DReSG: Diffusion Residuals for Stylized Gaussian Splatting

Zhongliang Liu[1], Wenjie Liu[2], and Yang Li[2,3]†

[1]School of Software Engineering, East China Normal University, Shanghai, China
[2]School of Computer Science and Technology, East China Normal University, Shanghai, China
[3]School of Intelligent Interaction, East China Normal University, Shanghai, China

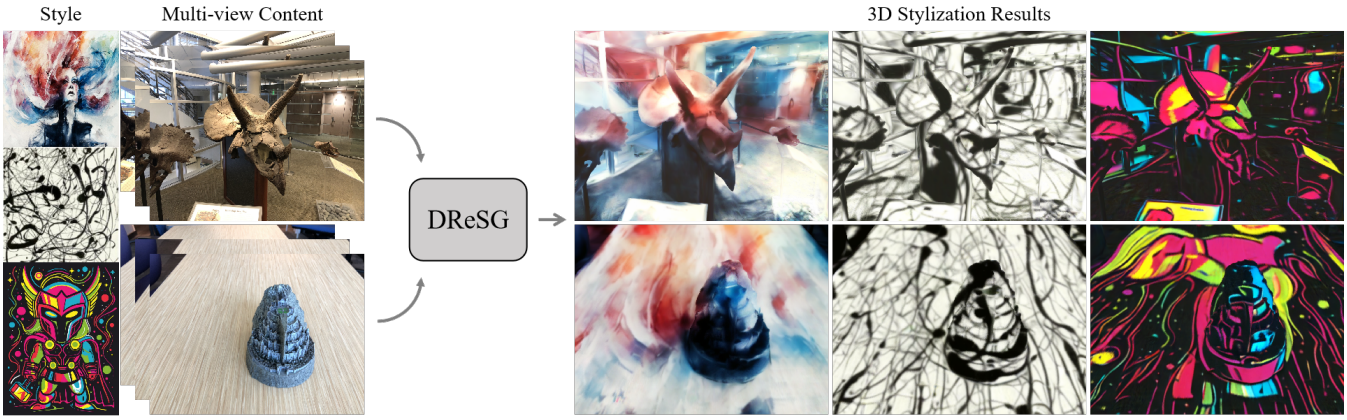

Figure 1: *DReSG fits diffusion-proposed stylization residuals into a reconstructed Gaussian scene, producing a renderable stylized Gaussian scene whose reference-specific appearance persists across novel views.*

Abstract

*Reference-guided stylization of scenes represented by 3D Gaussian Splatting (3DGS) is important for efficient and controllable 3D content creation. Existing VGG-feature-based 3D stylization methods provide stable rendered-view optimization, but often under-represent expressive reference style cues; diffusion models offer stronger image priors, yet direct per-view or score-based diffusion guidance can lead to view drift, local artifacts, and hard-to-control appearance updates. We present **DReSG**, a 3D-grounded residual-feedback framework for stylized Gaussian splatting. DReSG represents attention-guided diffusion proposals as residual targets relative to the current render, and progressively absorbs these residuals into a shared Gaussian scene through multi-view Gaussian feedback. To make this feedback stable and controllable, DReSG modulates residual strength during target construction and combines coverage-aware view selection with conflict-filtered color updates during multi-view fitting. Extensive experiments demonstrate that DReSG achieves competitive reference-guided stylization while better preserving scene structure and cross-view stability. Our project page is available at* https://vpx-ecnu.github.io/DReSG-website/.

CCS Concepts

• ***Computing methodologies*** → ***Computer graphics; Rendering;*** *Image processing;*

Keywords: 3D Gaussian Splatting; reference-guided stylization; diffusion models; multi-view consistency

1. Introduction

3D style transfer aims to transfer the artistic appearance of a reference image onto a 3D scene while preserving the scene's original structure and semantic content [ZKB*22, ZYC*25]. With the growing demand for editable 3D content in virtual reality, film

† Corresponding author

production, games, and digital asset creation, 3D stylization has become an important tool for efficient and controllable scene authoring. Compared with 2D image stylization, a stylized Gaussian scene must transfer the reference appearance, preserve the reconstructed structure, and maintain the 3D anchoring of local appearance changes under camera motion.

Most existing 3D stylization methods follow a rendered-view optimization paradigm: they render images from a 3D representation and optimize the scene with 2D style supervision. This paradigm supports stable optimization and multi-view constraints and has been widely used for both NeRF and Gaussian stylization [NPLX22, ZKB*22, LZC*23, LZX*24, ZYC*25]. Many methods rely on VGG-feature-based objectives, including Gram-based style losses, perceptual feature reconstruction, adaptive normalization, or feature transforms [GEB16, JAFF16, HB17, LFY*17]. These objectives effectively transfer global colors and local texture responses and provide gradients that can be readily propagated through the differentiable renderer. However, VGG feature objectives often express the reference style through local feature responses, normalization parameters, or aggregate correlations. As a result, they can under-represent spatially organized reference cues such as continuous strokes, coherent contours, directional structures, and region-wise color treatment, leading to weak stylization effects or fragmented textures instead of a persistent style update stored in the 3D scene.

More expressive image priors provide new opportunities for reference-guided 3D stylization. CLIP-based guidance introduces high-level semantic or multimodal image similarity [RKH*21, HWB*26], but it is usually used as a global matching objective rather than a local render-space target that can be directly fitted by 3D rendering. Diffusion models are generative image priors trained to synthesize images through iterative denoising [HJA20, RBL*22]. They can produce concrete stylization proposals with richer semantic, structural, and local appearance changes than VGG feature losses.

Despite this stronger prior, diffusion signals are not immediately stable targets for 3DGS stylization. Per-view diffusion stylization can introduce inconsistent colors, textures, or local structures across views [HTE*23, WFZ*24, CLV24, YWW*26], while directly optimizing 3D parameters with diffusion scores or timestep-dependent gradients [PJBM22, WDL*23, WLW*23] lacks an explicit render-space target and can produce saturated colors, local artifacts, or poorly controlled appearance updates. Our insight is to use diffusion as a source of observable reference-aware stylization proposals, and to convert these proposals into residual targets relative to the current render. The key novelty is this render-relative residual-feedback formulation: diffusion proposes reference-aware appearance changes, while the shared Gaussian scene repeatedly absorbs the residuals that can be explained across views.

We propose **DReSG**, a 3D-grounded residual-feedback framework for stylized Gaussian splatting. DReSG renders the current Gaussian scene, generates attention-guided diffusion proposals, computes render-relative residual targets, and fits these targets back to the shared Gaussian scene through differentiable multi-view Gaussian rendering. To control the feedback strength, DReSG constructs targets in RGB-logit space and applies SNR-balanced residual-strength modulation. To improve multi-view feedback with a compact active-view set, DReSG uses coverage-aware active views and conflict-filtered color updates. This process converts expressive diffusion proposals into appearance updates that are progressively absorbed by a shared renderable 3D scene.

In summary, our contributions are:

- We formulate reference-guided 3DGS stylization as render-relative diffusion residual feedback, converting attention-guided diffusion proposals into render-relative targets that are progressively fitted into a shared Gaussian scene.
- We design a controllable residual-target and multi-view feedback procedure with RGB-logit target scaling, SNR-balanced residual modulation, and coverage-aware active views.
- Extensive comparisons and ablations against representative VGG-feature-based, CLIP-guided, and diffusion-based 3D stylization methods demonstrate a better balance among reference-guided stylization, content preservation, and cross-view stability.

2. Related Work

2.1. 2D Reference-Guided Stylization

2D reference-guided stylization studies how to transfer the appearance of a style reference to an input image [GEB16, JAFF16, HB17, LFY*17]. Classical neural style transfer represents style with Gram-matrix correlations in VGG features and separates content and style in a deep feature space. Subsequent methods improve efficiency and generality through perceptual objectives, adaptive normalization, or feature transforms. More recently, diffusion-based image stylization and reference-guided generation use pre-trained generative priors to synthesize content-preserving images that follow a style or reference image [RBL*22, ZHT*23, YZL*23, WWB*24]. These methods commonly rely on latent diffusion, attention control, or reference-image conditioning to produce more concrete color, stroke, and local appearance changes than VGG-feature objectives alone.

2.2. 3D Scene Stylization

3D scene stylization transfers the appearance of a reference image to a renderable 3D representation, requiring the stylized appearance to remain consistent under camera motion. Existing approaches mainly build on two scene representations. Neural Radiance Fields (NeRFs) represent scenes as continuous neural radiance functions and render images through volumetric integration [MST*21]. NeRF-based stylization methods lift 2D style objectives to 3D by optimizing implicit radiance fields with rendered-view losses [MST*21, NPLX22, ZKB*22, LZC*23, FMH24]. Later work improves view consistency by constructing stylized observations or correspondences before or during 3D optimization [HTS*21, HHY*22, IKN24, ZCH*25]. These methods improve stylized novel views, but their optimization and rendering costs are less aligned with the explicit splatting representation used in 3DGS.

Recent 3DGS stylization methods can be grouped by their supervision form. VGG-feature losses and patch-level matching provide stable Gaussian appearance optimization [LZX*24, GWRM25, LLY*25], while texture transfer and geometry-aware constraints

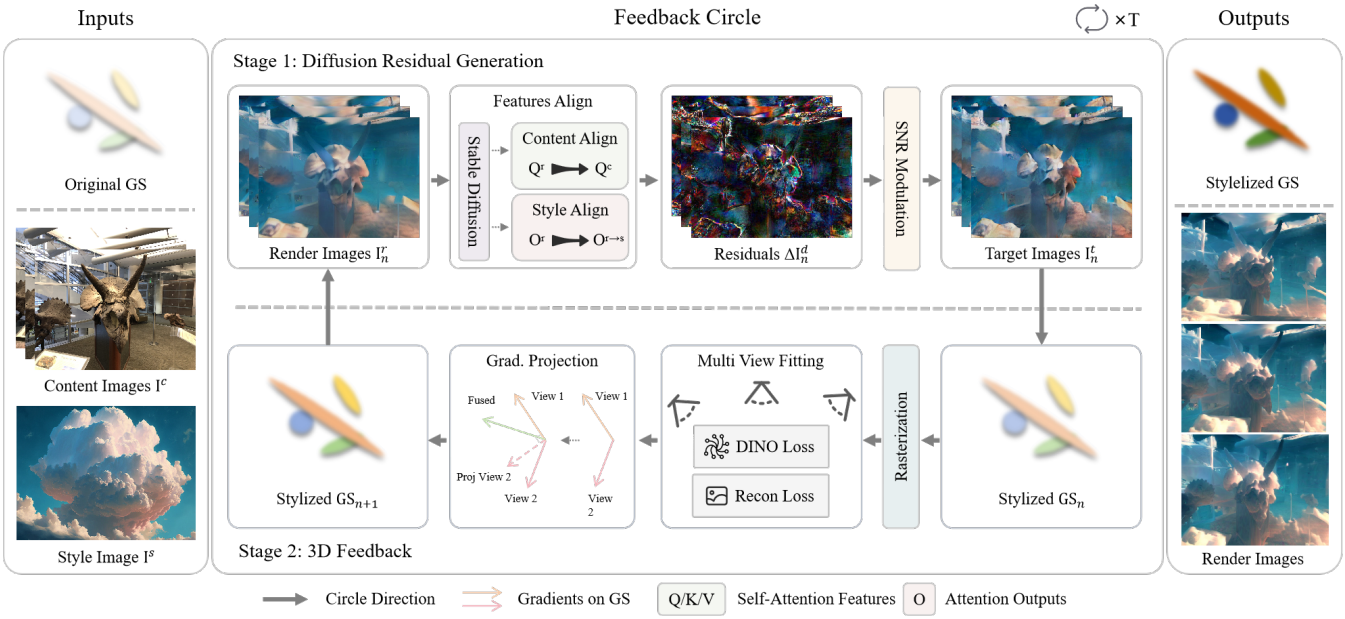

Figure 2: *Overview of DReSG. Given a scene represented by 3DGS, its content renderings, and a reference style image, DReSG generates attention-guided diffusion proposals, constructs render-relative residual targets, and fits them back to the scene through multi-view Gaussian rendering.*

improve controllability and structure preservation [LLS*26]. CLIP or prompt guidance introduces more flexible semantic control [HWB*26], and diffusion-related guidance or stylized distillation strengthens reference-style proposals [YWW*26]. However, these supervision forms still leave a gap between strong style signals and persistent scene-level appearance: feature losses are indirect, prompt guidance is global, and diffusion-generated stylized views can drift if their view-wise details are not absorbed by a shared 3D representation. DReSG addresses this gap with diffusion-derived, render-relative residual targets that are progressively fitted into the shared Gaussian scene.

2.3. Diffusion-Guided 3D Generation and Editing

Diffusion priors have been widely used for 3D generation and editing [PJBM22, SWY*23, HTE*23, CLV24]. Existing methods broadly follow two routes. One uses a pretrained 2D diffusion model as a prior for 3D optimization, optimizing NeRF, mesh, or Gaussian representations through score-based objectives or multi-view diffusion priors [PJBM22, WDL*23, LGT*22, WLW*23, SWY*23]. The other edits existing scenes by applying instruction- or prompt-guided image edits to rendered views and propagating those edits back to a 3D scene with multi-view constraints, scene-level optimization, or consistency regularization [HTE*23, WFZ*24, WBL*24, WYW*24, CLV24]. These approaches demonstrate the strength of diffusion priors as image-level guidance for 3D content generation and editing. Compared with general diffusion-guided generation or editing, reference-guided 3DGS stylization requires injecting reference appearance into a reconstructed scene while preserving structure and cross-view persistence. Score-based optimization can produce timestep-dependent gradients without an inspectable render-space target, whereas per-view edits may leave style changes in independent images. DReSG instead uses diffusion to form observable proposal-render residuals that are fitted by the shared Gaussian scene.

3. Preliminaries

This section reviews the two sets of notation used by DReSG: Gaussian rendering and diffusion scheduler quantities. 3D Gaussian Splatting represents a scene as a set of differentiable Gaussian primitives and renders images by projecting visible Gaussians to the image plane followed by front-to-back alpha compositing [KKLD23]. Let $\mathcal{G} = \{(\mu_i, \mathbf{s}_i, \mathbf{q}_i, \alpha_i, \mathbf{c}_i)\}_{i=1}^{N_\mathcal{G}}$ denote a Gaussian scene, where μ_i, $\mathbf{s}_i$, $\mathbf{q}_i$, α_i, and $\mathbf{c}_i$ are the mean, scale, rotation, opacity, and color attributes of the i-th Gaussian. Given a camera v, the rendered color at pixel p is

$$\mathcal{R}(\mathcal{G}_n, v)[p] = \sum_{i \in \mathcal{N}_p} T_{i,p} \alpha_{i,p} \mathbf{c}_{i,p}, \quad T_{i,p} = \prod_{j<i} (1 - \alpha_{j,p}), \tag{1}$$

where $\mathcal{N}_p$ is the depth-ordered set of Gaussians contributing to pixel p, $\alpha_{i,p}$ is the projected opacity, $T_{i,p}$ is the accumulated transmittance before Gaussian i, and $\mathbf{c}_{i,p}$ is the projected color. In the DReSG feedback loop, $\mathcal{G}_n$ denotes the current Gaussian scene at stage n, and the rendered image from view v is defined as

$$\mathbf{I}_n^{r,v} = \mathcal{R}(\mathcal{G}_n, v). \tag{2}$$

Superscripts r, c, s, and t denote the current render, content-guidance image, style image, and residual fitting target, respectively, while d denotes proposal-render residuals. Throughout the paper, n indexes the outer DReSG feedback stage and k indexes a selected diffusion scheduler state.

Algorithm 1 DReSG residual-feedback optimization.

Input: Base scene $\mathcal{G}$, active views $\mathcal{V}$, content views $\{\mathbf{I}^{c,v}\}$, style image $\mathbf{I}^s$, DDIM states $\{\tau_q\}_{q=1}^N$, feedback interval S, and fit length M

$\mathbf{z}^{r,v} \leftarrow \mathcal{E}_{\text{VAE}}(\text{Render}(\mathcal{G}, v)), \quad \forall v \in \mathcal{V}$
for $n \leftarrow 1$ **to** N/S **do**
  for $j \leftarrow 1$ **to** S **do**
    $k \leftarrow \tau_{S(n-1)+j}$
    $\mathbf{z}^{r,v} \leftarrow \text{Adam}(\mathbf{z}^{r,v}; \mathcal{E}^{\text{attn},v}(k)), \quad \forall v \in \mathcal{V}$
  end for
  $\mathbf{I}^{r,v} \leftarrow \text{Render}(\mathcal{G}, v), \quad \Delta\mathbf{I}^{d,v} \leftarrow \text{Decode}(\mathbf{z}^{r,v}) - \mathbf{I}^{r,v}$
  $\mathbf{I}^{t,v} \leftarrow \text{Target}_k(\mathbf{I}^{r,v}, \Delta\mathbf{I}^{d,v})$ using Eq. 12, $\forall v \in \mathcal{V}$
  for $m \leftarrow 1$ **to** M **do**
    Compute $\mathcal{L}_{\text{fit}}$ and view-wise color gradients.
    Project and fuse conflicting color gradients.
    $\mathcal{G} \leftarrow \text{Adam}(\mathcal{G}; \mathcal{L}_{\text{fit}})$
  end for
  $\mathbf{z}^{r,v} \leftarrow \mathcal{E}_{\text{VAE}}(\text{Render}(\mathcal{G}, v)), \quad \forall v \in \mathcal{V}$
end for

Output: Stylized Gaussian scene $\mathcal{G}$

Diffusion models define a sequence of scheduler states that trade off signal and noise during image generation [HJA20, RBL*22]. Under the standard forward-process notation, the noisy latent corresponding to a clean latent $\mathbf{z}_0$ at scheduler timestep k is written as

$$\mathbf{z}_k = \sqrt{\bar{\alpha}_k}\mathbf{z}_0 + \sqrt{1-\bar{\alpha}_k}\epsilon, \quad \bar{\alpha}_k = \prod_{i=1}^{k} \alpha_i. \tag{3}$$

The corresponding signal-to-noise ratio is

$$\text{SNR}(k) = \frac{\bar{\alpha}_k}{1-\bar{\alpha}_k}. \tag{4}$$

In DReSG, $\bar{\alpha}_k$ and $\text{SNR}(k)$ are used only to describe the scheduler state and to define the residual-strength modulation in Section 4.2; the Gaussian scene parameters are not treated as diffusion variables.

4. Method

DReSG formulates stylized Gaussian splatting as a closed-loop residual feedback problem grounded in rendered views. Given a scene represented by 3DGS and a reference style image, DReSG repeatedly renders active views from the current Gaussian scene, obtains attention-guided latent diffusion proposals from a frozen diffusion model, converts proposal-render differences into render-relative residual targets, and fits those targets into a shared Gaussian scene. Fig. 2 summarizes this pipeline. The following subsections describe the three main steps of DReSG: attention-guided proposal residual generation, SNR-modulated residual target construction, and multi-view Gaussian feedback. Algorithm 1 makes the alternating latent and Gaussian optimization schedule explicit.

4.1. Attention-Guided Proposal Residuals

An expressive style signal should be extracted without decoupling it from the current 3D scene. VGG-style rendered-view losses are stable for optimization, yet their supervision is often implicit in feature-space correlations or patch matches. Diffusion models can propose richer reference-aware appearance changes, but a full per-view stylized proposal may contain details that cannot be explained consistently by a shared Gaussian scene. DReSG therefore uses diffusion only to propose an appearance update for the current render; the resulting proposal-render residual then guides subsequent 3D feedback.

For each active view v, a frozen latent diffusion model with attention-guided latent optimization receives the current rendering $\mathbf{I}_n^{r,v}$, the original content view $\mathbf{I}^{c,v}$, and the style image $\mathbf{I}^s$. The index n identifies the current 3D scene state, whereas k identifies the diffusion feature-extraction state. The current render is the image being stylized, the content view provides layout guidance, and the style image provides reference appearance. At each feedback stage, the render latent is initialized from the VAE projection of the current render and optimized with attention guidance at the scheduler state used by that stage.

DReSG obtains the residual by optimizing the render latent along selected scheduler indices. Let $\mathbf{z}_k^{r,v}$ denote the current render latent state during this optimization at scheduler state k, $\mathbf{z}^{c,v}$ the VAE latent of $\mathbf{I}^{c,v}$, and $\mathbf{z}^s$ the VAE latent of $\mathbf{I}^s$. We use the frozen U-Net as a timestep-conditioned feature extractor: k selects the U-Net features used for attention guidance and also provides the scheduler state for residual-strength modulation. Let

$$\begin{aligned}(\mathbf{Q}_k^{r,v}, \mathbf{K}_k^{r,v}, \mathbf{V}_k^{r,v}) &= \Phi^{\text{attn}}(\mathbf{z}_k^{r,v}; k),\\ (\mathbf{Q}_k^{c,v}, \mathbf{K}_k^{c,v}, \mathbf{V}_k^{c,v}) &= \Phi^{\text{attn}}(\mathbf{z}^{c,v}; k),\\ (\mathbf{Q}_k^{s}, \mathbf{K}_k^{s}, \mathbf{V}_k^{s}) &= \Phi^{\text{attn}}(\mathbf{z}^{s}; k).\end{aligned} \tag{5}$$

The operator Φ^{attn} collects self-attention features from selected U-Net layers at timestep k into compact Q/K/V feature tensors. We use the content query for layout preservation and the style keys/values for reference appearance guidance; content and style features serve as fixed references during render-latent optimization. Content guidance matches the current render query to the content query,

$$\mathcal{E}^{c,v}(k) = \left\|\mathbf{Q}_k^{r,v} - \mathbf{Q}_k^{c,v}\right\|_1. \tag{6}$$

For style guidance, we compare the current self-attention output with an output formed by using the render query to attend to the style keys and values:

$$\begin{aligned}\mathbf{O}_k^{r,v} &= \text{Attn}(\mathbf{Q}_k^{r,v}, \mathbf{K}_k^{r,v}, \mathbf{V}_k^{r,v}),\\ \mathbf{O}_k^{r\to s,v} &= \text{Attn}(\mathbf{Q}_k^{r,v}, \mathbf{K}_k^{s}, \mathbf{V}_k^{s}).\end{aligned} \tag{7}$$

The operator $\text{Attn}(\cdot)$ denotes the standard attention-output operator of the frozen U-Net self-attention blocks. The style loss is

$$\mathcal{E}^{s,v}(k) = \left\|\mathbf{O}_k^{r,v} - \mathbf{O}_k^{r\to s,v}\right\|_1. \tag{8}$$

This formulation is inspired by the self-attention feature-transfer objective of Zhou et al. [ZGCH25]. DReSG then uses the optimized render latent to construct proposal-render residuals for Gaussian feedback. The attention guidance energy is

$$\mathcal{E}^{\text{attn},v}(k) = \mathcal{E}^{s,v}(k) + w_c\mathcal{E}^{c,v}(k). \tag{9}$$

The scalar w_c controls content preservation. At each feedback stage, DReSG updates the render latent for multiple Adam steps to reduce $\mathcal{E}^{\text{attn},v}$. We denote the resulting optimized proposal latent

as $\mathbf{z}_k^{r,v,*}$. The proposal-render residual is obtained by decoding the optimized latent and subtracting the current render:

$$\Delta \mathbf{I}_n^{d,v}(k) = \mathcal{D}_{\mathrm{VAE}}(\mathbf{z}_k^{r,v,*}) - \mathbf{I}_n^{r,v}. \tag{10}$$

4.2. SNR-Modulated Residual Targets

DReSG converts each proposal-render residual into a fitting target whose strength is explicitly controlled. A weak residual may fail to transfer recognizable style cues, whereas an overly amplified residual can saturate colors or force unstable local appearance changes into the Gaussian scene. To account for scheduler-dependent proposal behavior, DReSG constructs a bounded fitting target by modulating residual strength and amplifying the residual in RGB-logit space. Given the current render $\mathbf{I}_n^{r,v}$ and the residual $\Delta \mathbf{I}_n^{d,v}(k)$, the corresponding unscaled diffusion proposal is $\mathbf{I}_n^{r,v} + \Delta \mathbf{I}_n^{d,v}(k)$.

We define the channel-wise RGB logit transform, used only for target construction, as $\ell(\mathbf{I}) = \log \frac{\mathbf{I}}{1-\mathbf{I}}$. We parameterize the residual scale as $\gamma_k = 1 + p_k$, with $p_k \in [0,1]$, preserving the unamplified proposal as the baseline while allowing only bounded extrapolation. Prior diffusion studies report that guidance benefits and representation quality vary across scheduler states and are strongest over intermediate regimes [KAK*24, LZL*26]. We therefore use a state-dependent unimodal schedule rather than a constant gain. Under the variance-preserving scheduler, $\bar{\alpha}_k$ and $1-\bar{\alpha}_k$ are the signal and noise power fractions; their peak-normalized product defines the SNR-balanced schedule:

$$\gamma_k = 1 + p_k^{\mathrm{snr}} = 1 + 4\bar{\alpha}_k(1-\bar{\alpha}_k) = 1 + \frac{4\,\mathrm{SNR}(k)}{(1+\mathrm{SNR}(k))^2}. \tag{11}$$

The resulting scale is smooth and bounded in $[1,2]$, peaks at $\mathrm{SNR}(k) = 1$, and is symmetric under $\mathrm{SNR} \mapsto 1/\mathrm{SNR}$. It returns to $\gamma_k = 1$ when either signal or noise dominates, retaining the proposal while reducing only the extra amplification. As the schedule depends only on the scheduler state, equal-SNR states receive equal scales regardless of schedule discretization. Fig. 3 visualizes this modulation profile and the RGB-logit target construction; the alternative profiles in its top panel are evaluated in Section 5.5.

The final fitting target extrapolates from the current render toward the diffusion proposal in RGB-logit space:

$$\mathbf{I}_n^{t,v}(k) = \sigma\Big(\ell(\mathbf{I}_n^{r,v}) + \gamma_k \Big[\ell(\mathbf{I}_n^{r,v} + \Delta \mathbf{I}_n^{d,v}(k)) - \ell(\mathbf{I}_n^{r,v})\Big]\Big). \tag{12}$$

The target is defined relative to the current render rather than a fixed stylized image. As optimization proceeds across feedback stages, style changes that have already been absorbed by the 3D scene produce smaller residuals in later stages, whereas view-specific proposal details must be reproduced through the shared scene or appear as a remaining fitting discrepancy. RGB-logit scaling only defines the space in which these residual targets are amplified, making it a bounded target-construction step rather than a separate source of 3D structure.

4.3. Multi-View Gaussian Feedback

Proposal targets from multiple views must be absorbed by one shared Gaussian scene to become persistent 3D appearance. Even

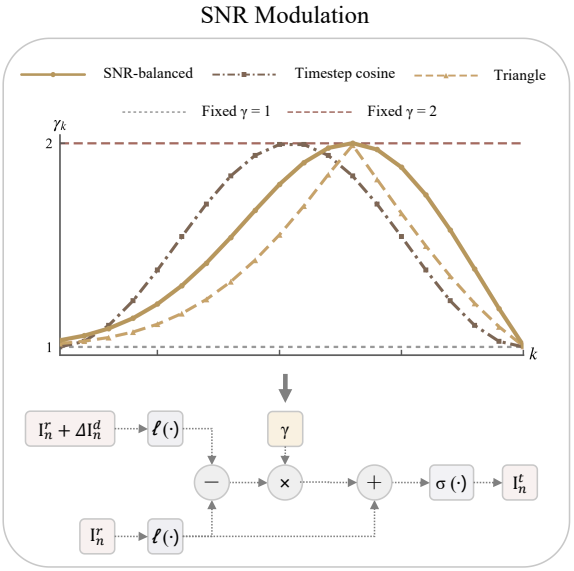

Figure 3: *Residual-strength modulation and RGB-logit target construction. Top: the SNR-balanced schedule and four alternatives evaluated in Section 5.5. Bottom: the RGB-logit construction of the stage target from the current render and diffusion proposal.*

when each view has a reasonable fitting target, different targets may supervise overlapping Gaussians with inconsistent color updates, and using all available views would make proposal generation unnecessarily expensive. DReSG therefore selects a compact set of active views, fits their targets through differentiable multi-view Gaussian rendering, and filters conflicting color updates before applying them to the shared scene. It does not introduce a new Gaussian representation; it optimizes the Gaussian color attributes $\mathbf{c}$ and bounded mean, scale, and rotation offsets, while keeping opacity frozen.

DReSG selects active views offline using visibility-based Gaussian coverage. Let $w_{v,i}$ measure how strongly Gaussian i is visibly observed in candidate view v. We compute this value from projected alpha and opacity contributions after excluding samples whose depth is inconsistent with the rendered depth. With minimum support $w_{\min}$ and target fraction ρ, we compute $m_i^\star = \max_v w_{v,i}$, keep the target-visible set $\mathcal{T} = \{i \mid m_i^\star \geq w_{\min}\}$, and define the per-Gaussian target support

$$q_i = \max(\rho m_i^\star, w_{\min}). \tag{13}$$

For a selected view set $\mathcal{A}$, we define the capped coverage of Gaussian i as

$$c_i(\mathcal{A}) = \min\left(\frac{\max_{u\in\mathcal{A}} w_{u,i}}{q_i}, 1\right). \tag{14}$$

For the empty initial set, we define $c_i(\emptyset) = 0$. Starting from an empty set, DReSG greedily adds the candidate view that maximizes the total increase of capped coverage:

$$v^\star = \arg\max_{v\in\mathcal{C}\setminus\mathcal{A}} \sum_{i\in\mathcal{T}} \left[c_i(\mathcal{A}\cup\{v\}) - c_i(\mathcal{A})\right]. \tag{15}$$

The procedure terminates when the prescribed coverage threshold is reached or the marginal gain falls below its threshold. The selected view set remains fixed during optimization and determines only where proposal targets are generated. The resulting fixed active-view set is denoted by $\mathcal{V}$.

Given the selected active views, DReSG uses a standard per-view image-space fitting objective for residual feedback:

$$\mathcal{L}^v_{\text{fit}} = \mathcal{L}^v_1 + \lambda_{\text{ssim}}\mathcal{L}^v_{\text{DSSIM}} + \lambda_{\text{tv}}\mathcal{L}^v_{\text{tv}} + \lambda_{\text{dino}}\mathcal{L}^v_{\text{DINO}}. \tag{16}$$

Each per-view loss is averaged over image pixels, and the DINO feature loss uses the base render as its content reference. Geometry is updated only through small offsets from the base Gaussian means, scales, and rotations, and these offsets are projected back to preset ranges after each update. This objective grounds independent diffusion proposals in 3D: every target must be reproduced by the same Gaussian scene under differentiable rendering.

Residual feedback from different active views may disagree on overlapping Gaussian regions. DReSG applies color-gradient projection to conflicting view updates during appearance-gradient fusion. Each per-view objective in Eq. 16 induces a gradient on the shared Gaussian color attributes $\mathbf{c}$:

$$\mathbf{g}^c_v = \nabla_{\mathbf{c}}\mathcal{L}^v_{\text{fit}}. \tag{17}$$

If two views u and v propose opposing updates,

$$(\mathbf{g}^c_u)^\top \mathbf{g}^c_v < 0, \tag{18}$$

we use color-gradient projection, inspired by gradient surgery [YKG*20], to remove the directly conflicting component, using $\epsilon > 0$ for numerical stability:

$$\tilde{\mathbf{g}}^c_u = \mathbf{g}^c_u - \frac{(\mathbf{g}^c_u)^\top \mathbf{g}^c_v}{\|\mathbf{g}^c_v\|_2^2 + \epsilon}\mathbf{g}^c_v. \tag{19}$$

The adjusted gradients are then averaged to obtain the final color-update direction:

$$\bar{\mathbf{g}}^c = \frac{1}{|\mathcal{V}|}\sum_{v\in\mathcal{V}} \tilde{\mathbf{g}}^c_v. \tag{20}$$

For multiple active views, DReSG processes gradients in the deterministic active-view order. For each view gradient, it sequentially checks conflicts against the other active-view gradients and updates the current gradient with Eq. 19 whenever the inner product is negative. After all view gradients are processed, the projected gradients are averaged as in Eq. 20. Bounded geometry gradients are fused by a simple mean and constrained by their offset ranges, while opacity receives no update. Thus color-gradient projection suppresses directly conflicting, view-specific local updates during fusion, helping reduce view-dependent artifacts and local style fragmentation without changing the residual targets or feedback schedule.

After each 3D feedback stage, the updated scene is rendered again and re-encoded to initialize the render latent for the next stage, so each residual describes the remaining stylization discrepancy of the current fitted scene rather than a fixed target generated from the initial rendering. A standard affine color-transfer fitting step can be applied at the end [ZKB*22, LLY*25, LLS*26], yielding a stylized Gaussian scene renderable along novel camera paths within the reconstructed scene.

5. Experiments

We evaluate DReSG for reference-guided 3DGS stylization with quantitative results on LLFF [MSOC*19] and qualitative comparisons on both LLFF and Tanks and Temples [KPZK17]. The main quantitative benchmark contains 48 LLFF scene-style pairs. We report main comparisons, ablations, and a user study. In the tables, colored cells indicate the top three entries per main-comparison column and the top two per ablation column.

5.1. Metrics

We report six quantitative metrics that separately assess style alignment, content preservation, and cross-view consistency. CLIP-S measures CLIP [RKH*21] feature similarity between the stylized render and the reference style image; CLIP features capture high-level image appearance and reference-style cues across visual domains. DINO-C measures DINO [CTM*21] feature similarity between the stylized render and the corresponding content image at the same camera pose; DINO features emphasize object- and scene-level correspondence rather than low-level color matching, providing a content-preservation measure. We compute CLIP-S with a pretrained CLIP ViT-B/32 backbone and DINO-C with a pretrained DINO ViT-S/16 backbone. Images are resized to 224×224 and normalized with the corresponding pretrained-model preprocessing. Scores are averaged over all evaluated views, styles, and scenes. The main quantitative evaluation uses 48 LLFF scene-style pairs. For each scene-style pair, all methods are evaluated on the same rendered camera views. We denote short-term and long-term temporal metrics as ST and LT in tables. In compact ablation tables, C-S and D-C abbreviate CLIP-S and DINO-C, while LP and RM abbreviate LPIPS and RMSE.

Table 1 also reports optimization time, peak optimization memory, peak inference memory, and inference FPS. All resource measurements are taken on a single NVIDIA H20 GPU at the same rendering resolution. Optimization time includes diffusion proposal generation and residual fitting, but excludes the initial base 3DGS reconstruction. Inference FPS and inference memory are measured on the final stylized Gaussian scene without diffusion calls.

For temporal consistency, we use the Princeton-VL RAFT optical-flow model [TD20] to align frame pairs. To avoid counting camera motion as stylization flicker, optical flow is estimated on the original base renders and then used to compare the corresponding stylized frames. Short-term metrics use adjacent frames with gap 1, and long-term metrics use gap $N_v/2$, where N_v is the number of evaluated views. RAFT forward–backward consistency and in-bounds checks define valid regions. RMSE is averaged over valid pixels; for LPIPS [ZIE*18], invalid pixels in the warped stylized frame are replaced with target-frame pixels before computing the full-image distance. Lower RAFT-aligned LPIPS and RMSE indicate less appearance drift.

5.2. Baselines

We compare DReSG with representative 3D scene stylization methods covering the main supervision and representation choices in this task. ARF [ZKB*22] is a NeRF-based rendered-view

Figure 4: *Main qualitative comparison on LLFF and Tanks and Temples scenes. Rows are selected to cover reference-style properties discussed in Sec. 1, including directional strokes, coherent contours, region-wise color treatment, and structured style details. All methods are rendered from matched camera poses.*

Table 1: *Main quantitative comparison averaged over evaluated scenes and styles. Temporal columns report RAFT-aligned temporal drift at short and long strides; memory is measured in GB.*

Method	Style	Content	Short-term		Long-term		Efficiency			
	CLIP-S ↑	DINO-C ↑	ST-LPIPS ↓	ST-RMSE ↓	LT-LPIPS ↓	LT-RMSE ↓	Opt. Time (min) ↓	Opt. Mem. (GB) ↓	Infer Mem. (GB) ↓	FPS ↑
Ours	0.6773	0.5803	0.0648	0.0321	0.1045	0.0501	7.80	15.65	1.80	248.88
FantasyStyle	0.6600	0.4260	0.0708	0.0890	0.1100	0.1342	112.90	33.30	2.26	168.60
CLIPGaussian	0.7251	0.4055	0.0993	0.0841	0.1292	0.1176	13.14	5.40	2.25	211.03
SGSST	0.7271	0.1836	0.0865	0.0680	0.1172	0.0986	21.79	2.40	2.26	212.06
ABC-GS	0.6282	0.4438	0.0832	0.0515	0.1140	0.0723	1.06	5.08	2.20	239.52
ARF	0.6674	0.3126	0.1117	0.0766	0.1610	0.1090	3.32	4.45	1.82	10.59

feature-loss baseline; SGSST [GWRM25] and ABC-GS [LLY*25] are recent Gaussian stylization pipelines driven by VGG-style feature losses and patch-level matching; CLIPGaussian [HWB*26] evaluates CLIP-guided multimodal Gaussian stylization; and FantasyStyle [YWW*26] provides a diffusion-based 3D stylization comparison. This set covers the principal alternatives to DReSG's residual-feedback formulation: rendered-view feature optimization, patch-level Gaussian fitting, CLIP guidance, and diffusion-guided 3D stylization.

Unless otherwise stated, all methods use the same input scene, reference style image, base Gaussian reconstruction, and evaluation cameras. Gaussian-based methods are initialized from the same base reconstruction for each scene, since reconstruction defects can otherwise dominate perceived stylization quality. ARF is evaluated with its method-specific NeRF checkpoint; this method is not defined for Gaussian checkpoints. We run baselines with official implementations and released configurations when available, and otherwise use the authors' default hyperparameters.

5.3. Implementation Details

DReSG starts from a Fast-PGSR all-view base built on FastGS acceleration and PGSR reconstruction [RWFL25, CLY*24]. It optimizes Gaussian color attributes $\mathbf{c}$ and small offsets to means, scales, and rotations, while opacity is frozen; geometry offsets are projected back to preset ranges after each update. Active views are

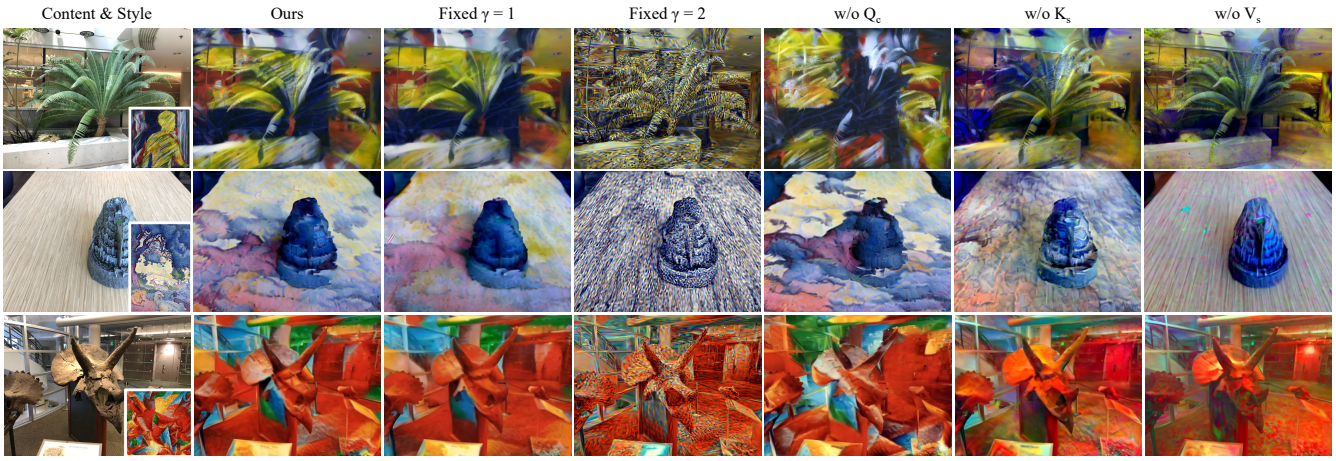

Figure 5: *Attention-guided residual construction and representative fixed-schedule ablations. The schedule columns show SNR-balanced and the two fixed endpoints; all five residual-strength schedules are compared quantitatively in Table 3.*

selected once before optimization using the offline coverage strategy in Section 4.3. The default setting uses $w_{\min} = 10^{-4}$, $\rho = 0.98$, coverage-stop ratio 0.9999, and marginal-gain threshold 0.001. Selection begins from the empty set and terminates when either prescribed stopping condition is met; it typically retains approximately 20 LLFF views and 56–70 Tanks and Temples views. LLFF scenes are rendered at factor four, whereas Tanks and Temples scenes use their native resolution. All renderings and reference images are resized to 448×320 before being passed to the diffusion model.

At each feedback stage, we render the current 3D scene from the active views, generate attention-guided diffusion proposals with Stable Diffusion v1.5 using the attention energy in Eq. 9, compute render-relative residual targets with RGB-logit scaling, and fit the scene to those targets. This logit scaling is applied only during target construction. The main setting uses 20 feedback stages, whose scheduler states are sampled uniformly from a 200-step DDIM schedule. Attention features are extracted from zero-indexed U-Net self-attention layers 10–15. After all attention-guided residual stages are complete, we run the post affine color-transfer fitting step described above. The default configuration uses γ_k from the SNR-balanced schedule in Eq. 11, content weight 0.15 for LLFF and 0.10 for Tanks and Temples, appearance learning rate 0.05, and the fitting loss in Eq. 16 with $\lambda_{\text{ssim}} = 0.2$, $\lambda_{\text{tv}} = 0.01$, and $\lambda_{\text{dino}} = 0.01$. DReSG performs one latent Adam update at each DDIM timestep, for a total of 10 updates per feedback stage; each feedback stage uses 30 inner steps for 3D residual fitting, and the final post color-transfer fitting uses 50 additional steps. We apply color-gradient projection to $\mathbf{c}$ when fusing view-wise appearance updates. For performance analysis, we measure optimization time, peak optimization memory, peak inference memory, and inference FPS.

5.4. Main Results

Table 1 reports quality and efficiency metrics for the final stylized 3D scene rather than only the selected active views. The measurements support a multi-objective conclusion: DReSG attains the highest DINO-C and the lowest short- and long-term drift, indicating strong content preservation and persistent stylized appearance under viewpoint changes, although its CLIP-S remains below the highest reported value. Its optimization is faster than CLIPGaussian and SGSST and requires substantially less time than FantasyStyle; at inference, it achieves the highest reported FPS and lowest memory. These resource measurements make the intended trade-off explicit: DReSG spends additional time constructing proposal-derived residual targets and fitting them into the shared scene, but the final edited scene remains compact and fast to render.

Figure 4 compares how the methods reproduce the style cues described in Section 1. VGG-feature-based baselines often preserve coarse layout but fragment directional strokes or coherent contours into local texture responses. The diffusion-based baseline can generate stronger style cues for region-wise color treatment or structured style details, but may over-saturate colors, overwrite content, or detach details from object boundaries. In these cases, DReSG better preserves object boundaries, occlusion relationships, and foreground-background separation while retaining recognizable style-reference cues. These observations align with the quantitative results, which show strong content preservation and reduced view-dependent drift alongside reference-style transfer.

5.5. Ablation Studies

We conduct component-wise ablations over four factors: attention-guided proposal construction, SNR-based residual-strength modulation, shared-scene feedback, and RGB-logit target construction. These ablations assess trade-offs rather than identify a single-metric winner: some variants improve an isolated metric by remaining closer to the input or weakening style, whereas the full model is selected for its balance across style strength, content preservation, and view stability. All comparisons hold the style image, selected active-view set, post color-transfer configuration, and optimization budget constant. Figures 5, 6, and 7 provide qualitative evidence for

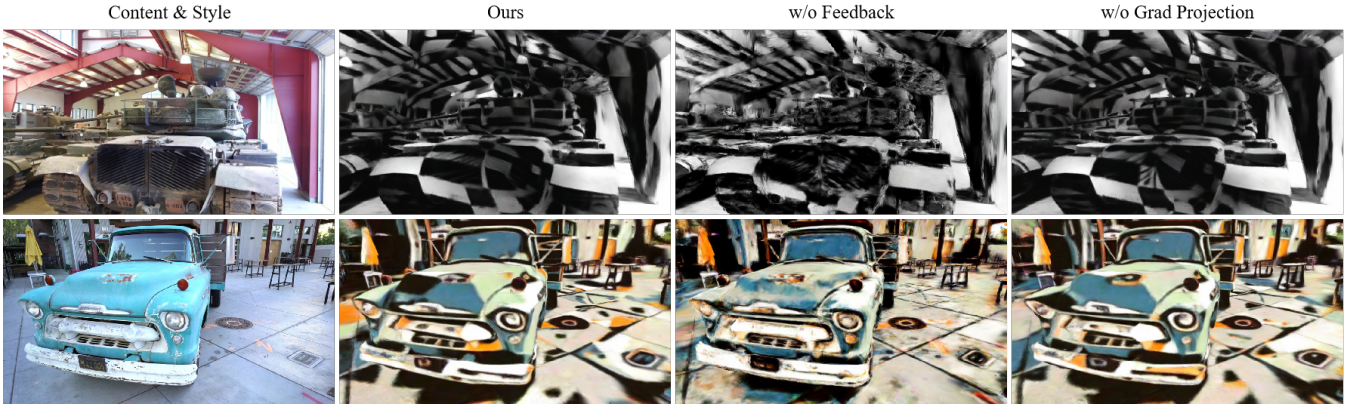

Figure 6: *3D-grounded feedback ablation. We compare DReSG with variants that remove residual feedback or color-gradient projection under matched scenes, styles, and viewpoints.*

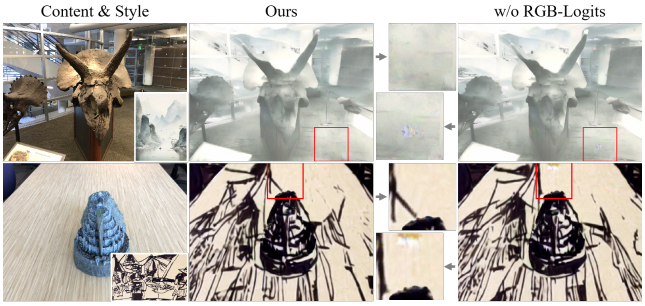

Figure 7: *RGB-logit target scaling ablation. We compare DReSG with an image-space residual-scaling variant under matched scenes, styles, and viewpoints.*

Table 2: *Attention-guided residual construction ablation. Q_c is the content query, and K_s, V_s are style keys and values. Metric abbreviations are defined in Section 5.1.*

Variant	C-S ↑	D-C ↑	ST-LP ↓	ST-RM ↓	LT-LP ↓	LT-RM ↓
DReSG	0.736	0.556	0.061	0.033	0.101	0.053
w/o Q_c	0.779	0.407	0.063	0.035	0.104	0.057
w/o K_s	0.655	0.625	0.064	0.033	0.104	0.053
w/o V_s	0.594	0.695	0.062	0.033	0.101	0.050

these factors, while Tables 4, 2, and 3 summarize the corresponding metrics.

For the schedule comparison, let $F = N/S$ be the number of feedback stages and k_n the scheduler state used at stage n; the stage-wise scale is $\gamma_{k_n} = 1 + p_n$. SNR-balanced and Triangle use the scheduler coordinate $\bar{\alpha}_{k_n}$, with Triangle matching the range of SNR-balanced but using a piecewise-linear profile. Timestep cosine is defined over the feedback-stage index and places its peak at the midpoint of that index range. The comparison also includes the fixed endpoints $\gamma = 1$ and $\gamma = 2$.

Table 2 reports the attention-guided residual construction ablation. Removing K_s or V_s reduces CLIP-S, while removing Q_c increases CLIP-S but substantially reduces DINO-C. The qualitative comparisons in Fig. 5 illustrate these trade-offs, supporting the complementary roles of the content query and style keys/values rather than a single-metric ranking of the variants.

Table 3: *Five-schedule residual-strength ablation under the same evaluation and optimization settings. Metric abbreviations are defined in Section 5.1.*

Schedule	Extra scale p_n	C-S ↑	D-C ↑
SNR-balanced	$4\bar{\alpha}_{k_n}(1-\bar{\alpha}_{k_n})$	0.7362	0.5557
Fixed $\gamma = 1$	0	0.7291	0.5467
Fixed $\gamma = 2$	1	0.6617	0.5316
Triangle	$1 - \lvert 2\bar{\alpha}_{k_n} - 1 \rvert$	0.7357	0.5370
Timestep cosine	$\sin^2(\pi(n-1)/(F-1))$	0.7361	0.5277

Table 4: *Quantitative feedback ablation on the matched scene-style subset used for feedback variants.*

Variant	C-S ↑	D-C ↑	ST-LP ↓	ST-RM ↓	LT-LP ↓	LT-RM ↓
DReSG	0.7394	0.5592	0.0624	0.0339	0.1027	0.0548
w/o color-grad proj.	0.7373	0.5549	0.0623	0.0341	0.1023	0.0548
w/o feedback	0.6670	0.6773	0.0718	0.0400	0.1087	0.0611

Table 3 isolates residual-strength scheduling. The three unimodal schedules yield nearly identical CLIP-S values. Among them, SNR-balanced attains the highest DINO-C and outperforms both fixed-scale baselines on both metrics. We therefore adopt SNR-balanced; its smooth, scheduler-aligned profile is consistent with the motivation in Section 4.2 and achieves the best measured style–content balance in this five-schedule comparison.

The feedback ablation isolates the render-to-latent path that closes the stage-wise loop. In the *w/o feedback* variant, each active view retains its current attention-optimized latent after 3D fitting instead of replacing it with the VAE encoding of the updated render; residual-target construction and shared-scene fitting are oth-

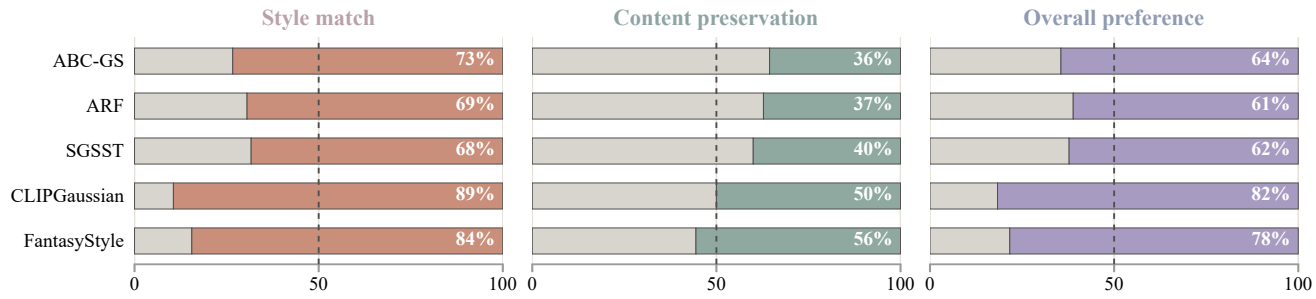

Figure 8: *Pairwise user study. We report the percentage of votes selecting DReSG over each baseline for style match, content preservation, and overall preference.*

erwise unchanged. Removing this path lowers style strength and increases short- and long-term drift in Table 4, while Fig. 6 shows local artifacts and view-inconsistent details. Without color-gradient projection, conflicting style updates from different views are fused directly. Although the aggregate metrics remain close, the paired examples in Fig. 6 show greater local style fragmentation without projection, qualitatively supporting its role in mitigating conflicting appearance updates during multi-view fusion.

The RGB-logit ablation compares the full model with a variant that scales residuals directly in image space. Direct image-space extrapolation can produce saturated or locally unstable targets; Fig. 7 shows that removing RGB-logit scaling introduces localized color artifacts, whereas RGB-logit construction yields cleaner, bounded appearance changes under the same feedback setting. This comparison explains why the residual target is formed in RGB-logit space before being fitted into the Gaussian scene.

5.6. User Study

We conduct a pairwise user study to evaluate perceived style match, content preservation, and overall quality. The study uses six scene-style pairs and compares DReSG against five baselines with 30 participants. For each baseline and criterion, this design yields 180 pairwise votes. Trial order and left-right order are randomized, and method names are hidden.

Participants answer three preference questions: which result has stronger style expression, which result better preserves the original content, and which result is preferred overall. This protocol separates the two main perceptual objectives from the final trade-off, since a result can appear strongly stylized while damaging the scene, or preserve content while appearing weakly stylized. As shown in Fig. 8, DReSG is preferred for style match against all baselines, with rates from 68.33% against SGSST to 89.44% against CLIPGaussian. Content-preservation votes reveal the style–content trade-off: conservative VGG-feature-based baselines can be favored for input preservation alone, while DReSG reaches parity with CLIPGaussian and is preferred over FantasyStyle. For overall preference, DReSG receives more than half of the votes against every baseline, from 61.11% against ARF to 81.67% against CLIPGaussian, suggesting that participants favor its combined style-transfer and scene-preservation trade-off.

6. Conclusion and Limitations

This paper introduced DReSG for reference-guided stylization of scenes represented by 3DGS. DReSG converts attention-guided diffusion proposals into proposal-render residual targets, constructs bounded RGB-logit fitting targets, modulates residual strength with an SNR-balanced schedule, and fits the targets to a shared Gaussian scene through multi-view rendering. Quantitative results on LLFF and qualitative comparisons on LLFF and Tanks and Temples show that DReSG improves the balance between reference-specific stylization, structure preservation, and cross-view persistence compared with representative 3D stylization baselines; the ablation results clarify how residual construction, residual modulation, and shared scene fitting contribute to that balance.

DReSG has three main limitations. First, its performance is limited by the proposal source: the quality and resolution of attention-guided diffusion proposals limit the residual targets and thus the final stylized 3D scene. Stronger high-resolution, multi-scale, or multi-view diffusion models may provide more reliable proposal targets. Second, DReSG depends on the quality of the input 3DGS. Reconstruction artifacts, missing geometry, or weakly anchored regions can affect stylization stability, and reconstruction-aware extensions could allocate Gaussians more effectively in thin structures or regions containing fine details or enrich their appearance and geometry attributes. Extending DReSG to geometry reconstruction or repair remains future work. Sparse active views can also under-supervise rarely visible regions, motivating adaptive view expansion for broader camera paths. Third, DReSG is not feed-forward. Each scene-style pair still requires iterative diffusion guidance and 3DGS optimization; distilling this process into a scene/style-conditioned predictor for Gaussian attributes is a promising direction.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant Nos. 62572194, 62472178, and 62376244) and the Shanghai Frontiers Science Center of Molecule Intelligent Syntheses. This work was also supported by the Key Technology Research and Development Program of the Shanghai Science and Technology Commission (Grant Nos. 25511107200 and 25511102400).

References

[CLV24] CHEN M., LAINA I., VEDALDI A.: Dge: Direct gaussian 3d editing by consistent multi-view editing. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part LXXIV* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 74–92. URL: https://doi.org/10.1007/978-3-031-72904-1_5, doi:10.1007/978-3-031-72904-1_5. 2, 3

[CLY*24] CHEN D., LI H., YE W., WANG Y., XIE W., ZHAI S., WANG N., LIU H., BAO H., ZHANG G.: Pgsr: Planar-based gaussian splatting for efficient and high-fidelity surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics PP* (01 2024), 1–12. doi:10.1109/TVCG.2024.3494046. 7

[CTM*21] CARON M., TOUVRON H., MISRA I., JÉGOU H., MAIRAL J., BOJANOWSKI P., JOULIN A.: Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (2021), pp. 9650–9660. 6

[FMH24] FUJIWARA H., MUKUTA Y., HARADA T.: Style-nerf2nerf: 3d style transfer from style-aligned multi-view images. In *SIGGRAPH Asia 2024 Conference Papers* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery. URL: https://doi.org/10.1145/3680528.3687643, doi:10.1145/3680528.3687643. 2

[GEB16] GATYS L. A., ECKER A. S., BETHGE M.: Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 2414–2423. doi:10.1109/CVPR.2016.265. 2

[GWRM25] GALERNE B., WANG J., RAAD L., MOREL J.-M.: Sgsst: scaling gaussian splatting style transfer. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 26535–26544. 2, 7

[HB17] HUANG X., BELONGIE S.: Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 1501–1510. 2

[HHY*22] HUANG Y.-H., HE Y., YUAN Y.-J., LAI Y.-K., GAO L.: Stylizednerf: consistent 3d scene stylization as stylized nerf via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 18342–18352. 2

[HJA20] HO J., JAIN A., ABBEEL P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems 33* (2020), 6840–6851. 2, 4

[HTE*23] HAQUE A., TANCIK M., EFROS A., HOLYNSKI A., KANAZAWA A.: Instruct-NeRF2NeRF: Editing 3D scenes with instructions. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)* (10 2023), pp. 19683–19693. doi:10.1109/ICCV51070.2023.01808. 2, 3

[HTS*21] HUANG H.-P., TSENG H.-Y., SAINI S., SINGH M., YANG M.-H.: Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021), pp. 13869–13878. 2

[HWB*26] HOWIL K., WACZYNSKA J., BORYCKI P., DZIARMAGA T., MAZUR M., SPUREK P.: Clipgaussian: Universal and multimodal style transfer based on gaussian splatting. *Advances in neural information processing systems 38* (2026), 112125–112168. 2, 3, 7

[IKN24] IBRAHIMLI N., KOOIJ J. F., NAN L.: Muviecast: multi-view consistent artistic style transfer. In *2024 International Conference on 3D Vision (3DV)* (2024), IEEE, pp. 1136–1145. 2

[JAFF16] JOHNSON J., ALAHI A., FEI-FEI L.: Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision* (2016), Springer, pp. 694–711. 2

[KAK*24] KYNKÄÄNNIEMI T., AITTALA M., KARRAS T., LAINE S., AILA T., LEHTINEN J.: Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *Advances in Neural Information Processing Systems 37* (2024), 122458–122483. 5

[KKLD23] KERBL B., KOPANAS G., LEIMKUEHLER T., DRETTAKIS G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph. 42*, 4 (July 2023). URL: https://doi.org/10.1145/3592433, doi:10.1145/3592433. 3

[KPZK17] KNAPITSCH A., PARK J., ZHOU Q.-Y., KOLTUN V.: Tanks and temples: benchmarking large-scale scene reconstruction. *ACM Trans. Graph. 36*, 4 (July 2017). URL: https://doi.org/10.1145/3072959.3073599, doi:10.1145/3072959.3073599. 6

[LFY*17] LI Y., FANG C., YANG J., WANG Z., LU X., YANG M.-H.: Universal style transfer via feature transforms. *Advances in neural information processing systems 30* (2017). 2

[LGT*22] LIN C.-H., GAO J., TANG L., TAKIKAWA T., ZENG X., HUANG X., KREIS K., FIDLER S., LIU M.-Y., LIN T.-Y.: Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440* (2022). 3

[LLS*26] LIU W., LIU Z., SHU J., WANG C., LI Y.: GT2-GS: Geometry-aware texture transfer for gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2026), vol. 40, pp. 7296–7304. 3, 6

[LLY*25] LIU W., LIU Z., YANG X., SHA M., LI Y.: ABC-GS: Alignment-based controllable style transfer for 3D gaussian splatting. In *2025 IEEE International Conference on Multimedia and Expo (ICME)* (06 2025), pp. 1–6. doi:10.1109/ICME59968.2025.11209430. 2, 6, 7

[LZC*23] LIU K., ZHAN F., CHEN Y., ZHANG J., YU Y., EL SADDIK A., LU S., XING E. P.: Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2023), pp. 8338–8348. 2

[LZL*26] LI X., ZHANG Z., LI X., CHEN S., ZHU Z., WANG P., QU Q.: Understanding representation dynamics of diffusion models via low-dimensional modeling. *Advances in Neural Information Processing Systems 38* (2026), 107365–107404. 5

[LZX*24] LIU K., ZHAN F., XU M., THEOBALT C., SHAO L., LU S.: Stylegaussian: Instant 3d style transfer with gaussian splatting. In *SIGGRAPH Asia 2024 Technical Communications* (New York, NY, USA, 2024), SA '24, Association for Computing Machinery. URL: https://doi.org/10.1145/3681758.3698002, doi:10.1145/3681758.3698002. 2

[MSOC*19] MILDENHALL B., SRINIVASAN P. P., ORTIZ-CAYON R., KALANTARI N. K., RAMAMOORTHI R., NG R., KAR A.: Local light field fusion: practical view synthesis with prescriptive sampling guidelines. *ACM Trans. Graph. 38*, 4 (July 2019). URL: https://doi.org/10.1145/3306346.3322980, doi:10.1145/3306346.3322980. 6

[MST*21] MILDENHALL B., SRINIVASAN P. P., TANCIK M., BARRON J. T., RAMAMOORTHI R., NG R.: Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM 65*, 1 (Dec. 2021), 99–106. URL: https://doi.org/10.1145/3503250, doi:10.1145/3503250. 2

[NPLX22] NGUYEN-PHUOC T., LIU F., XIAO L.: Snerf: stylized neural implicit representations for 3d scenes. *ACM Trans. Graph. 41*, 4 (July 2022). URL: https://doi.org/10.1145/3528223.3530107, doi:10.1145/3528223.3530107. 2

[PJBM22] POOLE B., JAIN A., BARRON J. T., MILDENHALL B.: Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022). 2, 3

[RBL*22] ROMBACH R., BLATTMANN A., LORENZ D., ESSER P., OMMER B.: High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2022), pp. 10684–10695. 2, 4

[RKH*21] RADFORD A., KIM J. W., HALLACY C., RAMESH A., GOH G., AGARWAL S., SASTRY G., ASKELL A., MISHKIN P., CLARK J., ET AL.: Learning transferable visual models from natural language

supervision. In *International conference on machine learning* (2021), PmLR, pp. 8748–8763. 2, 6

[RWFL25] REN S., WEN T., FANG Y., LU B.: Fastgs: Training 3d gaussian splatting in 100 seconds. *ArXiv abs/2511.04283* (2025). URL: https://api.semanticscholar.org/CorpusID:282812148. 7

[SWY*23] SHI Y., WANG P., YE J., MAI L., LI K., YANG X.: Mvdream: Multi-view diffusion for 3d generation. *arXiv:2308.16512* (2023). 3

[TD20] TEED Z., DENG J.: Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision* (2020), Springer, pp. 402–419. 6

[WBL*24] WU J., BIAN J.-W., LI X., WANG G., REID I., TORR P., PRISACARIU V. A.: Gaussctrl: Multi-view consistent text-driven 3d gaussian splatting editing. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XIV* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 55–71. URL: https://doi.org/10.1007/978-3-031-72630-9_4, doi:10.1007/978-3-031-72630-9_4. 3

[WDL*23] WANG H., DU X., LI J., YEH R., SHAKHNAROVICH G.: Score jacobian chaining: Lifting pretrained 2D diffusion models for 3D generation. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (06 2023), pp. 12619–12629. doi:10.1109/CVPR52729.2023.01214. 2, 3

[WFZ*24] WANG J., FANG J., ZHANG X., XIE L., TIAN Q.: GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions . In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (Los Alamitos, CA, USA, June 2024), IEEE Computer Society, pp. 20902–20911. URL: https://doi.ieeecomputersociety.org/10.1109/CVPR52733.2024.01975, doi:10.1109/CVPR52733.2024.01975. 2, 3

[WLW*23] WANG Z., LU C., WANG Y., BAO F., LI C., SU H., ZHU J.: Prolificdreamer: high-fidelity and diverse text-to-3d generation with variational score distillation. In *Proceedings of the 37th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2023), NIPS '23, Curran Associates Inc. 2, 3

[WWB*24] WANG H., WANG Q., BAI X., QIN Z., CHEN A.: Instantstyle: Free lunch towards style-preserving in text-to-image generation. *arXiv preprint arXiv:2404.02733* (2024). 2

[WYW*24] WANG Y., YI X., WU Z., ZHAO N., CHEN L., ZHANG H.: View-consistent 3d editing with gaussian splatting. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29 – October 4, 2024, Proceedings, Part XXXV* (Berlin, Heidelberg, 2024), Springer-Verlag, p. 404–420. URL: https://doi.org/10.1007/978-3-031-72761-0_23, doi:10.1007/978-3-031-72761-0_23. 3

[YKG*20] YU T., KUMAR S., GUPTA A., LEVINE S., HAUSMAN K., FINN C.: Gradient surgery for multi-task learning. *Advances in neural information processing systems 33* (2020), 5824–5836. 6

[YWW*26] YANG Y., WANG Y., WANG C., WANG H., HE S.: FantasyStyle: Controllable stylized distillation for 3D gaussian splatting. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2026), vol. 40, pp. 11784–11792. 2, 3, 7

[YZL*23] YE H., ZHANG J., LIU S., HAN X., YANG W.: Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721* (2023). 2

[ZCH*25] ZHU L., CAI S., HUANG S., WETZSTEIN G., KHOSRAVAN N., ARMENI I.: Scene-level appearance transfer with semantic correspondences. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers* (New York, NY, USA, 2025), SIGGRAPH Conference Papers '25, Association for Computing Machinery. URL: https://doi.org/10.1145/3721238.3730655, doi:10.1145/3721238.3730655. 2

[ZGCH25] ZHOU Y., GAO X., CHEN Z., HUANG H.: Attention distillation: A unified approach to visual characteristics transfer. In *Proceedings of the Computer Vision and Pattern Recognition Conference* (2025), pp. 18270–18280. 4

[ZHT*23] ZHANG Y., HUANG N., TANG F., HUANG H., MA C., DONG W., XU C.: Inversion-based style transfer with diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (2023), pp. 10146–10156. 2

[ZIE*18] ZHANG R., ISOLA P., EFROS A. A., SHECHTMAN E., WANG O.: The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2018), pp. 586–595. 6

[ZKB*22] ZHANG K., KOLKIN N., BI S., LUAN F., XU Z., SHECHTMAN E., SNAVELY N.: Arf: Artistic radiance fields. In *European conference on computer vision* (2022), Springer, pp. 717–733. 1, 2, 6

[ZYC*25] ZHANG D., YUAN Y.-J., CHEN Z., ZHANG F.-L., HE Z., SHAN S., GAO L.: Stylizedgs: Controllable stylization for 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025). 1, 2