DiDE:*D*irect *I*njection with Color-Texture *DE*coupling for 3D Stylization

Tao Wu[1,2] **Alexandra Gomez-Villa**[1,2]
Senmao Li[3] **Yaxing Wang**[4] **Joost van de Weijer**[1,2] **Kai Wang**[5,6*]

[1]Computer Vision Center, Spain
[2]Universitat Autònoma de Barcelona, Spain
[3]Mohamed bin Zayed University of Artificial Intelligence, United Arab Emirates
[4]Jilin University, China
[5]Program of Computer Science, City University of Hong Kong (Dongguan), China
[6]City University of Hong Kong, HK SAR, China

Project page: `https://tttao-uwu.github.io/DiDE/`



Figure 1: Our *training-free* framework *DiDE* enables the generation of stylized 3D assets, with flexible control over the independent modulation of color or texture.

Abstract

Recent advances in rectified flow-based image-to-3D generative models have enabled high-fidelity 3D asset generation. Building on this, a growing line of work has exploited these strong 3D priors for training-free stylization, transferring visual attributes from a reference image onto a generated 3D asset. However, existing methods enforce an all-or-nothing paradigm: color and texture are transferred jointly, with no mechanism to control them independently — a limitation we formalize as *Disentangled 3D Stylization (Disen3D)*. To address this, we propose *DiDE*, the first training-free framework for *Disen3D*. Key to our approach is the observation that the structured latent space of image-to-3D models is overcomplete with respect to texture: texture information occupies only a small subset of the style-significant channels, leaving a free subspace available for independent color encoding. *DiDE* exploits this via a channel partition mechanism that processes a content image, a texture reference, and a color reference through dedicated branches and composes both style signals interference-free at every self-attention layer, preserving content geometry throughout. Experiments on Disen3D-Bench, our newly collected multi-reference benchmark, show that *DiDE* consistently outperforms 2D and 3D stylization baselines in color fidelity, texture transfer, and content preservation.

*Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

1 Introduction

Generative models for 3D asset creation have witnessed significant advancements [43, 44, 49], facilitating their widespread adoption across diverse domains, including entertainment [53], robotics [23] and healthcare [26]. In particular, recent rectified flow-based approaches [28, 29, 45], exemplified by TRELLIS [49, 50], have demonstrated the ability to generate high-fidelity 3D objects by jointly modeling sparse structural representations and appearance features within a unified latent space, while explicitly disentangling geometric structures from appearance attributes. Beyond 3D asset generation, a growing line of work has extended generative models to support *3D stylization* [18, 34, 42] — the task of transferring stylistic attributes from a reference image onto a generated 3D asset. By conditioning generation on a style reference, these methods allow the produced asset to inherit visual properties such as surface material, color palette, and structural texture, while remaining semantically aligned with a content image. This capability carries significant practical value: a single base geometry can be re-stylized for different artistic contexts, production pipelines, or brand identities without requiring manual re-authoring. Although existing stylized 3D generation methods [34, 42] have achieved promising results, they still exhibit a critical limitation: style attributes (such as texture and color) cannot be controlled independently, which severely restricts artistic freedom in practice.

In practice, content creators rarely want to adopt a single reference style. Instead, they often wish to selectively borrow specific components while preserving the core of their own artistic vision [10, 14]. Color palettes play an essential role in this process: they are carefully designed to convey specific emotions or maintain brand consistency, and a creator may wish to preserve their own color scheme while borrowing surface texture from a reference. Equally, one might want to adopt the rough, weathered texture of aged stone from one reference while keeping the warm tones of an entirely different palette. These two attributes — *color and texture* — are visually and semantically distinct, yet current 3D stylization approaches [18, 34] enforce an all-or-nothing paradigm: applying a reference style requires accepting all its visual attributes simultaneously, which severely restricts artistic freedom and practical applicability. To the best of our knowledge, only one prior work addresses disentangled stylization in 2D [33], yet it cannot be generalized to 3D, as style embeddings from separate references cannot simply be composed in 3D latent space. This leaves the problem of fine-grained, decoupled 3D stylization entirely open.

To address these limitations, we introduce *Disentangled 3D Stylization* (*Disen3D*), which aims to decompose reference styles into independently controllable color and texture attributes, enabling artists to selectively transfer specific style elements while preserving content geometry. We further propose Direct Injection with color-texture DEcoupling (*DiDE*), the first training-free framework for this task, which takes three inputs — a content image, a texture reference, and a color reference — processed through dedicated branches. To achieve this, We build on prior findings [34] showing that feature channels of image-to-3D models [50] can be decomposed into style- and content-significant subsets, with one branch dedicated to style and another to content. However, directly extending this framework with an additional branch for color is non-trivial: combining two style branches through the same attention pathway leads to feature interference, making disentangled control of texture and color difficult. To address this problem, we further show that texture information occupies only a subset of the style-significant channels, leaving additional feature capacity available for encoding an independent color attribute. We allocate those informative channels from the texture branch, while populating the remaining free channels with color features from a separate color reference branch. Finally, this partition is applied at every self-attention layer of the texture generation stage, enabling composable, interference-free injection of texture and color while preserving full 3D consistency.

Since no existing method addresses the disentangled stylization problem directly in 3D, we evaluate against complementary baselines on our proposed Disen3D-Bench benchmark: SADis [33], the only prior work on color-texture disentanglement but operating in 2D, and StyleSculptor [34] and MorphAny3D [42], which perform 3D stylization but without attribute disentanglement. In summary, our contributions are as follows:

- We introduce *Disentangled 3D Stylization* (*Disen3D*), a novel problem formulation that decomposes reference styles into independently controllable color and texture attributes, which is neglected in prior works.

- We are the *first* to identify and exploit the overcomplete nature of the structured latent space in image-to-3D models, demonstrating that texture information saturates only a small subset of channels, leaving a free subspace available for independent color and texture injection.
- We present Direct Inject with color-texture DEcoupling (*DiDE*), the first tuning-free *color-texture disentanglement* framework for 3D stylization, which exploits this overcomplete structure via a channel partition mechanism to enable composable, interference-free style control while preserving content geometry.
- Experimental results demonstrate that *DiDE* consistently outperforms existing 2D and 3D stylization baselines in color fidelity, texture transfer, and content preservation on our new collected multi-reference disentangled 3D stylization benchmark Disen3D-Bench.

2 Related Work

3D Generative Models. Two dominant paradigms have been established for high-quality 3D asset generation. The first focuses on distilling useful priors from pretrained 2D generative models into feed-forward 3D reconstruction algorithms. They either fine-tune 2D models to synthesize multi-view images [39, 40, 55], which are further reconstructed into 3D assets via techniques such as 3D Gaussian splatting [2, 24]; or directly guides 3D optimization through gradient signals from 2D diffusion models [32]. Despite their effectiveness, they lack an explicit latent 3D space to achieve fine-grained structural control. To overcome this limitation, the second paradigm pursues unified native 3D generative models trained from scratch. Inspired by the remarkable success of diffusion models in visual synthesis [15, 25, 38], recent methods [5, 36, 46, 56] learn compact latent representations and perform generation within this compressed latent space, significantly advancing 3D generative modeling. Benefiting from large-scale 3D datasets [6, 7], modern 3D foundation models [3, 17, 27, 45] have been developed to capture strong geometric priors, enabling not only high-quality generation across diverse object categories but also various downstream applications [9, 19]. However, most existing native 3D generators are restricted to either explicit representations or implicit formulations. TRELLIS [50] addresses this constraint by introducing a Structured Latent Representation (SLAT) that supports flexible multi-format 3D generation, whose versatility we inherit to accommodate diverse 3D modalities in our framework. Although these 3D foundation models exhibit strong generalization abilities, they mainly generate samples from a learned data distribution and lack dedicated mechanisms for example-driven control. In this work, we leverage the powerful priors from the 3D foundation model TRELLIS [50] and introduce a decoupled personalization framework. Our approach enables flexible adaptation of both geometry and appearance, while fully retaining the high-quality generative capacity of the underlying models.

Controllable 3D Generation. Recent methods have been developed to enable image-guided control for 3D generation [8, 11, 21, 47, 52]. Among these, some approaches inject explicit shape priors to regulate the geometric attributes of synthesized 3D content. In parallel, another line of work unifies the stylistic appearance of generated objects by leveraging reference images [37, 41, 51, 54]. However, such style-guided 3D generation pipelines typically require users to provide predefined 3D geometry such as meshes, and enforce fixed geometric structures throughout generation to maintain spatial consistency. This strong restriction confines their controllability to texture style only, while leaving geometric style—another critical dimension of 3D asset customization—unchanged and fully uncontrollable. In this work, we employ TRELLIS as our backbone model, taking advantage of its expressive 3D latent representations that comprehensively encode both geometric structure and textural appearance. Built on these latent features, we achieve zero-shot style-guided 3D generation that supports simultaneous control over both texture and geometric style.

Image-based 3D Stylization. Given a style reference input and a target content shape, 3D stylization is defined as the task of distilling stylistic patterns from the reference and transferring these patterns to the target geometry to synthesize stylistically aligned 3D assets. This problem has been investigated across diverse 3D representation frameworks, including explicit meshes [13, 16, 19], neural implicit fields, and Gaussian splatting [20, 22]. To enhance generalizability, a prominent line of recent work [1, 8, 51] harnesses powerful 2D priors from large-scale image diffusion models [38] for 3D stylization. Concurrently, StyleSculptor [34] leverages strong 3D priors [50] to achieve view-consistent stylization. However, all these approaches share a critical limitation: contemporary 3D stylization methods [18, 34] enforce an all-or-nothing paradigm, where adopting a reference

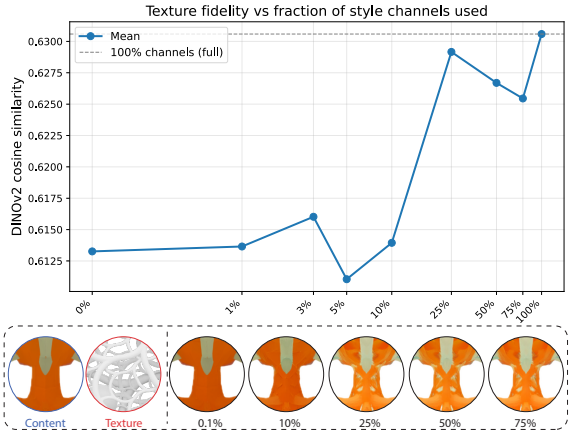

Figure 2: **Top:** DINOv2 texture similarity rises sharply up to 25% of channels retained, then plateaus — indicating that texture information is concentrated in a small channel subset. **Bottom:** Visual inspection shows that texture is reproduced beyond this threshold, with negligible improvement from additional channels.



Figure 3: Conventionally, the query is derived from the content image, with the key/value from the reference image. This leads to content bias. In contrast, using query/key/value all from the reference image improves style consistency.

style necessitates the simultaneous incorporation of all its visual attributes. This inherent constraint severely limits the methods' practical utility in real-world content creation scenarios.

3 Methods

To address the *Disen3D* problem, we propose a disentangled 3D style transfer framework *DiDE* built upon the TRELLIS backbone. Given a content image I_{cnt}, a texture reference I_{txu}, and a color reference I_{col}, our goal is to generate a 3D asset that (i) preserves the geometric structure of I_{cnt}, (ii) inherits texture from I_{txu}, and (iii) adopts color from I_{col}, while maintaining cross-view consistency. Unlike conventional 2D style transfer, our approach performs disentangled style modeling in the 3D latent space and injects style during rectified flow-based refinement.

3.1 Preliminaries

TRELLIS [50] is a foundation model for conditional 3D asset generation that synthesizes assets in multiple representations (e.g., NeRF, 3DGS, mesh) from text or image inputs. It adopts a two-stage rectified flow framework to progressively transform noise into structured 3D representations: Stage I predicts a sparse geometric structure, and Stage II refines per-voxel latent features for texture and material details.

Following this pipeline, we keep the first-stage sparse structure prediction unchanged to preserve geometric fidelity. Given an input image I_{cnt}, we use the DINOv2 model [30] to encode it into a latent $z_c = \mathcal{E}(I_{cnt})$. The Voxel Flow Transformer $\mathcal{VFT}$ then predicts a sparse voxel set

$$P = \{p_i\}_{i=1}^{L}, \quad p_i \in \{0, 1, \ldots, N-1\}^3 \tag{1}$$

where p_i is the positional index of an active voxel in the 3D grid intersecting with the surface of object, L is the total number of active voxels, and N is the maximum voxel index per dimension. We feed the coarse location information obtained in the first stage, along with the image embedding, into Stage II to generate features for each voxel. In the second stage, we complete the style transfer.

3.2 Overcompleteness of the TRELLIS Feature Space

StyleSculptor [34] showed that TRELLIS feature channels can be decomposed into style and content-relevant subsets via patch-wise variance. However, their formulation assumes a single style reference. In our setting, texture and color must be injected independently into the same style branch,

which can lead to feature interference. We therefore investigate whether the style-significant channels are overcomplete, asking whether all of them are actually required to faithfully encode texture. We conduct a channel ablation experiment in which we progressively reduce the number of channels used for texture injection at every DiT block of TRELLIS stage II, retaining only the top-k highest-variance channels from the style branch of Stage II. For each value of k, we generate outputs across the full set of style references from StyleSculptor [34] using a fixed content image, render a single view, and measure DINOv2 cosine similarity between the mean patch features of the rendered output and the corresponding style reference image. As shown in Fig. 2, DINOv2 texture similarity is ≈ 0.6125 when only a small fraction of channels is retained, rises to ≈ 0.625–0.63 once more than 25% of the channels are retained, and remains within this range up to 100%. To further support this finding, we conduct a PCA reconstruction analysis (see appendix), confirming the low intrinsic dimensionality of texture within the TRELLIS feature space, and validating that style-significant channels contain a large free subspace that can be independently allocated to a second style attribute.

3.3 Color-Texture Attributes Embedding

Given a content image I_{cnt}, a texture reference I_{txu}, and a color reference I_{col}, our framework processes each input through a dedicated branch. The content branch is kept unchanged from the standard image-to-3D pipeline [50] to preserve geometric structure, while the texture and color branches apply dedicated preprocessing to extract disentangled style embeddings, as described below.

Texture Reference. We convert the texture reference image I_{txu} into grayscale to suppress color information, yielding I_{txu}^{gray}. This grayscale image is then fed into the DINOv2 [30] encoder to obtain the texture embedding $f_{txu} = \mathcal{E}\big(I_{txu}^{\text{gray}}\big)$.

Color Reference. Our method supports two types of color reference inputs: *reference images*, common in style transfer applications, and *color palettes*, widely used in multimedia and design workflows. Both are processed to isolate color information from texture structure before encoding.

When a reference image is used as the color reference, we extract its low-frequency components via a Fourier transform while suppressing high-frequency signals. Intuitively, high-frequency components primarily encode textural details (e.g., edges and fine patterns), whereas low-frequency components capture global color distributions. By filtering out high-frequency signals, our method enforces a structure-invariant color conditioning, thereby improving the disentanglement between color and texture.

Given a color palette $\{\mathbf{c}_k\}_{k=1}^{n_{\text{colors}}}$, we construct a spatially varied image I_{vor} via Voronoi tessellation: we sample random seeds over the image plane, assign each a palette color, and map every pixel $\mathbf{x}$ to its nearest seed:

$$I_{\text{vor}}(\mathbf{x}) = \mathbf{c}_{a_{j^*(\mathbf{x})}}, \quad j^*(\mathbf{x}) = \arg\min_j \|\mathbf{x} - \mathbf{s}_j\|_2^2. \tag{2}$$

This produces an image that encodes the palette's color distribution without any texture structure. In both cases, the processed image is encoded via DINOv2 [30] to obtain the color embedding $f_{\text{col}} = \mathcal{E}(I_{\text{vor}})$.

3.4 Disentangled 3D Style Transfer (*Disen3D*)

DiDE runs three parallel branches through the transformer blocks of Stage II — one for the content image, one for the texture reference, and one for the color reference — each processing its input through independent self-attention layers. At every transformer block, the texture and color branches produce separate attention outputs, which are composed via a channel partition mechanism into a single fused representation. This fused output is then injected into the content branch, overwriting its self-attention output before the residual update. We describe each step in detail below.

Per-Branch Self-Attention. Given an input feature sequence $\mathbf{X} \in \mathbb{R}^{N \times d}$, a standard self-attention layer computes as:

$$\text{Attn}(\mathbf{X}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}, \quad \mathbf{Q} = \mathbf{X}W_Q,\ \mathbf{K} = \mathbf{X}W_K,\ \mathbf{V} = \mathbf{X}W_V. \tag{3}$$

Prior 3D stylization methods [34] adopt a cross-attention formulation in which the query is derived from the content image while keys and values come from the style reference. As shown in Fig. 3,

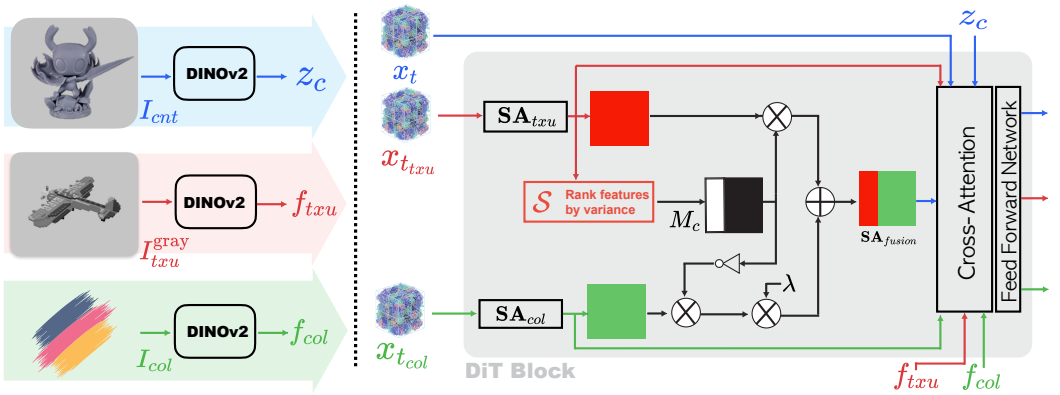

Figure 4: **Overview of *DiDE*.** *Left:* Our framework takes three inputs — a content image, a texture reference, and a color reference — processed through three independent branches: the content branch (blue), the texture branch (red), and the color branch (green). *Right:* At each DiT block, the texture and color branches produce independent self-attention outputs. A channel selection operator $\mathcal{S}$ partitions the feature channels into texture-allocated (high-variance) and color-allocated (low-variance) subsets, and the two attention outputs are composed via the binary mask $\mathbf{M_c}$ into a single fused representation $\mathbf{SA}_{\text{fusion}}$, which is then injected into the residual stream of the content branch through its gated update.

this leads to structural leakage: the content query biases the output toward the content geometry and weakens style transfer.

We instead run each style branch with its own standard self-attention, using query, key, and value all derived from the same branch input:

$$\mathbf{SA}_{txu} = \text{Attn}(\mathbf{Q}_{txu}, \mathbf{K}_{txu}, \mathbf{V}_{txu}), \tag{4}$$

$$\mathbf{SA}_{col} = \text{Attn}(\mathbf{Q}_{col}, \mathbf{K}_{col}, \mathbf{V}_{col}). \tag{5}$$

This ensures that each branch faithfully encodes the style of its own reference without contamination from the content image.

Texture-Color Composition. Building on our finding that texture information saturates only a subset of the style-significant channels (Sec. 3.2), we exploit the remaining free channels to encode an independent color attribute. This channel-wise decomposition is applied at every transformer block of Stage II, as illustrated in Fig. 4 (right).

We introduce a channel selection operator $\mathcal{S}$, where $\mathcal{S}(h) \subset \{1, \ldots, C\}$ denotes the indices of the top $k\%$ highest-variance channels of a feature map with C channels. These channels are allocated to the texture branch, while the complementary set is allocated to the color branch. We construct a binary mask $\mathbf{M} \in \{0, 1\}^C$:

$$M_c = \begin{cases} 1, & c \in \mathcal{S}(h), \\ 0, & \text{otherwise}, \end{cases} \tag{6}$$

and define the fused attention output as:

$$\mathbf{SA}_{\text{fusion}} = \text{Attn}(\mathbf{Q}_{txu}, \mathbf{K}_{txu}, \mathbf{M} \odot \mathbf{V}_{txu}) + \lambda \, \text{Attn}(\mathbf{Q}_{col}, \mathbf{K}_{col}, (1 - \mathbf{M}) \odot \mathbf{V}_{col}), \tag{7}$$

where $\odot$ denotes element-wise multiplication broadcast along the sequence dimension, and $\lambda > 0$ is a scalar hyperparameter that controls the trade-off between the color and texture branch.

Injection into the Content Branch. The fused representation $\mathbf{SA}_{\text{fusion}}$ is injected into the content branch by replacing its self-attention output at each transformer block. Following the standard gated residual mechanism of the DiT blocks in TRELLIS, the content features are updated as:

$$\mathbf{X}'_{cnt} = \mathbf{X}_{cnt} + \alpha_{cnt} \odot \mathbf{SA}_{\text{fusion}}, \tag{8}$$

where $\alpha_{cnt} \in \mathbb{R}^d$ is the learned per-channel gate derived from the content branch's adaptive layer norm conditioning, following the AdaLN-zero parameterization [31]. This preserves the spatial structure and diffusion conditioning of the content branch while fully transferring the composed style signal, keeping the generated geometry faithful to the content image throughout the generation.



Figure 5: Comparison with existing 3D stylization methods and adapted baselines for the *Disen3D* problem. The first two rows show results with a single style reference, while the remaining rows show results with disentangled texture and color references.



Figure 6: Zoomed-in comparison of the results. The enlarged views provide a closer inspection of geometry preservation and the disentangled transfer of texture and color.

4 Experiments

4.1 Experimental Setup

We conduct extensive experiments to validate three core claims of *DiDE*: (i) effective disentanglement of color and texture in the generated 3D assets, (ii) faithful preservation of the content geometry, and (iii) competitive style fidelity against existing methods. Since no prior benchmark addresses the *Disen3D* setting, we introduce Disen3D-Bench and compare against representative 2D and 3D stylization baselines.

Datasets. We collected 37 images to form a novel multi-image style transfer dataset named Disen3D-Bench for the *Disen3D* setup. These images include 10 content images, 12 texture reference images, and 15 palette images. The content images and texture reference images were selected from Stylesculptor [34] and Morph3D [42], and each palette image contains 3-5 different colors. These image combinations generate 1,800 3D assets for subsequent evaluation.

Table 1: Quantitative comparisons with existing style transfer methods. The best and second-best results are highlighted in **bold** and underlined, respectively. CD and HD denote Chamfer Distance and Hausdorff Distance, respectively.

Method	Geometry			Color			Texture		
	CLIP-I↑	CD$_{\times 10^2}$↓	HD$_{\times 10^2}$↓	MS-SWD↓	C-Hist↓	VLM↑	LPIPS↓	SSIM$_{\times 10^3}$↑	VLM↑
SADis [33]	0.73	18.17	41.59	18.38	0.99	37.17	**0.65**	116.10	18.00
StyleSculptor [34]	0.74	17.91	8.38	21.26	0.89	35.44	0.69	124.61	23.22
MorphAny3D [42]	0.71	198.52	44.50	**18.31**	0.91	40.05	**0.65**	117.21	19.72
DiDE (Ours)	**0.75**	**15.38**	**7.87**	20.40	**0.87**	**70.94**	0.68	**125.19**	**30.72**

Implementation Details. We mainly build our method *DiDE* upon TRELLIS [50] while also verified on TRELLIS.2 [49]. For the hyperparameters, we set $k = 25, \lambda = 1.2$. Both stages use a CFG strength of 3.0, with 25 inference steps. The reproducing code will be released after the peer review process. All experiments are conducted on a NVIDIA A40 GPU.

Baseline Methods. We compare against three representative training-free stylization approaches, each adapted to the *Disen3D* two-reference setting to the best extent their architectures permit. SADis [33] is the only prior method addressing color-texture disentanglement, though in 2D: we apply it to the content image and lift the stylized result to 3D using TRELLIS. MorphAny3D [42] natively supports two image inputs via its dual-target morphing setting, which we use to provide the texture and color references directly. StyleSculptor [34], originally designed for a single content-texture pair, is adapted by adding an extra reference branch for the color reference; the self-attention outputs of the texture and color branches are then summed to produce the fused style representation. We emphasize that StyleSculptor and MorphAny3D were not designed for disentangled two-reference stylization; these adaptations represent the closest reasonable comparison points rather than intended use cases. SADis, while designed for color-texture disentanglement, operates in 2D and requires an additional lifting step to produce 3D assets. Additional details on the experimental setup are provided in the appendix.

Evaluation Metrics. We adopt several quantitative metrics across four axes: (1) For *Geometry Semantics*, CLIP-I [35] measures semantic similarity between the generated asset and the content image; and Chamfer Distance and Hausdorff Distance are used to measure the geometric difference between two assets. (2) For *Texture preservation*: LPIPS [57] captures perceptual distortion and potential style leakage; and SSIM [48] evaluates structural similarity. (3) For *color fidelity*: MS-SWD [12] measures color distance between the generated asset and the color reference, and C-Hist computes the distance between color histograms. (4) Finally, we employ *VLM*-based evaluation using InternVL2.5-8B [4], which performs binary judgments on texture consistency (whether the generated asset shares structural style with the texture reference, ignoring color) and color fidelity (whether colors absent from the color reference are introduced). Full prompts and scoring protocol are provided in Appendix B. All metrics are computed over 1,800 generated assets.

4.2 Experimental Results

Qualitative Results. In our evaluation of generation quality, visual results are presented in Fig. 5 and Fig. 6. For SADis, the method exhibits subpar object detail preservation, while introducing redundant textures and color artifacts in the background region. After 3D rendering, the reconstructed textures and colors also fail to align closely with the reference images. For StyleSculptor, the model fails to achieve full disentanglement of texture and color. The generated 3D assets exhibit a mixed color distribution derived from both reference images; furthermore, the results often suffer from style bias, tending to favor either the texture reference or the color reference rather than achieving balanced, independent control over both attributes. Regarding MorphAny3D, its Morphing Cross-Attention mechanism causes the colors and styles inherent in the content image to exert a significant influence on the final generation result. Additionally, the method generates spurious colors that are not present in the provided color palette images. In contrast, our *DiDE* enables precise decoupling of texture structure and color attributes. For instance, we can accurately align the cutout structure of the texture reference with the target color— a capability not achievable by other baseline methods. Simultaneously, our method *DiDE* ensures that the generated 3D asset remains consistent with the content image, free from distortions, flips, or other visual artifacts.



Figure 7: Ablation on grayscale conversion and Voronoi tessellation preprocessing.



Figure 8: *DiDE* applied to the TRELLIS.2 backbone, showing generalization across model variants.



Figure 9: Ablation on channel selection variance source: texture vs. color branch.

Quantitative Results. Table 1 reports the full results. *DiDE* achieves the best CLIP-I score (0.75), indicating superior semantic alignment with the content image. For Geometry, *DiDE* achieves the lowest Chamfer (15.38) and Hausdorff (7.87) distances, improving over StyleSculptor by approximately 14.1% and 6.1%, respectively. Although SADis obtains a comparable Chamfer distance, its substantially higher Hausdorff distance reveals larger worst-case local deviations, showing that *DiDE* better preserves both global surface structure and local geometry. For color fidelity, *DiDE* leads on C-Hist (0.87) and by a large margin on VLM color accuracy (70.94 vs. 40.05 for the next best method), confirming faithful and disentangled color transfer. While MorphAny3D and SADis yield slightly better MS-SWD scores (18.31 and 18.38 vs. 20.40), this comes at the cost of degraded texture structure, as reflected in their lower SSIM scores. For texture quality, *DiDE* attains the highest SSIM (125.19) and VLM texture accuracy (30.72 vs. 23.22 for StyleSculptor), demonstrating superior structural preservation and texture fidelity. LPIPS scores are comparable across all methods (0.65–0.69), suggesting similar levels of perceptual distortion relative to the content image.

4.3 Ablation Study

Effect of Branch Preprocessing. We ablate the two key preprocessing components of *DiDE* — grayscale conversion of the texture reference and Voronoi tessellation of the color palette — by removing each in turn while keeping the rest of the pipeline unchanged. As shown in Fig. 7, removing grayscale conversion leaves residual color information in the texture branch, introducing undesired color interference in the generated output. Removing Voronoi tessellation causes color leakage, producing colors that deviate from the reference palette.

Model Adaptability. To verify the generalizability of *DiDE* beyond the base TRELLIS backbone, we apply it to TRELLIS.2 [49], which extends TRELLIS with a three-stage pipeline where texture and color features are processed in separate stages. As illustrated in Fig. 8, *DiDE* maintains robust and consistent disentanglement performance on TRELLIS.2, confirming that our channel partition mechanism transfers across model variants. The natural stage-wise separation in TRELLIS.2 further aligns with our texture-color decoupling design, making the adaptation straightforward.

Channel Selection. In *DiDE*, channel variance is computed from the texture branch to determine the channel partition. To assess the sensitivity of this design choice, we ablate by computing variance from the color branch instead, swapping which branch drives the partition. As shown in Fig. 9, both settings achieve consistent texture-color disentanglement, with only minor differences in color saturation. This robustness to the choice of variance source corroborates our earlier finding (Sec. 3.2) that the latent feature space is overcomplete, with texture and color information redundantly distributed across channels regardless of which branch is used for estimation. We use texture-branch variance as default since it produces marginally more faithful color rendition in practice.

Hyperparameter λ. We ablate the color branch weight λ in Eq. 7 by sweeping over the range $[0.5, 2.0]$. All results are averaged over 600 randomly samples. As reported in Table 2, too small a value of λ weakens color transfer fidelity, while too large a value begins to compromise texture quality. This reflects the expected trade-off between the two branches, and $\lambda = 1.2$ strikes the best balance, achieving the lowest C-Hist (0.89) and MS-SWD (19.25) while maintaining competitive texture scores.

Table 2: Ablation on color branch weight λ.

λ	Color		Texture	
	MS-SWD↓	C-Hist↓	LPIPS↓	SSIM$_{\times 10^3}$↑
0.5	20.24	0.92	0.68	83.36
1.0	19.47	0.89	0.69	83.34
1.2	19.25	0.89	0.69	83.45
1.5	19.32	0.90	0.70	83.30
2.0	19.38	0.90	0.70	83.40

User Study. Our method received the highest preference rates across all three aspects, with 51.7% for geometry preservation, 58.2% for texture consistency, and 72.4% for color consistency. These results provide complementary perceptual evidence that the quantitative improvements of our method translate into noticeable visual benefits, including better preservation of content geometry and more faithful, disentangled transfer of texture and color.

Table 3: User study results on geometry, texture, and color preference. Higher is better.

Method	Geometry ↑	Texture ↑	Color ↑
MorphAny3D	4.4%	1.2%	2.3%
SADiS	12.1%	3.2%	8.7%
StyleSculptor	31.8%	37.4%	16.6%
Ours	**51.7%**	**58.2%**	**72.4%**

Failure Cases. We further show a failure case in Fig. 10. Our method is sensitive to the luminance distribution of the texture reference. While the original texture reference produces reasonable texture and color transfer, increasing its brightness by $1.5\times$ causes the generated 3D assets to become nearly white, with substantial loss of both texture and color details. This indicates that large brightness shifts in the reference image remain a limitation of our current appearance-transfer formulation.



Figure 10: Failure case under texture-reference brightness shifts.

5 Conclusion

This work addresses a fundamental limitation in existing 3D stylization methods: the inability to independently control distinct stylistic attributes such as color and texture. We formalize this gap as *Disentangled 3D Stylization* (*Disen3D*) and propose Direct Injection with color-texture DEcoupling (*DiDE*), the first training-free framework to address it. *DiDE* decouples texture and color by processing each reference through a dedicated branch with tailored preprocessing and composes them via a channel partition mechanism injected into the self-attention layers of a pretrained image-to-3D model, preserving content geometry throughout. Extensive qualitative and quantitative evaluations on Disen3D-Bench confirm that *DiDE* outperforms state-of-the-art baselines in disentangled color and texture control. We hope this work opens the door to finer-grained artistic control in 3D content creation and inspires future research on multi-attribute style disentanglement.

6 Acknowledgments

Kai Wang acknowledges the financial support by Department of Science and Technology of Guangdong Province (No.2025QN12X071), funding No.R002026G0116 and No.R002026B0121 from Guangdong and Hong Kong Universities 1+1+1 Joint Research Collaboration Scheme, No.TIC2026SP005 from Seed Project under City University of Hong Kong (Dongguan) Technology and Innovation Center, NSFC Grant No.62606428 and the start-up grant B01040000108 from CityU-DG. Joost van de Weijer acknowledges Grant PID2025-175937NB-I00 and Grant AIA2025-163919-C52, both funded by MICIU/AEI/10.13039/501100011033 and the Càtedra ENIA UAB-Cruïlla (TSI-100929-2023- 2) from the Ministry of Economic Affairs and Digital Transition of Spain. Alexandra Gomez Villa acknowledges the EuroHPC Joint Undertaking for awarding access to the Leonardo supercomputer at CINECA, Italy, under project EHPC-AIF-2026SC01-004.

References

[1] Qimin Chen, Yuezhi Yang, Yifan Wang, Vladimir Kim, Siddhartha Chaudhuri, Hao Zhang, and Zhiqin Chen. Art-deco: Arbitrary text guidance for 3d detailizer construction. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

[2] Yulong Chen, Xiaoyun Dong, Haoyu Zhang, Zongxian Yang, Lewei Xie, Xinke Li, Yifan Zhang, Kai Wang, and Jianping Wang. One world, dual timeline: Decoupled spatio-temporal gaussian scene graph for 4d cooperative driving reconstruction. *arXiv preprint arXiv:2605.07910*, 2026.

[3] Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 26576–26586, 2025.

[4] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[5] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4456–4465, 2023.

[6] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023.

[7] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13142–13153, 2023.

[8] Wenqi Dong, Bangbang Yang, Lin Ma, Xiao Liu, Liyuan Cui, Hujun Bao, Yuewen Ma, and Zhaopeng Cui. Coin3d: Controllable and interactive 3d assets generation with proxy-guided conditioning. In *ACM SIGGRAPH 2024 Conference Papers*, pages 1–10, 2024.

[9] Keyu Du, Jingyu Hu, Haipeng Li, Hao Xu, Haibin Huang, Chi-Wing Fu, and Shuaicheng Liu. Hierarchical neural semantic representation for 3d semantic correspondence. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–11, 2025.

[10] Ziv Epstein, Aaron Hertzmann, Investigators of Human Creativity, Memo Akten, Hany Farid, Jessica Fjeld, Morgan R Frank, Matthew Groh, Laura Herman, Neil Leach, et al. Art and the science of generative ai. *Science*, 380(6650):1110–1111, 2023.

[11] Junyao Gao, Jiaxing Li, Wenran Liu, Yanhong Zeng, Fei Shen, Kai Chen, Yanan Sun, and Cairong Zhao. Charactershot: Controllable and consistent 4d character animation. *arXiv preprint arXiv:2508.07409*, 2025.

[12] Jiaqi He, Zhihua Wang, Leon Wang, Tsein-I Liu, Yuming Fang, Qilin Sun, and Kede Ma. Multiscale sliced wasserstein distances as perceptual color difference measures. *European Conference on Computer Vision*, 2024.

[13] Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. Deep geometric texture synthesis. *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 2020.

[14] Aaron Hertzmann. Toward modeling creative processes for algorithmic painting. *ICCC*, 2022.

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

[16] 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*, pages 6198–6208, 2022.

[17] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *International Conference on Learning Representations*, 2024.

[18] Jingyu Hu, Bin Hu, Ka-Hei Hui, Haipeng Li, Zhengzhe Liu, Daniel Cohen-Or, and Chi-Wing Fu. Pegasus: 3d personalization of geometry and appearance. *arXiv preprint arXiv:2602.08198*, 2026.

[19] Jingyu Hu, Ka-Hei Hui, Zhengzhe Liu, Ruihui Li, and Chi-Wing Fu. Neural wavelet-domain diffusion for 3d shape generation, inversion, and manipulation. *ACM transactions on graphics*, 43(2):1–18, 2024.

[20] Junchao Huang, Xinting Hu, Shaoshuai Shi, Zhuotao Tian, and Li Jiang. Edit360: 2d image edits to 3d assets from any angle. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16618–16628, 2025.

[21] Tianyu Huang, Yihan Zeng, Zhilu Zhang, Wan Xu, Hang Xu, Songcen Xu, Rynson WH Lau, and Wangmeng Zuo. Dreamcontrol: Control-based text-to-3d generation with 3d self-prior. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5364–5373, 2024.

[22] 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*, pages 18342–18352, 2022.

[23] Tsung-Wei Ke, Nikolaos Gkanatsios, and Katerina Fragkiadaki. 3d diffuser actor: Policy diffusion with 3d scene representations. *CoRL*, 2024.

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

[25] Levon Khachatryan, Andranik Movsisyan, Vahram Tadevosyan, Roberto Henschel, Zhangyang Wang, Shant Navasardyan, and Humphrey Shi. Text2video-zero: Text-to-image diffusion models are zero-shot video generators. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15954–15964, 2023.

[26] Firas Khader, Gustav Müller-Franzes, Soroosh Tayebi Arasteh, Tianyu Han, Christoph Haarburger, Maximilian Schulze-Hagen, Philipp Schad, Sandy Engelhardt, Bettina Baeßler, Sebastian Foersch, et al. Denoising diffusion probabilistic models for 3d medical image generation. *Scientific reports*, 13(1):7303, 2023.

[27] Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. Triposg: High-fidelity 3d shape synthesis using large-scale rectified flow models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *International Conference on Learning Representations*, 2023.

[29] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. *International Conference on Learning Representations*, 2023.

[30] Maxime Oquab, Timothée Darcet, Theo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, Russell Howes, Po-Yao Huang, Hu Xu, Vasu Sharma, Shang-Wen Li, Wojciech Galuba, Mike Rabbat, Mido Assran, Nicolas Ballas, Gabriel Synnaeve, Ishan Misra, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. Dinov2: Learning robust visual features without supervision, 2023.

[31] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4195–4205, 2023.

[32] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. DreamFusion: Text-to-3D using 2D Diffusion. *ICLR*, 2023.

[33] Jiang Qin, Senmao Li, Alexandra Gomez-Villa, Shiqi Yang, Yaxing Wang, Kai Wang, and Joost van de Weijer. Free-lunch color-texture disentanglement for stylized image generation. *nips*, 2025.

[34] Zefan Qu, Zhenwei Wang, Haoyuan Wang, Ke Xu, Gerhard Petrus Hancke, and Rynson WH Lau. Stylesculptor: Zero-shot style-controllable 3d asset generation with texture-geometry dual guidance. In *Proceedings of the SIGGRAPH Asia 2025 Conference Papers*, pages 1–12, 2025.

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

[36] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4209–4219, 2024.

[37] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In *ACM SIGGRAPH 2023 conference proceedings*, pages 1–11, 2023.

[38] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10684–10695, 06 2022.

[39] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *International Conference on Learning Representations*, 2024.

[40] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *3DV*, 2025.

[41] Bingjie Song, Xin Huang, Ruting Xie, Xue Wang, and Qing Wang. Style3d: Attention-guided multi-view style transfer for 3d object generation. *arXiv preprint arXiv:2412.03571*, 2024.

[42] Xiaokun Sun, Zeyu Cai, Hao Tang, Ying Tai, Jian Yang, and Zhenyu Zhang. Morphany3d: Unleashing the power of structured latent in 3d morphing. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2026.

[43] Tencent Hunyuan3D Team. Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation, 2024.

[44] Tencent Hunyuan3D Team. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation, 2025.

[45] Tencent Hunyuan3D Team. Hunyuan3d 2.1: From images to high-fidelity 3d assets with production-ready pbr material, 2025.

[46] Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, Karsten Kreis, et al. Lion: Latent point diffusion models for 3d shape generation. *Advances in neural information processing systems*, 35:10021–10039, 2022.

[47] Zhenwei Wang, Tengfei Wang, Gerhard Hancke, Ziwei Liu, and Rynson WH Lau. Themestation: Generating theme-aware 3d assets from few exemplars. In *Acm siggraph 2024 conference papers*, pages 1–12, 2024.

[48] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

[49] Jianfeng Xiang, Xiaoxue Chen, Sicheng Xu, Ruicheng Wang, Zelong Lv, Yu Deng, Hongyuan Zhu, Yue Dong, Hao Zhao, Nicholas Jing Yuan, and Jiaolong Yang. Native and compact structured latents for 3d generation. *Tech report*, 2025.

[50] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025.

[51] Zhiyu Xie, Yuqing Zhang, Xiangjun Tang, Yiqian Wu, Dehan Chen, Gongsheng Li, and Xiaogang Jin. Styletex: Style image-guided texture generation for 3d models. *ACM Transactions on Graphics (TOG)*, 43(6):1–14, 2024.

[52] Jiale Xu, Xintao Wang, Weihao Cheng, Yan-Pei Cao, Ying Shan, Xiaohu Qie, and Shenghua Gao. Dream3d: Zero-shot text-to-3d synthesis using 3d shape prior and text-to-image diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20908–20918, 2023.

[53] Yongzhi Xu, Yonhon Ng, Yifu Wang, Inkyu Sa, Yunfei Duan, Zhenhong Sun, Yang Li, Pan Ji, and Hongdong Li. Sketch2scene: Automatic generation of interactive 3d game scenes from user's casual sketches. *arXiv preprint arXiv:2408.04567*, 2024.

[54] Yu-Ying Yeh, Jia-Bin Huang, Changil Kim, Lei Xiao, Thu Nguyen-Phuoc, Numair Khan, Cheng Zhang, Manmohan Chandraker, Carl S Marshall, Zhao Dong, et al. Texturedreamer: Image-guided texture synthesis through geometry-aware diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4304–4314, 2024.

[55] Wangbo Yu, Jinbo Xing, Li Yuan, Wenbo Hu, Xiaoyu Li, Zhipeng Huang, Xiangjun Gao, Tien-Tsin Wong, Ying Shan, and Yonghong Tian. Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.

[56] Xianfang Zeng, Xin Chen, Zhongqi Qi, Wen Liu, Zibo Zhao, Zhibin Wang, Bin Fu, Yong Liu, and Gang Yu. Paint3d: Paint anything 3d with lighting-less texture diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4252–4262, 2024.

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

A Statements

Limitations. While *DiDE* effectively mitigates detail attenuation and enhances fine-grained texture fidelity for training-free disentangled 3D stylization, it has several limitations: the framework is only validated on rectified flow-matching based native 3D diffusion models (the TRELLIS/TRELLIS.2 family) and has not been tested for generalization to other mainstream 3D generation paradigms, including voxel-based models, signed distance field (SDF) generators, 3D Gaussian splatting diffusion pipelines, and knowledge distillation-based image-to-3D methods. Furthermore, although the computational overhead is negligible, the low-noise restart refinement step adds mild extra inference operations, restricting direct deployment on extremely memory-constrained edge devices. Finally, *DiDE* focuses on color-texture decoupling and texture enhancement without a dedicated geometric refinement module, so the geometric quality of the generated 3D assets is inherently dependent on the geometric reconstruction capability of the underlying pre-trained 3D foundation model; and the current design only supports independent control of color and texture, without extending to multi-attribute disentanglement (e.g., material, lighting, geometric style), which limits its flexibility for complex artistic creation.

Broader Impacts. *DiDE* advances controllable 3D content creation and delivers widespread practical value by enabling training-free, disentangled color-texture control for 3D stylization, which drastically lowers the technical barrier and production cost for generating high-fidelity and customizable 3D assets, benefiting diverse industries and scenarios including game development, film and television virtual production, industrial design, cultural heritage digitization, robotic perception, and metaverse content construction, empowering both professional creators and non-expert users to flexibly customize 3D assets with independent style control, improving creative efficiency and diversifying content supply; on the other hand, the strong detail-preserving and high-fidelity texture transfer capability of *DiDE* introduces potential intellectual property and ethical risks, as the model could be misused for unauthorized duplication, counterfeiting, or commercial exploitation of copyrighted objects, branded products, real-world artifacts, or cultural heritage items without permission, and to address such risks, we advocate for the responsible deployment of the technology, enforcing content authentication for generated assets, establishing clear usage norms and legal compliance guidelines, and promoting ethical adoption across industries to ensure that *DiDE* creates public value while minimizing harms from misuse.

Ethical Statement. We acknowledge the potential ethical implications associated with generative image-to-3D technologies, including risks related to privacy, impersonation, and misuse of synthetic 3D assets. All models used in this work are trained on publicly available datasets and follow the usage policies of those datasets. To promote transparency and responsible research, we will release the implementation details necessary to reproduce our results. We encourage researchers and practitioners to use *DiDE* responsibly and to consider the broader societal implications when deploying image-to-3D synthesis .

Reproducibility Statement. To ensure reproducibility, we will release the source code and training scripts required to reproduce the experimental results reported in this paper after the peer review process. All experiments are conducted using publicly available datasets, and detailed descriptions of the model architecture, training configuration, and evaluation procedures are provided in the main paper and appendices.

Inference efficiency Table 4 compares the average inference time per sample. Our method is the fastest among the evaluated methods, requiring 21.96 seconds per sample, a 13.5% reduction compared to the fastest baseline, SADis.

Table 4: **Inference time comparison.** Average runtime in seconds per sample. Lower is better.

Method	MorphAny3D	SADis	StyleSculptor	**Ours**
Time (s) ↓	64.70	25.39	39.76	**21.96**

B VLM-based Evaluation Protocol

Overview. To quantitatively evaluate the quality of generated images, we employ the InternVL2.5-8B [4] model as an automatic evaluator. The vision-language model (VLM) performs binary judgments on two complementary aspects:

- **Style Consistency**: structural similarity between images while being invariant to color differences.
- **Color Fidelity**: whether the generated image faithfully preserves the reference color distribution without introducing extraneous colors.

This enables semantic-level evaluation without requiring human annotation.

Input Construction. Given a generated image I_{gen}, we construct paired inputs with different references:

- I_{txu}: texture reference image
- I_{col}: color reference image

Each image is converted into a sequence of visual tokens (patches) using a vision encoder. Let:

$$V_{\text{gen}},\ V_{\text{txu}},\ V_{\text{col}} \tag{9}$$

denote the corresponding token representations. We concatenate tokens as:

$$V = [V_{\text{ref}}, V_{\text{gen}}] \tag{10}$$

and provide the number of patches per image to preserve image boundaries:

$$\texttt{num_patches_list} = [|V_{\text{ref}}|, |V_{\text{gen}}|] \tag{11}$$

Style Consistency Evaluation. To assess whether the generated image preserves structural/style characteristics, we prompt the VLM with:

```
Image-1: <Texture image>
Image-2: <Generation image>
Ignoring color differences, are the structures or styles of these
two images similar? Answer yes or no.
```

The input pair is $(I_{\text{txu}}, I_{\text{gen}})$, and the model outputs:

$$y_{\text{style}} \in \{\text{yes}, \text{no}\} \tag{12}$$

We define the style consistency score over N samples as:

$$S_{\text{style}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(y_{\text{style}}^{(i)} = \text{yes}\right) \tag{13}$$

Color Novelty Evaluation. To evaluate whether new colors are introduced, we use the prompt:

```
Image-1: <Color image>
Image-2: <Generation image>
If the second image shows a color that is not present in the first
image, answer yes; otherwise, answer no.
```

The input pair is $(I_{\text{col}}, I_{\text{gen}})$, and the output is:

$$y_{\text{color}} \in \{\text{yes}, \text{no}\} \tag{14}$$

We compute the color novelty score as:

$$S_{\text{color}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}\left(y_{\text{color}}^{(i)} = \text{no}\right) \tag{15}$$

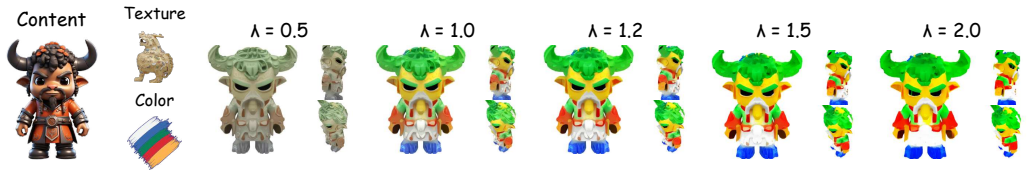

Figure 11: Ablation on λ.



Figure 12: Ablation on k.

C Hyperparameter visualization

We further analyze the influence of the hyperparameter λ in Equ. 7 on disentangled generation quality. Specifically, λ controls the contribution of the color branch during feature fusion. We evaluate five settings, $\lambda \in 0.5, 1.0, 1.2, 1.5, 2.0$, and visualize the corresponding results in Fig. 11. As shown in the figure, smaller values of λ (e.g., 0.5) lead to insufficient color guidance, causing the generated 3D assets to inherit excessive appearance information from the texture branch. In contrast, excessively large values of λ (e.g., 1.5 or 2.0) overemphasize color features, which weakens texture fidelity and introduces structural inconsistencies. We observe that $\lambda = 1.2$ achieves the best balance between texture preservation and color controllability. Under this setting, the generated results faithfully maintain the structural patterns from the texture reference while accurately matching the target color distribution, leading to the most visually coherent disentangled generation results.

D Ablation on k

We conduct an ablation study by varying $k \in \{0, 25, 50, 75, 100\}$, where only the top-$k$ channels are retained and the features of the remaining channels are set to zero. Fig. 12 shows that even when $k = 25$, the generated 3D assets already exhibit clearly recognizable texture patterns, indicating that a relatively small subset of channels is sufficient to preserve most of the texture information, which further demonstrates the overcompleteness of the TRELLIS feature space.

E Additional Visualization Results

we provide additional qualitative results to further demonstrate the effectiveness and robustness of our method across diverse texture and color references. As shown in Fig. 13, our method consistently achieves effective disentanglement between texture structure and color attributes under various challenging scenarios. Specifically, the generated 3D assets faithfully preserve the structural patterns and geometric details from the texture reference while accurately matching the target color distribution from the color reference image. Moreover, our method maintains strong content consistency without introducing noticeable distortions, structural artifacts, or undesired color leakage. These additional visualizations further validate the generalization capability and stability of *DiDE* for controllable disentangled 3D generation.

F PCA Reconstruction Analysis

To further validate the overcompleteness claim of Sec. 3.2, we perform a PCA analysis on the style-significant channels of the TRELLIS Stage II features. We project the style branch features onto an increasing number of principal components and measure DINOv2 cosine similarity between the rendered output and the texture reference. As shown in Fig. 14, similarity stabilizes after retaining as few as 10 principal components, despite the full channel dimensionality being much larger. This confirms that texture occupies an extremely low-dimensional subspace, leaving the majority of the feature space free to be independently repurposed for color encoding — directly motivating our channel partition design.

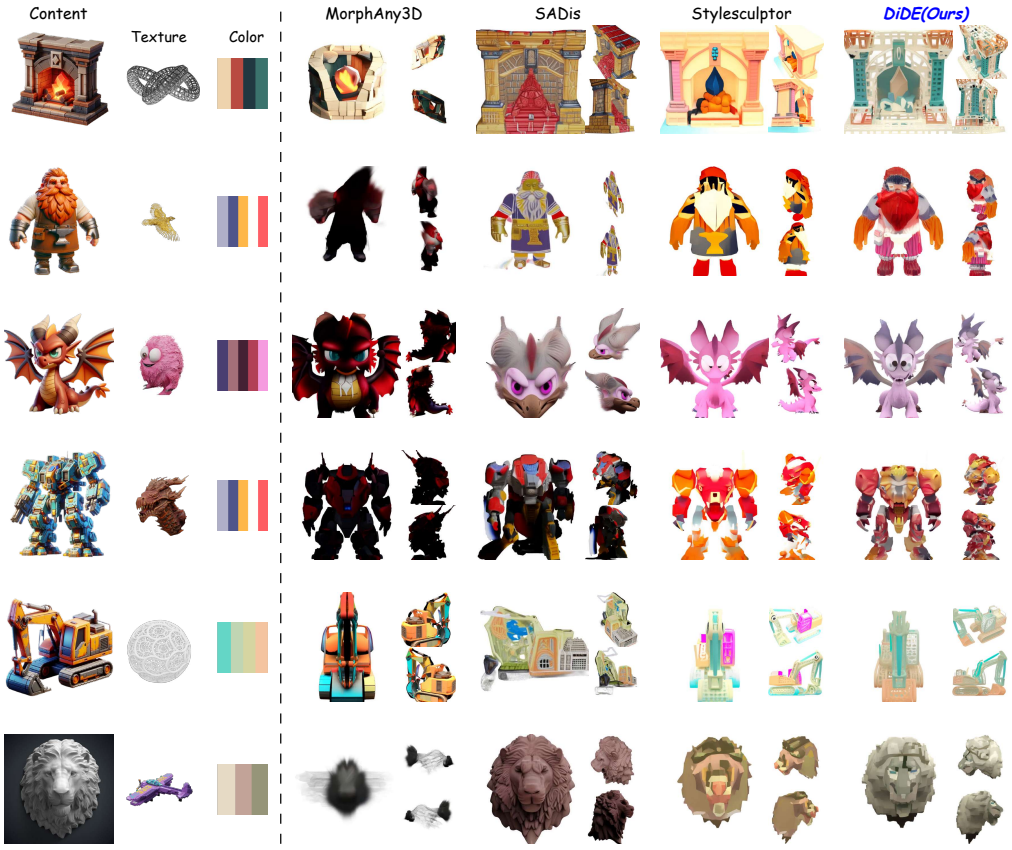

Figure 13: Additional qualitative comparisons

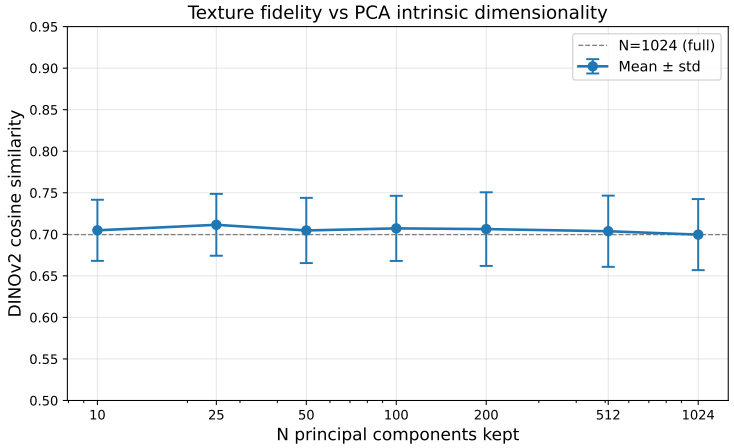

Figure 14: DINOv2 cosine similarity between rendered outputs and texture references as a function of the number of PCA components retained from TRELLIS texture-significant features. Similarity stabilizes after as few as 10 components, confirming the low intrinsic dimensionality of texture within the feature space.