holynski, and A. Kanazawa, "Instruct-nerf2nerf: Editing 3d scenes with instructions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023. 1, 2, 3, 6

[9] M. Mendiratta, X. Pan, M. Elgharib, K. Teotia, M. B. R, A. Tewari, V. Golyanik, A. Kortylewski, and C. Theobalt, "Avatarstudio: Text-driven editing of 3d dynamic human head avatars," *ACM Trans. Graph.*, vol. 42, no. 6, dec 2023. 1, 2

[10] Y.-H. Kwon, J. H. Yoon, and M.-G. Park, "Text2avatar: Articulated 3d avatar creation with text instructions," *IEEE Trans. Multimedia*, pp. 1–12, 2025. 1

[11] X. Huang, R. Shao, Q. Zhang, H. Zhang, Y. Feng, Y. Liu, and Q. Wang, "Humannorm: Learning normal diffusion model for high-quality and realistic 3d human generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2024, pp. 4568–4577. 1, 2, 3

[12] B. Sunagad, H. Zhu, M. Mendiratta, A. Kortylewski, C. Theobalt, and M. Habermann, "TEDRA: Text-based editing of dynamic and photoreal actors," in *Proc. Int. Conf. 3D Vis.*, 2025. 1, 2

[13] B. Poole, A. Jain, J. T. Barron, and B. Mildenhall, "Dreamfusion: Text-to-3d using 2d diffusion," in *Proc. Int. Conf. Learn. Represent.*, 2023. 1, 2, 3, 4, 5, 6

[14] N. Kolotouros, T. Alldieck, A. Zanfir, E. Bazavan, M. Fieraru, and C. Sminchisescu, "Dreamhuman: Animatable 3d avatars from text," *Adv. Neural Inf. Process. Syst.*, vol. 36, 2024. 1, 2, 5

[15] Z. Wang, C. Lu, Y. Wang, F. Bao, C. LI, H. Su, and J. Zhu, "ProlificDreamer: High-Fidelity and Diverse Text-to-3D Generation with Variational Score Distillation," in *Adv. Neural Inf. Process. Syst.*, A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36. Curran Associates, Inc., 2023, pp. 8406–8441. 1, 3

[16] C.-H. Lin, J. Gao, L. Tang, T. Takikawa, X. Zeng, X. Huang, K. Kreis, S. Fidler, M.-Y. Liu, and T.-Y. Lin, "Magic3d: High-resolution text-to-3d content creation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 300–309. 1

[17] N. Ruiz, Y. Li, V. Jampani, Y. Pritch, M. Rubinstein, and K. Aberman, "Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2023, pp. 22 500–22 510. 1

[18] X. Yu, Y.-C. Guo, Y. Li, D. Liang, S.-H. Zhang, and X. Qi, "Text-to-3D with Classifier Score Distillation," *arXiv.org*, Oct. 2023. 1, 2

[19] B. Tang, J. Wang, Z. Wu, and L. Zhang, "Stable score distillation for high-quality 3d generation," *arXiv preprint arXiv:2312.09305*, no. arXiv:2312.09305, 2024. 1, 2, 3, 4, 6

[20] H.-I. Ho, L. Xue, J. Song, and O. Hilliges, "Learning locally editable virtual humans," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 21 024–21 035. 1, 2, 5, 6, 7, 8, 9, 10

[21] T. Brooks, A. Holynski, and A. A. Efros, "Instructpix2pix: Learning to follow image editing instructions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2023, pp. 18 392–18 402. 1, 2, 3, 7, 9

[22] Y. Chen, Z. Chen, C. Zhang, F. Wang, X. Yang, Y. Wang, Z. Cai, L. Yang, H. Liu, and G. Lin, "Gaussianeditor: Swift and controllable 3d editing with gaussian splatting," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 21 476–21 485, 2023. 2, 3, 9, 11

[23] M. Chen, J. Xie, I. Laina, and A. Vedaldi, "Shap-editor: Instruction-guided latent 3d editing in seconds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 26 456–26 466. 2, 3

[24] S. Li, B. Zeng, Y. Feng, S. Gao, X. Liu, J. Liu, L. Li, X. Tang, Y. Hu, J. Liu, and B. Zhang, "Zone: Zero-shot instruction-guided local editing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2024, pp. 6254–6263. 2, 3

[25] C. Jambon, B. Kerbl, G. Kopanas, S. Diolatzis, G. Drettakis, and T. Leimkühler, "Nerfshop: Interactive editing of neural radiance fields," *Proc. ACM Comput. Graph. Interact. Tech.*, vol. 6, no. 1, 2023. 2

[26] J. Sun, X. Wang, Y. Shi, L. Wang, J. Wang, and Y. Liu, "Ide-3d: Interactive disentangled editing for high-resolution 3d-aware portrait synthesis," *ACM Trans. Graph.*, vol. 41, no. 6, pp. 1–10, 2022. 2

[27] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 10 684–10 695. 2

[28] C. Saharia, W. Chan, S. Saxena, L. Li, J. Whang, E. L. Denton, K. Ghasemipour, R. Gontijo Lopes, B. Karagol Ayan, T. Salimans *et al.*, "Photorealistic text-to-image diffusion models with deep language understanding," *Adv. Neural Inf. Process. Syst.*, vol. 35, pp. 36 479–36 494, 2022. 2

[29] S. Aneja, J. Thies, A. Dai, and M. Nießner, "Clipface: Text-guided editing of textured 3d morphable models," in *ACM SIGGRAPH Conf. Proc.*, 2023, pp. 1–11. 2

[30] A. Jain, B. Mildenhall, J. T. Barron, P. Abbeel, and B. Poole, "Zero-shot text-guided object generation with dream fields," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 867–876. 2

[31] W. Li, R. Chen, X. Chen, and P. Tan, "Sweetdreamer: Aligning geometric priors in 2d diffusion for consistent text-to-3d," in *Proc. Int. Conf. Learn. Represent.*, 2024. 2

[32] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," in *Proc. Int. Conf. Learn. Represent.*, 2024. 2
[33] H. Yi, Z. Zheng, X. Xu, and T.-s. Chua, "Progressive text-to-3d generation for automatic 3d prototyping," *arXiv preprint arXiv:2309.14600*, 2023. 2
[34] A. Hertz, K. Aberman, and D. Cohen-Or, "Delta denoising score," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, October 2023, pp. 2328–2337. 2, 3
[35] H. Kamata, Y. Sakuma, A. Hayakawa, M. Ishii, and T. Narihira, "Instruct 3d-to-3d: Text instruction guided 3d-to-3d conversion," *arXiv preprint arXiv:2303.15780*, 2023. 2
[36] J. Yu, H. Zhu, L. Jiang, C. C. Loy, W. Cai, and W. Wu, "Painthuman: Towards high-fidelity text-to-3d human texturing via denoised score distillation," in *Proc. AAAI Conf. Artif. Intell.*, vol. 38, no. 7, 2024, pp. 6800–6807. 2, 3, 5
[37] H. Zhang, Y. Feng, P. Kulits, Y. Wen, J. Thies, and M. J. Black, "TECA: Text-Guided Generation and Editing of Compositional 3D Avatars," in *International Conference on 3D Vision (3DV)*, 2024. 2
[38] J. Ho and T. Salimans, "Classifier-free diffusion guidance," *arXiv preprint arXiv:2207.12598*, 2022. 2, 3
[39] Y. Huang, J. Wang, Y. Shi, B. Tang, X. Qi, and L. Zhang, "Dreamtime: An improved optimization strategy for diffusion-guided 3d generation," in *Proc. Int. Conf. Learn. Represent.*, 2024. 2, 3, 4
[40] O. Katzir, O. Patashnik, D. Cohen-Or, and D. Lischinski, "Noise-free score distillation," in *Proc. Int. Conf. Learn. Represent.*, 2024. 2, 3, 5, 6, 7
[41] Y. Cao, Y.-P. Cao, K. Han, Y. Shan, and K.-Y. K. Wong, "Dreamavatar: Text-and-shape guided 3d human avatar generation via diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 958–968. 2
[42] S. Hwang, J. Hyung, D. Kim, M. Kim, and J. Choo, "Faceclipnerf: Text-driven 3d face manipulation using deformable neural radiance fields," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.* Los Alamitos, CA, USA: IEEE Computer Society, oct 2023, pp. 3446–3456. 2
[43] H. Wu, M. Zhao, Z. Hu, C. Fan, L. Li, W. Chen, R. Zhao, and X. Yu, "Ice: Interactive 3d game character facial editing via dialogue," *IEEE Trans. Multimedia*, pp. 1–14, 2025. 2
[44] C. Wang, R. Jiang, M. Chai, M. He, D. Chen, and J. Liao, "Nerf-art: Text-driven neural radiance fields stylization," *IEEE Trans. Vis. Comput. Graph.*, 2023. 2
[45] O. Michel, R. Bar-On, R. Liu, S. Benaim, and R. Hanocka, "Text2mesh: Text-driven neural stylization for meshes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, June 2022, pp. 13 492–13 502. 2
[46] B. Kim, P. Kwon, K. Lee, M. Lee, S. Han, D. Kim, and H. Joo, "Chupa: Carving 3d clothed humans from skinned shape priors using 2d diffusion probabilistic models," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, October 2023, pp. 15 965–15 976. 2
[47] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 6840–6851, 2020. 3
[48] J. Choi, J. Lee, C. Shin, S. Kim, H. Kim, and S. Yoon, "Perception prioritized training of diffusion models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 11 472–11 481. 3
[49] G. Pavlakos, V. Choutas, N. Ghorbani, T. Bolkart, A. A. A. Osman, D. Tzionas, and M. J. Black, "Expressive body capture: 3D hands, face, and body from a single image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10 975–10 985. 5
[50] Z. Pan, J. Lu, X. Zhu, and L. Zhang, "Enhancing high-resolution 3d generation through pixel-wise gradient clipping," in *Proc. Int. Conf. Learn. Represent.*, 2024. 5
[51] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Trans. Graph.*, vol. 42, no. 4, July 2023. 5
[52] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel, "Laplacian surface editing," in *Proc. Eurographics/ACM SIGGRAPH Symp. Geom. Process.*, 2004, pp. 175–184. 5
[53] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018. 9