LIFTING 2D STYLIZERS TO 3D: PLUG-AND-PLAY STYLE TRANSFER FOR GAUSSIAN SPLATTING

Anonymous authors
Paper under double-blind review

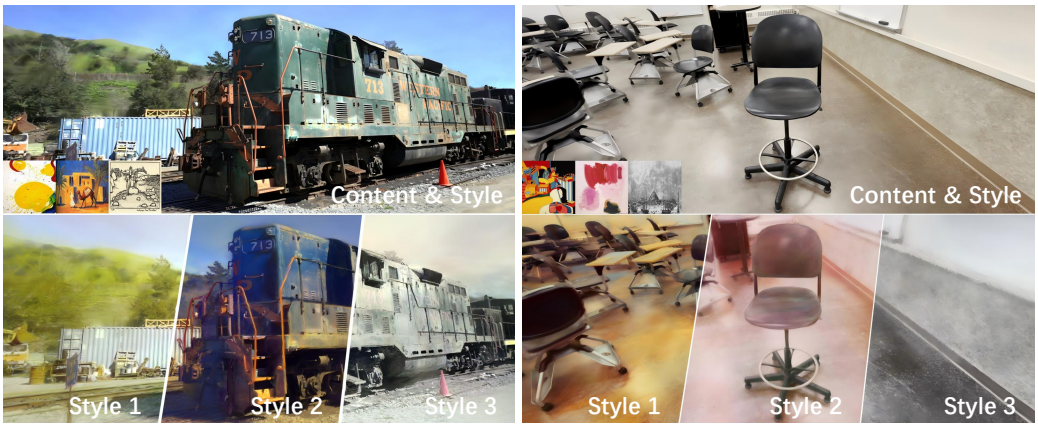


Figure 1: StyleLift3D lifts diverse styles from 2D stylizers into existing 3D Gaussian Splatting scenes, enabling consistent stylized 3D rendering across both indoor and outdoor environments.

ABSTRACT

We present StyleLift3D, a plug-and-play feed-forward framework for transferring expressive 2D stylization to existing 3D Gaussian Splatting scenes. Existing methods typically require per-scene optimization or reconstruct a new stylized representation, while independently stylized multi-view images provide inconsistent supervision across viewpoints. StyleLift3D instead lifts such 2D style cues directly into a coherent 3D representation in a single forward pass. Our framework combines three components: Geometry-Aligned Style Lifting aligns multi-view style features with the content scene; Octree-Local Style Fusion establishes fine-grained style-content correspondence through scalable local cross-attention; and Structure-Preserving Gaussian Decoding predicts residual appearance and bounded geometry updates, with optional dynamic splitting for additional capacity. By performing style fusion directly in 3D, StyleLift3D preserves scene structure while maintaining consistent stylization across novel viewpoints. Experiments on diverse scenes, styles, and 2D stylizers demonstrate improved content fidelity, favorable overall user-study preference, efficient single-forward lifting, and plug-and-play compatibility with multiple 2D stylizers, compared with the evaluated optimization-based and feed-forward baselines.

1 INTRODUCTION

3D Gaussian Splatting (3DGS) has emerged as a powerful representation for photorealistic reconstruction and real-time novel-view rendering (Kerbl et al., 2023). Its explicit Gaussian primitives offer high visual quality, efficient rendering, and direct control over appearance and geometry, making 3DGS an appealing foundation for editable 3D content. Artistic stylization of existing 3DGS scenes could benefit AR/VR, games, and virtual production (Fei et al., 2024; Chen et al., 2024a; Xu et al., 2025b). Unlike stylizing a single image, 3D stylization must preserve the scene structure while maintaining a coherent appearance across arbitrary viewpoints.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Most existing 3DGS stylization methods are optimization-based: they adapt a reconstructed scene to each target style through per-scene training (Liu et al., 2024; Galerne et al., 2025; Zhang et al., 2025; Lin et al., 2025). Although generally effective at preserving content, repeated optimization is costly and often confines stylization to appearance changes. Recent feed-forward approaches reduce this cost but remain tied to particular formulations: A3GS (Fang et al., 2025) transfers global appearance statistics to an existing Gaussian scene through AdaIN (Huang & Belongie, 2017), whereas Stylos (Liu et al., 2026) reconstructs a new stylized Gaussian representation from multi-view images. Meanwhile, diffusion-based image stylizers have achieved rich, diverse, and high-quality 2D stylization. A general feed-forward mechanism for spatially lifting their expressive style cues into an existing 3DGS scene, while retaining its structure, remains underexplored.

A seemingly straightforward route is to render an existing scene from multiple viewpoints, stylize each view independently with a pre-trained image stylizer, and optimize a 3DGS representation against the stylized views. This route, however, retains costly per-scene optimization and treats mutually inconsistent views as simultaneous supervision. Their disagreements are then projected into the same 3D scene, often producing blurred textures, ghosting, and unstable novel-view appearance. These limitations lead to our central question: *Can expressive 2D style information be efficiently lifted into an existing 3DGS scene while preserving its content and forming a coherent stylized representation?* Two challenges are central. First, spatially varying cues from inconsistent 2D views must be geometrically aligned with irregular Gaussian primitives without corrupting the content structure. Second, their fusion must scale to large scenes without discarding fine-grained, spatially local style patterns.

To address these challenges, we propose **StyleLift3D**, a plug-and-play feed-forward framework that lifts expressive 2D stylization into an existing 3DGS scene in three stages. Its modular interface accepts multi-view guidance from off-the-shelf 2D stylizers without modifying or retraining them. First, **Geometry-Aligned Style Lifting (GASL)** uses depth to anchor stylized multi-view features in the content coordinate frame, while a pointwise encoder converts the input Gaussians into geometry-preserving content tokens. Second, **Octree-Local Style Fusion (OLSF)** partitions the shared 3D space into octree leaves and performs cross-attention only among spatially aligned content and style features within each leaf, enabling local correspondence without the prohibitive cost of global interaction. Finally, **Structure-Preserving Gaussian Decoding (SPGD)** predicts residual appearance and bounded geometry updates, with optional dynamic splitting providing additional capacity for fine-grained stylization. By fusing style once in 3D, StyleLift3D produces a single coherent representation that can be rendered consistently from arbitrary viewpoints without per-scene optimization. Experiments across diverse scenes and styles demonstrate strong style fidelity, multi-view consistency, efficiency, and generalization relative to both optimization-based and feed-forward baselines.

Our main contributions are summarized as follows:

- We introduce StyleLift3D, a plug-and-play feed-forward framework that lifts the outputs of off-the-shelf 2D stylizers into existing 3DGS scenes. It produces a coherent stylized 3D representation in a single forward pass, requiring neither per-scene optimization nor reconstruction from stylized views.
- We develop a geometry-aware style propagation architecture comprising GASL, OLSF, and SPGD, which respectively lift multi-view style evidence into a shared 3D space, establish local style-content correspondence through scalable octree-local attention, and decode structure-preserving appearance and geometry updates with optional dynamic splitting.
- We construct a 9K-scale multi-view stylization dataset that pairs existing 3DGS scenes with independently stylized observations from diverse 2D stylizers and WikiArt styles, providing large-scale supervision for learning coherent 3D stylization from inherently inconsistent 2D views.

2 Related Work

2.1 2D Style Transfer

Neural style transfer was introduced by matching the Gram statistics of deep features (Gatys et al., 2015), then accelerated to real-time inference with perceptual feed-forward networks (Johnson et al.,

2016). AdaIN further enabled arbitrary-style transfer by aligning channel-wise feature statistics (Huang & Belongie, 2017). More recently, diffusion models have provided stronger generative priors and flexible conditional control (Zhang et al., 2023). Training-free methods such as StyleSSP (Xu et al., 2025a), StyleGallery (He et al., 2026), and AttenST (Huang et al., 2025) exploit these priors to produce expressive stylization from image references without task-specific model training. Despite their strong per-image results, these methods process views independently and provide no mechanism for enforcing 3D consistency. StyleLift3D treats such stylizers as interchangeable guidance generators and resolves their cross-view discrepancies in a shared 3D representation.

2.2 3DGS Stylization methods

Most 3DGS stylization methods optimize an already reconstructed scene for each target style. StyleGaussian decodes per-view VGG features into Gaussian colors (Liu et al., 2024); SGSST improves scalability with multi-scale Gram matching (Galerne et al., 2025); StylizedGS preserves geometry through depth-based regularization (Zhang et al., 2025); and Multi-StyleGS associates multiple styles with semantic regions (Lin et al., 2025). WaSt-3D further transfers geometric characteristics between Gaussian scenes through a Wasserstein objective (Kotovenko et al., 2024). Although these methods offer strong scene-specific stylization, their optimization must be repeated for new scene–style pairs. Feed-forward alternatives remove this repeated optimization but adopt different forms of global conditioning or scene generation: A3GS applies AdaIN to a graph-convolutional scene latent (Fang et al., 2025), Stylos predicts a stylized Gaussian scene from unposed multi-view images (Liu et al., 2026), AnyStyle injects CLIP embeddings into a reconstructed scene (Kaleta et al., 2026), and FPGS performs semantic-aware photorealistic transfer with AdaIN in 3D space (Kim et al., 2026). In contrast, StyleLift3D spatially lifts the outputs of pluggable 2D stylizers into an existing Gaussian scene and fuses them through geometry-aware local attention.

2.3 Feed-Forward 3DGS Reconstruction

Feed-forward geometry models such as VGGT (Wang et al., 2025), together with generalizable Gaussian predictors, have made single-pass 3D reconstruction practical. pixelSplat predicts Gaussian primitives from image pairs using epipolar constraints (Charatan et al., 2024), MVSplat builds plane-swept cost volumes for sparse multi-view reconstruction (Chen et al., 2024b), and AnySplat handles unconstrained, unposed image collections (Jiang et al., 2025). Recent work further improves the structure and coverage of predicted Gaussians: Z-Order Transformer organizes unstructured primitives into spatially coherent sequences for efficient sparse attention (Wang et al., 2026); LocusGS equips Gaussian queries with explicit 3D anchors to promote local spatial coherence (Li et al., 2026); and ReconSplat couples feed-forward 3DGS with multi-view latent diffusion to reconstruct plausible, geometrically consistent content beyond observed views (Stracquadanio et al., 2026). These advances demonstrate strong cross-scene generalization, but their objective is to reconstruct a scene from input observations. Reconstructing directly from independently stylized views entangles geometry estimation with cross-view appearance discrepancies. StyleLift3D instead takes an existing 3DGS scene as a geometric anchor and performs only feed-forward style propagation, avoiding reconstruction from inconsistent stylized views.

3 Method

Overview. StyleLift3D transfers the outputs of a pluggable 2D stylizer to an input 3DGS scene in a single forward pass. As illustrated in Figure 2, the input Gaussians and independently stylized views are first embedded in a shared 3D coordinate fram. Geometry-aligned lifting anchors multi-view style features to the input geometry, octree-local fusion establishes spatial correspondence between style voxels and Gaussian tokens, and structure-preserving decoding converts the fused tokens into bounded residual updates. The resulting Gaussians form one coherent stylized scene and can be rendered from arbitrary viewpoints without per-scene optimization.

3.1 Preliminaries

A 3DGS scene is represented by a set of Gaussian primitives:

$$G = \{(\mu_i, s_i, q_i, \alpha_i, c_i)\}_{i=1}^{N}, \tag{1}$$

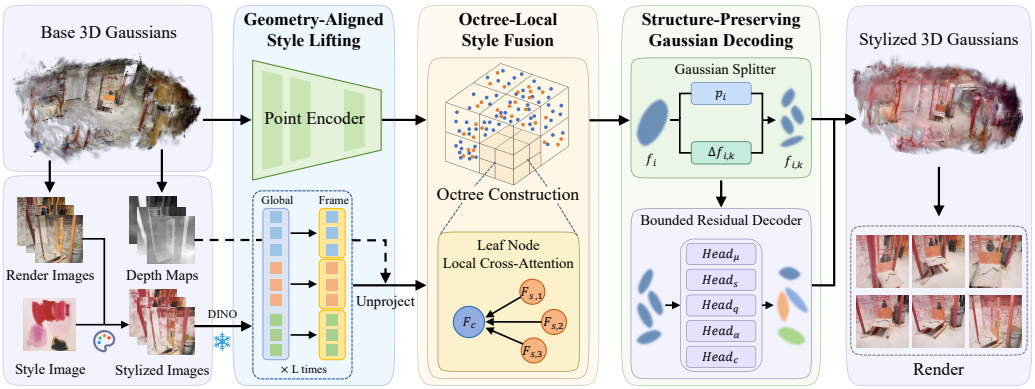


Figure 2: Overview of StyleLift3D. The base 3DGS is encoded into content tokens, while independently stylized views are lifted with depth into the same 3D space. Gaussian content and lifted style features within each octree leaf are fused and decoded into a stylized Gaussian scene.

where $\mu_i \in \mathbb{R}^3$ is the center, $s_i \in \mathbb{R}^3$ is the anisotropic scale, $q_i \in \mathbb{R}^4$ is a rotation quaternion, α_i is the opacity, and c_i contains the spherical-harmonic (SH) appearance coefficients (Kerbl et al., 2023). The scale and rotation define the covariance $\Sigma_i = R(q_i)\,\mathrm{diag}(s_i^2)R(q_i)^\top$. Given camera intrinsics K and pose P, differentiable Gaussian rasterization projects the primitives and renders an image $I = \mathcal{R}(G, K, P)$ via alpha blending. This explicit parameterization allows StyleLift3D to modify appearance and geometry directly while retaining the input scene as a structural reference.

3.2 Problem Formulation

Let G_c denote the input 3DGS scene. We render it from V calibrated viewpoints and independently stylize the rendered images with a frozen 2D stylizer, obtaining style views $\tilde{I}_v$ together with content depth D_v, camera intrinsics K_v, and poses P_v. StyleLift3D learns the scene-general mapping:

$$G_s = \Phi_\theta\Big(G_c, \{(\tilde{I}_v, D_v, K_v, P_v)\}_{v=1}^{V}\Big), \tag{2}$$

where G_s is the stylized Gaussian scene and θ is shared across scenes and styles. The mapping updates G_c in a single forward pass rather than optimizing per scene or reconstructing geometry from inconsistent style views.

3.3 Geometry-Aligned Style Lifting

Independently stylized views are rich in appearance but lack cross-view consistency, making their 2D features ambiguous supervision for unordered Gaussian primitives. Feed-forward reconstruction methods such as pixelSplat (Charatan et al., 2024), MVSplat (Chen et al., 2024b), and ReSplat (Xu et al., 2026) demonstrate that image features can be lifted into a 3D Gaussian representation. Inspired by this paradigm, we use the input 3DGS as a fixed geometric anchor and lift only the style evidence, rather than reconstructing a new scene. We implement this idea with **Geometry-Aligned Style Lifting (GASL)**, which contains a 3D content branch and a 2D style branch.

3D content branch. A pointwise encoder maps each input Gaussian and its attributes to a content token while retaining the Gaussian center as its coordinate:

$$f_i^c = E_c(\mu_i, s_i, q_i, \alpha_i, c_i), \qquad \mathcal{F}^c = \{(f_i^c, \mu_i)\}_{i=1}^{N}. \tag{3}$$

Here, $i \in \{1, \ldots, N\}$ indexes the input Gaussians, whose parameters $(\mu_i, s_i, q_i, \alpha_i, c_i)$ are defined in Section 3.1. E_c is the pointwise encoder, $f_i^c \in \mathbb{R}^d$ is its d-dimensional content token, and $\mathcal{F}^c$ is the set of content-token–center pairs. We implement E_c as a three-layer MLP. Although point-transformer designs have been explored for Gaussian modeling (Chen et al., 2025), we do not adopt PTv3 (Wu et al., 2024) because its memory consumption is prohibitive: in our setting, it cannot scale to 3DGS scenes containing more than one million Gaussians. We therefore defer spatial interaction to the subsequent local fusion module.

2D style branch. We first use the frozen DINOv2 encoder E_{DINO} (Oquab et al., 2024) to extract a feature map $F_v^{\mathrm{DINO}} = E_{\mathrm{DINO}}(\tilde{I}_v)$ from each style view $\tilde{I}_v$. We then aggregate these feature maps globally across views and lift them into the input 3DGS coordinate frame:

$$\begin{aligned} \{\bar{F}_v^s\}_{v=1}^V &= \mathrm{GlobalSelfAttn}\big(\{F_v^{\mathrm{DINO}}\}_{v=1}^V\big)\,, \\ \mathcal{F}^s &= \mathrm{Unproject}\big(\{(\bar{F}_v^s, D_v)\}_{v=1}^V\big) = \{(f_j^s, x_j)\}_{j=1}^M. \end{aligned} \tag{4}$$

Here, $v \in \{1, \ldots, V\}$ indexes the style views. The operator $\mathrm{GlobalSelfAttn}$ denotes a trainable six-layer global self-attention module following VGGT (Wang et al., 2025): its layers alternate between within-view and cross-view attention and output the enhanced feature map $\bar{F}_v^s$ for each view. The operator $\mathrm{Unproject}$ renders D_v from the input 3DGS via alpha blending and unprojects the features into 3D style tokens using the known camera calibration. The resulting set contains M style tokens, where $f_j^s \in \mathbb{R}^d$ is the j-th token and $x_j \in \mathbb{R}^3$ is its 3D position. GASL thus outputs the spatially aligned token sets $\mathcal{F}^c$ and $\mathcal{F}^s$.

3.4 Octree-Local Style Fusion

The central challenge is to establish spatial correspondence between two irregular token sets without sacrificing scalability. Global interaction couples every Gaussian token with every style token, incurring $O(NM)$ cost and allowing style evidence to leak across unrelated regions. ReSplat avoids global interaction through kNN attention (Xu et al., 2026); however, retrieving and materializing neighbors for every query remains memory-intensive for million-scale Gaussian scenes. Moreover, a fixed neighborhood size is insensitive to spatial density: it may cross object boundaries in sparse regions while discarding relevant evidence in dense regions. We therefore introduce **Octree-Local Style Fusion (OLSF)**, which organizes both token sets into a shared, density-adaptive spatial hierarchy and restricts interaction to spatially co-located tokens.

Let Ω_0 be the axis-aligned bounding box enclosing all Gaussian centers $\{\mu_i\}_{i=1}^N$ and lifted style positions $\{x_j\}_{j=1}^M$. For a node region Ω_l, we count its content and style tokens as n_l^c and n_l^s, respectively, and recursively construct the octree (Riegler et al., 2017) as:

$$\begin{aligned} n_l^c &= \sum_{i=1}^{N} \mathbf{1}[\mu_i \in \Omega_l], \quad n_l^s = \sum_{j=1}^{M} \mathbf{1}[x_j \in \Omega_l], \\ \mathcal{O}(\Omega_l) &= \begin{cases} \{\Omega_l\}, & \max\left\{\dfrac{n_l^c}{\tau_c}, \dfrac{n_l^s}{\tau_s}\right\} \le 1, \\ \displaystyle\bigcup_{b\in\{0,1\}^3} \mathcal{O}(\Omega_{l,b}), & \text{otherwise.} \end{cases} \end{aligned} \tag{5}$$

where $\mathbf{1}[\cdot]$ is the indicator function, τ_c and τ_s are the leaf capacities, and $\{\Omega_{l,b}\}_{b\in\{0,1\}^3}$ are the eight child regions obtained by bisecting Ω_l along all three spatial axes. This recursion yields the leaf set $\mathcal{L}(\mathcal{O})$. For each $\Omega_l \in \mathcal{L}(\mathcal{O})$, we stack its content and style tokens row-wise into $F_l^c \in \mathbb{R}^{n_l^c \times d}$ and $F_l^s \in \mathbb{R}^{n_l^s \times d}$, respectively. Their local cross-attention is explicitly defined as:

$$\widetilde{F}_l^c = F_l^c + \mathrm{CrossAtt}(F_l^c, F_l^s)\,, \qquad \Omega_l \in \mathcal{L}(\mathcal{O}). \tag{6}$$

Here, $\mathrm{CrossAtt}(F_l^c, F_l^s)$ fuses the content tokens with the style tokens from the same leaf, and the resulting style signal is added residually to the content tokens. If a content leaf contains no style token, we use the nearest occupied style leaf. The rows of $\widetilde{F}_l^c$ are finally restored to the original Gaussian order, yielding $\{\tilde{f}_i^c\}_{i=1}^N$. Because each content token interacts with at most τ_s style tokens, OLSF reduces the interaction cost to $O(N\tau_s)$ while preserving geometry-aligned locality.

3.5 Structure-Preserving Gaussian Decoding

The fused tokens must be converted into expressive Gaussian parameters without compromising the geometry that anchors multi-view consistency. A fixed set of primitives may be insufficient for fine geometric or textural patterns, while unconstrained parameter prediction may distort the underlying scene. We therefore introduce **Structure-Preserving Gaussian Decoding (SPGD)**, comprising two complementary components: adaptive Gaussian splitting expands the representation only where

additional capacity is required, and bounded residual decoding translates the resulting tokens into constrained updates around the input Gaussians.

Adaptive Gaussian splitting. We first predict whether Gaussian i should be expanded, together with bounded offsets for its potential children:

$$p_i = \text{gumbel-sigmoid}\Big(H_{\text{split}}(\tilde{f}_i^c); \tau\Big), \qquad \Delta\mu_{i,k} = \rho \tanh H_{\text{off}}^k(\tilde{f}_i^c), \quad k = 1, \dots, K. \tag{7}$$

Here, H_{split} predicts the split logit, $p_i \in [0, 1]$ is the differentiable splitting gate, and τ is its temperature. The head H_{off}^k predicts the offset $\Delta\mu_{i,k} \in \mathbb{R}^3$ of child k, whose magnitude is bounded by ρ. When activated, the gate allocates K child Gaussians that inherit the fused token and attributes of their parent.

Bounded residual decoding. After determining the adaptive representation, a pointwise decoder D maps the fused content token and original center to a latent code, $h_i = D(\tilde{f}_i^c, \mu_i)$. Parameter-specific heads then produce the stylized Gaussian parameters

$$\begin{aligned} c_i^s &= c_i + H_c(h_i), & \mu_i^s &= \mu_i + \beta d \tanh H_\mu(h_i), \\ \log s_i^s &= \log s_i + \gamma \tanh H_s(h_i), & q_i^s &= \text{norm}(q_i + H_q(h_i)), \\ \alpha_i^s &= \text{clip}(\alpha_i + H_\alpha(h_i), 0, 1), \end{aligned} \tag{8}$$

where the superscript s denotes stylized parameters and H_c, H_μ, H_s, H_q, and H_α are the appearance, position, scale, rotation, and opacity heads, respectively. All prediction heads denoted by H, including the splitting and offset heads, are implemented as single-layer MLPs. The scalar d is the input-scene diagonal, while β and γ bound the position and log-scale residuals. The operators $\text{norm}(\cdot)$ and $\text{clip}(\cdot, 0, 1)$ enforce a unit rotation quaternion and valid opacity, respectively. A split child is centered at $\mu_{i,k}^s = \mu_i^s + \Delta\mu_{i,k}$ and inherits the remaining decoded parameters of its parent. All residual heads are zero-initialized, so SPGD begins as an identity mapping and learns only bounded deviations from the input 3DGS.

3.6 Training Objective

Following the established supervision used in Stylos and G-Style (Liu et al., 2026; Kovács et al., 2024), we render the stylized scene as $\hat{I}_v = \mathcal{R}(G_s, P_v)$ and supervise it with the corresponding stylized view $\tilde{I}_v$. The overall objective is:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \lambda_{\text{style}} \mathcal{L}_{\text{style}}. \tag{9}$$

Reconstruction and style losses. We instantiate the two standard objectives as:

$$\begin{aligned} \mathcal{L}_{\text{rec}} &= \frac{1}{V} \sum_{v=1}^{V} \Big(\lambda_1 \|\hat{I}_v - \tilde{I}_v\|_1 + \lambda_2 \|\hat{I}_v - \tilde{I}_v\|_2^2 \Big), \\ \mathcal{L}_{\text{style}} &= \lambda_{\text{nnfm}} \mathcal{L}_{\text{NNFM}} + \lambda_{\text{content}} \mathcal{L}_{\text{VGG}} + \lambda_{\text{tv}} \mathcal{L}_{\text{TV}}. \end{aligned} \tag{10}$$

Here, λ_1 and λ_2 weight the pixel-wise $\mathcal{L}_1$ and $\mathcal{L}_2$ terms. We use the NNFM loss (Zhang et al., 2022; Kovács et al., 2024), VGG content loss (Johnson et al., 2016; Liu et al., 2026; Kovács et al., 2024), and total-variation loss (Liu et al., 2026; Kovács et al., 2024), weighted by λ_{nnfm}, λ_{content}, and λ_{tv}, respectively. λ_{style} controls the overall style contribution.

4 Experiments

Datasets. For training, we construct a 9K-scale dataset from 100 DL3DV scenes (Ling et al., 2024), each stylized by three stylizers using 30 distinct WikiArt style images (Saleh & Elgammal, 2016). For testing, We conduct both quantitative and qualitative evaluations on 15 held-out scenes drawn from Tanks and Temples (TnT) (Knapitsch et al., 2017), Mip-NeRF360 (Barron et al., 2022), and DL3DV-10K. In addition to evaluating on the 30 training style images, we use 20 additional WikiArt style images for unseen-style evaluation, with no overlap with the training styles. Each scene is represented by an existing 3DGS reconstruction (Kerbl et al., 2023) and rendered multi-view observations. Further details on the dataset are provided in Appendix A.5.

Table 1: Content fidelity on the Tanks and Temples datasets, evaluated using LPIPS, grayscale LPIPS (Gray), and PSNR. **Bold** and underline indicate the best and second-best results, respectively. Each stylization method category is visualized with a distinct color.

Method	Train			Truck			M60		
	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑
StyleGaussian	0.570	0.501	10.786	0.577	0.534	**11.826**	0.595	0.546	10.380
Stylos	0.775	0.657	9.542	0.797	0.655	9.901	0.768	0.665	9.978
AnyStyle	0.716	0.625	9.496	0.722	0.610	9.869	0.693	0.618	9.801
StyleLift3D	**0.552**	**0.409**	**11.239**	**0.553**	**0.385**	11.491	**0.508**	**0.400**	**11.336**
	Ballroom			**Museum**			**Palace**		
	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑
StyleGaussian	**0.517**	**0.455**	8.884	0.578	0.533	**11.051**	0.571	**0.399**	9.755
Stylos	0.826	0.755	7.286	0.786	0.703	8.391	0.769	0.654	10.613
AnyStyle	0.735	0.689	7.483	0.705	0.655	8.370	0.735	0.651	9.545
StyleLift3D	0.596	0.552	**9.884**	**0.520**	**0.445**	8.713	**0.556**	0.491	**12.023**

Table 2: User study results. **Bold** and underline indicate the best and second-best results, respectively. Each stylization method category is visualized with a distinct color.

Method	Style Consistency↑	Content Preservation↑	Visual Naturalness↑	Sum↑
StyleGaussian	3.264	3.496	2.960	9.720
Stylos	3.704	2.728	2.936	9.368
AnyStyle	**3.784**	3.632	3.472	10.888
StyleLift3D	3.592	**4.520**	**4.288**	**12.400**

Baselines. We compare against optimization-based StyleGaussian (Liu et al., 2024) and feed-forward Stylos (Liu et al., 2026) and AnyStyle (Kaleta et al., 2026). StyleLift3D directly stylizes existing 3DGS scenes using frozen 2D stylizers, with AttenST (Huang et al., 2025), StyleSSP (Xu et al., 2025a), and StyleGallery (He et al., 2026) as guidance generators. All reported StyleLift3D metrics are averaged across these three stylizers; per-stylizer results are provided in Appendix A.2 and A.3. Further implementation and training details are given in Appendix A.5.

4.1 Quantitative Comparison

We evaluate methods along three axes: content fidelity, view consistency, and human preference.

Content fidelity. Following Liu et al. (2024), we compare each stylized rendering with its corresponding original frame using LPIPS (Zhang et al., 2018), grayscale LPIPS (Gray), and PSNR. As shown in Table 1, StyleLift3D achieves the best or second-best results across all evaluated scenes and metrics. These results support StyleLift3D's ability to preserve the original scene structure while integrating style cues from view-inconsistent 2D stylizations.

View consistency. Following prior work (Liu et al., 2026), we test warp-based LPIPS and RMSE over short-range and long-range view pairs as a measure of multi-view consistency. However, after careful consideration and extensive experiments, we find that these metrics do not serve as good proxies for evaluating view consistency in the stylized domain. Detailed numerical results and discussion are provided in Appendix A.6.

Human preference. We conduct a user study with 25 participants, who rate randomly ordered results on style consistency, content preservation, and visual naturalness using a 5-point scale (Fang et al., 2025). Table 2 reports the mean opinion scores. StyleLift3D achieves the highest scores for content preservation and visual naturalness, as well as the highest summed score. Although its style consistency score is slightly lower than those of Stylos and AnyStyle, these results suggest a favorable balance between style consistency and the preservation of coherent, natural-looking scene content. Further study details are provided in Appendix A.4.

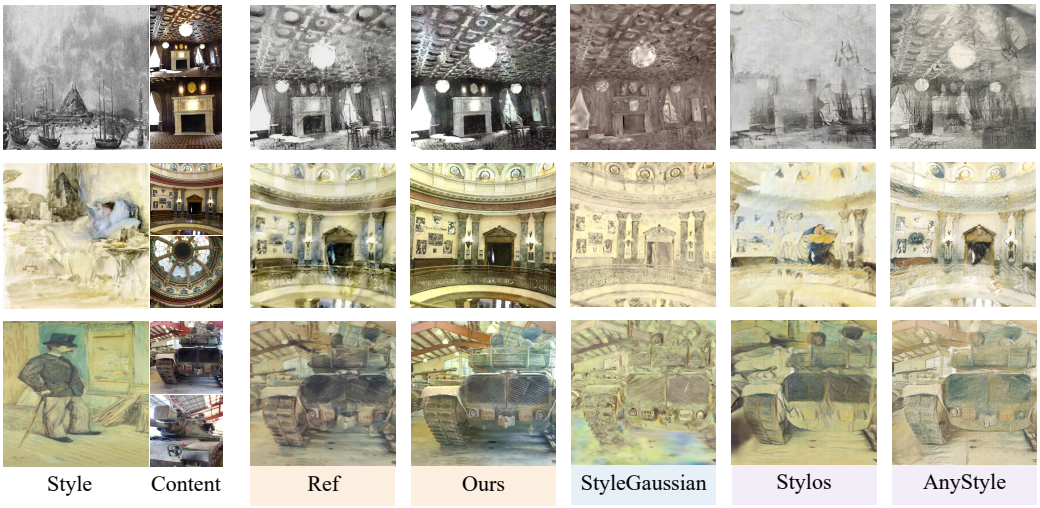


Figure 3: Visual comparison between our method and existing 3D stylization methods.

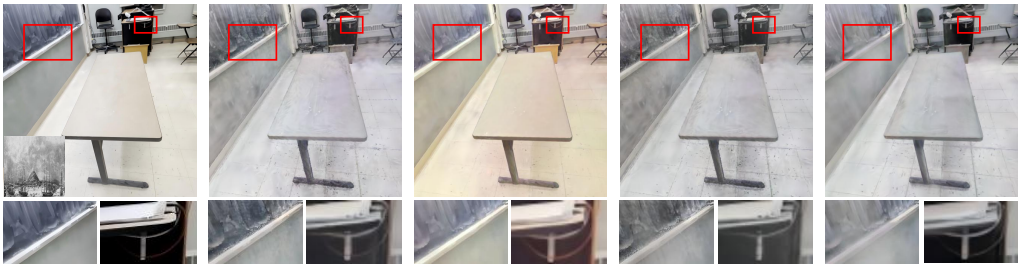


Figure 4: Qualitative ablations. From left to right: style/content references, full model, kNN attention, w/o splitter, and w/o geo residual. Under the same training-time budget, kNN (3 epochs) yields weak stylization. Removing the splitter degrades structure and fine details, while removing geometry residuals blurs edges and reduces surface detail.

4.2 Qualitative Comparison

Figure 3 presents qualitative comparisons across three scenes between StyleGaussian (Liu et al., 2024), Stylos (Liu et al., 2026), AnyStyle (Kaleta et al., 2026) and our method. Here, Ref denotes the 2D style reference generated by AttenST (Huang et al., 2025) to guide our method.

The results reflect the methods' different design choices. StyleGaussian modifies the appearance of existing Gaussians while keeping their geometry fixed. Stylos and AnyStyle reconstruct stylized scenes from stylized views, capturing the target appearance but occasionally introducing blur or geometric drift. In contrast, StyleLift3D directly stylizes an existing 3DGS scene by fusing style and content features in a shared 3D space. Its bounded residual updates (Section 3.5) help preserve sharp scene boundaries while accommodating geometric stylization and brushstroke effects. By anchoring stylization to the input geometry and integrating multi-view guidance into a unified 3D representation, StyleLift3D combines expressive stylization with structural preservation and supports coherent rendering across viewpoints. Additional qualitative results of our method are provided in Section A.3 and Figure 8 in the appendix.

4.3 Ablation Study

We ablate the three components and one hyperparameter of StyleLift3D on 50 scenes with 45 style cases each. Results are shown in Figure 4, Figure 5 and Table 3.

Table 3: Ablation study. We compare the full model with variants that replace octree-local style fusion with kNN fusion, disable dynamic Gaussian splitting, or remove residual geometry updates.

Variant	LPIPS↓	Gray↓	PSNR↑
Full model	0.228	0.148	19.010
kNN fusion	0.176	0.099	19.742
w/o dynamic splitter	0.242	0.178	18.792
w/o geometry residual	0.222	0.163	18.898

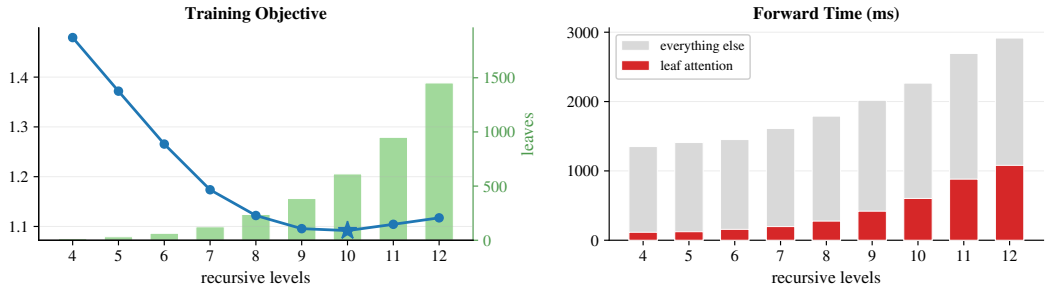


Figure 5: Leaf granularity of the octree partition.

Octree-local style fusion. We ablate it by replacing the fusion rule and varying octree granularity. (1) *kNN fusion.* Under the same training-time budget, exact kNN fusion completes only 3 epochs vs. 100 for the full model and produces substantially weaker stylization. Its outputs remain close to the original scene, so favorable content-fidelity scores primarily reflect undertraining rather than effective style transfer. These results highlight the computational advantage of octree-local fusion. (2) *Leaf granularity.* The maximum octree depth controls the spatial extent of local style fusion. Increasing the depth cap from 4 to 10 lowers the objective from 1.4794 to 1.0917, before it rises to 1.1043 and 1.1173 at depths 11 and 12, respectively. Although a depth of 10 yields the lowest objective, it requires approximately $1.3\times$ the forward-pass time of depth 8. We therefore adopt 8 as the default to balance quality and efficiency.

Dynamic splitter. Disabling adaptive splitting fixes the primitive count and limits local representational capacity. With geometry residuals still active, displaced Gaussians can leave gaps, degrading thin structures and fine stroke details (Figure 4, w/o splitter). All metrics worsen relative to the full model , highlighting the role of adaptive splitting in preserving structure during stylization.

Geometry residuals. Removing bounded position and scale residuals restricts geometric adaptation, blurring edges and reducing surface detail, particularly in the flat region of the left example (Figure 4, w/o geo residual). Although LPIPS improves slightly, Gray and PSNR worsen. Together with the visual results, this suggests that geometry residuals primarily benefit local structural fidelity.

5 Conclusion

We presented StyleLift3D, a plug-and-play feed-forward framework that directly stylizes existing 3DGS scenes. Geometry-aligned style lifting, octree-local fusion, and structure-preserving decoding integrate inconsistent multi-view style cues into a coherent 3D representation in a single forward pass, without per-scene optimization or reconstruction from stylized views. Experiments show strong style fidelity, multi-view consistency, efficiency, and generalization, with human preference favoring its balance between content preservation and naturalness. Ablations support our design choices and reveal a quality–efficiency trade-off controlled by octree depth. Future work could extend StyleLift3D to dynamic scenes and broader style domains.

AI USE STATEMENT

We used generative AI tools in three ways, as disclosed in the submission system. First, we used large language models to aid and polish writing, including drafting, editing, and improving the clarity, grammar, flow, and internal consistency of portions of the manuscript. All AI-assisted text was reviewed and revised by the authors. Second, we used large language models for retrieval and discovery, including identifying and locating related literature and background references. The authors independently verified the relevance, accuracy, and bibliographic details of all references. Third, we used generative AI tools to generate synthetic datasets. The generated data were subsequently inspected, filtered, and validated by the authors before being used in the experiments.

In all cases, AI tools were used under the authors' direction, and all AI-assisted outputs were reviewed and edited by the authors. The research idea, scientific narrative, core method, experimental design, and interpretation of results originated with the authors. All reported numerical results come from our own experiments and analyses and were not generated by AI. We did not use generative AI for any purpose beyond the three uses described above. The authors take full responsibility for the manuscript, including its text, references, synthetic data, and results.

ETHICS STATEMENT

We use publicly available datasets (DL3DV, WikiArt, TnT and Mip-NeRF360) and publicly released pretrained models, each under its own license, and we evaluate on existing 3DGS reconstructions rather than on data we captured ourselves. The user study involved 25 participants, who were informed of its purpose and consented to take part; responses were collected anonymously and no personally identifiable information was recorded. We acknowledge that transferring an artistic style onto a captured scene raises copyright and authenticity questions, and we encourage responsible use of our method.

REPRODUCIBILITY STATEMENT

Section 3 specifies the architecture and the training objective, and Section 4 reports the comparisons, the user study and the ablations. Section A.5 documents the training corpus — the 100 reconstructed DL3DV scenes, their 30 style references and the resulting $100 \times 30 \times 3$ stylized view sets — together with the optimizer, the loss weights, the batch composition and the schedule, and the distribution of scene complexity across the corpus. Section A.4 gives the user-study design, questionnaire and analysis; Section A.6 defines the multi-view consistency metric and the controlled counterexample used to characterise it; and Sections A.3 and A.7 give the extended qualitative results and the failure case. Inference is a single forward pass, so reproducing a result requires no per-scene optimization. Experiments were conducted with fixed random seeds, and we will release the complete code and the trained checkpoint.

REFERENCES

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*, pp. 5460–5469. IEEE, 2022.

David Charatan, Sizhe Lester Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19457–19467. IEEE, 2024.

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*, pp. 21476–21485. IEEE, 2024a.

Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *European conference on computer vision*, pp. 370–386. Springer, 2024b.

Yutong Chen, Marko Mihajlovic, Xiyi Chen, Yiming Wang, Sergey Prokudin, and Siyu Tang. Splatformer: Point transformer for robust 3d gaussian splatting. In *International Conference on Learning Representations*, volume 2025, pp. 56049–56075, 2025.

Jiwoo Chung, Sangeek Hyun, and Jae-Pil Heo. Style injection in diffusion: A training-free approach for adapting large-scale diffusion models for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8795–8805. IEEE, 2024.

Zhiyuan Fang, Rengan Xie, Xuancheng Jin, Qi Ye, Wei Chen, Wenting Zheng, Rui Wang, and Yuchi Huo. A3GS: Arbitrary artistic style into arbitrary 3d gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 17751–17760. IEEE, 2025.

Ben Fei, Jingyi Xu, Rui Zhang, Qingyuan Zhou, Weidong Yang, and Ying He. 3d gaussian splatting as a new era: A survey. *IEEE Transactions on Visualization and Computer Graphics*, 31(8): 4429–4449, 2024.

Bruno Galerne, Jianling Wang, Lara Raad, and Jean-Michel Morel. Sgsst: Scaling gaussian splatting style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26535–26544. IEEE, 2025.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

Boyu He, Yunfan Ye, Chang Liu, Weishang Wu, Fang Liu, and Zhiping Cai. Stylegallery: Training-free and semantic-aware personalized style transfer from arbitrary image references. *arXiv preprint arXiv:2603.10354*, 2026.

Bo Huang, Wenlun Xu, Qizhuo Han, Haodong Jing, and Ying Li. Attenst: A training-free attention-driven style transfer framework with pre-trained diffusion models. *arXiv preprint arXiv:2503.07307*, 2025.

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1510–1519. IEEE, 2017.

Lihan Jiang, Yucheng Mao, Linning Xu, Tao Lu, Kerui Ren, Yichen Jin, Xudong Xu, Mulin Yu, Jiangmiao Pang, Feng Zhao, et al. Anysplat: Feed-forward 3d gaussian splatting from unconstrained views. *ACM Transactions on Graphics*, 44(6):1–16, 2025.

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pp. 694–711. Springer, 2016.

Joanna Kaleta, Bartosz Świrta, Kacper Kania, Tomasz Trzciński, Przemysław Spurek, and Marek Kowalski. AnyStyle: Single-pass multimodal stylization for 3D gaussian splatting. In *Proceedings of the 34th ACM International Conference on Multimedia*, 2026.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4):139:1–139:14, 2023.

GeonU Kim, Kim Youwang, Lee Hyoseok, and Tae-Hyun Oh. Fpgs: Feed-forward semantic-aware photorealistic style transfer of large-scale gaussian splatting. *International Journal of Computer Vision*, 134(5):214, 2026.

Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4):1–13, 2017.

Dmytro Kotovenko, Olga Grebenkova, Nikolaos Sarafianos, Avinash Paliwal, Pingchuan Ma, Omid Poursaeed, Sreyas Mohan, Yuchen Fan, Yilei Li, Rakesh Ranjan, et al. Wast-3d: Wasserstein-2 distance for scene-to-scene stylization on 3d gaussians. In *European Conference on Computer Vision*, pp. 298–314. Springer, 2024.

Áron Samuel Kovács, Pedro Hermosilla, and Renata G Raidou. G-Style: Stylized gaussian splatting. *Computer Graphics Forum*, 43(7):e15259, 2024.

Wenyu Li, Sidun Liu, Tongrui Hu, Peng Qiao, and Yong Dou. Locusgs: Spatially grounded tokens for feed-forward 3d gaussian splatting. *arXiv preprint arXiv:2608.12825*, 2026.

Yangkai Lin, Jiabao Lei, and Kui Jia. Multi-StyleGS: Stylized gaussian splatting with multiple styles. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(5):5289–5297, 2025.

Lu Ling, Yichen Sheng, Zhi Tu, Wentian Zhao, Cheng Xin, Kun Wan, Lantao Yu, Qianyu Guo, Zixun Yu, Yawen Lu, et al. Dl3dv-10k: A large-scale scene dataset for deep learning-based 3d vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22160–22169. IEEE, 2024.

Hanzhou Liu, Jia Huang, Mi Lu, Srikanth Saripalli, and Peng Jiang. Stylos: Multi-view 3d stylization with single-forward gaussian splatting. In *International Conference on Learning Representations*, volume 2026, pp. 70768–70801, 2026.

Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. Stylegaussian: Instant 3d style transfer with gaussian splatting. In *SIGGRAPH Asia 2024 Technical Communications*, pp. 21:1–21:4, 2024.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024.

Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3577–3586, 2017.

Babak Saleh and Ahmed Elgammal. Large-scale classification of fine-art paintings: Learning the right metric on the right feature. *International Journal for Digital Art History*, 2016.

Giuseppe Stracquadanio, Kevin Raj, Julia Grabinski, and Stefan Roth. Reconsplat: Generalizable 3d scene reconstruction beyond observed views. In *European Conference on Computer Vision*, pp. 304–325. Springer, 2026.

Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European conference on computer vision*, pp. 402–419. Springer, 2020.

Can Wang, Lei Liu, Wei Jiang, and Dong Xu. Z-order transformer for feed-forward gaussian splatting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7362–7371, 2026.

Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5294–5306. IEEE, 2025.

Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4840–4851. IEEE, 2024.

Haofei Xu, Daniel Barath, Andreas Geiger, and Marc Pollefeys. Resplat: Learning recurrent gaussian splatting. In *European Conference on Computer Vision*, pp. 364–382. Springer, 2026.

Ruojun Xu, Weijie Xi, XiaoDi Wang, Yongbo Mao, and Zach Cheng. Stylessp: Sampling startpoint enhancement for training-free diffusion-based method for style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18260–18269. IEEE, 2025a.

Yuancheng Xu, Wenqi Xian, Li Ma, Julien Philip, Ahmet Taşel, Yiwei Zhao, Ryan Burgert, Mingming He, Oliver Hermann, Oliver Pilarski, et al. Virtually being: Customizing camera-controllable video diffusion models with volumetric performance captures. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pp. 1–12, 2025b.

Dingxi Zhang, Yu-Jie Yuan, Zhuoxun Chen, Fang-Lue Zhang, Zhenliang He, Shiguang Shan, and Lin Gao. Stylizedgs: Controllable stylization for 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Kai Zhang, Nick Kolkin, Sai Bi, Fujun Luan, Zexiang Xu, Eli Shechtman, and Noah Snavely. Arf: Artistic radiance fields. In *European conference on computer vision*, pp. 717–733. Springer, 2022.

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*, pp. 3813–3824. IEEE, 2023.

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*, pp. 586–595. IEEE, 2018.

A Appendix

A.1 Appendix overview

This Appendix is organized as follows:

- **Extended quantitative results** (Section A.2): detailed quantitative results of our methods on three Plug-and-Play 2D stylizers.
- **Extended qualitative results** (Section A.3): (1) comparison between our methods and naive optimization method, (2) one scene and camera held fixed while the 2D stylizer, the style reference and the pipeline stage vary, to demonstrate the plug-and-play generalization of StyleLift3D.
- **User study** (Section A.4): the study design, the questionnaire, and the statistical analysis of human preferences.
- **Training details** (Section A.5): implementation specifics, architecture choices, loss weights, optimization setup, dataset construction, and reproducibility information.
- **Multi-view consistency** (Section A.6): in-depth analysis of the warping-based consistency metric, its limitations in the stylized domain, and additional supporting evidence.
- **Limitations and failure cases** (Section A.7): a representative failure case for image-guided stylization, with a discussion of its cause and of the mitigation it suggests.

A.2 Extended quantitative results

Table 4 reports quantitative results for StyleLift3D on six scenes using guidance from AttenST (Huang et al., 2025), StyleSSP (Xu et al., 2025a), and StyleGallery (He et al., 2026). Unlike the metrics reported in the main text, which are averaged across these three stylizers, this table presents results separately for each scene–stylizer pair. This breakdown complements the qualitative comparisons and provides a detailed assessment of StyleLift3D's effectiveness across scenes and its plug-and-play compatibility with different 2D guidance sources.

Table 4: Content fidelity on the Tanks and Temples datasets. LPIPS, grayscale LPIPS (Gray), and PSNR are reported separately for each scene–stylizer pair, rather than averaged across stylizers as in the main text. **Bold** and underline indicate the best and second-best results, respectively. Each stylization method category is visualized with a distinct color.

Method	Train			Truck			M60		
	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑
StyleGaussian	0.570	0.501	10.786	0.577	0.534	**11.826**	0.595	0.546	10.380
Stylos	0.775	0.657	9.542	0.797	0.655	9.901	0.768	0.665	9.978
AnyStyle	0.716	0.625	9.496	0.722	0.610	9.869	0.693	0.618	9.801
StyleLift3D+AttenST	**0.537**	0.401	11.197	0.546	0.384	11.480	**0.495**	0.389	11.461
StyleLift3D+StyleSSP	0.574	0.431	11.028	0.571	0.395	11.339	0.533	0.425	11.028
StyleLift3D+StyleGallery	0.546	**0.395**	**11.491**	**0.542**	**0.377**	11.655	0.497	**0.387**	**11.518**
	Ballroom			**Museum**			**Palace**		
	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑	LPIPS↓	Gray↓	PSNR↑
StyleGaussian	**0.517**	**0.455**	8.884	0.578	0.533	**11.051**	0.571	**0.399**	9.755
Stylos	0.826	0.755	7.286	0.786	0.703	8.391	0.769	0.654	10.613
AnyStyle	0.735	0.689	7.483	0.705	0.655	8.370	0.735	0.651	9.545
StyleLift3D+AttenST	0.584	0.549	9.696	**0.517**	0.447	8.435	**0.545**	0.485	11.879
StyleLift3D+StyleSSP	0.612	0.556	**10.028**	0.524	**0.444**	8.828	0.574	0.506	11.869
StyleLift3D+StyleGallery	0.591	0.552	9.927	0.518	0.445	8.877	0.548	0.482	**12.322**

A.3 Extended qualitative results

Comparison with direct optimization. A straightforward alternative to StyleLift3D is to fit a 3DGS directly to independently stylized views, as discussed in Sec. 1. We instantiate this baseline

on Horse, using the same ten stylized views and style references as our method. We optimize all Gaussian parameters—means, scales, rotations, opacities, and SH coefficients—for $2{,}000$ iterations, allowing geometry to change without the bounded residual parameterization used by our decoder.

Figures 6 and 7 reveal two distinct failure modes. For the high-saturation reference, optimization is unstable: the loss evaluated on the supervision views rises from 0.030 to 0.069 for AttenST, 0.035 to 0.090 for StyleGallery, and 0.073 to 0.445 for StyleSSP. In the StyleSSP case, it jumps from 0.134 to 0.473 within a single 100-iteration interval. The corresponding renders exhibit blotches, smearing, and severe noise artifacts, with degradation already visible at the fitted viewpoints. For the low-saturation reference, all three fits converge to losses between 2.2×10^{-3} and 3.8×10^{-3}, roughly an order of magnitude below their values at iteration 100. Nevertheless, novel-view renders still exhibit haze, ghosting and duplication of the horse, and color speckle in regions that should be neutral gray.

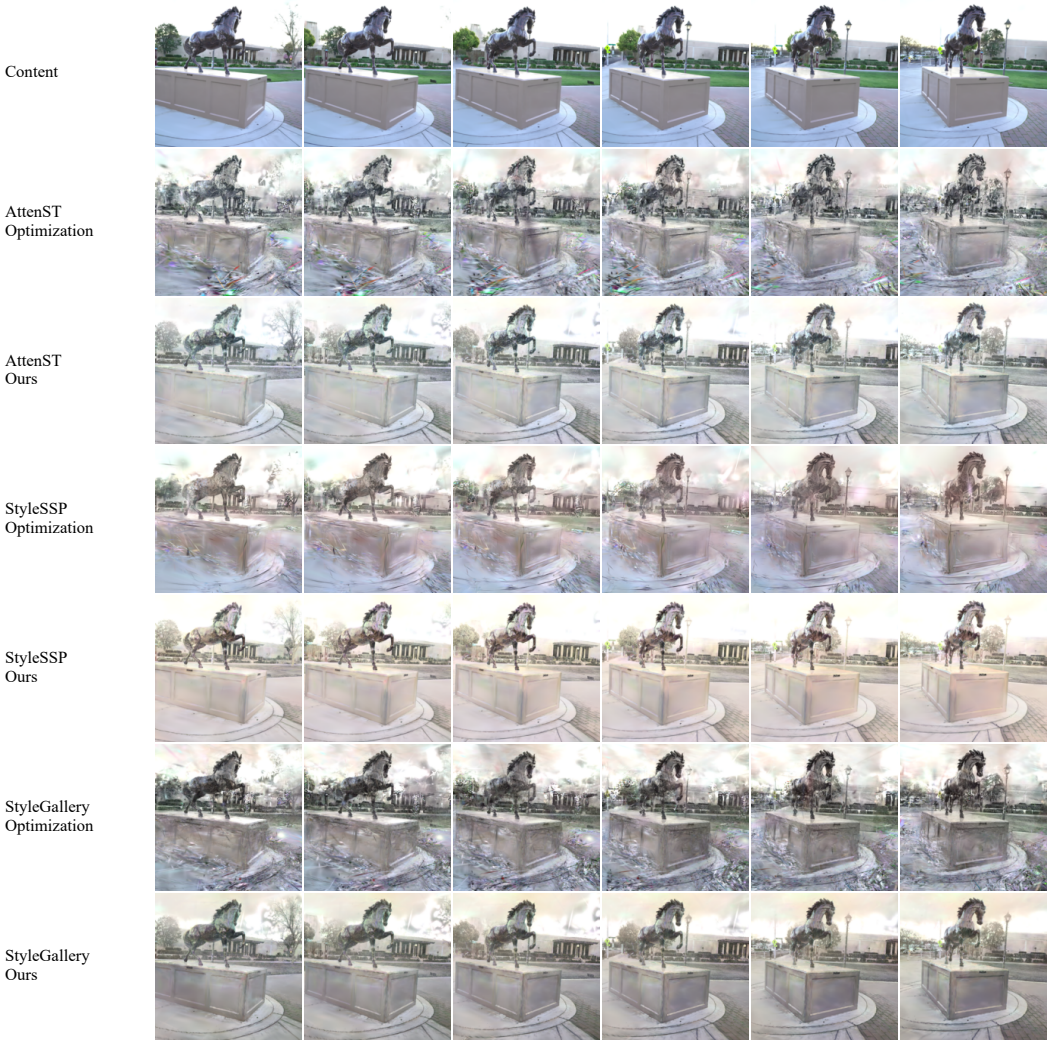


Figure 6: Comparison between per-scene optimization and StyleLift3D on Horse. Rows: the unstylized content, then the optimization and our inference for each of three 2D stylizers. Columns are six consecutive novel-view cameras. Optimization converges yet renders hazy and ghosted; ours stays clean and stable across the views.

The low-saturation case is particularly informative: minimizing the fitting loss does not ensure consistent novel-view appearance. Independently stylized views can provide conflicting appearance targets, and unrestricted geometry updates may accommodate these discrepancies at the expense of scene structure. The observed ghosting and duplication are consistent with this explanation, al-

though fitting losses alone cannot establish the underlying cause. Similar failure patterns across all three stylizers also suggest that the issue is not specific to a single guidance method.

In contrast, StyleLift3D fuses the guidance in 3D and bounds geometry updates. Its renders remain visually stable across the novel views for both references and all three stylizers, including the StyleSSP setting in which direct optimization becomes unstable. The closely spaced viewpoints expose visible popping in the optimization baseline, whereas StyleLift3D maintains more consistent appearance along the evaluated trajectory. A residual warm cast remains for the low-saturation reference, rather than fully matching its neutral gray palette. Nevertheless, these results are obtained in a single forward pass of the lifting network, without per-scene fitting or tuning.

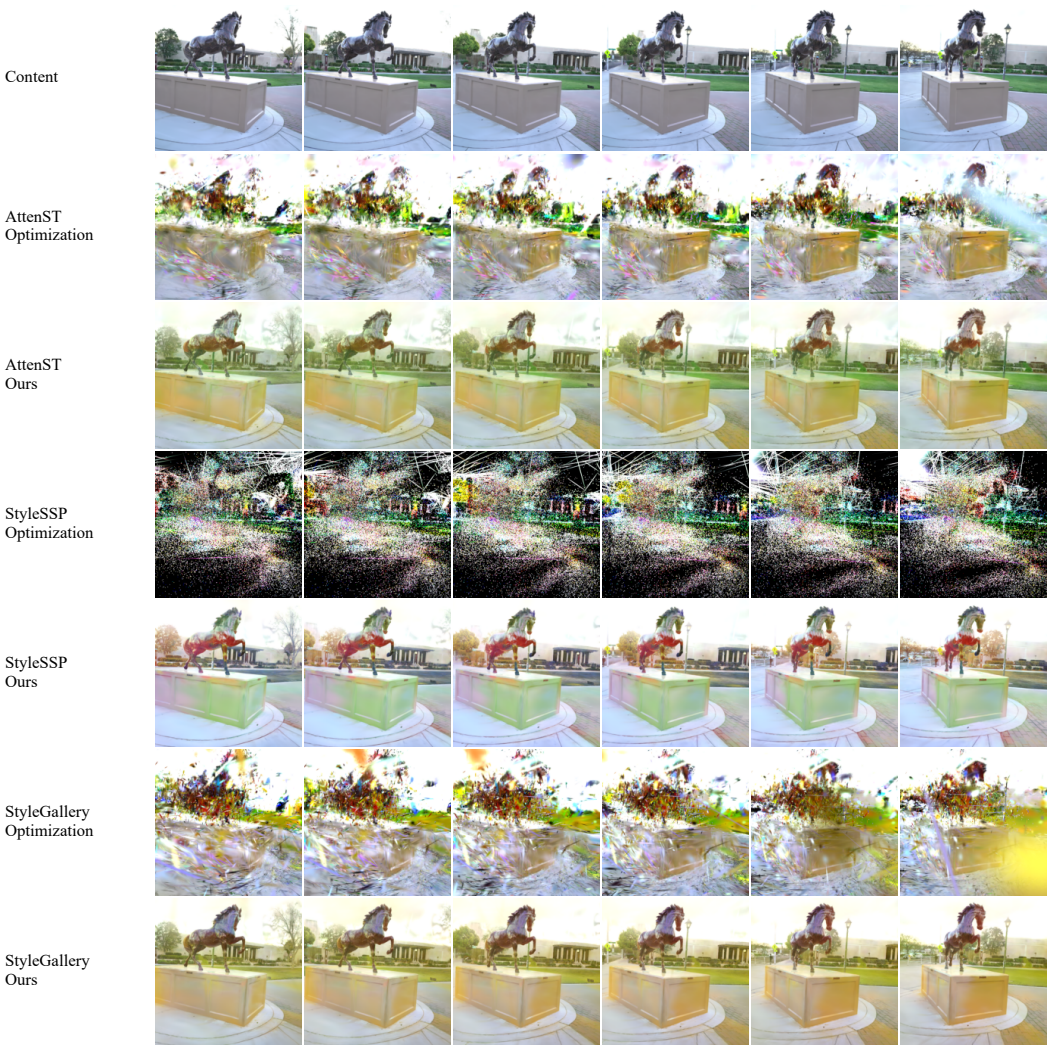


Figure 7: Same layout as Fig. 6, for the high-saturation style. Optimization no longer converges and degenerates into blotches, smears and noise; ours remains clean across all views for all stylizers.

Compatibility with different 2D stylizers. StyleLift3D accepts stylized multi-view images through a shared interface, allowing the 2D guidance generator to be replaced without modifying or retraining the lifting network. Figure 8 isolates this factor on a sofa scene by fixing the content scene, camera view, and three style references across AttenST, StyleSSP, and StyleGallery. The references are selected to illustrate both pronounced stylization and clearly differentiated appearances across stylizers.

The three methods produce distinct visual characteristics in these examples. AttenST introduces soft, woven textures and extends the reference palette beyond the sofa into the surrounding room: Style 5 shifts the white walls toward lilac, whereas Style 20 produces a much lighter appearance. StyleSSP induces stronger palette shifts while largely retaining the original shading, producing smoother, less granular textures. Its sofa becomes ochre under Style 20 and lilac-gray under Style 5. StyleGallery instead introduces finer painterly patterns, generally with lower saturation. Under Style 27, it largely retains the red sofa while cooling the surrounding walls; under Style 5, the walls shift toward green-gray. With the other inputs held fixed, these differences illustrate the range of appearance guidance available through the choice of 2D stylizer. The corresponding StyleLift3D renders retain these distinct characteristics.

As a preliminary probe of out-of-distribution compatibility with stylizers beyond those used to generate our training data, we additionally evaluate StyleID (Chung et al., 2024), which performs style transfer through reference attention injection in a pretrained diffusion model. We disable text conditioning so that appearance guidance is derived from the style-reference image, and directly feed its outputs into the unchanged lifting network. As shown in the rightmost column of Figure 8, StyleLift3D transfers the appearance of this unseen stylizer into 3D without retraining or stylizer-specific adaptation. Some of its more pronounced local changes are attenuated, as examined below.

Effects of lifting on structure and appearance. Figure 8 pairs the stylized guidance with the corresponding StyleLift3D renders, enabling direct comparison before and after lifting. We quantify these changes over all sixteen views and the three style references shown in the figure.

The most consistent improvement is in agreement with the original content structure. Averaged over all four stylizers, the gradient-magnitude correlation increases from 0.734 for the guidance to 0.875 for the renders. This increase holds for every stylizer: from 0.778 to 0.855 for AttenST, 0.714 to 0.855 for StyleSSP, 0.817 to 0.911 for StyleGallery, and 0.628 to 0.877 for StyleID. These results indicate stronger preservation of local image structure after lifting. Visually, mottling and splatter in regions such as the floor are reduced, while the underlying scene layout remains clear. This behavior is consistent with aggregating independently stylized views in a shared 3D representation while constraining geometry changes.

To assess the magnitude of appearance changes, we measure the mean absolute pixel deviation from the original content for both guidance and renders. The guidance deviations are 32.2, 38.8, and 25.8 on a 0–255 scale for AttenST, StyleSSP, and StyleGallery, respectively. The corresponding render-to-guidance ratios of these deviations are 103%, 100%, and 102%, indicating that lifting largely maintains the magnitude of their appearance changes. Although these ratios do not measure exact style fidelity, visual comparisons show that the reference-dependent palettes remain recognizable, including the plum tones of Style 5, the ochre palette of Style 20, and the red tones of Style 27.

StyleID exhibits stronger attenuation. Its guidance deviation is 35.1, within the range of the three training stylizers, but the render deviation decreases to 28.9, yielding a ratio of 82%. The resulting appearance-change magnitude remains within the 26.3–38.8 range observed for their renders, and the dominant palette changes remain visible. Thus, StyleLift3D can process StyleID guidance directly, but does not preserve its input changes to the same extent as those of the training stylizers.

Fine-scale image variation follows a similar pattern. The Laplacian standard deviation changes modestly for the three training stylizers: from 0.075 to 0.088 for AttenST, 0.027 to 0.032 for StyleSSP, and 0.100 to 0.094 for StyleGallery. For StyleID, it decreases more substantially, from 0.245 to 0.128. Together with its lower initial structural correlation, these measurements indicate that StyleID introduces stronger local image changes, whose high-frequency components are more heavily attenuated during lifting. This difference may partly reflect a mismatch with the training guidance. Broadening the training distribution to include stylizers with stronger local structural and texture changes could therefore help improve transfer fidelity for such inputs.

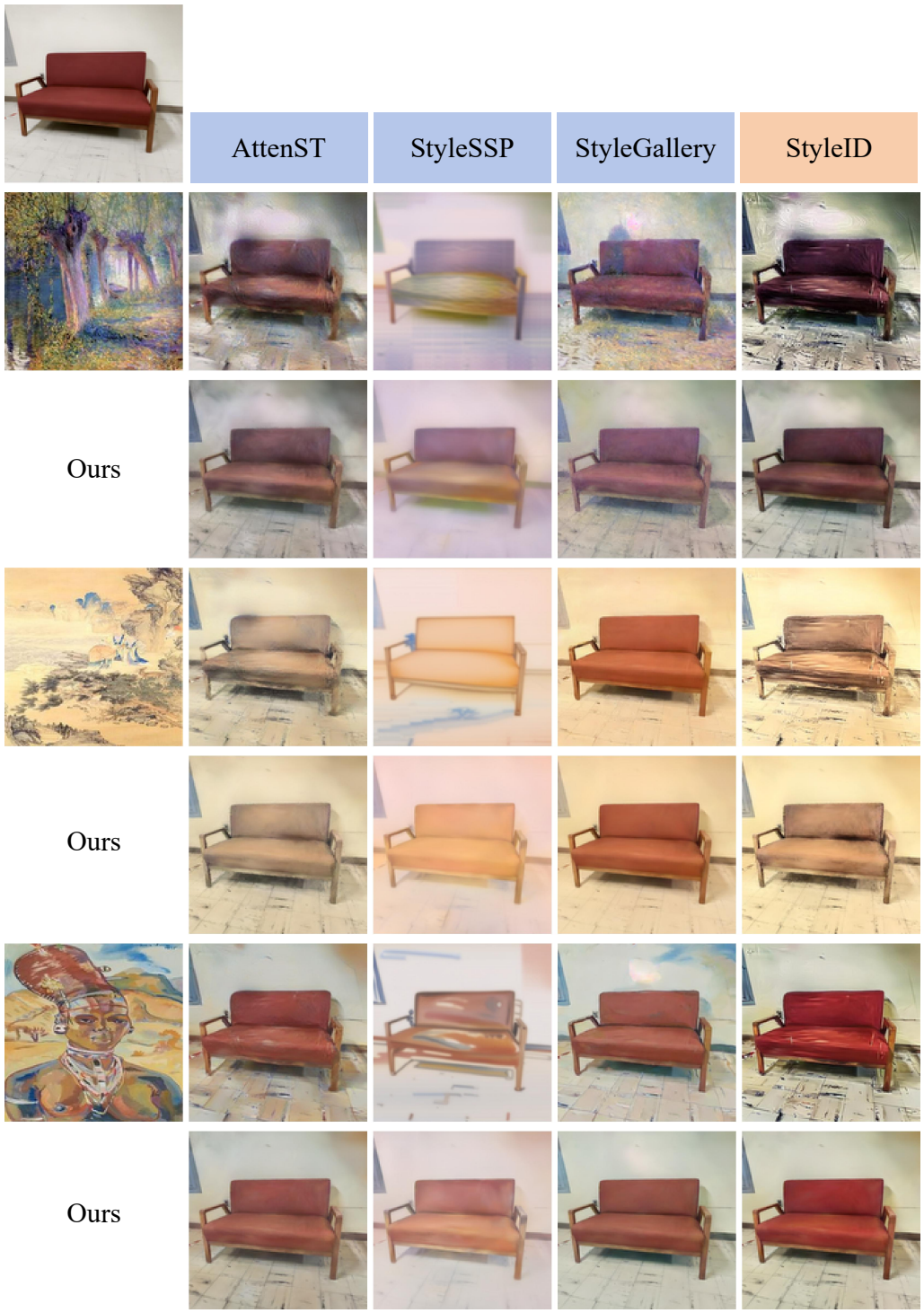


Figure 8: Qualitative results with different 2D stylizers. The four columns correspond to AttenST, StyleSSP, StyleGallery, and StyleID, while the three style references shown in the left column are Style 5, Style 20, and Style 27 from top to bottom. For each style, the upper row shows the stylized input and the lower row shows the corresponding StyleLift3D render. All results use the same content scene, camera viewpoint, and StyleLift3D checkpoint. StyleLift3D preserves the distinct appearance induced by each stylizer, including the unseen diffusion-based StyleID.

A.4 User study

To evaluate the perceptual quality of different 3D stylization methods, we conducted a user study with 25 participants. The study was designed to assess three key aspects: style consistency, content preservation, and visual naturalness.

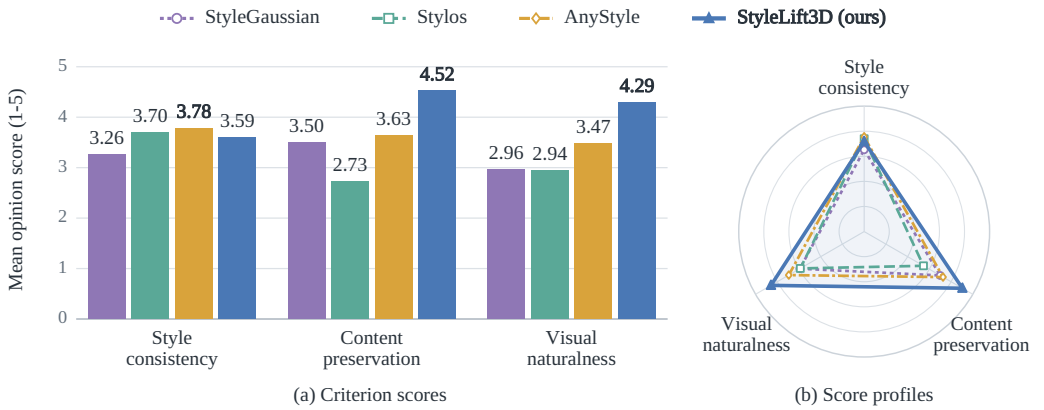
Study design. Participants were shown a sequence of stylization results on an online platform. In each question, we provided: (1) a style reference image, (2) the original multi-view renders of the content scene, and (3) the stylized multi-view results generated by an anonymous method. To avoid bias, the method names were concealed, and the order of scenes, styles, and methods was randomized for each participant. The scene and style assigned to each method were selected at random. All participants were instructed to answer based on their first impression and were informed that their responses would be anonymously stored for academic research purposes only.

Questionnaire. Each participant evaluated 20 groups of images. For each group, they were asked to rate the anonymized result on a 5-point Likert scale (1 = worst, 5 = best) across three dimensions:

- **Style Consistency**: "To what extent does the result reflect the style of the reference image?" (1 = Completely inconsistent, 5 = Completely consistent).
- **Content Preservation**: "To what extent does the result preserve the content structure of the original scene?" (1 = Completely lost, 5 = Completely preserved).
- **Visual Naturalness**: "How natural does the result look?" (1 = Very unnatural, 5 = Very natural).

Statistical analysis. We computed the Mean Opinion Score (MOS) for each method. Given the non-parametric nature of the ordinal ratings, we used the Friedman test to assess overall differences among methods, followed by the Nemenyi post-hoc test for pairwise comparisons. A significance level of $p < 0.05$ was adopted.

Results and discussion. As reported in Table 2 and Figure 9, StyleLift3D achieves the best content preservation and visual naturalness. Although its style consistency is slightly lower than that of Stylos and AnyStyle, it remains competitive. Notably, participants reported a clear trade-off (trilemma) between style consistency, content preservation, and visual naturalness. Methods with stronger style consistency often sacrifice content fidelity. StyleLift3D mitigates this tension by maintaining competitive style consistency while significantly improving content preservation and naturalness, yielding the highest overall preference (Equal weight aggregate score).



(a) Criterion scores (b) Score profiles

Figure 9: User study. Mean opinion scores on a 5-point Likert scale (higher is better) for StyleGaussian, Stylos, AnyStyle, and StyleLift3D, evaluated on style consistency, content preservation, and visual naturalness. (a) Category-wise scores, with exact values annotated above the bars and the highest score in each category shown in bold. (b) The same scores overlaid in a shared radar plot. StyleLift3D receives the highest scores for content preservation and visual naturalness.

A.5 TRAINING DETAILS

Objective and optimization. All trainable components are jointly optimized end-to-end using the objective described in Sec. 3.6, which combines pixel-level $\mathcal{L}_1/\mathcal{L}_2$ reconstruction against the stylized target views, the style loss, and the splitter gate regularizer. Both the 2D stylization and depth estimation are performed offline. For each (scene, style) pair, the multi-view stylized images are precomputed and treated as fixed supervision, so the 2D stylizers are not part of the training graph. The pretrained patch-level feature extractor in the lifting backbone is frozen, while its six-layer aggregator, the content encoder, octree-local attention module, decoder, prediction heads, and splitter are optimized jointly.

We use AdamW with a peak learning rate of 10^{-4} and weight decay of 10^{-2}. The learning rate is cosine-annealed to 15% of its peak value, with the content encoder using a learning rate equal to one tenth of the global rate. Gradients are clipped to a global ℓ_2 norm of 5. We set $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$ for the pixel-wise $\mathcal{L}_1$ and $\mathcal{L}_2$ terms, respectively. The overall style-loss weight is $\lambda_{\text{style}} = 1.0$, with $\lambda_{\text{nnfm}} = 2.0$, $\lambda_{\text{content}} = 0.1$, and $\lambda_{\text{tv}} = 10^{-4}$ for the NNFM, VGG content, and total-variation terms, respectively.

Training is performed for 100 epochs on 8 NVIDIA H200 GPUs and takes approximately four days to complete. Each micro-batch contains up to four style sets from the same content scene, allowing them to share scene geometry, content tokens, and the octree structure. Gradients are accumulated over three micro-batches before each optimizer update. For each style set, ten views provide style evidence for lifting into 3D, while all sixteen views are rendered for supervision. At inference time, StyleLift3D requires only a single forward pass and performs no per-scene optimization.

Training datasets. Our training dataset is constructed from 100 reconstructed 3DGS scenes. Each scene contains a base Gaussian representation with spherical harmonics of degree 3, sixteen registered content views with corresponding depth maps, and ninety sets of stylized multi-view images. These ninety sets are generated from thirty style references sampled from WikiArt (Saleh & Elgammal, 2016), each processed independently by three off-the-shelf 2D stylizers: AttenST (Huang et al., 2025), StyleSSP (Xu et al., 2025a), and StyleGallery (He et al., 2026). The 30 style references are shown in Figure 10 and cover a diverse range of artistic appearances. Consequently, each style reference yields three distinct stylization variants for the same scene, and the complete set is available for every training scene. The resulting dataset contains $100 \times 30 \times 3 = 9{,}000$ (scene, style, stylizer) combinations and $9{,}000 \times 16 = 144{,}000$ stylized images. For each combination, ten views are used as style evidence during lifting, while all sixteen views are used for rendering supervision.

The scenes cover a broad range of reconstruction complexity. Their base Gaussian representations contain between 168k and 1.50M primitives, with a median of 506k, a mean of 589k, and a standard deviation of 301k. The distribution is shown in Figure 11. The original content views and depth maps are stored at scene-specific native resolutions, ranging from 956×536 to 1600×899 across 45 distinct sizes. Within each scene, all sixteen views and their depth maps share the same resolution.

For training, the precomputed stylized images have a resolution of 512×512, following preprocessing with a centered square crop and resizing. The stored depth maps undergo the same crop and are bilinearly resized to the stylized-image resolution during loading. For an original image of size $W \times H$, the crop has side length $S = \min(W, H)$. Camera intrinsics are adjusted by subtracting the crop offset from the principal point and scaling both the focal lengths and the shifted principal point by $512/S$. This keeps the stylized images, depth maps, and camera projections aligned. Both the depth maps used for backprojection and the rendered RGB and depth maps are therefore 512×512 in our experiments.

Ablation datasets. For the ablation experiments, we construct a deterministic subset of the full training datasets containing 50 scenes and 15 style references. To preserve the distribution of scene complexity, all 100 scenes are first sorted by the number of Gaussian primitives, with scene name used as a secondary key, and adjacent scenes are grouped into pairs. One scene is then sampled from each pair using `random.Random(42)`, yielding 50 scenes in total. An independent `random.Random(42)` instance selects 15 of the 30 style references, with indices {00, 01, 02, 03, 04, 07, 08, 13, 17, 18, 20, 23, 24, 27, 28}. For every selected scene and style reference, we retain the outputs of all three training stylizers, AttenST, StyleSSP, and StyleGallery, together with all sixteen views. The ablation subset therefore contains $50 \times 15 \times 3 = 2{,}250$ (scene, style, stylizer)

combinations and $2{,}250 \times 16 = 36{,}000$ stylized images, exactly one quarter of the corresponding full training corpus.

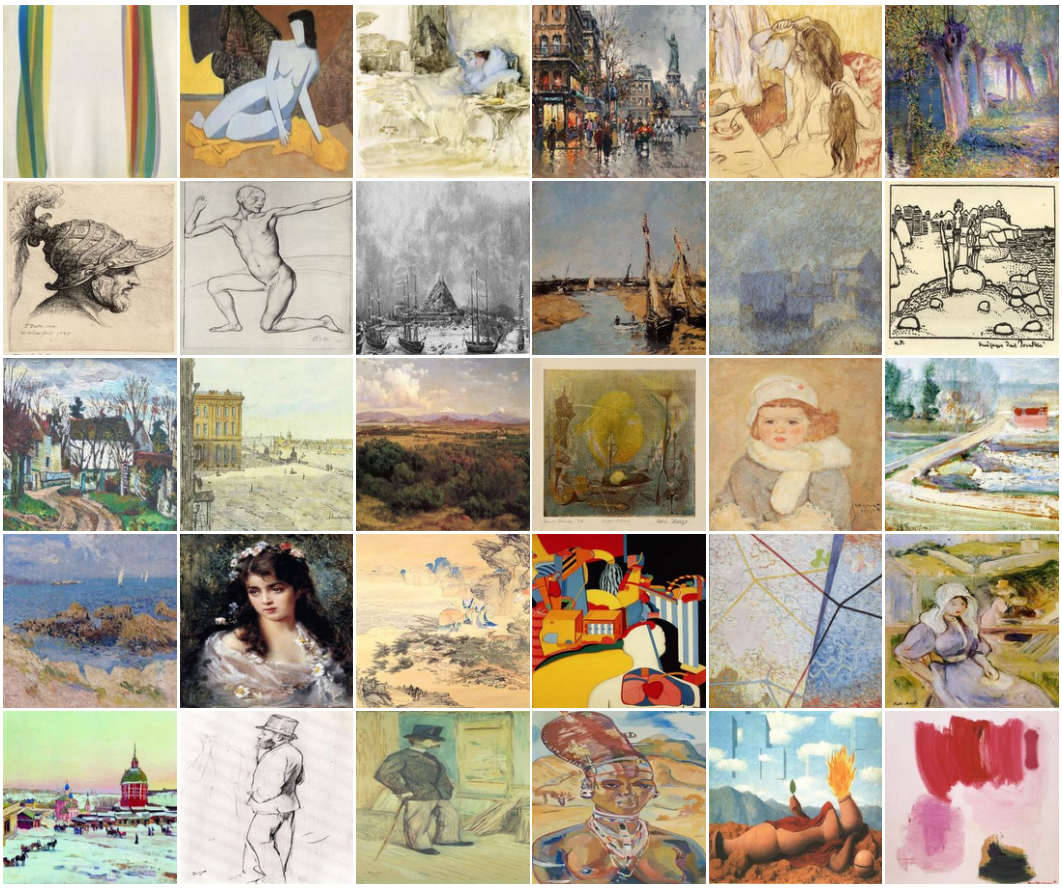


Figure 10: Style-reference gallery used to construct the training dataset. We sample 30 style images from WikiArt (Saleh & Elgammal, 2016), covering diverse artistic appearances including paintings, sketches, abstract compositions, and illustration-like styles.

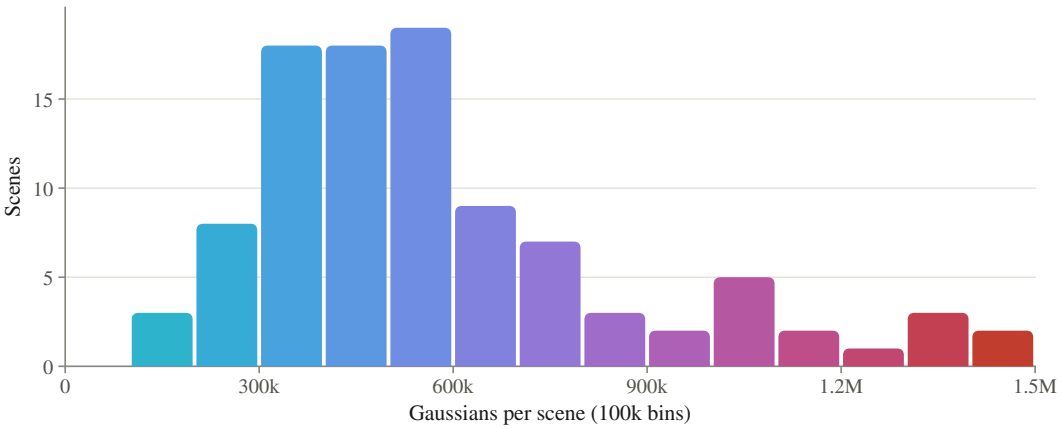


Figure 11: Distribution of Gaussian counts across the 100 training scenes. The reconstructed scenes contain 168k to 1.50M Gaussian primitives, with a median of 506k and a mean of 589k. The histogram uses 100k-wide bins, illustrating the substantial variation in scene reconstruction complexity.

A.6 MULTI-VIEW CONSISTENCY

Evaluation protocol. Following prior work (Liu et al., 2026), we evaluate multi-view consistency using optical-flow-based warping errors. RAFT (Teed & Deng, 2020), with the `raft-things` weights and 32 refinement iterations, estimates optical flow from frame i to frame j. We then forward-warp frame i using softmax splatting, with weights $\exp(-\alpha, \ell_1)$ and $\alpha = 10$ to downweight unreliable correspondences. LPIPS with VGG features (Zhang et al., 2018) and RMSE are computed between the warped frame and frame j, excluding pixels not covered by the splat. We evaluate both short-range pairs ($i \to i + 1$) and long-range pairs ($i \to i + 7$). All results in Tables 7 and 8 are obtained using this protocol.

Controlled diagnostic experiment. Optical-flow-based warping metrics are intended to measure whether rendered appearance remains coherent across viewpoints. To examine how strongly these metrics depend on image content itself, we construct a controlled setting in which the underlying 3D consistency is fixed by design. Specifically, we render a single cube using one camera trajectory and one lighting configuration, while varying only the spatial frequency of its surface albedo. Over 48 frames, the camera travels $120°$ around the cube, giving an angular separation of $2.55°$ for short-range pairs and $17.8°$ for long-range pairs. We compare three albedos: a crisp procedural checkerboard and two progressively smoothed versions obtained by low-pass filtering the same procedural pattern over balls of radius 0.05 and 0.10 in object space.

Importantly, the low-pass operation is applied *within the material*: the procedural pattern is averaged locally in object space and subsequently evaluated by the renderer as ordinary surface appearance. The right two columns of Figure 12 are therefore genuine renders of smoother albedos rather than filters applied to completed images. Geometry, camera trajectory, lighting, and the rendering process are otherwise identical across all three variants. Hence, all sequences are generated from equally coherent 3D representations by construction; only their surface-frequency statistics differ.

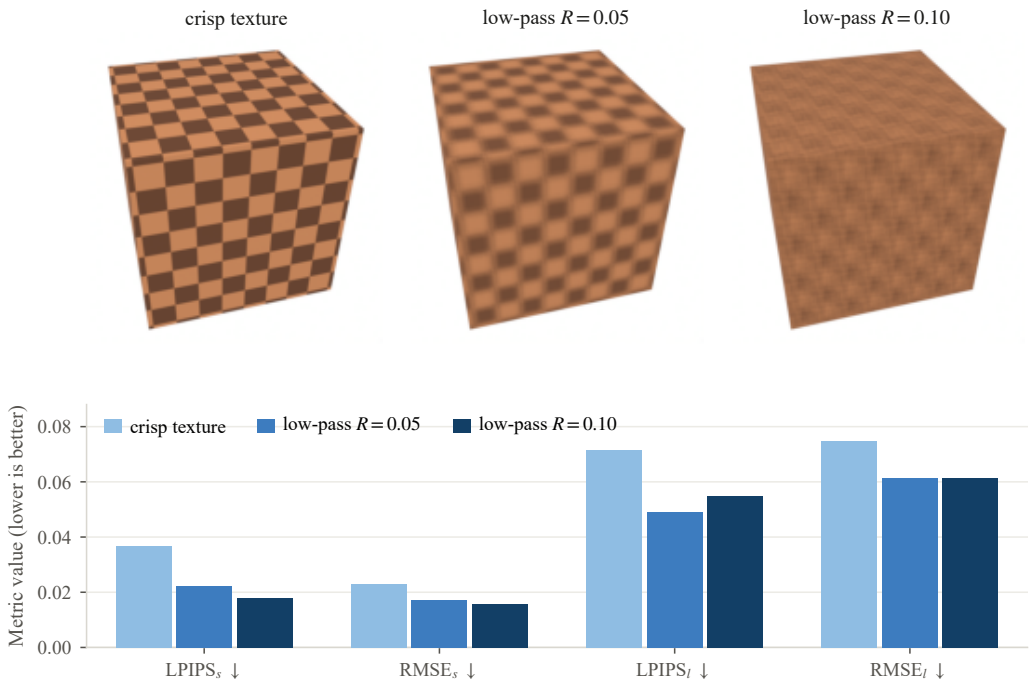


Figure 12: Surface-texture sensitivity of the warping-based evaluation. **Top:** the same cube view with a crisp checker and two smoothed albedos. Smoothing is applied in object space, not to rendered images. **Bottom:** the corresponding short- and long-range scores. Despite unchanged geometry, smoothing lowers errors.

Texture sensitivity under fixed geometry. Despite identical underlying geometry and camera motion, the resulting warping scores vary substantially with albedo frequency. Both low-pass variants obtain lower errors than the crisp checkerboard on all four measures. Relative to the crisp variant, the errors differ by factors of $1.66/1.35/1.46/1.22$ (LPIPS$_s$/RMSE$_s$/LPIPS$_l$/RMSE$_l$) for R=0.05 and $2.07/1.47/1.30/1.22$ for R=0.10 (Table 5). Thus, reducing high-frequency surface detail alone can substantially improve the reported consistency scores.

Two additional observations illustrate that this dependence is not a simple monotonic preference for smoother images. First, for long-range LPIPS, the more strongly smoothed R=0.10 variant performs worse than R=0.05 (0.05503 vs. 0.04922). Second, long-range RMSE is nearly insensitive to the difference between the two smoothed textures: their scores differ by less than 10^{-6} (0.0612560 vs. 0.0612567). At this larger viewpoint separation, the similarity of these values suggests that the residual is increasingly governed by image structures shared across the variants, such as silhouettes and large-scale motion, rather than by the remaining surface texture.

Table 5: Effect of surface smoothing on warping-based scores for the same cube, lighting, and camera trajectory. Subscripts s and l denote frame offsets of 1 and 7, respectively.

Surface albedo	**Short-range**		**Long-range**	
	LPIPS$_s$↓	RMSE$_s$↓	LPIPS$_l$↓	RMSE$_l$↓
Crisp checker	0.03685	0.02310	0.07166	0.07465
Low-pass, $R = 0.05$	0.02218	0.01712	0.04922	0.06126
Low-pass, $R = 0.10$	0.01783	0.01576	0.05503	0.06126

Warp residuals are strongly texture-dependent. This behavior can be further examined directly through the warp residual. Over the short-range pairs (Table 6), the mean absolute residual decreases from 16.35 for the crisp checkerboard to 13.40 and 9.98 as the low-pass radius increases. This trend closely accompanies the fraction of pixels containing an edge, which decreases from 8.74 to 2.63 and 1.65%, respectively. Removing the texture altogether, while keeping the same cube geometry and camera trajectory, further reduces the residual to 5.67. The resulting $2.9\times$ variation in warp error is therefore induced purely by changes in surface appearance, without any change to the underlying 3D geometry or camera motion.

This sensitivity is consistent with the classical ambiguity of optical flow in weakly textured regions. Under the brightness-constancy formulation, a local image gradient primarily constrains the component of motion normal to that gradient, while textureless regions provide little local evidence for a unique two-dimensional displacement. Consequently, flow errors in low-texture regions can incur only a small image-space penalty. A stylization that suppresses texture may therefore obtain a lower warp residual without corresponding to a more coherent underlying 3D representation. In this sense, the metric jointly reflects view consistency and the image statistics that determine how strongly the estimated correspondence is constrained.

Table 6: Edge density and mean absolute warp residual for short-range pairs ($i \to i + 1$) in the controlled cube experiment. Edge pixels have a gradient magnitude above $10/255$. Only surface texture varies across rows.

Surface albedo	**Edge pixels (%)**	**Warp residual**
Crisp checker	8.74	16.35
Low-pass, $R = 0.05$	2.63	13.40
Low-pass, $R = 0.10$	1.65	9.98
No texture	1.59	5.67

LPIPS and RMSE can induce opposite rankings. A particularly clear example is obtained by rendering the same cube geometry with flat white faces bounded by black edge lines, which we refer to as the cel sequence. This sequence achieves an LPIPS$_s$ of 0.01250, lower than all 32 method entries in Table 7, yet its RMSE$_s$ is 0.04585, nearly twice the crisp checkerboard's 0.02310. Thus, LPIPS and RMSE rank these two geometrically identical sequences in opposite orders.

Because both residuals are computed from the same estimated flow field, this reversal cannot be attributed to differences in optical-flow estimation between the two metrics. Instead, it demonstrates that perceptual and pixelwise warp errors respond differently to the appearance statistics of the rendered sequence. The ranking induced by either metric is therefore not invariant to stylization, even when the underlying geometry and camera trajectory remain unchanged.

Full benchmark results. For completeness, Tables 7 and 8 report the full measurements on eight scenes and four methods for both short-range ($i \rightarrow i{+}1$) and long-range ($i \rightarrow i{+}7$) pairs. Each table contains LPIPS and RMSE measurements, together with the unstylized content sequence evaluated under the same protocol as an unranked reference. All values are produced using exactly the evaluation procedure described above. Among the four compared stylization methods, StyleLift3D obtains the lowest error in four of the sixteen scene–metric entries at short range and three at long range, and the highest error in three and one entries, respectively. We report all scene-level measurements rather than only aggregate or favorable cases.

Table 7: Short-range warping-based evaluation ($i \rightarrow i+1$) on eight scenes. Content (ref.) is the unstylized video evaluated with the same protocol and is excluded from method comparisons. Each stylization method category is visualized with a distinct color. Best and second-best results among stylization methods are shown in **bold** and underline, respectively.

Method	Train		Truck		M60		Ballroom	
	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$
Content (ref.)	0.06029	0.07093	0.05717	0.06609	0.04146	0.04729	0.07516	0.06262
StyleGaussian	0.05818	0.05300	**0.04500**	0.04239	0.03994	0.04928	0.06140	0.05462
Stylos	**0.04585**	**0.04150**	0.04690	**0.04138**	0.04221	0.04885	**0.04467**	**0.03809**
AnyStyle	0.05567	0.05758	0.05740	0.06131	0.04780	0.07773	0.05412	0.05530
StyleLift3D	0.04736	0.05163	0.05734	0.06105	**0.03433**	**0.04597**	0.06166	0.05014

Method	Museum		Palace		Panther		Horse	
	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$	$\text{LPIPS}_s\downarrow$	$\text{RMSE}_s\downarrow$
Content (ref.)	0.07337	0.07947	0.04355	0.06129	0.03965	0.03783	0.06423	0.08235
StyleGaussian	0.05555	0.06785	0.04194	0.04906	**0.02767**	**0.02929**	0.08826	0.08424
Stylos	**0.02979**	**0.04217**	0.04225	**0.03992**	0.03754	0.04122	0.06437	**0.07969**
AnyStyle	0.03404	0.05163	0.03971	0.04547	0.04486	0.06543	0.07610	0.10799
StyleLift3D	0.06416	0.06855	**0.03575**	0.04703	0.03542	0.03887	**0.05415**	0.08004

Table 8: Long-range warping-based evaluation ($i \rightarrow i+7$) on the same eight scenes.

Method	Train		Truck		M60		Ballroom	
	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$
Content (ref.)	0.17291	0.17862	0.16212	0.17549	0.14093	0.18061	0.19328	0.17308
StyleGaussian	0.16514	**0.12381**	0.11728	**0.10974**	0.12201	0.15426	0.12752	0.10900
Stylos	**0.10780**	0.12464	**0.11601**	0.12696	**0.10624**	**0.13094**	**0.08067**	**0.10194**
AnyStyle	0.13837	0.15726	0.17450	0.18822	0.15266	0.19318	0.13091	0.15780
StyleLift3D	0.13742	0.14294	0.15807	0.16720	0.11953	0.16337	0.13850	0.13296

Method	Museum		Palace		Panther		Horse	
	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$	$\text{LPIPS}_l\downarrow$	$\text{RMSE}_l\downarrow$
Content (ref.)	0.20406	0.19427	0.13756	0.17640	0.13043	0.13240	0.13871	0.18673
StyleGaussian	**0.16911**	**0.16715**	0.10937	**0.10692**	**0.08469**	**0.08596**	0.18404	0.19018
Stylos	0.18175	0.24112	0.10668	0.12657	0.10430	0.13039	0.14392	0.18545
AnyStyle	0.21806	0.26104	0.11124	0.13935	0.15557	0.19476	0.18141	0.23611
StyleLift3D	0.21028	0.19607	**0.10569**	0.13765	0.11498	0.13037	**0.13000**	**0.18538**

Rationale for reporting these metrics. The controlled analysis above shows that optical-flow-based warping errors should not be interpreted as a texture-invariant measure of 3D view consistency in the stylized domain. Nevertheless, this does not make the measurements uninformative for comparison with prior work. The protocol was adopted by previous work (Liu et al., 2026), and the baselines considered here are evaluated using the same procedure. Reporting the corresponding measurements therefore enables a direct and reproducible comparison under the established protocol. For this reason, we follow the metric exactly as defined and report the complete results, including scene–metric combinations for which StyleLift3D does not obtain the lowest error.

Interpretation. We therefore do not use the ordering of Tables 7 and 8 as direct evidence of superior or inferior view continuity, nor do we select or tune models using these metrics. In stylized imagery, a lower optical-flow warping error can arise from reduced texture complexity as well as from improved cross-view coherence. We consequently rely on the user study in Section A.4 for perceptual evaluation of the rendered results, while reporting the warping metrics primarily for compatibility with existing evaluation protocols. More generally, when such metrics are applied to stylized scenes, reporting complementary information about output image statistics may help distinguish improvements in correspondence from changes caused primarily by texture simplification.

A.7 Limitations and failure cases

Reduced sky stylization in open scenes. Sky regions can exhibit weaker stylization than the rest of an open scene. In the Palace example in Figure 13, the sky occupies roughly one third of the image but is represented by only $23{,}931$ of $691{,}613$ input Gaussians (3.5%). Such a sparse representation may limit the capacity of StyleLift3D, which predicts residual updates to existing Gaussians, to reproduce localized appearance changes. In addition, the independently stylized guidance is less consistent across views: across all style variants, the stylization-induced color shift shows $1.84\times$ greater cross-view variation in the sky than in non-sky regions.

As illustrated in Figure 13, the 2D guidance suppresses the original blue sky and introduces a warmer palette, whereas our render retains a stronger blue-violet cast while preserving the stylized appearance elsewhere. Across the same variants, StyleLift3D retains 76% of the guidance-induced color-change magnitude in the sky, compared with 99% in non-sky regions. We find no missing depth values or unmatched Gaussian tokens, suggesting that this behavior is not caused by incomplete lifting or decoding. Instead, sparse sky geometry together with inconsistent multi-view guidance may encourage conservative updates toward the original appearance. Potential improvements include view-consistent sky guidance and agreement-aware weighting of lifted style features across views.



Figure 13: Reduced sky stylization on Palace with AttenST guidance. From left to right: original content, stylized 2D guidance, and the StyleLift3D render. The bottom row enlarges the boxed sky region. The render preserves the warm appearance of the facade, trees, and water but retains more of the original blue-violet sky than the guidance.