STYLOS: MULTI-VIEW 3D STYLIZATION WITH SINGLE-FORWARD GAUSSIAN SPLATTING

Hanzhou Liu[*] **Jia Huang** **Mi Lu** **Srikanth Saripalli** **Peng Jiang**[*]
Texas A&M University

ABSTRACT

We present ***Stylos***, a single-forward 3D Gaussian framework for 3D style transfer that operates on unposed content, from a single image to a multi-view collection, conditioned on a separate reference style image. Stylos synthesizes a stylized 3D Gaussian scene without per-scene optimization or precomputed poses, achieving geometry-aware, view-consistent stylization that generalizes to unseen categories, scenes, and styles. At its core, Stylos adopts a Transformer backbone with two pathways: geometry predictions retain self-attention to preserve geometric fidelity, while style is injected via global cross-attention to enforce visual consistency across views. With the addition of a voxel-based 3D style loss that aligns aggregated scene features to style statistics, Stylos enforces view-consistent stylization while preserving geometry. Experiments across multiple datasets demonstrate that Stylos delivers high-quality zero-shot stylization, highlighting the effectiveness of global style-content coupling, the proposed 3D style loss, and the scalability of our framework from single view to large-scale multi-view settings. Our codes are available at https://github.com/HanzhouLiu/Stylos.

1 INTRODUCTION

Image guided 3D stylization aims to preserve scene geometry and cross-view consistency while transferring the reference style. With the rise of immersive content, augmented and virtual reality, demand for this capability is growing. Nevertheless, achieving reliable 3D stylization remains challenging and continues to attract significant research attention.

Early attempts rely on explicit 3D geometry such as meshes or point clouds, but their stylization quality is often limited by geometric reconstruction accuracy, resulting in noticeable artifacts in complex scenes (Kato et al., 2018; Cao et al., 2020). The introduction of implicit radiance fields, most notably NeRF (Mildenhall et al., 2020), enables higher-fidelity rendering, and subsequent works extended it to artistic stylization (Zhang et al., 2022; Huang et al., 2022; Liu et al., 2023), but typically require costly per-scene optimization, constraining their generalization to unseen scenes. More recently, 3D Gaussian Splatting (3DGS) has emerged as a promising explicit representation that combines high reconstruction quality with real-time rendering efficiency (Kerbl et al., 2023). Stylization approaches built on 3DGS achieve efficient multi-view consistency (Liu et al., 2024; Galerne et al., 2025), yet still struggle to generalize beyond scene-specific training.

In contrast, we introduce ***Stylos*** (meaning *pens* in French), a single-forward framework for 3D style transfer that eliminates the need for per-scene optimization and precomputed camera parameters, and generalizes effectively to novel categories, scenes, and styles. Stylos employs a shared Transformer backbone with two pathways: content and style images are projected into a shared feature space, where content retains self-attention for geometric reasoning, and style is injected via global cross-attention to condition aesthetics. Notably, the global cross-attention also helps preserve fine visual details in the Gaussian splatting renderings. Geometry-related attributes, such as depth, camera intrinsics, and extrinsics, are derived from backbone features, whereas style conditioning guides the prediction of color coefficients. These outputs are estimated through prediction heads that serve as the interface between feature space and the final Gaussian representation. Prior works in 2D style transfer typically rely on distribution alignment (Li et al., 2017), which operate on image statistics

[*] Equal Contribution

but lack explicit multi-view or 3D consistency. To address this, we explore alternative objectives and propose a voxel-based 3D style loss that aligns aggregated scene features with style statistics, providing stronger view-consistent stylization while preserving geometric fidelity.

We evaluate Stylos across different scenarios, including category-level transfer and cross-scene generalization. Our assessment spans challenging real-world benchmarks, where Stylos produces stylized renderings with high visual fidelity and consistent geometry, demonstrating robustness even in previously unseen environments. Our main contributions are threefold:

- We propose a shared-backbone design with two pathways: geometry predictions retain self-attention for geometric reasoning, while style is injected through global cross-attention to condition aesthetics.
- We introduce a voxel-based 3D style loss that aligns aggregated scene features with style statistics, enforcing cross-view coherence and geometry-aware stylization.
- We develop Stylos, a single-forward-pass pipeline for 3D style transfer from unposed inputs, scaling from a single to hundreds of views with a single style image, and achieving zero-shot generalization to unseen categories, scenes, and styles.

2 Related Work

2.1 Single-Forward 3D Reconstruction

Early CNN models (Ummenhofer et al., 2017; Zhou et al., 2018; Teed & Deng, 2020) demonstrate the feasibility of end-to-end 3D reconstruction. While the Transformer-based DUSt3R (Wang et al., 2024) has shown impressive performance, yet remains constrained by pairwise inputs and reliance on traditional optimization for pose estimation. VGGT (Wang et al., 2025a) advances this line by jointly predicting camera parameters, depth, point maps, and tracks from one to hundreds of views in a single forward pass, while AnySplat (Jiang et al., 2025) complements it with a feed-forward Gaussian splatting pipeline trained with VGGT priors to directly predict Gaussian parameters along with depth and camera parameters from uncalibrated images.

2.2 3D Style Transfer and Losses

Explicit-representation approaches to 3D stylization (Kato et al., 2018; Cao et al., 2020; Guo et al., 2021; Huang et al., 2021; Höllein et al., 2022; Mu et al., 2022) have shown promising results, but their reliance on accurate geometry limits applicability in complex real-world scenes. Implicit neural representations like NeRF (Mildenhall et al., 2020) improve geometric consistency, inspiring a range of stylization methods based on stylized view supervision or auxiliary networks (Nguyen-Phuoc et al., 2022; Zhang et al., 2022; Chiang et al., 2022; Huang et al., 2022). StylizedNeRF (Huang et al., 2022) addresses the domain gap between style images and NeRF by introducing a mutual learning framework. StyleRF (Liu et al., 2023) further enables zero-shot transfer by applying style transformations in radiance-field feature space. However, training NeRF for scene reconstruction is typically computationally expensive, making NeRF-based 3D stylization far from real-time.

More recently, 3D Gaussian Splatting (3DGS) (Kerbl et al., 2023) has emerged as an efficient alternative to NeRF for novel view synthesis. Subsequent works further refine 3DGS for efficient stylization (Mei et al., 2024; Zhang et al., 2025; Kovács et al., 2024; Yu et al., 2024; Galerne et al., 2025). StyleGaussian (Liu et al., 2024) extends 3DGS with efficient feature rendering to enable real-time zero-shot stylization. However, despite being significantly faster to optimize than NeRF, 3DGS-based approaches still require per-scene fitting, which poses a fundamental barrier to achieving truly real-time 3D stylization. Styl3R (Wang et al., 2025b), the closest contemporaneous related work to ours, introduces a feedforward framework for instant 3D stylized reconstruction. While effective and efficient, Styl3R is primarily designed for 2–8 input views and does not specifically target strong multi-view consistency among the rendered stylization results.

Underlying these architectures, style transfer objectives play a crucial role. Classical style transfer relies on Gram-matrix correlations (Gatys et al., 2016), while channel-wise statistics such as mean and variance (Li et al., 2017) offer efficient alternatives, forming the basis of AdaIN (Huang & Belongie, 2017). Extensions to video and multi-view stylization (Ruder et al., 2016; Gupta et al., 2017;

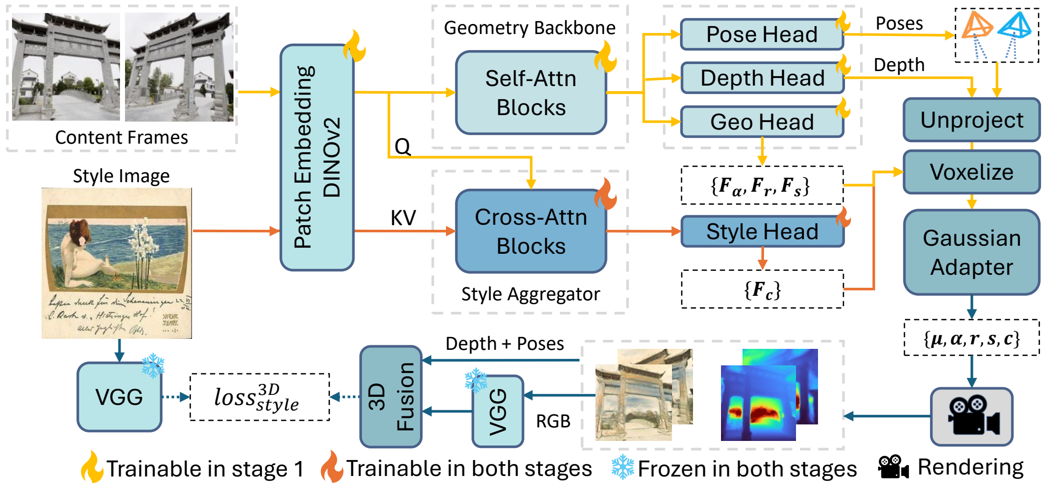


Figure 1: **Architecture overview**. Given multi-view content inputs and a style reference, Stylos enables instant 3D stylization without scene-specific training or post-optimization. A key component is the 3D style loss, matching voxelized 3D features with 2D style statistics.

Nguyen-Phuoc et al., 2022; Zhang et al., 2022) address temporal and cross-view consistency, and CLIP-based losses (Radford et al., 2021) introduce semantic alignment with text or image prompts. However, most objectives remain inherently 2D and cannot guarantee consistency across complex 3D scenes. To this end, we extend statistics-based style losses into voxel space, introducing a 3D-aware objective that aligns feature distributions after multi-view fusion.

3 Method

We propose ***Stylos***, a transformer-based framework for stylized 3D scene reconstruction. Given a style reference image and one or more content views (also referred to as context views), Stylos predicts a set of stylized 3D Gaussian primitives together with camera parameters, enabling faithful reconstruction of the observed scene while transferring the desired style. We first formulate the problem in Sec. 3.1, then present the network architecture in Sec. 3.2, and finally detail our training strategy in Sec. 3.3 and the used style loss in Sec. 3.4.

3.1 Problem Formulation

We aim to disentangle geometry and style within a unified 3D scene representation. The input to Stylos consists of a set of N context views $\{I_i\}_{i=1}^N$ of a scene, with $N \geq 1$ and $I_i \in \mathbb{R}^{H\times W\times 3}$, together with a single style reference image $S \in \mathbb{R}^{H\times W\times 3}$. The context views provide geometric cues, while the style image specifies the desired aesthetic.

Formally, Stylos defines a conditional mapping,

$$f_\theta : (\{I_i\}_{i=1}^N,\, S) \;\mapsto\; (G,\, \{g_i\}_{i=1}^N),$$

where the scene representation $G = \{(p_m,\, c_m)\}_{m=1}^M$ is parameterized by M anisotropic 3D Gaussians. Each Gaussian has geometry attributes $p_m = (\mu_m,\, \alpha_m,\, r_m,\, s_m)$, comprising 3D position $\mu_m \in \mathbb{R}^3$, opacity $\alpha \in \mathbb{R}^+$, orientation quaternion $r_m \in \mathbb{R}^4$, and anisotropic scale $s_m \in \mathbb{R}^3$, as well as a style-dependent color embedding $c_m \in \mathbb{R}^{3\times(k+1)^2}$ represented with spherical harmonics of degree k. In addition, Stylos jointly predicts camera parameters $\{g_i \in \mathbb{R}^9\}_{i=1}^N$ for each input view, leveraging pose cues estimated by the VGGT backbone.

3.2 Network Architecture

Stylos employs a geometry backbone that alternates frame-wise and global attention to process multiple input views and infer geometry-related parameters such as positions, scales, orientations, and opacities. To enable stylization, we introduce a dedicated conditioning branch, the *Style Aggregator*, which fuses content and style features through cross-attention and predicts style-aware color embeddings for each Gaussian. In this way, geometry remains derived solely from the backbone, while style representation is conditioned on the style reference. We next detail the geometric backbone in Sec. 3.2.1, the style aggregation module in Sec. 3.2.2, and the prediction heads in Sec. 3.2.3.

3.2.1 GEOMETRIC BACKBONE

The geometric backbone follows the alternating-attention design of VGGT (Wang et al., 2025a), which interleaves frame and global self-attention layers to capture both intra-frame structure and cross-view consistency. This component is kept unchanged to retain strong geometric reasoning, serving as the foundation on which we integrate style information.

3.2.2 STYLE AGGREGATOR

The Style Aggregator builds on the geometric backbone inherited from VGGT, which is composed of standard Transformer Block containing self-attention followed by feed-forward layers. To enable style conditioning, we replace this standard block with an adapted $\mathrm{CrossBlock}$, which inserts a cross-attention operation between the self-attention and MLP stages (Deng et al., 2022). In $\mathrm{CrossBlock}$, content tokens serve as queries ($\mathcal{Q}$) while style tokens provide keys and values ($\mathcal{KV}$). This structure allows the content representations to be refined by their own spatial context (via the preserved self-attention) while being explicitly conditioned on the target style (via the inserted cross-attention) before the final feed-forward projection.

Content–Style Coupling. Let $\mathcal{Q}_{b,v} \in \mathbb{R}^{L_q \times C}$ denote the content tokens extracted from the v-th view of the b-th sample, and let $\mathcal{KV}_b \in \mathbb{R}^{L_{kv} \times C}$ denote the style tokens. We use $\mathrm{CrossBlock}(\mathcal{Q}, \mathcal{KV})$ to denote our block operator that updates $\mathcal{Q}$ using $\mathcal{KV}$ as key–value pairs. Depending on how the query set is formed, we consider two primary topological strategies.

(1) Frame cross attention. Each view independently interacts with the style tokens,

$$\widetilde{\mathcal{Q}}_{b,v} = \mathrm{CrossBlock}(\mathcal{Q}_{b,v},\, \mathcal{KV}_b). \tag{1}$$

Since the internal self-attention of the block operates only on $\mathcal{Q}_{b,v}$, this strategy preserves view-specific geometric structure but prevents information propagation between different views.

(2) Global cross attention. We first concatenate all views to obtain a global sequence $\mathcal{Q}_b^{\mathrm{global}} \in \mathbb{R}^{VL_q \times C}$ by $\mathcal{Q}_b^{\mathrm{global}} = \mathrm{Concat}_{v=1}^{V} \mathcal{Q}_{b,v}$, and perform the block operation simultaneously,

$$\widetilde{\mathcal{Q}}_b^{\mathrm{global}} = \mathrm{CrossBlock}(\mathcal{Q}_b^{\mathrm{global}},\, \mathcal{KV}_b). \tag{2}$$

We then reshape the updated tokens back to per-view tensors, $\widetilde{\mathcal{Q}}_{b,v}$, by splitting $\widetilde{\mathcal{Q}}_b^{\mathrm{global}}$ along the sequence dimension. This approach enables long-range reasoning: the internal self-attention ensures multi-view geometric consistency, while the cross-attention broadcasts style information globally.

(3) Hybrid cross attention. We first apply frame attention to refine each view independently as in Eq. 1, and then perform global attention by concatenating the resulting $\widehat{\mathcal{Q}}_{b,v}$ across views and applying the same operator as in Eq. 2. This hierarchical coupling combines per-view refinement with multi-view aggregation.

3.2.3 PREDICTION HEADS

To connect the geometric backbone and Style Aggregator with the Gaussian scene, we introduce prediction heads that translate feature tokens into explicit parameters. These heads serve as modular interfaces, keeping geometry and style distinct in feature space while ensuring their integration in the final Gaussian representation. We next describe the individual heads in detail.

Geometry Head. Geometric backbone features are passed through a DPT-style regression head that outputs the Gaussian geometry parameters $p_m = (\mu_m, s_m, r_m, \alpha_m)$, i.e., position, scale, orientation, and opacity (as defined in Sec. 3.1). By relying on this established design, structural predictions are derived from backbone features alone, without direct influence from style conditioning.

Style Head. The outputs of the Style Aggregator are subsequently processed by a color head to predict the spherical-harmonic coefficients c_m that define appearance. This pathway injects style information directly into Gaussian colors while leaving the geometry parameters p_m unaffected, enabling the two factors to be recombined seamlessly at the Gaussian level.

Auxiliary Heads, Adapter, and Voxelization. We employ the existing VGGT camera head to estimate camera intrinsics and extrinsics, and a depth head to predict scene geometry cues, which are unprojected into 3D anchors for Gaussian placement (Kerbl et al., 2023; Ren et al., 2025). A Gaussian adapter then consolidates geometry and style outputs into a unified set of primitives $\{(p_m, c_m)\}_{m=1}^{M}$ for differentiable rendering. Finally, to reduce redundancy and balance density, we follow the voxelization step introduced by AnySplat (Jiang et al., 2025), nearby points are clustered within a discretized 3D grid and fused using confidence-aware weighting. This operation depends only on the unprojected 3D points and features, and is independent of camera parameters.

3.3 TRAINING STRATEGY

We adopt a two-stage training strategy for structure-aware stylization.

Stage 1: Geometry Pretraining. We initialize the geometric backbone of Stylos with VGGT weights (Wang et al., 2025a) and train the network end-to-end to learn geometry and photometric appearance. To avoid trivial identity mapping and improve robustness to color variations, one input view is randomly selected and color-jittered as the style reference. A frozen VGGT teacher provides pose and depth supervision. The objective combines reconstruction and distillation,

$$\mathcal{L}_{\text{stage1}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{distill}}\,\mathcal{L}_{\text{distill}}.$$

Stage 2: Stylization Fine-tuning. We freeze all geometry-related modules and only update the Style Aggregator and the color head. Following ArtFlow (An et al., 2021), we use feature-level style and content losses in VGG space, matching channel-wise statistics for style and feature activations for content. We further extend these objectives with a 3D voxel-space style loss for cross-view consistency, and add a CLIP-based loss for semantic alignment. A total variation (TV) regularizer is also included to suppress high-frequency artifacts and stabilize optimization. The total loss is,

$$\mathcal{L}_{\text{stage2}} = \mathcal{L}_{\text{rec}} + \lambda_{\text{style}}\,\mathcal{L}_{\text{style}}^{3D} + \lambda_{\text{cnt}}\,\mathcal{L}_{\text{content}} + \lambda_{\text{clip}}\,\mathcal{L}_{\text{clip}} + \lambda_{\text{tv}}\,\mathcal{L}_{\text{TV}}.$$

In our experiments, we set $\lambda_{\text{distill}} = 1.0$, and use $\{\lambda_{\text{style}}, \lambda_{\text{cnt}}, \lambda_{\text{clip}}, \lambda_{\text{tv}}\} = \{1.0, 0.1, 1.0, 10.0\}$

3.4 STYLE LOSSES

We denote by $\mathcal{R}_{b,v}^{l} \in \mathbb{R}^{C_l \times H_l \times W_l}$ the VGG feature map of the rendered image for the b-th scene and v-th view at layer l, and by $\mathcal{S}_b^l \in \mathbb{R}^{C_l \times H_l \times W_l}$ the feature map of the single style image. Building on the Batch Normalization statistics method (Li et al., 2017), which interprets BN mean and variance as style descriptors, we progressively extend it matching beyond standard 2D settings. This progression starts from independent image-level matching, moves to multi-view feature aggregation that promotes global style coherence across views, and culminates in a voxel-space loss that directly constrains the fused 3D representation. Next, we formulate the three losses studied in this paper.

Image-Level Style Loss. The simplest baseline aligns each rendered frame independently with the style reference, which encourages per-frame stylization but does not enforce multi-view consistency,

$$\mathcal{L}_{\text{sty}}^{\text{img}} = \frac{1}{BV} \sum_{b=1}^{B} \sum_{v=1}^{V} \sum_{l=1}^{5} \alpha_l \Big(\|\mu(\mathcal{R}_{b,v}^{l}) - \mu(\mathcal{S}_b^l)\|_2^2 + \|\sigma(\mathcal{R}_{b,s}^{l}) - \sigma(\mathcal{S}_b^l)\|_2^2 \Big), \tag{3}$$

where α_l denotes the weight at each level l, μ is the mean and σ is the standard deviation.

Scene-Level Style Loss. To introduce consistency into multi-view 3D stylization, we concatenate the per-view features $\{\mathcal{R}_{b,v}^{l}\}_{v=1}^{V}$ along the spatial dimension to form $\widetilde{\mathcal{R}}_b^l$, and calculate statistics on this aggregated map. However, this method still operates in 2D feature space and is computed by,

$$\mathcal{L}_{\text{sty}}^{\text{scn}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{l=1}^{5} \alpha_l \Big(\|\mu(\widetilde{\mathcal{R}}_b^l) - \mu(\mathcal{S}_b^l)\|_2^2 + \|\sigma(\widetilde{\mathcal{R}}_b^l) - \sigma(\mathcal{S}_b^l)\|_2^2 \Big). \tag{4}$$

Algorithm 1 3D Voxel-based AdaIN Style Loss

```
Require: Rendered features $\mathcal{R}$, Style features $\mathcal{S}$, 3D points $\mathcal{P}$, Confidence $\mathcal{C}$, Valid mask $\mathcal{M}$
Ensure: 3D style loss $\mathcal{L}_{sty}^{\mathrm{3D}}$
Initialize $\mathcal{L}_{sty}^{3D} \leftarrow 0$
for $b = 1$ to $B$ do
    for $l = 1$ to $5$ do
        Resize $(\mathcal{C}, \mathcal{M}, \mathcal{P})$ to match the resolution of $\mathcal{R}_b^l$
        // Next, fuse all views into a single voxel grid
        $\mathcal{G}_b^l \leftarrow \text{VoxelizeAndFuse}\left(\{\mathcal{R}_{b,v}^l\}_{v=1}^V, \{\mathcal{P}_{b,v}\}_{v=1}^V, \{\mathcal{C}_{b,v}\}_{v=1}^V, \{\mathcal{M}_{b,v}\}_{v=1}^V\right)$
        // Next, compute the BN statistics (mean and standard deviation) of two feature maps
        $(\mu_g, \sigma_g) \leftarrow \text{MeanStd}(\mathcal{G}_b^l)$, $(\mu_s, \sigma_s) \leftarrow \text{MeanStd}(\mathcal{S}_b^l)$
        $\mathcal{L}_{sty}^{3D} \leftarrow \mathcal{L}_{sty}^{3D} + \alpha_l \|\mu_g - \mu_s\|_2^2 + \alpha_l \|\sigma_g - \sigma_s\|_2^2$
    end for
end for
return $\mathcal{L}_{sty}^{3D} = \lambda_{sty} \mathcal{L}_{sty}^{3D}$
```

Voxel-level 3D Style Loss. Finally, to further enforce multi-view consistency, we fuse multi-view features into a voxel grid using differentiable unprojection, where features from different views are accumulated into spatial bins of a discretized 3D volume. Let $\mathcal{G}_b^l$ denote the voxelized features for the b-th scene at layer l. We then compute style statistics directly in voxel space,

$$\mathcal{L}_{\text{sty}}^{\text{3D}} = \frac{1}{B} \sum_{b=1}^{B} \sum_{l=1}^{5} \alpha_l \Big(\|\mu(\mathcal{G}_b^l) - \mu(\mathcal{S}_b^l)\|_2^2 + \|\sigma(\mathcal{G}_b^l) - \sigma(\mathcal{S}_b^l)\|_2^2 \Big). \tag{5}$$

By operating on voxelized features, this loss explicitly encodes geometry and enforces style consistency across both views and the underlying 3D structure. We provide pseudo codes in Algorithm 1.

4 Experiment

Datasets. We evaluate cross-category generalization on the CO3D (Reizenstein et al., 2021) dataset by training on 17 categories and testing on 3 held-out ones, and cross-scene generalization by training on the full DL3DV-10K (Ling et al., 2024) and testing on Tanks & Temples (Knapitsch et al., 2017). Style images are provided by WikiArt (WikiArt, 2025) and DELAUNAY (Gontier et al., 2023), with 50 diverse style images reserved as unseen styles that are never used during training.

Baselines. We compare ***Stylos*** with recent 3D stylization models. The baselines include, (1) a per-scene training and zero-shot method StyleGaussian (Liu et al., 2024), (2) per-scene and per-style optimization approaches G-Style (Kovács et al., 2024), StylizedGS (Zhang et al., 2025), and SGSST (Galerne et al., 2025), and (3) the closest related work Styl3R (Wang et al., 2025b).

Evaluation Metrics. Our evaluation covers three aspects. (1) To assess geometry reconstruction in the style-free setting, we report *PSNR*, *SSIM*, and *LPIPS* (Zhang et al., 2018) between content views and predictions, conditioned on the original first frame of each scene as the style input. (2) To measure stylization consistency, following prior work (Chiang et al., 2022), we compute *LPIPS* and *RMSE* in both short-range and long-range settings. (3) To further evaluate stylization quality, we report *ArtScore* (Chen et al., 2024), a recent metric specifically designed for reference-free evaluation of artness in generated images, and *ArtFID* (Wright & Ommer, 2022; Chung et al., 2024).

4.1 Ablation Study

In this section, we analyze the impact of each design choice on the CO3D dataset. To ensure fairness, all variants are trained with identical configurations in each experimental round.

Style-Content Cross Attention. Take the pizza scene for example. Table 1 shows that the baseline method, combining frame-level and global attention, yields a PSNR of 19.78 dB and LPIPS of 0.3326. The frame variant produces slightly lower PSNR and higher LPIPS. By contrast, applying cross-attention at the global level significantly boosts the PSNR value by 0.79 dB and achieves

Table 1: Ablation on the style-content coupling strategies, comparing frame-level, global-level and hybrid cross-attention. The first frame of each scene is used as the style reference. Reconstruction quality is evaluated with PSNR↑, SSIM↑, and LPIPS↓ on the CO3D dataset.

Strategies		Skateboard			Pizza			Donut		
Global	Frame	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓	PSNR↑	SSIM↑	LPIPS↓
✓	✓	21.12	0.6858	0.2821	19.78	0.5939	0.3326	21.39	0.7198	0.3264
	✓	20.93	0.6917	0.2912	19.72	0.5940	0.3405	21.40	0.7167	0.3340
✓		**21.68**	**0.7043**	**0.2684**	**20.57**	**0.6177**	**0.3110**	**22.09**	**0.7362**	**0.3125**

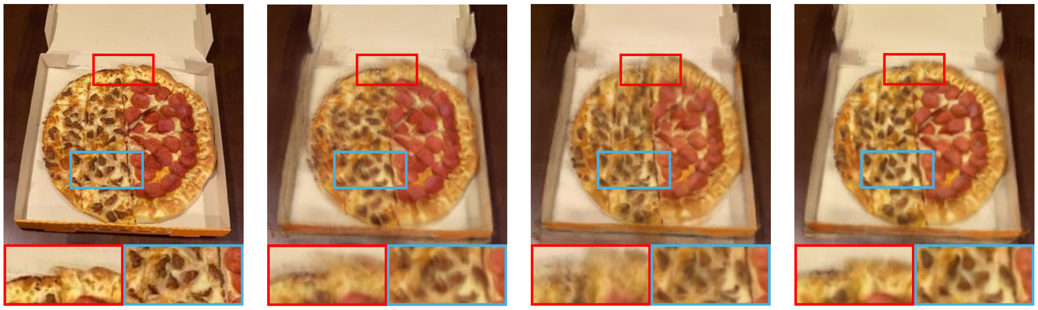

(a) GT (b) Hybrid (c) Frame **(d) Global (Ours)**

Figure 2: CO3D pizza scene comparing different style–content cross-attention strategies.

consistently improved metrics. Qualitative comparisons in Fig. 2 highlight these differences. Frame-only attention produces a poorly defined crust with noticeable artifacts, the hybrid variant recovers part of the structure but leaves the box edges blurred, while global attention delivers the most faithful result, with clear toppings and a well-preserved crust boundary.

Image vs. Scene vs. 3D Style Losses. As shown in Table 2, both scene-level and 3D style losses clearly outperform the image-level baseline in terms of artistic quality. Overall, the 3D losses exhibit superior and more stable consistency performance. These results indicate that enforcing style consistency in a voxel-level yields more coherent stylized textures across views. We observe that the image-level supervision may fail to transfer style. For example, in Fig. 3, the donut surface is not well synthesized with the expected style, whereas the scene and 3D losses successfully apply the target style. The 3D style loss produces sharper boundaries and a stronger sense of 3D geometry, as observed in the skateboard and pizza examples, leading to the most coherent multi-view stylization. We release these model weights to support reproducibility and further analysis.

From a Single View to Dozens of Views. While Stylos can process up to dozens (even hundreds) of views, we observe a gradual decrease in visual quality once the number of views per batch (denoted as "# views / batch") exceeds 32. As shown in Fig. 4, very small batches (1 view) fail to synthesize parts of the scene like the bench, while overly large batches (64 views) introduce edge artifacts on the tall building, potentially due to the gap from our training settings (no more than 24 views). Additionally, we provide A.2 Fig. 7 to show the efficiency trend with respect to the number of views.

Table 2: Comparison of consistency and artistic quality among different style loss designs on CO3D. The frame stride is set to 3, and 15 held-out scenes are randomly selected for evaluation.

Style Loss	Short-range		Long-range		ArtScore↑
	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	
Image loss (Eq. 3)	0.048	0.038	0.157	**0.142**	4.78
Scene loss (Eq. 4)	**0.047**	0.036	0.156	0.148	9.12
3D loss (Eq. 5)	**0.047**	**0.034**	**0.153**	**0.142**	**9.15**

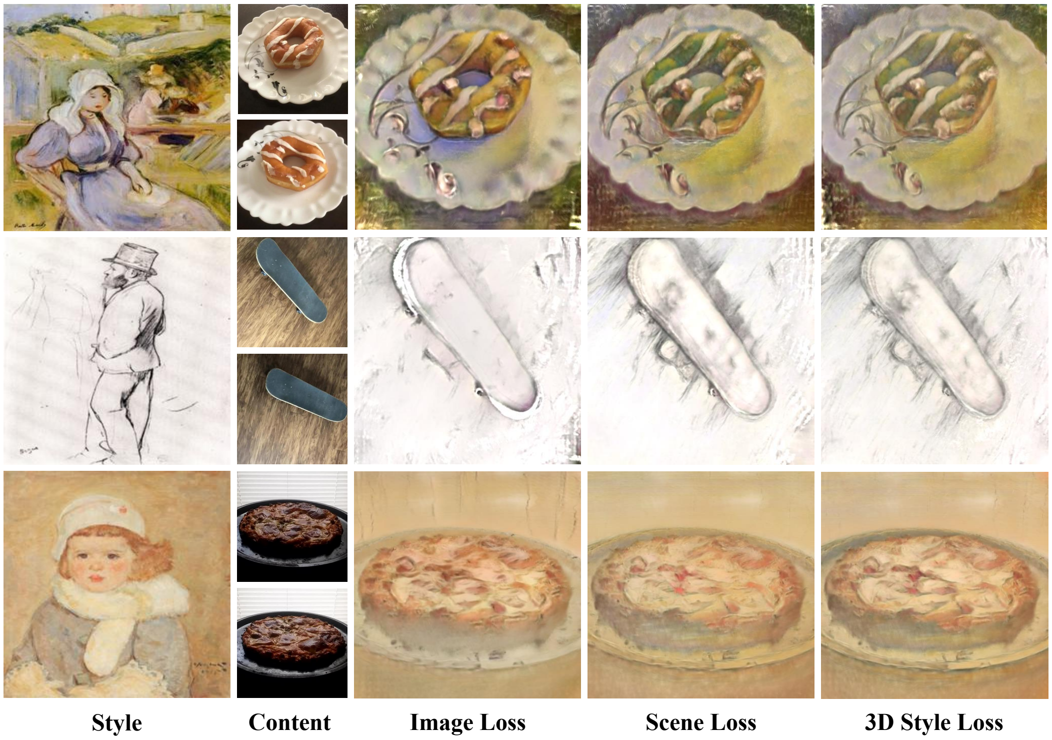


Figure 3: Comparison of style losses on unseen donut, skateboard, and pizza scenes from the CO3D dataset. Both scene and 3D style losses yield cleaner stylized textures compared to image-level matching, while the 3D loss further conveys a stronger sense of 3D geometry. We encourage readers to Appendix Fig. 8-10 for more visual comparisons under varing scenes and styles.

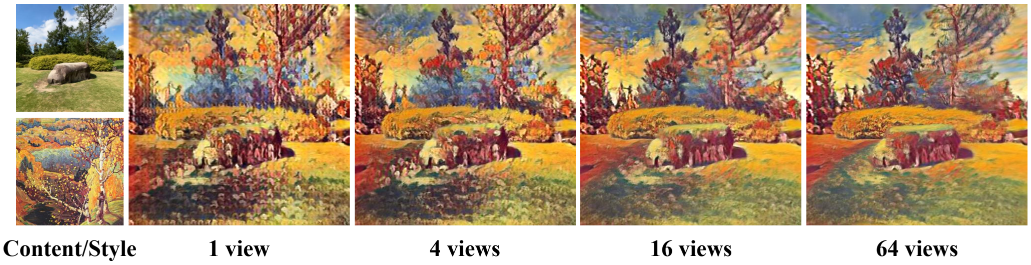


Figure 4: Effect of varying # views / batch on the Lighthouse scene from Tanks and Temples.

4.2 Comparison with State-of-the-Art

To retrain StyleGaussian (Liu et al., 2024), G-Style (Kovács et al., 2024), StylizedGS (Zhang et al., 2025), and SGSST (Galerne et al., 2025), we strictly follow their released codes. Since the pretrained-weights of StyleGaussian (Liu et al., 2024) and Styl3R (Wang et al., 2025b) are publicly available, we use them to generate the visual results. Stylos is trained once on DL3DV (Ling et al., 2024), and tested in a zero-shot manner without prior knowledge of either the scenes or styles.

Quantitative Evaluation. As shown in Table 3, Stylos achieves strong and stable consistency scores, ranking the first across all consistency metrics and all four scenes. This indicates that Stylos provides markedly improved cross-view stylization consistency. Furthermore, Table 4 shows that Stylos attains either the best or second-best artistic metric values, as rated by ArtScore and ArtFID, on all four Tanks and Temples scenes (Knapitsch et al., 2017), while maintaining the fastest stylization speed. Overall, Stylos demonstrates a favorable balance between visual quality, consistency, and efficiency, suggesting its potential for practical real-time 3D stylization.

Table 3: For consistency comparisons, we report short-range and long-range LPIPS↓ and RMSE↓ on the four scenes from the Tanks & Temples dataset. We clarify experiment details in A.4. In the following tables, the best results are **highlighted** and the second best results are underlined. Each stylization method category is visualized with a distinct color. The proposed ***Stylos*** demonstrates improved short-range and long-range consistency scores across the four scenes.

Method	Train		Truck		M60		Garden	
	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓
Short-range consistency								
StyleGaussian 2024	0.033	0.038	0.031	0.034	0.038	0.037	0.069	0.061
G-Style 2024	0.042	0.052	0.032	0.035	0.038	0.034	0.066	0.070
SGSST 2025	0.038	0.047	0.039	0.047	0.044	0.049	0.084	0.090
Styl3R 2025b	–	–	0.061	0.036	0.066	0.040	0.105	0.067
Stylos (ours)	**0.030**	**0.026**	**0.028**	**0.021**	**0.035**	**0.024**	**0.047**	**0.044**
Long-range consistency								
StyleGaussian 2024	0.067	0.072	0.086	0.077	0.091	0.091	0.177	0.141
G-Style 2024	0.098	0.120	0.095	0.093	0.104	0.095	0.180	0.175
SGSST 2025	0.087	0.108	0.119	0.120	0.130	0.128	0.221	0.222
Styl3R 2025b	–	–	0.116	0.100	0.147	0.143	0.146	0.145
Stylos (ours)	**0.051**	**0.056**	**0.074**	**0.069**	**0.083**	**0.082**	**0.139**	**0.134**

StylizedGS (Zhang et al., 2025) is not included in quantitative comparisons due to its multiple failure cases observed on our test styles. Nevertheless, its quantitative results are reported in A.4 Table 5 and Table 6 for readers' reference. Reproduced results are available at https://github.com/HanzhouLiu/Stylos.

Table 4: Stylization quality, as measured by ArtScore and ArtFID (abbreviated as Score and FID respectively), and stylization time comparisons with recent 3D stylization models. ***Stylos*** achieves consistently favorable metric scores across the four scenes. Additionally, we follow StyleID (Chung et al., 2024) and calculate additional metrics as reported in A.4 Table 5 and Table 6.

Method	Train		Truck		M60		Garden		Time↓
	Score↑	FID↓	Score↑	FID↓	Score↑	FID↓	Score↑	FID↓	
StyleGaussian 2024	0.78	52.79	5.76	44.93	8.63	47.48	**9.38**	41.14	165 m[*]
G-Style 2024	**9.52**	**23.24**	9.67	**22.15**	**9.73**	**22.36**	8.98	**25.76**	14.7 m[*]
SGSST 2025	1.84	38.24	5.34	32.34	5.26	38.73	4.89	33.54	35.2 m[*]
Styl3R 2025b	–	–	2.94	34.11	2.96	29.86	4.09	38.28	0.16 s[†]
Stylos (ours)	9.50	26.40	**9.70**	28.71	9.37	27.44	9.34	28.06	**0.05** s[†]

[*] For methods with per-scene fitting required, the training phase accounts towards stylization time (see A.4).
[†] As to single-forward 3D stylization approaches, training is not considered for stylization time.

Qualitative Evaluation. Taking the truck scene with the *desert-town* style (yellow foreground objects with blue backgrounds) in Fig. 5 as an example, we observe that nearly all existing approaches could reasonably produce structured results. However, their differences are discussed as follows. The zero-shot method StyleGaussian fail to generate color-consistent scenes, several portions of the truck retain undesired color blocks. Per-scene and per-style fitting approaches such as G-Style, StylizedGS, and SGSST often fail to achieve complete style transfer, the truck door remaining predominantly blue instead of adopting the target yellow hue. The other feedforward solution Styl3R similarly struggles to propagate the correct yellow color throughout the truck. In contrast, Stylos successfully renders the truck in coherent yellow while preserving clean geometry, and correctly assigns blue tones to the distant background regions, producing an appearance that closely resembles the intended desert-town aesthetic. These observations underscore the strength of Stylos in achieving both faithful style expression and robust 3D structural preservation.

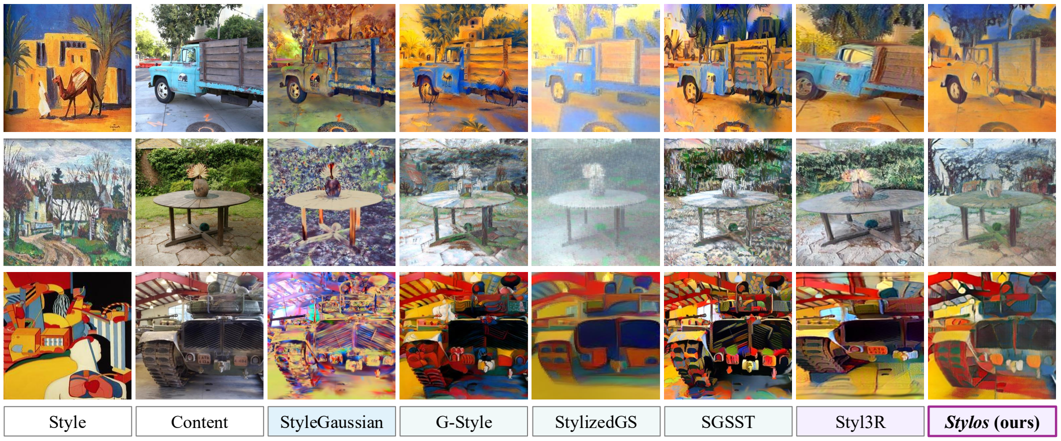


Figure 5: Visual comparisons between ***Stylos*** and recent 3D stylization approaches. ***Stylos*** successfully transfers diverse artistic styles to the scenes while preserving fine structural details.

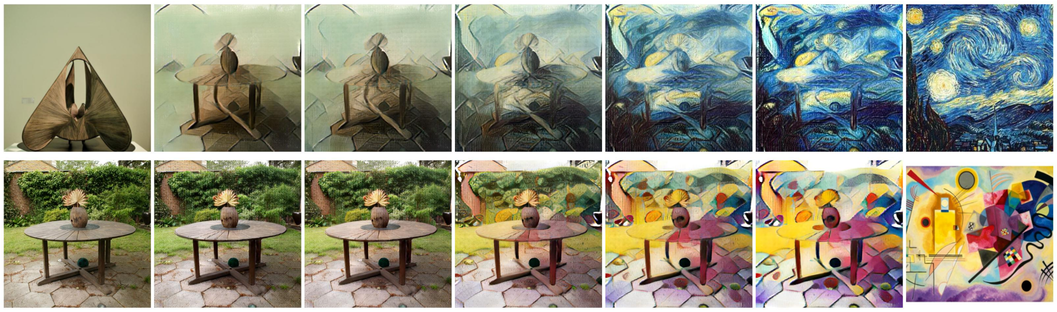


Figure 6: Evaluation of multi-style blending and controllable stylization. Top: interpolation between two style embeddings. Bottom: interpolation between the content embedding and a style embedding.

4.3 Evaluation of Multi-Style Blending and Stylization Control

We validate the model's controllable stylization capability through two interpolation experiments. First, by linearly interpolating between the embeddings of two distinct style images, we observe smooth, coherent transitions that demonstrate the model's natural support for multi-style blending (see Fig. 6). Second, because the model is trained to reconstruct the original appearance when the style image matches the content image, we interpolate between the reconstruction (content) embedding and a stylized embedding Fig. 6. This produces a continuous spectrum of outputs with gradually increasing stylization strength, confirming that our approach enables fine-grained, post-inference control over the content–style trade-off. Together, these results show that our method supports both multi-style fusion and adjustable stylization without any additional optimization.

5 Conclusion

In this work, we propose Stylos, a single-forward-pass framework for multi-view 3D stylization. By keeping geometry predictions on the self-attentive backbone path and conditioning color via global cross-attention, together with a voxel-space 3D style loss that aggregates multi-view features, Stylos achieves geometry-aware and view-consistent stylization. Our ablations show that the global-only cross-attention between style and content better preserves geometric details than alternative coupling strategies. Extensive experiments across category-level and large-scale scene datasets demonstrate the generalization ability of our approach, achieving aesthetically pleasing stylization with strong cross-view consistency. In the future, we aim to scale Stylos to support higher-resolution inputs while improving efficiency, paving the way for practical 3D content creation.

6 Acknowledgement

This research used both the DeltaAI advanced computing and data resource, which is supported by the National Science Foundation (award OAC 2320345) and the State of Illinois, and the Delta advanced computing and data resource which is supported by the National Science Foundation (award OAC 2005572) and the State of Illinois. Delta and DeltaAI are joint efforts of the University of Illinois Urbana-Champaign and its National Center for Supercomputing Applications. The allocations on Delta and DeltaAI [allocation number CIS250529] were made by the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

7 Reproducibility Statement

We have taken several steps to ensure the reproducibility of our work. The main paper describes all model architectures in Sec. 3. We provide detailed pseudo-code for the proposed 3D style loss function in Algo. 1, and will release the full implementation upon acceptance. All datasets used are publicly available, as described in Sec. 4. These resources should allow researchers to reproduce and extend our findings. In Appendix A.4, we clarify the training and inference details of the state-of-the-art methods for reproducible comparisons. In addition, we provide extensive qualitative results in Appendix A.6, and analyze efficiency trends with respect to the number of views per batch in Appendix A.2, where efficiency experiments are averaged over 100 iterations per setting.

References

Jie An, Siyu Huang, Yibing Song, Dejing Dou, Wei Liu, and Jiebo Luo. Artflow: Unbiased image style transfer via reversible neural flows. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 862–871, 2021.

Xu Cao, Weimin Wang, Katashi Nagao, and Ryosuke Nakamura. Psnet: A style transfer network for point cloud stylization on geometry and color. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer vision*, pp. 3337–3345, 2020.

Junyu Chen, Jie An, Hanjia Lyu, Christopher Kanan, and Jiebo Luo. Learning to evaluate the artness of ai-generated images. *IEEE Transactions on Multimedia*, 26:10731–10740, 2024.

Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pp. 1475–1484, 2022.

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, 2024.

Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11326–11336, 2022.

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

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

Camille Gontier, Jakob Jordan, and Mihai A Petrovici. Delaunay: A dataset of abstract art for psychophysical and machine learning research. In *International Conference on Soft Computing and Pattern Recognition*, pp. 134–143. Springer, 2023.

Jie Guo, Mengtian Li, Zijing Zong, Yuntao Liu, Jingwu He, Yanwen Guo, and Ling-Qi Yan. Volumetric appearance stylization with stylizing kernel prediction network. *ACM Trans. Graph.*, 40 (4):162–1, 2021.

Agrim Gupta, Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Characterizing and improving stability in neural style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 4067–4076, 2017.

Lukas Höllein, Justin Johnson, and Matthias Nießner. Stylemesh: Style transfer for indoor 3d scene reconstructions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6198–6208, 2022.

Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 13869–13878, 2021.

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

Yi-Hua Huang, Yue He, Yu-Jie Yuan, Yu-Kun Lai, and Lin Gao. 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*, pp. 18342–18352, 2022.

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 (TOG)*, 44(6):1–16, 2025.

Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3907–3916, 2018.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

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

Áron Samuel Kovács, Pedro Hermosilla, and Renata G Raidou. $\mathcal{G}$-style: Stylized gaussian splatting. In *Computer Graphics Forum*, volume 43, pp. e15259. Wiley Online Library, 2024.

Yanghao Li, Naiyan Wang, Jiaying Liu, and Xiaodi Hou. Demystifying neural style transfer. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 2230–2236, 2017.

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, 2024.

Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8338–8348, 2023.

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. 1–4. ACM, 2024.

Yiqun Mei, Jiacong Xu, and Vishal Patel. Regs: Reference-based controllable scene stylization with gaussian splatting. *Advances in Neural Information Processing Systems*, 37:4035–4061, 2024.

Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

Fangzhou Mu, Jian Wang, Yicheng Wu, and Yin Li. 3d photo stylization: Learning to generate stylized novel views from a single image. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16273–16282, 2022.

Thu Nguyen-Phuoc, Feng Liu, and Lei Xiao. Snerf: stylized neural implicit representations for 3d scenes. *ACM Transactions on Graphics*, 41:1–11, 07 2022. doi: 10.1145/3528223.3530107.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10901–10911, 2021.

Kerui Ren, Lihan Jiang, Tao Lu, Mulin Yu, Linning Xu, Zhangkai Ni, and Bo Dai. Octree-gs: Towards consistent real-time rendering with lod-structured 3d gaussians. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

Manuel Ruder, Alexey Dosovitskiy, and Thomas Brox. Artistic style transfer for videos. In *German conference on pattern recognition*, pp. 26–36. Springer, 2016.

Zachary Teed and Jia Deng. Deepv2d: Video to depth with differentiable structure from motion. In *8th International Conference on Learning Representations, ICLR 2020*. International Conference on Learning Representations, ICLR, 2020.

Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5038–5047, 2017.

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

Peng Wang, Xiang Liu, and Peidong Liu. Styl3r: Instant 3d stylized reconstruction for arbitrary scenes and styles. *arXiv preprint arXiv:2505.21060*, 2025b.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 20697–20709, 2024.

WikiArt. Wikiart: Visual art encyclopedia. `https://www.wikiart.org/`, 2025. Accessed: 2025-09.

Matthias Wright and Björn Ommer. Artfid: Quantitative evaluation of neural style transfer. In *DAGM German Conference on Pattern Recognition*, pp. 560–576. Springer, 2022.

Xin-Yi Yu, Jun-Xin Yu, Li-Bo Zhou, Yan Wei, and Lin-Lin Ou. Instantstylegaussian: Efficient art style transfer with 3d gaussian splatting. *arXiv preprint arXiv:2408.04249*, 2024.

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.

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, 2018.

Huizhong Zhou, Benjamin Ummenhofer, and Thomas Brox. Deeptam: Deep tracking and mapping. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 822–838, 2018.

A Appendix

A.1 Implementation

Implementation Details. For the geometry branch, we utilize VGGT as the backbone, comprising $L = 24$ Alternating-Attention Transformer layers. Both the geometry transformer and the depth DPT head are initialized using pretrained VGGT weights, while all remaining layers are initialized randomly. The style aggregator similarly employs 24 Transformer layers, configured with content-global self-attention followed by content-style cross-attention. For voxelization, each output of Geo Head associates a confidence value, which is converted into a normalized intra-voxel weight via softmax over Gaussians within the same voxel; these weights determine each Gaussian's contribution when aggregating voxel-level properties (opacity or color). The Gaussian Adapter maps the Geometry head and Style head predicted vector into explicit 3D Gaussian parameters. The module first splits the predicted vector into (i) isotropic scales, (ii) rotation quaternions, and (iii) degree-d spherical harmonic coefficients. Finally, we construct full covariance matrices using the predicted scale and rotation via the standard 3DGS covariance formulation. This yields the final set of Gaussian parameters: means, covariances, SH harmonics, opacities, scales, and normalized rotations.

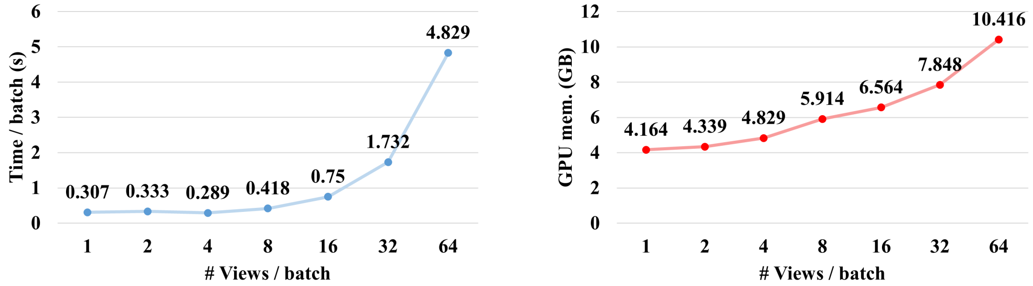
In the geometry training stage, the entire network is pre-trained. The geometry outputs, poses and depth, are guided by a frozen VGGT teacher. The DINO encoder is trained to capture color and style features. For stylization fine-tuning, we optimize only the style aggregator and the style head. We set the voxel size to 0.002 for differentiable voxelization. For both geometry pre-training and stylization fine-tuning, we train the model using the AdamW optimizer for 15k iterations. We employ a cosine learning-rate schedule with a peak learning rate of 2×10^{-4} and a 1k-iteration warmup.

Data & Augmentation. Following Anysplat (Jiang et al., 2025), we randomly sample between 2 and 20 frames per clip, maintaining a fixed total of 20 frames per GPU. The input resolution is constrained to 448 pixels on the longest side, with the aspect ratio randomized between 0.5 and 1.0. Intrinsic augmentation is applied via random center-cropping (77%–100% of the original size) and random horizontal flipping. During Geometry Pretraining, we encourage the DINO encoder to better learn color and style tokens by occasionally replacing the content image with a style image and by applying color jitter to perturb its appearance, while keeping the original image as ground truth to supervise appearance-change learning.

Loss Weights. Style weight $\lambda_{style} = 1.0$, content-reconstruction weight $\lambda_{cnt} = 0.1$, CLIP consistency weight $\lambda_{clip} = 1.0$, and total-variance regularization weight $\lambda_{tv} = 10.0$.

A.2 Efficiency Scaling

We further analyze the efficiency of Stylos by varying the number of views processed per batch. The results in Fig. 7 indicate a clear scaling pattern. For very small batches, such as 1–2 views, the average runtime per batch is low, around 0.3 s, but the measurements fluctuate heavily due



(a) Time per batch vs. # views per batch

(b) GPU memory usage vs. # views per batch

Figure 7: Scaling of inference time and GPU memory with the number of views per batch. Time is averaged over 100 iterations, and memory is reported as the peak allocation, without rendering cost.

to overheads and GPU scheduling. This makes single-view or extremely small-batch inference unstable, even though the raw latency is minimal. As the batch size increases, runtime grows roughly linearly, reaching 0.75 s at 16 views and 4.83 s at 64 views. Importantly, once the batch size exceeds 4–8 views, the runtime becomes much more stable, showing only minor variation across repeated runs. Memory usage shows a similarly consistent trend: GPU memory rises steadily from 4.16 GB at 1 view to 10.42 GB at 64 views. The growth is close to linear with respect to batch size, which makes resource requirements easy to estimate when scaling to larger workloads.

A.3 Ablation Analysis

Content-Style Coupling. Stylos consists of a style pathway which contributes to the color information for the predicted 3DGS representation, and a content pathway for the other 3DGS parameters. In the style pathway, we introduce a style aggregator with cross-attention to entangle the content and style tokens. Our training pipeline consists of two stages. The first stage focuses on 3D scene reconstruction, where a random content view is used as the style reference; Stage 2 performs actual stylization using a real artistic style image. When designing Stylos, it is therefore essential to verify that introducing content-style cross-attention in the style aggregator does not harm the reconstruction process.

So, we compare three content–style coupling mechanisms (global, frame, and both) by calculating PSNR, SSIM and LPIPS between the rendered multi-view rgbs and their corresponding groundthruth after Stage 1 training, using the first content view as the style reference. As shown in the Table 1, the global-only variant achieves the best reconstruction accuracy. This behavior is expected, because each view attends to the style tokens independently without interacting with the other views in frame attention. Treating a single view as the style reference for all views forces each view to match its patch-level representation to the same reference individually, which distorts the rendered images. Fig. 2 shows that the frame attention significantly blurs the pizza box and toppings.

Image-, Scene-, and Voxel-level Style Losses. In this section, we provide additional visual comparisons showing that the proposed scene and 3D style losses significantly outperform the baseline image style loss. Specifically, we observe that the scene- and 3D-level losses produce cleaner and more coherent textures, such as the wooden floor in Fig. 8. As shown in Fig. 9, the 3D style loss is also able to stylize a larger portion of the table surface compared with the image loss. In Fig. 10, the seams on the tablecloth are well preserved under the 3D loss, while they become partially distorted when using the image or scene losses. We will release our implementations of all three style losses to facilitate further exploration by the community.

A.4 Reproducible Experiments

Reproduce State-of-the-art Models. For StyleGaussian (Liu et al., 2024), SGSST (Galerne et al., 2025), StylizedGS (Zhang et al., 2025), and GStyle (Kovács et al., 2024), we use their officially released codebases and pretrained weights. We either run inference directly with the provided models or follow the authors' instructions to train the models when necessary. All of them are evaluated on GH200 GPUs for fair comparisons.

We notice that StylizedGS (Zhang et al., 2025) fail in serveral style cases, producing stylized images with pure colors with barely visible geomtry details. It significantly improves its consistency scores in our experiments. For fair comparisons, we exclude these failure cases when calculating consistency and ArtScore for StylizedGS, as they could lead to very small values.

Stylization Quality Comparisons. In Table 3 and 4, * denotes that failure stylization cases for StylizedGS* (Zhang et al., 2025) are excluded for fair comparisons. Please refer to our released codes for additional visual examples. Table 5 and Table 6 are provided as supplementary quantitative results to Table 4.

Stylization Time Comparisons. We run 48 samples as a hardware warm-up phase, and calculate the average inference speeds over the following 112 samples. For fair comparisons, all methods are evaluated on the same hardware, using an NVIDIA GH200 GPU with identical configurations. The reported stylization time of the per-scene fitting required method considers per-scene training time

Table 5: Following StyleID (Chung et al., 2024), we additionally report sylization quality metrics, FID, LPIPS, LPIPS-gray, CFSD, color matching loss (HistoGAN loos), as supplementary to Table 4.

Method	Train					Truck				
	FID↓	LPIPS↓	Gray↓	CFSD↓	CM Loss↓	FID↓	LPIPS↓	Gray↓	CFSD↓	CM Loss↓
StyleGaussian	34.59	0.483	0.377	0.190	0.418	28.53	0.522	0.367	0.131	0.382
G-Style	**14.80**	**0.471**	**0.364**	**0.165**	**0.185**	**13.65**	**0.512**	**0.353**	0.087	**0.176**
StylizedGS	22.90	0.707	0.648	0.216	0.396	22.65	0.779	0.713	**0.084**	0.417
SGSST	24.15	0.520	0.409	0.220	0.257	19.50	0.577	0.445	0.177	0.196
Styl3R	19.77	0.670	0.598	0.244	0.364	19.28	0.682	0.575	0.102	0.350
Stylos (ours)	15.30	0.620	0.529	0.223	0.241	16.67	0.625	0.471	**0.084**	0.237

Table 6: Following StyleID (Chung et al., 2024), we additionally report sylization quality metrics, FID, LPIPS, LPIPS-gray, CFSD, color matching loss (HistoGAN loos), as supplementary to Table 4.

Method	M60					Garden				
	FID↓	LPIPS↓	Gray↓	CFSD↓	CM Loss↓	FID↓	LPIPS↓	Gray↓	CFSD↓	CM Loss↓
StyleGaussian	30.54	0.506	0.413	0.138	0.467	25.23	0.569	0.454	0.189	0.480
G-Style	**13.93**	**0.498**	**0.395**	**0.092**	**0.208**	**16.17**	**0.500**	**0.364**	0.103	**0.179**
StylizedGS	28.33	0.815	0.728	0.102	0.443	33.73	0.876	0.827	**0.074**	0.570
SGSST	23.95	0.552	0.458	0.204	0.264	20.93	0.529	0.415	0.233	0.228
Styl3R	17.14	0.646	0.573	0.124	0.314	22.38	0.637	0.556	0.097	0.335
Stylos (ours)	16.61	0.558	0.457	0.098	0.252	16.26	0.625	0.509	0.080	0.242

besides the rendering time. We strictly follow the original authors' training setups and inference pipelines. All 3DGS-based methods (Liu et al., 2024; Galerne et al., 2025; Zhang et al., 2025; Kovács et al., 2024) are evaluated on the full-resolution images, whereas single-forward methods, including Styl3R (Wang et al., 2025b) and Stylos, are evaluated at their preset input resolutions, i.e., 256×256 and 448×448, respectively.

A.5 Limitations

Stylos tends to underperform on scenes dominated by high-frequency or highly cluttered structures, such as dense foliage, thin branches, or wire-like elements, where preserving fine geometric detail is particularly challenging. The method also struggles with style categories that introduce strong global lighting shifts or extreme color palettes, which can lead to over-saturation or loss of subtle appearance cues. In addition, performance decreases as the number of input views increases. This degradation primarily stems from limitations of the VGGT backbone: as view count grows, its underlying geometric reconstruction becomes less stable, which in turn negatively affects the quality of the stylized outputs. More specifically, the failure cases can be grouped into three categories: (1) Failures caused by inaccurate scene reconstruction, such as missing geometric details or incorrect region colors; (2) Failures caused by generating colors that are absent from the original style images. (3) Failures caused by introducing blurry or over-smoothed style colors.

A.6 Additional Visual Results

Figure 14-15 exhibit all the 50 test style images for quantitative comparisons. Figure 16-25 presents additional qualitative results of Stylos on diverse style–content pairs. Across different styles and object categories, Stylos produces visually pleasing stylizations that respect the input content geometry while transferring the target artistic appearance. These examples further demonstrate the versatility and robustness of our approach beyond the main results in the paper.

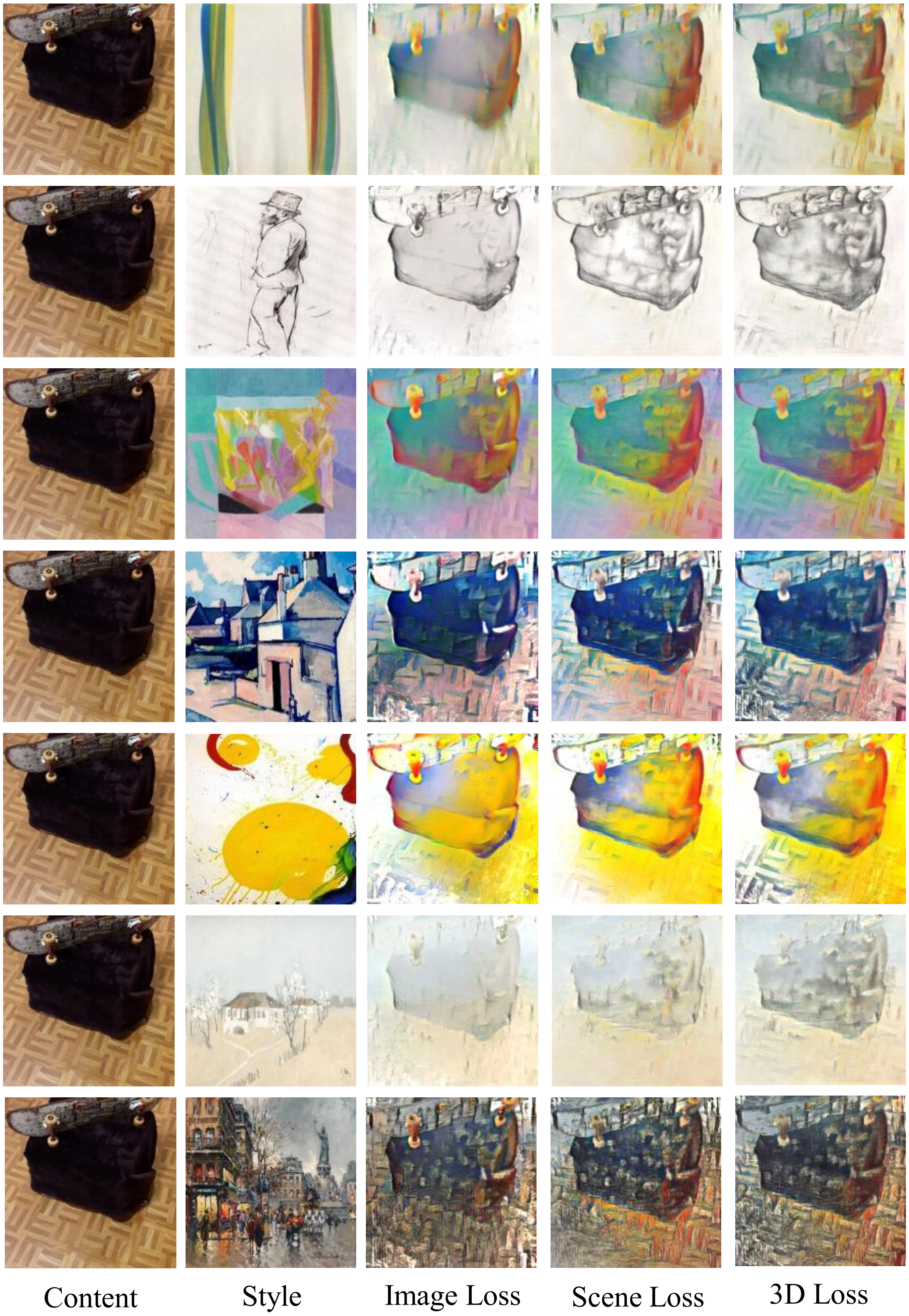


Figure 8: Visual comparisons of different style losses on the CO3D skateboard scene. It is clear that the 3D style loss provides a stronger sense of 3D geometry and cleaner wooden floor textures evaluated on varying style images.

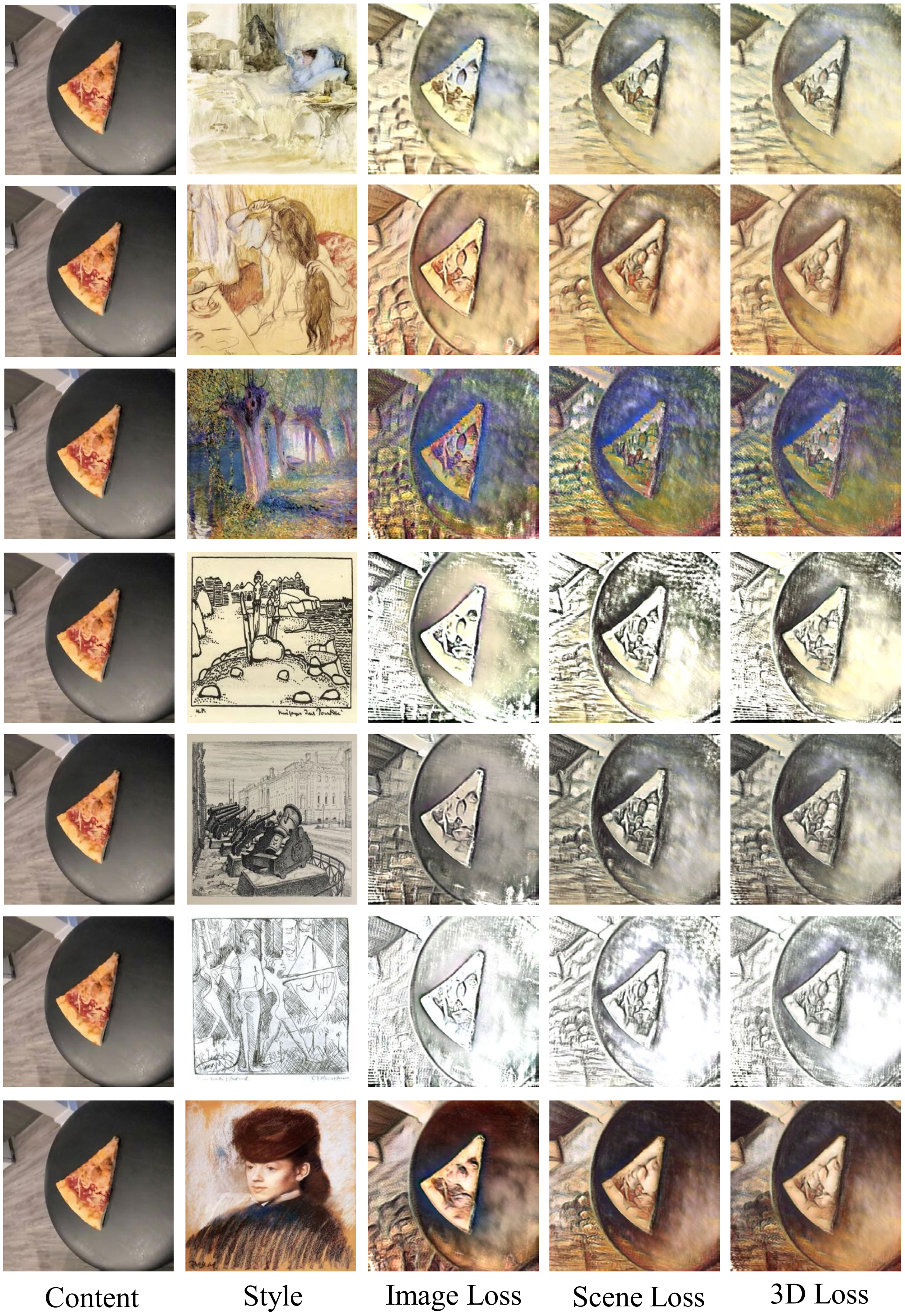


Figure 9: Visual comparisons of different style losses on the CO3D pizza scene. It is clear that the 3D style loss provides stronger artistic stylization to the table surface, while the image style loss often fails to impose the expected styles to cover the whole table surface.

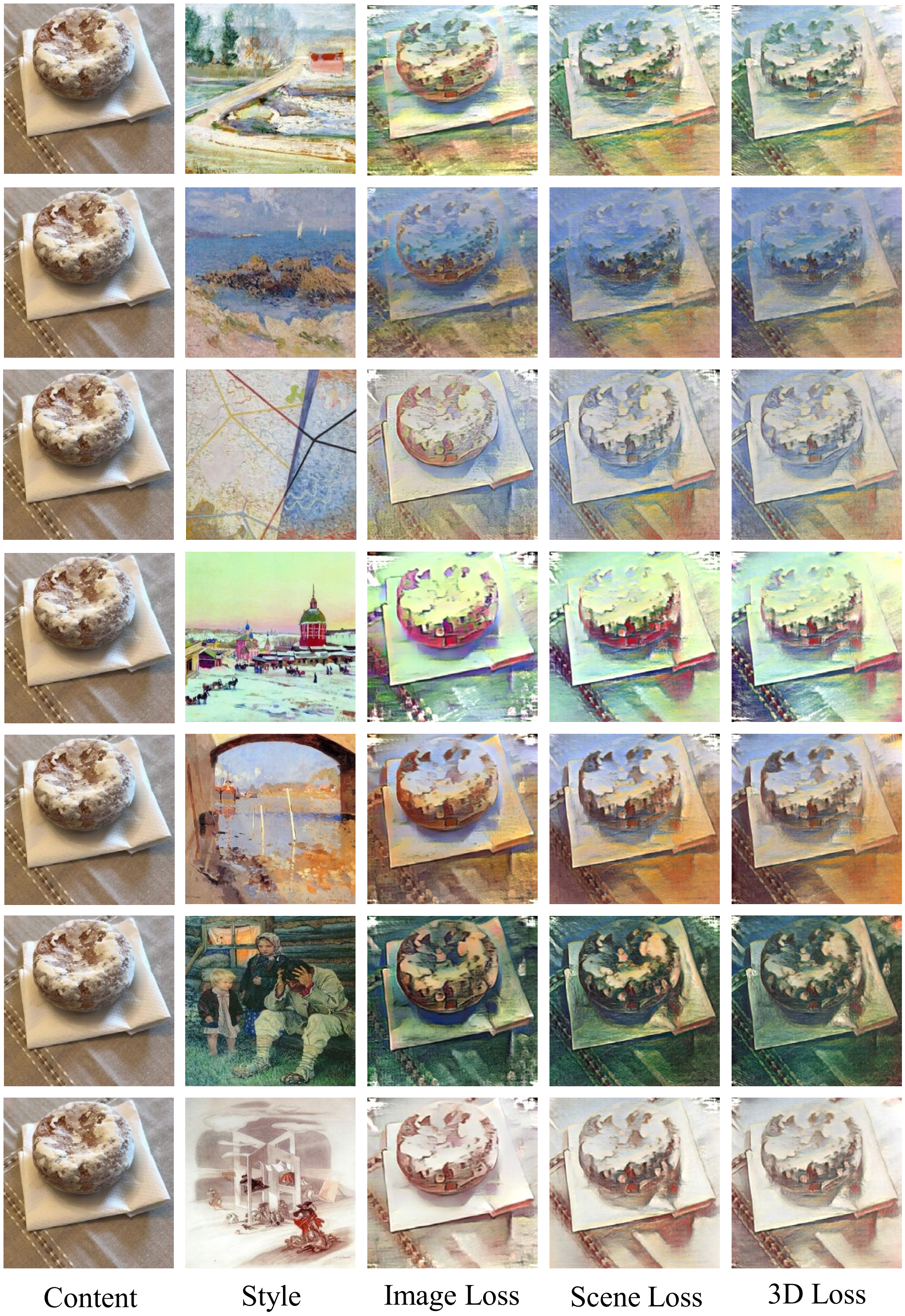
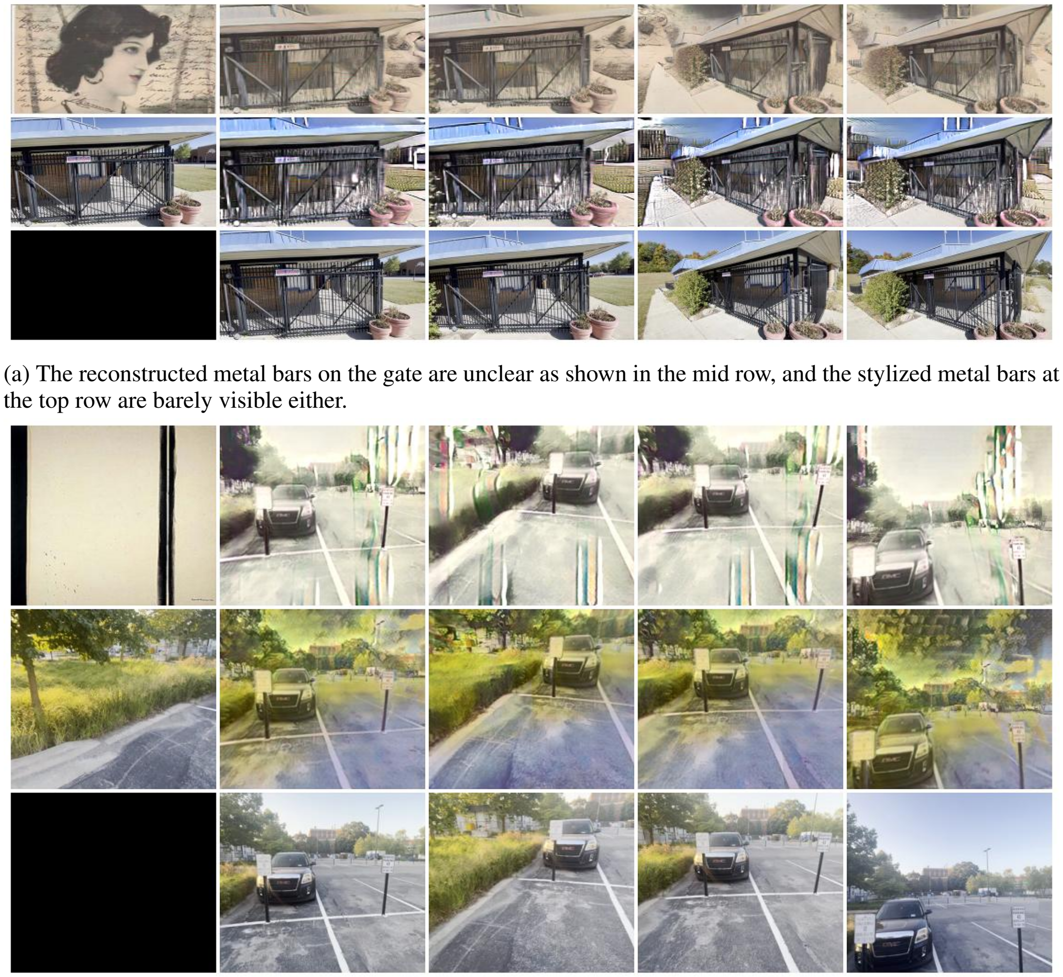


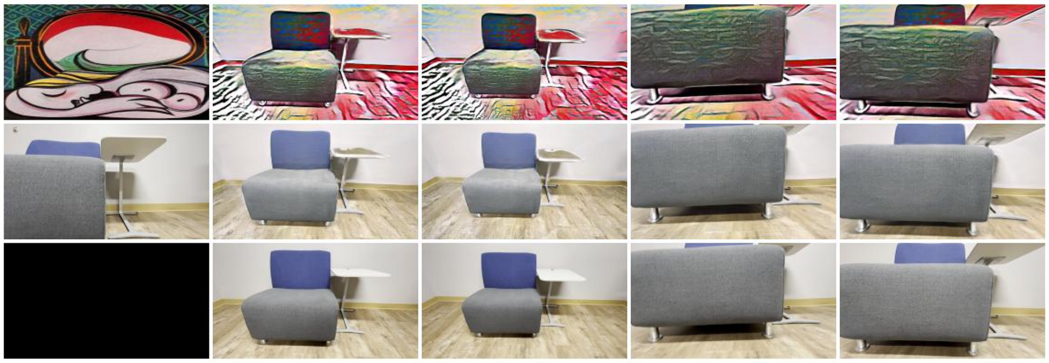
Figure 10: Visual comparisons of different style losses on the CO3D donut scene. It is clear that the 3D style loss provides faithful stylization to the table cover while preserving the seams on the tablecloth compared to both the image style loss and the scene style loss.



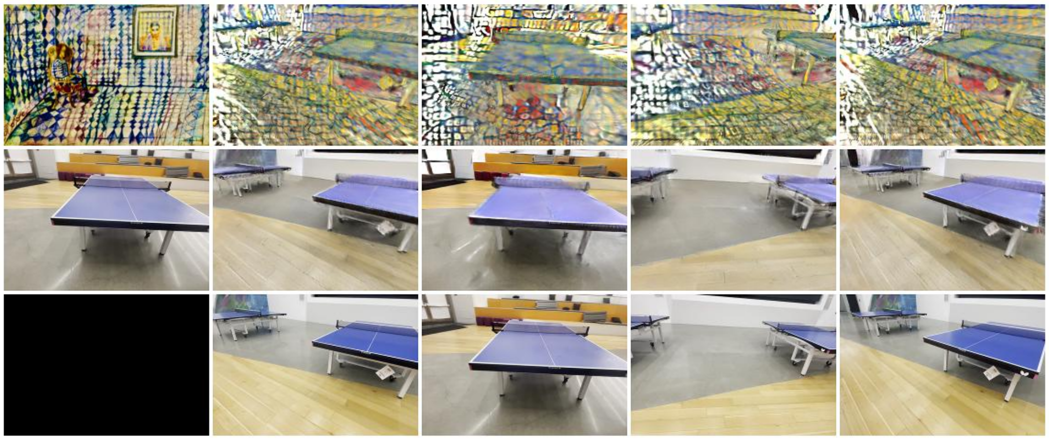
(a) The reconstructed metal bars on the gate are unclear as shown in the mid row, and the stylized metal bars at the top row are barely visible either.

(b) The green colors fail to correspond to the intended regions in the reconstructed scene (at the mid row), and the stylized multi-view images mistakenly interpret the green area as part of the target pattern, leading the model to apply stylization to it, (at the top row).

Figure 11: Failures caused by inaccurate scene reconstruction, such as missing geometric details or incorrect region colors. Top row: The input style image and the stylized multi-view results; mid row: a reference frame from the input contents and the reconstructed multi-view images; bottom row: the input contents.

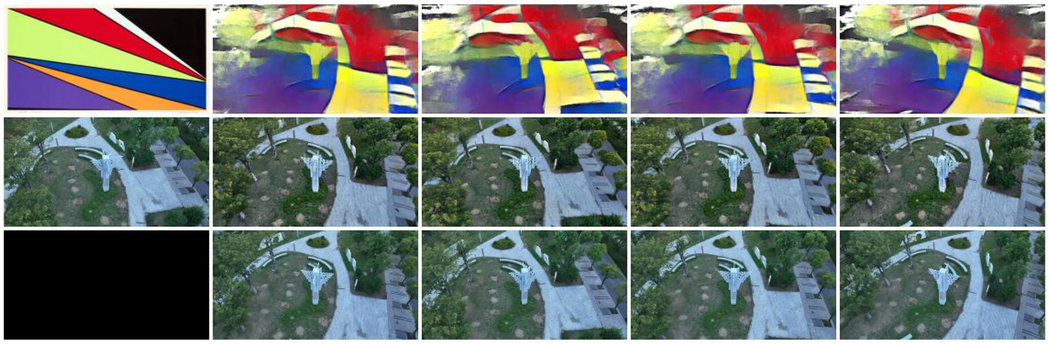


(a) The reconstructed scene is generally accurate. However, our model introduces a pure white region in the bottom-left area (at the top row), which does not appear in the original style image.



(b) The reconstructed scene is generally accurate. However, our model hallucinates a pure white region on the left side (at the top row), which is absent from the original style image.

Figure 12: Failures caused by generating colors that are absent from the original style images. Top row: The input style image and the stylized multi-view results; mid row: a reference frame from the input contents and the reconstructed multi-view images; bottom row: the input contents.



(a) Our model produces a blurred stylized region in the upper-left area of the results (as shown in the top row).

Figure 13: Failures caused by introducing blurry or over-smoothed style colors. Top row: The input style image and the stylized multi-view results; mid row: a reference frame from the input contents and the reconstructed multi-view images; bottom row: the input contents.

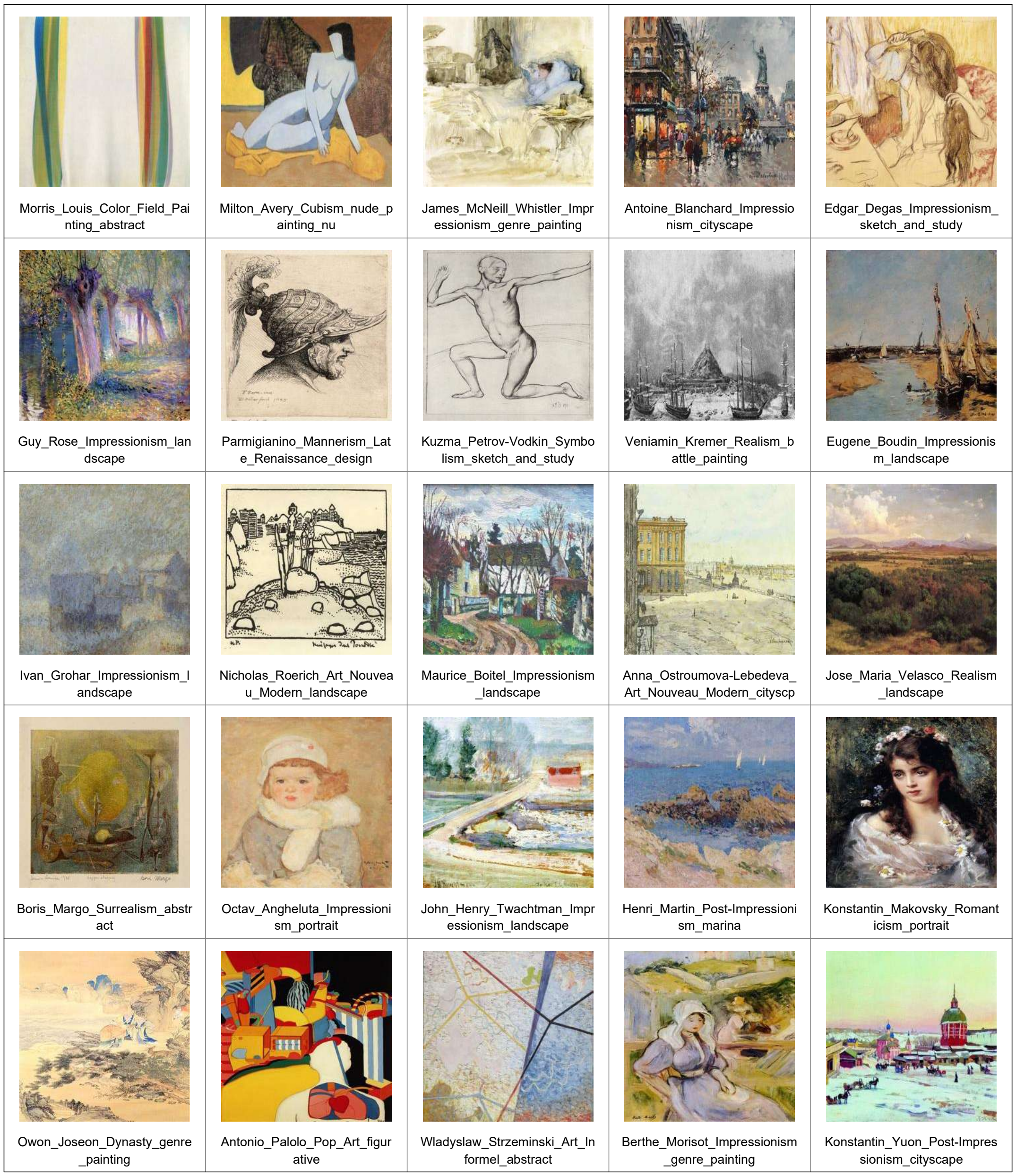

Morris_Louis_Color_Field_Painting_abstract
Milton_Avery_Cubism_nude_painting_nu
James_McNeill_Whistler_Impressionism_genre_painting
Antoine_Blanchard_Impressionism_cityscape
Edgar_Degas_Impressionism_sketch_and_study
Guy_Rose_Impressionism_landscape
Parmigianino_Mannerism_Late_Renaissance_design
Kuzma_Petrov-Vodkin_Symbolism_sketch_and_study
Veniamin_Kremer_Realism_battle_painting
Eugene_Boudin_Impressionism_landscape
Ivan_Grohar_Impressionism_landscape
Nicholas_Roerich_Art_Nouveau_Modern_landscape
Maurice_Boitel_Impressionism_landscape
Anna_Ostroumova-Lebedeva_Art_Nouveau_Modern_cityscp
Jose_Maria_Velasco_Realism_landscape
Boris_Margo_Surrealism_abstract
Octav_Angheluta_Impressionism_portrait
John_Henry_Twachtman_Impressionism_landscape
Henri_Martin_Post-Impressionism_marina
Konstantin_Makovsky_Romanticism_portrait
Owon_Joseon_Dynasty_genre_painting
Antonio_Palolo_Pop_Art_figurative
Wladyslaw_Strzeminski_Art_Informel_abstract
Berthe_Morisot_Impressionism_genre_painting
Konstantin_Yuon_Post-Impressionism_cityscape

Figure 14: Style images from WikiArt (2025) used for model evaluation in all quantitative comparisons, while not used for training Stylos.

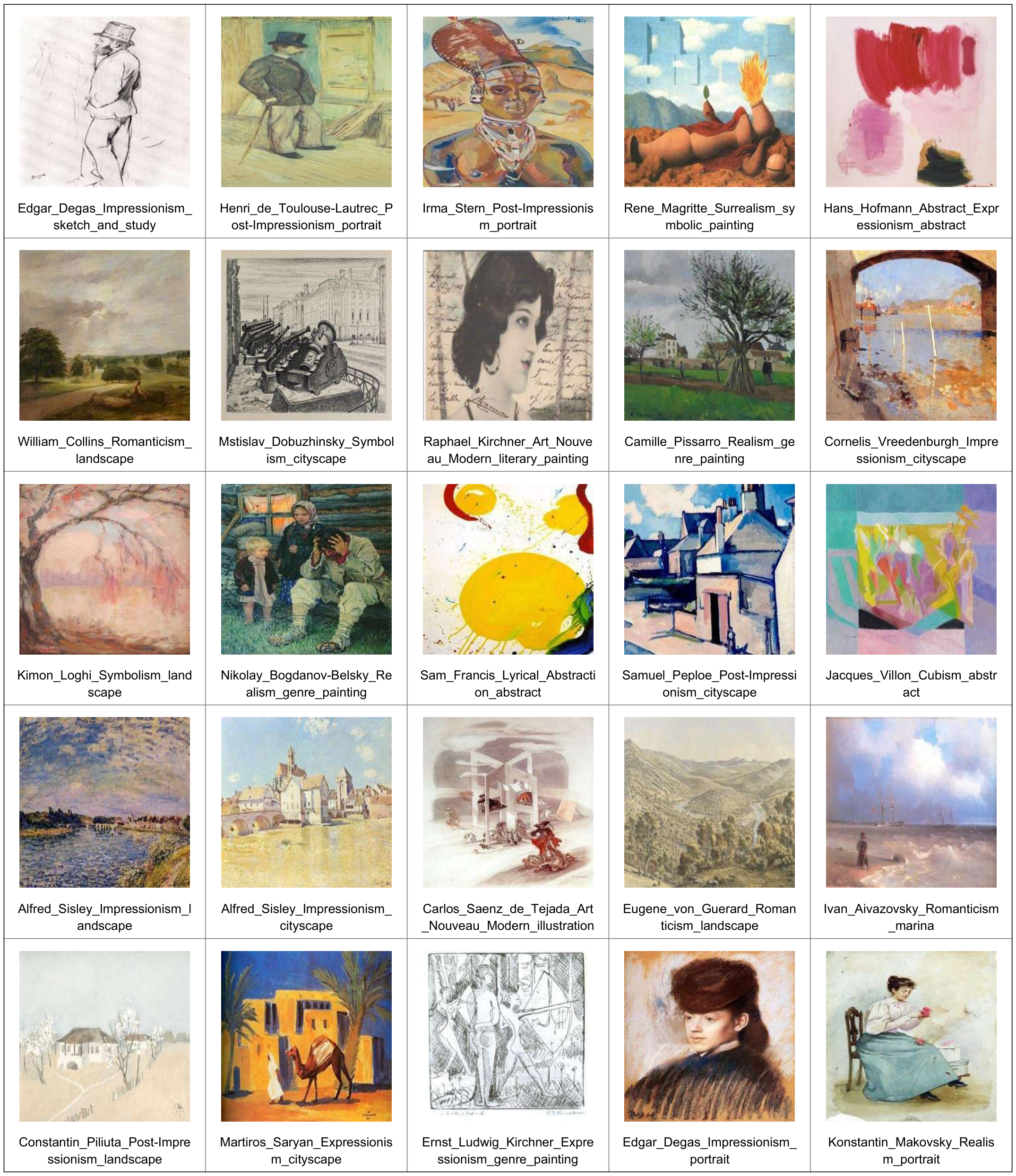

Edgar_Degas_Impressionism_sketch_and_study
Henri_de_Toulouse-Lautrec_Post-Impressionism_portrait
Irma_Stern_Post-Impressionism_portrait
Rene_Magritte_Surrealism_symbolic_painting
Hans_Hofmann_Abstract_Expressionism_abstract
William_Collins_Romanticism_landscape
Mstislav_Dobuzhinsky_Symbolism_cityscape
Raphael_Kirchner_Art_Nouveau_Modern_literary_painting
Camille_Pissarro_Realism_genre_painting
Cornelis_Vreedenburgh_Impressionism_cityscape
Kimon_Loghi_Symbolism_landscape
Nikolay_Bogdanov-Belsky_Realism_genre_painting
Sam_Francis_Lyrical_Abstraction_abstract
Samuel_Peploe_Post-Impressionism_cityscape
Jacques_Villon_Cubism_abstract
Alfred_Sisley_Impressionism_landscape
Alfred_Sisley_Impressionism_cityscape
Carlos_Saenz_de_Tejada_Art_Nouveau_Modern_illustration
Eugene_von_Guerard_Romanticism_landscape
Ivan_Aivazovsky_Romanticism_marina
Constantin_Piliuta_Post-Impressionism_landscape
Martiros_Saryan_Expressionism_cityscape
Ernst_Ludwig_Kirchner_Expressionism_genre_painting
Edgar_Degas_Impressionism_portrait
Konstantin_Makovsky_Realism_portrait

Figure 15: Style images from WikiArt (2025) used for model evaluation in all quantitative comparisons, while not used for training Stylos.

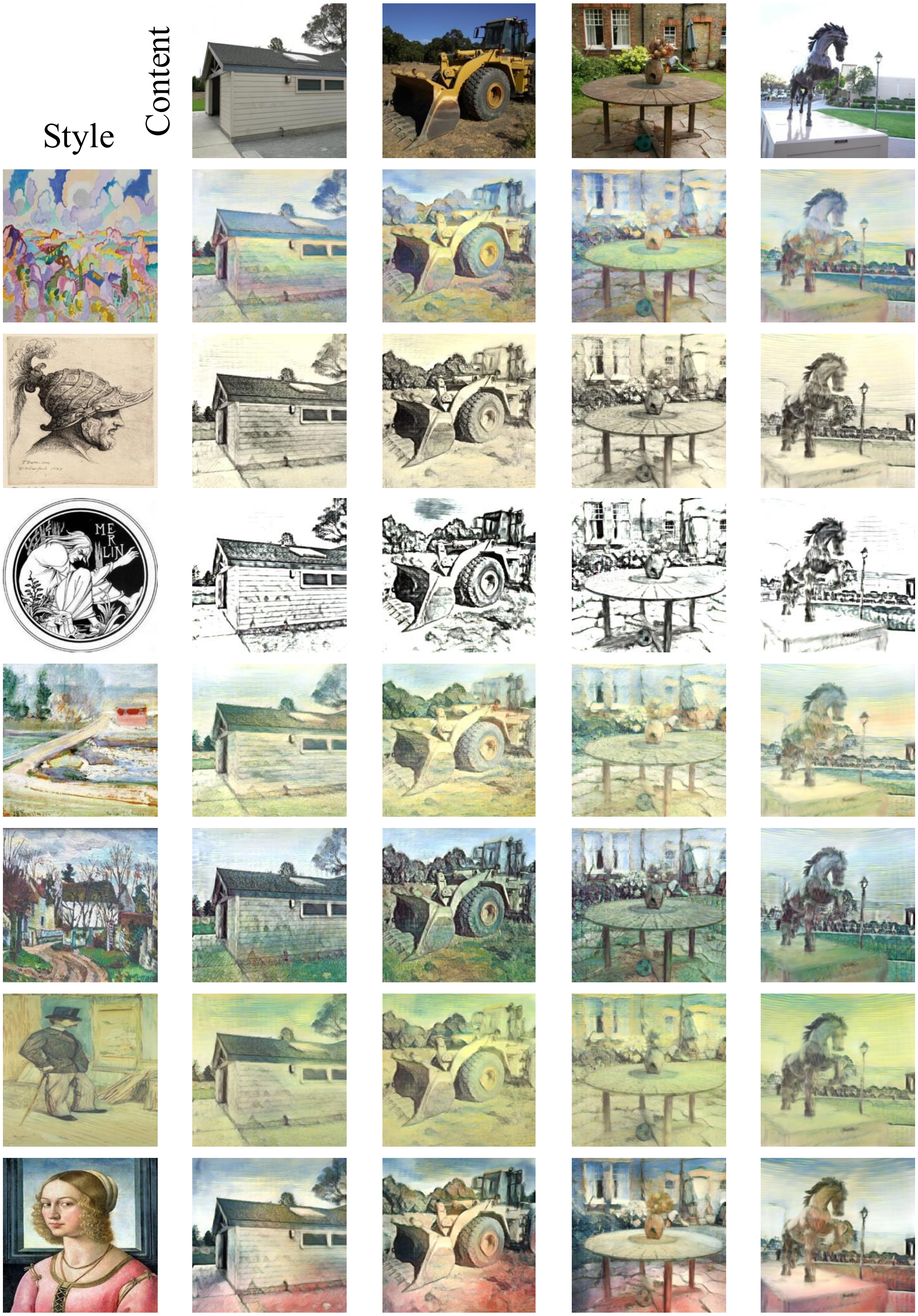


Figure 16: Additional visual results of Stylos on diverse style–content pairs on Tanks and Temples Dataset (Knapitsch et al., 2017). Stylos consistently produces visually pleasing stylizations that preserve content geometry while transferring the target style.

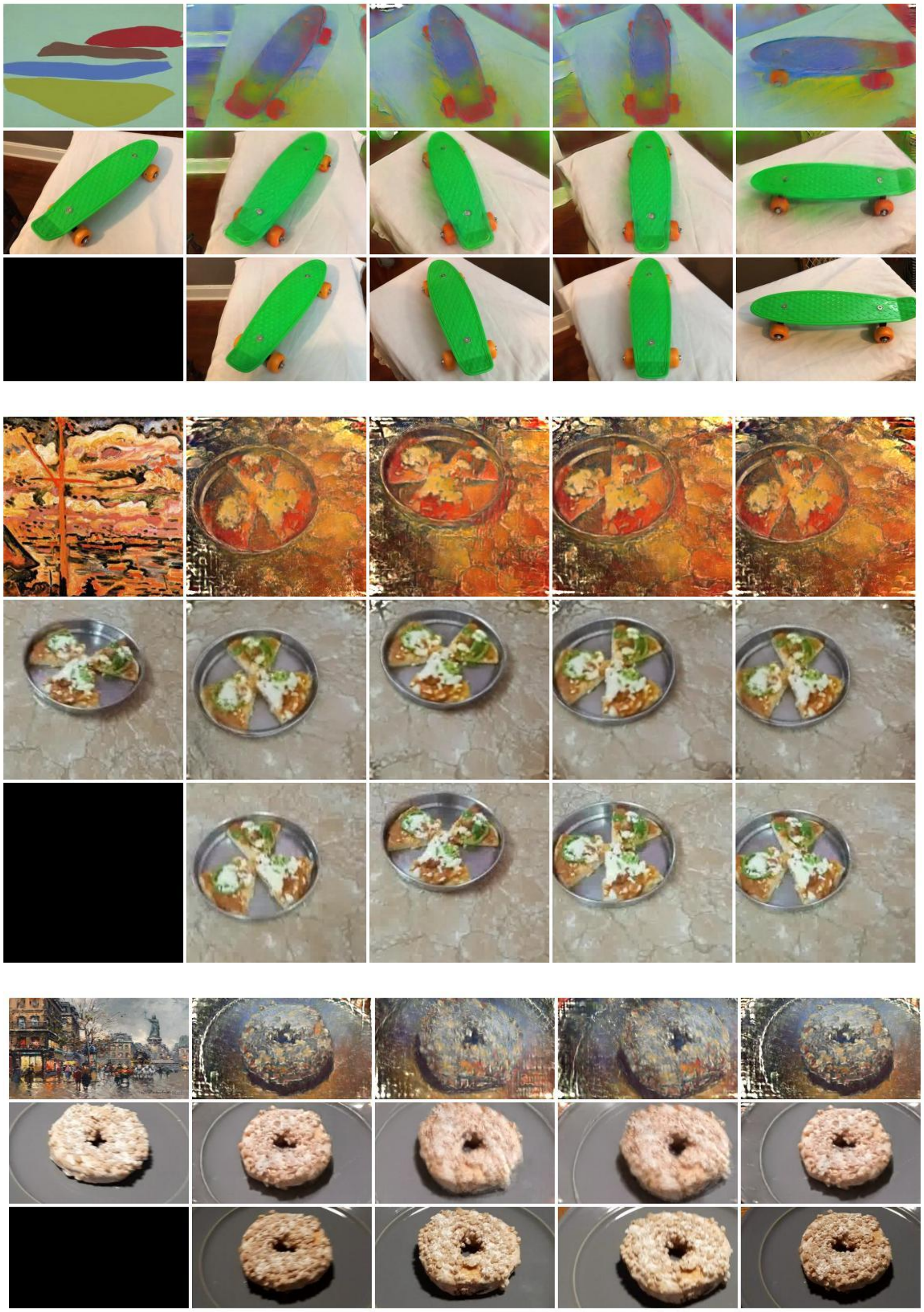


Figure 17: Additional visual results of Stylos on the CO3D dataset. The top left is the style image while the rest of the top row are the rendered images. The mid row are the rendered images conditioned on the mid left imamge. The bottom row is the content inputs.

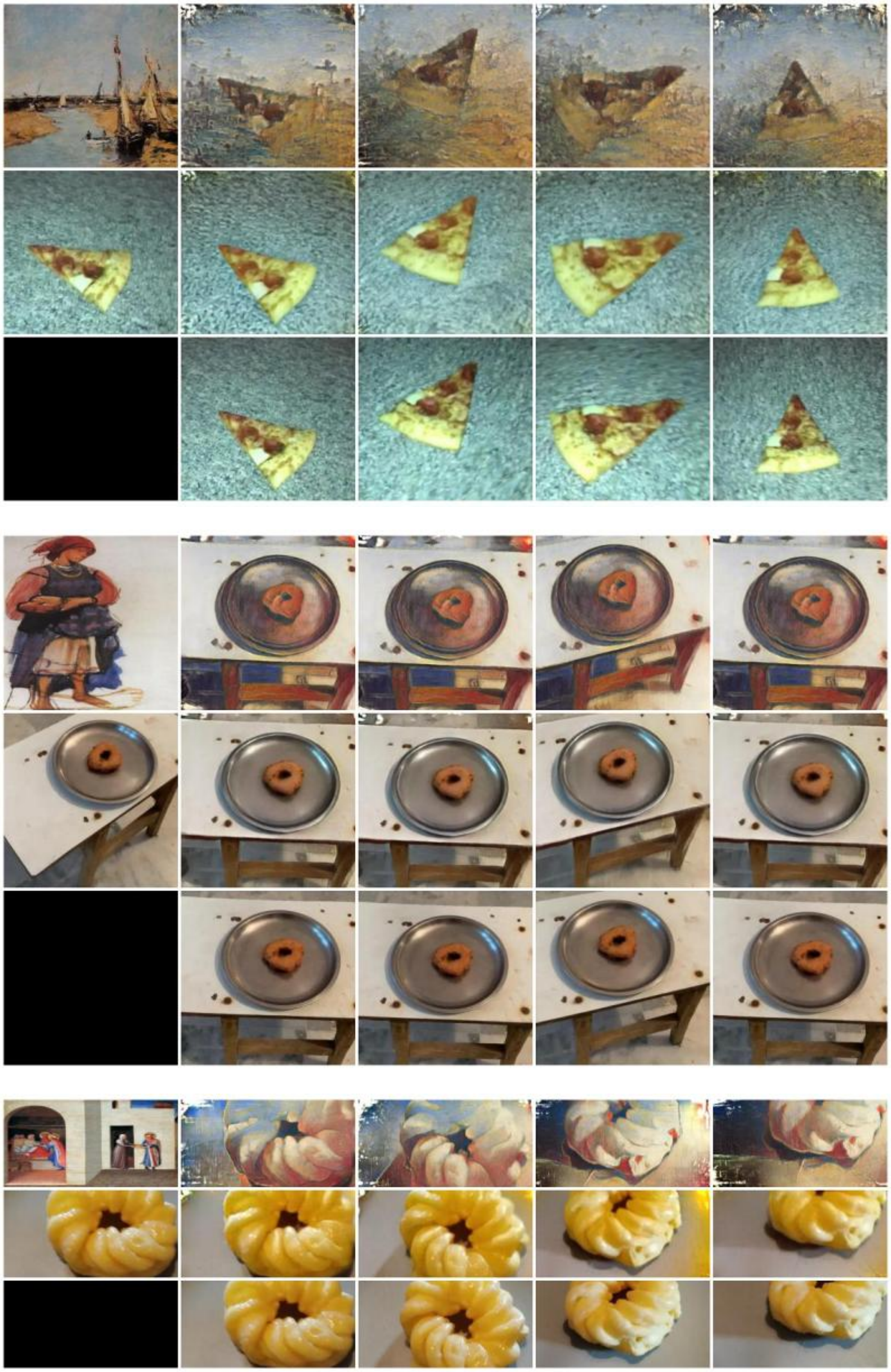


Figure 18: Additional visual results of Stylos on diverse style–content pairs.

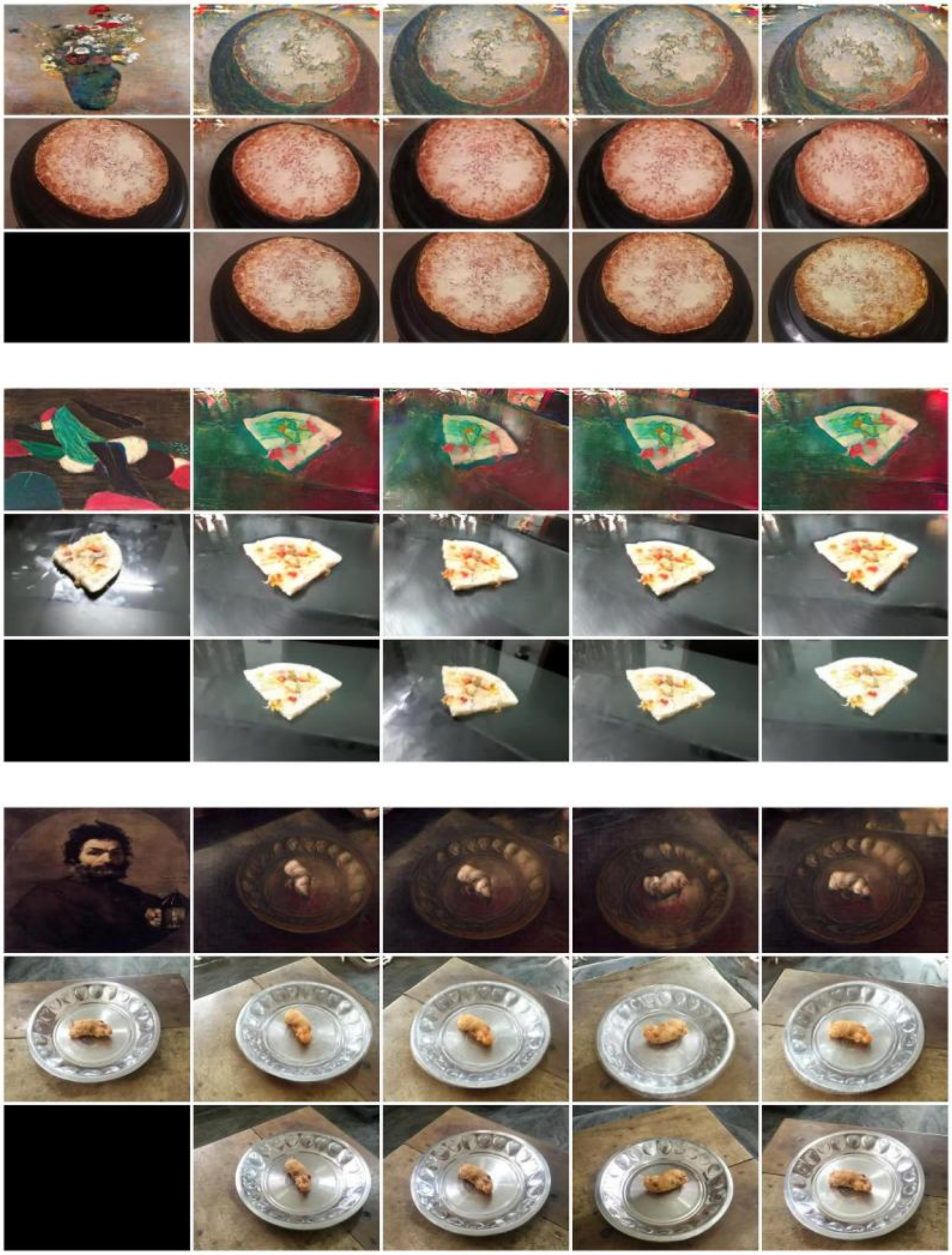


Figure 19: Additional visual results of Stylos on diverse style–content pairs.

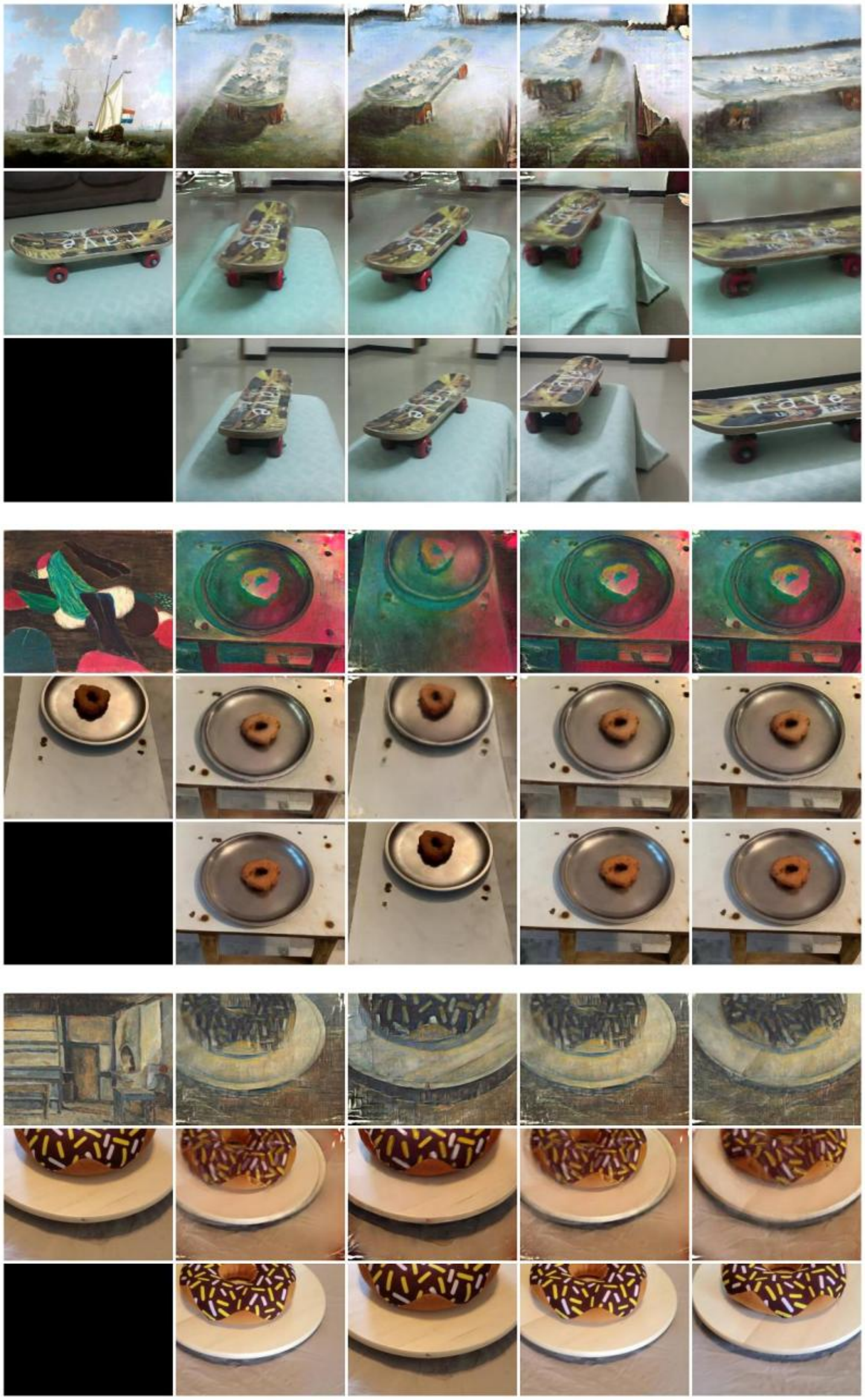


Figure 20: Additional visual results of Stylos on diverse style–content pairs.

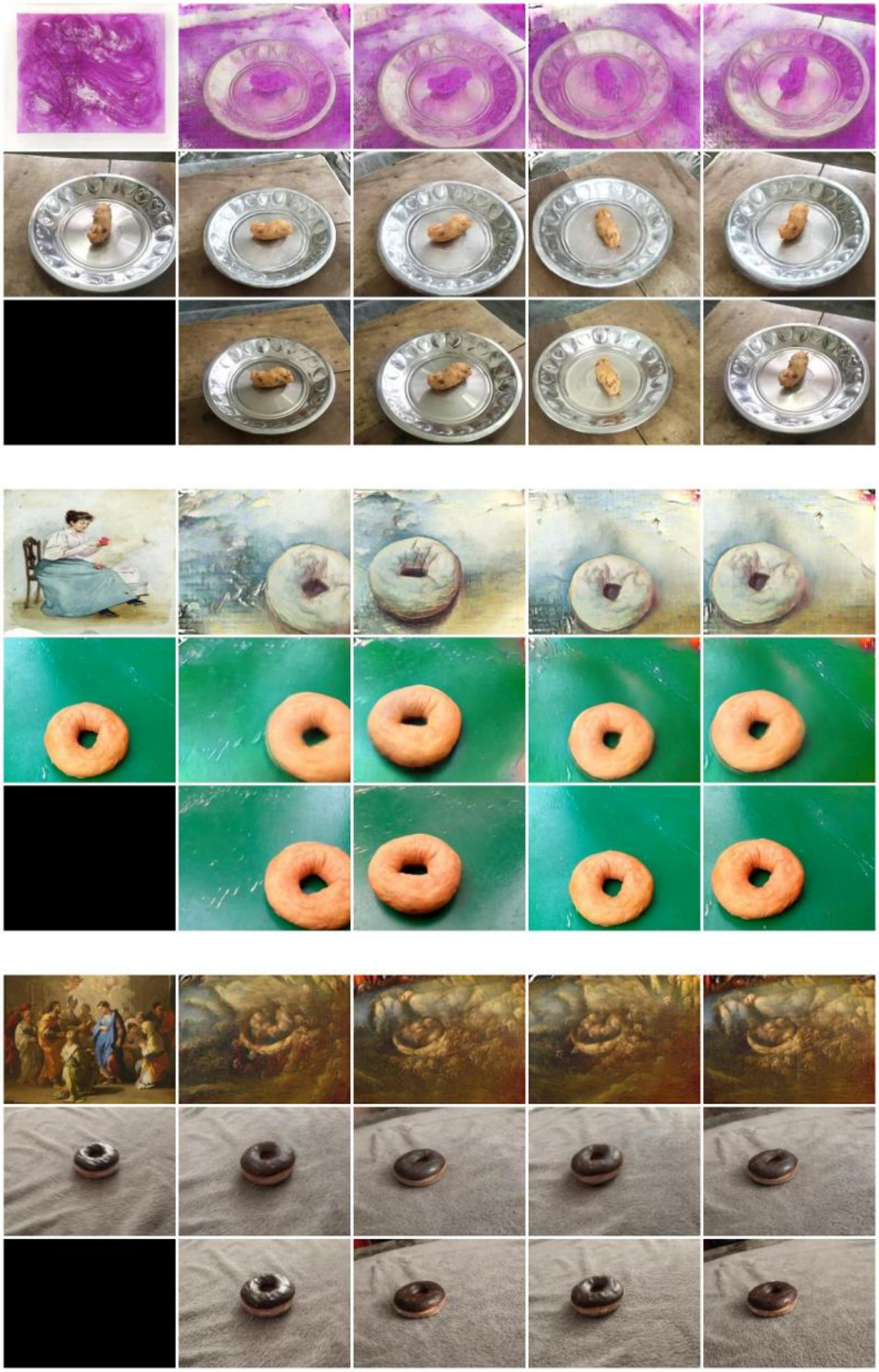


Figure 21: Additional visual results of Stylos on diverse style–content pairs.

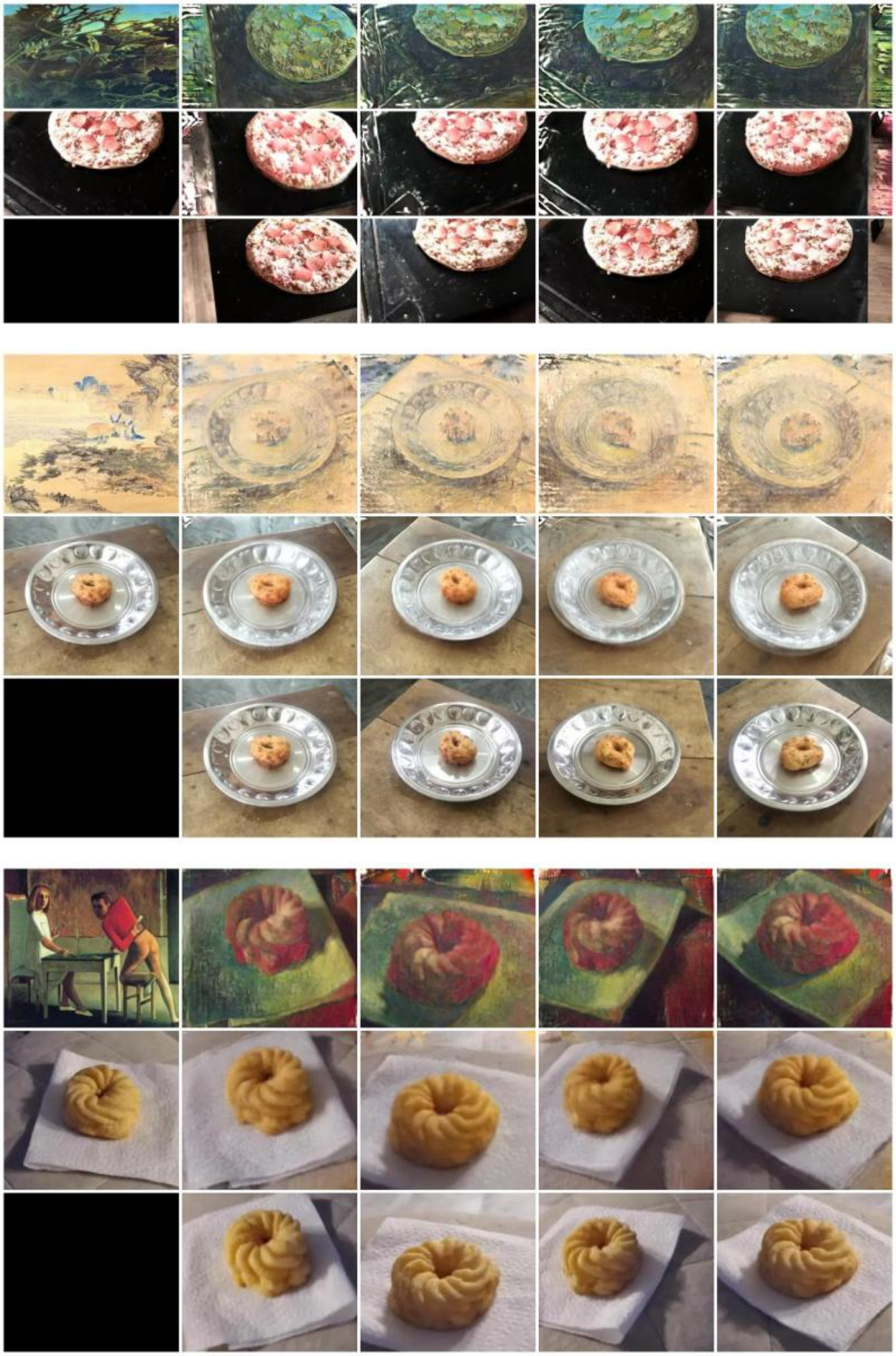


Figure 22: Additional visual results of Stylos on diverse style–content pairs.

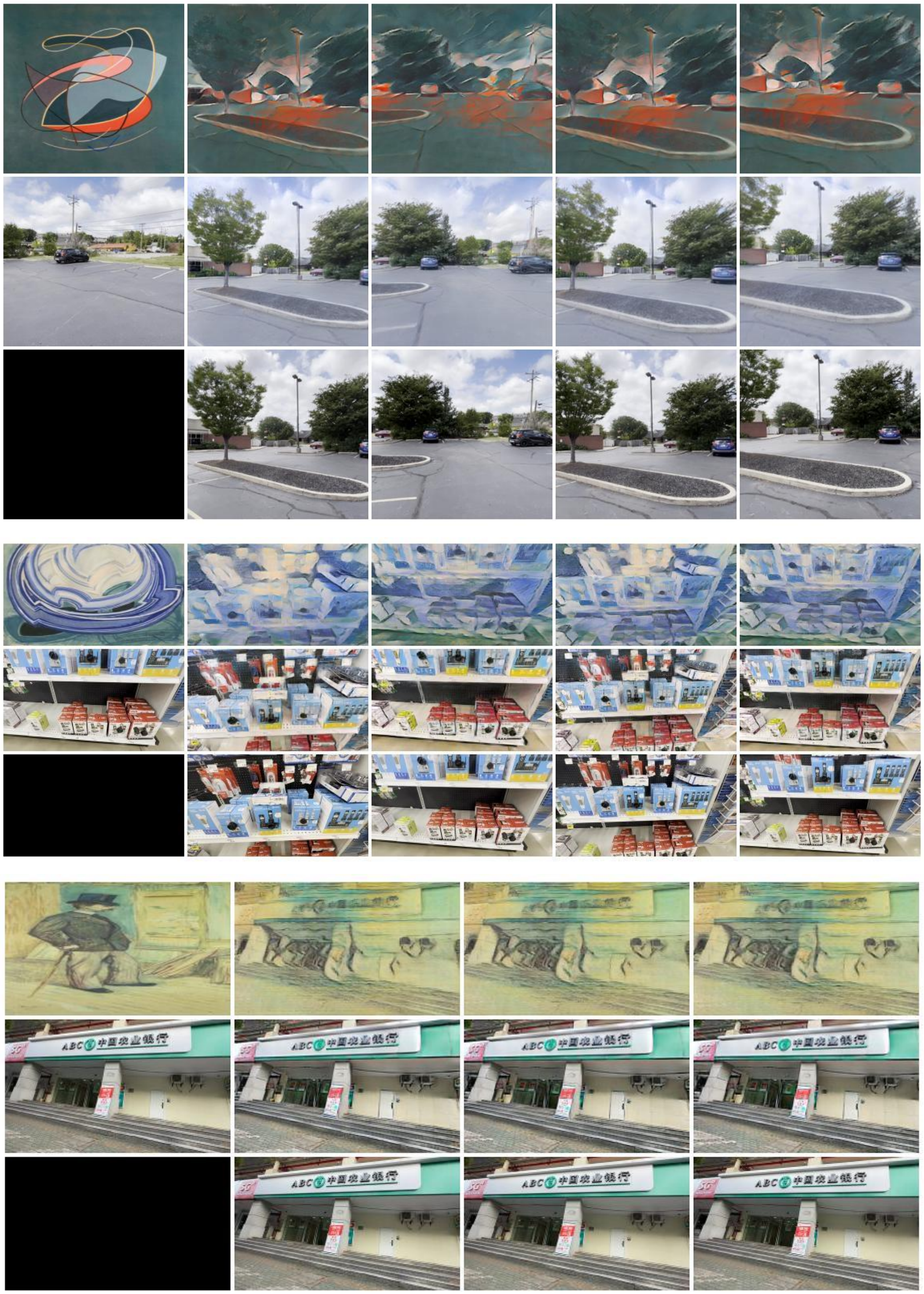


Figure 23: Additional visual results of Stylos on the DL3DV-10K dataset. The top left is the style image while the rest of the top row are the rendered images. The mid row are the rendered images conditioned on the mid left imamge. The bottom row is the content inputs.

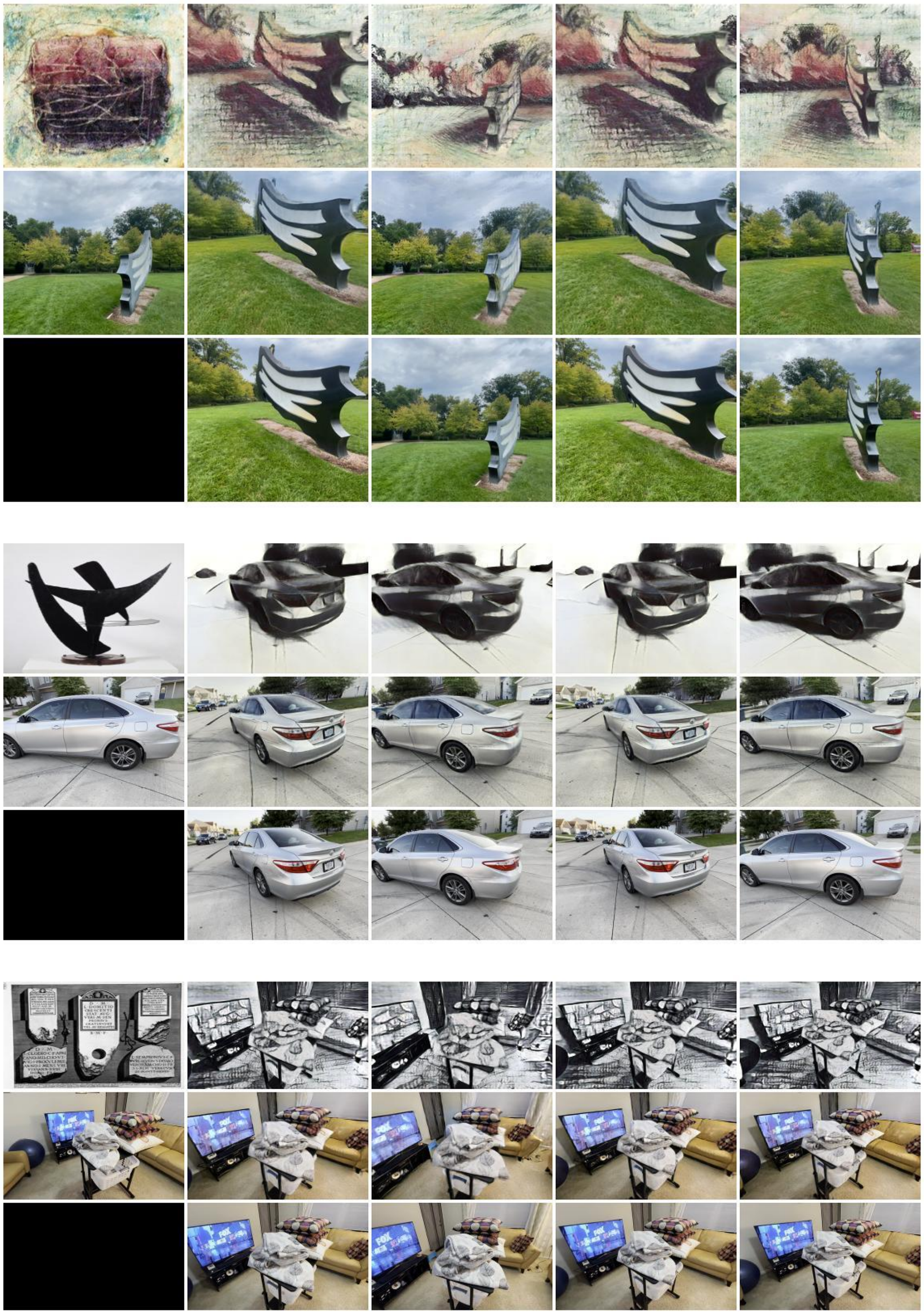


Figure 24: Additional visual results of Stylos on diverse style–content pairs.

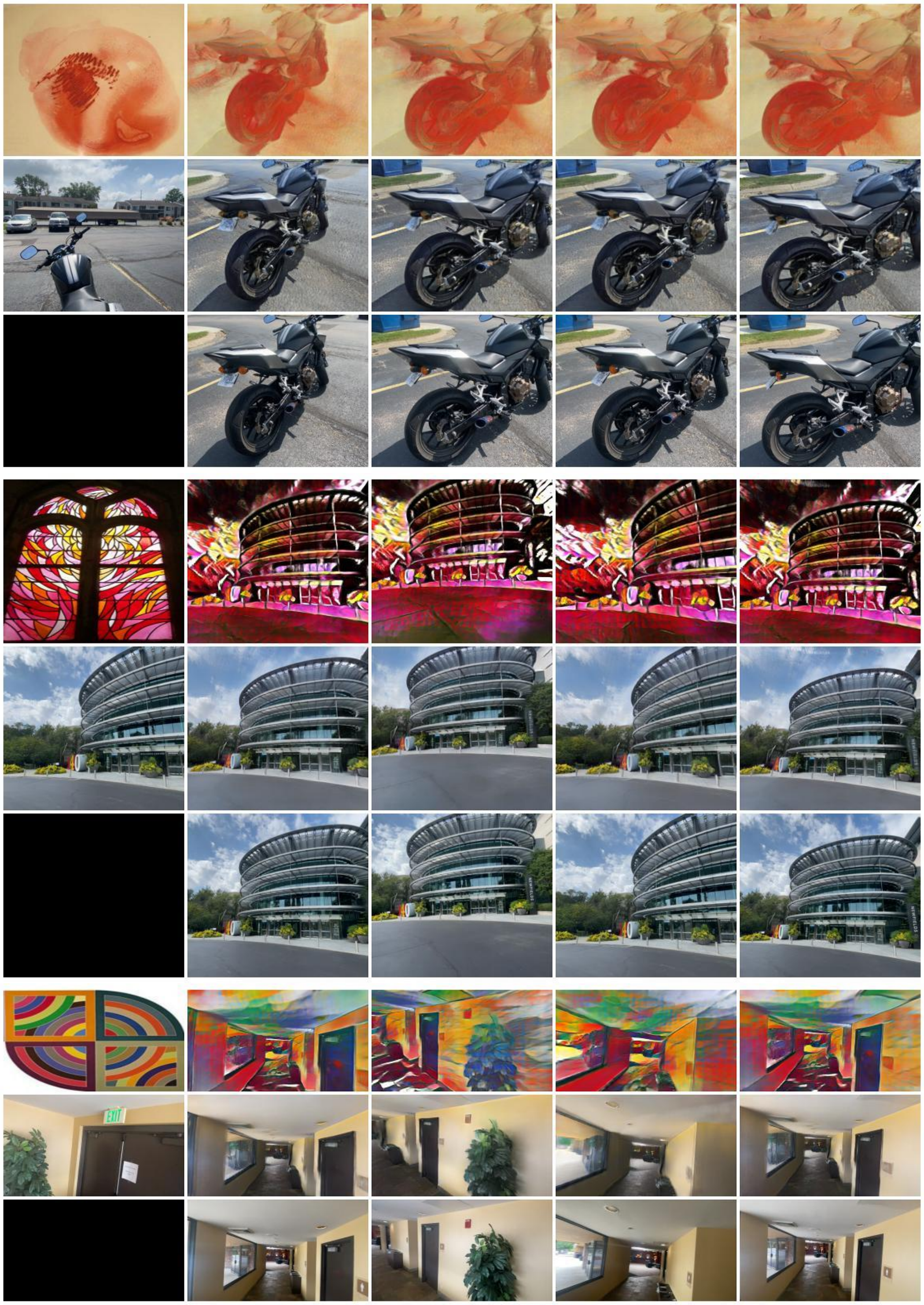


Figure 25: Additional visual results of Stylos on diverse style–content pairs.