Supplementary Material for
Robust 3D-Masked Part-level Editing in 3D Gaussian Splatting with Regularized Score Distillation Sampling

S.1. Additional details on quantitative results

S.1.1. Experimental setting

S.1.1.1. Comparison with 3D Gaussian editing models

We collected human face scenes from the IN2N [12] and Nerf-Art [33] datasets. For each facial part: 'eyes', 'nose', 'mouth' and 'hair', we applied five editing prompts: 'silver-textured', 'gold-textured', 'diamond', 'green', 'pink' to evaluate editing success. Additionally, we designed five prompts requiring drastic changes: 'delicious croissant nose', 'hair made of metallic gears, steampunk style', 'hair on fire, red and blue flame', 'hair covered with beautiful butterfly', 'left blue and right green eye', and categorized them as 'hard' to assess extreme editing performance. For models incorporating InstructPix2Pix [2] in their pipelines, we adapted the prompts to the format: "Turn ... into ...".

S.1.1.2. Comparison with 3D Gaussian generation models

To prove that our local 3D editing method enhances controllability in 3D content generation, we designed prompts for samples that were challenging for previous 3D generation methods to create. The prompts included: 'A beautiful woman with a cheek's beak', 'A woman with cloudy hair', 'A beautiful woman with butterfly hair', 'A snail with skyscapes inside its shell', and 'A vase with a yellow tulip and a stained glass-textured rose'. We tasked 3D generation models with directly generating 3D content from these prompts. In our approach, we first generated the base objects, such as 'A snail', then applied these prompts as editing instructions to assess whether our method could successfully produce the desired samples.

User Study We conducted a user study across three categories: (1) Alignment - Is the 3D Gaussian edited to match the text? (2) Fidelity - Does the image look visually appleaing? (3) Accuracy - Were only the specified parts edited correctly?. Users were asked to score a 4-point scale, and we averaged it for mean opinion score (MOS). For reconstructed scene, participants evaluated all three criteria, collecting 4,680 responses from 260 respondents. For generated 3D, evaluations were based on alignment and fidelity, yielding 2,600 responses from 260 respondents.

S.1.1.3. Metrics

CLIP and CLIP directional score The CLIP-based metrics calculate the cosine similarity between text and image features extracted using CLIP [26]. CLIP scores are commonly utilized in evaluating text-to-3D [20, 24, 31]. CLIP directional scores are specifically employed to evaluate whether the changes occurred in the desired direction, first introduced by [10] and adopted mostly by editing models [3, 4, 7, 35]. We used the ViT-L/14 version of the model, with images cropped to 512 pixels and resized to 336 pixels before being input into the model.

TIFA and BLIP score While CLIP-based metrics effectively evaluate coarse similarity between image and text, they have limitations in assessing fine-grained correspondences [1, 7, 13, 14, 29]. To address this, we adopted two additional evaluation metrics focused on fine-grained visual-textual alignment, based on visual question answering (VQA). The TIFA score, introduced in [13], measures the faithfulness of generated image to text input by generating questions with LLaMA2 [30], answering with UnifiedQA-v2 [16]. BLIP-VQA, proposed in [14] breaks down a prompt into multiple questions, assigning a score based on the probability of answering 'yes' to each question, leveraging the vision-language understanding and generation capabilities of BLIP [19].

S.1.1.4. Implementation details

Our method is implemented in PyTorch [23], based on Threestudio [11]. We employ Stable Diffusion 3 [9]. All experiments are conducted on a single A100.

S.1.2. Experimental Results

Quantitative results Detailed quantitative results are shown in Tab. S.1, Tab. S.2, Tab. S.3 and Tab. S.4. The tables present quantiative results for each part editing. Our approach outperformed all other baselines in NeRF and Gaussian Splatting editing across all parts and metrics [3, 4, 8, 12, 17, 35]. Notably, considering that our models achieve strong performance on both CLIP-based and VQA-based scores, we can conclude that our models perform well in editing at both coarse and fine levels. Detailed results of user study for each evaluation criterion are provided in Table. S.5 and Table. S.6. Validity of the user

method	part											
	eye		nose		mouth		hair		hard		avg	
	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}		CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	
GaussCtrl [35]	0.191	0.042	0.183	0.035	0.173	0.056	0.195	0.060	0.168	0.026	0.182	0.044
GaussianEditor [4]	0.190	0.068	0.130	0.057	0.140	0.086	0.232	0.144	0.202	0.083	0.179	0.087
DGE [3]	0.193	0.076	0.190	0.058	0.182	0.070	0.232	0.161	0.211	0.110	0.201	0.095
RoMaP(ours)	**0.246**	**0.150**	**0.263**	**0.210**	**0.311**	**0.265**	**0.277**	**0.211**	**0.291**	**0.188**	**0.277**	**0.205**

Table S.1. **Comparison with GS editing methods.** CLIP score and CLIP directional score value for each method and part.

method	part											
	eye		nose		mouth		hair		hard		avg	
	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA
GaussCtrl [35]	0.194	0.422	0.195	0.561	0.223	0.389	0.239	0.494	0.098	0.292	0.190	0.432
GaussianEditor [4]	0.361	0.561	0.301	0.633	0.448	0.572	0.593	0.722	0.148	0.368	0.370	0.571
DGE [3]	0.517	0.539	0.427	0.717	0.512	0.5	0.774	0.683	0.255	0.388	0.497	0.565
RoMaP(ours)	**0.700**	**0.667**	**0.797**	**0.733**	**0.935**	**0.711**	**0.796**	**0.717**	**0.399**	**0.543**	**0.723**	**0.674**

Table S.2. **Comparison with GS editing methods.** BLIP-VQA score and TIFA score value for each method and part.

method	part											
	eye		nose		mouth		hair		hard		avg	
	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}	CLIP	CLIP_{dir}
iN2N [12]	**0.247**	0.067	0.257	0.071	0.258	0.084	0.253	0.079	0.227	0.060	0.248	0.072
VICA [8]	0.224	0.050	0.225	0.040	0.219	0.052	0.229	0.049	0.217	0.051	0.223	0.048
PDS [17]	0.162	-0.033	0.171	0.014	0.177	0.007	0.176	0.008	0.152	-0.020	0.167	-0.005
RoMaP(ours)	0.246	**0.150**	**0.263**	**0.210**	**0.311**	**0.265**	**0.277**	**0.211**	**0.291**	**0.188**	**0.277**	**0.205**

Table S.3. **Comparison with NeRF editing methods.** CLIP score and CLIP directional score value for each method and part.

method	part											
	eye		nose		mouth		hair		hard		avg	
	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA	B-VQA	TIFA
iN2N [12]	0.168	0.589	0.168	0.489	0.163	0.471	0.139	0.671	0.072	**0.623**	0.142	0.565
VICA [8]	0.277	0.436	0.204	0.507	0.292	0.387	0.228	0.396	0.205	0.41	0.241	0.427
PDS [17]	0.267	0.2	0.287	0.173	0.264	0.147	0.333	0.160	0.034	0.380	0.237	0.212
RoMaP(ours)	**0.700**	**0.667**	**0.797**	**0.733**	**0.935**	**0.711**	**0.796**	**0.717**	**0.399**	0.543	**0.723**	**0.674**

Table S.4. **Comparison with GS editing methods.** BLIP-VQA score and TIFA score value for each method and part.

study result is evaluated using pairwise Wilcoxon tests and the Friedman test, as shown in Fig.S.3. The test results confirm that our method significantly outperforms other editing and generation methods with strong statistical significance and validating the effectiveness of our method.

Qualitative results We included more qualitative results of our approach in Fig. S.1, Fig. S.2, Fig. S.8, Fig. S.9, and

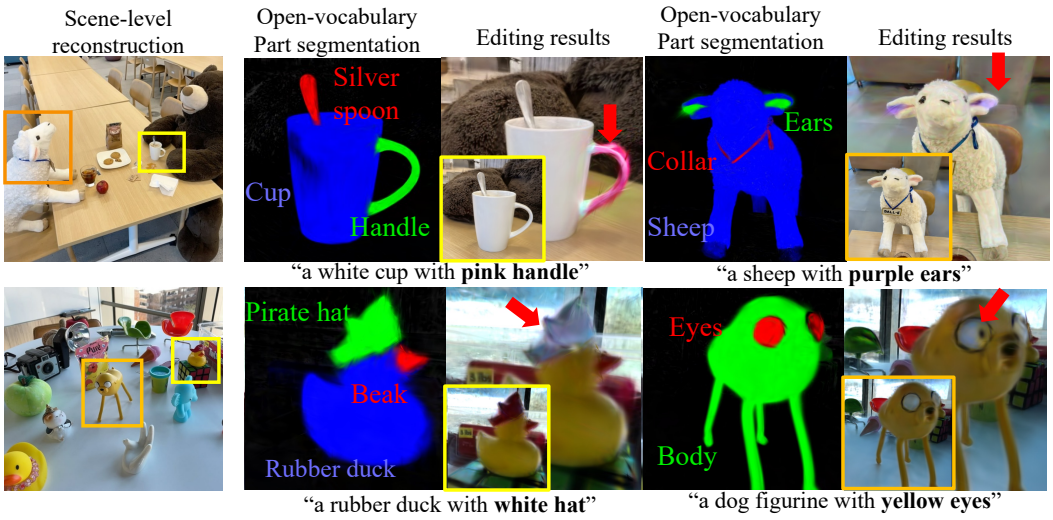


Figure S.1. **3DGS part editing results in complex 3DGS scenes.** We performed RoMaP editing on complex 3DGS scenes from the LERF dataset. As shown above, our RoMaP achieved precise open-vocabulary part segmentation for parts of varying sizes, such as the collar, eyes, body, and rubber duck. Additionally, we achieved accurate part editing based on prompts like 'a sheep with purple ears' and 'a rubber duck with a white hat'.

Figure S.2. **3DGS part editing results in complex scenes.** We demonstrate RoMaP editing results on complex 3D Gaussian Splatting (3DGS) scenes from both the 3D-OVS and LERF datasets. As shown above, RoMaP achieves high-quality normal editing, effectively handling diverse and practical edits such as 'with blue hair' or 'with a 'Hi' name tag'. These results highlight RoMaP's ability to generalize across various scene complexities.

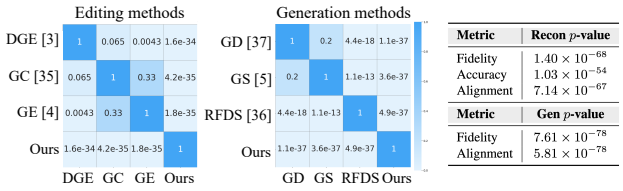
Fig. S.10. As shown in Fig. S.8, Fig. S.9 and Fig. S.10, our RoMaP can generate diverse 3D assets by editing the original 3D Gaussian Splatting (3DGS). Also, Fig. S.1 and Fig. S.2 show part-editing of our RoMaP in complex scenes. The results demonstrate that our 3D-GALP and editing strategies achieve high precision in 3D segmentation and enable precise modifications to the targeted regions, highlighting the scalability of our method to more complex and cluttered 3D scenes.

Method	Alignment	Fidelity	Accuracy
GaussCtrl [35]	19.70%	19.98%	20.6%
GaussianEditor [4]	19.61%	19.98%	20.72%
DGE [3]	23.18%	23.62%	20.24%
RoMaP (Ours)	**36.73%**	**36.31%**	**38.43%**

Table S.5. **User study results on comparison with 3D Gaussian editing models.**

Method	Alignment	Fidelity
GSGEN [5]	20.48%	20.09%
GaussianDreamer [37]	19.61%	19.98%
RFDS [36]	23.18%	23.62%
RoMaP (Ours)	**36.73%**	**36.31%**

Table S.6. **User study results on comparison with 3D Gaussian generation models.**



(a) Pairwise wilcoxon test with editing and generation methods

Metric	Recon p-value
Fidelity	1.40×10^{-68}
Accuracy	1.03×10^{-54}
Alignment	7.14×10^{-67}
Metric	**Gen p-value**
Fidelity	7.61×10^{-78}
Alignment	5.81×10^{-78}

(b) Friedman test on both results

Figure S.3. **Statistical results from user study.** (a) Pairwise Wilcoxon test results for editing and generation methods. (b) Friedman test p-values for fidelity, accuracy, and alignment. Our approach (Ours) achieves significantly better performance in both reconstruction and generation compared to existing methods.

Qualitative results of baselines We visualized qualitative results of Gaussian and NeRF-editing baselines in Fig. S.15 and Fig. S.16. For the NeRF baseline model, we present result from IN2N [12]. Due to the implicit nature of NeRF, precisely selecting the target region is challenging, often resulting in unintended global changes. For example, when applying the prompt 'Turn his hair into silver-textured hair', the entire scene shifts to a silver hue S.15. Similarly, prompts such as 'hair on fire' or 'left eye blue and right eye green' lead to incorrect region selection, causing widespread color alterations across the scene. For the Gaussian Splatting baseline, we show results from GaussianEditor [4]. Inconsistencies in 2D part segmentation lead to unreliable 3D part segmentation, as shown in Fig. S.16. Additionally, 2D editing results demonstrate difficulties in precisely modifying the desired regions. For instance, a croissant appears in the background instead of the intended edit, or the entire scene turns pink rather than just his eyes.

S.2. Additional results in complex scene

To further validate the robustness and generalizability of RoMaP, we present additional editing results on complex 3DGS scenes from both the 3D-OVS [21] and LERF [15] datasets. These scenes contain multiple objects with intricate part-level structures and diverse contextual settings.

As illustrated in Fig. S.1, RoMaP demonstrates precise open-vocabulary part segmentation and editing across a wide range of object types and part granularity. Examples include edits guided by prompts such as a 'white cup with pink handle', 'a rubber duck with white hat', and 'a dog figurine with yellow eyes'. RoMaP effectively identifies and modifies fine-grained parts such as handles, beaks, collars, and ears, even under cluttered backgrounds and occlusions.

In addition, Fig. S.2 further showcases our model's ability to perform practical part editing tasks involving realistic human and animal figures. Prompts such as 'with blue hair', 'with purple dress', and 'with 'Hi' name tag' illustrate RoMaP's capability to generalize beyond common categories and execute attribute-level modifications across highly complex scenes. These results collectively highlight RoMaP's strength in both semantic understanding and fine-grained spatial localization, making it a versatile tool for open-vocabulary 3D scene editing.

S.3. Additional validation and details of pipeline

S.3.1. Attention map extraction

Unlike the naive reverse flow-matching process used in text-to-3D generation, we adopted a controlled forward ODE to extract more accurate attention maps for real images, thereby enhancing robustness. Controlled forward ODE, proposed in [27], helps maintain consistency with the given image while aligning with the distribution of typical images. This balancing mechanism allows for effective inversion and editing across various inputs, especially real images, even when the given image is corrupted or atypical. Additionally, we adopted the approach proposed in [34] for dense prediction. This method allows for faster and more accurate extraction of attention maps.

Post-processing We post-processed extracted attention maps by normalizing them with a softmax temperature and utilizing a refiner [6]. Adjusting softmax temperature allowed us to segment regions with varying granularity, while the refiner, by incorporating the original image features, enabled segmentation of parts with more precise edges, as shown in Fig. S.4.

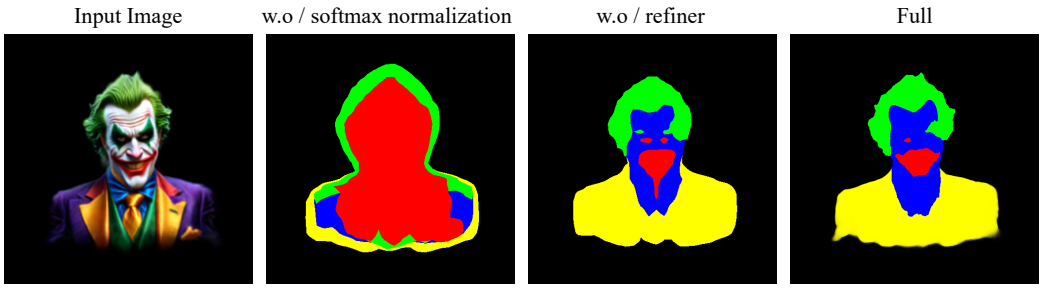


Figure S.4. **Ablation study of attention map post-processing procedure** By adjusting the softmax temperature, we achieved segmentation with varying levels of granularity, while the refiner, leveraging the original image features, facilitated the segmentation of parts with sharper and more defined edges.

S.3.2. 3D-geometry aware label prediction

S.3.2.1. Details of 3D-geometry aware label prediction

The detailed algorithm for 3D-Geometry Aware Label Prediction (3D-GALP) is provided in Algo. 1. 3D-GALP produces high-quality 3D segmentation maps even when part segmentation maps from multiple views are noisy, by applying a neighbor consistency loss that considers the soft-label property of Gaussian segmentation. Label softness is typically higher at part boundaries due to abrupt shape changes, which can lead to substantial variation in segmentation results across different views. Moreover, in practice, the Gaussians at these part boundaries may simultaneously represent pixels belonging to multiple parts depending on the viewpoint, further complicating consistent segmentation. To address this, Gaussians with both high and low softness are sampled, enabling continuous refinement of ambiguous as well as more view-invariant regions while taking surrounding information into account.

S.3.2.2. Part segmentation performance of 3D-GALP compared with other language-embedded 3DGS model in complex scenes

Experimental setting To evaluate how effectively 3D-GALP performs part segmentation in complex scenes, we annotated part segmentation for every object in all scenes of the 3D-OVS dataset [21]. We compared 3D-GALP with two text-aligned segmentation models for 3D Gaussians, LangSplat [25] and LeGaussian [28]. We kept hyperparameter, the softmax value for our 2D attention map extraction, to 0.2 during segmentation. We then evaluated part-segmentation results for each object from three different views, comparing them against ground truth using the mean Intersection over Union (mIoU). Examples of part-segmentation annotation are presented in Fig. S.5.

Experimental results As shown in Tab. S.7, our 3D segmentation method, 3D-GALP, achieves the highest mIoU, outperforming other 3DGS segmentation baselines across all scenes. Furthermore, 3D-GALP successfully performs open-vocabulary 3DGS segmentation for parts of varying sizes in complex scenes, as illustrated in Fig. S.11.

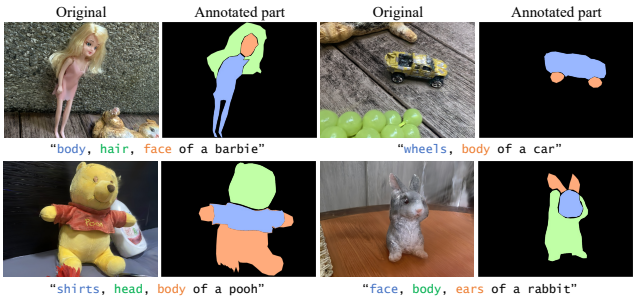


Figure S.5. **Examples of part segmentation annotation in 3D-OVS dataset.**

Scene	Bench	Blue sofa	Cov.desk	Room	Average
LangSplat [25]	0.005	0.076	0.093	0.129	0.076
LeGaussian [28]	0.320	0.312	0.264	0.257	0.288
3D-GALP (Ours)	**0.607**	**0.580**	**0.546**	**0.502**	**0.559**

Table S.7. **Comparison of 3D-GALP with part segmentation on complicated 3D scenes.**

S.3.2.3. Ablation study on SH degree

Experimental setting We ablated the SH order to analyze its effect on part-level segmentation. While low-order SH is typically sufficient for modeling lighting in color representation, part-level segmentation requires sharper spatial transitions, particularly around object boundaries. To evaluate this, we conducted experiments using the same experimental settings as in S.3.2.2 with different SH degree settings.

Experimental results As shown in Tab. S.8 and Fig. S.6, SH=3 consistently provides the best average mIoU across

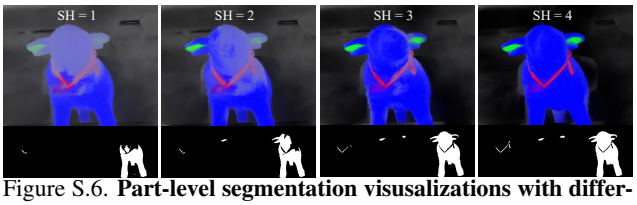


Figure S.6. **Part-level segmentation visusalizations with different SH orders.**

scenes and captures fine-grained parts more clearly than lower orders. Although SH=4 performs best in some scenes, it introduces more noise and higher memory usage, leading to slightly worse overall performance. Based on these observations, we fix SH=3 for all segmentation experiments, as it provides the best trade-off between detail preservation and stability.

Order of SH	**1**	**2**	**3**	**4**
mIoU	0.4777	0.5306	**0.5587**	0.5506

Table S.8. **mIoU average scores across the scenes per SH degree.** Best per scene is in **bold**.

S.3.3. Scheduled latent mixing and part editing

S.3.3.1. Scheduled latent mixing and part editing

The detailed algorithm is provided in Algo. 2. This method leverages the property of rectified flow that is more faithful to the original image. During the editing process, α_{base} is multiplied by the mask to ensure that regions outside the target editing area retain their original information. This introduces weak conditioning at intermediate steps of image generation, guiding the generated regions to align with the original context. At the timestep t_s, α_{last} is applied to ensure that most of the $\mathcal{M}_{\text{inv}}$ regions are replaced with z_{target}, preserving the majority of the reference image's information in the final output. Further results on the selection of t_s are shown in Fig. S.14. A low t_s induces dramatic changes based on the prompt, while a high t_s ensures faithful adherence to the mask, taking into account the original content and its context. In the t_s selection described in the main paper, we randomly selected 100 person images from the CelebAMaskHQ [18] dataset, performed part-level editing using 25 prompts, and evaluated the results using CLIP$_{dir}$ [10] and SSIM to assess the direction of change while preserving the original content. The full experimental results with 25 prompts are shown in Fig. S.7.

S.3.3.2. Comparison of SLaMP with other image editing models

Experimental setting To evaluate the effectiveness of our SLaMP in preserving non-target regions while accurately modifying only the specified parts compared to other

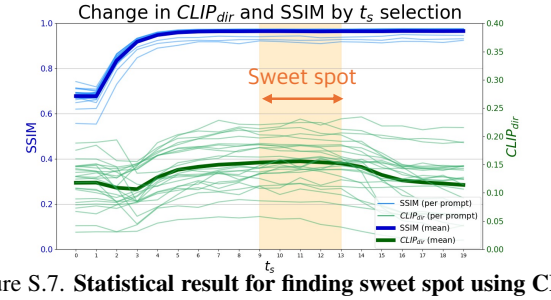


Figure S.7. **Statistical result for finding sweet spot using CLIP and SSIM results.**

models, we randomly selected 15 male and female images from the CelebAMaskHQ [18] dataset. For each image, we performed image editing using 25 prompts as described in Sec. S.1.1.1. For comparison, we selected SD3-based models (SD3-inpainting [36], Plug&Play [9], RF-inversion [27]), as well as an editing model based on naive latent mixing (RePaint [22]), in contrast to our scheduled latent mixing approach. Additionally, we include a training-based model, InstructPix2Pix (IP2P [2]), which is commonly adopted in 3DGS and NeRF editing approaches. For RePaint, we used a Stable Diffusion-integrated variant from HuggingFace Diffusers [32] library since RePaint is not originally designed for text-based image editing. We evaluated how well the changes aligned with the prompts using the CLIP$_{dir}$ [10] and B-VQA [14] metrics.

Metrics	**RePaint** [22]	**iP2P** [2]	**SD3-inp.** [9]	**Plug&Play** [36]	**RF-inv.** [27]	**SLaMP (Ours)**
CLIP$_{dir}$ ↑	0.111	0.117	0.147	0.044	0.089	**0.165**
B-VQA ↑	0.439	0.668	0.693	0.564	0.740	**0.758**

Table S.9. **Quantitative comparison of SLaMP with other 2D part editing baselines.**

Experimental results The quantitative experimental results are presented in Tab. S.9, and the qualitative results in Fig. S.12. SLaMP outperforms all other 2D image editing baselines across all metrics, including CLIP$_{dir}$ [10] and BLIP-VQA [14]. Unlike baselines that either fail to reflect the prompt or fail to preserve the original context, SLaMP produces significant changes in the target part while accurately maintaining the untouched regions, achieving strong alignment with the text prompt.

As shown in Fig. S.12, the widely used 2D image editing baseline for 3D editing research, iP2P [2], struggles to perform meaningful part edits and often deviates from the original image context. This helps explain why existing 3D editing models often produce no visible changes in part editing tasks. RePaint [22] employs a fixed blending ratio for harmonized inpainting, making it unsuitable for strong, prompt-driven part-level edits. In contrast, SLaMP adopts

a scheduled blending strategy that enables bold edits early on and gradually preserves global context, achieving both precise modifications and faithful preservation. Additional results of SLaMP editing can be found in Fig. S.13.

S.4. Social Impact and Limitations

In our methodology, we utilized existing datasets from prior works [2, 33]. These datasets include information about real individuals, and if the results of our editing approach are misused, it could lead to concerns regarding negative societal impacts. Therefore, we strongly advocate for the responsible use of our methodology in adherence to ethical guidelines and relevant laws. In perspective on limitation, our approach relies on 3D segmentation based on attention maps observed from 360-degree viewpoints. Consequently, it may not perform well when dealing with objects with highly complex geometries (*e.g.*, a Klein bottle), leading to unintended editing results. Additionally, if the Gaussian Splatting scene is inherently blurry or poorly reconstructed, it becomes difficult to distinguish individual components. This can cause SD3 to fail in accurately interpreting the scene, resulting in incorrect 3D segmentation or undesired editing outcomes.

Algorithm 1: Algorithm of 3D-geometry aware label prediction (3D-GALP).

Input: Gaussian Representation Ω, Camera Parameters $\mathcal{C}$, Number of Anchors K, Nearest Neighbors k, Segmentation Labels $\mathbf{s}_{labels}$

Output: Segmentation Loss $\mathcal{L}_{\text{3D}}$

// Initialize multi-view camera dataset

$\mathcal{D}_{\text{test}} \leftarrow \text{LoadMultiviewDataset}(\mathcal{C})$

// Compute SH consistency

$\mathbf{S} \leftarrow \Omega.\text{get\_sh\_objects}()$

$\mathbf{T} \leftarrow \emptyset$ // Store SH values for different views

foreach $\mathbf{b}$ *in* $\mathcal{D}_{test}$ **do**

  $\mathbf{d} \leftarrow \text{ComputeViewDirection}(\mathbf{b}, \mathcal{C})$

  $\mathbf{s}_b \leftarrow \text{EvalSH}(\Omega, \mathbf{S}, \mathbf{d})$

  $\mathbf{T} \leftarrow \mathbf{T} \cup \mathbf{s}_b$

// Compute variance and entropy for each Gaussian

foreach *Gaussian* i *in* Ω **do**

  Compute variance: $\boldsymbol{v_i} \leftarrow \frac{1}{|\mathbf{T}|} \sum_{\mathbf{r} \in \mathbf{T}} \|\mathbf{r} - \bar{\mathbf{r}}\|^2$, where $\bar{\mathbf{r}} = \frac{1}{|\mathbf{T}|} \sum_{\mathbf{r} \in \mathbf{T}} \mathbf{r}$

  Compute entropy: $\mathbf{sim} \leftarrow \frac{\bar{\mathbf{r}} \cdot \mathbf{R}_{\text{labels}}}{\|\bar{\mathbf{r}}\| \|\mathbf{R}_{\text{labels}}\|}$

  $\mathbf{p_i} \leftarrow \frac{e^{\mathbf{sim}}}{\sum e^{\mathbf{sim}}}$

  $\mathbf{H_i} \leftarrow -\sum \mathbf{p_i} \log(\mathbf{p_i} + \epsilon)$ // Compute entropy

  Compute label softness: $\mathbf{U_i} \leftarrow \mathbf{H_i} \cdot \boldsymbol{v_i}$

// Anchor Selection Based on label softness

Sort all Gaussians by U_i in descending order

Select $\lfloor K/2 \rfloor$ anchors with highest U_i

Select $\lfloor K/2 \rfloor$ anchors with lowest U_i

Define set of selected anchors: S

// Compute Anchor-Based Neighbor Consistency Loss

foreach *anchor* $i \in S$ **do**

  Find nearest neighbors $\mathcal{N}_k(i) = \{j_1, \ldots, j_k\}$ using Euclidean distance

  Compute L1 loss: $\mathcal{L}_{\text{3D}} \leftarrow \sum_{i \in S} \left[\frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} \|\mathbf{r}_i - \mathbf{r}_j\|_1 \right]$

return $\mathcal{L}_{3D}$

Algorithm 2: Scheduled latent mixing and part editing Algorithm

Input: Latents $\mathbf{z}$, Text Embeddings $\mathbf{E}$, Camera Condition $\mathbf{C}$, Timestep $\mathbf{T}$, Noise $\mathbf{n}_{\text{target}}$, Cfg scale c, γ, η_{values}, α_{base}, α_{last}, Mask $\mathcal{M}$, , Mix timestep t_s

Output: Model Prediction $\mathbf{m}_{\text{pred}}$

// Latent Initialization and Noise Target

for t_{curr}, t_{prev} *in timesteps*$[:-1]$, *timesteps*$[1:]$ **do**

  $\mathbf{t} \leftarrow t_{\text{curr}} \times 1000$

  $\mathbf{v}_{\text{pred}} \leftarrow \text{transformer}(\mathbf{z}_{\text{noisy}}, \mathbf{t}, \mathbf{E}_{\text{uncond}})$

  $\mathbf{v}_{\text{target}} \leftarrow (\mathbf{n}_{\text{target}} - \mathbf{z}_{\text{noisy}})/(1 - t_{\text{curr}})$

  $\mathbf{v}_{\text{interp}} \leftarrow \gamma \cdot \mathbf{v}_{\text{target}} + (1 - \gamma) \cdot \mathbf{v}_{\text{pred}}$

  $\mathbf{z}_{\text{noisy}} \leftarrow \mathbf{z}_{\text{noisy}} + (t_{\text{prev}} - t_{\text{curr}}) \cdot \mathbf{v}_{\text{interp}}$

$\mathbf{z}_{\text{target}} \leftarrow \mathbf{z}.\text{clone}$

for t *in timesteps* **do**

  $\mathbf{t} \leftarrow t/1000$

  $\mathbf{v}_{\text{pred}} \leftarrow \text{transformer}(\mathbf{z}_{\text{noisy}}, \mathbf{t}, \mathbf{E}_{\text{mix}})$

  $\mathbf{v}_{\text{target}} \leftarrow -(\mathbf{z}_{\text{target}} - \mathbf{z}_{\text{noisy}})/t$

  $\eta \leftarrow \eta_{\text{values}}[i]$

  $\mathbf{v}_{\text{interp}} \leftarrow \mathbf{v}_{\text{pred}} + \eta \cdot (\mathbf{v}_{\text{target}} - \mathbf{v}_{\text{pred}})$

  $\mathbf{z}_{\text{noisy}} \leftarrow \text{scheduler.step}(\mathbf{v}_{\text{interp}}, t, \mathbf{z}_{\text{noisy}})$

  $\mathcal{F} \leftarrow \alpha_{last}$ if $i > |\text{timesteps}| - t_s$ else α_{base}

  $\mathcal{M}_{inv} \leftarrow \mathcal{F} \times (1 - \mathcal{M})$

  $\mathbf{z}_{\text{noisy}} \leftarrow \mathbf{z}_{\text{noisy}} \times (1 - \mathcal{M}_{inv}) + \mathbf{z}_{\text{target}} \times \mathcal{M}_{inv}$

$\mathbf{m}_{\text{pred}} \leftarrow \mathbf{z}_{\text{noisy}}$

return $\mathbf{m}_{pred}$

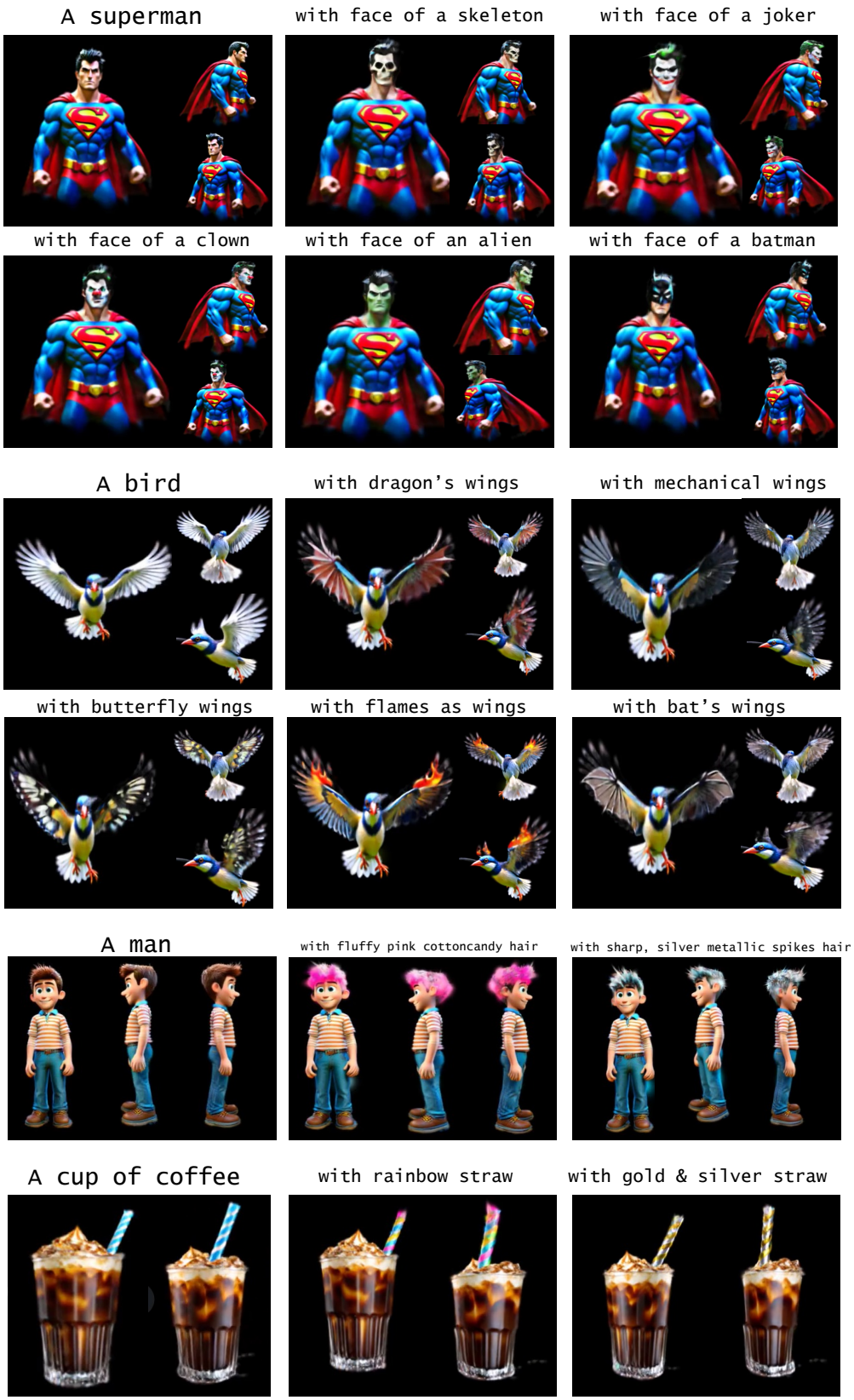
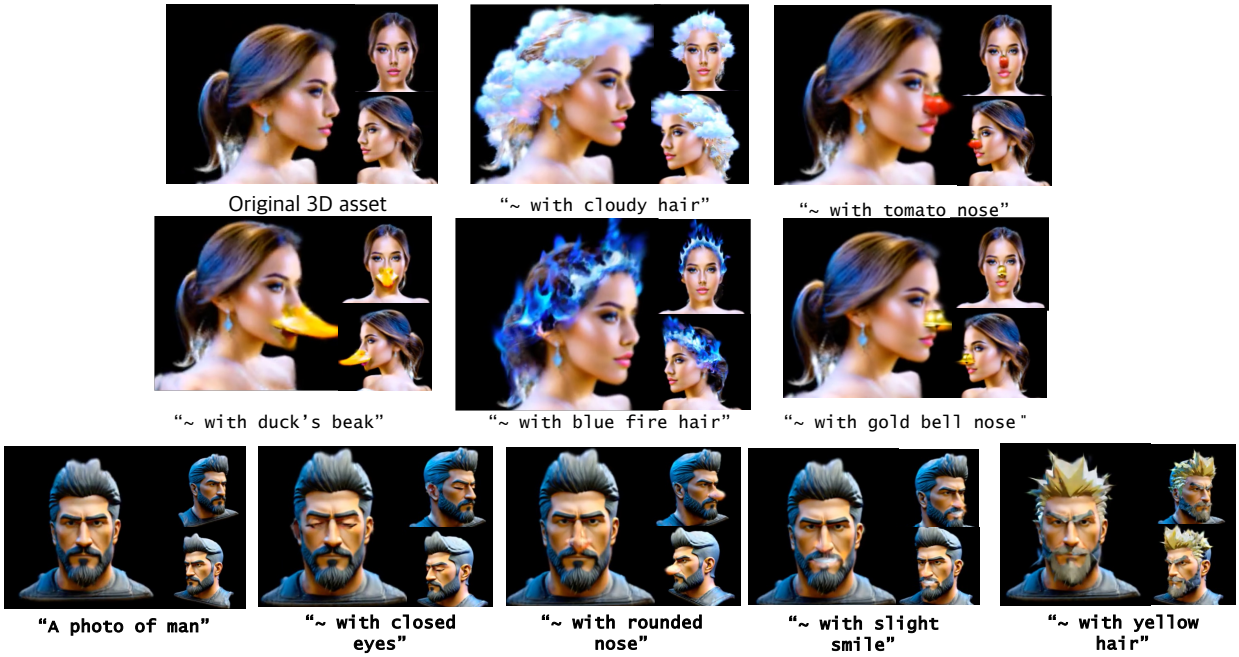
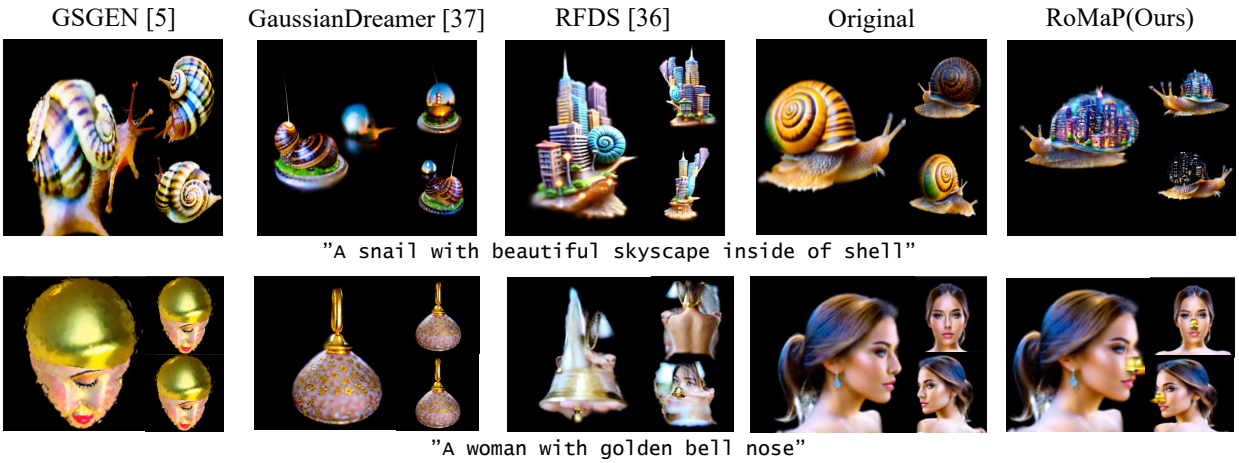


Figure S.8. **Additional qualitative results of RoMaP.** Our approach, RoMaP, enables editing across a wide range of parts, objects, and prompts in generated 3D Gaussians, further providing users with enhanced controllability over 3D content generation.



(a) Additional results for enhanced controllability in 3D asset generation



(b) Additional qualitative comparison with 3DGS generation models

Figure S.9. **Additional qualitative results of RoMaP.** Our approach, RoMaP, enables editing across a wide range of parts, objects, and prompts in generated 3D Gaussians, further providing users with enhanced controllability over 3D content generation.

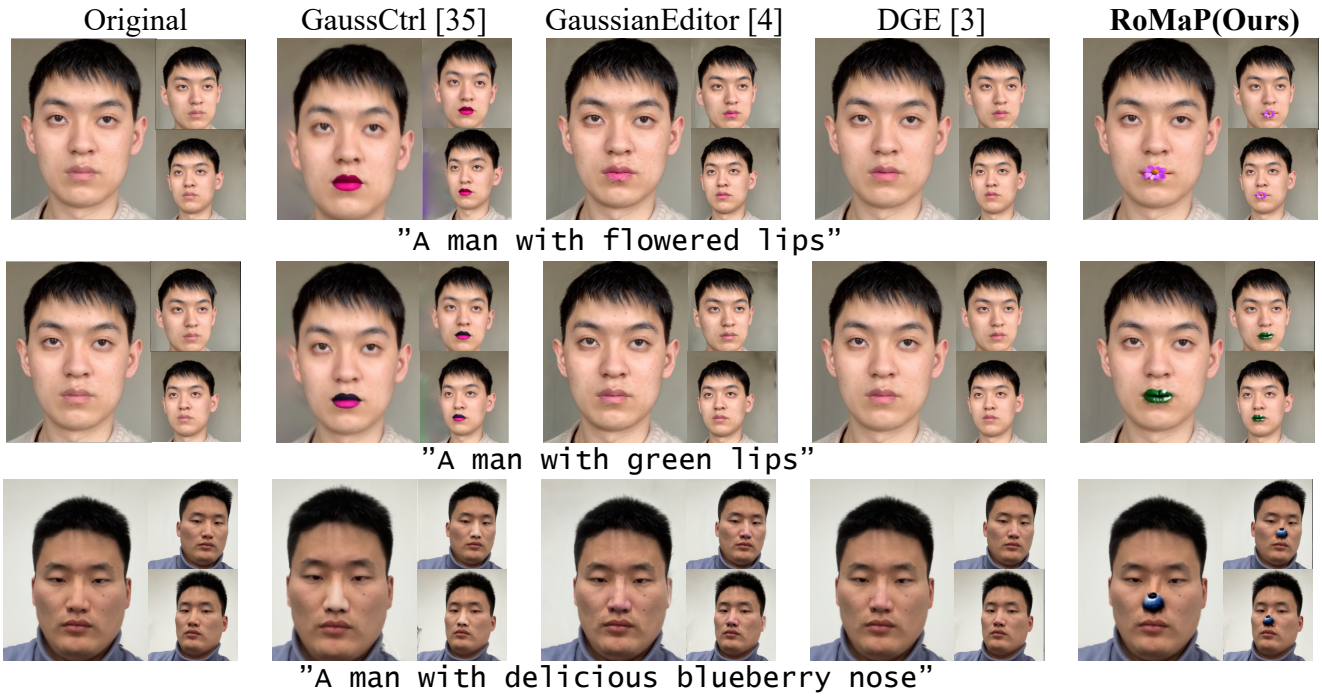


Figure S.10. **Additional comparison results of RoMaP.** Our approach, RoMaP, enables editing across a wide range of parts, objects, compare to other methods in 3D scene reconstruction settings.

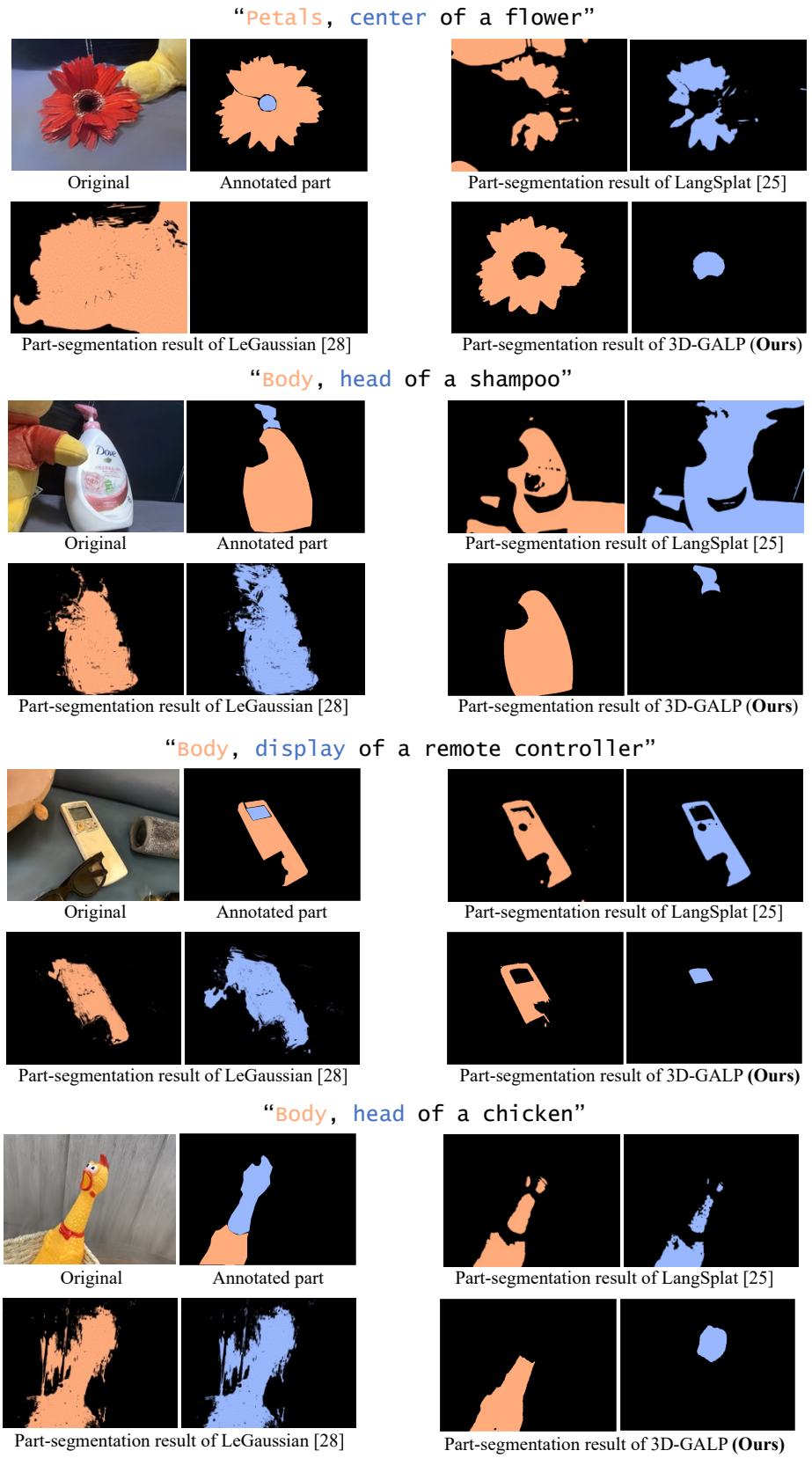


Figure S.11. **Open-voca part segmentation results comparison in complicated 3DGS scenes of 3D-OVS dataset.**

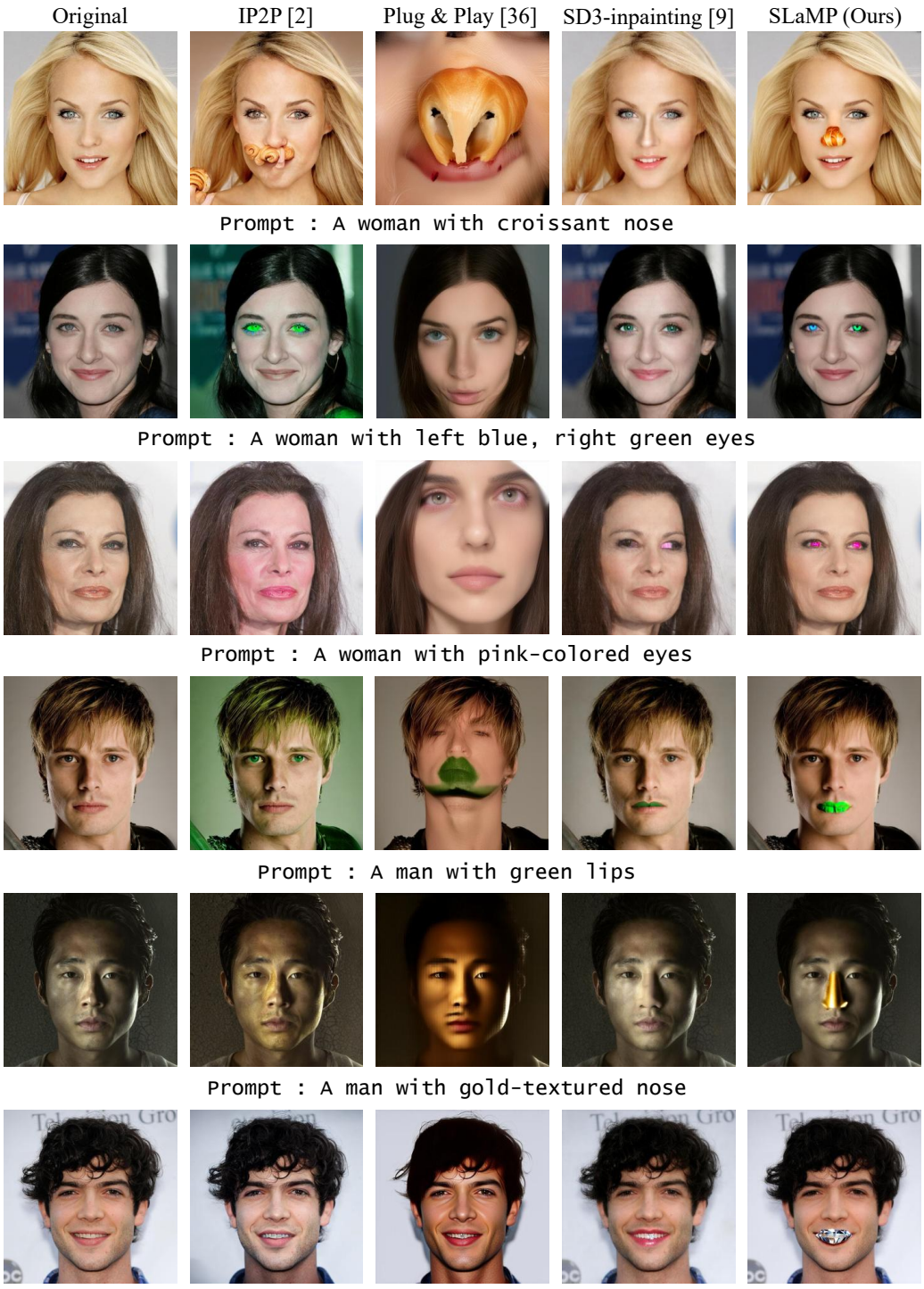


Figure S.12. **Local editing results between SLaMP and 2D image editing methods.** SLaMP editing employs rectified flow inversion to achieve effective modifications while maintaining the original context in unedited regions. This contrasts with 2D image editing baselines, which struggle to edit the specified part in alignment with the text prompt.

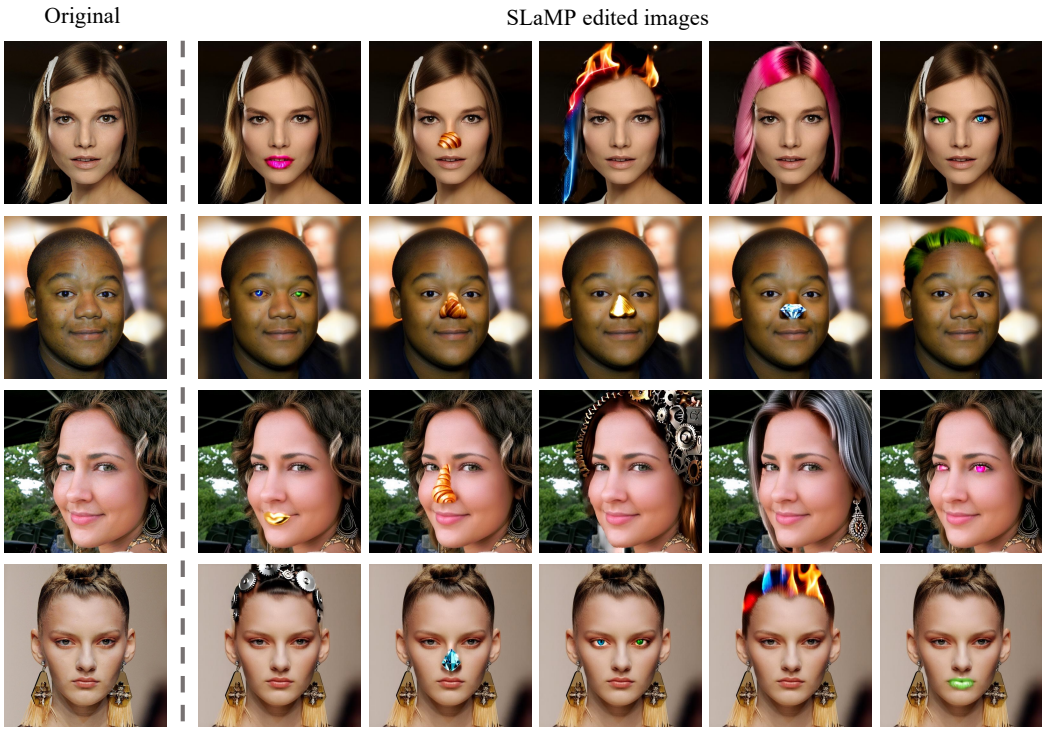


Figure S.13. **More 2D part editing results with SLaMP.**

Figure S.14. **Effect of different t_s in SLaMP editing.**

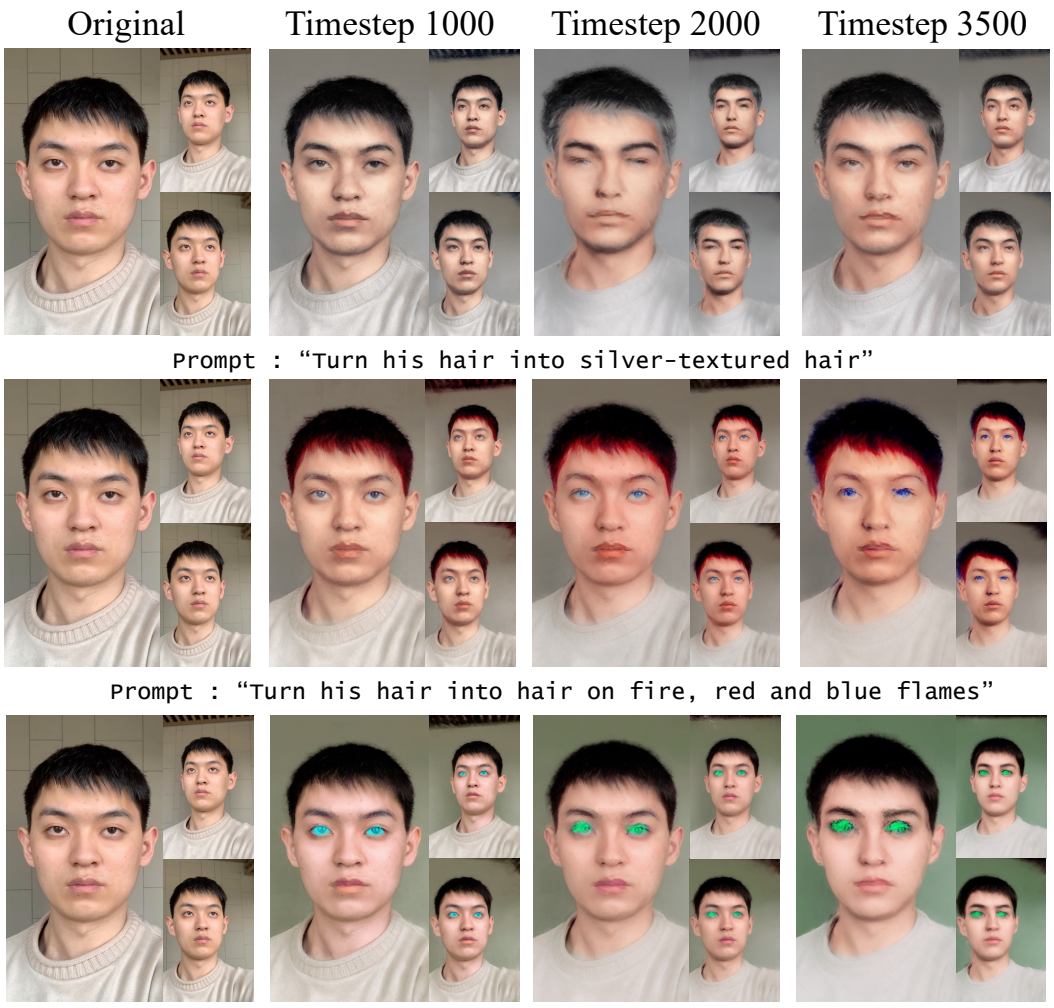


Figure S.15. **Qualitative results of nerf baseines [12] in 3D part editing.**

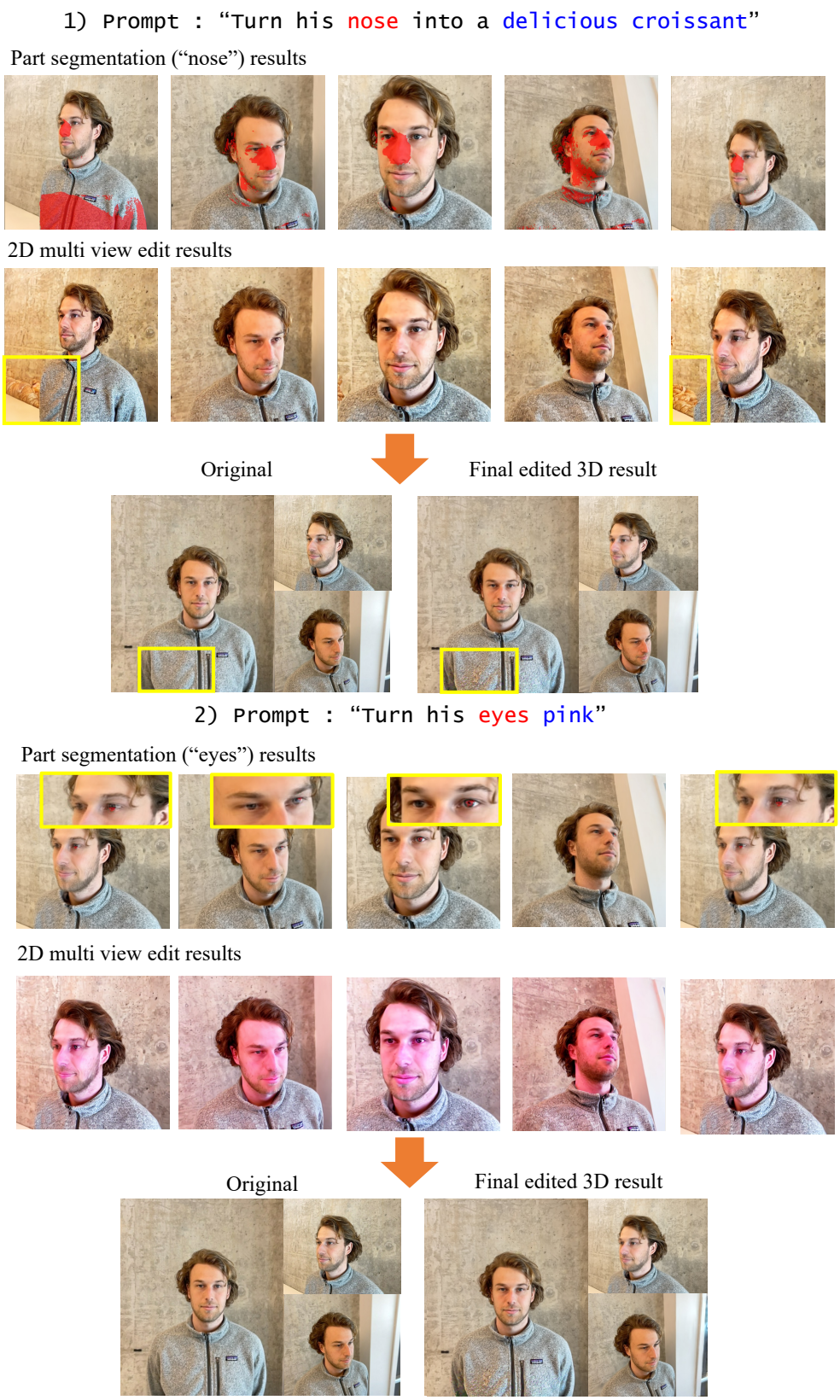


Figure S.16. **Qualitative results of 3DGS baseine [4] in 3D part editing.**

References

[1] Saba Ahmadi and Aishwarya Agrawal. An examination of the robustness of reference-free image captioning evaluation metrics. *ACL Anthology*, 2023. 1

[2] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 1, 6, 7

[3] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. *ECCV*, 2024. 1, 2, 4

[4] 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 *CVPR*, 2024. 1, 2, 4, 17

[5] Zilong Chen, Feng Wang, Yikai Wang, and Huaping Liu. Text-to-3d using gaussian splatting. In *CVPR*, 2024. 4

[6] Ho Kei Cheng, Jihoon Chung, Yu-Wing Tai, and Chi-Keung Tang. Cascadepsp: Toward class-agnostic and very high-resolution segmentation via global and local refinement. In *CVPR*, 2020. 4

[7] Xinhua Cheng, Tianyu Yang, Jianan Wang, Yu Li, Lei Zhang, Jian Zhang, and Li Yuan. Progressive3d: Progressively local editing for text-to-3d content creation with complex semantic prompts. *ICLR*, 2023. 1

[8] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *NeurIPS*, 2023. 1, 2

[9] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *ICML*, 2024. 1, 6

[10] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. 2022. 1, 6

[11] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 1

[12] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *CVPR*, 2023. 1, 2, 4, 16

[13] Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A Smith. Tifa: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *CVPR*, 2023. 1

[14] Kaiyi Huang, Kaiyue Sun, Enze Xie, Zhenguo Li, and Xihui Liu. T2i-compbench: A comprehensive benchmark for open-world compositional text-to-image generation. 2023. 1, 6

[15] Justin Kerr, Chung Min Kim, Ken Goldberg, Angjoo Kanazawa, and Matthew Tancik. Lerf: Language embedded radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19729–19739, 2023. 4

[16] Daniel Khashabi, Yeganeh Kordi, and Hannaneh Hajishirzi. Unifiedqa-v2: Stronger generalization via broader cross-format training. 2022. 1

[17] Juil Koo, Chanho Park, and Minhyuk Sung. Posterior distillation sampling. In *CVPR*, 2024. 1, 2

[18] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *CVPR*, 2020. 6

[19] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*, 2022. 1

[20] Yuhan Li, Yishun Dou, Yue Shi, Yu Lei, Xuanhong Chen, Yi Zhang, Peng Zhou, and Bingbing Ni. Focaldreamer: Text-driven 3d editing via focal-fusion assembly. In *AAAI*, 2024. 1

[21] Kunhao Liu, Fangneng Zhan, Jiahui Zhang, Muyu Xu, Yingchen Yu, Abdulmotaleb El Saddik, Christian Theobalt, Eric Xing, and Shijian Lu. Weakly supervised 3d open-vocabulary segmentation. *NeurIPS*, 2023. 4, 5

[22] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11461–11471, 2022. 6

[23] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 2019. 1

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. 2023. 1

[25] Minghan Qin, Wanhua Li, Jiawei Zhou, Haoqian Wang, and Hanspeter Pfister. Langsplat: 3d language gaussian splatting. In *CVPR*, 2024. 5

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 1

[27] Litu Rout, Yujia Chen, Nataniel Ruiz, Constantine Caramanis, Sanjay Shakkottai, and Wen-Sheng Chu. Semantic image inversion and editing using rectified stochastic differential equations. *arXiv preprint arXiv:2410.10792*, 2024. 4, 6

[28] Jin-Chuan Shi, Miao Wang, Hao-Bin Duan, and Shao-Hua Guan. Language embedded 3d gaussians for open-vocabulary scene understanding. In *CVPR*, 2024. 5

[29] Yaya Shi, Xu Yang, Haiyang Xu, Chunfeng Yuan, Bing Li, Weiming Hu, and Zheng-Jun Zha. Emscore: Evaluating video captioning via coarse-grained and fine-grained embedding matching. In *CVPR*, 2022. 1

[30] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov,

Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. 2023. 1

[31] Christina Tsalicoglou, Fabian Manhardt, Alessio Tonioni, Michael Niemeyer, and Federico Tombari. Textmesh: Generation of realistic 3d meshes from text prompts. In *3DVS*, 2024. 1

[32] Patrick von Platen, Suraj Patil, Anton Lozhkov, Pedro Cuenca, Nathan Lambert, Kashif Rasul, Mishig Davaadorj, Dhruv Nair, Sayak Paul, William Berman, Yiyi Xu, Steven Liu, and Thomas Wolf. Diffusers: State-of-the-art diffusion models, 2022. 6

[33] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *TVCG*, 2023. 1, 7

[34] Feng Wang, Jieru Mei, and Alan Yuille. Sclip: Rethinking self-attention for dense vision-language inference. In *ECCV*, 2025. 4

[35] 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. *ECCV*, 2024. 1, 2, 4

[36] Xiaofeng Yang, Cheng Chen, Xulei Yang, Fayao Liu, and Guosheng Lin. Text-to-image rectified flow as plug-and-play priors. *arXiv preprint arXiv:2406.03293*, 2024. 4, 6

[37] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *CVPR*, 2024. 4