T3HG-EDITOR: TEXT-DRIVEN 3D HUMAN GARMENT EDITING WITH BODY PRIORS EMBEDDED IN SMPL-X

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite recent advances in 3D Gaussian Editing (3DGE), text-driven 3D human garment editing remains underexplored. Existing methods edit multi-view renderings with 2D techniques and update the underlying Gaussians, often causing geometric distortions and cross-view inconsistencies. SMPL eXpressive (SMPL-X) provides rich human geometry and joint priors to address these challenges. Motivated by this insight, we propose a text-driven 3D human garment editor termed T3HG-Editor, which delivers high-fidelity and garment consistent results by leveraging geometry and joint priors embedded in SMPL-X. Specifically, T3HG-Editor contains three stages, namely obtainment of editable Gaussians, garment consistent editing, and Gaussian updating with overflow pruning. The obtainment of editable Gaussians begins with seeding Gaussians along SMPL-X normals to generate sufficient near surface Gaussians, followed by a 2D mask constraint that precisely localizes the target Gaussians to be edited. The garment consistent editing aggregates tokens corresponding to the same SMPL-X vertex across multiple views and propagates them to their original views, enforcing garment consistency without requiring additional training. Gaussian updating with overflow pruning employs a Signed Distance Function (SDF) defined on SMPL-X to construct a human distance field, which is then integrated with a 2D semantic mask to prune overflowing Gaussians, thus preventing contamination of non-target regions. Experiments on multiple subjects and diverse garment types demonstrate that T3HG-Editor outperforms state-of-the-art methods in both editing quality and garment consistency.

1 INTRODUCTION

3D human garment editing holds significant value in meeting personalized demands across various domains like AIGC asset reuse, film production, e-commerce display, and more Wang et al. (2025); Bao et al. (2025); Chen & Wang (2024). Recent advances in 2D diffusion models Chan et al. (2024); Li et al. (2025); Ho et al. (2020) and 3D Gaussian splatting (3DGS) Kerbl et al. (2023) have greatly advanced the development of 3D Gaussian Editing (3DGE). However, 3D human garment editing, which requires fine-grained and garment consistent modifications without disrupting background or human structure, remains underexplored due to the diversity of garment types and the complexity of their geometries Sun et al. (2025); Chen et al. (2024a). Existing 3DGE works typically apply 2D editing techniques to multi-view rendered images and subsequently update 3D Gaussians based on the modified images Vachha & Haque (2024); Karim et al. (2024). A straightforward extension of such methods to human garments suffers from low-fidelity outcomes,

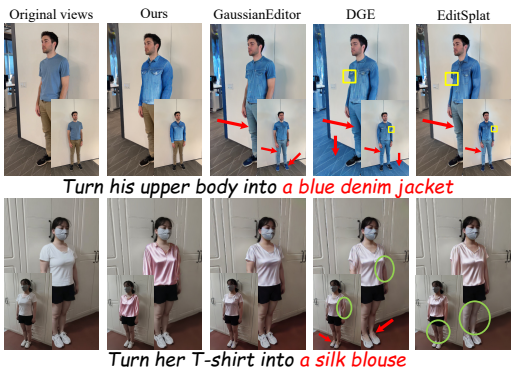


Figure 1: Failure cases of existing methods. Top: background and pants color changes (red arrows) and retained short-sleeve details (yellow box). Bottom: garment inconsistencies around the sleeves and shoes (green ellipses).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

primarily on account of two key issues. As shown in Figure 1, on the one hand, complex non-rigid deformations in garment regions may lead to inaccurate shape adaptation, while unintended Gaussian contamination may occur in non-garment regions. On the other hand, garment diversity and independent editing across different views may result in inconsistencies in garment shape, texture, and appearance.

The SMPL-X (Skinned Multi-Person Linear Model with eXpressive hands and face) model Pavlakos et al. (2019); Loper et al. (2015), which is widely adopted in the virtual human domain, encapsulates rich prior information including human body geometry, pose joints, and garment-body interaction constraints Lin et al. (2025); Svitov et al. (2024); Shao et al. (2025). The priors embedded in SMPL-X enable restricting edits within target garment regions and establishing consistency constraints across views. Motivated by this insight, we propose a text-driven 3D human garment editor termed T3HG-Editor. T3HG-Editor leverages SMPL-X's body geometry prior to constrain Gaussian position adjustment and target-non-target region decoupling, and utilizes the pose joint prior to achieve cross-view consistent feature fusion. T3HG-Editor contains three stages, namely obtainment of editable Gaussians, garment consistent editing, and Gaussian updating with overflow pruning. The obtainment of editable Gaussians begins by seeding 3D Gaussians along the normals of the SMPL-X model's surface, thus anchoring the Gaussians to the geometric structure of the human body. Then, 2D mask-based filtering eliminates Gaussians outside the target garment region to obtain accurately positioned Gaussians. For garment consistent editing, T3HG-Editor aggregates feature tokens associated with the same SMPL-X vertex across views. By tying token aggregation to the unified geometric framework of SMPL-X for information sharing, garment consistent editing aligns Gaussian representations across different views without requiring additional model training. Gaussian updating with overflow pruning follows a dual-constraint paradigm. One constraint is a 3D human distance field, primarily derived from a Signed Distance Function (SDF) defined on the SMPL-X model, while the other is 2D semantic masks. This dual-constraint paradigm confines Gaussians to the intended regions, thus preventing the contamination of unintended regions. In summary, the contributions of this work are as follows:

- We propose a text-driven 3D human garment editor (T3HG-Editor) based on the SMPL-X model. To the best of our knowledge, T3HG-Editor is an early systematic exploration of text-driven 3DGS editing specifically designed for human garment editing.
- Based on SMPL-X's body geometric prior, we propose Gaussian seeding along the surface normals and target-non-target region decoupling, solving shape deviations in the target garment region and unintended Gaussian contamination in non-target regions.
- Based on the pose joint prior, we design a garment consistent editing scheme by aggregating tokens corresponding to the same SMPL-X vertex across key views.

2 Related work

3D Human Gaussian Splatting with SMPL-X. By providing a parametric skeleton, vertices, and meshes, SMPL-X has been widely adopted in 3D human Gaussian modeling Prospero et al. (2025); Hu et al. (2024b). HumanGaussian Liu et al. (2024) initializes Gaussian positions with SMPL-X meshes and employs a dual-branch SDS scheme for Gaussian updating. DreamWaltz-G Huang et al. (2025) uses the SMPL-X skeleton as a conditioning signal and SMPL-based Gaussian initialization to reduce multi-face artifacts and implausible poses. ToMiE Zhan et al. (2024) identifies garment regions not covered by the base model and grows external data-driven joints for decoupled animation of loose garments and hand-held objects. GaussianAvatar Hu et al. (2024a) binds animatable Gaussians to the SMPL-X skeleton, applies forward LBS for reposing, and jointly refines SMPL-X pose parameters for monocular human reconstruction. Across these works, SMPL-X plays a central role in 3D human generation Pan et al. (2024), reconstruction Li et al. (2024a), and animation Kocabas et al. (2024) with Gaussian representations. In this paper, we adopt SMPL-X priors to facilitate garment editing of 3D human Gaussians.

Text-driven 3D Editing. The scarcity of large-scale 3D training data has motivated most text-driven 3D editing methods to leverage pretrained 2D diffusion models Couairon et al. (2022); Li et al. (2024b); Narasimhaswamy et al. (2024); Chen et al. (2024b). InstructPix2Pix (IP2P) Brooks et al. (2023), conditioned on an input image and text instruction, is a widely adopted editing backbone. Instruct-NeRF2NeRF Haque et al. (2023) iteratively edits rendered views with IP2P and uses them

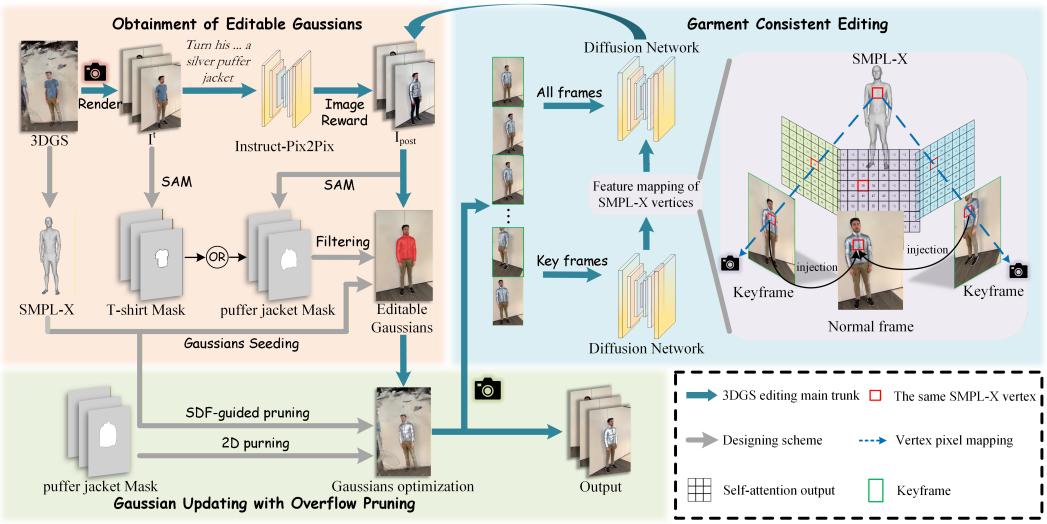


Figure 2: The overall framework of **T3HG-Editor**, which consists of three components: (1) Obtainment of Editable Gaussians; (2) Garment Consistent Editing; and (3) Gaussian Updating with Overflow Pruning.

to update a NeRF representation. However, the implicit representation of NeRF makes local editing difficult to control and computationally expensive. With the emergence of 3D Gaussian Splatting, GaussianEditor Chen et al. (2024d) back-projects 2D editing masks to label target Gaussians and employs anchor constraints to regulate Gaussian densification. Nevertheless, independently editing individual views often causes cross-view inconsistencies. Subsequent methods therefore introduce geometric or feature-level constraints. View-consistent 3D editing Wang et al. (2024b) exploits cross-view geometric correspondence, while DGE Chen et al. (2024c) jointly attends to key views and propagates edited features to other views via epipolar constraints. EditSplat Lee et al. (2025) incorporates multi-view depth information and attention-guided Gaussian pruning, whereas GaussCtrl Wu et al. (2024) jointly edits multiple views with depth conditioning and attention alignment. Other approaches align instruction-related regions with Gaussian representations to support localized updates Wang et al. (2024a).

The aforementioned methods ensure garment consistency of 3D scenes or localization of editing regions, but dedicated studies on text-driven garment editing for 3DGS humans remain limited. Garment editing may cause expansion or shrinkage, where existing methods modify Gaussians in non-editing regions or fail to adapt the editing area to garment changes. In contrast, T3HG-Editor leverages SMPL-X to improve editable Gaussian localization, garment consistency, and Gaussian updating within intended regions.

3 METHOD

As illustrated in Figure 2, T3HG-Editor comprises three stages: obtainment of editable Gaussians, garment consistent editing, and Gaussian updating with overflow pruning. The first stage combines SMPL-X-based Gaussian seeding with mask-based filtering to obtain editable Gaussians. The second stage establishes cross-view correspondence through SMPL-X-vertex guided attention and feature propagation. The third stage updates the editable Gaussians while applying 2D and 3D pruning to prevent contamination of unintended regions. The following subsections describe these stages.

3.1 OBTAINMENT OF EDITABLE GAUSSIANS

Existing 3D editing methods obtain editable Gaussians under the guidance of masks or attention weights. However, such schemes are unsuitable for garment editing because garment coverage may change after editing. Restricting Gaussian updating to the pre-edit region overlooks newly extended

parts, whereas focusing on the post-edit region ignores contracted parts. To address these limitations, T3HG-Editor seeds Gaussians along the SMPL-X normals to generate sufficient Gaussians proximate to the body surface, followed by 2D mask filtering to precisely position the target Gaussians.

SMPL-X-based Gaussian Seeding. This operation can generate a dense set of editable Gaussians based on an SMPL-X mesh, effectively mitigating shape deviations that arise after editing operations. As shown in Figure 3, given an SMPL-X mesh, the SMPL-X-based Gaussian Seeding operation begins by uniformly sampling J points within a distance $D_{\max}$ along the normal direction of each vertex. Then, each sampled point is used to anchor a plane whose normal is aligned to the corresponding vertex normal. Afterwards, for each plane, a set of seeded points is generated through random sampling centered at the corresponding anchor point and confined within radius r. Formally, the position of the seeded points is expressed as follows:

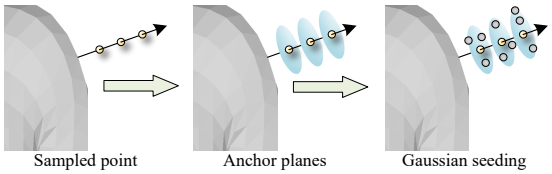


Figure 3: SMPL-X-based Gaussian seeding. The yellow circles indicate the points sampled along the normals, while the gray circles are the positions of the seeded Gaussians.

$$\mathbf{x}_{i,j} = \mathbf{v}_i + s_j \cdot \mathbf{n}_i + u \cdot \mathbf{t}_{i,j}, \quad u \sim \mathcal{U}[0, r], \tag{1}$$

$$s_j = \frac{j-1}{J-1} \cdot D_{\max}, \quad j = 1, 2, \ldots, J. \tag{2}$$

Here, $\mathbf{n}_i$ represents the normal of vertex $\mathbf{v}_i$, $\mathbf{t}_{i,j}$ denotes a random unit vector, u denotes a random number sampled from the interval $[0, r]$. Finally, each seeded point is assigned appearance and geometric properties through KNN Cover & Hart (1967) attribute aggregation, thus defining the initial state of the seeded Gaussians. The color parameter $\mathbf{c}_{i,j}$, scaling factor $\mathbf{s}_{i,j}$ and opacity $\alpha_{i,j}$ are initialized as the average of the corresponding attributes from the several nearest Gaussians, while the rotation parameter $\mathbf{q}_{i,j}$ inherits directly from the single nearest Gaussian in the original model.

Mask-based filtering. For the seeded Gaussians, pre-edit and post-edit garment changes are jointly considered to accurately localize the editable Gaussians. Specifically, the 3DGS scene is rendered from T views to obtain $\{I^t_{\text{pre}}\}^T_{t=1}$, and Segment Anything Model (SAM) Kirillov et al. (2023) is applied to derive the pre-edit garment mask $\mathcal{M}_{\text{pre}}$. Then, $\{I^t_{\text{pre}}\}^T_{t=1}$ together with the text prompt are fed into InstructPix2Pix (IP2P) Brooks et al. (2023) to generate the edited images. IP2P generates diverse results across viewpoints, and such diversity facilitates selecting the outputs that align with the textual description, thereby establishing the garment's overall shape. Accordingly, ImageReward Xu et al. (2023) is used to evaluate the editing results, and the top N images are selected as supervision for subsequent Gaussian updates, denoted by $\{I^n_{\text{post}}\}^N_{n=1}$. Subsequently, these images are fed into SAM to obtain the post-edit garment mask $\mathcal{M}_{\text{post}}$. Finally, the filtering mask is obtained by $\mathcal{M}_{\text{filter}} = \mathcal{M}_{\text{pre}} \vee \mathcal{M}_{\text{post}}$, which accommodates garment expansion or shrinkage, as illustrated in Figure 4. The filtered Gaussians are assigned semantic masks that are retained during subsequent splitting or duplication.

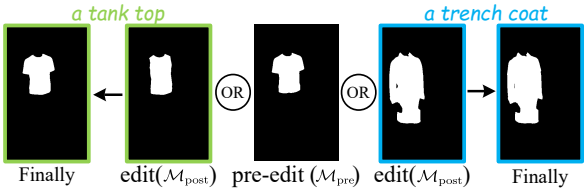


Figure 4: Obtaining target masks for garments that shrink or expand after editing.

According to the back-projection formulation in GaussianEditor Chen et al. (2024d), the computed Gaussian weights determine whether a Gaussian is assigned a semantic mask, thereby integrating garment shapes from all edited views. To obtain a unified 2D garment representation, the labeled Gaussians are separately rendered to the selected camera views, and the resulting 2D mask $\mathcal{M}_{\text{final}}$ is used both as the loss region and as the supervision region for subsequent pruning. The editing loss is defined as follows:

$$\mathcal{L}_{\text{edit}} = \min \sum_{n=1}^{N} \left\| I^n_{\text{post}}(\mathcal{M}_{\text{final}}) - I'_n(\mathcal{M}_{\text{final}}) \right\|. \tag{3}$$

Here, I'_n denotes the rendered image of the current 3D Gaussians at the n-th view. Through the above mechanism, T3HG-Editor can precisely obtain the editable Gaussians, thereby establishing a reliable foundation for subsequent multi-view garment consistency enhancement and pruning.

3.2 GARMENT CONSISTENT EDITING

The independent single-view editing inevitably introduces garment inconsistencies in texture and fine details. Therefore, a garment consistency enhancement mechanism is designed comprising two core modules, SMPL-X-vertex guided attention and SMPL-X-vertex guided feature propagation, achieving unified semantic and visual coherence across views.

In the context of human garment editing, the fitted SMPL-X mesh carries effective structural priors for establishing cross-view correspondences. Given the camera parameters, SMPL-X vertices are rendered into each view and every pixel patch is annotated with its associated vertex index. This yields a bidirectional lookup between pixel patches and mesh vertices, furnishing a unified and queryable coordinate reference for subsequent cross-view feature alignment.

SMPL-X-vertex Guided Attention. Unlike editing a single image, T3HG-Editor jointly edits multi-view images by concatenating the tokens corresponding to the same SMPL-X vertex before applying attention, thereby enhancing consistency. Specifically, to reduce computational cost, one key view out of every five views is selected in order and all key views are fed into the U-Net simultaneously. Then, within the U-Net's self-attention module, for each key view m with queries $\{Q_m\}_{m=1}^{M}$, keys $\{K_m\}_{m=1}^{M}$, and values $\{V_m\}_{m=1}^{M}$, the tokens corresponding to the same vertex l across key views are determined from the vertex–pixel index list. Next, these tokens for the same vertex are concatenated and processed through self-attention, enabling each view to attend to the features of other views corresponding to the same SMPL-X vertex. The formulation is as follows:

$$\mathrm{STAttn}\big(Q^l, K^l, m\big) = \mathrm{Softmax}\left(\frac{Q^l_m \cdot [K^l_1, \ldots, K^l_M]}{\sqrt{d}}\right), \tag{4}$$

$$\Phi^l_m = \mathrm{STAttn}\big(Q^l, K^l, m\big) \cdot [V^l_1, \ldots, V^l_M]. \tag{5}$$

Here, d denotes the dimension used for computing queries and keys. $l \in \{1, \cdots, L\}$ denotes the same vertex across key views. After computing cross-view attention for each vertex, the output features are returned to their positions. Finally, tokens that cannot be matched to the same vertex across views, along with background tokens, are concatenated and processed through self-attention across key views.

By performing the vertex-level cross-view attention only within the selected key views and processing vertex groups in batches, the memory footprint and computational overhead can be effectively controlled. This operation explicitly enhances the semantic and textural consistency of the same vertex across key views, providing a well-aligned foundation for subsequent feature propagation.

SMPL-X-vertex Guided Feature Propagation. For the remaining views, the features from the key views are propagated according to the principle of sharing the same vertex, thereby ensuring consistency across views. The vertex-pixel index list is used to query and compute the tokens corresponding to the same vertex in both normal views and key views. Then, the features Φ^l_m obtained from Eq. 5 are fused through weighted aggregation and injected into the remaining views as self-attention outputs, as formulated below:

$$\psi^l_n = w_1 \cdot \Phi^l_1 + \cdots + w_M \cdot \Phi^l_M. \tag{6}$$

The weight w_m of each key view is determined by normalizing the number of vertices. It shares with the n-th normal view, where more shared vertices imply closer views and thus higher weights. For the remaining tokens without corresponding vertices, the cosine similarity is computed between the tokens of the normal and key views, and the most similar tokens are selected as learning targets.

Through the above mechanism, SMPL-X establishes information association and mutual learning across views, thereby improving consistency. In addition, some vertices may be occluded or invisible in certain views but remain visible in others. During cross-view self-attention and feature learning, to avoid losing information from invisible vertices on the human body, the tokens of the same vertex are processed together with their neighboring features for attention and feature learning.

3.3 Gaussian Updating with Overflow Pruning

During Gaussian updating, some Gaussians may deviate from the target regions, causing contamination of non-target regions. To address this issue, 2D and 3D pruning are designed by combining a Signed Distance Function (SDF)-based human distance field with 2D masks, thereby pruning Gaussians that overflow beyond the target editing regions.

2D Pruning. After periodic updating, the Gaussians labeled with the semantic mask are rendered from the selected camera views, and the resulting current Gaussian mask map is compared with $\mathcal{M}_{\text{final}}$ to identify out-of-bound pixel areas. These pixel areas are then back-projected into 3D space to track the Gaussians that have moved beyond the target region for pruning. It should be noted that this process applies only to the labeled Gaussians.



Figure 5: The original pants and edited human are segmented to identify pruning regions, followed by Gaussian pruning and updating to obtain the final rendering.

Furthermore, in the case of garment shrinkage after editing, for example, "*Make him in a shorts*". The Gaussians corresponding to the lower part of the pants gradually become transparent or are pruned. However, relying solely on gradient-based updating may introduce artifacts. To accelerate the pruning of redundant Gaussians, a human segmentation mask is generated for the edited human, denoted as $\mathcal{M}_{\text{hum}}$ and it no longer includes the lower part of the pants. In contrast, $\mathcal{M}_{\text{final}}$ corresponds to the mask obtained by performing a logical OR between the pants and shorts masks, which still retains the overall shape of the pants. By comparing $\mathcal{M}_{\text{hum}}$ and $\mathcal{M}_{\text{final}}$, the Gaussians associated with the missing lower pants can be efficiently localized and pruned. As illustrated in Figure 5.

3D Pruning. Due to viewpoint limitations and occlusion, some Gaussians may drift away from the human body while their 2D projections remain within $\mathcal{M}_{\text{final}}$. To address this, an SMPL-X-derived SDF constraint is imposed in 3D space. For Gaussians generated through splitting or duplication, the signed distance from each centroid to the SMPL-X surface is computed, and those exceeding a predefined threshold β are pruned. This constraint removes out-of-body drift that is difficult to detect in 2D projections.

Dual-domain pruning can significantly reduce the risk of noisy Gaussians contaminating non-target regions. However, given the large number of Gaussians, missed drifting Gaussians may still cause local contamination that propagates "from points to surfaces". To prevent such cases, a loss is introduced in non-target regions to keep their appearance consistent with the original scene, as formulated below:

$$\mathcal{L} = \mathcal{L}_{\text{edit}} + \min \sum_{n=1}^{N} \left\| I^{n}_{\text{pre}}(\overline{\mathcal{M}_{\text{final}}}) - I'_{n}(\overline{\mathcal{M}_{\text{final}}}) \right\|. \tag{7}$$

Here, $\overline{\mathcal{M}_{\text{final}}}$ denotes the complement of $\mathcal{M}_{\text{final}}$, representing the non-target regions. Gradients are computed between the rendered images I'_n of the current 3DGS and the original I^n_{pre} in the unintended regions, thereby preserving these regions unchanged. This is also facilitated by the accurate localization of the 2D mask.

4 Experiments

4.1 Experimental Setup

Implementation Details. T3HG-Editor adopts 3D Gaussian Splatting (3DGS) to represent humans and employs InstructPix2Pix Brooks et al. (2023) as the image editing backbone. For each scene, the number of selected views N is set to 20, and the Gaussian updating is performed for 1200

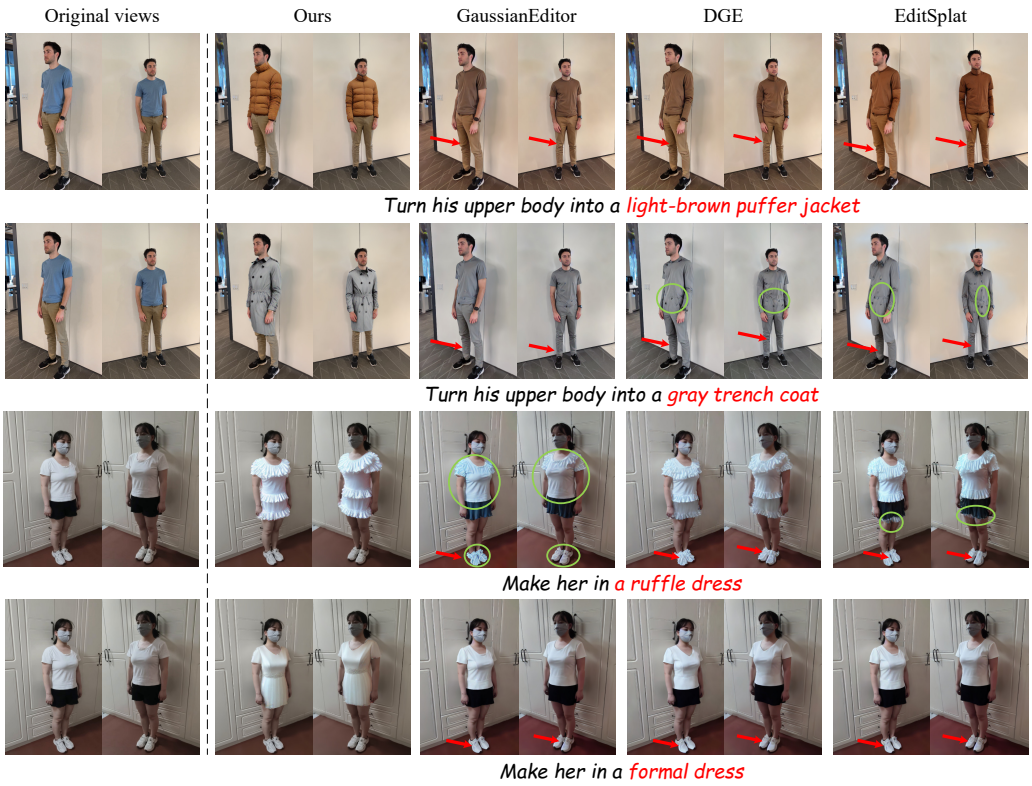


Figure 6: Qualitative comparison of garment editing. The regions that require attention have been annotated in the figure. The red arrows indicate unintended region changes, while the green ellipses highlight garment inconsistencies.

iterations. Following the recommendations of IN2N Haque et al. (2023), both LPIPS and L1 losses are utilized to update the 3DGS human scenes. The maximum seeding distance $D_{\max}$ and the SDF-based distance threshold β are determined according to the size and thickness of the target garment to be edited, and are typically set within the range of 0 to $0.1\,\mathrm{m}$ (see the Appendix for detailed parameter selection criteria and settings). All experiments are conducted on a workstation equipped with two NVIDIA GeForce RTX 4080 SUPER GPU (32 GB VRAM), and the editing process takes approximately 7 to 9 minutes.

Compared Methods. T3HG-Editor is compared with state-of-the-art 3D scene editing approaches, including GaussianEditor Chen et al. (2024d), DGE Chen et al. (2024c), and EditSplat Lee et al. (2025). All methods adopt InstructPix2Pix as the image editor and have been applied to garment edits in human scenes. GaussianEditor is a representative approach in this field. DGE and EditSplat introduce consistency enhancing strategies, making them competitive methods for comparison.

Evaluation Metrics. The quantitative evaluation follows the criteria introduced in He et al. (2024); Wang et al. (2024b). Similarity to the target text is measured using CLIP Radford et al. (2021), while CLIP directional similarity evaluates the alignment between the image and text editing directions. All methods are assessed across ten text prompts. Editing quality and cross-view consistency are further evaluated using PSNR, SSIM, and LPIPS. To more clearly reveal the advantages of the proposed designs, these metrics are computed separately for the target and non-target regions. We also analyze the evolution of CLIP-based metrics during optimization on a representative garment editing case (see the Appendix for details).

4.2 Qualitative Comparisons

Figure 6 presents qualitative results of garment editing. As we can see, unintended changes occur in the regions indicated by the red arrows for GaussianEditor, DGE, and EditSplat. In addition,

for the text prompts "*Turn his upper body into a gray trench coat*", DGE and EditSplat exhibit garment inconsistencies in sleeve or button editing, as indicated by the green ellipses. In contrast, T3HG-Editor confines modifications strictly to the target garment regions by jointly considering masks before and after editing, thereby avoiding contamination of irrelevant areas. Moreover, when editing a T-shirt into a long-sleeved coat, the three baseline methods still leave visible short-sleeve contours. For large garments, such as the prompt "*Turn his upper body into a light-brown puffer jacket*", they do not capture the bulky texture characteristic of puffer jackets. These limitations stem from the insufficient distribution of Gaussians in garment-related regions such as sleeves, hems, and other structural components in the original 3D scenes. For female subject, similar limitations remain in the compared methods, and they fail to produce edits consistent with textual instructions and often alter Gaussian features in unintended regions, as indicated by the red arrows. As shown in Figure 1, under the prompt "*Turn her T-shirt into a silk blouse*", DGE yields inconsistent sleeves between the left and right arms, while EditSplat produces mismatched colors on the legs. With the prompt "*Make her in a formal dress*", all three baselines fail to generate results resembling a formal dress, producing outputs largely similar to the original garment.

By contrast, T3HG-Editor leverages SMPL-X normals to seed Gaussians, effectively supplementing missing regions and precisely obtaining Gaussians in the target region using 2D masks, thereby validating the effectiveness of the proposed obtainment of editable Gaussians scheme. Moreover, T3HG-Editor achieves garment consistent editing through cross-view attention among tokens corresponding to the same SMPL-X vertices, while pruning overflowing Gaussians to prevent contamination of unintended regions. For a fairer comparison, we additionally compare T3HG-Editor with GaussianEditor and DGE when both methods are provided with target masks. More comparison results and qualitative examples are provided in the Appendix.

4.3 Quantitative Comparisons

Table 1 presents the CLIP similarity, CLIP directional similarity, and average running time of T3HG-Editor and the compared approaches across two scenarios and ten text prompts. The results show that T3HG-Editor outperforms state-of-the-art approaches in both CLIP similarity and CLIP directional similarity. T3HG-Editor requires time for Gaussian seeding and cross-view attention computation among tokens corresponding to the same SMPL-X vertices within the U-Net. Therefore, it takes more time than DGE, while remaining competitive with the other methods.

Table 1: Quantitative comparisons. The best results are highlighted in bold.

Method	CLIP_{sim}	CLIP_{dir}	Time (min)
GaussianEditor (CVPR 2024)	0.2524	0.0889	$\sim$8
DGE (ECCV 2024)	0.2573	0.1445	$\sim$**4**
EditSplat (CVPR 2025)	0.2554	0.1674	$\sim$24
T3HG-Editor (Ours)	**0.2660**	**0.2394**	$\sim$8

Table 2 reports the average PSNR, SSIM, and LPIPS over ten text instructions. For each method, 16 of the 20 edited views are used for 3DGS reconstruction, while the remaining four are used for testing. The rendered test views are compared with the edited images in the target garment regions and with the original images in the non-target regions. T3HG-Editor achieves the best SSIM and LPIPS in the target garment regions, demonstrating improved multi-view garment consistency. Its PSNR is slightly lower than that of DGE because DGE fails on some instructions and produces less detailed textures. In the non-target garment regions, T3HG-Editor performs best on all metrics, benefiting from accurate obtainment of editable Gaussians and overflow pruning.

Table 2: Quantitative comparisons on the target and non-target garment regions.

Method	PSNR↑	SSIM↑	LPIPS↓
Target Garment Region			
GaussianEditor (CVPR 2024)	19.4225	0.5290	0.2904
DGE (ECCV 2024)	**22.0834**	0.5783	0.2085
EditSplat (CVPR 2025)	20.9711	0.5731	0.2157
T3HG-Editor (Ours)	21.2198	**0.5796**	**0.1738**
Non-Target Garment Region			
GaussianEditor (CVPR 2024)	20.0683	0.7963	0.2279
DGE (ECCV 2024)	23.3919	0.8701	0.1644
EditSplat (CVPR 2025)	19.7180	0.7392	0.3143
T3HG-Editor (Ours)	**28.8880**	**0.9300**	**0.0828**

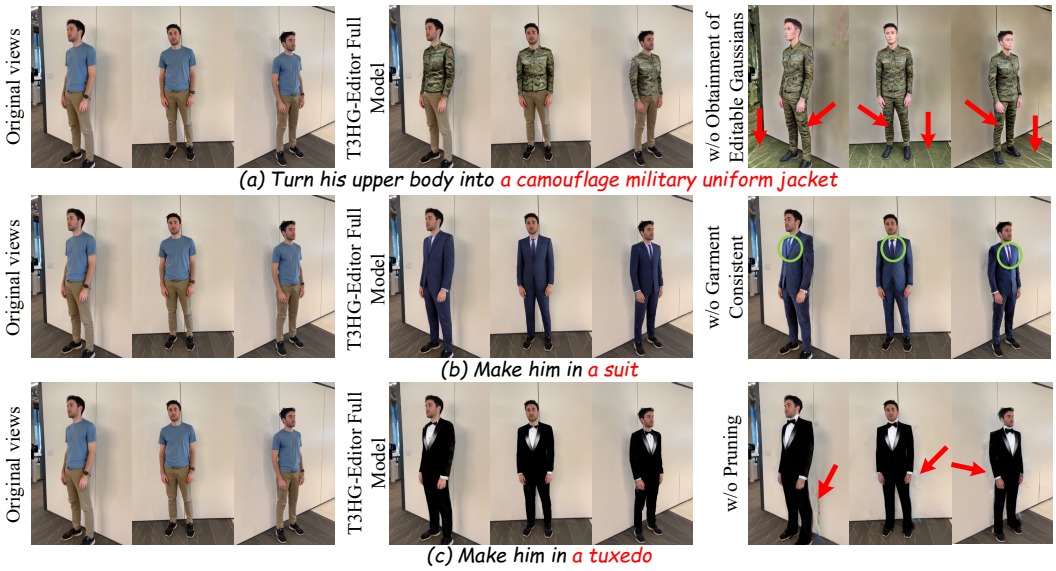


Figure 7: **Ablation Study.** (a)–(c) illustrate the effects of editable Gaussian obtainment, garment consistency enforcement, and Gaussian overflow pruning, respectively. The red arrows indicate unintended changes, while the green ellipses highlight garment inconsistencies.

4.4 Ablation Study

To validate the proposed obtainment of editable Gaussians, garment consistent editing, and pruning of Gaussian overflow, ablation studies are conducted on each mechanism (see the Appendix for quantitative comparisons).

First, as shown in Figure 7(a), removing the obtainment of editable Gaussians causes facial distortion and background corruption. The edited garment retains a T-shirt contour because the original scene lacks Gaussians around the sleeves, demonstrating the necessity of Gaussian seeding and the effectiveness of editable Gaussian obtainment.

Second, Figure 7(b) demonstrates the contribution of garment consistent editing. Without this mechanism, the tie exhibits inconsistent colors across views, and artifacts appear on the suit, as highlighted by green ellipses. Applying this strategy produces a uniform tie color and fewer artifacts, confirming its effectiveness in enhancing garment consistency.

Finally, Figure 7(c) verifies the effectiveness of Gaussian overflow pruning. Without pruning, numerous noisy Gaussians and artifacts appear around the human body, as indicated by the red arrows, whereas the proposed pruning effectively suppresses these issues.

5 Conclusion

To address complex shape transformations and garment inconsistencies in 3DGS-based garment editing, we propose T3HG-Editor, a text-driven 3D human garment editor. It seeds Gaussians along SMPL-X surface normals and filters them using fused pre- and post-edit masks to localize editable regions. Cross-view attention aggregates features associated with the same SMPL-X vertices, while an SDF-based distance field and 2D masks prune overflowing Gaussians. Experiments on garments show that T3HG-Editor outperforms state-of-the-art methods qualitatively and quantitatively.

T3HG-Editor relies on IP2P, whose ability to handle complex text instructions remains limited, while simultaneous multi-view editing increases runtime. Future work will focus on accelerating editing and supporting more diverse garment modifications with complex instructions.

AI USE STATEMENT

Generative AI tools were used only to assist with manuscript preparation, including language refinement, sentence restructuring, and LaTeX formatting. They were not used to formulate the proposed method, design the experiments, generate or alter experimental results, implement the core algorithms, or draw scientific conclusions. All AI-assisted content was carefully checked and revised by the authors, who take full responsibility for the final manuscript.

ETHICS STATEMENT

This work evaluates 3D human garment editing using one publicly available human scene and one scene captured specifically for research purposes. The participant involved in the recorded scene provided consent for the collection and use of the data in research and publication. T3HG-Editor is developed for controllable garment editing of 3D human representations and is not intended to identify individuals or infer sensitive personal information. As appearance-editing technologies could be misused to generate misleading or unauthorized visual content, their use should respect participant consent, privacy, and applicable intellectual-property regulations.

REPRODUCIBILITY STATEMENT

The paper specifies the architecture and optimization procedures of T3HG-Editor in the Method section, while the Experiments section reports the editing backbone, number of selected views, optimization iterations, loss functions, and hardware configuration. Additional implementation details are provided in the Appendix, including dataset construction and preprocessing, view selection, hyperparameter ranges, and sensitivity analyses. Together, these descriptions are intended to provide sufficient information for reproducing the main experimental settings and evaluating the proposed method.

REFERENCES

Zhenyu Bao, Guibiao Liao, Kaichen Zhou, Kanglin Liu, Qing Li, and Guoping Qiu. LoopSparseGS: Loop-based sparse-view friendly Gaussian splatting. *IEEE Trans. Image Process.*, 34:3889–3902, 2025. doi: 10.1109/TIP.2025.3574929.

Tim Brooks, Aleksander Holynski, and Alexei A Efros. InstructPix2Pix: Learning to follow image editing instructions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 18392–18402, 2023.

Cheuk-Yiu Chan, Wan-Chi Siu, Yuk-Hee Chan, and H. Anthony Chan. AnlightenDiff: Anchoring diffusion probabilistic model on low light image enhancement. *IEEE Trans. Image Process.*, 33: 6324–6339, 2024. doi: 10.1109/TIP.2024.3486610.

Haodong Chen, Yongle Huang, Haojian Huang, Xiangsheng Ge, and Dian Shao. GaussianVTON: 3D human virtual try-on via multi-stage gaussian splatting editing with image prompting, 2024a.

Jun-Kun Chen and Yu-Xiong Wang. Proedit: Simple progression is all you need for high-quality 3D scene editing. *Adv. Neural Inf. Process. Syst.*, 37:4934–4955, 2024.

Jun-Kun Chen, Samuel Rota Bulo, Norman Müller, Lorenzo Porzi, Peter Kontschieder, and Yu-Xiong Wang. ConsistDreamer: 3D-consistent 2D diffusion for high-fidelity scene editing. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 21071–21080, 2024b.

Minghao Chen, Iro Laina, and Andrea Vedaldi. DGE: Direct gaussian 3D editing by consistent multi-view editing. In *Eur. Conf. Comput. Vis.*, pp. 74–92. Springer, 2024c.

Yiwen Chen, Zilong Chen, Chi Zhang, Feng Wang, Xiaofeng Yang, Yikai Wang, Zhongang Cai, Lei Yang, Huaping Liu, and Guosheng Lin. GaussianEditor: Swift and controllable 3D editing with gaussian splatting. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 21476–21485, 2024d.

Guillaume Couairon, Jakob Verbeek, Holger Schwenk, and Matthieu Cord. DiffEdit: Diffusion-based semantic image editing with mask guidance, 2022.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory*, 13 (1):21–27, 1967.

Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-NeRF2NeRF: Editing 3D scenes with instructions. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 19740–19750, 2023.

Runze He, Shaofei Huang, Xuecheng Nie, Tianrui Hui, Luoqi Liu, Jiao Dai, Jizhong Han, Guanbin Li, and Si Liu. Customize your NeRF: Adaptive source driven 3D scene editing via local-global iterative training. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6966–6975, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Adv. Neural Inf. Process. Syst.*, 33:6840–6851, 2020.

Liangxiao Hu, Hongwen Zhang, Yuxiang Zhang, Boyao Zhou, Boning Liu, Shengping Zhang, and Liqiang Nie. GaussianAvatar: Towards realistic human avatar modeling from a single video via animatable 3D gaussians. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 634–644, 2024a.

Shoukang Hu, Tao Hu, and Ziwei Liu. GauHuman: Articulated gaussian splatting from monocular human videos. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 20418–20431, 2024b.

Yukun Huang, Jianan Wang, Ailing Zeng, Zheng-Jun Zha, Lei Zhang, and Xihui Liu. DreamWaltz-G: Expressive 3D gaussian avatars from skeleton-guided 2D diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2025.

Nazmul Karim, Hasan Iqbal, Umar Khalid, Chen Chen, and Jing Hua. Free-Editor: Zero-shot text-driven 3D scene editing. In *Eur. Conf. Comput. Vis.*, pp. 436–453. Springer, 2024.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3D Gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139:1–139:14, 2023.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, pp. 4015–4026, 2023.

Muhammed Kocabas, Jen-Hao Rick Chang, James Gabriel, Oncel Tuzel, and Anurag Ranjan. HUGS: Human Gaussian splats. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 505–515, 2024.

Dong In Lee, Hyeongcheol Park, Jiyoung Seo, Eunbyung Park, Hyunje Park, Ha Dam Baek, Sangheon Shin, Sangmin Kim, and Sangpil Kim. EditSplat: Multi-view fusion and attention-guided optimization for view-consistent 3D scene editing with 3D gaussian splatting. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 11135–11145, 2025.

Lisha Li, Jingwen Hou, Weide Liu, Yuming Fang, and Jiebin Yan. Diffusion-based facial aesthetics enhancement with 3D structure guidance. *IEEE Trans. Image Process.*, 34:1879–1894, 2025. doi: 10.1109/TIP.2025.3551077.

Mengtian Li, Shengxiang Yao, Zhifeng Xie, and Keyu Chen. GaussianBody: Clothed human reconstruction via 3D Gaussian splatting, 2024a.

Wenrui Li, Fucheng Cai, Yapeng Mi, Zhe Yang, Wangmeng Zuo, Xingtao Wang, and Xiaopeng Fan. Scenedreamer360: Text-driven 3D-consistent scene generation with panoramic Gaussian splatting, 2024b.

Zonghao Lin, Ruiyan Wang, Jun Ling, and Li Song. Siavatar: Animatable 3D gaussian avatar from a single image. In *Proc. IEEE Int. Conf. Image Process.*, pp. 1067–1072, 2025.

Xian Liu, Xiaohang Zhan, Jiaxiang Tang, Ying Shan, Gang Zeng, Dahua Lin, Xihui Liu, and Ziwei Liu. HumanGaussian: Text-driven 3D human generation with gaussian splatting. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6646–6657, 2024.

Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: A skinned multi-person linear model. *ACM Trans. Graph.*, 34(6):1–16, 2015.

Supreeth Narasimhaswamy, Uttaran Bhattacharya, Xiang Chen, Ishita Dasgupta, Saayan Mitra, and Minh Hoai. HanDiffuser: Text-to-image generation with realistic hand appearances. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 2468–2479, 2024.

Panwang Pan, Zhuo Su, Chenguo Lin, Zhen Fan, Yongjie Zhang, Zeming Li, Tingting Shen, Yadong Mu, and Yebin Liu. HumanSplat: Generalizable single-image human gaussian splatting with structure priors. *Adv. Neural Inf. Process. Syst.*, 37:74383–74410, 2024.

Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3D hands, face, and body from a single image. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 10975–10985, 2019.

Lorenza Prospero, Abdullah Hamdi, Joao F Henriques, and Christian Rupprecht. GST: Precise 3D human body from a single image with gaussian splatting transformers. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 6007–6017, 2025.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Mach. Learn.*, pp. 8748–8763. PMLR, 2021.

Zhijing Shao, Duotun Wang, Qing-Yao Tian, Yao-Dong Yang, Hengyu Meng, Zeyu Cai, Bo Dong, Yu Zhang, Kang Zhang, and Zeyu Wang. DEGAS: Detailed expressions on full-body gaussian avatars. In *Int. Conf. 3D Vis.*, pp. 1529–1540, 2025.

Yuexin Sun, Zhenhua Hao, Zhaohui Wang, Jiping Jin, Qinwen Ye, and Yingrui Lyu. Deep learning for 3D garment generation: A review. *Textile Research Journal*, 2025.

David Svitov, Pietro Morerio, Lourdes Agapito, and Alessio Del Bue. HAHA: Highly articulated gaussian human avatars with textured mesh prior. In *Proc. Asian Conf. Comput. Vis.*, pp. 4051–4068, 2024.

Cyrus Vachha and Ayaan Haque. Instruct-GS2GS: Editing 3D gaussian splats with instructions. *URL https://instruct-gs2gs.github.io*, 2024.

Junjie Wang, Jiemin Fang, Xiaopeng Zhang, Lingxi Xie, and Qi Tian. GaussianEditor: Editing 3D gaussians delicately with text instructions. In *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, pp. 20902–20911, 2024a.

Yu Wang, Xiaobao Wei, Ming Lu, and Guoliang Kang. PLGS: Robust panoptic lifting with 3D Gaussian splatting. *IEEE Trans. Image Process.*, 34:3377–3388, 2025. doi: 10.1109/TIP.2025.3573524.

Yuxuan Wang, Xuanyu Yi, Zike Wu, Na Zhao, Long Chen, and Hanwang Zhang. View-consistent 3D editing with gaussian splatting. In *Eur. Conf. Comput. Vis.*, pp. 404–420. Springer, 2024b.

Jing Wu, Jia-Wang Bian, Xinghui Li, Guangrun Wang, Ian Reid, Philip Torr, and Victor Adrian Prisacariu. GaussCtrl: Multi-view consistent text-driven 3D gaussian splatting editing. In *Eur. Conf. Comput. Vis.*, pp. 55–71. Springer, 2024.

Jiazheng Xu, Xiao Liu, Yuchen Wu, Yuxuan Tong, Qinkai Li, Ming Ding, Jie Tang, and Yuxiao Dong. ImageReward: Learning and evaluating human preferences for text-to-image generation. *Adv. Neural Inf. Process. Syst.*, 36:15903–15935, 2023.

Yifan Zhan, Qingtian Zhu, Muyao Niu, Mingze Ma, Jiancheng Zhao, Zhihang Zhong, Xiao Sun, Yu Qiao, and Yinqiang Zheng. ToMiE: Towards modular growth in enhanced smpl skeleton for 3D human with animatable garments, 2024.

APPENDIX

A DATASET AND EXPERIMENTAL DETAILS

Our experiments are conducted on two human scenes, consisting of one male subject and one female subject. The male scene is obtained from the publicly available dataset released by Instruct-NeRF2NeRF. For the female scene, we capture a monocular video and process the recorded frames using the Nerfstudio pipeline to estimate the camera parameters and prepare the multi-view images. For both scenes, images from all viewpoints are cropped to a resolution of 320×512 and subsequently used to reconstruct the corresponding scenes with 3D Gaussian Splatting (3DGS). Following the official implementation settings, the input images are automatically resized to 512×512 during both scene reconstruction and the iterative 3DGS updating process of T3HG-Editor.

B HYPERPARAMETER SETTINGS AND SENSITIVITY ANALYSIS

B.1 HYPERPARAMETER SETTINGS

During the initial InstructPix2Pix editing pass, T3HG-Editor uniformly samples 40 views at approximately equal intervals from all available viewpoints. This provides ImageReward with a sufficiently diverse set of candidate results, facilitating the selection of edited images that better align with the input text instruction. The 20 highest-ranked views are subsequently retained and used for image editing and supervision during the remaining optimization process. Unless otherwise specified, the hyperparameters are configured following the official implementations of GaussianEditor and DGE, while the segmentation threshold of SAM is set to its official default value. The additional hyperparameters introduced by T3HG-Editor, including the maximum seeding distance $D_{\max}$ and the SDF-based distance threshold β, are determined according to the size, thickness, and geometric extent of the target garment. Both parameters are typically selected within the range of $[0, 0.1]\,\mathrm{m}$.

B.2 SENSITIVITY ANALYSIS

Figure 8 analyzes the sensitivity of the maximum seeding distance $D_{\max}$ and the SDF-based distance threshold β, which are set to the same value and jointly varied from 0 to 0.1 m. This range accommodates garments with different thicknesses and geometric extents. Three representative garments are evaluated. For the green work jacket and silver puffer jacket, setting both parameters to 0.06 m and 0.08 m, respectively, yields more complete sleeves and better garment shapes. For the long wine-red trench coat, smaller values excessively restrict the garment extent and cause visible dents around the lower hem, whereas 0.10 better preserves its overall silhouette. These results indicate that the parameters should be adjusted according to garment thickness and spatial extent.

C CLIP-BASED OPTIMIZATION TRENDS

We randomly render 20 camera views from the edited 3DGS to compute CLIP similarity and CLIP directional similarity. In addition, one garment editing case is selected to compare the trends of CLIP and CLIP directional similarity, thereby illustrating the efficiency of updating and the trajectory of performance improvement. Following the experimental protocol described above, Figure 9 and Figure 10 depict the iterative trends of CLIP similarity and CLIP directional similarity, using "*Turn his upper body into a gray trench coat*" as the text prompt. In early iterations, T3HG-Editor rapidly captures the coarse garment shape, leading to faster convergence than baselines; in later iterations, after enabling the consistency module, the curves become smoother as updating focuses on enforcing cross-view consistency in fine-grained garment textures. The results demonstrate that the proposed obtainment of editable Gaussians and garment consistent editing schemes provide reliable support at different updating stages, ultimately leading to superior garment editing performance.



Figure 8: Qualitative parameter sensitivity analysis of the maximum seeding distance $D_{\max}$ and the SDF-based distance threshold β. The two parameters are set to the same value and jointly varied from 0 to 0.1 m for three garments with different thicknesses and geometric extents.

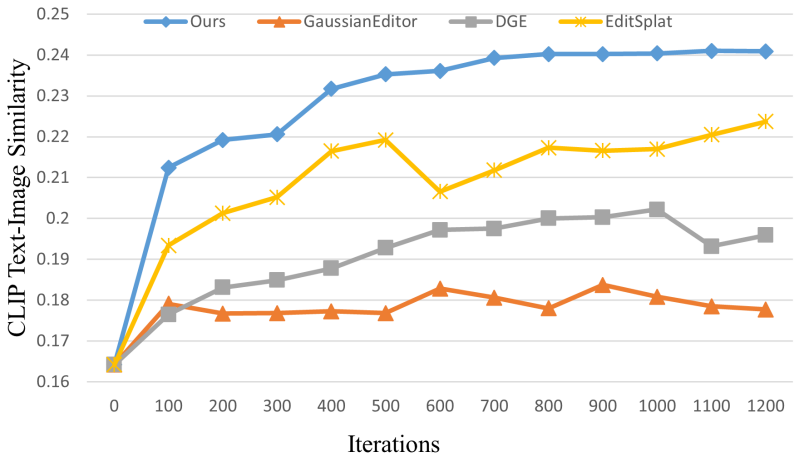


Figure 9: Iterative trends of CLIP similarity.

D Additional Experimental Results

Figure 11 presents additional qualitative comparisons on both male and female subjects under diverse garment-editing instructions. For the instruction "Turn his upper body into a dark green work jacket with practical pockets," GaussianEditor, DGE, and EditSplat introduce noticeable changes to

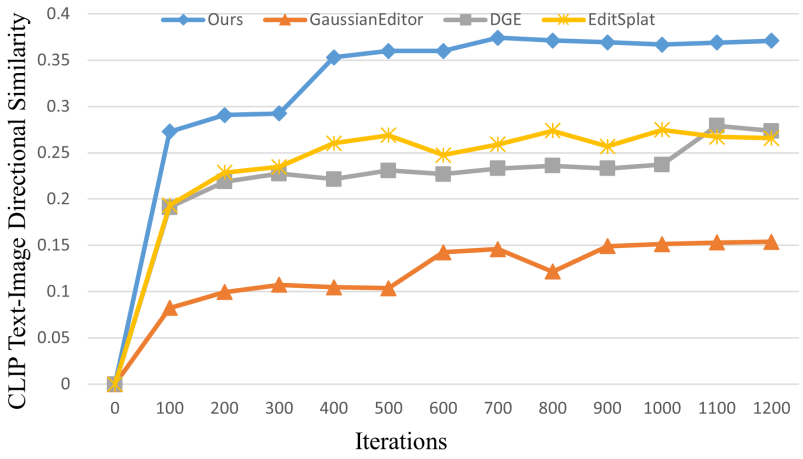


Figure 10: Iterative trends of CLIP directional similarity.

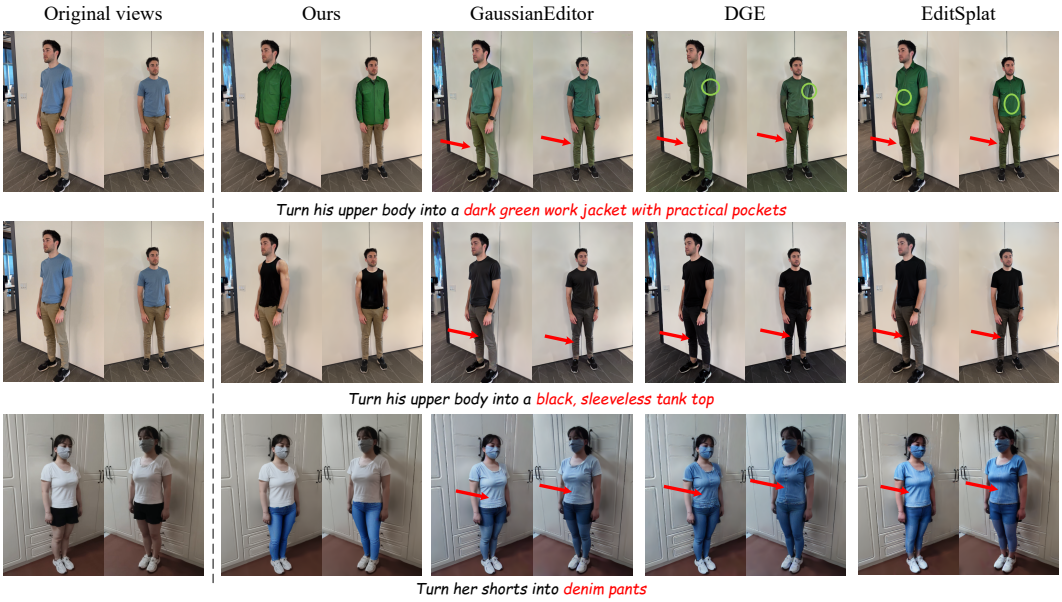


Figure 11: Additional qualitative comparisons under diverse garment-editing instructions. Red arrows indicate unintended changes in non-target regions, while green circles highlight inconsistent garment structures across views.

the pants and other non-target regions, as indicated by the red arrows. Moreover, DGE and EditSplat retain short-sleeve contours and exhibit inconsistent garment structures across viewpoints, as highlighted by the green circles. In contrast, T3HG-Editor produces a more complete jacket structure while better preserving the appearance of the pants and background. For the instruction "Turn his upper body into a black, sleeveless tank top," the compared methods largely preserve the original short-sleeve structure and simultaneously alter the color of the pants. T3HG-Editor, by contrast, successfully removes the sleeves and confines the modification to the target upper-body garment region. A similar advantage is observed for "Turn her shorts into denim pants." Although the compared methods generate blue pants, they also modify the color and appearance of the original T-shirt, resulting in unintended changes outside the target region. T3HG-Editor produces the desired denim pants while maintaining the original upper-body garment and surrounding scene. These results further demonstrate that T3HG-Editor can accommodate diverse garment transformations



Figure 12: Qualitative comparison with mask. All methods adopt a 2D mask segmentation module and restrict updating to the Gaussians identified by the mask.

Method	PSNR↑	SSIM↑	LPIPS↓
Target Garment Region			
w/o Garment Consistency	19.5949	0.5405	0.2014
T3HG-Editor (Ours)	**21.2198**	**0.5796**	**0.1738**
Non-Target Garment Region			
w/o Editable Gaussians	17.0431	0.7619	0.3023
w/o Overflow Pruning	22.6321	0.8179	0.1372
T3HG-Editor (Ours)	**28.8880**	**0.9300**	**0.0828**

Table 3: Ablation study on the target and non-target garment regions. The best results are highlighted in **bold**.

while effectively suppressing unintended changes in non-target regions and maintaining garment consistency across views.

E Comparisons under Mask Guidance

Although GaussianEditor and DGE provide an optional segmentation strategy to localize editing regions, they only segment the pre-editing area and cannot adapt to post-edit clothing expansion or shrinkage. Figure 12 illustrates the results of mask-based localization of editable regions with GaussianEditor and DGE. Given the prompts "*Turn his upper body into a Plaid Sweater*" and "Turn her upper body into a silver puffer jacket," both baselines fail to generate sleeves for the edited garments and do not exhibit perceptible garment thickness. These findings further substantiate the effectiveness of the editable Gaussian acquisition scheme in T3HG-Editor, which robustly accommodates shape changes introduced by the edits.

F Quantitative Ablation Study

Table 3 further quantifies the contribution of each component. Removing the garment consistent editing module degrades all metrics in the target garment regions, confirming its effectiveness in improving garment consistency. In the non-target regions, removing either obtainment of editable Gaussians or overflow pruning causes clear performance drops, while the full T3HG-Editor achieves the best PSNR, SSIM, and LPIPS. These results verify that the proposed components jointly improve editing consistency and suppress contamination in non-target regions.