Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

BindStyle: mesh-bound dynamic Gaussian stylization with parameter-routed visual supervision

Yongang Yu
Central South University
Zhigang Chen
czg@csu.edu.cn

Central South University
Lingfeng Liu
Central South University

Article

Keywords: dynamic Gaussian splatting, artistic stylization, mesh binding, gradient routing, video diffusion

Posted Date: November 25th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11540095/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

BindStyle: mesh-bound dynamic Gaussian stylization with parameter-routed visual supervision

Yongang Yu[1], Zhigang Chen[1*], Lingfeng Liu[1]

[1]School of Computer Science and Engineering, Central South University, Changsha, China.

*Corresponding author(s). E-mail(s): czg@csu.edu.cn;

Abstract

Stylizing a dynamic three-dimensional (3D) scene from a single reference image requires transferring both artistic appearance and local geometric structure while keeping these edits persistently associated with moving scene elements. We present BindStyle, a framework that learns shared Gaussian appearance and local geometry on a temporally corresponding mesh sequence. By anchoring primitives to persistent triangle faces while holding mesh vertices, attachment identities, and opacities fixed, BindStyle transports edited Gaussian parameters coherently across supplied scene motion. To resolve the supervisory ambiguity between color and shape in differentiable rendering, we introduce explicit gradient routing across two complementary visual teachers: local VGG feature matching updates spherical-harmonic (SH) appearance coefficients, whereas a reference-conditioned video diffusion teacher—guided by mesh depth and short-window temporal context—updates mesh-local positions, scales, and rotations. Optimization follows a two-stage curriculum that adapts appearance prior to joint geometric refinement. Once trained, the stylized dynamic scene renders directly via mesh-bound Gaussian splatting without diffusion inference or per-frame optimization. Experiments on the D-NeRF and DG-Mesh benchmarks demonstrate that combining mesh-bound parameter sharing with routed supervision improves artistic fidelity, content preservation, and multi-view temporal coherence over existing static and dynamic baselines.

Keywords: dynamic Gaussian splatting, artistic stylization, mesh binding, gradient routing, video diffusion

1 Introduction

Dynamic three-dimensional (3D) stylization aims to transfer the visual characteristics of a reference artwork to a moving scene. Achieving convincing results requires preserving recognizable underlying content while ensuring that stylized colors, brush textures, and local geometric abstractions remain persistently attached to the same surface elements across viewpoints and time. Applying image-based stylization independently across frames fails to establish such correspondence. Conversely, while a shared 3D scene representation provides persistent parameters, differentiable rendering introduces an inherent supervisory ambiguity: a discrepancy in rendered 2D projections can be explained by modifying either surface radiance or spatial primitive support. Effective dynamic stylization must therefore address both temporal correspondence and the assignment of supervisory responsibility across parameter groups.

A central insight of BindStyle is that feature-level texture matching and generative video guidance are best suited to supervising distinct parameter groups. We therefore route local VGG feature matching exclusively to spherical-harmonic (SH) appearance coefficients, and source-referenced video diffusion guidance exclusively to mesh-local Gaussian position, scale, and rotation. Both gradients are evaluated from an identical rendered snapshot before separate optimizers step. Furthermore, an appearance-first optimization curriculum establishes a style-adapted radiance initialization before

unlocking local geometric support alongside continued appearance refinement. During inference, persistent face attachments propagate the learned parameters across arbitrary mesh frames without diffusion queries or per-frame optimization.

2 Related Work

2.1 Dynamic scene representations

Three-dimensional Gaussian splatting (3DGS) exposes primitive appearance, position, scale, and orientation through efficient differentiable rasterization [1]. Building upon neural radiance field foundations [2], dynamic scene representations extend 3DGS and radiance fields to four dimensions (4D) using deformation networks, factorized space–time feature planes, or persistent primitive trajectories [3–6]. In parallel, surface-aligned and mesh-bound formulations explicitly couple Gaussian primitives to underlying geometric proxies [7, 8]. In this work, we adopt the mesh-binding formulation of MaGS [8] to decouple style learning from motion transport. Specifically, we address style optimization under known mesh motion rather than pose-free surface reconstruction or motion prediction.

2.2 Image and three-dimensional stylization

Classical and neural image stylization methods transfer artistic traits via feature statistics, perceptual losses, adaptive normalization, or attention mechanisms [9–12]. Video stylization frameworks incorporate optical flow or temporal constraints to suppress flicker [13–15], while recent diffusion-based approaches such as StyleID and StyleMaster leverage generative priors for reference-guided synthesis [16, 17]. In 3D space, StylizedNeRF, ARF, and StyleRF adapt radiance field appearance to match reference statistics [18–20], and static Gaussian stylization methods—including StyleGaussian, G-Style, ReGS, SGSST, StylizedGS, and Stylos—optimize or modulate Gaussian attributes for view-consistent transfer [21–26]. Dynamic extensions such as 4DStyleGaussian, ZDySS, and Style4D further propagate appearance transfer across time-varying scenes [27–29], whereas controllable mesh stylization directly deforms vertex positions [30]. To avoid geometric degradation during style optimization, however, most radiance- and Gaussian-based approaches restrict updates to appearance parameters. BindStyle instead preserves the supplied mesh motion while jointly learning shared appearance and local Gaussian spatial support relative to the moving surface.

2.3 Diffusion-guided stylization

To supervise geometric adaptation without explicit 3D style targets, generative diffusion priors offer rich structural and perceptual cues [31–33]. Score distillation and iterative editing frameworks bridge these 2D priors with 3D representations [34–37], enabling recent diffusion-driven 3D stylization and shape-editing methods such as FantasyStyle, DiffStyle3D, SIC3D, and GeoStyle [38–41]. Inspired by classifier score distillation (CSD) and delta denoising score (DDS) guidance [42, 43], we employ a frozen video diffusion teacher conditioned on the style reference, mesh depth, and a short temporal window. By contrasting noise predictions between stylized and cached source renders, this teacher provides a structure- and motion-aware update direction for local Gaussian geometry without requiring ground-truth 3D stylized shapes.

3 Methods

The overall learning and propagation architecture is illustrated in Figure 1, and the gradient-routing and face-attachment mechanics are detailed in Figure 2.

3.1 Inputs and shared scene parameters

Given a single style reference image S, BindStyle takes as input a pretrained mesh-bound Gaussian scene, multi-view or monocular content frames $\{J_t\}$, calibrated camera parameters $\{C_t\}$, foreground segmentation masks, and a temporally corresponding triangle mesh sequence $M_t = (V_t, F)$ sharing a fixed connectivity F and time-varying vertex positions V_t. The mesh sequence supplies underlying scene motion; neither the meshes nor the content photographs serve as ground-truth targets for stylized 3D shape. For each of the N Gaussian primitives, the parent face index f_i and opacity remain

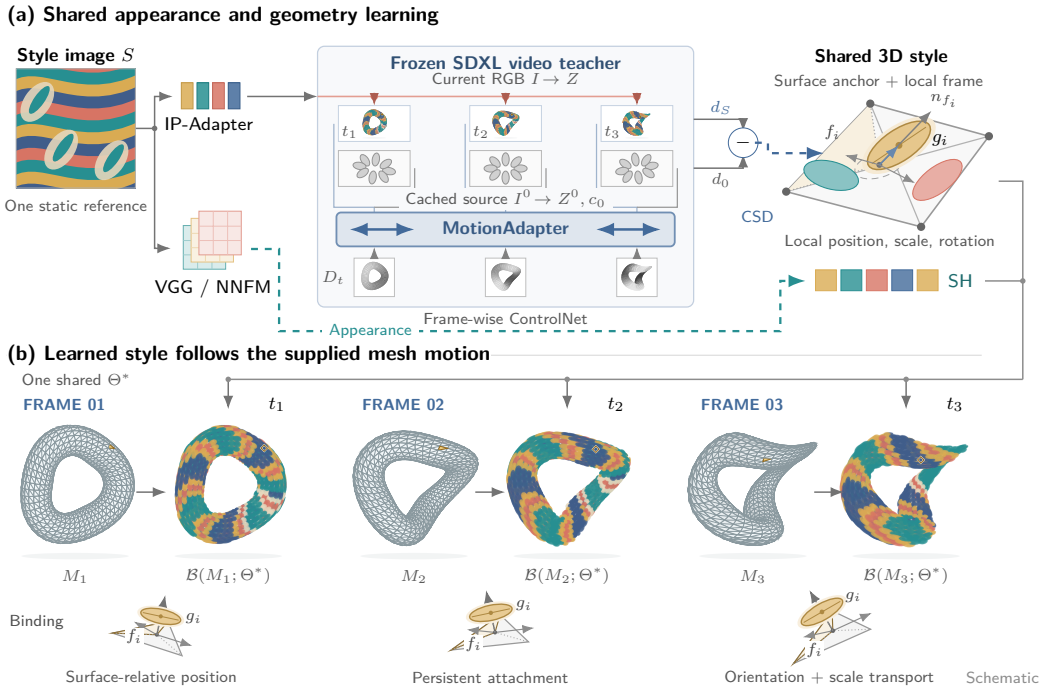

Figure 1 BindStyle framework. a, A single style reference supplies VGG features for appearance matching and image conditioning for a frozen video diffusion teacher. Current stylized renders and cached source renders share mesh depth, diffusion timestep, and noise realizations; their guidance difference updates mesh-local Gaussian geometry, whereas VGG feature matching updates SH appearance. **b**, Persistent triangle-face attachments transport the learned style parameters across the supplied mesh sequence. Gold markers track an identical primitive attachment across time.

fixed throughout stylization. The trainable parameter set $\Theta = (A, G)$ is partitioned into appearance parameters A and mesh-local geometric parameters G:

$$\Theta = (A, G), \quad A = \{c_i^{\mathrm{DC}}, c_i^{\mathrm{rest}}\}_{i=1}^{N}, \quad G = \{u_i, v_i, w_i, \ell_i, q_i\}_{i=1}^{N}, \tag{1}$$

where c_i^{DC} and c_i^{rest} denote the zero-order and higher-order spherical-harmonic (SH) coefficients, (u_i, v_i) are local tangential coordinates on face f_i, w_i is the scalar offset along the face normal, $\ell_i \in \mathbb{R}^3$ represents log-scale, and $q_i \in \mathbb{R}^4$ is a local rotation quaternion. Optimization updates these shared tensors directly without introducing auxiliary neural deformation or style-prediction networks, while keeping mesh vertices, primitive count N, and face attachments invariant.

3.2 Mesh binding and motion propagation

Let $\mathcal{B}$ denote the mesh-binding transformation adopted from MaGS [8] and $\mathcal{R}$ denote differentiable Gaussian splatting rasterization [1]. The forward rendering map at timestamp t and camera view v is given by

$$\hat{I}_{t,v}(\Theta) = \mathcal{R}\big(\mathcal{B}(M_t; \Theta), C_{t,v}\big), \tag{2}$$

where a single time-invariant parameter set Θ renders all timestamps and viewpoints. Specifically, given the vertex positions $v^0_{f_i,t}, v^1_{f_i,t}, v^2_{f_i,t}$ and unit normal $n_{f_i,t}$ of attachment face f_i at time t, the world-space Gaussian center $x_{i,t}$ is computed as

$$x_{i,t} = u_i v^0_{f_i,t} + v_i v^1_{f_i,t} + (1 - u_i - v_i) v^2_{f_i,t} + w_i n_{f_i,t}. \tag{3}$$

To transport full 3D covariance across motion, the normalized local quaternion $q_i/\|q_i\|$ is composed with the instantaneous local tangent-normal frame of face f_i, and the positive local scale $\exp(\ell_i)$ is multiplied by a face-area scaling factor (Figure 2b).

During optimization, (u_i, v_i) are not clamped to the standard simplex and w_i is not hard-bounded; thus, mesh binding serves as a persistent coordinate parameterization for motion transport rather than a rigid surface-distance constraint. At inference time, transforming Θ via equation (3) across target meshes M_t directly yields stylized dynamic renders in real time without diffusion model queries or deformation network evaluations.

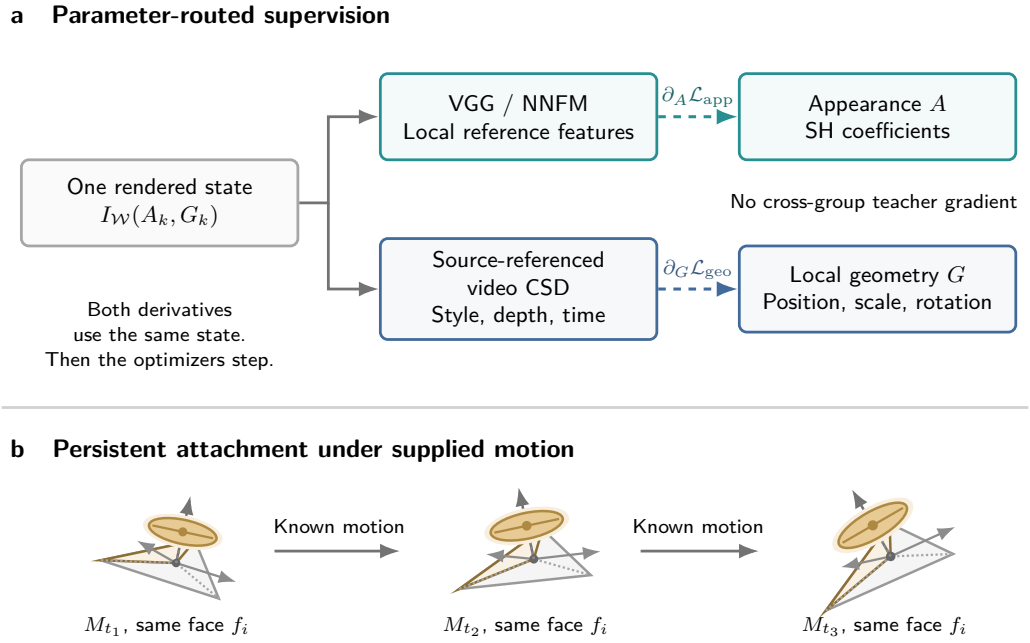

Figure 2 Parameter routing and persistent attachment. a, Local VGG matching supervises SH appearance (A), whereas source-referenced video guidance supervises mesh-local geometry (G). Both partial derivatives are evaluated from the same rendered state prior to optimizer updates. **b**, A persistent face attachment transports a Gaussian primitive's center, orientation, and scale across supplied mesh deformations without introducing additional surface-distance constraints.

3.3 Appearance supervision from reference features

To transfer fine-grained color palettes and local brush textures, we extract spatial feature maps using a frozen VGG16 network [44] and apply Nearest-Neighbor Feature Matching (NNFM) [19]. Let $a^l_{t,p}$ and s^l_r denote ℓ_2-normalized feature vectors at spatial location p of rendered frame t and location r of the style reference S within VGG layer $l \in \mathcal{L}$. Over a temporal window $\mathcal{W}$ of F_w frames, the appearance loss is defined as

$$\mathcal{L}_{\text{app}} = \frac{1}{|\mathcal{L}|} \sum_{l \in \mathcal{L}} \frac{1}{F_w P_l} \sum_{t \in \mathcal{W}} \sum_{p=1}^{P_l} \min_r \left(1 - \langle a^l_{t,p}, s^l_r \rangle\right), \tag{4}$$

where P_l is the number of spatial feature locations in layer l. Rendered frames are cropped to the bounding box of the union of source foreground masks with a 10% margin prior to feature extraction, whereas style features are extracted from the full reference image. Nearest-neighbor searches are evaluated in memory-bounded spatial chunks and averaged equally across selected VGG layers. Our formulation relies purely on NNFM without auxiliary Gram matrix, AdaIN, or pixel-level content reconstruction terms.

3.4 Source referenced video guidance

To supervise local geometric adaptation with temporal and structural awareness, we construct a frozen video diffusion teacher integrating Stable Diffusion XL (SDXL) [33], an IP-Adapter for reference-image conditioning [45] via CLIP image embeddings [46], a frame-wise depth ControlNet [47], and AnimateDiff temporal motion modules [48]. Each training step samples a contiguous window $\mathcal{W}$ of F_w dataset frames along with their corresponding meshes and cameras (allowing camera pose to vary naturally across the window). We rasterize the unedited input meshes using nvdiffrast [49] to obtain clean, sequence-normalized inverse depth maps $D_{\mathcal{W}}$ that condition the ControlNet, while gradients flow through the RGB Gaussian renders.

Prior to stylization, we cache initial unstyled Gaussian renders $I^0_t = \hat{I}_t(\Theta_0)$ for all timestamps using the initial scene parameters Θ_0. During training, a frozen variational autoencoder (VAE) encodes the current stylized render window $I_{\mathcal{W}}$ into latent representations $Z = E(I_{\mathcal{W}})$ and encodes the cached

source window into detached latents Z^0. Both Z and Z^0 are perturbed to diffusion timestep τ using an identical Gaussian noise realization ϵ to form noisy latents $Z_{S,\tau}$ and $Z_{0,\tau}$.

For each branch $b \in \{S, 0\}$, we compute classifier-free guidance noise residuals $d_b = s_b\big(\epsilon_\theta(Z_{b,\tau}, c_b, D_\mathcal{W}, \tau) - \epsilon_\theta(Z_{b,\tau}, \varnothing, D_\mathcal{W}, \tau)\big)$ using shared depth $D_\mathcal{W}$ and teacher weights θ. Here, the stylized branch $b = S$ uses the style reference embedding c_S, whereas the source branch $b = 0$ uses the mean embedding c_0 averaged over cached source renders. Following a source-referenced formulation related to CSD and DDS [34, 42, 43], the clipped latent guidance direction g_Z and differentiable surrogate loss $\mathcal{L}_{\text{geo}}$ are formulated as

$$g_Z = \text{clip}\big((1 - \bar{\alpha}_\tau)(d_S - d_0), -c, c\big), \qquad \mathcal{L}_{\text{geo}} = \frac{1}{BF_w}\langle \text{sg}(g_Z), Z\rangle, \tag{5}$$

where $\bar{\alpha}_\tau$ is the cumulative diffusion signal schedule, c is an element-wise clipping threshold, B is the batch size, and $\text{sg}(\cdot)$ denotes the stop-gradient operator. Differentiating $\mathcal{L}_{\text{geo}}$ backpropagates the guidance vector g_Z through the VAE encoder and Gaussian rasterizer without differentiating through the U-Net denoiser ϵ_θ. Note that the scalar value of the linear surrogate $\mathcal{L}_{\text{geo}}$ serves solely to construct the backward pass and is not a perceptual quality metric.

3.5 Gradient routing and staged updates

At optimization step k, both supervisory branches evaluate the same parameter snapshot $\Theta_k = (A_k, G_k)$. Rather than summing $\mathcal{L}_{\text{app}}$ and $\mathcal{L}_{\text{geo}}$ into a single undifferentiated objective, BindStyle explicitly routes each supervisory signal to its designated parameter group (Figure 2a):

$$g_A = \lambda_A \partial_A \mathcal{L}_{\text{app}}(A_k, G_k), \qquad g_G = \lambda_G \partial_G \mathcal{L}_{\text{geo}}(A_k, G_k). \tag{6}$$

Specifically, geometric parameters G_k are detached when backpropagating $\mathcal{L}_{\text{app}}$ to obtain g_A, and SH appearance parameters A_k are detached when backpropagating $\mathcal{L}_{\text{geo}}$ to obtain g_G. Both partial derivatives are computed from the identical rendered state $\hat{I}_\mathcal{W}(A_k, G_k)$ before the respective Adam optimizers step. Eliminating the cross-group partial derivatives $\partial_G \mathcal{L}_{\text{app}}$ and $\partial_A \mathcal{L}_{\text{geo}}$ prevents high-frequency VGG texture gradients from destabilizing primitive geometry and prevents coarse diffusion guidance from corrupting fine SH color matching.

Training proceeds in two stages. An initial appearance-only warmup updates A while holding G fixed at its reconstruction initialization G_0. Subsequently, joint routed optimization unlocks local coordinates, scales, and rotations in G while continuing to update A. Before applying any parameter update at window boundaries, gradients undergo finite-value verification and separate group-wise ℓ_2-norm clipping. As analyzed in Section 4.9 and Figure 5B, appearance-first staging serves as a stable initialization curriculum rather than a guarantee of optimal terminal value across every metric.

4 Experiments

4.1 Experimental setup

We evaluate BindStyle on dynamic scenes from the D-NeRF and DG-Mesh benchmarks [3, 50]. Multi-view and temporal consistency evaluations span six dynamic sequences—Jumping Jacks, Bouncing Balls, LEGO, Beagle, Bird, and Girlwalk—weighted equally across scenes. Perceptual quality metrics and routing ablations are conducted on dedicated, matched evaluation sets that are kept strictly separate without cross-population pooling. Within each comparison, camera trajectories, timestamps, style references, and cropping protocols are held identical across competing methods (see Experiments Sections 4.3–4.5 for mask definitions, aggregation rules, and sampling units).

We compare BindStyle against representative static 3DGS stylization baselines applied per timestamp (StyleGaussian [21], StylizedGS [25], and FantasyStyle [38]) as well as native dynamic 4D stylization frameworks (4DStyleGaussian [27] and Style4D [29]). While all methods evaluate on identical views, timestamps, and masks, BindStyle additionally leverages the supplied mesh sequence for correspondence and depth conditioning; we explicitly account for this input structural prior when interpreting comparative performance.

4.2 Implementation settings

In our reference configuration, each training window comprises $F_w = 2$ consecutive frames. Optimization runs for 500 appearance-only warmup steps followed by 2,000 joint routed steps (2,500 optimizer updates, corresponding to 5,000 rendered frame evaluations). We use Adam optimizers with learning rates of 2×10^{-3} for c^{DC}, 2×10^{-4} for c^{rest}, 1×10^{-4} for local coordinates (u, v, w), and 5×10^{-5} for log-scales ℓ and rotations q. Loss weights are set to $\lambda_A = \lambda_G = 1.0$, and group-wise gradients g_A and g_G are clipped to a maximum ℓ_2 norm of 1.0. To ensure numerical stability without external geometric regularizers, log-scales ℓ_i are clamped to $[-20, 8]$ prior to exponentiation and quaternions q_i are ℓ_2-normalized at each step.

For appearance supervision, NNFM extracts features from VGG16 layers 11, 13, and 15 (`relu3_1`, `relu3_2`, `relu3_3`) with equal weighting. Both the 10%-padded foreground union crop of the rendered image and the full style reference are resized to a long edge of 256 pixels. For geometric supervision, the video teacher combines SDXL Base 1.0, AnimateDiff SDXL-Beta, SDXL depth ControlNet, and standard SDXL IP-Adapter with classifier-free guidance scale 7.0, IP-Adapter scale 1.0, and ControlNet conditioning scale 0.35. Teacher inputs are resized to 256×256, and diffusion timesteps τ are sampled uniformly within $[0.20, 0.65]$ of the noise schedule. The VAE encodes images in `float32` using the posterior mode, whereas the U-Net denoiser executes in `float16`. Noise realizations are shared between the current and source branches at each step but sampled independently across frames within the window.

4.3 Evaluation populations and metrics

Our quantitative evaluation employs ArtScore [51], ArtFID [52], histogram-based CM Loss [26, 53], and aligned LPIPS [54] and pixel RMSE. Single-sequence quality evaluations and single-case ablations are conducted on the 200-frame Jumping Jacks sequence evaluated at 5,000 frame iterations. The broader paired-quality benchmark evaluates nine scene–style combinations: Jumping Jacks (styles A, B), Bouncing Balls (styles A, C), Beagle (styles A, B), Bird (style C), and Girlwalk (styles A, B). Within each combination, all competing methods are evaluated on identical camera views, timestamps, and style references, with final scores averaged equally across the nine pairs.

For ArtFID computation, each dynamic method generates 216 stylized frames paired with 72 unique content images and 9 reference artworks, evaluated using 512×512 center crops in sRGB space. While content preservation is computed between each stylized frame and its exact corresponding content frame, the Fréchet distance for style fidelity is estimated against the 9-artwork reference set; given this compact sample size, ArtFID is interpreted alongside paired metrics.

Multi-view consistency is evaluated across camera pairs partitioned into narrow ($(0°, 10°]$) and wide ($[30°, 60°]$) angular baselines using exact mesh-surface correspondence and z-buffer visibility filtering. Temporal consistency is evaluated on fixed-camera sequences at frame intervals of $\Delta t = 2$ and $\Delta t = 20$ using optical flow estimated by RAFT [55] on unstyled source renders (note that the single-scene adjacent-frame curve in Figure 5A follows a separate diagnostic protocol). All multi-view and temporal consistency metrics are averaged with equal weight across the six benchmark scenes (Jumping Jacks, Bouncing Balls, LEGO, Beagle, Bird, and Girlwalk). Quality, consistency, and ablation populations are maintained strictly separate.

4.4 Common visible support

To prevent methods with shrunken or fragmented silhouettes from gaining an unfair advantage in consistency metrics, we establish a method-agnostic common visible support mask for every directed evaluation pair. Eligible source pixels are defined as foreground pixels possessing finite positive depth in the unstyled source reconstruction. Common masks are precomputed once from unstyled source renders and input meshes and cached identically across all evaluated methods.

For multi-view pairs, an eligible source pixel is marked valid if its 3D surface point projects within the target image boundaries, falls inside the target foreground mask, and exhibits a projected depth within 1% relative error of the target mesh z-buffer. For temporal pairs, validity requires falling within the target foreground mask under a shared RAFT flow field computed on unstyled renders and satisfying a forward–backward flow consistency threshold of 1.5 pixels (defined at a 400-pixel short edge and scaled proportionally with image resolution).

Crucially, stylized RGB values, modified opacities, and method-specific silhouettes play no role in selecting common evaluation pixels. LPIPS is computed by area-resampling the binary validity mask to match the spatial dimensions of each internal perceptual feature map, while RMSE is evaluated

directly over the valid pixel support. Pair directions, image interpolation, and preprocessing are held identical across methods. Reported coverage percentages represent the ratio of valid to eligible pixels averaged equally across scenes. View or frame pairs with zero valid support are excluded uniformly across all methods, whereas non-finite renders are logged as run failures rather than masked out. Both the method-specific mask protocol and the common-mask protocol are reported side by side in Table 2.

4.5 Statistical reporting

Quantitative routing ablations in Table 3 are evaluated over three matched scene–style cases. Variants marked with an asterisk (⋆) are repeated across three random seeds per case, with reported SDs reflecting variation across seed-wise case means. Reported differences represent descriptive contrasts across matched conditions rather than formal hypothesis tests. In separate stability sweeps evaluating four style references at fixed seed 42 and five random seeds on style reference A, BindStyle completed 24/24 style-sweep runs and 29/30 seed-sweep runs with finite outputs (one seed run encountered non-finite values and is retained in the attempt count but excluded from finite score averages), compared to 24/24 and 30/30 for Style4D.

4.6 Artistic appearance and content preservation

Figure 3a presents qualitative comparisons evaluating color palette transfer, motif fidelity, and semantic recognizability. Rather than producing a uniform global color cast, BindStyle transfers structured reference patterns onto coherent surface regions while preserving fine-scale articulated geometry and curved contours. Quantitative evaluations in Table 1 corroborate these visual observations across multiple perceptual dimensions, while recognizing that the degree of geometric abstraction naturally varies with the structural characteristics of each reference artwork.

To provide a multifaceted assessment, Table 1 reports complementary automated metrics. ArtScore [51] evaluates learned artistic appeal, CM Loss [26] quantifies reference palette alignment via the Hellinger distance between RGB-uv color histograms, and ArtFID [52] assesses distributional style fidelity and content preservation across an image collection. Across both single-sequence and paired multi-scene protocols, BindStyle achieves favorable performance across these rendered perceptual criteria. Because these 2D metrics evaluate projected appearance rather than explicit 3D shape deformation—and because ArtFID is computed over a compact reference collection distinct from the paired quality set—we interpret ArtFID jointly with the paired metrics rather than in isolation.

Table 1 Quality and reference matching. The first block reports single-sequence ArtScore and image-set ArtFID; the second averages nine scene–style pairs. The evaluation populations and aggregation are specified in Experiments section 4.3; n/a denotes unevaluated entries.

	Artistic quality		Nine pairs	
Method	ArtScore↑	ArtFID↓	ArtScore↑	CM Loss↓
StyleGaussian (2024)†	0.583	42.7	*n/a*	*n/a*
StylizedGS (2025)†	0.641	35.2	*n/a*	*n/a*
FantasyStyle (2026)†	0.689	29.8	*n/a*	*n/a*
4DStyleGaussian (2024/2026)	0.712	26.4	0.698	0.817
Style4D (2026)	0.738	23.1	0.724	0.767
BindStyle	**0.781**	**19.6**	**0.769**	**0.686**

4.7 Consistency across viewpoints and motion

Evaluating consistency in stylization methods that modify primitive geometry requires care, as changes in opacity or footprint size can alter the rendered foreground silhouette. Table 2a therefore reports multi-view consistency under both method-specific masks and a shared common visible support mask derived strictly from unstyled source geometry (Experiments Section 4.4). BindStyle achieves the lowest aligned LPIPS and RMSE errors across both narrow (0°–10°) and wide (30°–60°) camera baselines under both masking protocols. Retaining superior coherence under the common mask confirms that the performance gain stems from consistent 3D representation rather than selective silhouette shrinkage. As expected, wider angular baselines yield higher absolute error due to visibility changes

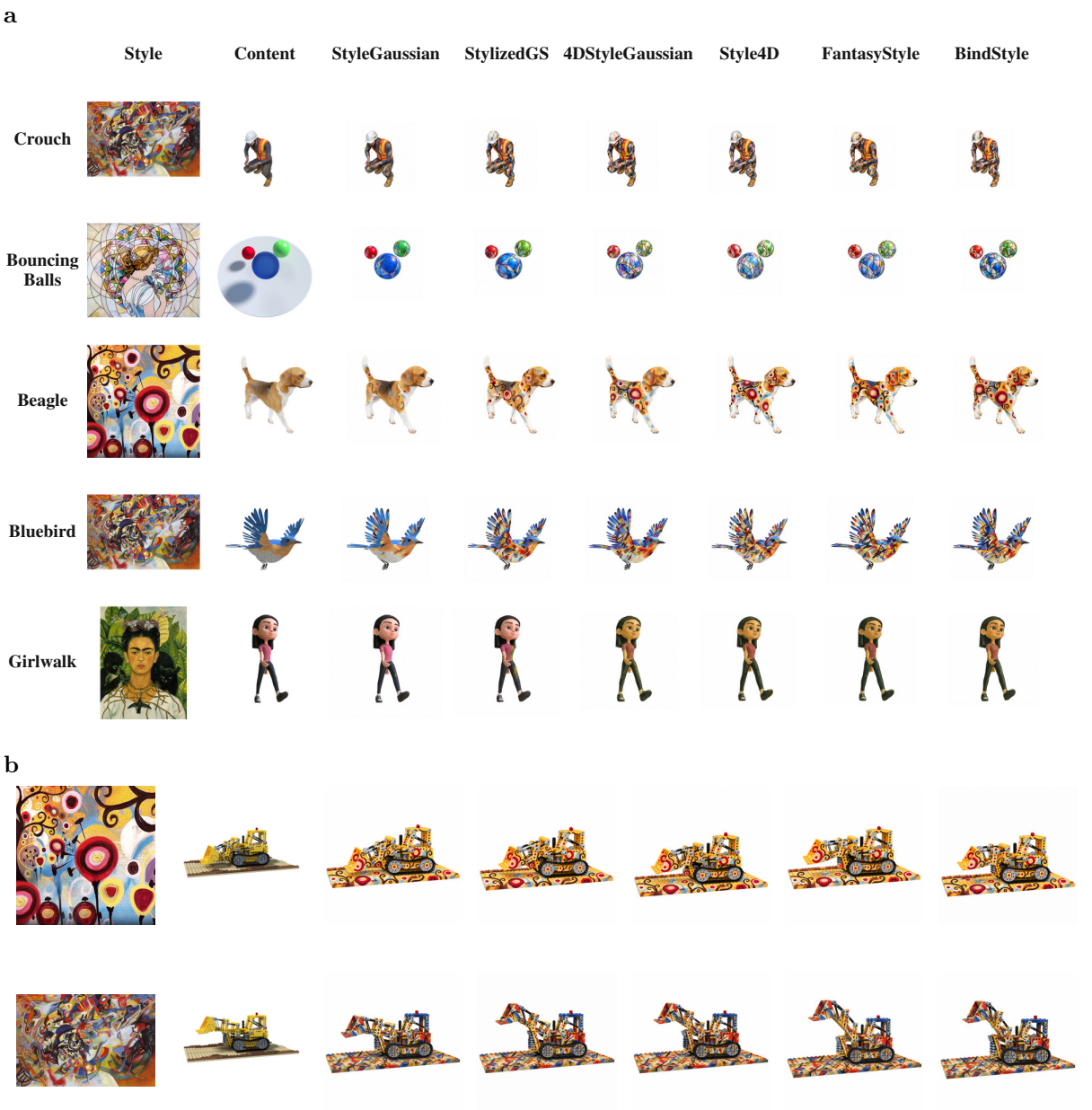

Figure 3 Visual quality and style propagation. a, Reference, source, and results from StyleGaussian, StylizedGS, 4DStyleGaussian, Style4D, FantasyStyle, and BindStyle. The comparison shows palette transfer, pattern coverage, and recognizable structure. **b**, Two references and successive BindStyle renders of LEGO. Circular motifs and angular brushwork remain associated with articulated parts.

and foreshortening. While the common source mask ensures fair pixel-level correspondence, qualitative inspection remains essential for assessing stylized footprints that extend beyond the unstyled silhouette.

Figure 3b demonstrates temporal style transport on the articulated LEGO sequence. As the loader bucket, chassis, and base articulate across frames, stylized motifs remain firmly anchored to their respective mechanical components. Quantitative temporal evaluations at short ($\Delta t = 2$) and long ($\Delta t = 20$) frame intervals in Table 2b similarly favor BindStyle over both static per-frame baselines and deformation-field-based dynamic methods. Anchoring shared Gaussian parameters to persistent mesh faces provides a deterministic mechanism for temporal style propagation without per-frame re-optimization, whereas residual pixel variations across time reflect natural visibility transitions and view-dependent SH shading.

Table 2 Consistency across viewpoints and motion. a, Multi-view evaluation. **b**, Temporal evaluation. Method-specific and common-mask scores retain their original values and are not pooled. Coverage is averaged over scenes. † denotes per-time processing; bold marks the lowest reported error. Experiments section 4.4 defines correspondence, masks, and exclusions.

a Multi-view consistency

	Method-specific masks				Common masks			
	Narrow-baseline		Wide-baseline		Narrow-baseline		Wide-baseline	
Method	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓
StyleGaussian (2024)†	0.4539	0.0412	0.5788	0.1414	0.4612	0.0435	0.5894	0.1487
StylizedGS (2025)†	0.4098	0.0337	0.5225	0.1179	0.4175	0.0356	0.5331	0.1242
FantasyStyle (2026)†	0.3708	0.0266	0.4659	0.0964	0.3784	0.0283	0.4762	0.1021
4DStyleGaussian (2024/2026)	0.3450	0.0213	0.4127	0.0808	0.3526	0.0228	0.4235	0.0859
Style4D (2026)	0.3220	0.0175	0.3791	0.0694	0.3297	0.0189	0.3894	0.0741
BindStyle	**0.2963**	**0.0138**	**0.3383**	**0.0598**	**0.3031**	**0.0149**	**0.3472**	**0.0643**

Narrow/wide angles are $(0, 10]/[30, 60]$ degrees. Shared coverage (%): narrow 94.6, wide 91.2.

b Temporal consistency

	Method-specific masks				Common masks			
	Short-range		Long-range		Short-range		Long-range	
Method	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓	LPIPS↓	RMSE↓
Per-frame extensions								
StyleGaussian (2024)†	0.2544	0.0356	0.4226	0.0787	0.2618	0.0379	0.4342	0.0835
StylizedGS (2025)†	0.2316	0.0298	0.3881	0.0652	0.2387	0.0318	0.3994	0.0694
FantasyStyle (2026)†	0.2090	0.0252	0.3530	0.0565	0.2161	0.0270	0.3638	0.0603
Native dynamic methods								
4DStyleGaussian (2024/2026)	0.1878	0.0214	0.3137	0.0485	0.1945	0.0229	0.3241	0.0518
Style4D (2026)	0.1702	0.0189	0.2898	0.0421	0.1768	0.0203	0.2995	0.0452
BindStyle	**0.1431**	**0.0147**	**0.2521**	**0.0347**	**0.1492**	**0.0158**	**0.2613**	**0.0374**

Gaps are 2 and 20 frames. Shared coverage (%): short 95.1, long 92.4.

4.8 Geometry supervision and parameter routing

To disentangle the impact of trainable geometric capacity from that of supervisory gradient routing, Table 3 evaluates controlled architectural variants. Under VGG-only appearance supervision, unlocking local Gaussian geometry ($A + G$) improves rendered metrics over SH-only optimization (A). To test whether this gain simply reflects adding the video diffusion teacher, the same-teacher SH-only control retains both teachers but restricts updates to A. Conversely, the unrouted joint control retains full trainable capacity ($A + G$) and backpropagates both VGG and video diffusion losses indiscriminately to both parameter groups. Comparing BindStyle against this matched unrouted baseline directly isolates the effect of parameter-specific gradient routing without confounding changes in model capacity or teacher availability.

Across the matched routing evaluation cases, explicit gradient routing improves both color histogram alignment (CM Loss) and temporal coherence over unrouted optimization. Qualitative comparisons in Figure 4a illustrate how SH-only fitting, editable Gaussian support, and teacher conditioning shape the stylized output. Jointly removing temporal and depth conditioning degrades structural coherence, although this control does not isolate the contribution of each condition. Figure 4b visualizes the effect of gradient routing on rendered appearance and Gaussian depth maps, providing a qualitative comparison of local support rather than a direct measurement of style-induced 3D displacement.

4.9 Temporal context and optimization order

Figure 5A analyzes the trade-off between the video teacher's temporal window length, adjacent-frame temporal error (aligned LPIPS), iteration runtime, and peak GPU memory. Extending the temporal window consistently improves adjacent-frame coherence at the cost of increased computational overhead and memory footprint, with diminishing perceptual returns at longer horizons. Under the tested hardware configuration, a four-frame window represents the largest context within a 15 GB memory

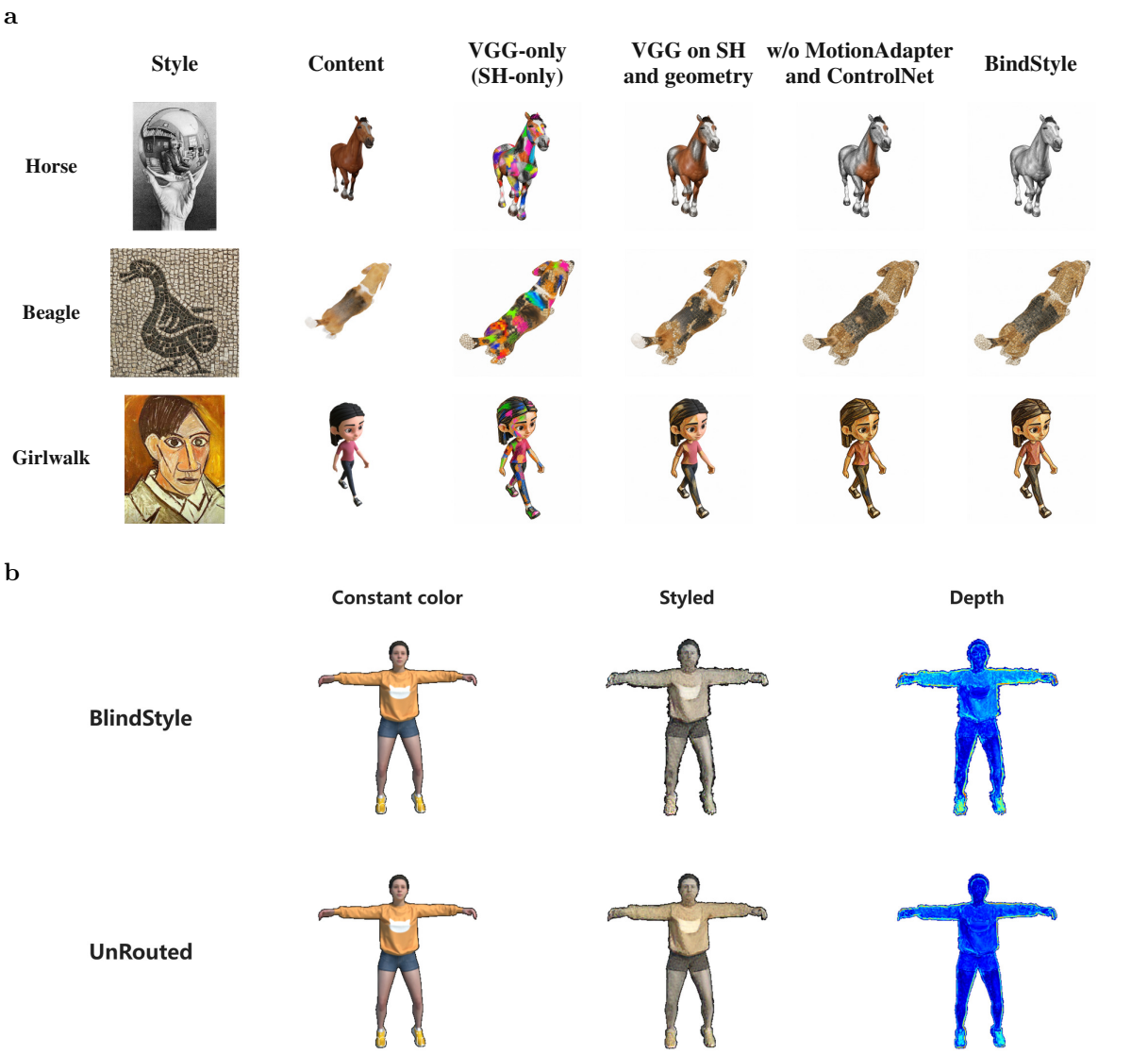

Figure 4 Qualitative ablation of supervision and routing. a, After reference and source, columns show VGG on SH, VGG on SH and geometry, joint removal of MotionAdapter and ControlNet, and BindStyle. Rows show Horse, Beagle, and Girlwalk. **b**, BindStyle (top row, originally labeled "BlindStyle") and the unrouted control through source appearance, stylized appearance, and depth. The original "Constant color" column contains spatially varying source colors, not a neutral-color geometry diagnostic. The character in panel **b** is a synthetic model from the Jumping Jacks sequence (D-NeRF [3]); the data were accessed through https://github.com/Isabella98Liu/DG-Mesh/releases/tag/ datasets. These panels support analysis of rendered appearance and support but do not measure style-only displacement.

Table 3 Supervision and gradient routing ablations. The same-teacher SH-only control retains both teachers but freezes geometry. In the three-case block, $*$ denotes three seeds per case; SD is computed across seed-wise case means. VGG + video on both groups is the unrouted control.

	Single case			Three matched cases		
Variant	CM↓	T-LPIPS↓	RMSE↓	CM↓	T-LPIPS↓	RMSE↓
VGG-only (SH-only)*	0.9743	0.2684	0.0391	0.981 ± 0.012	0.274 ± 0.008	0.041 ± 0.002
VGG + video (SH-only) *	0.9126	0.2415	0.0342	0.921 ± 0.014	0.248 ± 0.007	0.036 ± 0.002
VGG on SH and geometry w/o MotionAdapter and ControlNet	0.8567	0.2013	0.0276	0.869	0.208	0.029
VGG + video on SH and geometry*	0.9215	0.2358	0.0334	0.934	0.241	0.035
BindStyle*	0.7934	0.1782	0.0221	0.801 ± 0.015	0.183 ± 0.006	0.023 ± 0.001
	0.7082	**0.1426**	**0.0147**	$\mathbf{0.715 \pm 0.009}$	$\mathbf{0.146 \pm 0.004}$	$\mathbf{0.015 \pm 0.001}$

budget. Note that this single-sequence adjacent-frame diagnostic uses a protocol distinct from the multi-scene, multi-gap evaluation in Table 2b, and temporal smoothness must always be balanced against stylistic expressiveness.

Figure 5B investigates the optimization curriculum by tracking CM Loss across normalized training progress for three schedules: appearance-first (our default), simultaneous joint optimization, and geometry-first. All three schedules retain explicit gradient routing. Appearance-first adaptation achieves the fastest initial reduction in color-matching error, providing a well-aligned radiance initialization before geometric updates commence, whereas simultaneous optimization reaches a slightly lower terminal CM Loss and geometry-first converges intermediately. Because normalized training progress does not equate the exact number of SH and geometric update steps across schedules, these trajectory curves characterize convergence dynamics rather than serving as fixed-budget substitutes for the ablation controls in Table 3.

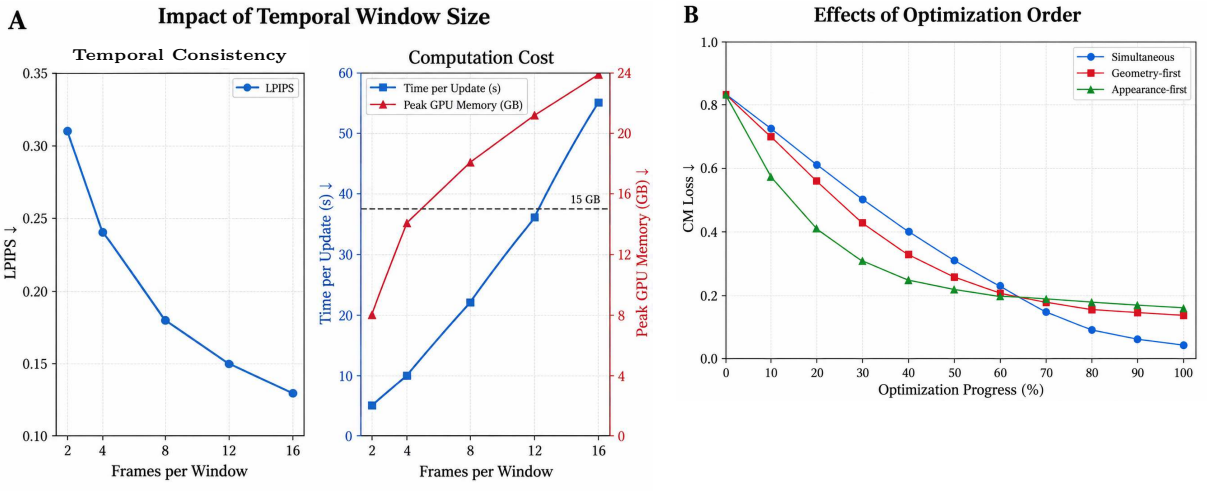

Figure 5 Temporal context and optimization order. A, Increasing the teacher window lowers aligned adjacent-frame LPIPS while increasing update time and peak GPU memory. The dashed line marks 15 GB in the plotted configuration. **B**, CM Loss under simultaneous, geometry-first, and appearance-first schedules. Appearance-first leads early; simultaneous optimization has the lowest endpoint. These analyses use evaluation configurations distinct from those in Table 2.

5 Conclusion

BindStyle separates the challenge of learning a 3D artistic style from that of transporting it across dynamic motion. Persistent triangle-face attachments establish deterministic temporal correspondence, while shared Gaussian parameters encode stylized appearance and local spatial support. Across the evaluated benchmarks, this formulation yields consistent improvements in perceptual quality, multi-view coherence, and temporal stability. At the same time, recognizing the methodological boundaries of the framework is essential for interpreting these results accurately.

6 Limitations

First, parameter-routed supervision governs direct gradient ownership at each optimization step: VGG feature matching computes partial derivatives exclusively with respect to SH coefficients, and video diffusion guidance computes derivatives exclusively with respect to local geometric parameters. Nevertheless, the two parameter groups interact indirectly across iterations through the shared forward rendering state—modifying primitive geometry alters the rendered image from which subsequent appearance gradients are extracted, and vice versa. Consequently, the matched unrouted control (which preserves identical teachers and parameter capacity) provides the most rigorous validation of gradient routing.

Second, evaluating dynamic stylization via source-derived common visibility masks carries inherent trade-offs. Computing correspondence and occlusion masks from unstyled source geometry prevents methods from artificially lowering error metrics by shrinking foreground opacity. However, source-based masks cannot track stylized Gaussian footprints that intentionally expand beyond the original

surface silhouette to form painterly strokes. Low error within the common visible support must therefore be interpreted alongside mask coverage ratios and qualitative visual inspection.

Finally, BindStyle operates within explicit structural assumptions. The method requires a pre-reconstructed scene and a topologically consistent mesh sequence across all target timestamps; it does not predict novel motion or handle topological tearing. Because local barycentric coordinates and normal offsets are not hard-constrained to the triangle surface, extreme guidance gradients could still displace primitives if unregularized, and short diffusion windows capture only local temporal context. Conversely, strong depth conditioning preserves recognizable content structure at the potential cost of limiting extreme geometric abstraction.

Data availability

The D-NeRF [3] and DG-Mesh [50] datasets are publicly available at https://github.com/albertpumarola/D-NeRF and https://github.com/Isabella98Liu/DG-Mesh, respectively, subject to their original terms of use. The numerical source data and evaluation records supporting the findings of this study are available from the corresponding author upon reasonable request.

Code availability

The code is publicly available at https://github.com/yongangYu/BindStyle-public.

References

[1] Kerbl, B., Kopanas, G., Leimkühler, T. & Drettakis, G. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* **42** (2023). URL https://arxiv.org/abs/2308.04079.

[2] Mildenhall, B. *et al.* NeRF: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision* (2020). URL https://arxiv.org/abs/2003.08934.

[3] Pumarola, A., Corona, E., Pons-Moll, G. & Moreno-Noguer, F. D-NeRF: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021). URL https://arxiv.org/abs/2011.13961.

[4] Cao, A. & Johnson, J. HexPlane: A fast representation for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 130–141 (2023). URL https://openaccess.thecvf.com/content/CVPR2023/html/Cao_HexPlane_A_Fast_Representation_for_Dynamic_Scenes_CVPR_2023_paper.html.

[5] Wu, G. *et al.* 4d gaussian splatting for real-time dynamic scene rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). URL https://arxiv.org/abs/2310.08528.

[6] Luiten, J., Kopanas, G., Leibe, B. & Ramanan, D. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *International Conference on 3D Vision* (2024). URL https://dynamic3dgaussians.github.io/.

[7] Guédon, A. & Lepetit, V. SuGaR: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2024). URL https://arxiv.org/abs/2311.12775.

[8] Ma, S., Luo, Y., Yang, W. & Yang, Y. MaGS: Reconstructing and simulating dynamic 3d objects with mesh-adsorbed gaussian splatting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025). URL https://arxiv.org/abs/2406.01593.

[9] Gatys, L. A., Ecker, A. S. & Bethge, M. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016). URL https://openaccess.thecvf.com/content_cvpr_2016/html/Gatys_Image_Style_Transfer_CVPR_2016_paper.html.

[10] Johnson, J., Alahi, A. & Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision* (2016). URL https://arxiv.org/abs/1603.08155.

[11] Huang, X. & Belongie, S. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision* (2017). URL https://arxiv.org/abs/1703.06868.

[12] Liu, S. *et al.* AdaAttN: Revisit attention mechanism in arbitrary neural style transfer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2021). URL https://arxiv.org/abs/2108.03647.

[13] Ruder, M., Dosovitskiy, A. & Brox, T. Artistic style transfer for videos. In *German Conference on Pattern Recognition*, 26–36 (2016). URL https://arxiv.org/abs/1604.08610.

[14] Huang, H. *et al.* Real-time neural style transfer for videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017). URL https://openaccess.thecvf.com/content_cvpr_2017/html/Huang_Real-Time_Neural_Style_CVPR_2017_paper.html.

[15] Gupta, A., Johnson, J., Alahi, A. & Fei-Fei, L. Characterizing and improving stability in neural style transfer. In *Proceedings of the IEEE International Conference on Computer Vision*, 4067–4076 (2017). URL https://openaccess.thecvf.com/content_iccv_2017/html/Gupta_Characterizing_and_Improving_ICCV_2017_paper.html.

[16] Chung, J., Hyun, S. & Heo, J.-P. 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* (2024). URL https://arxiv.org/abs/2312.09008.

[17] Ye, Z. *et al.* StyleMaster: Stylize your video with artistic generation and translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2025). URL https://arxiv.org/abs/2412.07744.

[18] Huang, Y.-H., He, Y., Yuan, Y.-J., Lai, Y.-K. & Gao, L. StylizedNeRF: Consistent 3d scene stylization as stylized NeRF via 2d-3d mutual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18342–18352 (2022). URL https://openaccess.thecvf.com/content/CVPR2022/html/Huang_StylizedNeRF_Consistent_3D_Scene_Stylization_As_Stylized_NeRF_via_2D-3D_CVPR_2022_paper.html.

[19] Zhang, K. *et al.* ARF: Artistic radiance fields. In *European Conference on Computer Vision* (2022). URL https://arxiv.org/abs/2206.06360.

[20] Liu, K. *et al.* StyleRF: Zero-shot 3d style transfer of neural radiance fields. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8338–8348 (2023). URL https://openaccess.thecvf.com/content/CVPR2023/html/Liu_StyleRF_Zero-Shot_3D_Style_Transfer_of_Neural_Radiance_Fields_CVPR_2023_paper.html.

[21] Liu, K. *et al.* StyleGaussian: Instant 3d style transfer with gaussian splatting. In *SIGGRAPH Asia 2024 Technical Communications* (2024). URL https://kunhao-liu.github.io/StyleGaussian/.

[22] Kovács, Á. S., Hermosilla, P. & Raidou, R. G. G-Style: Stylized gaussian splatting. *Computer Graphics Forum* **43**, e15259 (2024). URL https://diglib.eg.org/items/758761fc-b693-4242-bd3b-198bb4730ba5.

[23] Mei, Y., Xu, J. & Patel, V. M. ReGS: Reference-based controllable scene stylization with gaussian splatting. In *Advances in Neural Information Processing Systems*, Vol. 37, 4035–4061 (2024). URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/076c1fa639a7190e216e734f0a1b3e7b-Abstract-Conference.html.

[24] Galerne, B., Wang, J., Raad, L. & Morel, J.-M. SGSST: Scaling gaussian splatting style transfer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 26535–26544 (2025). URL https://openaccess.thecvf.com/content/CVPR2025/html/Galerne_SGSST_Scaling_Gaussian_Splatting_Style_Transfer_CVPR_2025_paper.html.

[25] Zhang, D. *et al.* StylizedGS: Controllable stylization for 3d gaussian splatting. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025). URL https://kristen-z.github.io/stylizedgs/.

[26] Liu, H., Huang, J., Lu, M., Saripalli, S. & Jiang, P. Stylos: Multi-view 3d stylization with single-forward gaussian splatting. In *International Conference on Learning Representations* (2026). URL https://arxiv.org/abs/2509.26455.

[27] Liang, W., Xu, H., Chen, W., Xiao, F. & Kang, W. 4DStyleGaussian: Generalizable 4d style transfer with gaussian splatting. *Pattern Recognition* **172**, 112422 (2026). URL https://www.sciencedirect.com/science/article/pii/S0031320325010830.

[28] Saroha, A. *et al.* ZDySS: Zero-shot dynamic scene stylization using gaussian splatting. *arXiv preprint arXiv:2501.03875* (2025). URL https://arxiv.org/abs/2501.03875.

[29] Chen, B. *et al.* Style4D-Bench: A benchmark suite for 4d stylization. In *AAAI Conference on Artificial Intelligence*, Vol. 40, 2769–2777 (2026). URL https://ojs.aaai.org/index.php/AAAI/article/view/37266.

[30] Haetinger, G. G., Tang, J., Ortiz, R., Kanyuk, P. & Azevedo, V. Controllable neural style transfer for dynamic meshes. In *ACM SIGGRAPH Conference Papers* (2024). URL https://studios.disneyresearch.com/2024/07/28/controllable-neural-style-transfer-for-dynamic-meshes/.

[31] Ho, J., Jain, A. & Abbeel, P. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, Vol. 33, 6840–6851 (2020). URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

[32] Rombach, R., Blattmann, A., Lorenz, D., Esser, P. & Ommer, B. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10684–10695 (2022). URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

[33] Podell, D. *et al.* SDXL: Improving latent diffusion models for high-resolution image synthesis. In *International Conference on Learning Representations* (2024). URL https://arxiv.org/abs/2307.01952.

[34] Poole, B., Jain, A., Barron, J. T. & Mildenhall, B. DreamFusion: Text-to-3d using 2d diffusion. In *International Conference on Learning Representations* (2023). URL https://arxiv.org/abs/2209.14988.

[35] Tang, J., Ren, J., Zhou, H., Liu, Z. & Zeng, G. DreamGaussian: Generative gaussian splatting for efficient 3d content creation. In *International Conference on Learning Representations* (2024). URL https://dreamgaussian.github.io/.

[36] Wang, Z. *et al.* ProlificDreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. In *Advances in Neural Information Processing Systems*, Vol. 36 (2023). URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1a87980b9853e84dfb295855b425c262-Abstract.html.

[37] Haque, A., Tancik, M., Efros, A. A., Holynski, A. & Kanazawa, A. Instruct-NeRF2NeRF: Editing 3d scenes with instructions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023). URL https://instruct-nerf2nerf.github.io/.

[38] Yang, Y., Wang, Y., Wang, C., Wang, H. & He, S. FantasyStyle: Controllable stylized distillation for 3d gaussian splatting. In *AAAI Conference on Artificial Intelligence*, Vol. 40, 11784–11792 (2026). URL https://ojs.aaai.org/

index.php/AAAI/article/view/38164.

[39] Yang, Y. *et al.* DiffStyle3D: Consistent 3d gaussian stylization via attention optimization. *arXiv preprint arXiv:2601.19717* (2026). URL https://arxiv.org/abs/2601.19717.

[40] He, M., Chen, Z. & Maddock, S. SIC3D: Style image conditioned text-to-3d gaussian splatting generation. *arXiv preprint arXiv:2604.08760* (2026). URL https://arxiv.org/abs/2604.08760.

[41] Choi, C., Lee, H., Jambon, C., Vinker, Y. & Kim, Y. M. Image-guided geometric stylization of 3d meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2026). URL https://arxiv.org/abs/2604.07795.

[42] Yu, X. *et al.* Text-to-3d with classifier score distillation. In *International Conference on Learning Representations* (2024). URL https://arxiv.org/abs/2310.19415.

[43] Hertz, A., Aberman, K. & Cohen-Or, D. Delta denoising score. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023). URL https://arxiv.org/abs/2304.07090.

[44] Simonyan, K. & Zisserman, A. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations* (2015). URL https://arxiv.org/abs/1409.1556.

[45] Ye, H., Zhang, J., Liu, S., Han, X. & Yang, W. IP-Adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arXiv:2308.06721* (2023). URL https://arxiv.org/abs/2308.06721.

[46] Radford, A. *et al.* Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, Vol. 139, 8748–8763 (2021). URL https://proceedings.mlr.press/v139/radford21a.html.

[47] Zhang, L., Rao, A. & Agrawala, M. Adding conditional control to text-to-image diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2023). URL https://arxiv.org/abs/2302.05543.

[48] Guo, Y. *et al.* AnimateDiff: Animate your personalized text-to-image diffusion models without specific tuning. In *International Conference on Learning Representations* (2024). URL https://arxiv.org/abs/2307.04725.

[49] Laine, S. *et al.* Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics* **39** (2020). URL https://arxiv.org/abs/2011.03277.

[50] Liu, I., Su, H. & Wang, X. Dynamic gaussians mesh: Consistent mesh reconstruction from dynamic scenes. In *International Conference on Learning Representations* (2025). URL https://openreview.net/forum?id=LuGHbK8qTa.

[51] Chen, J., An, J., Lyu, H., Kanan, C. & Luo, J. Learning to evaluate the artness of AI-generated images. *IEEE Transactions on Multimedia* **26**, 10731–10740 (2024). URL https://arxiv.org/abs/2305.04923.

[52] Wright, M. & Ommer, B. ArtFID: Quantitative evaluation of neural style transfer. In *German Conference on Pattern Recognition* (2022). URL https://arxiv.org/abs/2207.12280.

[53] Afifi, M., Brubaker, M. A. & Brown, M. S. HistoGAN: Controlling colors of GAN-generated and real images via color histograms. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7941–7950 (2021). URL https://arxiv.org/abs/2011.11731.

[54] Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018). URL https://arxiv.org/abs/1801.03924.

[55] Teed, Z. & Deng, J. RAFT: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision* (2020). URL https://arxiv.org/abs/2003.12039.

Funding

This work was supported by the Major Special Project of Changsha Science and Technology Plan under Grant kh2103016.

Author contributions

Yongang Yu conceived the study, developed the methodology, and implemented the code. Lingfeng Liu performed the experiments, analysed the results, and wrote the initial draft. Zhigang Chen supervised the research and reviewed the manuscript. All authors reviewed and approved the final manuscript.

Additional information

Competing interests

The authors declare no competing interests.

Ethics approval and consent to participate

The evaluations reported in this article are computational experiments using publicly available benchmark datasets. Ethics approval and consent to participate are not applicable to these computational evaluations.

Consent for publication

The character in Figure 4b is a synthetic three-dimensional model from the publicly available Jumping Jacks sequence (D-NeRF [3]), accessed through the dataset release of the DG-Mesh project at https://github.com/Isabella98Liu/DG-Mesh/releases/tag/datasets. It does not depict a real human participant and was not photographed specifically for this study. Consent for publication of identifiable participant images is not applicable to this synthetic character.