Creating Your Editable 3D Photorealistic Avatar with Tetrahedron-constrained Gaussian Splatting

Supplementary Material

In our supplementary material, we provide:

- Details of 3D avatar instantiation.
- Details of localized spatial adaptation.
- Details of texture generation.
- Details of reference image-based editing.
- More edited results.
- Experimental details on baseline comparison.
- Applications.
- Limitations.

A. Details of 3D avatar instantiation

To provide accurate geometry and appearance prior for TetGS initialization and pave the way for the following editing phase, we perform high-quality 3D avatar instantiation from captured real-world monocular videos.

A.1. Architecture of implicit reconstruction with SDF field

To obtain precise geometric surface for TetGS initialization, we conduct multi-view surface reconstruction utilizing an SDF field, which is instantiated by a 4-layer MLP ψ with 512 hidden units per layer. Given a spatial point x, the SDF field ψ maps it to its signed distance value $\hat{s}$ to the object surface. A predicted normal $\hat{\mathbf{n}}'$ and a geometric feature $\hat{z}$ is also output by ψ. The SDF field is followed by an appearance field ψ_{app} which predicts the view-dependent color $\hat{c}$ for point x under view direction d. The appearance MLP ψ_{app} has 2 layers with 128 hidden units. The network architecture is illustrated in Fig. 10. To improve sampling efficiency, we apply two rounds of proposal sampling and then a NeRF sampling following Mip-NeRF 360 [3]. The overall training loss contains a color reconstruction loss L_c, an eikonal loss [16] L_{reg}, and two normal regularization losses L_p and L_o:

$$L_{rec} = L_c + \lambda_{reg} L_{reg} + \lambda_p L_p + \lambda_o L_o, \tag{11}$$

where $\{\lambda_{reg}, \lambda_p, \lambda_o\}$ are set to $\{0.1, 10^{-6}, 10^{-3}\}$.

A.2. TetGS initialization

The reconstructed geometry is directly converted into tetrahedron grids, where we embed a different number of Gaussians for each tetrahedron. The number of Gaussians assigned to each tetrahedron is based on its extracted face area relative to the average size: faces larger than average are assigned three Gaussians, while smaller faces receive one Gaussian. The optimization of the embedded Gaussians $\mathcal{G}$ follows the original 3DGS method [27], where we apply the pixel-wise reconstruction loss between the multi-view renderings $\hat{I}$ and the training images I_{gt} sampled from the input monocular video:

$$L = L_1(\hat{I}, I_{gt}) + \lambda L_{SSIM}(\hat{I}, I_{gt}). \tag{12}$$

To better capture high-frequency geometry and texture details, we perform TetGS initialization inside the subdivided tetrahedron grids [56].

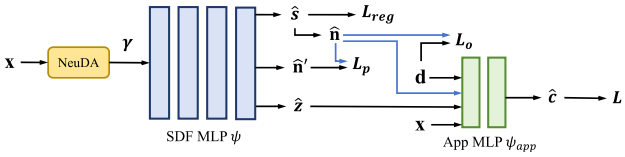

Figure 10. The architecture of the implicit reconstruction with SDF field.

Table 3. Quantitative evaluation (test-set view) of our method compared to 3DGS averaged on our collected dataset. 7K and 30K denote training iterations.

Method|Metric	PSNR ↑	SSIM ↑	LPIPS ↓
3DGS-7K	26.12	0.933	0.195
3DGS-30K	**27.31**	0.941	0.175
Ours-7K	26.67	**0.947**	**0.157**

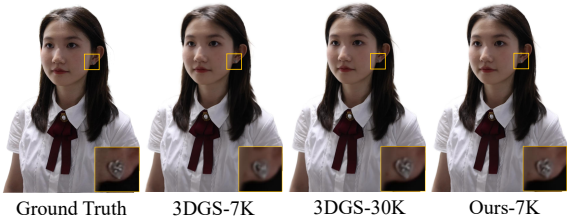

Figure 11. Qualitative reconstruction comparison of TetGS against 3DGS.

A.3. Reconstruction performance of TetGS against 3DGS

Comparison of TetGS with 3DGS is shown in Fig. 11 and Tab. 3. TetGS demonstrates comparable reconstruction performance to 3DGS, while achieving faster convergence of Gaussian parameters, benefiting from the guidance of tetrahedral grids on 3D Gaussians' spatial allocation.

B. Details of localized spatial adaptation

During the localized spatial adaptation of TetGS, we adopt a view sampling strategy similar to [24], focusing on both

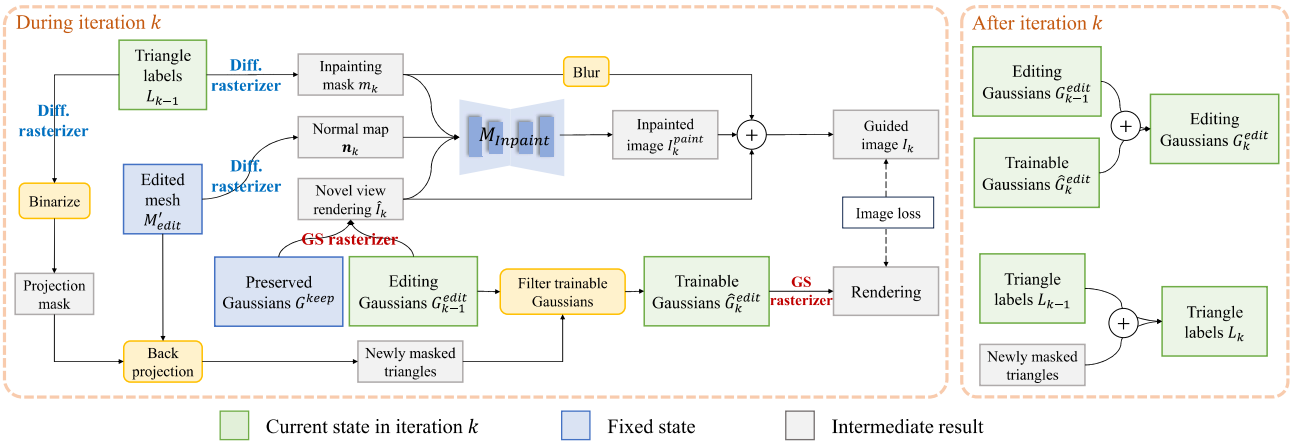

Figure 12. An overview of the coarse texture generation stage.

global and local regions to calculate the dual spatial constraints. The resolution of the rendered normals is 512×512. We set the global and local text prompts y^G and y^L as *"photo of a man/woman wearing a ... garment"* and *"photo of a ... garment"*, respectively. For calculating the geometric guidance L_{SDS}^G and L_{SDS}^L, we use a publicly available normal-adapted Stable Diffusion V1.5 model [24] to generate more detailed geometry. We set the guidance scale at 50, and anneal the timestep t from $t \sim \mathcal{U}(0.02, 0.80)$ into $t \sim \mathcal{U}(0.02, 0.20)$. Inspired by [26], during the early phase of the spatial adaptation, we render normals on the coarse tetrahedron grids with a resolution of 512^3 to encourage fast and large-scale deformation, and optimize detailed geometry within subdivided high-resolution tetrahedrons in later steps of training.

C. Details of texture generation

With the reallocated editing Gaussians $\mathcal{G}^{edit}$ with already learned spatial distribution, we propose to generate texture within the editing regions in a coarse-to-fine manner. The optimizer and training hyperparameters for $\mathcal{G}^{edit}$ are shared with the previous TetGS initialization stage. The guided text prompts used for texture generation are shared with the global prompt y^G during the spatial adaptation stage.

C.1. Overview of coarse texture generation

We show an overview of the coarse texture generation stage in Fig. 12, where we iteratively optimize trainable editing Gaussians under the supervision of the few-shot inpainted images. Detailed descriptions are included in Sec. 3.2.3 of the main paper.

C.2. Diffusion guidance during texture generation

We use the publicly available SDXL-based ControlNet-Plus [68] provided on Hugging Face for both normal-based coarse texture inpainting and I2I augmentation, which is a unified all-in-one ControlNet for image generation and editing. We integrate the normal branch and inpainting branch for the normal-based inpainter to generate consistent texture faithful to the underlying geometry, and apply the tile super resolution branch for I2I augmentation that boosts local details while preserving the original contents. We generate the inpainted and augmented images focusing at the editing region with a resolution of 1024×1024.

D. Details of reference image-based editing

The proposed controllable TetGS and decoupled editing strategy naturally support reference image-guided 3D virtual try-on. We collect diverse types of garments from e-commerce websites as the reference images. The corresponding text prompts are generated by GPT-4 [1] with the command *"describe the color and style of the garment"*. We generate front and back-view try-on images I_f and I_b separately using IDM-VTON [13], where we input the reconstructed front or back rendering, the editing mask, the reference garment, and its corresponding text description. Since the individual and reference clothing remain consistent for both views, the generated images inherently maintain a globally coherent style. Specifically, for the back view, we add *"backview"* to the text prompt to enhance image quality. The generated I_f and I_b serve as the guidance images for the appearance learning of the editing Gaussians, which facilitate direct texture transfer of the specific garment styles.

We also apply additional geometric supervision L_{vton} during the localized spatial adaptation stage to recover faithful geometric design (Eq.10 in the main paper). The loss weights $\{\lambda_{norm}, \lambda_{mask}\}$ are set to $\{0.03, 1.0\}$.

E. More edited results

We showcase more edited results in Fig. 15 and Fig. 16, including both text-guided 3D editing and reference image-

based 3D virtual try-on. We demonstrate that our proposed method can handle various editing scenarios, covering upper garments, lower garments, and dresses. Our generated editable 3D avatars exhibit accurate region localization, flexible geometric adaptation, and coherent renderings with high fidelity and photorealism comparable to real-world individuals.

F. Experimental details

We compare our method with three 3D editing baselines: GaussianEditor [60], DGE [11] and TIP-Editor [77], which are state-of-the-art methods for text or image-guided 3D scene editing. The selected baselines cover various approaches for 3D editing with Gaussian splatting, including supervisions based on the iN2N [18], the SDS loss [47], and multi-view consistent edited images.

GaussianEditor. GaussianEditor pioneers in 3D scene editing with Gaussian Splatting. It facilitates two editing models, which utilize delta denoising score [20] (GaussianEditor-DDS) and Instruct-NeRF2NeRF [18] (GaussianEditor-iN2N) as the generative guidance for editing, respectively. Since GaussianEditor-iN2N exhibits better editing performance across various scenarios, we compare our method with GaussianEditor-iN2N. To meet the instruction requirement of iN2N, we convert our text prompts into the format of *"put him/her into ..."* or *"give him/her a ..."*. To specify the local editing region with text-guided SAM [30], we manually provide segmentation prompts describing the interested area for GaussianEditor's semantic tracing process, which is the same prompt that we use in our method to segment multi-view masks for tetrahedron partitioning.

DGE. DGE is a representative 3DGS editing method that uses a multi-view consistent 2D image editor for a more stable generative supervision with spatial consistency. It utilizes Instruct-pix2pix [5] as the underlying image editor and adopts spatiotemporal attention for view-consistent editing following video editing methods. DGE is built on the implementation of GaussianEditor, which also enables local semantic tracing. Thus, the editing text prompts and segmentation prompts of DGE can be shared with GaussianEditor.

TIP-Editor. TIP-Editor enables text-and-image-guided 3DGS editing under the supervision of the SDS loss [47] propagated by a personalized T2I model, where a DreamBooth model [52] is used for original scene personalization and a LoRA layer [22] is added for editing content personalization. We use the collected reference garment images for the LoRA training. To adapt to the concept-driven models used in TIP-Editor, we convert our editing text prompts to meet the format of *"a V_1 man/woman wearing a V_2 garment"*. For fair comparison on the localized editing task, we manually set its editing bounding box close to the editing region localized by our method.

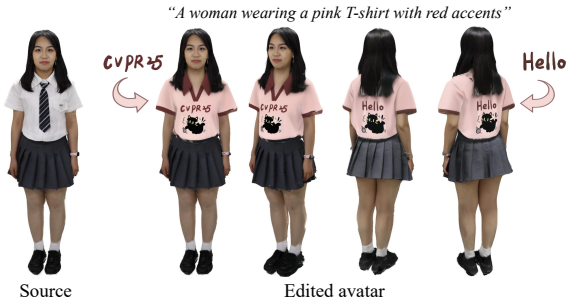

Figure 13. Application on customized texture doodling.

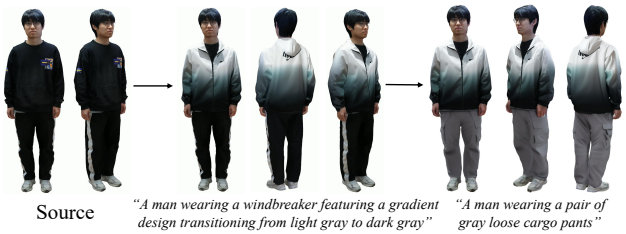

Figure 14. Application on continuous editing.

G. Applications

Texture doodling. Benefit from the controllable TetGS representation which naturally supports decoupled geometry and appearance editing, we achieve customized texture doodling by manually editing the front and back guidance images as the supervision of the texture generation stage. Users can paint any pattern to the guidance images, and the modified texture can be directly transferred into the edited 3D avatar by optimizing Gaussian appearance under the supervision of the painted images. We show an example in Fig. 13, where we paint *'CVPR25'* and *'Hello'* on the front and back views of the woman's T-shirt.

Continuous editing. Our method can continuously edit the source avatar, benefiting from our localized editing strategy. Fig. 14 shows results of changing the upper garment followed by the pants.

H. Limitations

Static human scene. Our method is proposed for static human scenes and the performers should stay still during the video capture. Dynamic portraits and obvious jittering may bring confused surfaces and blurred textures, due to the misalignment between multi-view observations on the pixel level.

Extreme editing case. Our method may struggle to generate proper geometric changes when editing from loose garments (e.g., dresses) to tight garments, as the pose and shape of the individual's inner body are ambiguous in those situations. Adding estimated inner body prior during editing can be a potential solution to mitigate this issue.

Figure 15. More results on text-guided 3D avatar editing.

Figure 16. More results on reference image-guided 3D avatar editing.