3DOT: Texture Transfer for 3DGS Objects from a Single Reference Image

Xiao Cao[1] **Beibei Lin**[1] **Bo Wang**[2] **Zhiyong Huang**[1] **Robby T. Tan**[1,3]
[1] National University of Singapore [2] University of Mississippi [3] ASUS Intelligent Cloud Services
{xiaocao, beibei.lin}@u.nus.edu
hawk.rsrch@gmail.com {dcshuang,robby.tan}@nus.edu.sg

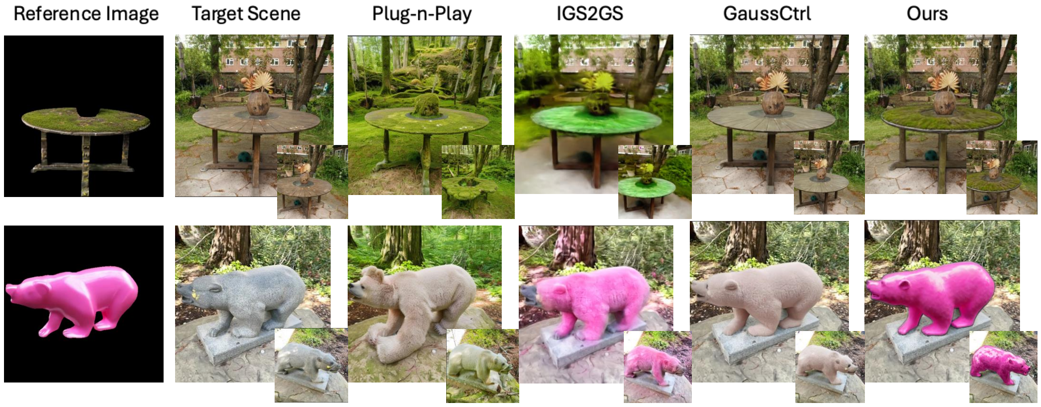


Figure 1: Comparison of 2D and 3D image-based texture editing methods. Prompts are "**moss-covered table**" and "**pink plastic bear**". 2D methods Plug-n-Play [36] suffers from view inconsistency problem; 3D text-driven editing methods IGS2GS [37] and GaussCtrl [41] struggle to preserve texture characteristics. Ours faithfully edit the texture, material appearance, and color.

Abstract

Image-based 3D texture transfer from a single 2D reference image enables practical customization of 3D object appearances with minimal manual effort. Adapted 2D editing and text-driven 3D editing approaches can serve this purpose. However, 2D editing typically involves frame-by-frame manipulation, often resulting in inconsistencies across views, while text-driven 3D editing struggles to preserve texture characteristics from reference images. To tackle these challenges, we introduce **3DOT**, a **3D** Gaussian Splatting **O**bject **T**exture Transfer method based on a single reference image, integrating: 1) progressive generation, 2) view-consistency gradient guidance, and 3) prompt-tuned gradient guidance. To ensure view consistency, progressive generation starts by transferring texture from the reference image and gradually propagates it to adjacent views. View-consistency gradient guidance further reinforces coherence by conditioning the generation model on feature differences between consistent and inconsistent outputs. To preserve texture characteristics, prompt-tuning-based gradient guidance learns a token that describes differences between original and reference textures, guiding the transfer for faithful texture preservation across views. Overall, **3DOT** combines these strategies to achieve effective texture transfer while maintaining structural coherence across viewpoints. Extensive qualitative and quantitative evaluations confirm that our three components enable convincing and effective 2D-to-3D texture transfer. Our project page is available here: `https://massyzs.github.io/3DOT_web/`.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

1 Introduction

Transferring texture from a 2D image to a 3D object is a valuable yet underexplored capability in 3D editing. It enables efficient texture manipulation and benefits applications such as virtual reality, CG films, and 3D games [2, 31, 40, 23, 28, 39, 34, 5]. Despite advances in 2D texture and 3D editing techniques, transferring texture from a single 2D image to a 3D object remains challenging due to difficulties in ensuring view consistency and preserving texture characteristics, particularly for unseen views beyond the reference image.

2D image-based editing methods [36, 46, 25, 32, 14, 51, 24, 42] perform texture transfer by finetuning a diffusion model (e.g., DreamBooth [30], Textual Inversion [13]) and editing images rendered from a 3D object to create a finetuning dataset. The resulting 3D object often suffers from view inconsistency and identity loss due to the absence of constraints enforcing multi-view coherence and identity preservation, as shown in Figure 1. *3D editing* methods [15, 37, 9, 41, 8, 27, 6], especially text-driven ones, guide editing using prompts derived from reference images via visual language models or manual descriptions. However, these prompts are typically coarse and miss fine-grained features, resulting in identity mismatch and inconsistent appearance across views.

Motivated by these challenges, we propose **3DOT**, a novel framework for transferring texture from a single 2D reference image to a 3D object represented by 3D Gaussian Splatting [21]. *3DOT* comprises three key components: 1) a progressive generation process, 2) view-consistency gradient guidance, and 3) prompt-tuning-based gradient guidance. The first two components enforce view consistency, while the third preserves texture characteristics.

In the progressive generation process, we first obtain reference images either by directly pasting the reference image onto the 3D object or by generating candidate views using a depth-conditioned model [47] based on the unedited view's depth. The image that best matches the target attributes is then selected. To facilitate prompt tuning and sparse cross-attention, we remove backgrounds from both the unedited training images and the reference images, and project them into the latent space for k-step partial diffusion. The generation begins from the reference view and progressively propagates to neighboring views, guided by sparse cross-attention on previously edited views. This strategy maximizes overlap between adjacent reference images to enforce view consistency.

To enhance view consistency in 3D editing, we introduce view-consistency gradient guidance. The core idea is to guide the diffusion model toward view-consistent generation by minimizing texture inconsistency features in intermediate outputs. Specifically, we initialize two diffusion modules: one conditioned on reference views via cross-attention, and the other guided only by a text prompt. Since cross-attention is the only differing component, the discrepancy between their intermediate results captures view-consistency features. During each denoising step, these features are scaled and injected as gradient guidance, steering the generation toward consistent outputs across views.

Since the reference image reveals no texture for unseen views, coarse text prompts often lead to inconsistency. To overcome this, we propose prompt-tuning-based gradient guidance that captures texture differences as additional prompt tokens. Specifically, we compute the difference between reference and unedited images in the CLIP feature space [11], encoding the texture transformation direction. This signal is injected into the diffusion denoising process as gradient guidance, enabling consistent texture transfer across views. The fine-tuned prompt improves style coherence in unseen views while preserving details in the reference view.

We evaluate our method on the face-forwarding [38] and 360-degree [3] datasets. Results show effective texture transfer with fine detail preservation and strong view consistency. Our key contributions:

- **3DOT**, an image-based 3D Guassian Splatting (3DGS) texture transfer framework that enables efficient and flexible texture editing.
- A progressive generation process with view-consistency gradient guidance to address view inconsistency across novel views.
- Prompt-tuning-based gradient guidance preserves texture characteristics in seen views and enforces style consistency in unseen views.
- Extensive experiments demonstrate that **3DOT** achieves state-of-the-art visual quality and quantitative performance.

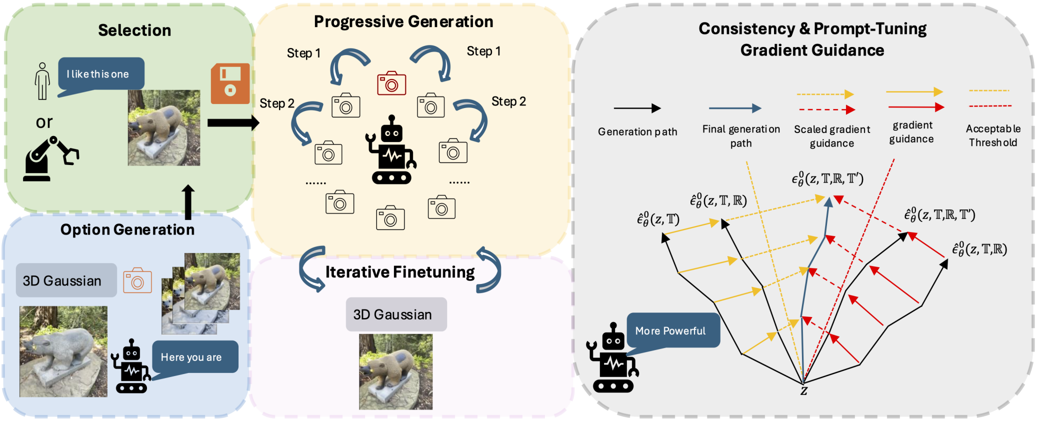

Selection
I like this one
or
Progressive Generation
Step 1
Step 1
Step 2
Step 2
......
......
Option Generation
3D Gaussian
Here you are
Iterative Finetuning
3D Gaussian
Consistency & Prompt-Tuning Gradient Guidance
Generation path
Final generation path
Scaled gradient guidance
gradient guidance
Acceptable Threshold
$\epsilon_\theta^0(z, \mathbb{T}, \mathbb{R}, \mathbb{T}')$
$\hat{\epsilon}_\theta^0(z, \mathbb{T}, \mathbb{R})$
$\hat{\epsilon}_\theta^0(z, \mathbb{T})$
$\hat{\epsilon}_\theta^0(z, \mathbb{T}, \mathbb{R}, \mathbb{T}')$
$\hat{\epsilon}_\theta^0(z, \mathbb{T}, \mathbb{R})$
More Powerful
z

Figure 2: **3DOT.** Our framework enables texture transfer from a single image to a 3D object. The left panels illustrate the selection of the reference image using a generative approach. Then, our method employs a progressive generation process guided by view-consistency and prompt-tuning-based gradient guidance to preserve both cross-view consistency and texture identity. $\mathbb{R}$, $\mathbb{T}$, and $\mathbb{T}'$ denote the reference set, text prompt, and learned texture difference token, respectively.

2 Related Work

2D Diffusion-based Editing DragDiffusion [32] defines target edits using keypoints and replaces them with reference images. A-Tale-of-Two-Features [46] combines dense DINO [7] features and sparse diffusion features by merging reference-view semantics with target-view structures. Plug-and-Play [36] refines fine-grained details by injecting diffusion features into DINO features, while DiffEditor [25] improves 2D editing precision via differential equation-based sampling with regional gradient guidance. The most relevant work, SwapAnything [14], leverages DreamBooth [30] and AdaIN [19] to encode source images and maintain style consistency during 2D edits. Although effective for image editing, these methods operate on individual views without enforcing view consistency, highlighting the need for 3D-aware texture editing techniques.

3D Editing Most 3D editing methods leverage 2D diffusion models for guidance and adopt dataset-updating strategies to finetune pretrained 3D scenes. Instruct-NeRF2NeRF [15] and Instruct-GS2GS [37] use instruct-pix2pix [4] to guide updates for NeRF or Gaussian Splatting. GaussianEditor [9] introduces hierarchical representations for more stable edits under stochastic guidance. Direct Gaussian Editor (DGE) [8] addresses view consistency via epipolar cross-attention, but its initial independent generation introduces artifacts. GaussCtrl [41] injects features from unedited views to preserve consistency, but this can cause the diffusion model to retain original textures, limiting editability. StyleSplat [20] achieves texture edits without a generative model but requires altering the 3D representation, which falls outside our setting of editing a fixed 3D object using a single reference image. Methods that ignore view consistency [15, 37, 9] can be extended with image captioning, while consistency-aware approaches [8, 41] can inject latent reference features during denoising. However, such modifications offer only coarse control. High-quality, identity-preserving edits require more precise and targeted designs.

3 Proposed Method

Fig. 2 illustrates our **3DOT** pipeline, consisting of three key modules: 1) a progressive generation process, 2) view-consistency gradient guidance for enforcing texture coherence across different views, and 3) prompt-tuning-based gradient guidance for preserving object identity.

To obtain the reference image, we either generate depth-conditioned candidates or extract textures by directly cropping texture into object shape in a certain rendered view. In the generative approach, users select the candidate that best matches the desired attributes. In the texture-based approach, extracted textures are directly mapped onto the object surface. Following [41], both reference and unedited images are encoded into latent space to initialize the denoising process. We then apply prompt-tuning to capture texture differences between the reference and the 3D object, guiding diffusion to preserve identity. Edited views are progressively generated, starting from the reference view. The resulting dataset is utilized to finetune 3D Gaussian model and the above procedure is iteratively conducted [15] for smooth results.

3.1 Progressive Generation

Existing methods [15, 37, 9, 41, 8] struggle to balance view consistency and editing flexibility. For example, GaussCtrl [41] conditions diffusion on unedited images to enforce consistency but often retains original textures, limiting editability. DGE [8] avoids reliance on unedited inputs but edits non-adjacent views, introducing inconsistencies.

To overcome these limitations, we propose a progressive generation process that removes dependency on unedited images and avoids isolated generation steps, achieving both consistency and flexibility.

We first generate reference images by conditioning a generative model on depth maps with background masking to ensure geometric alignment. To improve quality, we refine depth maps using dilated and blurred masks to address black-edge artifacts and apply the original mask to remove redundant content from the outputs.

For a selected reference view τ and target view $\mathbb{I}_i$, we construct a sparse reference set $\mathbb{R}_i = \{\mathbb{I}_\tau, \mathbb{I}_{i-1}, \mathbb{F}(\mathbb{I})_\tau\}$, excluding backgrounds. Including $\mathbb{I}_{i-1}$ maintains local consistency via minimal angular changes. As edits propagate to distant views, errors accumulate. For symmetric case, we can include $\mathbb{F}(\mathbb{I})_\tau$, a horizontally flipped variant of the reference, to preserve alignment with fewer conditioning views.

The generative model is conditioned on $\mathbb{R}$ using weighted fused cross-attention:

$$\mathrm{WeightedAttn}_e = \lambda \mathrm{Attn}_{e,e} + (1-\lambda) \sum_{i \in \mathbb{R}} w_i \mathrm{Attn}_{e,i}, \tag{1}$$

where e denotes the image that is currently editing, $\mathrm{Attn}_{i,j}$ denotes the attention score between images i and j, and λ balances self- and cross-attention.

Partial denoising (Eq. 3) begins with the reference view and progressively extends to adjacent views. These edited, view-consistent images are then used to finetune the 3D Gaussian model, and the process is repeated iteratively.

3.2 View-Consistency Gradient Guidance

Existing generative methods [47, 45, 30, 13] rely on many reference views to maintain consistency. In contrast, our progressive generation begins with a single reference and uses only a few views. To enhance cross-attention effectiveness under this constraint, we propose a consistency-aware gradient guidance mechanism inspired by classifier-free guidance [16], modifying the noise estimate [16] to amplify cross-view signals without additional training.

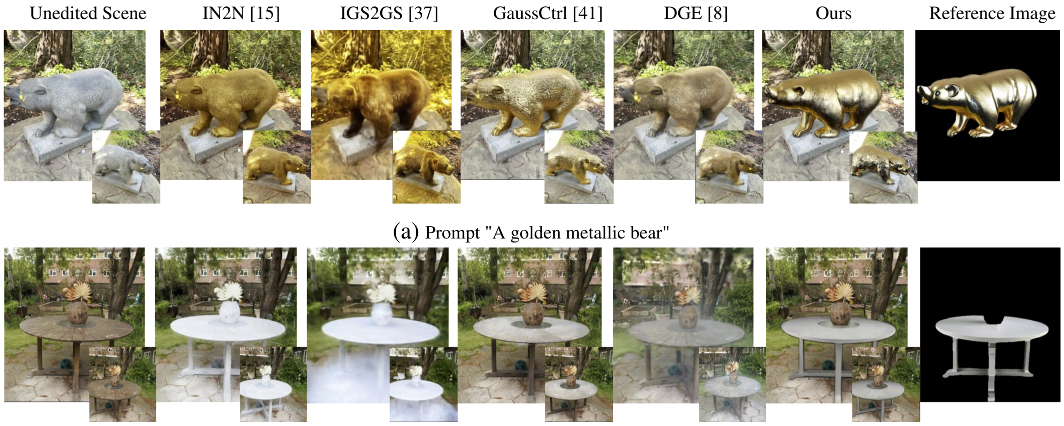
Given a target view $\mathbb{I}_i$ and reference set $\mathbb{R}_i = \{\mathbb{I}_\tau, \mathbb{I}_{i-1}, \mathbb{F}(\mathbb{I})_\tau\}$ (as in Sec. 3.1), we define the denoising prediction as:

$$\begin{aligned}\epsilon^t_\theta(z_\lambda, \mathbb{T}, \mathbb{R}) = \epsilon^t_{\hat\theta}(z_\lambda) \\ &+ w_\mathbb{T}\big(\epsilon^t_\theta(z_\lambda, \mathbb{T}, \mathbb{R}) - \epsilon^t_\theta(z_\lambda, \mathbb{R})\big) \\ &+ w_\mathbb{R}\big(\epsilon^t_\theta(z_\lambda, \mathbb{T}, \mathbb{R}) - \epsilon^t_{\hat\theta}(z_\lambda, \mathbb{T})\big),\end{aligned} \tag{2}$$

where $\mathbb{T}$ is the text prompt, θ and $\hat\theta$ refer to diffusion with and without fused cross-attention, and $w_\mathbb{T}, w_\mathbb{R}$ are scaling factors.

We perform partial denoising as:

$$\begin{aligned}z^{(t-1|\kappa)} = \sqrt{\alpha_{t-1|\kappa}} \frac{z^{t|\kappa} - \sqrt{1-\alpha_{t|\kappa}} \cdot \epsilon^{t|\kappa}_\theta(z, \mathbb{T}, \mathbb{R})}{\sqrt{\alpha_{t|\kappa}}} \\ + \sqrt{1-\alpha_{t-1|\kappa}} \epsilon^{t|\kappa}_\theta(z, \mathbb{T}, \mathbb{R}),\end{aligned} \tag{3}$$



(a) Prompt "A golden metallic bear"

(b) Prompt "A white table"

Figure 3: Qualitative comparison on 360-degree scenes (material and color edits): Our 3DOT method faithfully edits 3D objects' texture based on reference images.

where $t \in [0, \kappa]$, $\kappa_\tau < \kappa_{i \neq \tau}$, and α is the DDIM scheduler coefficient. The latent input z^t is initialized via:

$$z^{t+1} = \sqrt{\alpha_{t+1}} \frac{z^t - \sqrt{1 - \alpha_t} \cdot \epsilon^t}{\sqrt{\alpha_t}} + \sqrt{1 - \alpha_{t+1}} \epsilon^t. \tag{4}$$

In Eq. 2, the second term reflects differences between guidance with and without the unconditional prompt [16], improving adherence to textual instructions. The third term captures variations induced by reference conditioning, and amplifying it strengthens view consistency. This gradient-based mechanism guides generation toward coherent multi-view results, enhancing consistency without additional training overhead.

3.3 Prompt-Tuning-Based Gradient Guidance

Text prompts provide only coarse control during diffusion, often leading to identity loss and inconsistent texture fidelity. For example, the phrase "stone bear" may yield highly variable textures across different generations. These coarse text descriptions result in view-inconsistent generations and further cause 3D inconsistency.

Among fine-tuning methods [30, 45, 17, 13], textual inversion [13] learns a custom token to represent object-specific textures but requires multiple images to achieve reasonable quality (see supplementary for single-image results).

To address this, we introduce prompt-tuning-based gradient guidance, which reduces the need for multiple images while encoding texture differences more effectively. The key idea is to learn a new token that captures the texture discrepancy between the unedited 3D object and the reference image, and to use this token to guide denoising toward the desired style.

Given the reference image $\mathbb{I}_\tau$ and its corresponding unedited rendering $\hat{\mathbb{I}}_\tau$, we compute the texture difference in CLIP feature space:

$$\Delta_{\hat{\mathbb{I}}_\tau \to \mathbb{I}_\tau} = \text{CLIP}(\hat{\mathbb{I}}_\tau) - \text{CLIP}(\mathbb{I}_\tau). \tag{5}$$

We initialize the text token $\hat{\mathbb{T}}$ using a base prompt (e.g., from a VLM), and optimize it by aligning with the texture difference via:

$$L_{\text{clip}} = \text{cosine}(\Delta_{\hat{\mathbb{I}}_\tau \to \mathbb{I}_\tau}, \hat{\mathbb{T}}). \tag{6}$$

To reduce misalignment between image and text representations in CLIP space, we apply further prompt tuning in the diffusion feature space, following [26]:

$$L_{\text{diff}} = \boldsymbol{\epsilon}_\theta(z_\lambda, \mathbb{T}', \mathbb{R}) - \boldsymbol{\epsilon}'_\theta(z_\lambda, \mathbb{T}', \mathbb{R}). \tag{7}$$

Metrics	IN2N	IGS2GS	GaussCtrl	DGE	Ours
CLIP Score ↑	0.8917	0.8908	0.8638	0.8572	**0.9333**
Lpips(Alex) ↓	0.1708	0.1683	0.1692	0.1713	**0.1166**
Lpips(VGG) ↓	0.1676	0.1594	0.1591	0.1603	**0.1247**
Vision-GPT ↑	45.5	52	48	54	**76**
User study ↑	2.0375	2.4375	2.3750	2.0000	**4.5750**

Table 1: Quantitative results evaluated by CLIP score, VGG-based and Alex-based LPIPS scores, Vision-GPT and user studies given reference image with rendered edited objects. **Bold** text refers to the best performance and underlined text refers to the second best performance. Detailed results can be found in Supplementary Material Section 2.

The fine-tuned token $\mathbb{T}'$ acts as a style-aware prompt enriched by texture differences. While not meaningful in textual form, it encodes critical style information for guiding generation. During inference, we extract and amplify this information at t-step via the difference:

$$\epsilon_\theta^t(z_\lambda, \mathbb{T}', \mathbb{R}) - \epsilon_{\hat{\theta}}^t(z_\lambda, \mathbb{T}, \mathbb{R}),$$

and integrate it into the denoising process. The final prediction becomes:

$$\begin{aligned} \epsilon_\theta^t(z_\lambda, \mathbb{T}, \mathbb{R}, \mathbb{T}') = \epsilon_{\hat{\theta}}^t(z_\lambda) & \\ + w_{\mathbb{T}}\big(\epsilon_\theta^t(z_\lambda, \mathbb{T}, \mathbb{R}) - \epsilon_\theta^t(z_\lambda, \mathbb{R})\big) & \\ + w_{\mathbb{R}}\big(\epsilon_\theta^t(z_\lambda, \mathbb{T}, \mathbb{R}) - \epsilon_{\hat{\theta}}^t(z_\lambda, \mathbb{T})\big) & \\ + w_{\mathbb{T}'}\big(\epsilon_\theta^t(z_\lambda, \mathbb{T}', \mathbb{R}) - \epsilon_{\hat{\theta}}^t(z_\lambda, \mathbb{T}, \mathbb{R})\big). & \end{aligned} \tag{8}$$

This term strengthens style consistency across views while preserving fine texture details aligned with the reference.

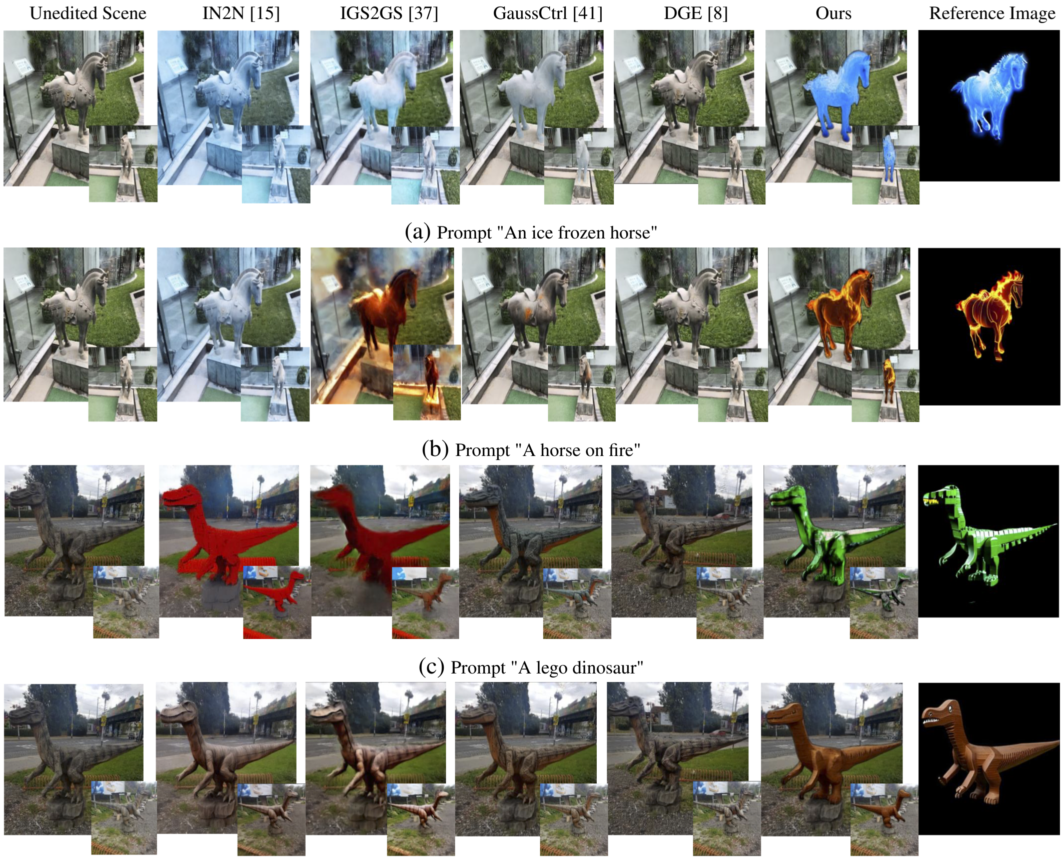
4 Experiments

We compare our method with state-of-the-art text-driven editing approaches, including GaussCtrl [41], DGE [8], IGS2GS [37], and IN2N [15]. Since these methods rely on text inputs, we use captioned descriptions as editing prompts to enable image-based 3D texture editing functionality. For quantitative evaluation, we employ AlexNet-based [22] and VGG-based [33] LPIPS scores [48], CLIP score [29], and Vision-GPT score [1], supplemented by user studies. Comparisons are conducted across multiple scenes from different datasets to ensure a comprehensive assessment following [41].

4.1 Evaluation

Quantitative For each edit, we compute AlexNet-based and VGG-based LPIPS scores, CLIP score, Vision-GPT score, and conduct user studies, as summarized in Table 1. Detailed per-scene scores are provided in the supplementary material. LPIPS and CLIP scores serve as perceptual evaluation metrics, measuring feature similarity between rendered edited objects and reference images. LPIPS ranges from 0 to 1, with lower values indicating better perceptual quality, while higher CLIP scores are preferred. Vision-GPT assesses the faithfulness of edited textures from the reasoning perspective, scoring from 0 to 100, where higher values indicate better alignment. For user studies, participants are informed of the edited object and required to rate the 3D result on a scale of 1 to 5, with higher scores reflecting better quality. Quantitative results show that our method achieves the highest performance across all metrics.

Qualitative We present qualitative results of 360-degree dataset in Figs.1, 3 and 4. Figs.3 and 3 includes reference images with texture color or material variations, while Fig.4 features those with complex textures and significant semantic changes. Fig.5 shows the results of "face-forward" case. Our method enables more precise 3D object editing without unintended texture leakage between objects. In the 360-degree color and material editing scenario (e.g., bear and table), IN2N [15] and IGS2GS [37] suffer from incorrect color saturation and inaccurate material representation. In the bear scenarios (Fig.1, 3a), their results are undersaturated, whereas in the table scenarios (Fig.3b, 1),



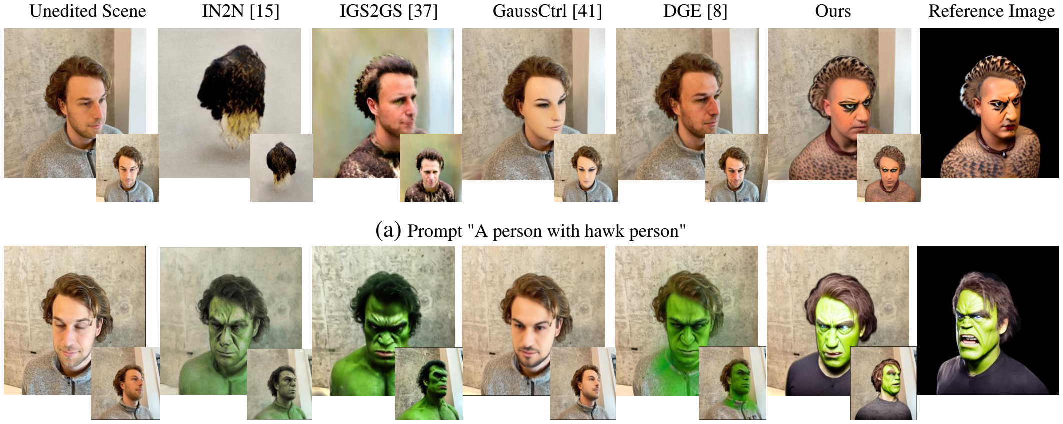
(a) Prompt "An ice frozen horse"

(b) Prompt "A horse on fire"

(c) Prompt "A lego dinosaur"

(d) Prompt "A wooden dinosaur"

Figure 4: Qualitative comparison on 360-degree scenes (complicated texture edits): Our 3DOT method successfully edits 3D objects' texture to complicated reference textures.

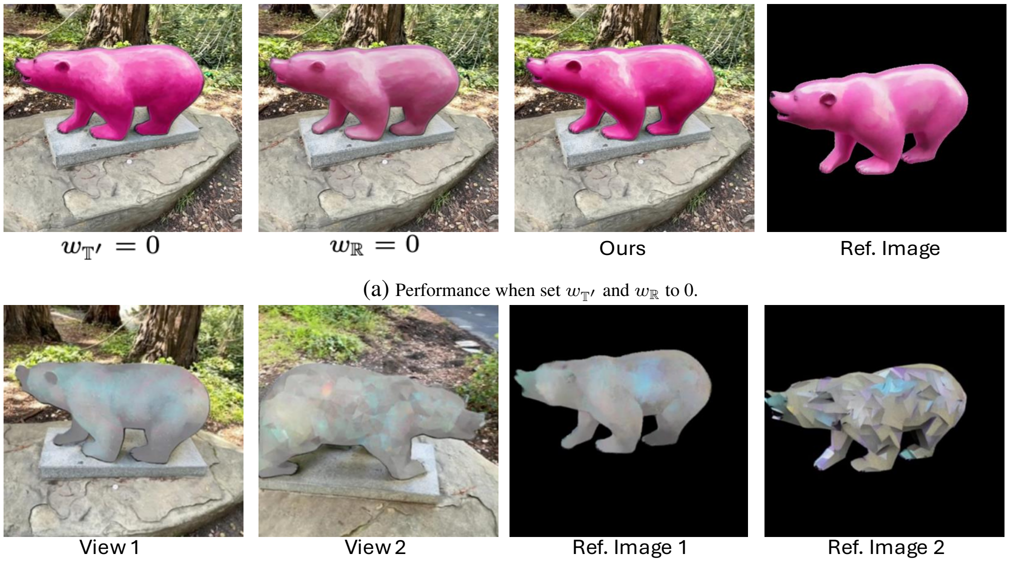


(a) Prompt "A person with hawk person"

(b) Prompt "A person with Hulk face"

Figure 5: Qualitative comparison on face-forwarding scenes: Our 3DOT method faithfully edits 3D objects' texture to reference textures and generates the most plausible texture edits for unseen views.

they are over-saturated. None of the baseline methods accurately reproduces the intended material attributes (i.e., plastic, moss in Fig. 1 and metallic in Fig. 3a). GaussCtrl [41] excessively preserves the original 3D object's appearance, resulting in minimal modifications due to its unedited reference set. Our method effectively edits textures while achieving realistic material appearances, such as



(a) Performance when set $w_{\mathbb{T}'}$ and $w_{\mathbb{R}}$ to 0.

(b) Using Ref.1 for prompt-tuning guidance and Ref.2 as reference image

Figure 6: Ablation studies on proposed two gradient guidances.

specular highlights on the bear and the lush, velvety moss on the table. In *moss-covered table* scenario (Fig.1), all 3D baseline methods only attempt to edit texture while ours can also modify geometry to better match the "moss material".

In the large semantic change editing scenario (Fig.4), baseline methods struggle with significant transformations. DGE [8] often fails, as its edits remain nearly unchanged. Its initial independent editing stage leads to inconsistent results and further causes the epipolar attention mechanism to break down in highly dissimilar views, resulting in minimal overall changes. Our method achieves precise 3D object editing with a texture style that closely matches the reference image, enabled by our proposed prompt-tuning, consistency guidance, and progressive process. Prompt-tuning preserves intricate texture details, while consistency guidance and progressive generation mitigate blurriness from view inconsistency.

In the face-forward case (Fig.5), our method preserves fine details, such as the black lower eyelid and feather-like cloth in the hawk scenario (Fig.5a). IN2N [15] and IGS2GS [37] generate erroneous results due to their independent diffusion process and full diffusion steps. The independent generation process leads to inconsistent images, while full diffusion steps cause excessive texture changes and identity loss. Finetuning NeRF with these inconsistent and identity-lost images can result in network collapse. For GaussCtrl [41] and DGE [8], particularly in the hawk case, large texture differences break their view-consistency mechanisms, resulting in outputs that retain the original object's appearance instead of the intended modifications.

4.2 Editing Speed

We compare the editing time of our method with two baselines: the Gaussian Splatting model (GaussCtrl [41]) and a representative NeRF-based model (IN2N [15]). GaussCtrl requires 15min 47s, while IN2N takes 20h 51min 20s. Our approach preserves the efficiency advantage of Gaussian Splatting, with an editing time of 23min 33s, introducing only a modest additional overhead from the incorporation of view-consistency and prompt-tuning-based guidance. In particular, the additional time required by our guidance components is approximately 8min in total. Since our full pipeline typically involves two iterations of image editing, the added cost per iteration is about 4min. We consider this overhead efficient given the improvements in texture fidelity and view consistency achieved by our method.

Ablation	LPIPS (Alex)↓	LPIPS (VGGT)↓	CLIP Score↑
W/O prompt-tuning (Sec.3.3)	0.0355	0.0679	0.9203
W/O consistency (Sec.3.2)	0.0558	0.0844	0.9340
W/O Prog. Gen.	0.1330	0.1290	0.9160
Ours	**0.0351**	**0.0620**	**0.9445**

Table 2: Ablation studies: Performance evaluation when removing (1) prompt-tuning guidance, (2) view-consistency guidance, and (3) progressive generation mechanism.

4.3 Ablation Studies

We evaluate the effectiveness of prompt-tuning-based gradient guidance and view-consistency gradient guidance from both qualitative and quantitative perspectives. We further illustrate the effectiveness of prompt-tuning-based guidance by using two distinct images: one serving as the reference image during image generation, and the other as the reference for prompt tuning.

Prompt-tuning based Gradient Guidance We first evaluate the effect of removing prompt-tuning-based guidance by setting the guidance scale $w_{\mathbb{T}'} = 0$, as shown in Table 2 and Figure 6a. Without this guidance, the rendered images exhibit blurry surface highlights, and the distribution direction does not align with the reference image.

Additionally, we demonstrate the effectiveness of prompt-tuning guidance by using a prompt token trained on reference image Ref-2 to edit the 3D object and setting Ref-1 as the target reference, as illustrated in Figure 6b. Ref-2 depicts a bear characterized by sharp metallic edges, whereas Ref-1 shows a bear with a rusted metallic texture. In Figure 6b, View-1 aligns closely with Ref-1, producing a rendering that closely matches the rusted metal appearance, which demonstrates the effectiveness of the utilized fused cross-attention. Conversely, View-2, which represents an unseen viewpoint without explicit texture guidance, utilizes the learned token to guide the rendering towards the "colorful metallic bear" appearance consistent with Ref-2. This demonstrates the effectiveness of the proposed prompt-tuning-based guidance when editing parts of the 3D object not visible in the reference image.

View-consistency Gradient Guidance We evaluate the effectiveness of view-consistency guidance by setting $w_{\mathbb{R}} = 0$, as reported in Table 2 and illustrated in Figure 6a. When this guidance is disabled, the performance significantly degrades, resulting in edited images exhibiting notable undersaturation. This undersaturation primarily arises due to inconsistencies within overlapping regions in the intermediate outputs. These findings underscore the crucial role of view-consistency gradient guidance in maintaining editing quality and color fidelity.

Progressive Generation To evaluate the effectiveness of the progressive generation, we disable progressive view propagation and perform editing using only the initial reference image. The degraded performance (as shown in Table 2) highlights the importance of propagating texture across neighboring views to enforce view consistency and mitigate artifacts from single-view editing.

4.4 Discussion

Differences from Multi-View Diffusion While existing multi-view diffusion methods are designed to generate multi-view consistent 3D objects, the task setting, constraints, and diffusion components in our work are fundamentally different. Specifically, **3DOT** takes as input a single 2D reference image and a fixed 3D Gaussian Splatting (3DGS) representation, and transfers high-fidelity texture onto this existing geometry. In contrast, multi-view diffusion methods are typically designed to synthesize novel views or reconstruct 3D scenes from a few texture-consistent input images, without anchoring to an explicit 3D representation. These methods rely on implicit geometry learned from priors, which makes them unsuitable for editing tasks that require consistency with a given 3D structure. Moreover, multi-view diffusion approaches assume that texture appearance is consistent across views, an assumption that does not hold in texture transfer scenarios where the reference image and the 3D representation exhibit different textures. Directly applying them in this setting leads to collapsed reconstructions, highlighting the need for a specialized framework such as ours.

Different Geometric Reference Image Geometry mismatches between the 2D reference image and the 3D object can occur when the reference is generated via a depth-conditioned diffusion model with a small conditioning scale factor. We consider two common scenarios:

- **Slightly Different Geometry:** Minor deviations (e.g., slight pose or scale differences, such as a standing bear with legs in a different position) can be effectively handled by **3DOT**. (1) During progressive generation, cross-attention extracts texture features while being constrained by the depth map and partially reversed latent features, preserving appearance cues from the original image. (2) During 3D fine-tuning, overlapping regions from adjacent edited views iteratively correct geometric discrepancies and reinforce consistent texture transfer.
- **Significantly Different Geometry:** When the reference depicts a substantially different shape (e.g., a running bear instead of a standing one), transfer quality may degrade. Such cases usually result from failures in depth-conditioned option generation or from unsuitable user-provided references. In practice, these poor references can be easily identified during the interactive selection stage, and regenerated using our partial denoising strategy with a larger depth control factor.

In summary, **3DOT** is robust to minor geometric mismatches, and mitigates larger discrepancies through reference regeneration and iterative correction during 3D fine-tuning.

Limitation Dark border around edited regions occurs in some cases. We attribute this artifact to language-based object segmentation methods (e.g., LangSAM) to generate masks for isolating objects in the reference views. These segmentation methods often include a narrow band of background pixels near object boundaries due to imperfect boundary localization. As a result, during the diffusion-based editing stage, this narrow band is misinterpreted as valid texture content, leading to the appearance of dark borders in the final renderings even after applying the soften mask. This may be addressed by utilizing a more advanced language-based segmentation method [49, 43, 35], depth as additional information [44, 50, 10] with a semantic 3D representation [18, 12]. The quality of unedited 3D Gaussians also impacts editing performance. Undertrained Gaussian spheres (e.g., floating Gaussians in empty space) degrade rendered images, disrupting the mask generation process. Incorrect segmentation can result in edits with significantly altered geometry, ultimately causing 3D Gaussian collapse.

5 Conclusion

We introduced **3DOT**, a framework for image-based 3DGS texture transfer from a single reference image, an underexplored capability in the 3D editing domain. To enable high-quality and view-consistent texture transfer, we proposed three key components: (1) progressive generation, (2) view-consistency gradient guidance, and (3) prompt-tuning-based gradient guidance. These components effectively address challenges of view-consistency and texture characteristic preservation during transfering process. We evaluated **3DOT** across various scenes involving color, material, and large semantic changes. **3DOT** consistently outperforms existing baselines, both visually and quantitatively.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Shivangi Aneja, Justus Thies, Angela Dai, and Matthias Nießner. Clipface: Text-guided editing of textured 3d morphable models. In *ACM SIGGRAPH 2023 Conference Proceedings*, pages 1–11, 2023.

[3] 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.

[4] 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] Xiao Cao, Beibei Lin, Bo Wang, Zhiyong Huang, and Robby T Tan. Ssnerf: Sparse view semi-supervised neural radiance fields with augmentation. *arXiv preprint arXiv:2408.09144*, 2024.

[6] Xiao Cao, Yuyang Zhao, Robby T Tan, and Zhiyong Huang. Bridging 3d editing and geometry-consistent paired dataset creation for 2d nighttime-to-daytime translation. In *[CVPR 2025 Workshop] SyntaGen: 2nd Workshop on Harnessing Generative Models for Synthetic Visual Datasets*.

[7] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[8] Minghao Chen, Iro Laina, and Andrea Vedaldi. Dge: Direct gaussian 3d editing by consistent multi-view editing. *arXiv preprint arXiv:2404.18929*, 2024.

[9] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21476–21485, 2024.

[10] Zitan Chen, Zhuang Qi, Xiao Cao, Xiangxian Li, Xiangxu Meng, and Lei Meng. Class-level structural relation modeling and smoothing for visual representation learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 2964–2972, 2023.

[11] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023.

[12] Shaohui Dai, Yansong Qu, Zheyan Li, Xinyang Li, Shengchuan Zhang, and Liujuan Cao. Training-free hierarchical scene understanding for gaussian splatting with superpoint graphs. *arXiv preprint arXiv:2504.13153*, 2025.

[13] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *arXiv preprint arXiv:2208.01618*, 2022.

[14] Jing Gu, Yilin Wang, Nanxuan Zhao, Wei Xiong, Qing Liu, Zhifei Zhang, He Zhang, Jianming Zhang, HyunJoon Jung, and Xin Eric Wang. Swapanything: Enabling arbitrary object swapping in personalized visual editing. *arXiv preprint arXiv:2404.05717*, 2024.

[15] 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.

[16] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

[17] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

[18] Chi Huang, Xinyang Li, Shengchuan Zhang, Liujuan Cao, and Rongrong Ji. Nerf-dets: Enhancing multi-view 3d object detection with sampling-adaptive network of continuous nerf-based representation. *arXiv e-prints*, pages arXiv–2404, 2024.

[19] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision*, pages 1501–1510, 2017.

[20] Sahil Jain, Avik Kuthiala, Prabhdeep Singh Sethi, and Prakanshul Saxena. Stylesplat: 3d object style transfer with gaussian splatting. *arXiv preprint arXiv:2407.09473*, 2024.

[21] 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.

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[23] Xinyang Li, Zhangyu Lai, Linning Xu, Yansong Qu, Liujuan Cao, Shengchuan Zhang, Bo Dai, and Rongrong Ji. Director3d: Real-world camera trajectory and 3d scene generation from text. *Advances in Neural Information Processing Systems*, 37:75125–75151, 2024.

[24] Beibei Lin, Tingting Chen, and Tan Robby T. Geocomplete: Geometry-aware diffusion for reference-driven image completion. *The Thirty-Ninth Annual Conference on Neural Information Processing Systems*, 2025.

[25] Chong Mou, Xintao Wang, Jiechong Song, Ying Shan, and Jian Zhang. Diffeditor: Boosting accuracy and flexibility on diffusion-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8488–8497, 2024.

[26] Thao Nguyen, Yuheng Li, Utkarsh Ojha, and Yong Jae Lee. Visual instruction inversion: Image editing via image prompting. *Advances in Neural Information Processing Systems*, 36, 2024.

[27] Yansong Qu, Dian Chen, Xinyang Li, Xiaofan Li, Shengchuan Zhang, Liujuan Cao, and Rongrong Ji. Drag your gaussian: Effective drag-based editing with score distillation for 3d gaussian splatting. *arXiv preprint arXiv:2501.18672*, 2025.

[28] Yansong Qu, Shaohui Dai, Xinyang Li, Jianghang Lin, Liujuan Cao, Shengchuan Zhang, and Rongrong Ji. Goi: Find 3d gaussians of interest with an optimizable open-vocabulary semantic-space hyperplane. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 5328–5337, 2024.

[29] 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.

[30] Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. Dreambooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22500–22510, 2023.

[31] Taha Samavati and Mohsen Soryani. Deep learning-based 3d reconstruction: a survey. *Artificial Intelligence Review*, 56(9):9175–9219, 2023.

[32] Yujun Shi, Chuhui Xue, Jun Hao Liew, Jiachun Pan, Hanshu Yan, Wenqing Zhang, Vincent YF Tan, and Song Bai. Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8839–8849, 2024.

[33] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[34] Weilin Sun, Manyi Li, Peng Li, Xiao Cao, Xiangxu Meng, and Lei Meng. Sequential selection and calibration of video frames for 3d outdoor scene reconstruction. *CAAI Transactions on Intelligence Technology*, 9(6):1500–1514, 2024.

[35] Lei Tan, Pingyang Dai, Jie Chen, Liujuan Cao, Yongjian Wu, and Rongrong Ji. Partformer: Awakening latent diverse representation from vision transformer for object re-identification. *arXiv preprint arXiv:2408.16684*, 2024.

[36] Narek Tumanyan, Michal Geyer, Shai Bagon, and Tali Dekel. Plug-and-play diffusion features for text-driven image-to-image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1921–1930, 2023.

[37] Cyrus Vachha and Ayaan Haque. Instruct-gs2gs: Editing 3d gaussian splats with instructions, 2024.

[38] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *IEEE Transactions on Visualization and Computer Graphics*, 2023.

[39] Yuze Wang, Junyi Wang, Ruicheng Gao, Yansong Qu, Wantong Duan, Shuo Yang, and Yue Qi. Look at the sky: Sky-aware efficient 3d gaussian splatting in the wild. *IEEE Transactions on Visualization and Computer Graphics*, 2025.

[40] Yuze Wang, Junyi Wang, Yansong Qu, and Yue Qi. Rip-nerf: Learning rotation-invariant point-based neural radiance field for fine-grained editing and compositing. In *Proceedings of the 2023 ACM international conference on multimedia retrieval*, pages 125–134, 2023.

[41] 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.

[42] Yihang Wu, Xiao Cao, Kaixin Li, Zitan Chen, Haonan Wang, Lei Meng, and Zhiyong Huang. Towards better text-to-image generation alignment via attention modulation. In *International Conference on Neural Information Processing*, pages 332–347. Springer, 2024.

[43] Jiaer Xia, Lei Tan, Pingyang Dai, Mingbo Zhao, Yongjian Wu, and Liujuan Cao. Attention disturbance and dual-path constraint network for occluded person re-identification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 6198–6206, 2024.

[44] Weilong Yan, Ming Li, Haipeng Li, Shuwei Shao, and Robby T Tan. Synthetic-to-real self-supervised robust depth estimation via learning with motion and structure priors. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21880–21890, 2025.

[45] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721*, 2023.

[46] Junyi Zhang, Charles Herrmann, Junhwa Hur, Luisa Polania Cabrera, Varun Jampani, Deqing Sun, and Ming-Hsuan Yang. A tale of two features: Stable diffusion complements dino for zero-shot semantic correspondence. *Advances in Neural Information Processing Systems*, 36, 2024.

[47] 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.

[48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.

[49] Xin Zhang and Robby T Tan. Mamba as a bridge: Where vision foundation models meet vision language models for domain-generalized semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 14527–14537, 2025.

[50] Xin Zhang, Jinheng Xie, Yuan Yuan, Michael Bi Mi, and Robby T Tan. Heap: unsupervised object discovery and localization with contrastive grouping. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 7323–7331, 2024.

[51] Chenyang Zhu, Kai Li, Yue Ma, Longxiang Tang, Chengyu Fang, Chubin Chen, Qifeng Chen, and Xiu Li. Instantswap: Fast customized concept swapping across sharp shape differences. *arXiv preprint arXiv:2412.01197*, 2024.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer:[Yes]

   Justification: Please refer to abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to Discussion section

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theory assumption.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to implementation detail and code upon acceptance.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: We will release our code upon acceptance.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to experimental section.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our experiment section does not include error bar related experiment.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We include in the supplementary material.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We use public datasets.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: Our work is unrelated to social matters.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: We does not pose such risk.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All datasets are public datasets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [No]

Justification: No new asset.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [No]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: Not applicable.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification: We only use it to revise our paper.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.