EditSplat: Multi-View Fusion and Attention-Guided Optimization for View-Consistent 3D Scene Editing with 3D Gaussian Splatting

Supplementary Material

Overview

This supplementary material introduces further details and experimental results of our proposed method, EditSplat.

- Sec. A introduces additional results of EditSplat, including comparison of CLIP score trends over iteration with baselines.
- Sec. B provides a detailed explanation of Multi-View Fusion Guidance (MFG), including the multi-view fusion process and the formulation of MFG.
- Sec. C elaborates on Attention-Guided Trimming (AGT) details, including preparing attention maps and further analysis of ablation study on pruning method, a key component of AGT, across iterations.
- Sec. D outlines the experimental setup, including implementation details and user study.

A. Additional Results

A.1. Extensive Results

We present extensive results to demonstrate the capability of EditSplat to handle a variety of scenarios, including large-scale scenes and complex text instructions, as shown in Fig. 9 and Fig. 10.

Video Results and Supplementary Files. To further demonstrate our method with additional results not included in the main and this supplementary paper, we provide rendered videos and a project page.

Comparison of CLIP Score Trends. We present a graph in Fig. 1 that illustrates the trends of CLIP [13] text-image directional similarity and CLIP text-image similarity across iterations for baselines on the "Face" scene in IN2N [5] with text prompt "Make his face resemble that of a marble sculpture." The graph highlights the optimization effectiveness and performance trajectory of EditSplat compared to the baselines throughout iterations.

The results demonstrate that EditSplat achieves superior semantic alignment with the given instructions and improved optimization efficiency compared to the baselines. Both EditSplat's CLIP text-image directional similarity and CLIP text-image similarity scores are the highest and increase significantly faster, indicating superior convergence efficiency. These results suggest that the AGT technique improves optimization efficiency, while the MFG editing process ensures multi-view consistency.

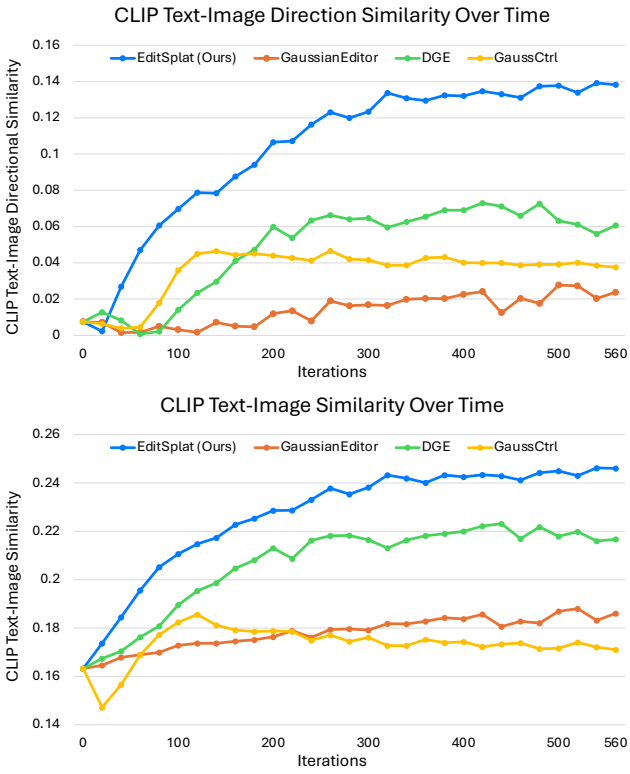

Figure 1. **Comparison of CLIP Scores Across Iterations.**. This figure compares the CLIP similarities between EditSplat and other baseline models over iterations. The results highlight the superior effectiveness of EditSplat in maintaining semantic alignment with the given instructions.

A.2. Comparison with NeRF-based Method

We compare our EditSplat framework with recent Neural Radiance Fields (NeRF)[11]-based approaches that were not included in the main paper's baselines, as those focused on state-of-the-art models that utilize 3D Gaussian Splatting (3DGS)[7]. Specifically, we evaluate EditSplat against InstructNeRF2NeRF (IN2N)[5], Vica-NeRF[4], and WatchYourSteps [12], as illustrated in Fig. 8.

The results demonstrate that NeRF-based methods produce outputs that are less aligned with target prompts and exhibit inferior editing quality compared to EditSplat. These methods often suffer from blurriness, artifacts, and minimal edits, with limited capability for precise local editing. In contrast, EditSplat achieves clear and high-quality rendered results with accurate local and global edits. Moreover, EditSplat completes the 3D editing process for the "Face" scene in the IN2N dataset in approximately 6 minutes, while NeRF-based methods require over 50 minutes,

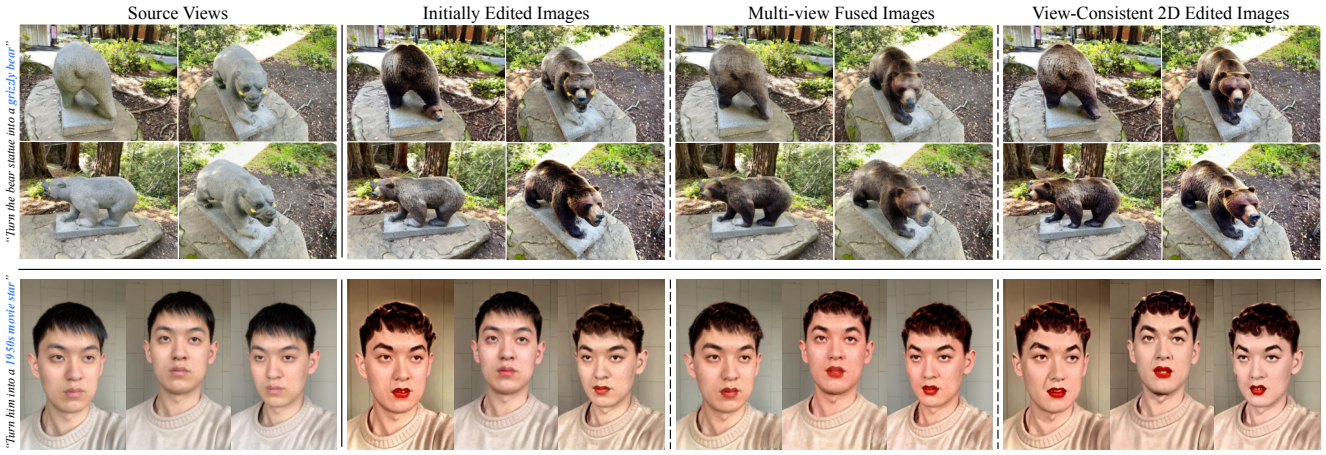

Figure 2. **Multi-View Fusion Guidance (MFG) 2D Editing Process.** MFG resolves multi-view inconsistency in the initially edited images through a multi-view fusion process that integrates information from multiple views to produce multi-view fused images. These fused images are subsequently incorporated into the diffusion process, ensuring consistent editing across all views.

with WatchYourSteps taking nearly four hours. These comparisons underscore EditSplat's superior efficiency, higher-quality edits, and better alignment with target prompts.

B. Multi-View Fusion Guidance (MFG) Details

B.1. Multi-View Fusion

Filtering. The ImageReward model [18] is trained on 137k pairs of expert comparisons and evaluates image fidelity and text-image alignment based on human preferences. This model's scoring capabilities surpass those of CLIP [13], Aesthetic [14], and BLIP [10], as demonstrated in the original paper. We leverage the ImageReward model to rank the initially edited images by their alignment with the text prompt and select the top 85% of high-scoring images. This filtering step improves the quality of the multi-view fused images, ensuring more accurate integration across views. We provide the ImageReward's capability in our filtering process in Fig. 7.

Depth-based Multi-View Fusion. Following the filtering step, we project the top-ranked initially edited images to each target view. Specifically, we select the top 5 adjacent views based on proximity to the target view. A naive projection from multiple views to a single target view leads to suboptimal results, including inaccuracies and blurriness caused by the improper handling of overlapping pixels. To address this, we employ an iterative alpha blending strategy guided by depth values, enhancing both consistency and accuracy in the fused results. The algorithm for multi-view fusion is detailed in Algorithm 1. In Algorithm 1, the input selected source data $\{(\mathcal{I}^i_{\text{src}}, \mathcal{D}^i_{\text{src}}, \xi^i_{\text{src}}, K^i_{\text{src}})\}_{i=1}^{N'}$ represents the selected initially edited images, their corresponding depth maps from 3DGS [6], extrinsic camera parameters, and intrinsic camera parameters, respectively. The target camera parameters $\{(\xi^j_{\text{trg}}, K^j_{\text{trg}})\}_{j=1}^{N}$ define the extrinsic ξ^j_{trg} and intrinsic K^j_{trg} parameters for each of the N target views. Here, N represents the total number of views, while N' denotes the number of selected images.

Algorithm 1: Depth-based Multi-View Fusion in MFG

Input : Selected source data $\{(\mathcal{I}^i_{\text{src}}, \mathcal{D}^i_{\text{src}}, \xi^i_{\text{src}}, K^i_{\text{src}})\}_{i=1}^{N'}$, Target camera parameters $\{(\xi^j_{\text{trg}}, K^j_{\text{trg}})\}_{j=1}^{N}$

Output: Multi-view fused images $\{\mathcal{I}^j_{\text{trg}} \in \mathbb{R}^{H\times W\times 3}\}_{j=1}^{N}$

foreach *target view* $j = 1$ *to* N **do**
  Select Nearest Source Views:
  $\mathcal{I}_{\text{sel}} \leftarrow$ Select 5 nearest views to ξ^j_{trg} from $\{\mathcal{I}^i_{\text{src}}\}_{i=1}^{N'}$
  Initialization:
  $\mathcal{I}^j_{\text{trg}} \leftarrow \mathbf{0}$ tensor of size $(3, H, W)$
  Reprojection:
  foreach *source view* i *in* $\mathcal{I}_{sel}$ **do**
    // P_i: 3D points, C_i: RGB colors.
    $(P_i, C_i) \in \mathbb{R}^{HW\times 3}, u_i \in \mathbb{R}^{HW\times 2}$
    $(P_i, C_i) \leftarrow \text{Reproject}(\mathcal{I}^i_{\text{src}}, \mathcal{D}^i_{\text{src}}, \xi^i_{\text{src}}, K^i_{\text{src}}, \xi^j_{\text{trg}}, K^j_{\text{trg}})$
    $u_i \leftarrow \text{MapToPixelCoordinates}(P_i)$
    // u_i within image bounds.
    // $\mathcal{I}_{\text{list}}, \mathcal{D}_{\text{list}}$: List of RGB, depth.
    $\mathcal{I}_{\text{list}}[i][u_i] \leftarrow C_i$, $\mathcal{D}_{\text{list}}[i][u_i] \leftarrow$ depth of P_i
  end
  Blending Based on Depth:
  Sort $(\mathcal{D}_{\text{list}}, \mathcal{I}_{\text{list}})$ in descending order by $\mathcal{D}_{\text{list}}$.
  foreach $(\mathcal{D}_l, \mathcal{I}_l)$ *in* $(\mathcal{D}_{list}, \mathcal{I}_{list})$ **do**
    if $l = 0$ **then**
      $w \leftarrow 1$
    else
      $w \leftarrow \frac{\mathcal{D}_l}{\mathcal{D}_l + \mathcal{D}_{prev}}$
    end
    $\mathcal{I}^j_{\text{trg}} \leftarrow (1 - w) \cdot \mathcal{I}_l + w \cdot \mathcal{I}^j_{\text{trg}}$
  end
end
return $\{\mathcal{I}^j_{trg}\}_{j=1}^{N}$

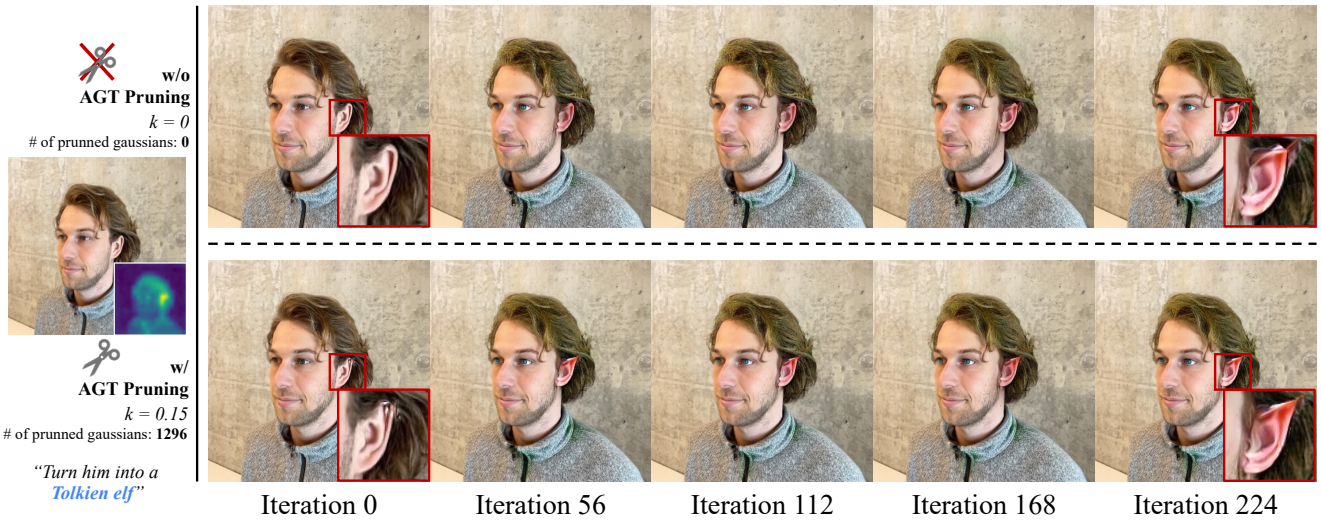

Figure 3. **Ablation Study on Pruning in AGT**. The top row illustrates the inefficiency of 3DGS optimization during editing. High-attention regions, which require significant modifications, remain under-edited due to the presence of numerous source Gaussians. In contrast, the application of pruning through AGT effectively removes Gaussians in high-attention areas, enabling more accurate and efficient editing. As demonstrated in the bottom row, this approach allows for better optimization even with fewer iterations.

Background Refinement. The initial multi-view fused images I_{trg} often exhibit a sparse background, while the target editing object in the image, which is often located in the center of the scene, appears dense. This issue arises due to the limitations of reprojection caused by discrepancies in camera viewpoints. To address this, we refine I_{trg} by replacing its sparse background with the background from the corresponding source image, using SAM [9] to preserve the original background. First, as the source image's object is generally misaligned with the target object's range, we extract a binary mask M_{trg} of the target object from I_{trg}:

$$M_{trg}(x,y) = \begin{cases} 1 & \text{if } (x,y) \text{ belongs to the target object,} \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Using M_{trg}, we isolate the background from the source image $\mathcal{I}_{\text{src}}$ as:

$$B_{src}(x,y) = \mathcal{I}_{\text{src}}(x,y) \cdot (1 - M_T(x,y)). \tag{2}$$

Finally, the refined multi-view fused image h_M is obtained by combining the target object from $\mathcal{I}_{\text{trg}}$ with the background from $\mathcal{I}_{\text{src}}$, ensuring seamless integration:

$$h_M(x,y) = \begin{cases} \mathcal{I}_{\text{trg}}(x,y), & \text{if } M_{trg}(x,y) = 1, \\ \mathcal{I}_{\text{src}}(x,y), & \text{if } M_{trg}(x,y) = 0. \end{cases} \tag{3}$$

This process effectively replaces the sparse background in $\mathcal{I}_{\text{trg}}$ while preserving the target object, resulting in smoother and more cohesive multi-view fused images h_M.

We illustrate the intermediate results of the MFG editing process in Fig. 2. The initially edited images exhibit misaligned edits with the text prompt and lack consistency across views. In contrast, the multi-view fused images produced through the multi-view fusion process are well-aligned with the text prompt, consistent across views, and incorporate multi-view information. Finally, the 2D edited images ensure multi-view consistency and precise alignment with the text prompt.

B.2. Alignment with Multi-View Information

To seamlessly incorporate multi-view information during the editing process, we extend the classifier-free guidance method. This allows the integration of multi-view fusion details into the diffusion process to facilitate multi-view consistent editing. Below is our Multi-view Fusion Guidance, an extended score estimate for multi-view aligned editing with classifier-free guidance, as specified in the main paper:

$$\begin{aligned} \tilde{\epsilon_\theta}\left(z_t, h_S, h_T, h_M\right) = \; & \epsilon_\theta\left(z_t, \varnothing, \varnothing\right) \\ & + s_T\left(\epsilon_\theta\left(z_t, h_M, h_T\right) - \epsilon_\theta\left(z_t, h_M, \varnothing\right)\right) \\ & + s_M\left(\epsilon_\theta\left(z_t, h_M, \varnothing\right) - \epsilon_\theta\left(z_t, \varnothing, \varnothing\right)\right) \\ & + s_S\left(\epsilon_\theta\left(z_t, h_S, \varnothing\right) - \epsilon_\theta\left(z_t, \varnothing, \varnothing\right)\right), \end{aligned} \tag{4}$$

Here, h_M represents the multi-view fusion image, h_S is the source image, and h_T corresponds to the text prompt. The guidance strength for each conditioning is modulated by the respective scale factors s_M, s_S, and s_T.

Conditional Probability Formulation. In our approach, the conditional probability distribution $P(z|h_M, h_S, h_T)$ can be expressed as:

$$\begin{aligned} P(z|h_M, h_S, h_T) &= \frac{P(z, h_M, h_S, h_T)}{P(h_M, h_S, h_T)} \\ &= \frac{P(h_T|h_M, h_S, z)P(h_M|h_S, z)P(h_S|z)P(z)}{P(h_M, h_S, h_T)} \end{aligned} \tag{5}$$

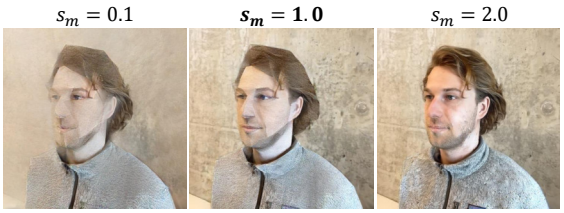

Figure 4. **Influence of Guidance Scale s_M.** Qualitative analysis of s_M in MFG for 3D editing. Higher s_M improves multi-view consistency, but excessively large value reduces the editing effect.

Figure 5. **Influence of w_{thres}.** Qualitative comparison of w_{thres} in AGT for local editing. Lower w_{thres} fails to preserve original structures, while higher values overly restrict the editing scope.

Log Probability and Score Estimation. Taking the logarithm of the conditional probability results in:

$$\begin{aligned}\log(P(z|h_M,h_S,h_T)) &= \log(P(h_T|h_M,h_S,z)) \\ &+ \log(P(h_M|h_S,z)) \\ &+ \log(P(h_S|z)) \\ &+ \log(P(z)) \\ &- \log(P(h_M,h_S,h_T)) \end{aligned} \tag{6}$$

Calculating the gradient with respect to z and rearranging terms, we obtain:

$$\begin{aligned}\nabla_z \log(P(z|h_M,h_S,h_T)) &= \nabla_z \log(P(z)) \\ &+ \nabla_z \log(P(h_S|z)) \\ &+ \nabla_z \log(P(h_M|h_S,z)) \\ &+ \nabla_z \log(P(h_T|h_M,h_S,z)) \end{aligned} \tag{7}$$

Guidance Interpretation. Each guidance scale (e.g., s_M, s_S, and s_T) effectively shifts the probability mass toward outputs that align with the corresponding conditioning. For instance, s_M biases the implicit classifier p_θ toward assigning higher probabilities to multi-view information, thereby ensuring consistent 2D editing across views and enhancing the quality of the resulting 3D edits (see Fig. 4). However, excessively increasing s_M reduces the influence of the text guidance scale s_T, resulting in less pronounced editing effects. Conversely, s_S preserves the original content from the source image, and s_T promotes adherence to the provided text prompts. By carefully balancing these guidance scales, our model effectively achieves multi-view consistent edits that accurately reflect both structural and color details specified by textual instructions.

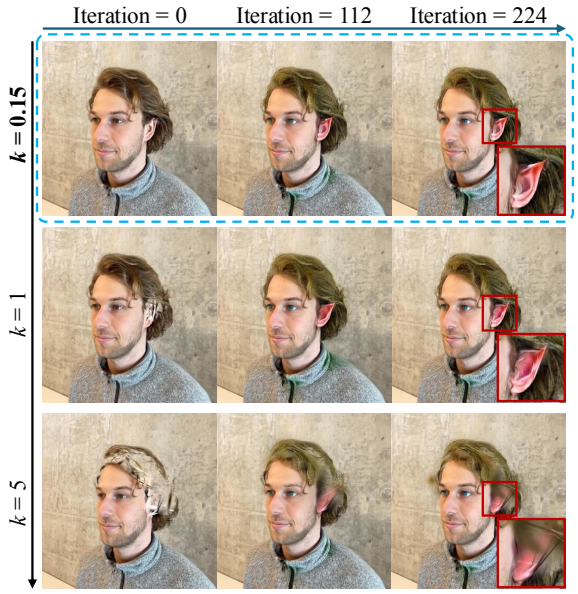

Figure 6. **Ablation study on pruning threshold k in AGT.** The figure illustrates the training progress under different pruning thresholds k using the same source images and text prompt as in Fig. 3. Excessively high values of k result in the removal of too many Gaussians, hindering the editing process.

C. Attention-Guided Trimming (AGT) Details

C.1. Extracting Attention Map

When the latent representation of the input image added noise and the text prompt are fed into the diffusion model for denoising, we select the semantic keyword that represents the intended editing outcome (e.g., "autumn" in the instruction "Make it autumn" or "clown" in "Turn him into a clown"). We then extract all cross-attention maps associated with this keyword, which are computed during the MFG editing process. These attention maps are resized to rendering resolution using bilinear interpolation, aggregated, and normalized to the [0, 1] range using Min-Max normalization. This process allows us to accurately assign these semantic maps to each Gaussian, ensuring they contain meaningful regions for pruning, facilitating efficient optimization, and selectively optimizing semantically rich local editing.

C.2. Qualitative Analysis over Iterations

We further validate the effectiveness of AGT by analyzing the editing results across iterations. Notably, the ablation study presented here analyzes the effect of pruning independently within AGT while maintaining its selective optimization for local editing.

Fig. 3 highlights the contrast in edited results with and without pruning the Gaussians. Based on the instruction *"Turn him into a Tolkien elf"*, the attention map on the far left identifies the ear as the most prominent region, requiring significant modifications in the source scene. For optimal editing, our AGT first assigns attention weights to each

source Gaussians. Then, the top 0.15% of Gaussians (1,296 in total) based on assigned weights, are pruned. The rendered image in the bottom row at iteration 0 shows that the upper part of the ear has been cleared. As a result, the ear is edited more effectively into the desired elf ear. In contrast, the top row shows that the remaining source Gaussians in the unpruned case interfere with the convergence.

Fig. 6 illustrates how editing quality varies with different pruning threshold $k\%$. Especially in the third row, pruning the top 5% of Gaussians results in excessively empty regions, requiring additional iterations to fill these gaps. Conversely, when $k\%$ is appropriately set to 0.15%, optimal editing is achieved. Through heuristic analysis across various scenes, we find that pruning the top 0.15% of Gaussians is the most suitable approach for overall scenes.

In addition, both Fig. 3 and Fig. 6 demonstrate that the attention weights assigned by AGT effectively reflect the semantic importance of the editing regions. Given the text instruction *"Turn him into a Tolkien elf"*, the ear, rather than the head, should undergo more significant changes. Our results highlight that our AGT accurately reflects this semantic importance during editing and that pruning an appropriate percentage of Gaussians ensures effective convergence toward the target.

C.3. Analysis of Local Editing

We provide an analysis of local editing in Attention-Guided Trimming (AGT), focusing specifically on the impact of the threshold w_{thres} for semantic local editing. As illustrated in Fig. 5, varying w_{thres} significantly affects the balance between preserving original scene details and effectively applying the intended semantic edits.

When the threshold w_{thres} is set too low, attention-guided selection becomes excessively inclusive, causing unintended modifications beyond the target area. Consequently, critical structural elements of the original scene—such as the table in Fig. 5—fail to be properly preserved. In contrast, higher values of w_{thres} significantly narrowing the scope of editing. Although this helps maintain structural integrity by protecting essential regions, it simultaneously limits the effectiveness of the semantic editing. As demonstrated in our qualitative evaluation, the edits within regions of secondary semantic importance (e.g., grass or background foliage in Fig. 5) become overly constrained, resulting in less pronounced visual changes and diminishing the overall impact of the edit.

Therefore, selecting an appropriate w_{thres} is critical to achieving an optimal trade-off between preserving essential structures of the original scene and effectively implementing localized semantic editing. Our experiments reveal that intermediate values of w_{thres} successfully balance these competing objectives, enabling precise and semantic localized editing.

Figure 7. **Filtering** To prevent misaligned images from negatively affecting MFG, we leverage ImageReward [18] to rank the initially edited images. Only the top 85% of these images, based on their rankings, are utilized for MFG, ensuring high-quality inputs.

D. Experiment Set Up

D.1. Implementation Details

EditSplat utilizes the vanilla 3DGS [6] framework for 3D reconstruction. For each scene, we train the model for 30,000 iterations to serve as the source scene. All 3DGS training phases employ the Adam optimizer [8], an identical learning rate with vanilla 3DGS. We apply the same densification strategy across all scenes, with a densification interval of 100 and a gradient threshold of 0.01.

For the 2D image editor, we use InstructPix2Pix(IP2P) [3] from Diffusers library [16] with our novel method, MFG. In a more detailed setup, we perform 20 sampling steps using the DDIM scheduler [15], with noise sampled from $t \in [0.7,\ 0.98]$.

D.2. Evaluation

Quantitative. We adopt a train/test split for our datasets following the methodology suggested by Mip-NeRF360 [2], taking every 8th image for test. Our evaluation metrics include measuring text-image directional similarity and text-image similarity using CLIP [13]. The text descriptions used for calculating CLIP similarity are detailed in Tab. 1. In Tab. 2, we summarize the instructions employed for 3D editing. For one of our comparison baselines, GaussCtrl [17], which does not support instruction-based editing, Tab. 2 provides the source and target scene descriptions used in its evaluation. Additionally, since GaussCtrl leverages a different 2D image editor, such as ControlNet [19], we set its guidance scale to the recommended value of 5.

User Study. To ensure a fair and rigorous user study, we recruit 100 participants through Amazon Mechanical Turk [1], a widely-used platform for human evaluation. Participants are provided with images rendered from the source scene along with the corresponding text prompt used for editing. They are tasked with evaluating how well the edited images align with the given text prompt and assessing the overall quality of the edits. The choices include rendered views edited by our method and those generated by baseline models, all presented in a randomized order to prevent participants from inferring which model produced which images. Furthermore, the randomization is applied separately for each scene to ensure unbiased evaluation across different examples. This setup allows us to measure both the semantic accuracy of the edits and the perceived quality of the rendered images, as shown in Fig. 11.

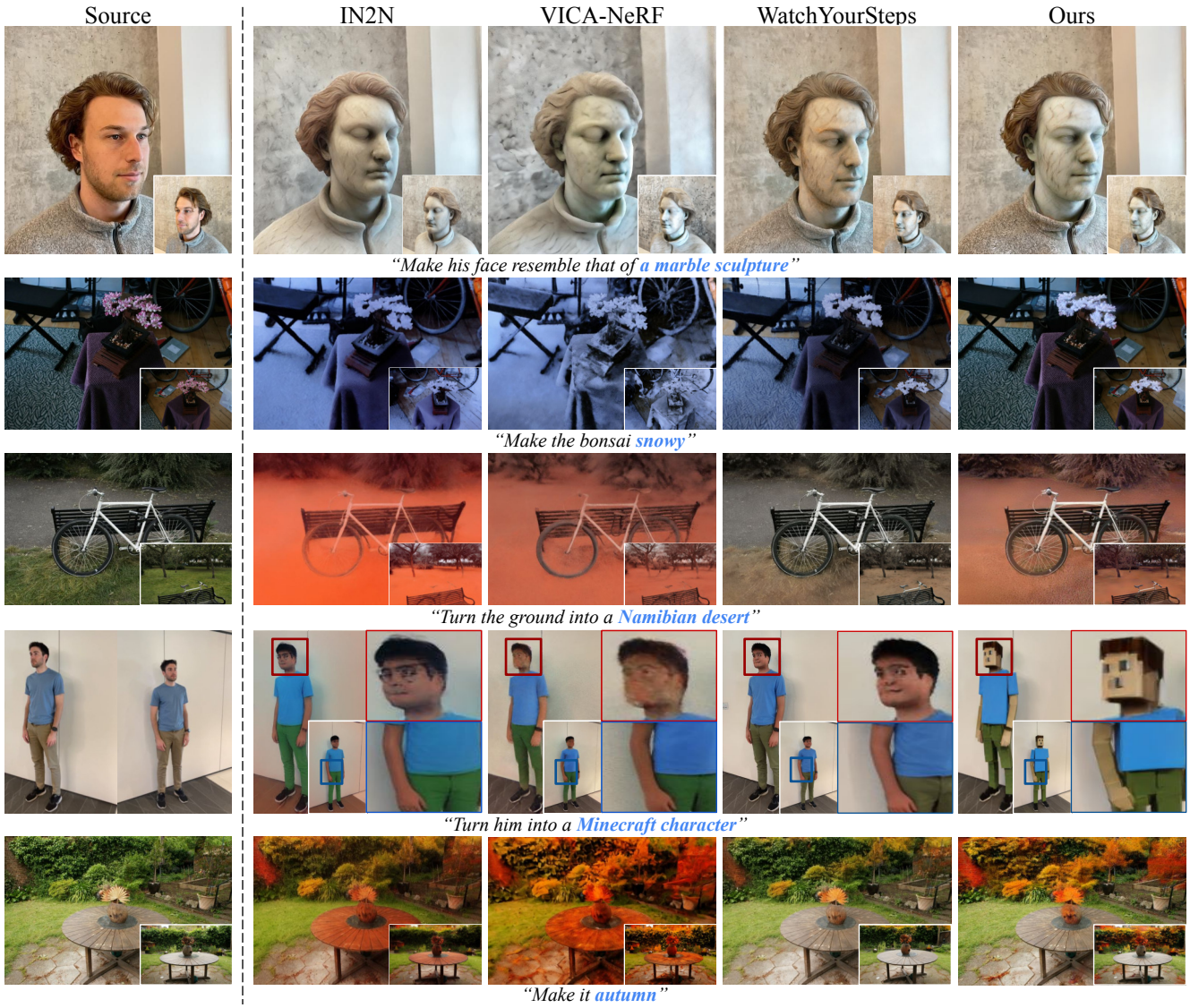

Figure 8. **Qualitative Comparison with NeRF-Based Methods.** EditSplat provides more intense and precise editing compared to recent NeRF-based methods. The leftmost column displays the source images, while the subsequent columns show rendered images. To evaluate multi-view consistency, different views of the corresponding images are included in each corner. Notably, EditSplat demonstrates superior performance in both local and global editing tasks.

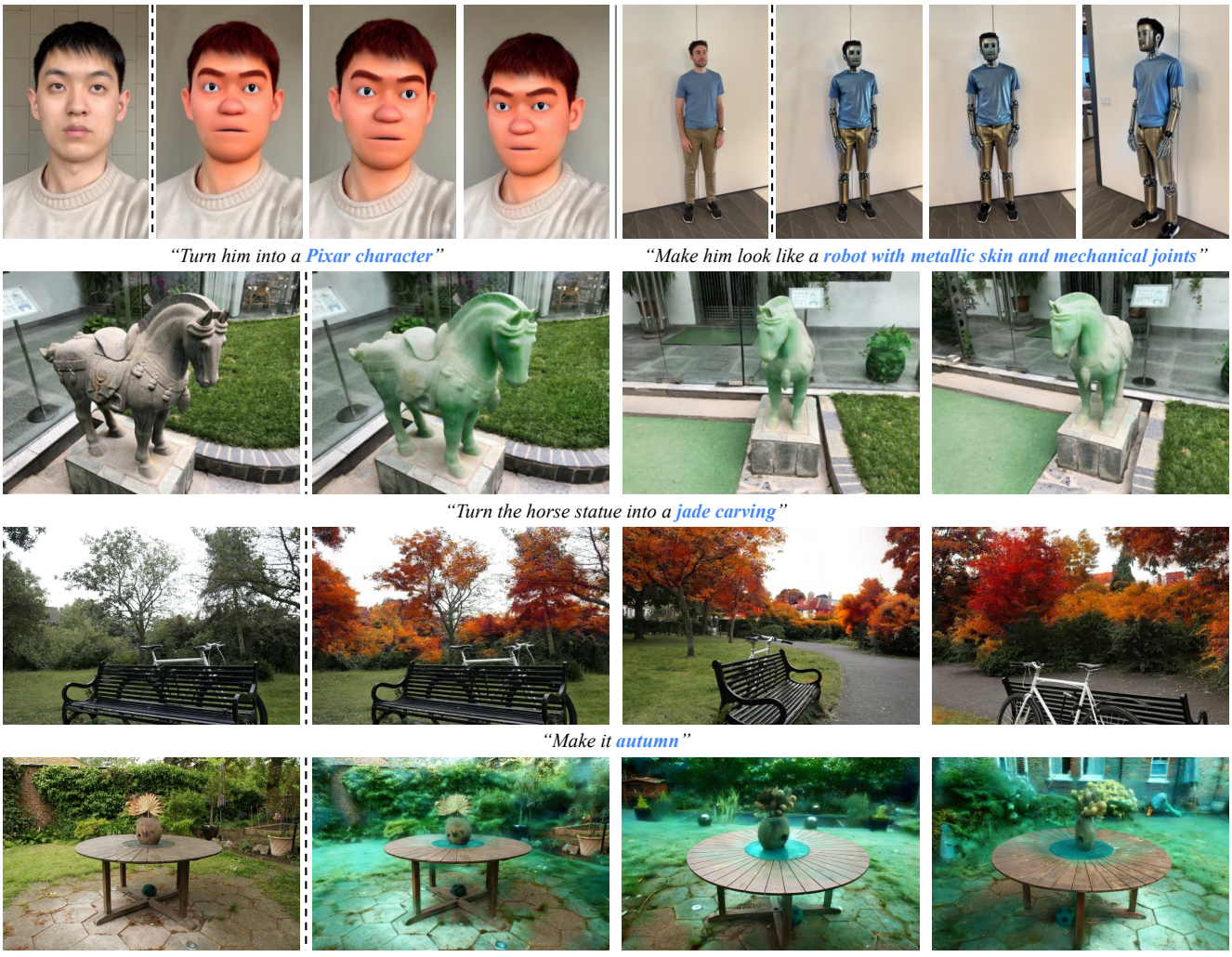

Figure 9. **Additional extensive results 1**. We present extensive qualitative results to highlight the robustness and versatility of our proposed method. Our EditSplat ensures multi-view consistency and provides flexible editing, ranging from fine-grained modifications to global stylization.

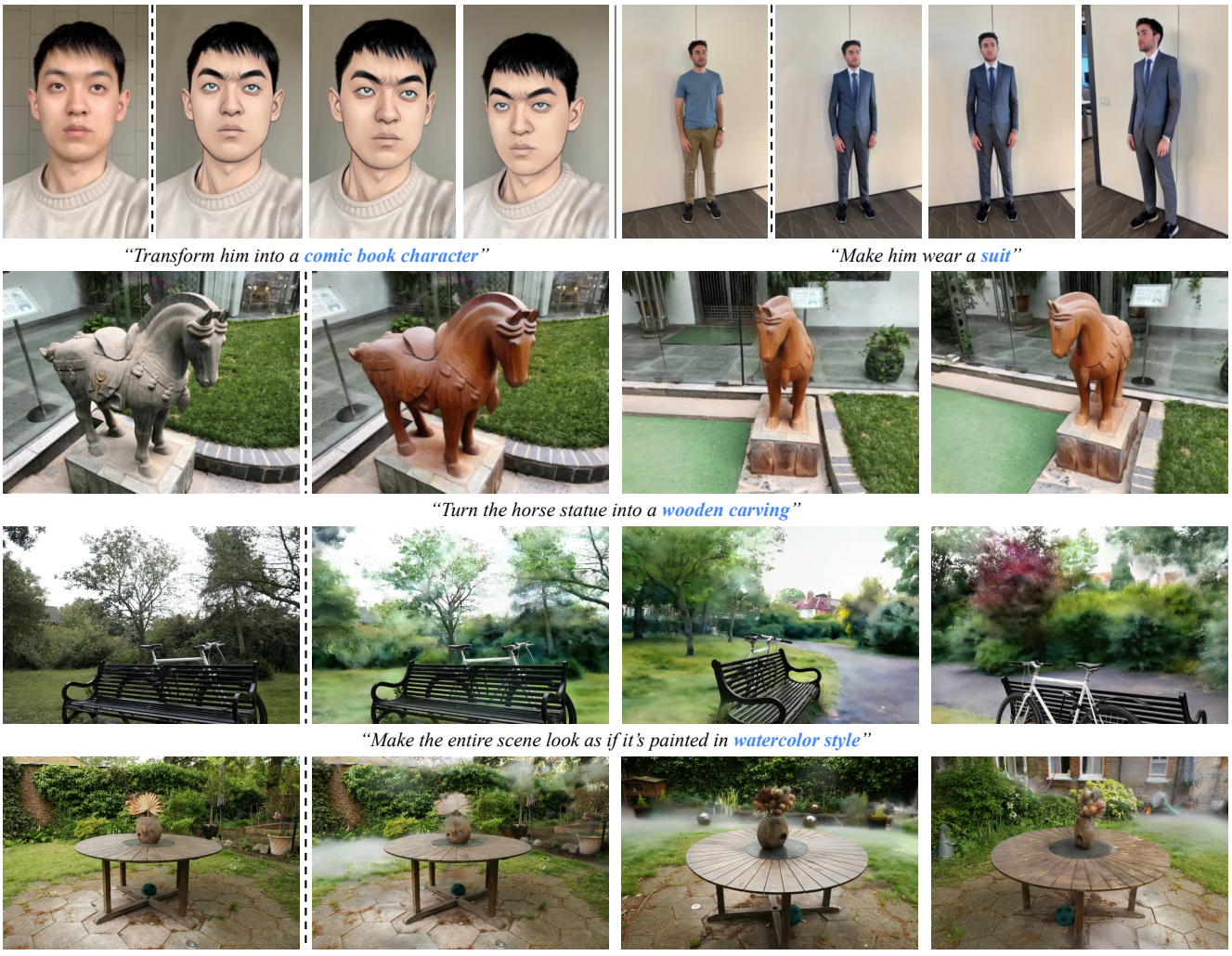

Figure 10. **Additional extensive results 2**. We present extensive qualitative results to highlight the robustness and versatility of our proposed method. Our EditSplat ensures multi-view consistency and provides flexible editing, ranging from fine-grained modifications to global stylization.

Scene	Source Description	Target Description	Editing Instruction
bear	a photo of a bear	a photo of a wild boar	Turn the bear statue into a wild boar
bear	a photo of a bear	a photo of a metallic robot bear	Turn the bear statue into a metallic robot
bicycle	a photo of a park	a photo of a Namibian desert	Turn the ground into a Namibian desert
bicycle	a photo of a park	a watercolor style paint of a park	Make the entire scene look as if it's painted in watercolor style
bonsai	a photo of a plant	a photo of a plant, snowy	Make the bonsai snowy
bonsai	a photo of a plant	a photo of a plant made of paper	Change the bonsai to look like it's made of paper, folded into intricate origami shapes
face	a photo of a face of a man	a photo of a marble sculpture	Make his face resemble that of a marble sculpture
face	a photo of a face of a man	a photo of a face of a man, made of paper	Make him appear like he's made of paper with folded edges
fangzhou	a photo of a face of a man	a photo of a face of Jocker	Turn him into a Jocker
fangzhou	a photo of a face of a man	a photo of a face of Steve Jobs	Turn him into a Steve Jobs
garden	a photo of an outdoor garden	a photo of an outdoor garden in autumn	Make it autumn
garden	a photo of an outdoor garden	a photo of a garden in underwater	Make the scene appear as though it's underwater
person	a photo of a person	a photo of a person wearing a suit	Make him wear a suit
person	a photo of a person	a photo of a person in Minecraft	Turn him into a Minecraft character
stone horse	a photo of a horse statue	a photo of a horse made of wood	Turn the horse statue into a wooden carving
stone horse	a photo of a horse statue	a photo of a horse made of jade	Turn the stone horse into a jade carving

Table 1. **Details of the descriptions used for the CLIP similarity**. CLIP similarity is computed as the cosine similarity between embeddings in the CLIP space. The source description depicts the scene before editing, while the target description represents the desired edited scene. Both descriptions are transformed into text embeddings in the CLIP space and are used to evaluate the semantic alignment of the 3D scene.

Scene	Editing Instruction	Source Description	Target Description
bear	Turn the bear statue into a wild boar	a photo of a bear statue in the forest	a photo of a wild boar in the forest
bear	Turn the bear statue into a metallic robot	a photo of a bear statue in the forest	a photo of a metallic robot in the forest
bicycle	Turn the ground into a Namibian desert	a photo of a bicycle at grass	a photo of the bicycle at the namibian desert
bicycle	Make the entire scene look as if it's painted in watercolor style	a photo of a bicycle at grass	a photo of a bicycle scene as if it's painted in watercolor style
bonsai	Make the bonsai snowy	a photo of a bonsai in the desk	a photo of a snowy bonsai
bonsai	Change the bonsai to look like it's made of paper, folded into intricate origami shapes	a photo of a bonsai in the desk	a photo of a tree made of paer, folded into intricate origianl shapes
face	Make his face resemble that of a marble sculpture	a photo of a face of a man	a photo of a marble sculpture
face	Make him appear like he's made of paper with folded edges	a photo of a face of a man	a photo of a man made of paper with folded edges
fangzhou	Turn him into a Jocker	a photo of a face of a man	a photo of a Joker
fangzhou	Turn him into a Steve Jobs	a photo of a face of a man	a photo of a Steve Jobs
garden	Make it autumn	a photo of a fake plant on a table in the garden	a photo of a garden scene with autumn
garden	Make the scene appear as though it's underwater	a photo of a fake plant on a table in the garden	a photo of a garden scene appear as though underwater
person	Make him wear a suit	a photo of a person	a photo of a man wearing a suit
person	Turn him into a Minecraft character	a photo of a person	a photo of a Minecraft character
stone horse	Turn the horse statue into a wooden carving	a photo of a stone horse statue in front of the museum	a photo of a wooden carving statue in front of the museum
stone horse	Turn the stone horse into a jade carving	a photo of a stone horse statue in front of the museum	a photo of a stone horse of a jade carving

Table 2. **Detailed text prompts used for 3D scene editing**. While models using IP2P as a 2D image editor can perform editing directly based on instructions, model employing inversion-based 2D image editor such as GaussCtrl [17] requires both source and target descriptions. To ensure a fair comparison, we designed the source and target descriptions to ensure that the semantic difference between source and target accurately reflects the given editing instructions.

Figure 11. **User Study Survey Form**. Participants are presented with the source 3D scene and four edited scenes, which include results from three baselines and our EditSplat. These edited scenes are randomly shuffled for each question to prevent bias. Participants evaluate the edits based on how well they align with the given text prompt and their overall quality.

References

[1] Amazon mechanical turk. https://www.mturk.com/, 2005. 6

[2] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5470–5479, 2022. 5

[3] Tim Brooks, Aleksander Holynski, and Alexei A Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18392–18402, 2023. 5

[4] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. *Advances in Neural Information Processing Systems*, 36, 2024. 1

[5] Ayaan Haque, Matthew Tancik, Alexei A Efros, Aleksander Holynski, and Angjoo Kanazawa. Instruct-nerf2nerf: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19740–19750, 2023. 1

[6] 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, 2023. 2, 5

[7] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42 (4), 2023. 1

[8] Diederik P Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[9] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023. 3

[10] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 2

[11] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, 65(1):99–106, 2021. 1

[12] Ashkan Mirzaei, Tristan Aumentado-Armstrong, Marcus A Brubaker, Jonathan Kelly, Alex Levinshtein, Konstantinos G Derpanis, and Igor Gilitschenski. Watch your steps: Local image and scene editing by text instructions. In *European Conference on Computer Vision*, pages 111–129. Springer, 2025. 1

[13] 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 *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1, 2, 5

[14] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 2

[15] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 5

[16] 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. https://github.com/huggingface/diffusers, 2022. 5

[17] 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. *arXiv preprint arXiv:2403.08733*, 2024. 5, 11

[18] 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. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 5

[19] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3836–3847, 2023. 5