
GuideFlow3D: Optimization-Guided Rectified Flow For Appearance Transfer

Sayan Deb Sarkar
Stanford University

Sinisa Stekovic
ENPC, IP Paris

Vincent Lepetit
ENPC, IP Paris

Iro Armeni
Stanford University

Abstract

Transferring appearance to 3D assets using different representations of the appearance object—such as images or text—has garnered interest due to its wide range of applications in industries like gaming, augmented reality, and digital content creation. However, state-of-the-art methods still fail when the geometry between the input and appearance objects is significantly different. A straightforward approach is to directly apply a 3D generative model, but we show that this ultimately fails to produce appealing results. Instead, we propose a principled approach inspired by universal guidance. Given a pretrained rectified flow model conditioned on image or text, our training-free method interacts with the sampling process by periodically adding guidance. This guidance can be modeled as a differentiable loss function, and we experiment with two different types of guidance including part-aware losses for appearance and self-similarity. Our experiments show that our approach successfully transfers texture and geometric details to the input 3D asset, outperforming baselines both qualitatively and quantitatively. We also show that traditional metrics are not suitable for evaluating the task due to their inability of focusing on local details and comparing dissimilar inputs, in absence of ground truth data. We thus evaluate appearance transfer quality with a GPT-based system objectively ranking outputs, ensuring robust and human-like assessment, as further confirmed by our user study. Beyond showcased scenarios, our method is general and could be extended to different types of diffusion models and guidance functions. Project Page: <https://sayands.github.io/guideflow3d>

1 Introduction

Transferring appearance—including both texture and fine geometric detail—to a 3D object is a challenging and increasingly relevant problem across applications in gaming, augmented reality, and digital content creation. While style transfer has seen substantial progress in the 2D domain [20, 64, 65, 68], extending it to 3D introduces unique challenges due to the irregular, sparse, and diverse nature of 3D representations [26, 40, 22, 85]. Style cues may originate from 3D shapes, 2D images, or natural language, further increasing task complexity. An additional challenge lies in transferring the appearance across objects with substantial geometric differences, required in practical 3D design.

Several existing methods address 3D generation by reformulating it as a multi-view task, leveraging 2D diffusion models and conditioning on, e.g., rendered depth images to preserve input geometry [47, 36, 84]. However, these methods often produce geometrically inconsistent results due to discrepancies across different views. Recent advances in 3D generative modeling—particularly denoising-based approaches—enable high-quality synthesis of shapes and appearances conditioned on inputs like text [54, 5, 72]. Still, these models are constrained by conditioning signals and data distributions used during training. As a result, direct application to appearance transfer yields poor generalization

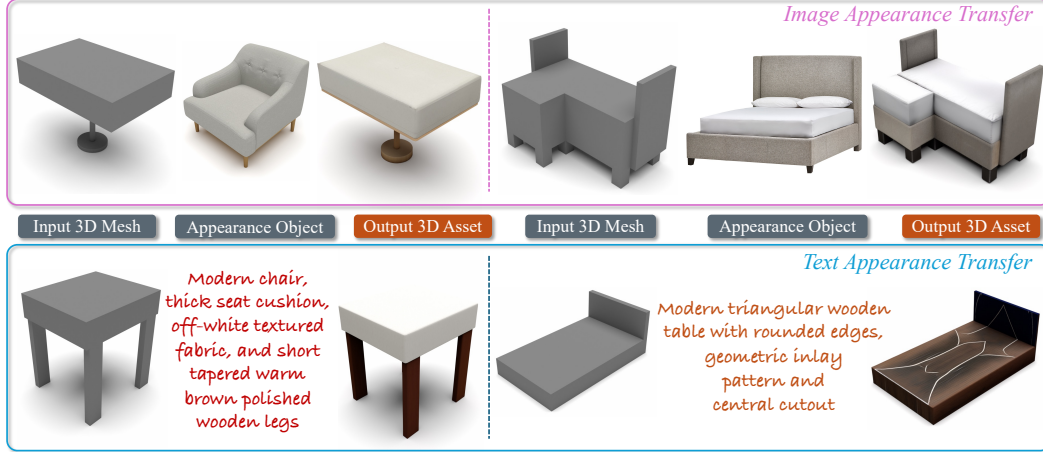


Figure 1: **GuideFlow3D is a method for 3D appearance transfer robust to strong geometric variations between objects.** Given an input 3D mesh, e.g., designed using simple 3D primitives, it transfers the texture and fine geometric details of an appearance object (e.g., the rounded edges of the table on the top left and the base and mattress distinction of the bed on the top right) but preserves the geometric form of the input mesh. Its flexibility across appearance modalities like meshes or text makes GuideFlow3D efficient for generating diverse 3D assets.

and limited control, particularly when the geometries between the input and appearance objects differ significantly, as we show in our experiments. Note that, in the appearance transfer setting, the input object dictates the global geometry and the appearance one the texture and finer geometric details.

In this work, we present *GuideFlow3D*, a training-free framework for 3D appearance transfer that adaptively steers a pretrained generative model at inference time. Our key insight is that the inductive bias of a pretrained generative model can be repurposed for a new task through *guided rectified flow sampling*, an inference-time mechanism that interleaves flow updates with latent-space optimization. This strategy extends the concept of universal guidance to 3D generation and enables conditioning on objectives for which the base model was not originally trained. Our method builds upon rectified flow models and structured latent representations [72], and introduces differentiable guidance functions that modulate the generation process without requiring retraining. To address prior work challenges in transferring appearance to an input 3D object while maintaining its global geometry, we propose two novel forms of guidance for appearance transfer to 3D shapes that significantly improve robustness to geometric variations between given input and appearance objects: (i) a part-aware appearance loss, which co-segments the input and appearance 3D shapes into semantically meaningful parts and enforces localized texture and geometry correspondence between the shapes; and (ii) a self-similarity loss, which preserves intrinsic structure within regions of the input during transfer. Our method supports various representations for the appearance object—including mesh-image pairs or text. When a mesh is available, the appearance loss is applied; otherwise, the self-similarity loss guides the generation using either an image or text. In this way, users can control whether appearance affects both geometry and texture (with mesh) or texture alone (with image or text). When only an image is provided, a mesh can be generated via existing methods such as [72]. This flexibility allows GuideFlow3D to unify multiple appearance modalities under a single framework, bridging geometric and perceptual style transfer in 3D.

Quantitatively evaluating 3D appearance transfer remains challenging due to the absence of ground truth input shapes that have the transferred appearance and the difficulty of comparing across dissimilar geometries. While metrics like DINOv2 [48], CLIP [29], and DreamSim [25] are typically used to assess perceptual similarity and others like PSNR [31], SSIM [69], LPIPS [79], and FID [15], for reconstruction quality, they require ground truth data that does not exist in our setting. To address this, we use a GPT-based evaluation scheme [15, 71] that performs pairwise output ranking in a human-aligned manner. We further validate the consistency of these rankings through a user study, confirming that GPT-based judgments strongly correlate with human evaluation on this task. Our experiments demonstrate that GuideFlow3D consistently outperforms baselines, generating visually appealing results that accurately reflect the intended style, while also capturing fine-grained geometry. Our contributions are as follows:

- We introduce *GuideFlow3D*, a novel framework for 3D appearance transfer that applies universal, differentiable guidance to a pretrained rectified flow model, enforcing the generation process to respect given constraints for which it was not originally trained.
- We propose part-aware and self-similarity loss functions as effective forms of guidance, enabling localized and structure-preserving style transfer across diverse 3D assets.
- Our method is training-free, generalizable to different appearance representations and, it could be, in principle, extended to a variety of 3D generative models.

By decoupling style control from the generation process and enabling inference-time conditioning on novel objectives, GuideFlow3D opens new directions in controllable 3D generation and asset stylization. We make our code and benchmark publicly available on our [project website](#).

2 Related Work

3D Generative Models. A common method for learning 3D shape generation uses autoencoders, where an encoder maps a point cloud to an embedding, which a decoder reconstructs into a 3D shape. Previous approaches rely on normalized point flows [74] or gradient fields [9]. GAN formulations are common in 3D shape generation [70, 50, 10, 27]. Luo et al. [43] and more recent approaches [13, 83, 11, 46] formulate 3D asset generation as a probabilistic diffusion processes. Some works take advantage of 2D generative models to create 3D assets [54, 63, 37]. Others [12, 67, 61] propose auto-regressive models for generating 3D meshes. As we discuss further in Sec. 3, Trellis [72] uses rectified flow formulation to learn the generation of structured 3D latents that can effectively be decoded in different 3D representations. Due to its ability to capture fine details in both geometry and texture, we develop our appearance transfer framework around this model.

2D Style Transfer. Transferring style between images has been extensively studied in computer vision. Early work [28] formulates it as an optimization problem over pretrained CNN features, later replaced by efficient feed-forward networks [32, 66]. With multimodal embeddings, StyleCLIP [49] enables text-driven style manipulation. Diffusion-based approaches [81, 68, 73, 55] now dominate the field, offering fine-grained control through score-based modeling. Cross-Image Attention [2] performs zero-shot appearance transfer by injecting cross-image attention into the diffusion process, while MambaST [8] leverages state-space models for efficient and expressive structure–style fusion. [64] trains a generator on a specific content–style example, where structure and appearance constraints are derived from pretrained vision transformer features. Such instance-specific formulations highlight the challenge of achieving generalizable appearance transfer across diverse geometries – a limitation our approach directly addresses by introducing a guided generative mechanism that preserves geometric consistency while adapting fine-grained style cues.

3D Style Transfer. Style transfer has been explored across diverse 3D representations, including point clouds [75], meshes [7], implicit fields [30, 14, 24, 39], and Gaussian splatting [40]. 3DStyleNet [75] separates geometry and texture through dual networks, while Mesh2Tex [7] maps style images into a learned texture manifold. Recent methods such as StyleRF [39] and StyleGaussian [40] represent major progress but operate on implicit or point-based representations, producing render-only outputs that lack explicit mesh control and often require multi-view optimization. StyleGaussian [40] stylizes only color while keeping geometry fixed, with performance degrading as Gaussian count increases. In contrast, GuideFlow3D performs training-free stylization in a structured latent space, enabling direct mesh editing, part-aware control, and constant inference cost. In diffusion-based 3D stylization, ControlNet-style adapters [77] have become standard for injecting additional cues such as images, edges, depth, or pose into pretrained models. For example, TEXTure [56] employs score distillation for text-guided texturing while EASI-Tex [52] extends this with depth-aware conditional diffusion. TriTex [15] learns a semantic texture field for single-mesh appearance transfer. While effective, these methods are tied to specific training setups and conditioning modalities, limiting their generality. Building on the broader line of diffusion guidance research [21, 6, 76], GuideFlow3D combines the concept of universal guidance and rectified flow model from Trellis [72]. By incorporating geometric priors from PartField [41], our method achieves robust and geometry-aware appearance transfer without the need for task-specific finetuning.

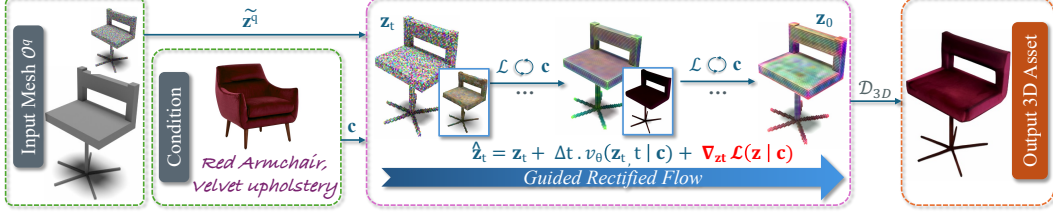


Figure 2: **GuideFlow3D introduces guided rectified flow for appearance transfer between input object $\mathcal{O}^q$ and an appearance object.** We extend the denoising process of structured latents $\tilde{z}^q$, conditioned by $\mathbf{c}$, by introducing an objective function $\mathcal{L}$ that enforces strong geometric and semantic priors during the process. We show denoised structured latents at different stages of the process, along with corresponding meshes decoded using a pretrained decoder $\mathcal{D}_{3D}$. The output 3D asset displays robustness to strong geometric variations between input and appearance objects.

3 Preliminaries

Structured Latent From 3D Assets. The geometry and appearance of a 3D object mesh $\mathcal{O}$ can be encoded using a structured latent (SLAT) representation $\mathbf{z}$ [72], composed of local latent codes anchored on a 3D voxel grid. This representation is defined as:

$$\mathbf{z} = \{(z_i, p_i)\}_{i=1}^L, \quad z_i \in \mathbb{R}^C, \quad p_i \in \{0, 1, \dots, N-1\}^3, \quad (1)$$

where p_i denotes the position of an active voxel intersecting the surface of $\mathcal{O}$, and z_i is the latent vector associated with that voxel. N is the spatial resolution of the grid and L is the number of active voxels, intersecting with the object’s surface. Since the number of active voxels is significantly lesser than that of a full grid, this representation is computationally less demanding, allowing to work at a higher resolution. The set of active voxel positions p_i outlines the coarse structure of the object, while the corresponding latents z_i capture fine-grained geometric and visual features, as shown in [72]. This representation thus effectively captures both the global shape and detailed surface characteristics of the object. In practice, following Trellis [72], multi-view images of $\mathcal{O}$ are rendered, and DinoV2 [48, 18] features are extracted for each image. After back-projection and aggregation of these features per voxel, a shallow transformer encoder [57, 35] compresses them into SLAT. Structured latents can be decoded into 3D Gaussians [33], Radiance Fields [45], or Meshes [60] using decoders $\mathcal{D}_{3D}$ that share a common architecture, differing only in output layers and trained with representation-specific losses. Please note that, in our case, voxel positions p_i remain fixed to preserve the coarse input geometry, while only the latent codes z_i are steered during generation.

Universal Diffusion Guidance. We briefly introduce the concept of universal guidance from [6]. Diffusion models are generative models that reverse the process of adding noise, typically Gaussian noise, to the data over time steps. A noisy version of the input x_0 at time step t is defined as:

$$x_t = \sqrt{\beta_t} x_0 + \sqrt{1 - \beta_t} \mathcal{N}(0, \mathbf{I}), \quad (2)$$

where β_t represents the noise scale at time step t , determined by a scheduling mechanism. Hence, x_t is a weighted sum of the original image and Gaussian noise. As t increases, β_t decreases, and x_t approaches pure noise as $t \rightarrow T$. With some generalization, a trained diffusion network $f_\theta(x_t)$ reverses the process by predicting the added noise. The denoising process can then be defined as:

$$\hat{x}_{t-1} = \hat{x}_t + f_\theta(\hat{x}_t) \quad (3)$$

While diffusion models can be typically guided by adding a condition to the diffusion model $f_\theta(x_t, \mathbf{c})$, e.g., $\mathbf{c}$ can be text or an image, this limits applications to more general conditions. In universal guidance from [6], any differentiable guidance function $\mathcal{L}(x_t, \cdot)$ can be used to condition diffusion. Then, the update in the reverse step is defined as:

$$\hat{x}_{t-1} = \hat{x}_t + f_\theta(\hat{x}_t, \mathbf{c}) + \nabla_{\hat{x}_t} \mathcal{L}(\hat{x}_t, \cdot) \quad (4)$$

Such an approach enables general conditioning without altering the diffusion training process.

4 GuideFlow3D

We present an overview of our approach for appearance transfer in Fig. 2. Given an input 3D object mesh $\mathcal{O}^q$ and an appearance object $\mathcal{O}^a$, we would like to modify the appearance of $\mathcal{O}^q$ based on $\mathcal{O}^a$,

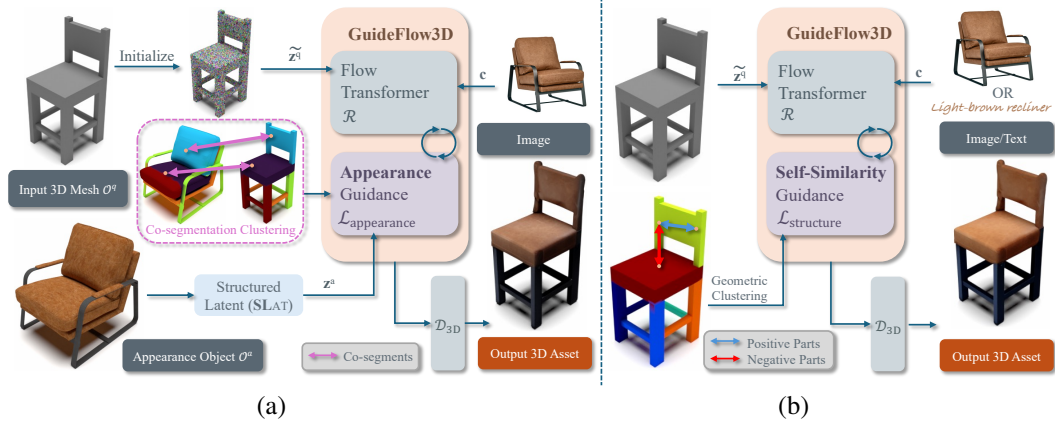


Figure 3: **GuideFlow3D with different guidance objectives.** (a) When a textured mesh is available for appearance object $\mathcal{O}^a$, we use our co-segmentation based objective $\mathcal{L}_{appearance}$ to guide appearance transfer. It encourages consistency between structured latents z^a and noisy latents $\tilde{z}^q$. In such case, we use an image of object $\mathcal{O}^a$ to condition the generative model $\mathcal{R}$. (b) When textured mesh is not available, we use our geometric clustering based objective $\mathcal{L}_{structure}$ for guidance. It encourages intra-cluster similarity and inter-cluster disparity when denoising $\tilde{z}^q$. We use text or image to condition $\mathcal{R}$ in such case. For both cases, we use decoder $\mathcal{D}_{3D}$ to obtain the output 3D asset.

while respecting the geometric structure of $\mathcal{O}^q$. $\mathcal{O}^a$ can be represented as an image-mesh pair or text. We assume here that the appearance object provides the mesh alongside the image. In practice, the mesh can be generated by running [72] on the image. We first discuss how we use structured latents from [72] for our appearance transfer problem. Then, we show how we can guide the rectified flow sampling using our objectives to transfer appearance while accounting for structural consistency.

4.1 Structured Latents for Appearance Transfer

We leverage the latent structure from [72] to represent object $\mathcal{O}^q$, but in general, different representations could be considered. We initialize structured latents $\tilde{\mathbf{z}}^q = \{(\tilde{z}_i^q, p_i^q)\}_{i=1}^{L_q}$ for the input object $\mathcal{O}^q$, where p_i^q are the positional indices of the active voxels, as in Eq. (1). The $\tilde{z}_i^q$'s are initialized by sampling from a normal distribution. In practice, p_i^q is not a trainable parameter but the $\tilde{z}_i^q$'s are learnable in our appearance transfer formulation. This effectively enforces global geometric consistency of the output with $\mathcal{O}^q$. Following, we define two appearance transfer objectives (Fig. 3).

Appearance-based objective. When the appearance object is an image-mesh pair, we can extract latents $\mathbf{z}^a$ for the mesh of $\mathcal{O}^a$ using a sparse VAE encoder [72]. Ideally, we seek an ‘oracle’ mapping that aligns each input latent $\tilde{z}_i^q$ with a semantically and geometrically corresponding appearance latent in $\mathbf{z}^a$: e.g., in Fig. 3 (a), we would like the $\tilde{z}_i^q$ for voxels of the back leg of an input chair to be mapped to the z_j^a for the voxels of the appearance chair. Since such correspondences are not readily available, we approximate them by assigning each $\tilde{z}_i^q$ to its nearest neighbor in $\mathbf{z}^a$ based on feature similarity. Formally, our objective for appearance transfer is defined as:

$$\mathcal{L}_{appearance} = \frac{1}{L_q} \sum_{i=1}^{L_q} \|\tilde{z}_i^q - z_m^a\|_2^2, \quad (5)$$

where m is the index of the corresponding feature in structured latents $\mathbf{z}^a$ for $\tilde{z}_i^q$. While it could be possible to extract correspondences based on these latents, they are not trained to be part-aware, and this would lead to false assignments, as shown in Sec. C. Instead, we establish correspondences based on geometric co-segmentation clustering using part-based feature fields from [41]. In practice, we compute PartField [41] features per voxel and run k -means clustering, thus, relying on approximate matching rather than one-to-one correspondences.

Self-similarity objective. In the second scenario, we assume that the mesh is not available, hence $\mathcal{O}^a$ is represented as an image or text. We observe that structure-based self-similarity descriptors have been shown to effectively capture structural information while being invariant to appearance [59, 4,

34]. We, thus, rely on a loss that encourages similarity between same object parts while promoting inter-part separability in the feature space. As the appearance mesh is not available, this loss aligns structural features with the part-aware semantics implied by the textual description or image for the source. To do so, we first perform geometric clustering that assigns each voxel p_i^q to a cluster $\mathcal{C}_q(i)$. Given $\mathcal{O}^q$, for voxels p_i and $p_j \in \mathbf{z}_q$, we compute pairwise cosine similarity between geometric features sim_{ij} and define a part-aware contrastive loss based on self-similarity. Specifically, for each voxel p_i , the set of positive samples is defined as all $j \neq i$ such that $\mathcal{C}_q(i) = \mathcal{C}_q(j)$, and all other voxels serve as negatives. The objective can then be defined by:

$$\mathcal{L}_{\text{structure}} = -\frac{1}{L_q} \sum_{i=1}^{L_q} \log \frac{\sum_{j \in \mathcal{C}_q(i), j \neq i} \exp(\text{sim}_{ij})}{\sum_{j \in \mathcal{C}'_q(i)} \exp(\text{sim}_{ij})} \quad (6)$$

where L_q denotes the set of all voxels and $\mathcal{C}'$ is the complement set of $\mathcal{C}$. Due to its inherent contrastive nature, $\mathcal{L}_{\text{structure}}$ promotes local consistency without homogenizing appearance globally.

4.2 Guiding Structured Latents for Appearance Transfer

Our objective is defined using a Bayesian formulation. Given the input 3D object mesh $\mathcal{O}^q$ and appearance object $\mathcal{O}^a$, we would like to maximize the posterior $P(\mathcal{O}^q | \mathcal{O}^a)$:

$$\log P(\mathcal{O}^q | \mathcal{O}^a) = \log P(\mathcal{O}^q) + \log P(\mathcal{O}^a | \mathcal{O}^q), \quad (7)$$

where prior $\log P(\mathcal{O}^q)$ models the geometric prior of $\mathcal{O}^q$ while likelihood $\log P(\mathcal{O}^a | \mathcal{O}^q)$ models the appearance. The objectives defined in the previous section are not enough to perform appearance transfer. While $\mathcal{L}_{\text{appearance}}$ models the posterior $\log P(\mathcal{O}^q | \mathcal{O}^a)$, solely optimizing it proves insufficient. It tends to shift the latent distribution away from the one modeled by the generative network, often leading to implausible or incoherent results across diverse shapes (Fig. 4 (a)). Meanwhile, $\mathcal{L}_{\text{structure}}$ only models $P(\mathcal{O}^q)$ of our Bayesian formulation. To address these issues and better align with the generative prior, we introduce appearance transfer as guidance in rectified flow.

Rectified Flow Guidance. Rectified flow models [42, 3, 38] define a linear forward process, $\mathbf{z}(t) = (1 - t)\mathbf{z}_0 + t\epsilon$, which interpolates between data samples $\mathbf{z}_0$ and noise ϵ over time t . The corresponding backward process is modeled as a time-dependent vector field $\mathbf{v}(\mathbf{z}, t) = \nabla_t \mathbf{z}$, which transports noisy samples back toward the data distribution. This vector field can be approximated by a neural network $\mathbf{v}_\theta$, trained via the Conditional Flow Matching (CFM) objective [38]:

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t, \mathbf{z}_0, \epsilon} \|\mathbf{v}_\theta(\mathbf{z}, t) - (\epsilon - \mathbf{z}_0)\|_2^2. \quad (8)$$

While rectified flow enables sample generation that remains close to the learned data distribution, using it alone is insufficient for semantic control, particularly when transferring fine-grained appearance or structure. As shown in Fig. 4 (b), applying rectified flow without additional constraints often yields outputs that are within the learned distribution but fail to reflect the style texture or semantic structure, especially across diverse object categories and shapes. To enforce semantic control while preserving realism, we interleave latent-space optimization with sampling steps from a rectified flow model $\mathbf{v}_\theta(\mathbf{z}, t | \mathbf{c})$, conditioned on a global signal $\mathbf{c}$. Based on [72], the signal $\mathbf{c}$ can either be an image or text. Starting from an initial latent $\mathbf{z}_T \sim \mathcal{N}(0, I)$, reverse-time flow is defined as:

$$\hat{\mathbf{z}}_t = \mathbf{z}_t + \Delta t \cdot \mathbf{v}_\theta(\mathbf{z}_t, t | \mathbf{c}) \quad (9)$$

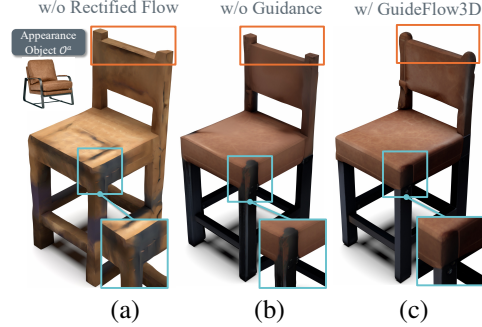


Figure 4: **Effect of different modules.** (a) Using only optimization with our objective functions to transfer appearance is insufficient as it does not enforce realistic distribution over the structured latent space. (b) The rectified flow model from [72] fails to transfer appearance when appearance and input objects have significantly different geometries. (c) We obtain appealing 3D assets when using rectified flow guidance of our GuideFlow3D.

where $t \in [0, 1]$ is linearly spaced. We adapt the equation to include guidance:

$$\hat{\mathbf{z}}_t = \mathbf{z}_t + \Delta t \cdot \mathbf{v}_\theta(\mathbf{z}_t, t \mid \mathbf{c}) + \nabla_{\mathbf{z}_t} \mathcal{L}(\mathbf{z} \mid \mathbf{c}) \quad (10)$$

where $\mathcal{L}$ is a general guidance objective. The optimized latent $\hat{\mathbf{z}}$ is then used for the next flow step. Such optimization-flow scheme allows for flexible conditioning via differentiable objectives without retraining the model, extending universal guidance [6] to any rectified flow model $\mathcal{R}$. The final output can be decoded into the 3D representation of choice based on 3D decoders from [72]. The guidance can be fit into our Bayesian formulation, where conditioned rectified flow models both prior $P(\mathcal{O})$ and likelihood $P(\mathbf{c}|\mathcal{O})$, and the guidance terms become additional factors in the prior and likelihood terms. We find that this interleaved scheme preserves global realism while enabling fine-grained control over both appearance and structure, as shown in Fig. 4 (c). Qualitative “in-the-wild” transfers (Fig. 6) further illustrate robustness under large geometric discrepancies, e.g., animal $\rightarrow$ furniture.

Condition-Specific Guidance. Our method supports flexible conditioning depending on the form of the input. When a reference image is provided, we may employ either appearance- or structure-based guidance. For appearance transfer, we assume access to the appearance mesh $\mathcal{O}^a$ from which structured latents $\mathbf{z}^a$ are extracted, and we optimize $\mathcal{L}_{\text{appearance}}$ to transfer appearance. Alternatively, even in the absence of a mesh, the same reference image can be used to compute geometric features, enabling structure-aware transfer via $\mathcal{L}_{\text{structure}}$. When the condition is a text prompt, we rely exclusively on $\mathcal{L}_{\text{structure}}$, using semantic clustering of geometric features to induce part-aware correspondences. This setup enables our framework to seamlessly handle both visual and textual inputs while maintaining geometric fidelity and semantic consistency.

5 Experiments

Dataset. Since there are no publicly available datasets for our task of transferring appearance across different shapes, we create a benchmark for evaluation. First, for input mesh, we generate synthetic objects using procedural models from [62]. Second, for the appearance mesh and images, we leverage the ABO dataset [16], with the text captions provided by [72]. ABO contains $\sim 8\text{K}$ artist-designed 3D models from Amazon, featuring complex geometries and high-resolution materials across 63 categories, mainly focused on furniture and interior decor. We use 5 categories: bed, cabinet, chair, table, and sofa, randomly sampling 100 objects from each dataset. Here onwards, we refer to the dataset of procedurally generated objects as *simple* and the ABO subset as *complex*. Using *simple* and *complex* meshes for the different categories, we create 250 input-appearance object pairs for each of our 4 experimental setups: (i) *simple-complex intra-category*, (ii) *simple-complex inter-category*, (iii) *complex-complex intra-category*, and, (iv) *complex-complex inter-category*.

Evaluation Metrics. In absence of ground truth transferred texture for the input 3D mesh, encoder-based metrics are not representative of appearance transfer for 3D shapes of very different geometries (see more in Appendix Sec. F). LLMs have been previously shown to be a human aligned evaluator for generation tasks capturing structural preservation and content alignment [71, 80, 51]. We use GPT-5 to carefully develop a more nuanced human-aligned ranking system. This system evaluates results across six criteria—Style Fidelity, Structure Clarity, Style Integration, Detail Quality, Shape Adaptation, and Overall Quality—capturing both semantic intent and the perceptual quality of texture transfer in 3D. Following prior work [15, 71], we use multi-view renders of input, appearance, and output for all methods and prompt GPT-5 to give us a ranking (details in Appendix Sec. H). We show that GPT-5 is aligned with human preferences on this task via a user study in Appendix Sec. G.

Baselines. We compare our GuideFlow3D for appearance transfer against multiple baselines: **(1) UV Nearest Neighbor:** We find the nearest neighbors in Euclidean space for each point in the input mesh $\mathcal{O}^q$ to the mesh of the appearance object $\mathcal{O}^a$, and map the coordinates of the UV texture map accordingly; **(2) Image-to-3D:** We render the input mesh from multiple views, apply state-of-the-art 2D style transfer models [2, 8], and lift the stylized renderings to 3D via the image-conditioned Trellis model; **(3) EasiTex** [52]: is a conditional diffusion-based method that uses edge-aware conditioning and ControlNet [77] to texture an existing 3D mesh from a single RGB image, **(4) Trellis** [72]: serves as our baseline without guidance, enabling detailed local texture transfer using structured latents; and **(5) Text-to-3D:** For text-based appearance, we first generate a reference image using Stable Diffusion [53] and then apply the same Image-to-3D pipeline described above.

5.1 Appearance Transfer

Our first goal is to transfer appearance from an object given as image and textured mesh, or as text, to an untextured 3D object, within the *same* semantic category. This setting allows us to systematically assess how well each method preserves stylistic intent while adapting to varied geometries within each semantic category. Tab. 1 shows the results on the *simple-complex intra-category* set under both image-mesh ($\mathcal{L}_{\text{appearance}}$) and text conditioning ($\mathcal{L}_{\text{structure}}$). As illustrated in Fig. 5 (top row), transferring the appearance from a chair to another chair reveals clear differences in quality. MambaST [8] produces textures that are globally consistent due to its state-space backbone, ensuring smooth overall color and material coherence. However, residual gray tones from the input mesh persist, causing uneven blending and inconsistent local texture alignment. Cross Image Attention [2] effectively transfers local appearance patterns through image-conditioned attention, yet fails to maintain consistent mapping when applied to 3D surfaces, introducing artifacts during transfer. EASI-Tex [52] performs competitively with its ControlNet-based edge and depth conditioning, yet struggles under large geometric deviations due to limited generalization beyond its training setup. Trellis [72] improves upon this with its probabilistic structured latent generation but struggles to integrate appearance seamlessly across large geometric variations. Although it associates some textures reasonably (e.g., the chair seat transferred between the two), it often results in loss of fine-grained details and patchy surfaces. In contrast, GuideFlow3D achieves strong improvements in both appearance transfer and structural clarity preservation in image and text settings. The transferred appearance aligns well with object structure, as seen in the smooth material transitions and consistent fabric textures on the chair. This demonstrates the ability of our approach to effectively translate both visual and semantic cues into high-fidelity, structure-aware textures across diverse 3D objects (more results in Appendix Sec. I).

Table 1: Quantitative comparison based on our GPT-based ranking metrics that rank quality of appearance transfer based on different criteria. Results are on the *simple-complex intra-category* set.

Methods	Ranking metrics					
	Fidelity ↓	Clarity ↓	Integration ↓	Quality ↓	Adaptation ↓	Overall ↓
<i>w/ Image Condition ($\mathcal{L}_{\text{appearance}}$)</i>						
UV Nearest Neighbor	4.12	3.84	4.30	4.10	4.43	4.33
MambaST [8]	4.94	3.55	4.56	4.90	4.42	4.87
Cross Image Attention [2]	3.56	3.48	3.32	3.83	3.47	3.59
EasiTex [52]	3.18	4.30	4.08	3.17	4.18	3.81
Trellis [72]	<u>2.51</u>	<u>2.58</u>	<u>2.53</u>	<u>2.69</u>	<u>2.61</u>	<u>2.62</u>
GuideFlow3D (Ours)	1.89	2.41	2.07	2.23	2.28	2.12
<i>w/ Text Condition ($\mathcal{L}_{\text{structure}}$)</i>						
UV Nearest Neighbor	3.12	3.21	3.82	3.61	3.43	3.64
SDXL + Cross Image Attention	2.88	2.52	3.25	3.38	3.29	2.98
Trellis [72]	<u>2.01</u>	<u>1.89</u>	<u>2.67</u>	<u>2.75</u>	<u>2.55</u>	<u>2.39</u>
GuideFlow3D (Ours)	1.54	1.63	2.01	2.15	2.44	1.95

Furthermore, to understand cross-category generalization, we evaluate a more challenging setting where the structure and appearance inputs come from different object categories, e.g., transferring the appearance of a cabinet to a bed (Fig. 5, bottom row). Results under image conditioning for both *simple-complex* and *complex-complex* sets are in Tab. 2. We show results for text conditioning in the Appendix (Sec. A). This scenario tests not only fidelity to style but also the ability to adapt and maintain perceptual coherence under stronger geometric variations. GuideFlow3D maintains good performance across all metrics despite the added difficulty. In Fig. 5, compared to Cross Image Attention [2] and EASI-Tex [52], which degrade under large shape discrepancies, our method preserves both global coherence and fine structural fidelity. Trellis [72] not only fails to transfer the cabinet’s appearance effectively but also introduces a major geometry change by closing a hole on the side and texturing the altered region. Our robust performance highlights the ability to disentangle style from structure, enabling us to transfer appearance features even when spatial priors do not align. This demonstrates that our method is applicable to a variety of real-world applications.

5.2 Application: In-the-wild Appearance Transfer

To assess the generalization of GuideFlow3D beyond furniture-to-furniture, we explore appearance transfer in-the-wild using structurally diverse 3D assets from Objaverse-XL [19] and ABO [16],



Figure 5: **Qualitative Comparisons showing quality of appearance transfer.** Top and bottom rows show intra-class (chair to chair) and inter-class (cabinet to bunk bed) results respectively. In both examples, MambaST [8] blends the textures from both input and appearance objects, giving a grey hue to the final result. EasiTex [52] generates non-smooth repetitions of texture and fails to generate textures for the entire object (e.g., the handles of the chair or the bottom part of the bunk bed). Cross Image Attention [2] performs better but omits texture details (cabinet’s wood texture) and fails to preserve good local geometry. Trellis [72] preserves better the texture of the appearance object and does better on matching it to the input object, however, it fails at providing a uniform texture on the arms of the chair and does not preserve the overall geometry of the bunk bed by closing the side hole. Ours performs the best by adhering to the appearance object’s texture and matching it on the input object, while preserving the overall geometry of the input object.

Table 2: Quantitative comparisons ranking of our GuideFlow3D against baselines for different experimental settings. Results are shown with image conditioning ($\mathcal{L}_{\text{appearance}}$).

Methods	Ranking metrics					
	Fidelity ↓	Clarity ↓	Adaptation ↓	Fidelity ↓	Clarity ↓	Adaptation ↓
	Intra-Category			Inter-Category		
Simple-Complex						
UV Nearest Neighbor	4.12	3.84	4.43	4.06	3.51	4.17
MambaST [8]	4.94	3.55	4.42	4.87	3.57	4.38
Cross Image Attention [2]	3.56	3.48	3.47	3.54	3.55	3.52
EasiTex [52]	3.18	4.30	4.18	3.25	4.21	4.10
Trellis [72]	<u>2.51</u>	<u>2.58</u>	<u>2.61</u>	<u>2.64</u>	<u>2.85</u>	<u>2.76</u>
GuideFlow3D (Ours)	1.89	2.41	2.28	1.99	2.75	2.45
Complex-Complex						
UV Nearest Neighbor	3.31	3.11	3.41	3.54	2.99	3.49
Cross Image Attention	4.00	4.13	4.07	3.79	3.99	3.91
MambaST	4.63	3.88	3.42	4.54	3.33	3.92
EasiTex	3.29	4.21	4.23	3.19	4.26	4.21
Trellis [72]	<u>2.82</u>	<u>2.73</u>	<u>2.81</u>	<u>2.99</u>	<u>3.15</u>	<u>3.09</u>
GuideFlow3D (Ours)	2.21	2.31	2.34	2.24	2.69	2.56

where either dataset can provide the input or appearance object. This setting poses significant robustness challenges due to extreme variation in object categories, mesh quality, and geometric complexity. In this setting, appearance objects are represented as mesh-image pairs, allowing us to use the part-aware appearance loss $\mathcal{L}_{\text{appearance}}$, as guidance. As illustrated in Fig. 6, our approach successfully transfers texture and fine geometric details while preserving the original structure. It transfers the appearance between objects in various semantic categories, such as animals, vehicles, and furniture. Despite the lack of semantic alignment or category overlap, we are able to apply texture in a part-aware and structurally consistent manner, highlighting the strength of co-segmentation and geometric clustering-based guidance mechanism. The resulting outputs show high visual fidelity with textures that adhere naturally to the surface topology of the target mesh. Compared to Trellis [72], which often loses part-aware texture continuity under large semantic shifts, GuideFlow3D maintains structurally aligned and visually coherent material transfer across categories. Notably, this application does not require additional assumptions or adaptations, and generalizes to unseen shape categories and object styles, reinforcing the adaptability of GuideFlow3D. The ability to generate visually coherent, stylized outputs has relevance for 3D asset stylization in AR/VR and digital twin generation.

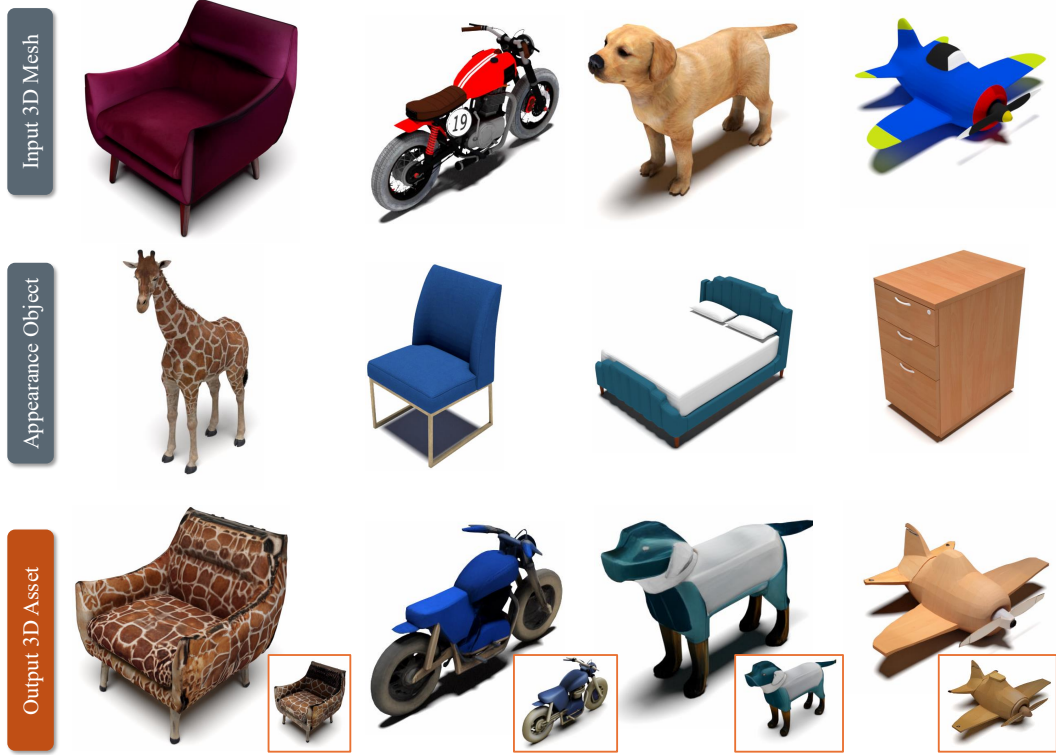


Figure 6: **Qualitative Comparisons showing in-the-wild appearance transfer.** Our GuideFlow3D robustly transfers appearance between diverse semantic categories. Patterns from the body of the giraffe are transferred to the seat, arms and backrest of the chair, while the appearance of giraffe legs is transferred to chair legs. Across other categories we observe interesting appearance transfers: chair legs to bike wheels, bed legs to golden retriever legs, and cabinet handles to airplane propeller. The smaller insets in orange boxes show Trellis [72] results, which exhibit weaker structural grounding and inconsistent material mapping, demonstrating GuideFlow3D’s advantage in preserving geometry-aware texture alignment even under large semantic shifts.

6 Conclusion

Our GuideFlow3D shows promising results in the field of 3D appearance transfer, possibly enabling exciting new applications. Given that our approach can generate realistic 3D assets from simplistic CAD designs, it could play a role in democratizing and simplifying 3D content creation. Tools for 3D content creation would become more accessible to artists which would significantly reflect on rapid development, prototyping, and creativity in XR and gaming platforms for example.

Limitations and Future Work. Our method is optimization based, thus it is not meant for real-time use cases. Our runtime is 96s on an NVIDIA 4090 GPU, compared to 78s for baseline [72]. In the future, we could train self-supervised feed-forward models for faster inference. Our implementation depends on the performance of [72] and [41], and failures of these models would impact our performance. Our approach assumes noiseless meshes which is a limiting factor for some future applications scenarios. Developing novel guidance objectives for new applications is an interesting future research direction to address such scenarios. The scope of our main experiments includes furniture objects from [16] and [62], but in practice, our method can be applied to a variety of object categories, as shown in Sec. 5.2, and downstream tasks such as scene editing (see Appendix Sec. D).

Ethical Considerations. Next to the exciting possibilities, there are considerable risks that should be addressed including manipulation and Deepfakes for spreading misinformation, concerns regarding intellectual property, and bias amplifications. Ethical usage of our method includes aspects of disclosing when 3D content is generated using AI, respecting and attributing source content licenses, and building systems for understanding biases are some of the ways for tackling these issues.

7 Acknowledgements

We thank Nicolas Dufour and Arijit Ghosh from Imagine Labs for helpful discussions on universal guidance, and Liyuan Zhu and Jianhao Zheng from Gradient Spaces Research Group for help with conducting the user study.

References

- [1] Stefan Ainetter, Sinisa Stekovic, Friedrich Fraundorfer, and Vincent Lepetit. Automatically annotating indoor images with cad models via rgb-d scans. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 3156–3164, 2023. 18
- [2] Yuval Alaluf, Daniel Garibi, Or Patashnik, Hadar Averbuch-Elor, and Daniel Cohen-Or. Cross-image attention for zero-shot appearance transfer, 2023. 3, 7, 8, 9, 16, 22
- [3] Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In ICLR, 2023. 6
- [4] Shir Amir, Yossi Gandelsman, Shai Bagon, and Tali Dekel. Deep vit features as dense visual descriptors. arXiv preprint arXiv:2112.05814, 2021. 5
- [5] Sudarshan Babu, Richard Liu, Avery Zhou, Michael Maire, Greg Shakhnarovich, and Rana Hanocka. Hyperfields: Towards zero-shot generation of nerfs from text. In Forty-first International Conference on Machine Learning. 1
- [6] Arpit Bansal, Hong-Min Chu, Avi Schwarzschild, Soumyadip Sengupta, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Universal guidance for diffusion models. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pages 843–852, 2023. 3, 4, 7
- [7] Alexey Bokhovkin, Shubham Tulsiani, and Angela Dai. Mesh2tex: Generating mesh textures from image queries. In International Conference on Computer Vision, 2023. 3
- [8] Filippo Botti, Alex Ergasti, Leonardo Rossi, Tomaso Fontanini, Claudio Ferrari, Massimo Bertozzi, and Andrea Prati. Mamba-st: State space model for efficient style transfer. In 2025 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pages 7797–7806, 2025. doi: 10.1109/WACV61041.2025.00757. 3, 7, 8, 9, 22
- [9] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snively, and Bharath Hariharan. Learning gradient fields for shape generation. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, pages 364–381. Springer, 2020. 3
- [10] Eric R Chan, Connor Z Lin, Matthew A Chan, Koki Nagano, Boxiao Pan, Shalini De Mello, Orazio Gallo, Leonidas J Guibas, Jonathan Tremblay, Sameh Khamis, et al. Efficient geometry-aware 3d generative adversarial networks. In Conference on Computer Vision and Pattern Recognition, 2022. 3
- [11] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In International Conference on Computer Vision, 2023. 3
- [12] Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. Meshanything: Artist-created mesh generation with autoregressive transformers. International Conference for Learning Representations, 2025. 3
- [13] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [14] Pei-Ze Chiang, Meng-Shiun Tsai, Hung-Yu Tseng, Wei-Sheng Lai, and Wei-Chen Chiu. Stylizing 3d scene via implicit representation and hypernetwork. In IEEE Winter Conference on Applications of Computer Vision, 2022. 3
- [15] Dana Cohen-Bar, Daniel Cohen-Or, Gal Chechik, and Yoni Kasten. Tritex: Learning texture from a single mesh via triplane semantic features. In CVPR, 2025. 2, 3, 7, 19
- [16] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, Matthieu Guillaumin, and Jitendra Malik. Abo: Dataset and benchmarks for real-world 3d object understanding. CVPR, 2022. 7, 8, 10, 19

- [17] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In Proc. Computer Vision and Pattern Recognition (CVPR), IEEE, 2017. 18, 19
- [18] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers, 2023. 4
- [19] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, Eli VanderBilt, Aniruddha Kembhavi, Carl Vondrick, Georgia Gkioxari, Kiana Ehsani, Ludwig Schmidt, and Ali Farhadi. Objaverse-xl: A universe of 10m+ 3d objects. arXiv preprint arXiv:2307.05663, 2023. 8
- [20] Yingying Deng, Fan Tang, Weiming Dong, Chongyang Ma, Xingjia Pan, Lei Wang, and Changsheng Xu. Stytr2: Image style transfer with transformers. In Conference on Computer Vision and Pattern Recognition, 2022. 1
- [21] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. Advances in Neural Information Processing Systems, 34, 2021. 3
- [22] Jiahua Dong and Yu-Xiong Wang. Vica-nerf: View-consistency-aware 3d editing of neural radiance fields. In NeurIPS, 2023. 1
- [23] Shivam Duggal, Yushi Hu, Oscar Michel, Aniruddha Kembhavi, William T. Freeman, Noah A. Smith, Ranjay Krishna, Antonio Torralba, Ali Farhadi, and Wei-Chiu Ma. Eval3d: Interpretable and fine-grained evaluation for 3d generation. arXiv preprint, 2024. 21
- [24] Zhiwen Fan, Yifan Jiang, Peihao Wang, Xinyu Gong, Dejia Xu, and Zhangyang Wang. Unified implicit neural stylization. In European Conference on Computer Vision, 2022. 3
- [25] Stephanie Fu, Netanel Tamir, Shobhita Sundaram, Lucy Chai, Richard Zhang, Tali Dekel, and Phillip Isola. Dreamsim: Learning new dimensions of human visual similarity using synthetic data. In Advances in Neural Information Processing Systems, volume 36, pages 50742–50768, 2023. 2, 20
- [26] Haruo Fujiwara, Yusuke Mukuta, and Tatsuya Harada. Style-nerf2nerf: 3d style transfer from style-aligned multi-view images. In SIGGRAPH Asia 2024 Conference Papers, 2024. 1
- [27] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. Advances in Neural Information Processing Systems, 2022. 3
- [28] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. arXiv preprint arXiv:1508.06576, 2015. 3
- [29] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: a reference-free evaluation metric for image captioning. In EMNLP, 2021. 2, 20
- [30] Hsin-Ping Huang, Hung-Yu Tseng, Saurabh Saini, Maneesh Singh, and Ming-Hsuan Yang. Learning to stylize novel views. In International Conference on Computer Vision, 2021. 3
- [31] Sadeep Jayasumana, Srikumar Ramalingam, Andreas Veit, Daniel Glasner, Ayan Chakrabarti, and Sanjiv Kumar. Rethinking fid: Towards a better evaluation metric for image generation. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9307–9315, 2023. 2
- [32] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016. 3
- [33] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics, 42(4), July 2023. 4
- [34] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019. 6
- [35] Yuhan Li, Yishun Dou, Xuanhong Chen, Bingbing Ni, Yilin Sun, Yutian Liu, and Fuzhen Wang. Generalized deep 3d shape prior via part-discretized diffusion process. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 16784–16794, 2023. 4
- [36] Zhiqi Li, Yiming Chen, Lingzhe Zhao, and Peidong Liu. Controllable text-to-3d generation via surface-aligned gaussian splatting, 2024. 1

- [37] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [38] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In ICLR, 2023. 6
- [39] Kunhao Liu, Fangneng Zhan, Yiwen Chen, Jiahui Zhang, Yingchen Yu, Abdulmotaleb El Saddik, Shijian Lu, and Eric P Xing. Stylerf: Zero-shot 3d style transfer of neural radiance fields. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [40] Kunhao Liu, Fangneng Zhan, Muyu Xu, Christian Theobalt, Ling Shao, and Shijian Lu. Stylegaussian: Instant 3d style transfer with gaussian splatting. In SIGGRAPH Asia 2024 Technical Communications. 2024. 1, 3
- [41] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. 2025. 3, 5, 10, 18, 19
- [42] Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In ICLR, 2023. 6
- [43] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In Conference on Computer Vision and Pattern Recognition, 2021. 3
- [44] Shalini Maiti, Lourdes Agapito, and Filippos Kokkinos. Gen3deval: Using vllms for automatic evaluation of generated 3d objects. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2025. 21
- [45] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM, 65(1):99–106, 2021. 4
- [46] Norman Müller, Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Peter Kotschieder, and Matthias Nießner. Diffrf: Rendering-guided 3d radiance field diffusion. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [47] Yeongtak Oh, Jooyoung Choi, Yongsung Kim, Minjun Park, Chaehun Shin, and Sungroh Yoon. Control-dreamer: Blending geometry and style in text-to-3d. arXiv:2312.01129, 2023. 1
- [48] 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. 2, 4, 20
- [49] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In International Conference on Computer Vision, 2021. 3
- [50] Dario Pavllo, Jonas Kohler, Thomas Hofmann, and Aurelien Lucchi. Learning generative models of textured 3d meshes from real-world images. In International Conference on Computer Vision, 2021. 3
- [51] Yang Peng, Yuxin Cui, Haomiao Tang, Zekun Qi, Runpei Dong, Jing Bai, Chunrui Han, Zheng Ge, Xiangyu Zhang, and Shu-Tao Xia. Dreambench++: A human-aligned benchmark for personalized image generation. In The Thirteenth International Conference on Learning Representations, 2025. 7, 21, 22
- [52] Sai Raj Kishore Perla, Yizhi Wang, Ali Mahdavi-Amiri, and Hao Zhang. Easi-tex: Edge-aware mesh texturing from single image. ACM Transactions on Graphics (Proceedings of SIGGRAPH), 43(4), 2024. 3, 7, 8, 9, 22
- [53] Dustin Podell, Zion English, Kyle Lacey, A. Blattmann, Tim Dockhorn, Jonas Muller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. ArXiv, abs/2307.01952, 2023. 7, 22
- [54] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. International Conference for Learning Representations, 2022. 1, 3
- [55] Tianhao Qi, Shancheng Fang, Yanze Wu, Hongtao Xie, Jiawei Liu, Lang Chen, Qian He, and Yongdong Zhang. Deadiff: An efficient stylization diffusion model with disentangled representations. In Conference on Computer Vision and Pattern Recognition, 2024. 3

- [56] Elad Richardson, Gal Metzer, Yuval Alaluf, Raja Giryes, and Daniel Cohen-Or. Texture: Text-guided texturing of 3d shapes. In ACM SIGGRAPH, 2023. 3
- [57] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models, 2021. 4
- [58] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In Computer Vision (ICCV), IEEE International Conference on, 2019. 19
- [59] Eli Shechtman and Michal Irani. Matching local self-similarities across images and videos. In CVPR, 2007. 5
- [60] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. ACM Trans. Graph., 42(4), jul 2023. ISSN 0730-0301. doi: 10.1145/3592430. 4
- [61] Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. Meshgpt: Generating triangle meshes with decoder-only transformers. In Conference on Computer Vision and Pattern Recognition, 2024. 3
- [62] Sinisa Stekovic, Arslan Artykov, Stefan Ainetter, Mattia D’Urso, and Friedrich Fraundorfer. Pytorchgeonodes: Enabling differentiable shape programs for 3d shape reconstruction. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025. 7, 10, 18
- [63] Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. International Conference for Learning Representations, 2022. 3
- [64] Narek Tumanyan, Omer Bar-Tal, Shai Bagon, and Tali Dekel. Splicing vit features for semantic appearance transfer. In CVPR, pages 10748–10757, 2022. 1, 3, 18
- [65] Narek Tumanyan, Omer Bar-Tal, Shir Amir, Shai Bagon, and Tali Dekel. Disentangling structure and appearance in vit feature space. ACM Trans. Graph., nov 2023. ISSN 0730-0301. 1
- [66] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. International Conference on Machine Learning, 2016. 3
- [67] Zhengyi Wang, Jonathan Lorraine, Yikai Wang, Hang Su, Jun Zhu, Sanja Fidler, and Xiaohui Zeng. Llama-mesh: Unifying 3d mesh generation with language models. arXiv preprint arXiv:2411.09595, 2024. 3
- [68] Zhizhong Wang, Lei Zhao, and Wei Xing. Stylediffusion: Controllable disentangled style transfer via diffusion models. In International Conference on Computer Vision, 2023. 1, 3
- [69] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004. doi: 10.1109/TIP.2003.819861. 2
- [70] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. Advances in Neural Information Processing Systems, 2016. 3
- [71] Tong Wu, Guandao Yang, Zhibing Li, Kai Zhang, Ziwei Liu, Leonidas Guibas, Dahua Lin, and Gordon Wetzstein. Gpt-4v(ision) is a human-aligned evaluator for text-to-3d generation. In CVPR, 2024. 2, 7, 21, 22
- [72] 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 CVPR, 2025. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 16, 18, 19, 20, 21, 22, 30
- [73] Binxin Yang, Shuyang Gu, Bo Zhang, Ting Zhang, Xuejin Chen, Xiaoyan Sun, Dong Chen, and Fang Wen. Paint by example: Exemplar-based image editing with diffusion models. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [74] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In International Conference on Computer Vision, 2019. 3
- [75] Kangxue Yin, Jun Gao, Maria Shugrina, Sameh Khamis, and Sanja Fidler. 3dstylenet: Creating 3d shapes with geometric and texture style variations. In International Conference on Computer Vision, 2021. 3

- [76] Jiwen Yu, Yinhuai Wang, Chen Zhao, Bernard Ghanem, and Jian Zhang. Freedom: Training-free energy-guided conditional diffusion model. In International Conference on Computer Vision, 2023. 3
- [77] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models, 2023. 3, 7
- [78] Muru Zhang, Ofir Press, William Merrill, Alisa Liu, and Noah A. Smith. How language model hallucinations can snowball. In Proceedings of the 41st International Conference on Machine Learning, ICML’24. JMLR.org, 2024. 22
- [79] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In CVPR, 2018. 2
- [80] Xinlu Zhang, Yujie Lu, Weizhi Wang, An Yan, Jun Yan, Lianke Qin, Heng Wang, Xifeng Yan, William Yang Wang, and Linda R. Petzold. Gpt-4v(ision) as a generalist evaluator for vision-language tasks. ArXiv, abs/2311.01361, 2023. 7, 21, 22
- [81] Yuxin Zhang, Nisha Huang, Fan Tang, Haibin Huang, Chongyang Ma, Weiming Dong, and Changsheng Xu. Inversion-based style transfer with diffusion models. In Conference on Computer Vision and Pattern Recognition, 2023. 3
- [82] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23, Red Hook, NY, USA, 2023. Curran Associates Inc. 22
- [83] Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. Locally attentional sdf diffusion for controllable 3d shape generation. ACM SIGGRAPH, 2023. 3
- [84] Linqi Zhou, Andy Shih, Chenlin Meng, and Stefano Ermon. Dreampropeller: Supercharge text-to-3d generation with parallel sampling. arXiv preprint arXiv:2311.17082, 2023. 1
- [85] Liyuan Zhu, Shengqu Cai, Shengyu Huang, Gordon Wetzstein, Naji Khosravan, and Iro Armeni. Scene-level appearance transfer with semantic correspondences. In ACM SIGGRAPH 2025 Conference Papers. Association for Computing Machinery, 2025. 1

Appendix

In the appendix, we provide the following:

- Intra- and inter-category results using text condition with $\mathcal{L}_{\text{structure}}$ (Sec. A)
- Results on using a rendered image as a condition with $\mathcal{L}_{\text{appearance}}$ (Sec. B)
- Ablation study on design choices (Sec. C)
- Application of GuideFlow3D on scene editing (Sec. D)
- Experimental and evaluation setup details (Sec. E)
- Issue with perceptual similarity evaluation metrics (Sec. F)
- Results of Human Evaluation (Sec. G)
- Details and analysis of the GPT-based evaluation setup (Sec. H)
- Additional qualitative results (Sec. I)

A Text Conditioning with $\mathcal{L}_{\text{structure}}$

Similar to image conditioned results in Tab. 2, we report results on using text prompts as the condition of the rectified flow model in Tab. A.1. As shown in Fig. 7, UV Nearest Neighbor fails to capture the described materials or object semantics, often ignoring features such as drawers, metal frames, or color attributes. SDXL + Cross-Image Attention [2] produces locally coherent patterns but lacks structural grounding, resulting in oversaturated or unrealistic textures. Trellis [72] improves the plausibility of the output yet misses fine-grained attributes—such as metallic elements, accent colors, or multi-material compositions and frequently distorts local part boundaries. In contrast, GuideFlow3D achieves consistently superior performance across all ranking metrics, both in *intra-category* and *inter-category*. In particular, our method preserves structural clarity and convincingly adapts texture, even when appearance is only described in abstract terms. The intra-category results reveal GuideFlow3D’s capacity to maintain semantic coherence between structure and style despite geometric variation. More strikingly, in the inter-category setting, where structure and appearance belong to different object types, our method still maintains better ranking scores (i.e., higher quality), demonstrating its robustness to both semantic drift and structural misalignment. This showcases the strength of our self-similarity guidance, which encourages intra-part consistency and inter-part contrast, enabling meaningful texture placement without explicit mesh supervision. Our flexible guidance based sampling process bridges the gap between structure and high-level semantic intent.

Table A.1: Quantitative comparison ranking of our GuideFlow3D against baselines for different experimental settings. Results are shown with text conditioning ($\mathcal{L}_{\text{structure}}$).

Methods	Ranking metrics					
	Fidelity ↓	Clarity ↓	Adaptation ↓	Fidelity ↓	Clarity ↓	Adaptation ↓
	Intra-Category			Inter-Category		
Simple-Complex						
UV Nearest Neighbor	3.12	3.21	3.43	3.45	3.66	3.72
SDXL + Cross-Image Attention	2.88	2.52	3.29	3.18	2.96	3.33
Trellis [72]	<u>2.01</u>	<u>1.89</u>	<u>2.55</u>	2.48	2.38	2.69
GuideFlow3D (Ours)	1.54	1.63	2.04	1.85	2.01	1.98
Complex-Complex						
UV Nearest Neighbor	3.45	3.52	3.61	3.58	3.67	3.75
SDXL + Cross-Image Attention	2.97	2.78	3.22	3.15	3.06	3.31
Trellis [72]	<u>2.16</u>	<u>2.05</u>	<u>2.49</u>	<u>2.35</u>	<u>2.41</u>	<u>2.66</u>
GuideFlow3D (Ours)	1.36	1.48	1.53	1.72	1.87	1.90

B Rendered Image Conditioning with $\mathcal{L}_{\text{appearance}}$

In many practical scenarios, an appearance mesh is provided without the corresponding reference image—either because the asset was procedurally generated, sourced from a CAD dataset, or previews were never created. To address this scenario, we render the appearance mesh from a fixed viewpoint to synthesize an image, which serves as the conditioning input for $\mathcal{L}_{\text{appearance}}$. For all objects we

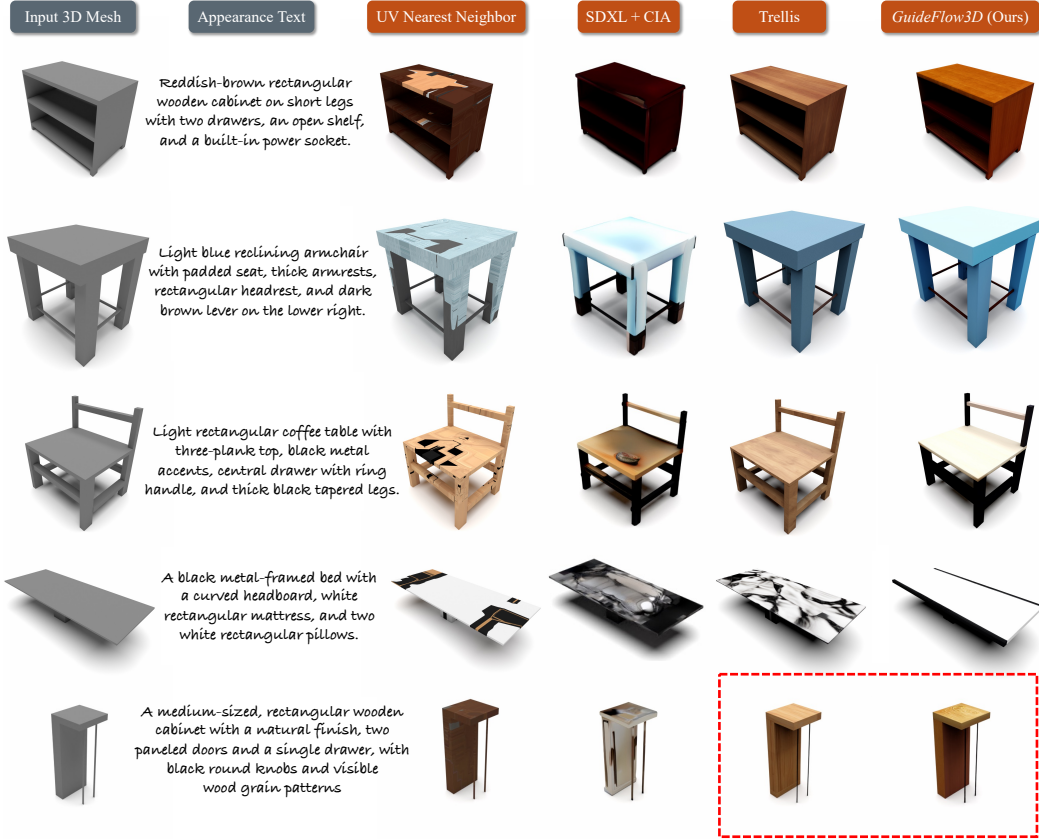


Figure 7: **Qualitative Comparisons showing results using text condition with $\mathcal{L}_{\text{structure}}$.** Each row shows an input 3D mesh, a descriptive appearance text, and textured results generated by different methods. We show challenging examples with considerable discrepancies between the appearance text prompts and input geometries. GuideFlow3D grounds textual descriptions in geometry, producing coherent, part-aware textures. The fifth row illustrates a failure case where abstract terms like “metal” or “cushioned” are not fully captured, highlighting the difficulty of interpreting underspecified text. Overall, GuideFlow3D delivers detailed, realistic appearances aligned with object semantics.



Figure 8: **Examples of rendered mesh views.**

use the same viewpoint, chosen to provide a slightly angled perspective on the object as shown in Fig. 8. Such view of the appearance mesh captures sufficient visual cues—such as texture, material, and shape context—to serve as effective guidance for our framework. This strategy enables mesh-based guidance in the absence of external visual data. Table B.1 shows results for this setup across intra- and inter-category pairs. Note that these ranking scores are not directly comparable to those in Tab. 2, as this experiment includes only the top performing baseline (Trellis); the absence of lower-ranked outputs can shift absolute values due to the relative nature of GPT-based evaluation.

While the reference image variant performs best overall, our rendered-view approach still consistently outperforms Trellis [72], demonstrating GuideFlow3D’s robustness against limited image data.

Table B.1: Quantitative comparison ranking of our GuideFlow3D against baselines for different experimental settings. Results are shown with image conditioning ($\mathcal{L}_{\text{appearance}}$).

Methods	Ranking metrics					
	Fidelity ↓	Clarity ↓	Adaptation ↓	Fidelity ↓	Clarity ↓	Adaptation ↓
	Intra-Category			Inter-Category		
Simple-Complex						
Trellis [72]	2.94	2.96	2.95	2.93	2.97	2.96
w/ Rendered Image	1.96	1.98	1.99	1.97	1.99	1.98
w/ Reference Image	1.08	1.03	1.02	1.06	1.02	1.01
Complex-Complex						
Trellis [72]	2.94	2.97	2.96	2.95	2.97	2.97
w/ Rendered Image	1.96	1.99	1.98	1.97	1.99	1.98
w/ Reference Image	1.01	1.02	1.02	1.07	1.02	1.01

C Ablation Study

We conduct an ablation study to analyze the impact of various design choices in our framework. Results in Tab. C.1 show performance on the *simple-complex intra-category* set under image conditioning. Inspired by the use of global appearance loss in [64], (i) we model global appearance for guidance. We do this using a combination of minimum, maximum, and average pooling operations which we found insufficient for our application. This highlights the ineffectiveness of global latents in capturing rich semantic correspondence required for high-quality appearance transfer. (ii) Replacing such global features with nearest-neighbor (NN) matching in SLAT space slightly improves performance, especially in fidelity, but lacks robustness in integration and adaptation, confirming that unstructured similarity fails to align content meaningfully across different shapes. (iii) Next, we evaluate a version using K-means-based co-segmentation over SLAT features instead of the PartField-driven segmentation [41] in GuideFlow3D. While guided flow improves generation realism, this variant still underperforms relative to the full method, indicating that semantically informed segmentation is crucial for establishing accurate part correspondences. (iv-v) Finally, we compare the two guidance objectives used in our framework: $\mathcal{L}_{\text{appearance}}$, and $\mathcal{L}_{\text{structure}}$, both using image as a condition. Each guidance improves transfer quality in its respective setting, with appearance guidance achieving stronger detail fidelity and structure guidance yielding better alignment and adaptability. These results validate our design of condition-specific guidance, where the choice of loss is tailored to the nature of the input, enabling flexible and semantically consistent transfer.

Table C.1: Ablation study ranking design choices using image conditioning. Results are on the *simple-complex intra-category* set.

Methods	Ranking metrics					
	Fidelity ↓	Clarity ↓	Integration ↓	Quality ↓	Adaptation ↓	Overall ↓
<i>w/o Rectified Flow</i>						
(i) Co-segmentation [41] & global feat.	4.52	4.51	4.53	4.51	4.49	4.50
(ii) Co-segmentation [41] & NN	3.58	3.62	3.61	3.60	3.61	3.63
<i>w/ Guidance in Rectified Flow</i>						
(iii) K-means Co-segmentation + NN	2.57	2.65	2.60	2.63	2.61	2.66
(iv) w/ Image Condition ($\mathcal{L}_{\text{structure}}$) - Ours	<u>2.17</u>	<u>2.05</u>	<u>2.12</u>	<u>2.05</u>	<u>2.11</u>	<u>2.03</u>
(v) w/ Image Condition ($\mathcal{L}_{\text{appearance}}$) - Ours	1.23	1.08	1.16	1.11	1.12	1.06

D Application: Scene Editing

To demonstrate the broader applicability of GuideFlow3D beyond isolated object transfer, we explore its use in full-scene editing. In this setting, we start with indoor scans from ScanNet [17] and utilize per-object CAD mesh annotations from PyTorchGeoNodes [62] and Scannotate [1] to obtain 3D mesh geometries of objects in the scene. For each semantic category present in the scene, we select

a representative appearance object from the ABO dataset [16] and perform appearance transfer individually using $\mathcal{L}_{\text{appearance}}$ on the corresponding meshes. As shown in Fig. 9, our method is capable of stylizing multiple objects across a cluttered scene while preserving their geometric structure. Note that objects with inaccurate pose annotations or high computational demands for Trellis processing are excluded from the transfer process. Despite relying on clean input meshes, our framework generalizes well in complex, real-world environments and opens avenues for future extensions with learned guidance in noisy or incomplete settings. This highlights GuideFlow3D’s utility for interactive scene customization, where designers can selectively restyle environments by swapping appearance while preserving structural layout.

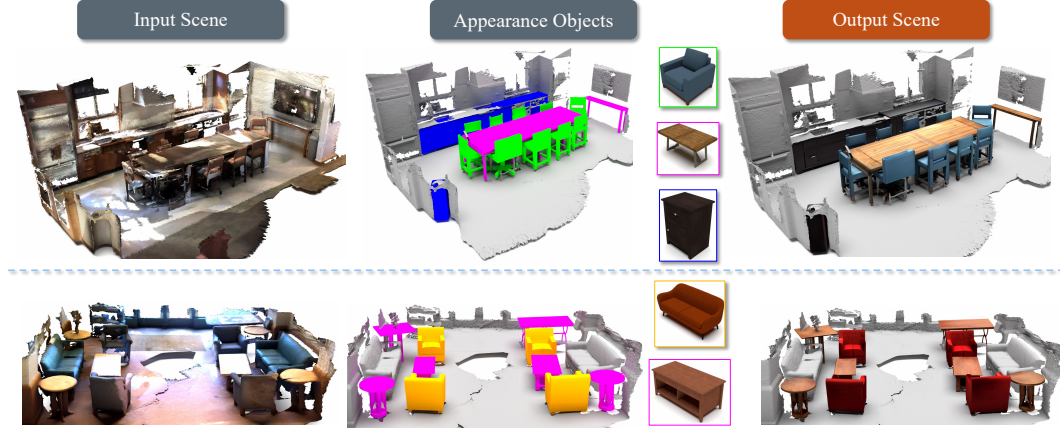


Figure 9: **Scene editing on scenes from ScanNet [17]**. For each semantic category, represented with a unique color in the middle column, we select a single appearance object and perform appearance transfer using our GuideFlow3D. Appearance is robustly transferred to different objects in the scene, showing another potential application of our approach.

E Experimental Details

Evaluation Setup. For evaluation, we render all assets—including the input, appearance, and output meshes—using Blender with smooth area lighting. Each object is rendered from 4 viewpoints, sampled around the origin at a fixed radius of 2 units and a pitch of 30° , with yaw angles spaced every 90° starting from 45° . All meshes are in a canonical pose to ensure input, appearance reference, and output are aligned, avoiding mismatches like comparing the front side of one object to the back side of another. For the rendered mesh-view guidance experiments (Sec. B), we use a pitch and yaw of 30° and 45° , respectively, to obtain a single consistent view of the appearance mesh. However, evaluation setup remains the same. Following standard practice [58, 15], we compute every metric, perceptual and GPT-based per view and per object, and report the average across all objects. This ensures that evaluations account for view-dependent variations.

Implementation Details. Our implementation is based on the pre-trained models and configurations from Trellis [72] and PartField [41]. We use `trellis-image-large` for image conditioning and `trellis-text-large` for text, and adhere to their default settings for any configuration. For part-aware guidance, we compute part feature fields using PartField [41] and query them using the voxel coordinates p_i of each mesh. We run GuideFlow3D for single-instance optimization interleaved with rectified flow sampling over 300 steps. Optimization is performed using AdamW with a learning rate of 5×10^{-4} . All experiments are conducted on a single NVIDIA RTX 4090 GPU. We use identical optimization settings across all conditioning types to ensure fairness and consistency.

F Issues with Perceptual Similarity Evaluation

Quantitatively evaluating appearance transfer in 3D is challenging due to the absence of ground truth outputs. Unlike traditional image-to-3D pipelines, where the generated asset can be directly compared to a reference view or textured model, our setting lacks a definitive target for what the



Figure 10: **Issue with Perceptual Similarity.** Even though our GuideFlow3D visually outperforms the Trellis baseline [72], CLIP scores do not reflect these improvements. In this example, we observe that the geometric forms of the input and appearance mesh are very different. Therefore CLIP score is not a suitable metric for evaluating similarity of output. We observe that the difference in scores increases for a cropped part of the chair, showing the difference between local and global geometry on the metric.

transferred appearance "should" look like. As illustrated in Fig. 10, the input and output geometries are intentionally dissimilar, and the appearance is adapted—not reconstructed—making traditional reconstruction-based metrics inapplicable. To this end, we report commonly used perceptual similarity scores such as DINOv2 [48], CLIP score [29], and DreamSim [25] in Table F.1, on image-conditioned appearance transfer for intra-category on *simple-complex* set. Due to large geometric differences between appearance and input geometries, these metrics offer only a coarse proxy for evaluating style transfer quality and often fail to capture the localized and semantic nuances essential to this task. Furthermore, in case of conditioning using text prompt, our CLIP score is actually slightly lower than the score for the Trellis baseline which is deceiving. This is because the text prompt often describes a very different geometry which is not aligned with the geometric form of the input object making such metrics less applicable for our experimental setting (e.g. "padded reclining armchair" vs. input mesh of a simple stool, as shown in second row of Fig. 7).

For instance, Fig. 10 shows an example where our method produces visibly better texture alignment and part-aware fidelity, yet perceptual similarity scores (e.g., CLIP) remain close or even favor inferior outputs. Comparisons on image crops, where scores between methods show larger gaps, highlight how such metrics can overlook structural misalignments, texture stretching, or poor integration when focusing on the global scale that are perceptually obvious to humans. While such encoder-based metrics could be useful in setups with ground-truth part-level correspondences that would enable an evaluation on cropped regions, their effectiveness is limited in our setting where no such reference exists. These limitations underscore the need for our fine-grained, human-aligned evaluation.

G Human Evaluation

We conducted a user study on Amazon Mechanical Turk with 59 participants to compare our approach against baselines. The evaluation set consisted of 100 randomly selected object and image pairs from

Table F.1: **Quantitative comparison based on perceptual similarity metrics.** Results are on the *simple-complex intra-category* set. Note that DinoV2 and DreamSim can only be evaluated in an image setting.

Methods	Perceptual similarity		
	DinoV2 $\uparrow$	CLIP $\uparrow$	DreamSim $\downarrow$
<i>w/ Image Condition ($\mathcal{L}_{appearance}$)</i>			
UV Nearest Neighbor	36.32	81.23	42.27
Cross Image Attention	44.67	84.60	41.67
MambaST	38.33	81.73	47.55
EasiTex	44.42	83.80	41.13
Trellis [72]	52.22	87.15	36.52
<i>GuideFlow3D (Ours)</i>	52.74	87.37	36.06
<i>w/ Text Condition ($\mathcal{L}_{structure}$)</i>			
UV Nearest Neighbor	-	25.32	-
SDXL + Cross-Image Attention	-	25.96	-
Trellis [72]	-	<u>27.64</u>	-
<i>GuideFlow3D (Ours)</i>	-	27.23	-

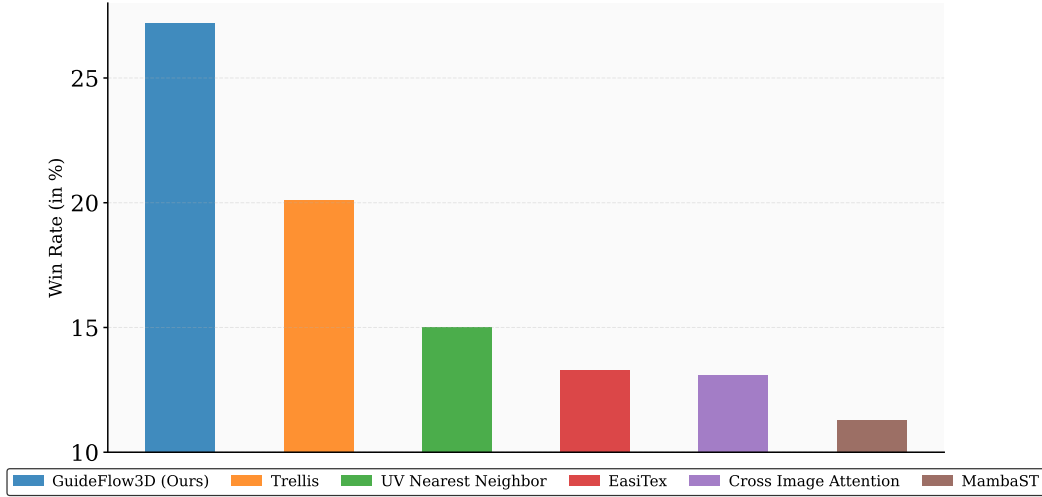


Figure 11: **Results of Human Evaluation** on randomly sampled outputs from both intra- and inter-category on *simple-complex* set.

the *simple-complex* intra- and inter-category, on the image condition setting. The participants were shown the appearance image and two views (front and back at a camera angle of 45 degrees) each of the structure mesh + outputs from different methods, without any knowledge about the methods. Then, they were provided with the same prompts as the LLM (Figs. 13 and 14) and asked to rate the best performing considering overall quality. As shown in Fig. 11, out of approximately 1000 evaluations, GuideFlow3D received the highest preference (27.2%), demonstrating its ability to balance structural preservation and appearance fidelity. Interestingly, UV Nearest Neighbor was ranked third, despite its lower quantitative performance in Tab. 1, likely due to its visually appealing but less semantically grounded results. Overall, the relative ranking across methods remains consistent with our LLM-based evaluation, confirming strong alignment between LLM-based and human judgments of texture fidelity and perceptual quality.

H GPT-based Evaluation

We outline the issues with automated perceptual similarity evaluation in Sec. F and resort to a GPT-based evaluation mechanism for analysis. GPT-4V has shown strong alignment with human judgments across vision-language tasks, including text-to-3D generation [51], image understanding, and 3D content evaluation. [71, 44, 23, 80]. In designing the GPT-based evaluator, we follow the human-aligned evaluation protocol proposed in GPT-Eval3D [71] and adopt it for appearance transfer using *gpt-5-mini*. We begin by clearly explaining the task to GPT evaluator: ranking textured 3D

outputs based on the quality of appearance transfer from a style source (image or text) to a structure mesh, using a single rendered view per mesh. All images are rendered from a consistent viewpoint, and the evaluator is instructed to imagine each output as a complete 3D object when judging texture consistency and structure preservation. As shown in Fig. 13 and Fig. 14, the instruction prompt defines six specific evaluation criteria: Style Fidelity, Structure Clarity, Style Integration, Detail Quality, Shape Adaptation, and Overall Quality. Each criterion is accompanied by a detailed explanation, emphasizing semantic consistency (e.g., appropriately mapping wood textures to structural elements like legs or seats, depending on the design), 3D geometric clarity, texture sharpness, and perceptual coherence. This structured rubric guides the LLM to perform fine-grained comparisons across outputs. To ensure consistency in output formatting and facilitate large-scale evaluation, we explicitly define a fixed output format corresponding to the six criteria. An example format is provided directly in the prompt to reduce ambiguity and improve parsing during automatic result aggregation. Additionally, we support in-context learning by optionally breaking the prompt into multiple stages with example completions shown before evaluation begins. These design choices—structured rubric, visual clarity, strict formatting, and optional in-context examples—ensure that our GPT-based evaluator produces reliable, human-aligned rankings across all experiments. Please note that we use output from all methods for comparison at a time (**three** is shown as an example in Figs. 13 and 14).

Limitations. While our GPT-based evaluation protocol proved practical and consistent for our purposes, prior work highlights several factors that future users should consider. Large multimodal models are known to exhibit hallucinations [71, 80] and systematic biases, such as position sensitivity [82, 78], and preference for high-contrast textures. These factors may distort rankings in ways that do not reflect actual quality. Moreover, GPT-based evaluation could be vulnerable to adversarial artifacts designed to exploit the multimodal model’s visual priors. Ensuring such metrics remain ‘ungamable’ is an open challenge. Future research can explore more efficient comparison strategies such as single method evaluation using a scale, e.g. 1 to 5 [51].

I Qualitative Results

We provide qualitative comparisons for both text- and image-conditioned appearance transfer. Figs. 7 and 12 highlight GuideFlow3D’s ability to produce semantically faithful and geometrically aligned textures across diverse scenarios. In the text-conditioned setting (Fig. 7), we evaluate challenging examples with significant discrepancies between textual descriptions and input geometries. UV Nearest Neighbor and SDXL [53] + Cross Image Attention [2] produce unrealistic outputs, often failing to capture semantic or color cues. Trellis [72] shows improvements in texture plausibility but struggles with global color consistency and local part correspondence, e.g., missing “reddish-brown” tones or “black metal frames.” In contrast, GuideFlow3D effectively grounds textual descriptions in geometry, yielding coherent, part-aware textures that accurately reflect material and structural cues. Nonetheless, when descriptions are abstract (e.g., “metal,” “mirror,” or “cushioned”), all methods show limitations, as shown in Fig. 7 (Row 5), highlighting the inherent ambiguity in interpreting such prompts. Under image conditioning (Fig. 12), UV Nearest Neighbor frequently misplaces textures due to limited geometric reasoning, though it performs reasonably when a single uniform texture dominates. MambaST [8], Cross Image Attention, and EasiTex [52] tend to oversmooth details or blend appearance cues across regions, while Trellis improves realism yet lacks explicit geometric grounding, sometimes introducing artifacts, such as dark textures from correspondence failures. GuideFlow3D achieves semantically and structurally consistent transfers, faithfully reproducing upholstery and wood patterns while maintaining sharp boundaries and material coherence. However, certain edge cases remain challenging (Fig. 12, Column 5), such as strong geometry-appearance mismatches or texture flattening under large shape variations. Overall, GuideFlow3D demonstrates robust semantic and structural fidelity across both modalities, effectively bridging the gap between appearance semantics and geometric form. These results underscore its capability to generate realistic, production-ready 3D assets.

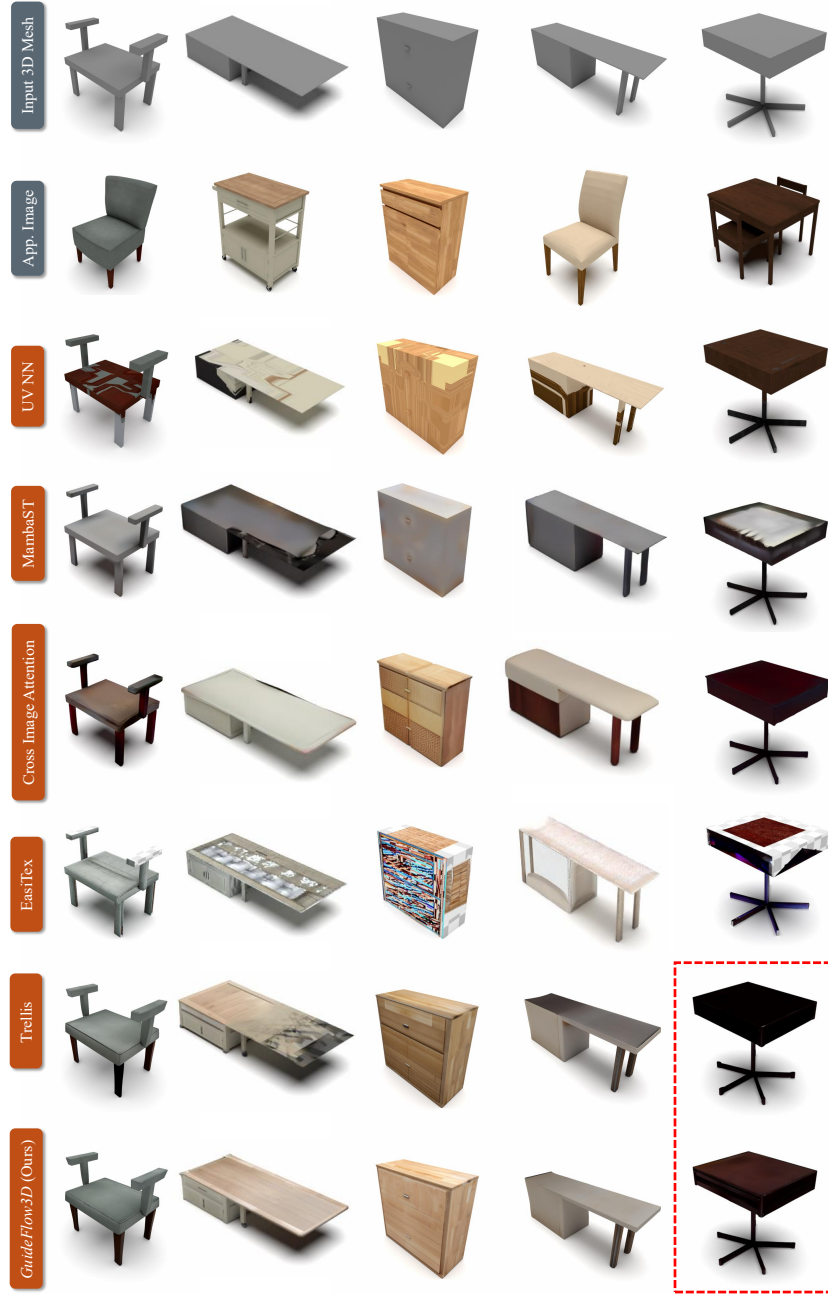


Figure 12: **Qualitative Comparisons showing results using image condition with $\mathcal{L}_{\text{appearance}}$.** We show 5 examples (one per column) and the results for all methods. UV Nearest Neighbor (UV NN) often misplaces textures across parts but performs better with uniform materials. MambaST, Cross Image Attention, and EasiTex blend or oversmooth textures, while Trellis improves realism yet lacks geometric grounding. GuideFlow3D achieves semantically and structurally consistent transfers, accurately mapping materials like upholstery and wood. Column 5 illustrates a failure case for all methods, though UV NN yields the most visually coherent result and our interleaved guidance scheme cannot recover from the black output produced by Trellis.

Our task is to rank **three** textured 3D outputs each created by transferring texture from a style mesh onto a structure mesh. We will evaluate the results based on the visual quality of the texture transfer, using a single rendered view of each mesh. All images are rendered from the same viewpoint. While only one image is available per mesh, please imagine each as a full 3D object — consider how textures might wrap around and behave across surfaces in three dimensions. The method should not only transfer color and texture from the style mesh, but also do so with semantic awareness — applying textures meaningfully to appropriate parts of the structure. Additionally, the geometry of the structure mesh must be preserved and visually clear. We would like to compare based on the following criteria:

Instruction

1. Style Fidelity: How well does the output capture the visual essence of the style mesh? Look for colour accuracy, material feel (e.g., matte, glossy, metallic), and stylistic patterns or motifs. A strong transfer should respect the semantic intent of the texture — e.g., wood textures mapped to handles, not faces. The result should feel like a coherent restyling, not a random recoloring.

2. Structure Clarity: Does the texture preserve the recognizable geometry of the structure mesh? Key parts (arms, legs, surfaces, joints) should remain distinguishable. Textures should enhance, not obscure. Imagine rotating the object: would the structure remain clear? Preservation of 3D form and part boundaries is critical.

3. Style Integration: How smoothly and appropriately is the style applied to the structure? Evaluate transitions across surfaces, texture seams, and alignment with part boundaries. Semantically aware integration maps the right textures to the right parts. Bad integration looks pasted or mismatched.

4. Detail Quality: Are local textures (grains, brushwork, ornamentation) clean, sharp, and artifact-free? Look for noise, blur, or visual inconsistencies. Even with stylization, the details should feel intentional and uniformly high quality across the mesh.

5. Shape Adaptation: Does the texture naturally follow the 3D geometry? Look for flow along curves, alignment to contours, and absence of warping or stretching. Imagine wrapping the style across a full 3D object. Well-adapted textures maintain realism and part continuity.

6. Overall Quality: Considering all of the above, which output delivers the best result overall? Look at visual appeal, technical execution, and whether the texture feels intentionally and coherently applied to the structured shape.

Output Format

For each criterion, rank outputs from best (1) to worst (3). Ties are allowed (e.g., 2 2 3). Summarize in this format:

Final answer: rankA / rankB / rankC / rankD / rankE / rankF

(Style Fidelity / Structure Clarity / Style Integration / Detail Quality / Shape Adaptation / Overall)

Example:

3 1 2 / 2 3 1 / 1 2 2 / 2 1 3 / 1 3 2 / 1 1 3

Figure 13: **Image GPT prompt.** Prompt provided to GPT-5-mini for providing a ranking of textured 3D outputs given an appearance object in the form of an image.

Our task is to rank **three** textured 3D outputs, based on a descriptive style caption onto a structure mesh. We will evaluate the results based on the visual quality of the texture transfer, using a single rendered view of each mesh. All images are rendered from the same viewpoint. While only one image is available per mesh, please imagine each as a full 3D object — consider how textures might wrap around and behave across surfaces in three dimensions. The method should translate the style described in the caption — including material, color palette, pattern, and visual feel — and apply it meaningfully and coherently to the structure. This includes respecting semantic cues (e.g., wood for handles, fabric for seats) and maintaining the clarity of the original geometry.

Instruction

We would like to compare based on the following criteria:

- 1. Style Fidelity:** How well does the output capture the visual essence of the style caption? Look for color accuracy, material feel (e.g., matte, glossy, metallic), and stylistic patterns or motifs. A strong transfer should respect the semantic intent of the texture — e.g., wood textures mapped to handles, not faces. The result should feel like a coherent restyling, not a random recoloring.
- 2. Structure Clarity:** Does the texture preserve the recognizable geometry of the structure mesh? Key parts (arms, legs, surfaces, joints) should remain distinguishable. Textures should enhance, not obscure. Imagine rotating the object: would the structure remain clear? Preservation of 3D form and part boundaries is critical.
- 3. Style Integration:** How smoothly and appropriately is the described style applied to the structure? Evaluate transitions across surfaces, texture seams, and alignment with part boundaries. Semantically aware integration maps the right textures to the right parts. Bad integration looks pasted or mismatched.
- 4. Detail Quality:** Are local textures (grains, brushwork, ornamentation) clean, sharp, and artifact-free? Look for noise, blur, or visual inconsistencies. Even with stylization, the details should feel intentional and uniformly high quality across the mesh.
- 5. Shape Adaptation:** Does the texture naturally follow the 3D geometry? Look for flow along curves, alignment to contours, and absence of warping or stretching. Imagine wrapping the style across a full 3D object. Well-adapted textures maintain realism and part continuity.
- 6. Overall Quality:** Considering all of the above, which output delivers the best result overall? Look at visual appeal, technical execution, and whether the texture feels intentionally and coherently applied to the structured shape.

Output Format

For each criterion, rank outputs from best (1) to worst (3). Ties are allowed (e.g., 2 2 3). Summarize in this format:
 Final answer: rankA / rankB / rankC / rankD / rankE / rankF
 (Style Fidelity / Structure Clarity / Style Integration / Detail Quality / Shape Adaptation / Overall)

Example:

3 1 2 / 2 3 1 / 1 2 2 / 2 1 3 / 1 3 2 / 1 1 3

Figure 14: **Text GPT prompt.** Prompt provided to GPT-5-mini for providing a ranking of textured 3D outputs given an appearance object in the form of a text.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#)

Justification: In Abstract and Introduction (Sec 1), we claim a novel method for appearance transfer between 3D objects based on guidance for rectified flow models. We claim that our method achieves very good results outperforming other baselines. We support the claims with our experimental analysis in Sec. 5.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: We discuss different limitation aspects of our method in Sec. 5.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: Our paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Next to open-sourcing our code and data necessary for the experiments as claimed in Sec 1, our manuscript provides information needed to reproduce the results. Additionally, we provide implementation details in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open source access to code and dataset, as claimed in in Sec 1, at this github repo: <https://github.com/GradientSpaces/GuideFlow3D>.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Sec. 5, we describe experimental settings while providing full details in the Appendix. We make all dataset information regarding splits and pairs publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We do not include error bars in our experiments. Instead, in Sec. 5, we evaluate our method on a range of settings, pointing to the statistical significance of our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide compute information when discussing limitations in Sec. 5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: We followed the NeurIPS Code of Ethics in this work. We took care to handle data responsibly, consider possible impacts, and follow all relevant ethical guidelines.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: We discuss positive and negative societal impacts in Sec. 6.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification: Our paper does not pose such risk.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We use assets from the ABO dataset and we cite the corresponding paper in Sec. 5. We generate assets with PytorchGeoNodes and provide the corresponding citation in Sec 5. We leverage the Trellis model [72] in our implementation and fully acknowledge this by citing and explaining its use (Sec 4). All use of the above has been within the licensing terms of the data and methods.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: We do not release new assets in the paper directly, but we will document our code release accordingly.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [Yes]

Justification: The paper consists of a user study conducted via Amazon Mechanical Turk (details in Appendix Sec. G).

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [Yes]

Justification: The study posed no foreseeable risks to participants, and the institution confirmed that formal IRB review was not required.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLM for our human-aligned evaluator in our quantitative experiments in Sec. 5.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.