ART-DECO: Arbitrary Text Guidance for 3D Detailizer Construction

QIMIN CHEN, Simon Fraser University, Canada
YUEZHI YANG, University of Texas at Austin, USA
YIFANG WANG, Adobe Research, USA
VLADIMIR KIM, Adobe Research, USA
SIDDHARTHA CHAUDHURI, Adobe Research, USA
HAO ZHANG, Simon Fraser University, Canada
ZHIQIN CHEN, Adobe Research, USA

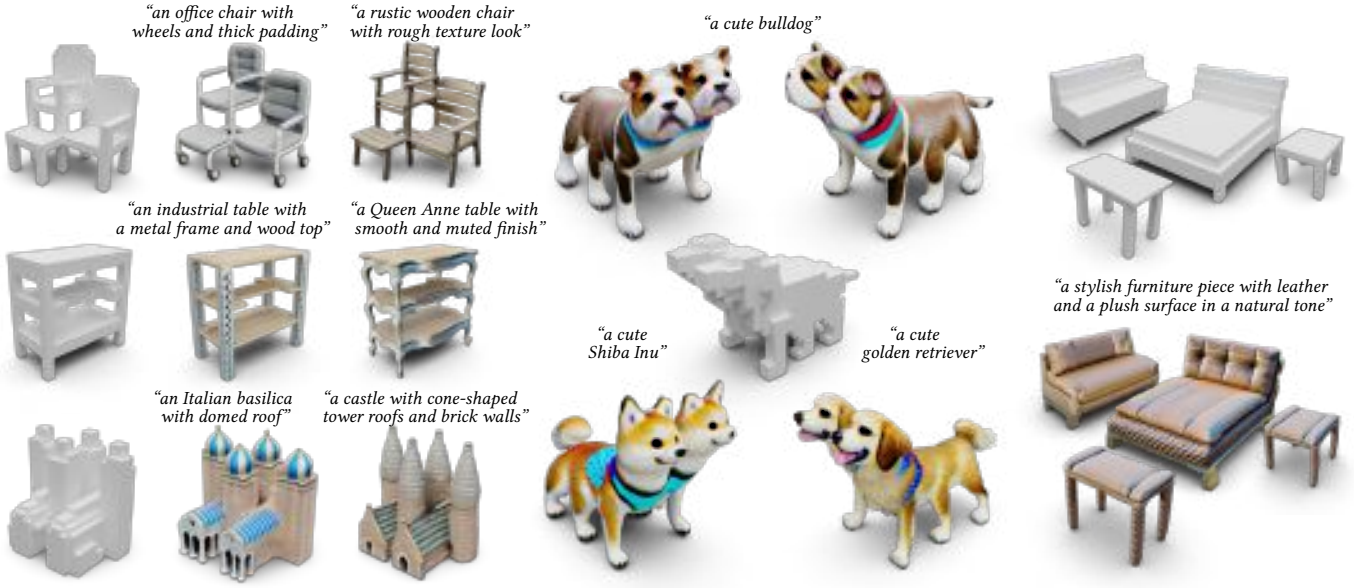


Fig. 1. Our 3D *detailizer* is trained using a text prompt, which defines the shape class and guides the stylization and detailization of any number of coarse 3D shapes with varied structures. Once trained, our detailizer can instantaneously (in <1s) transform a coarse proxy into a detailed 3D shape, whose overall structure respects the input proxy and the appearance and style of the generated details follows the prompt. We show results for both human-made (left) and organic (middle) shapes, with clearly out-of-distribution structures (e.g., the multi-seat "chair", letter-shaped shelvings, and two-headed dogs with six legs). On the right, when the training prompt references a generic term such as "furniture," which encompasses multiple object categories with diverse structures, such as chairs, beds, stools, etc., the 3D detailizer can be reused, as a feed-forward model without retraining, to produce a collection of detailized 3D models spanning all of these categories with structural variations. These models can then be arranged to form a style-consistent 3D scene.

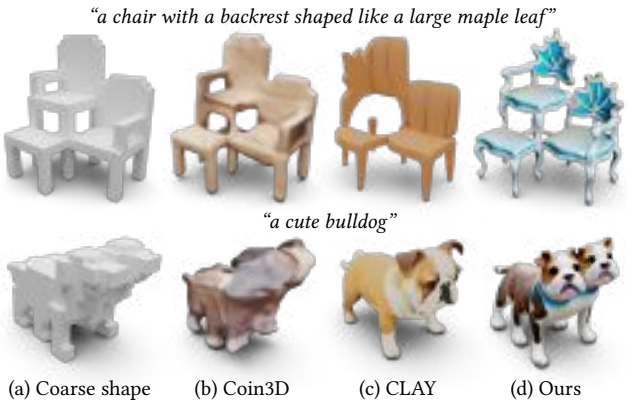
We introduce a *3D detailizer*, a neural model which can *instantaneously* (in <1s) transform a coarse 3D shape proxy into a high-quality asset with detailed geometry and texture as guided by an input text prompt. Our model is trained using the text prompt, which defines the shape class and characterizes the appearance and fine-grained style of the generated details. The coarse 3D proxy, which can be easily varied and adjusted (e.g., via user editing), provides structure control over the final shape. Importantly, our detailizer is not optimized for a single shape; it is the result of *distilling a generative model*, so that it can be reused, without retraining, to generate any number of shapes, with varied structures, whose local details all share a consistent style and appearance. Our detailizer training utilizes a pre-trained multi-view image diffusion model, with text conditioning, to distill the foundational knowledge therein into our detailizer via Score Distillation Sampling (SDS). To improve SDS and enable our detailizer architecture to learn generalizable features over complex structures, we train our model in two training stages to generate shapes with increasing structural complexity. Through extensive experiments, we show that our method generates shapes of superior quality and details compared to existing text-to-3D models under varied structure control. Our detailizer can refine a coarse shape in less than a second, making it possible to interactively author and adjust 3D shapes. Furthermore, the user-imposed structure control can lead to creative, and hence out-of-distribution, 3D asset generations that are beyond the current capabilities of leading text-to-3D generative models. We demonstrate an interactive 3D modeling workflow our method enables, and its strong generalizability over styles, structures, and object categories.

CCS Concepts: • **Computing methodologies** → **Shape modeling**.

Additional Key Words and Phrases: 3D generative model, shape detailization, shape refinement, text-to-3D, knowledge distillation

1 INTRODUCTION

3D generative models are becoming increasingly powerful, enabling the creation of 3D content with ease from texts [Li et al. 2024b; Poole et al. 2023; Shi et al. 2024] or images [Hong et al. 2024; Liu et al.



(a) Coarse shape (b) Coin3D (c) CLAY (d) Ours

Fig. 2. Comparing to state-of-the-art generators on out-of-distribution coarse structures (a). Coin3D [Dong et al. 2024] (b) is unable to produce proper textures based on the prompts, while CLAY [Zhang et al. 2024] (c) falls short in respecting the input structures. Our method performs better on both fronts, e.g., maple leaf-shaped backs with curved armrests and legs as matching styles, and the multiple legs and heads of the bulldog.

2023; Xiang et al. 2024]. However, artistic creation involves realizing the design vision of the artist, which often requires precise control over both the coarse structure and the local details of the generated object. Such precision cannot be fully achieved through text or image inputs alone. In addition, the artist's creative exploration of the design space is greatly facilitated by the ability to quickly generate detailed 3D assets via structural variations.

Prior works propose generative models which rely on user-defined coarse shapes to control the structure of the generated shapes, e.g., by formulating the problem as that of 3D voxel up-resolution [Chen et al. 2024a, 2023b, 2021; Ren et al. 2024; Shen et al. 2021a], while some others [Hui et al. 2024; Zhang et al. 2024] develop 3D generative models with coarse shapes as conditioning. However, as their models were trained on limited 3D shapes, they face generalizability issues and cannot produce local details of arbitrary desired styles. Some approaches [Chen et al. 2023a; Metzer et al. 2023] optimize the initial coarse shape via Score Distillation Sampling (SDS) using text-to-image diffusion models, which possess much stronger generalizability as they were trained on large image collections. Yet the structures of their generated shapes often deviate from those of the input shapes. In addition, each shape takes a significant amount of time to optimize, ranging from minutes to hours. More recently, some approaches [Chen et al. 2024b; Dong et al. 2024] have improved the structure adherence by adopting finetuned multi-view diffusion models conditioned on coarse shape inputs.

Despite such advances, a common issue with all methods relying on image diffusion models is that they can only generate shapes that are "ordinary" with respect to the pretrained diffusion models — they typically fail when the structure of the conditioning coarse shape is out of distribution, e.g., see examples from Figures 1 and a comparison to current generation models in Figure 2. Figure 3 further shows that state-of-the-art text-to-3D and text-to-image models are unable to deal with such out-of-distribution examples. Also, as these models perform *per-shape* optimization with no style consistency across different shapes, they cannot generate a coherent

(a) TRELLIS (b) Hunyuan3D (c) Stable Diffusion (d) MVDream

Fig. 3. State-of-the-art text-to-3D and text-to-image models are unable to generate ***"a cute bulldog with two heads and six legs"*** from text.

collection of 3D shapes with varied structures under the same style prompt, as shown by the "furniture" example in Figure 1-right.

In this work, we aim to tackle the above issues by introducing a 3D detailization model, or a *detailizer*, which is trained using a text prompt and can transform a coarse 3D shape proxy into a high-quality asset with detailed geometry and texture. The text prompt defines the shape class and guides the stylization and detailization of any number of coarse 3D shapes with varied structures. On the other hand, the shape proxy, which can be easily adjusted (e.g., via user editing), provides structure control over the final shape.

Our method relies on a pretrained multi-view image diffusion model with text conditioning, to achieve the generation of different possible styles. Given a text description of the desired style, instead of optimizing a single shape with SDS as in prior works [Chen et al. 2023a, 2024b; Dong et al. 2024; Metzer et al. 2023], we *distill a 3D generative model* as our detailizer, so that it can be reused, without retraining, to generate any number of shapes, with varied structures, whose local details all share a consistent style. We leverage a 3D convolutional neural network (CNN) as the backbone for our detailizer, so that the locality of the convolution operators helps our model learn localized features and enables it to handle coarse shapes of arbitrary structures. During training, we utilize a structure-matching loss defined on the rendered images of the generated shape and the input coarse shape, so the generated shape matches the structure of the input. To facilitate the learning of generating complex structures, we train the detailizer in two training stages to generate shapes with increasing structural complexity. When the training is done for the given text prompt, our detailizer is deployable as a feed-forward model that can detailize a coarse 3D shape in *less than a second*. This allows interactive exploration of structurally varying 3D shape designs in a common style space.

Our method is coined ART-DECO: Arbitrary Text guidance for 3D Detailizer Construction. We demonstrate both quantitatively and qualitatively that ART-DECO generates 3D shapes of superior quality and details compared to state-of-the-art 3D generative models with structure control, such as ShaDDR [Chen et al. 2023b], CLAY [Zhang et al. 2024], and Coin3D [Dong et al. 2024], especially when the input structure may be out-of-distribution and exhibit creativity; see Figure 2. We also show that ART-DECO can be trained using a single prompt to reference a *generic term* such as "furniture." Then, the same detailizer can be reused to quickly generate structurally varying shapes spanning *multiple furniture categories* such as chairs, tables, beds, etc., which share a common style, e.g., "leather, plush surface", as also prescribed in the training prompt; see Figure 1. We further demonstrate an interactive 3D modeling

workflow ART-DECO enables, and its generalizability to different out-of-distribution structures in an interactive application.

2 RELATED WORK

3D generative models. 3D generation methods went through drastic development in the recent decade. Early works mainly focus on building generative models under various 3D representations, such as voxels [Choy et al. 2016; Wu et al. 2016], point clouds [Achlioptas et al. 2018; Fan et al. 2017; Nichol et al. 2022], implicit fields [Chen and Zhang 2019; Mescheder et al. 2019; Park et al. 2019], and polygon meshes [Nash et al. 2020; Shen et al. 2024; Siddiqui et al. 2024]. These methods are trained with various generative frameworks such as variational autoencoders [Kingma and Welling 2014], generative adversarial networks [Goodfellow et al. 2020], and denoising diffusion probabilistic models [Ho et al. 2020; Song et al. 2021]. Due to data scarcity, these methods have limited generalization ability beyond the training distribution.

With the success of image diffusion models [Rombach et al. 2022; Saharia et al. 2022], DreamFusion [Poole et al. 2023] and subsequent works [Chen et al. 2023a; Lin et al. 2023; Melas-Kyriazi et al. 2023; Wang et al. 2023] distill the 2D image prior in the diffusion models into 3D shapes by optimizing individual shapes with Score Distillation Sampling (SDS), therefore achieving zero-shot text-to-3D generation. Several works [Lorraine et al. 2023; Qian et al. 2024; Xie et al. 2024] also attempt to distill feed-forward text-to-3D generative models with SDS. While their generalization ability has been greatly improved, they still lack 3D understanding, which often leads to suboptimal results such as the multi-face Janus problem [Chen et al. 2023a]. Zero123 [Liu et al. 2023] finetunes image diffusion models on large 3D object datasets such as Objaverse [Deitke et al. 2023], to make the diffusion model capable of generating novel-view images conditioned on a single input image and the target viewpoints. Later works including MVDream [Shi et al. 2024] and others [Liu et al. 2024; Long et al. 2024; Shi et al. 2023; Xiang et al. 2023] focus on improving multi-view consistency by generating all the views together and introducing extra information to condition the diffusion process. These multi-view images can then be used to reconstruct a 3D shape via differentiable rendering [Kerbl et al. 2023; Mildenhall et al. 2021; Wang et al. 2021] or via a feed-forward 3D shape reconstruction network [Hong et al. 2024; Li et al. 2024b].

Recently, a few methods [Chen et al. 2024c; Hui et al. 2024; Li et al. 2024a; Ren et al. 2024; Wu et al. 2024; Xiang et al. 2024; Zhang et al. 2024] have been proposed to train 3D diffusion models directly from large 3D datasets, bypassing the intermediate image diffusion model. Nevertheless, the existing 3D generative models primarily focus on text or image-conditioned generation. They often lack precise control over the overall structure of the generated 3D shapes, therefore making it difficult to be incorporated into an artist's workflow. In contrast, our method trains a feed-forward model that detailizes a coarse shape with a style specified by a text prompt, making it possible to create 3D shapes in an interactive manner where an artist can make adjustments to the coarse shape for refinement.

Geometric detailization. Many traditional [Kajiya and Kay 1989; Neyret 1998; Zhou et al. 2006] and learning-based [Berkiten et al. 2017; Hertz et al. 2020; Yifan et al. 2022] approaches have been proposed to synthesize geometric details on coarse shapes by transferring geometric textures, which are often represented as displacement maps or geometric texture patches, from detailed shapes. In addition, neural subdivision [Liu et al. 2020] and subsequent works [Chen et al. 2023c; Chen and Zhang 2021; Shen et al. 2021b] can also learn local geometric details in training shapes and apply them to new shapes. In another line of work, methods that are based on voxel representation can generate geometric details by replicating local voxel patches in detailed reference shapes [Chen et al. 2024a, 2023b, 2021; Sun et al. 2022]. However, these methods all require detailed 3D exemplar shapes provided as style references, which limits their capability when the exemplar shapes with desired style are scarce or unavailable. To mitigate the issue of data scarcity, some recent methods [Chen et al. 2023a; Gao et al. 2023; Metzer et al. 2023; Michel et al. 2022] propose to utilize image-space supervision provided by CLIP [Radford et al. 2021] or SDS, where the style is provided by an input text. However, these methods take a long time to converge, hindering the possibility of an interactive modeling experience. Most recent works [Chen et al. 2024b; Dong et al. 2024] finetune multi-view image diffusion models conditioned on input coarse shapes, which provide faster convergence and better structure adherence. Despite these advancements, a common issue with these methods is that they can only generate shapes within the training distribution and fail when the structure of the conditioning coarse shape is uncommon. Our work addresses this issue by distilling a generative model which is designed to be generalizable and trained with increasing structure complexity.

3 METHOD

In this section, we detail the design and the training of our detailizer. An overview is provided in Figure 4. Our detailizer upsamples an input coarse shape represented as binary occupancy voxels into a detailed 3D shape represented by a volume radiance field. Specifically, given an input coarse voxel grid of resolution k^3, our detailizer networks will generate two voxel grids of resolution K^3 to store a density field and a albedo field. The generated shape can then be rendered via volumetric rendering from the two fields; alternatively, a mesh with textures can be exported to be visualized. In this paper, we always export meshes for visualization. We use $k = 32$ and $K = 128$ in our experiments.

3.1 Network architecture

We utilize 3D convolutional networks as backbone for our detailizer, as inspired by prior voxel detailization works [Chen et al. 2024a, 2023b, 2021] which showed great generalizability on arbitrary input coarse shapes. The detailed network architecture can be found in the supplementary. The input to our detailizer is a coarse occupancy voxel grid $v \in \{0,1\}^{k\times k\times k}$. The detailizer has two separate upsampling networks: $\mathbf{G}_d$ for generating the density field $v'_d = \mathbf{G}_d(v) \in [0,+\infty)^{K\times K\times K}$, and $\mathbf{G}_a$ for the albedo field $v_a = \mathbf{G}_a(v) \in [0,1]^{K\times K\times K\times 3}$, where each voxel stores an RGB color. To enforce that the generated shape follows the structure of the input voxels, we dilate the input voxel grid v by one voxel and then upsample it to K^3 resolution via nearest neighbor to obtain a binary mask v_m. We then apply the mask to the output density field v'_d

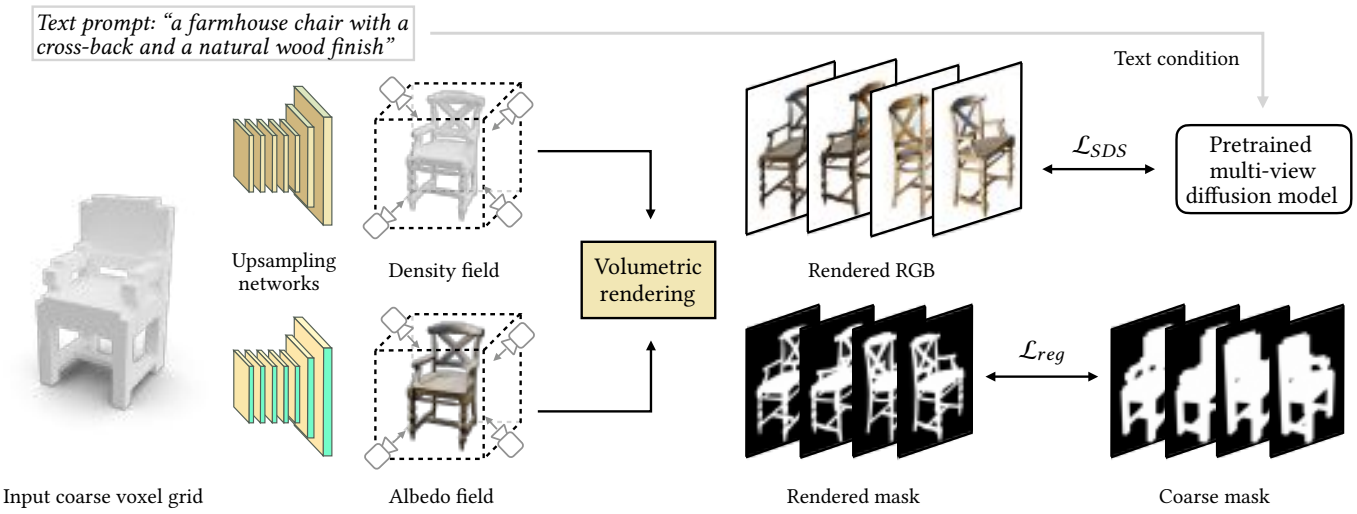


Fig. 4. Overview of the training of our detailizer. Given a coarse voxel grid and a text prompt that describes a style, two 3D convolutional networks upsample the coarse voxels into high-resolution density and albedo fields, respectively. Multi-view images are then rendered from the density and albedo fields, and a pretrained multi-view diffusion model conditioned on the text prompt is used as a prior for Score Distillation Sampling ($\mathcal{L}_{SDS}$). The regularization loss ($\mathcal{L}_{reg}$) measures the similarity between the masks rendered from the generated shape and those from the input coarse voxel grid, thus enforcing the structure of the generated shape to be consistent with the input coarse voxels.

via element-wise multiplication to obtain the final density field: $v_d = v'_d \cdot v_m$. This masking step prevents the network from generating artifacts or unnecessary details far away from the structure of the input coarse shape, and allows the network to focus its capacity on generating plausible details only within the valid region v_m.

3.2 Training

To train the detailizer, we run the networks on a training set of coarse voxel grids. For each voxel grid, we generate the density field v_d and the albedo field v_a, render multi-view images from them via volumetric rendering [Mildenhall et al. 2021], and leverage a pretrained multi-view diffusion model to provide training supervision via SDS [Poole et al. 2023; Shi et al. 2024]. Formally, given the density field v_d, the albedo field v_a, the camera parameters $\mathbf{c}_i$ (for the i-th view), the volumetric rendering function $\mathbf{R}(\cdot)$, a text prompt y, and the pretrained multi-view diffusion model ϵ_θ, we first render multi-view images $\mathbf{x}_i = \mathbf{R}(v_d, v_a; \mathbf{c}_i)$. Note that ϵ_θ operates on multi-view images, so in each iteration, we render a set of images $\{\mathbf{x}_i\}_{i=1,\ldots,N}$ from the same shape, and then we can use ϵ_θ to obtain the denoised images of $\mathbf{x}_i$: $\{\hat{\mathbf{x}}_i\} = \epsilon_\theta(\{\mathbf{x}_i\}_t; y, \{\mathbf{c}_i\}, t)$, where t is the time step and $\{\mathbf{x}_i\}_t$ are input images after adding noise ϵ for time step t. Now we can define our multi-view SDS loss as

$$\mathcal{L}_{SDS} = \mathbb{E}_{t,\epsilon,\mathbf{c}_i} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2. \tag{1}$$

We use MVDream [Shi et al. 2024] as our pretrained multi-view diffusion model, which is a model fine-tuned from Stable Diffusion [Rombach et al. 2022] on the Objaverse dataset [Deitke et al. 2023] to generate $N = 4$ views at the same time.

SDS alone does not guarantee that the structure of the generated shape is consistent with the input coarse voxels. Even with the masking step described in Section 3.1 so that the network can only generate details in the coarse voxels' vicinity, the network often produces 3D shapes that miss certain structures, such as armrests in a chair. Therefore, to ensure the generated shape fully respects the structure of the input coarse voxels, we enforce the rendered mask (the alpha/transparency channel in the rendered image) of the generated shape to be similar to the rendered mask of the input coarse voxel grid when the masks are rendered from the same camera pose. We formulate this constraint as a regularization loss

$$\mathcal{L}_{reg} = \mathbb{E}_{\mathbf{c}_i} \|\mathbf{m}_i - \hat{\mathbf{m}}_i\|_2^2 \tag{2}$$

where $\mathbf{m}_i$ is the rendered mask of the generated shape from camera pose $\mathbf{c}_i$, and $\hat{\mathbf{m}}_i$ is the rendered mask from the input coarse voxels.

The final loss function to train our model is defined as

$$\mathcal{L} = \mathcal{L}_{SDS} + \lambda_{reg} \cdot \mathcal{L}_{reg} \tag{3}$$

where λ_{reg} is a hyper-parameter balancing between distilling plausible shapes from SDS and preserving the structure of the inputs. During training, we set $\lambda_{reg} = 10^4$, and gradually reduce the value to 10, allowing the network to focus on generating structures in the early stages of training and refining local details later.

3.3 Data

Our input text prompt specifies the desired style and indicates the generic shape category of the generated shapes, such as chairs; however, it does not provide detailed structural descriptions. During training, we need coarse voxel grids of that shape category to serve as input to our detailizer. For a specific shape category, if a decently large 3D dataset is available, we simply voxelize the existing shapes into coarse voxel grids as our training set. Otherwise, we download a few shapes (16-32) from the Internet, an adopt a similar strategy to that of DECOLLAGE [Chen et al. 2024a] to generate a large number of coarse voxel grids with diverse structures via data augmentation. Specifically, we randomly scale the existing shapes in x, y, and z

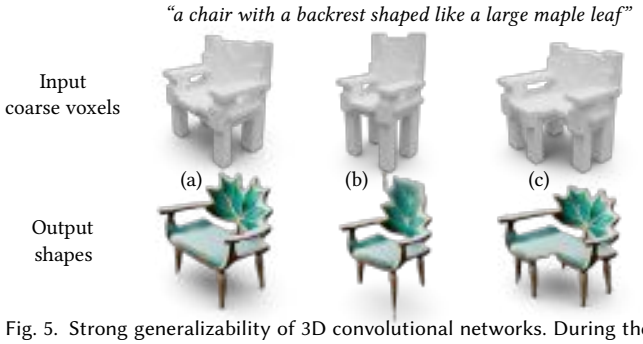


Fig. 5. Strong generalizability of 3D convolutional networks. During the first training stage, our detailizer is trained on a single coarse voxel grid to generate a single detailed shape (a). After training, when tested on different structures, our detailizer demonstrates strong generalizability and produces reasonable results (b, c). These results are not perfect, but they can be good initial points for the image diffusion model to refine in our second training stage, where we use multiple coarse shapes.

directions, and apply random rotations and merge multiple shapes into a single shape to enrich the geometric variation. See Section 4 for more details about the shapes used in our experiments.

3.4 Two-stage Learning

When our detailizer was trained with all the coarse shapes in the training set (which contains both simple and complex structures; see Figure 11 for examples), it failed to generate certain structures, as shown in Figure 9. This is because the multi-view image diffusion model tends to prioritize generating shapes that align with its learned biases, i.e., shapes with simple and common structures. Therefore, when the model is trained with complex structures from the very beginning, it learns to simplify the structure rather than adhering to it. To address this, we have observed that 3D convolutional networks have strong generalizability even when trained on a single simple shape, thanks to the locality and inductive bias of convolution operations. As shown in Figure 5, our detailizer trained on a single coarse voxel grid generalizes reasonably well on other inputs during inference, even when the input has complex structures. This model with single-shape training can be a good initialization for multi-shape training by providing good initial shapes for the image diffusion model to refine. Therefore, we employ a two-stage training scheme. In the first stage, we train the model on a single input coarse voxel grid with a simple structure. And in the second stage, we keep training the model on all the coarse shapes in our training set until convergence.

In our experiments, we train individual models for different text prompts. The first stage of training takes about 1.5 hours and the second stage takes about 3.5 hours on a single NVIDIA 4090 GPU.

4 EXPERIMENTS

In this section, we test our method on text prompts describing different styles from various shape categories. We show that our detailizer can generate high-quality shapes that adhere to both the text prompt and the structure of the input coarse shape. We compare with state-of-the-art 3D generative models with text and structural control in Section 4.1 to demonstrate the effectiveness of our proposed approach. We then validate our design via ablation study in Section 4.2.

Table 1. Quantitative comparison of text-guided 3D generation with structural control. Metrics are averaged over the evaluated text prompts.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	201.780	20.129	0.620	0.715
Coin3D	200.935	20.250	0.566	0.678
CLAY	175.402	23.943	0.626	0.725
Ours	**171.703**	**26.047**	**0.639**	**0.739**

Finally, we showcase several applications enabled by our method in Section 4.3, including style-consistent detailization for shapes in multiple categories, and an interactive modeling application.

Datasets. We evaluate our model on seven shape categories: chair, table, couch, bed, building, animal, and cake. For each category, we collect 1,000 to 2,000 shapes from ShapeNet [Chang et al. 2015], 3D Warehouse [Trimble Inc. 2014], and Objaverse [Deitke et al. 2023] and voxelize them to obtain coarse voxels for training. See supplementary material for details.

Evaluation metrics. For quantitative evaluation, we adopt the Render-FID and CLIP Score proposed in LATTE3D [Xie et al. 2024] to evaluate the fidelity of the generated shapes and their consistency against the text prompt. *Render-FID* measures the Fréchet Inception Distance (FID) between the rendered images of the generated shapes and the images sampled from Stable Diffusion with the same text prompts. It evaluates how closely the generated shapes align with the 2D prior in the image diffusion model. *CLIP Score* measures the average CLIP score [Radford et al. 2021] between the text prompt and each rendered image of the generated shape. It evaluates how closely the generated shapes align with the input text prompt. We also adopt Strict-IoU and Loose-IoU proposed in DECOR-GAN [Chen et al. 2021] to measure the consistency between the structures of the generated shape and its input coarse voxel grid. *Strict-IoU* computes the Intersection over Union (IoU) between the the input voxels and the occupancy voxels obtained by voxelizing the generated shape. *Loose-IoU* is a less restrictive version of Strict-IoU which computes the proportion of occupied voxels in the input voxels that are also occupied in the occupancy voxels of the generated shape. Generating a single shape using CLAY's commercial product, Rodin, takes about 5-10 minutes, while Coin3D requires about 25 minutes to optimize one, which makes both approaches impractical for large-scale automatic evaluation. Therefore, we randomly select 8 text prompts for testing. To prepare the voxel grids for testing, for each text prompt, we randomly sample 10 coarse shapes in the same way we obtained the training coarse shapes. We run all methods on those shapes, compute the metrics, and take the average. We report the average across all text prompts in the paper, and the average for each text prompt in the supplementary material.

4.1 Text-guided detailization

We compare our method with several methods that can generate 3D shapes with structural control: CLAY [Zhang et al. 2024], Coin3D [Dong et al. 2024], and ShaDDR [Chen et al. 2023b]. *CLAY* is a feed-forward model trained directly on large 3D datasets. It is able to take a coarse voxel grid as structural guidance in addition to its text conditioning. However, its code is not publicly available. We

(a) Input coarse voxels (b) Ours + SD (c) Ours + MVDream

Fig. 6. Ablation study on the choice of image diffusion models. (a) We train both models with the same coarse input in the first stage. (b) Our method with a single-view diffusion model (Stable Diffusion) as SDS guidance struggles to produce a complete shape. (c) Our method with a multi-view diffusion model (MVDream) as guidance produces high-quality results.

instead use Rodin[1], a commercial product built on and powered by CLAY, as a substitute for testing. *Coin3D* is an optimization-based framework for refining coarse shapes. We run the officially released code for testing in our experiments. *ShaDDR* is a shape detailization model whose network architecture and input/output closely resemble those of our detailizer. However, ShaDDR requires ground truth 3D shapes as style reference. Therefore, for each text prompt, we use one detailed shape generated by our method for training ShaDDR. We provide the training shapes in the supplementary and verify that the shapes are in good quality.

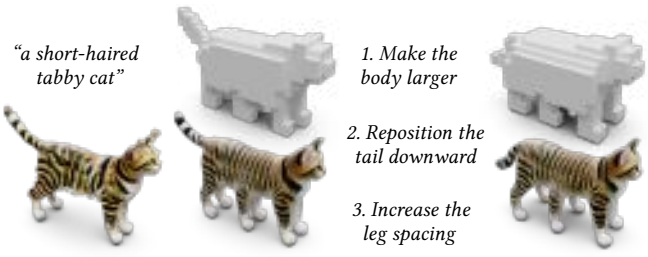
We showcase qualitative results of our method in Figure 12, and compare with other methods in Figure 14. In Figure 14, Coin3D struggles to generate accurate geometry and texture when conditioned on complex structures. This is primarily because the multi-view diffusion model that Coin3D is based on does not generalize well to uncommon structures. We provide additional results in the supplementary material to show the intermediate outputs from Coin3D and to demonstrate that Coin3D works on simple structures but fails on complex structures. ShaDDR was designed to be trained with clean, high-quality 3D shapes. Therefore, when trained with noisy shapes obtained from SDS optimization, the quality of ShaDDR's output becomes unsatisfactory. CLAY has cleaner geometry and better structure preservation as it was trained on real 3D data. Nonetheless, it often fails to generate detailed local geometries and textures. CLAY can also deviate significantly from the structure of the input coarse shape if the structure is uncommon, see the table and animal examples in Figure 14. In comparison, our method can effectively handle both out-of-distribution coarse structures and creative text prompts. It can provide users with flexible control over the structure and the style while generating high-quality shapes. We also report quantitative comparisons in Table 1, which further demonstrates the superior performance of our method.

4.2 Ablation Study

Image diffusion model: single-view vs. multi-view. Since we leverage pretrained image diffusion models and distill their knowledge into our detailizer, different diffusion models can significantly affect the generation quality of 3D shapes. Therefore, we conduct an ablation study using two different types of diffusion models: Stable Diffusion, a single-view image diffusion model trained on natural images, which possesses 2D image prior; and MVDream [Shi et al. 2024], a multi-view image diffusion model finetuned on multi-view rendered images of 3D objects, thus incorporating additional 3D prior. In both settings, we use the same network and loss functions, with the only difference being the choice of image diffusion models.

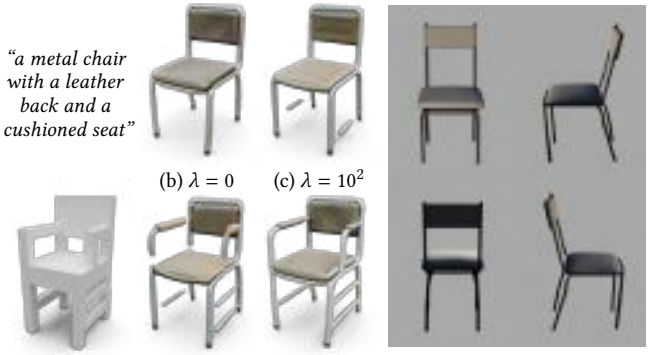
Figure 6 shows the qualitative comparison of the 3D shapes generated by our method using different image diffusion models. Stable Diffusion cannot synthesize a reasonable shape while MVDream can generate a 3D shape with significantly improved geometric and texture details, which demonstrates the importance of the 3D prior in the image diffusion model and the benefit of having multi-view-consistent distillation during SDS.

[1] https://hyper3d.ai/



(a) Baseline (b) Ours (c) Ours

Fig. 7. Ablation study on the masked MVDream baseline. The baseline (a) fails to preserve the input structure (see (b) top). Our result (b) fully respects the structure, and changes its structure according to new edits (c).



(a) Input voxels (b) $\lambda = 0$ (c) $\lambda = 10^2$ (d) $\lambda = 10^3$ (e) $\lambda = 10^4$ (f) Images from MVDream

Fig. 8. Ablation study on different regularization weights λ_{reg}. (b-e) show that with increasing λ_{reg}, the generated shape becomes more consistent with the input coarse structure (a). We also show example images sampled from MVDream in (f), which tends to produce simple structures.

Masked MVDream baseline. To show the necessity of distilling a generative neural network rather than distilling a single shape, we adopt MVDream to directly optimize a neural field while enforcing structural control by masking the neural field with the input coarse voxels. As shown in Figure 7 (a), the baseline fails to preserve the input coarse structure, as MVDream's multi-view diffusion model lacks prior knowledge of atypical structures, such as a cat with six legs. In contrast, our model, although trained on coarse voxels of *four-legged* animals using the same multi-view diffusion model,

(a) Input coarse voxels (b) *w/o* two-stage learning (c) *w* two-stage learning

Fig. 9. Ablation study on two-stage learning. When skipping the first-stage training, the detailizer learns to simplify the structure of the generated shape (b) so the shape resembles a round, real-life cake; it does not strictly follow the structure of the input coarse shape (a). With both training stages, the detailizer correctly learns to generate the shape (c) in an uncommon structure. The regularization loss is used in both settings.

Table 2. Quantitative ablation study on the proposed regularization loss. Metrics are averaged over the evaluated text prompts.

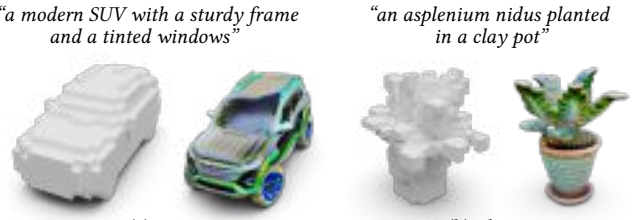
	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{reg} = 0$	**171.388**	25.718	0.585	0.693
$\lambda_{reg} = 10^2$	171.684	25.621	0.584	0.690
$\lambda_{reg} = 10^3$	171.552	25.423	0.623	0.728
$\lambda_{reg} = 10^4$	171.703	**26.047**	**0.639**	**0.739**

demonstrates strong generalization ability to unseen structures, thanks to the inductive bias of convolutional networks. Moreover, our model can refine new edits in a single forward pass in less than a second without retraining, whereas optimization-based methods require retraining for each new structural change; see Figure 7 (c).

Opacity regularization loss. As explained in Section 3.2, the pretrained image diffusion model tends to generate simple and common shapes due to its learned biases. In Figure 8 (f), we show that when armrests and stretchers are not explicitly mentioned in the text prompt, images sampled from MVDream often lack these structures, leading to missing parts in the generated 3D shapes, even though the parts exist in the input coarse voxel grid; see Figure 8 (a) and (b). Since our model is trained on shapes with diverse structures, such parts should be handled automatically by the network without explicit text descriptions. Therefore, we use the regularization loss defined on the rendered masks to ensure structural consistency.

Figure 8 (b-e) show qualitative results using the regularization loss with different λ_{reg} values. By using a larger λ_{reg}, the generated shape can faithfully preserve the structure of the coarse voxels, even when the structure is asymmetric, e.g., with different numbers of stretchers and varying armrest lengths on each side. This is also reflected by the higher IoU in Table 2. Note that the Render-FID of the model without the regularization loss, i.e. $\lambda_{reg} = 0$, is slightly higher, since the generated shapes are more aligned with the preferences of the image diffusion model.

Two-stage learning. As discussed in Section 3.4, the image diffusion model's learned prior exhibits strong biases toward common structures. In Figure 9 (b), when the text prompt indicates that the shape is to be a cake, despite that the input coarse shape is not round, the model attempts to generate a round cake, although a quarter of the cake is cut off by our masking (see Section 3.1). By using our two-stage training strategy, the model effectively preserves the structure of the input coarse voxel grids while maintaining high-quality geometry and texture.



(a) Car (b) Plant

Fig. 10. Limitations. Our current representation is not suitable for generating shapes with glossy/reflective material or shapes with thin threads/surfaces.

4.3 Applications

Cross-category detailization. By using a highly generic text prompt that can describe multiple shape categories, such as "furniture" for chairs, tables, couches, and beds, our detailizer can generate a diverse collection of detailed 3D shapes across these categories, while maintaining consistent styles throughout, as shown in Figure 1, where we have trained a single model using the coarse voxel grids from all four categories. Note that we did not provide any class labels to the network during training. Yet once trained, our detailizer can automatically identify the category of the input coarse voxel grid and upsample it into the corresponding detailed 3D shape.

Interactive modeling. Leveraging the feed-forward model design, our detailizer is capable of upsampling an input coarse voxel grid into a detailed 3D shape in under one second, making it possible to be incorporated into an interactive modeling application where users can refine and iterate on designs efficiently; see Figure 13. We have developed an interactive modeling interface that enables users to edit a coarse voxel grid, select a text prompt, and visualize the resulting detailed and textured 3D shape; see supplementary video.

5 CONCLUSIONS

We introduce ART-DECO, a shape detailization model that can refine a coarse shape of arbitrary structure into a detailed 3D shape. By distilling the knowledge in a pretrained multi-view image diffusion model, our detailizer learns to generate local details with the style described in an input text prompt. Once trained, our detailizer can be used to detailize given coarse shapes into detailed shapes with a consistent style. We demonstrate the superior performance of our method in the experiments and in an interactive modeling workflow.

The dense voxel representation that our method adopts enables us to take advantage of convolutional networks. However, it also limits the resolution of our generated shapes. Alternative representations and network design can be explored to address this issue. Our representation stores a fixed color in each voxel instead of view-dependent colors, therefore it cannot represent shapes with glossy or reflective surfaces, see Figure 10. Our detailizer requires re-training for each different text prompt, which can be inconvenient for fast prototyping. With sufficient compute, we believe it is

possible to distill a more generic detailizer that can take both the text prompt and the coarse shape as input during inference, and generate detailed shapes at an interactive speed.

ACKNOWLEDGMENTS

This work was done during the first author's internship at Adobe Research, and it is supported in part by an NSERC grant (No. 611370) and a gift fund from Adobe Research.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. 2018. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*. PMLR, 40–49.

S. Berkiten, M. Halber, J. Solomon, C. Ma, H. Li, and S. Rusinkiewicz. 2017. Learning detail transfer based on geometric features. *Computer Graphics Forum* 36, 2 (May 2017), 361–373.

Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. 2015. ShapeNet: An Information-Rich 3D Model Repository. *CoRR* abs/1512.03012 (2015). arXiv:1512.03012 http://arxiv.org/abs/1512.03012

Qimin Chen, Zhiqin Chen, Vladimir G Kim, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. 2024a. DECOLLAGE: 3D Detailization by Controllable, Localized, and Learned Geometry Enhancement. In *European Conference on Computer Vision*.

Qimin Chen, Zhiqin Chen, Hang Zhou, and Hao Zhang. 2023b. ShaDDR: Interactive Example-Based Geometry and Texture Generation via 3D Shape Detailization and Differentiable Rendering. In *SIGGRAPH Asia 2023 Conference Papers*.

Rui Chen, Yongwei Chen, Ningxin Jiao, and Kui Jia. 2023a. Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22246–22256.

Yun-Chun Chen, Vladimir Kim, Noam Aigerman, and Alec Jacobson. 2023c. Neural Progressive Meshes. In *ACM SIGGRAPH 2023 Conference Proceedings*. 1–9.

Yun-Chun Chen, Selena Ling, Zhiqin Chen, Vladimir G. Kim, Matheus Gadelha, and Alec Jacobson. 2024b. Text-guided Controllable Mesh Refinement for Interactive 3D Modeling. *SIGGRAPH Asia (Conference track)* (2024).

Zhiqin Chen, Vladimir G. Kim, Matthew Fisher, Noam Aigerman, Hao Zhang, and Siddhartha Chaudhuri. 2021. DECOR-GAN: 3D shape detailization by conditional refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 15740–15749.

Zhaoxi Chen, Jiaxiang Tang, Yuhao Dong, Ziang Cao, Fangzhou Hong, Yushi Lan, Tengfei Wang, Haozhe Xie, Tong Wu, Shunsuke Saito, et al. 2024c. 3dtopia-xl: Scaling high-quality 3d asset generation via primitive diffusion. *arXiv preprint arXiv:2409.12957* (2024).

Zhiqin Chen and Hao Zhang. 2019. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5939–5948.

Zhiqin Chen and Hao Zhang. 2021. Neural marching cubes. *ACM Transactions on Graphics (TOG)* 40, 6 (2021), 1–15.

Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 2016. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer, 628–644.

Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. 2023. Objaverse: A universe of annotated 3d objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13142–13153.

W. Dong, B. Yang, L. Ma, X. Liu, L. Cui, H. Bao, and Z. Cui. 2024. Coin3D: Controllable and Interactive 3D Assets Generation with Proxy-Guided Conditioning. In *ACM SIGGRAPH 2024 Conference Papers*. 1–10.

Haoqiang Fan, Hao Su, and Leonidas J Guibas. 2017. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 605–613.

William Gao, Noam Aigerman, Thibault Groueix, Vladimir G. Kim, and Rana Hanocka. 2023. TextDeformer: Geometry Manipulation using Text Guidance. *SIGGRAPH (Conference track)* (2023).

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM* 63, 11 (2020), 139–144.

Amir Hertz, Rana Hanocka, Raja Giryes, and Daniel Cohen-Or. 2020. Deep Geometric Texture Synthesis. *ACM Trans. Graph.* 39, 4, Article 108 (2020). https://doi.org/10.1145/3386569.3392471

Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33 (2020), 6840–6851.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2024. LRM: Large Reconstruction Model for Single Image to 3D. (2024). https://openreview.net/forum?id=sllU8vvsFF

Ka-Hei Hui, Aditya Sanghi, Arianna Rampini, Kamal Rahimi Malekshan, Zhengzhe Liu, Hooman Shayani, and Chi-Wing Fu. 2024. Make-a-shape: a ten-million-scale 3d shape model. In *Forty-first International Conference on Machine Learning*.

James T. Kajiya and Timothy L. Kay. 1989. Rendering fur with three dimensional textures. *ACM Siggraph Computer Graphics* 23, 3 (1989), 271–280.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 2023. 3D Gaussian Splatting for Real-Time Radiance Field Rendering. *ACM Transactions on Graphics (TOG)* 42, 4 (2023), 139:1.

Diederik P. Kingma and Max Welling. 2014. Auto-Encoding Variational Bayes. (2014). http://arxiv.org/abs/1312.6114

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. 2024b. Instant3D: Fast Text-to-3D with Sparse-view Generation and Large Reconstruction Model. (2024). https://openreview.net/forum?id=2lDQLiH1W4

Wenhao Li, Jiahui Liu, Rongjie Chen, Yunchao Liang, Xuaner Chen, Ping Tan, and Xiaohu Long. 2024a. CraftsMan: High-Fidelity Mesh Generation with 3D Native Generation and Interactive Geometry Refiner. *arXiv preprint arXiv:2405.14979* (2024).

Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 300–309.

Hsueh-Ti Derek Liu, Vladimir G. Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. 2020. Neural Subdivision. *ACM Trans. Graph.* 39, 4 (2020).

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9298–9309.

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2024. SyncDreamer: Generating Multiview-consistent Images from a Single-view Image. (2024). https://openreview.net/forum?id=MN3yH2ovHb

Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2024. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9970–9980.

Jonathan Lorraine, Kevin Xie, Xiaohui Zeng, Chen-Hsuan Lin, Towaki Takikawa, Nicholas Sharp, Tsung-Yi Lin, Ming-Yu Liu, Sanja Fidler, and James Lucas. 2023. Att3d: Amortized text-to-3d object synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 17946–17956.

Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. 2023. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 8446–8455.

Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.

Guy Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. 2023. Latent-NeRF for Shape-Guided Generation of 3D Shapes and Textures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12663–12673.

Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. 2022. Text2Mesh: Text-Driven Neural Stylization for Meshes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 13492–13502.

Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

Charlie Nash, Yaroslav Ganin, SM Ali Eslami, and Peter Battaglia. 2020. Polygen: An autoregressive generative model of 3d meshes. In *International conference on machine learning*. PMLR, 7220–7229.

Fabrice Neyret. 1998. Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics* 4, 1 (1998), 55–70.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-E: A System for Generating 3D Point Clouds from Complex Prompts. *CoRR* abs/2212.08751 (2022). https://doi.org/10.48550/ARXIV.2212.08751 arXiv:2212.08751

Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. 2019. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 165–174.

Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. 2023. DreamFusion: Text-to-3D using 2D Diffusion. (2023). https://openreview.net/forum?id=FjNys5c7VyY

Guocheng Qian, Junli Cao, Aliaksandr Siarohin, Yash Kant, Chaoyang Wang, Michael Vasilkovsky, Hsin-Ying Lee, Yuwei Fang, Ivan Skorokhodov, Peiye Zhuang, Igor Gilitschenski, Jian Ren, Bernard Ghanem, Kfir Aberman, and Sergey Tulyakov. 2024. AToM: Amortized Text-to-Mesh using 2D Diffusion. *CoRR* abs/2402.00867 (2024). https://doi.org/10.48550/ARXIV.2402.00867 arXiv:2402.00867

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

Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. 2024. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4209–4219.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. 2022. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10684–10695.

Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. 2022. Photorealistic text-to-image diffusion models with deep language understanding. *Advances in neural information processing systems* 35 (2022), 36479–36494.

Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. 2021a. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems* 34 (2021), 6087–6101.

Tianchang Shen, Chiyu Jiang, Yifan Jiang, Jun-Yan Zhu, Zhixin Shu, Andrea Tagliasacchi, Leonidas J. Guibas, and Shichen Liu. 2021b. Deep marching tetrahedra: a hybrid representation for high-resolution 3D shape synthesis. In *Advances in Neural Information Processing Systems*, Vol. 34. 6087–6101.

Tianchang Shen, Zhaoshuo Li, Marc Law, Matan Atzmon, Sanja Fidler, James Lucas, Jun Gao, and Nicholas Sharp. 2024. SpaceMesh: A Continuous Representation for Learning Manifold Surface Meshes. In *SIGGRAPH Asia 2024 Conference Papers*. 1–11.

Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023. Zero123++: a Single Image to Consistent Multi-view Diffusion Base Model. *CoRR* abs/2310.15110 (2023). https://doi.org/10.48550/ARXIV.2310.15110 arXiv:2310.15110

Yichun Shi, Peng Wang, Jianglong Ye, Long Mai, Kejie Li, and Xiao Yang. 2024. MVDream: Multi-view Diffusion for 3D Generation. (2024). https://openreview.net/forum?id=FUgrjq2pbB

Yawar Siddiqui, Antonio Alliegro, Alexey Artemov, Tatiana Tommasi, Daniele Sirigatti, Vladislav Rosov, Angela Dai, and Matthias Nießner. 2024. Meshgpt: Generating triangle meshes with decoder-only transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 19615–19625.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. 2021. Score-Based Generative Modeling through Stochastic Differential Equations. (2021). https://openreview.net/forum?id=PxTIG12RRHS

Bo Sun, Vladimir G. Kim, Qixing Huang, Noam Aigerman, and Siddhartha Chaudhuri. 2022. PatchRD: Detail-Preserving Shape Completion by Learning Patch Retrieval and Deformation. *ECCV* (2022).

Trimble Inc. 2014. 3D Warehouse. https://3dwarehouse.sketchup.com/.

Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. 2023. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12619–12629.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. NeuS: Learning Neural Implicit Surfaces by Volume Rendering for Multi-view Reconstruction. (2021), 27171–27183. https://proceedings.neurips.cc/paper/2021/hash/e41e164f7485ec4a28741a2d0ea41c74-Abstract.html

Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. 2016. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. *Advances in neural information processing systems* 29 (2016).

Shuang Wu, Youtian Lin, Feihu Zhang, Yifei Zeng, Jingxi Xu, Philip Torr, Xun Cao, and Yao Yao. 2024. Direct3D: Scalable Image-to-3D Generation via 3D Latent Diffusion Transformer. *arXiv preprint arXiv:2405.14832* (2024).

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

Jiangning Xiang, Jingbo Yang, Binglin Huang, and Xin Tong. 2023. 3D-Aware Image Generation Using 2D Diffusion Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2383–2393.

Kevin Xie, Jonathan Lorraine, Tianshi Cao, Jun Gao, James Lucas, Antonio Torralba, Sanja Fidler, and Xiaohui Zeng. 2024. LATTE3D: Large-scale Amortized Text-To-Enhanced3D Synthesis. *The 18th European Conference on Computer Vision (ECCV)* (2024).

Wang Yifan, Lukas Rahmann, and Olga Sorkine-hornung. 2022. Geometry-Consistent Neural Shape Representation with Implicit Displacement Fields. In *International Conference on Learning Representations*. https://openreview.net/forum?id=yhCp5RcZD7

Longwen Zhang, Ziyu Wang, Qixuan Zhang, Qiwei Qiu, Anqi Pang, Haoran Jiang, Wei Yang, Lan Xu, and Jingyi Yu. 2024. CLAY: A Controllable Large-scale Generative Model for Creating High-quality 3D Assets. *ACM Transactions on Graphics (TOG)* 43, 4 (2024), 1–20.

K. Zhou, X. Huang, X. Wang, Y. Tong, M. Desbrun, B. Guo, and H. Y. Shum. 2006. Mesh quilting for geometric texture synthesis. In *ACM SIGGRAPH 2006 Papers*. 690–697.

Fig. 11. Training set examples of coarse shapes: simple structures are shown in white, and complex ones in light yellow.

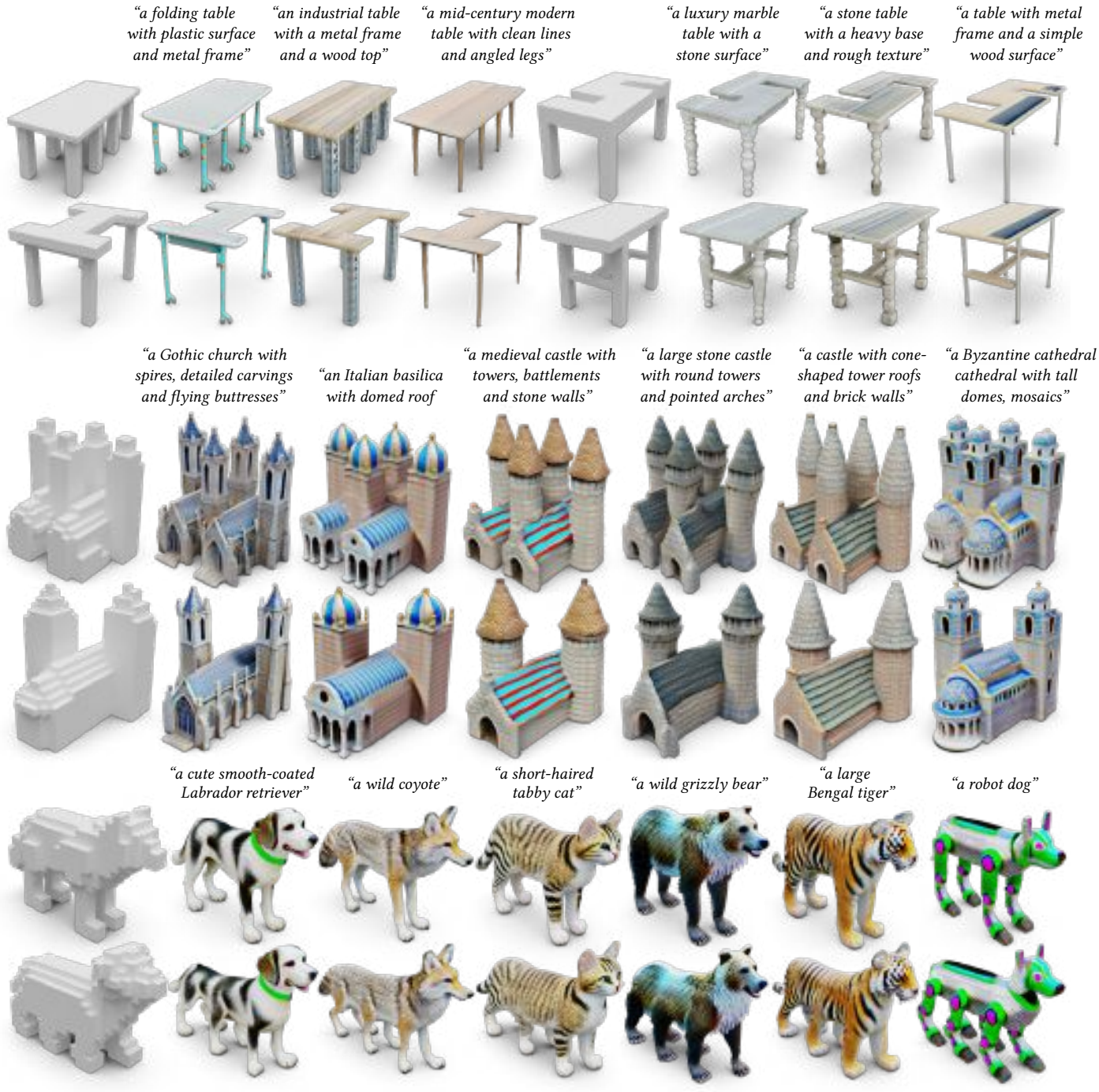


Fig. 12. Results of text-guided detailization with input coarse voxels control. We show the input coarse voxels on the left and the text prompts on the top.

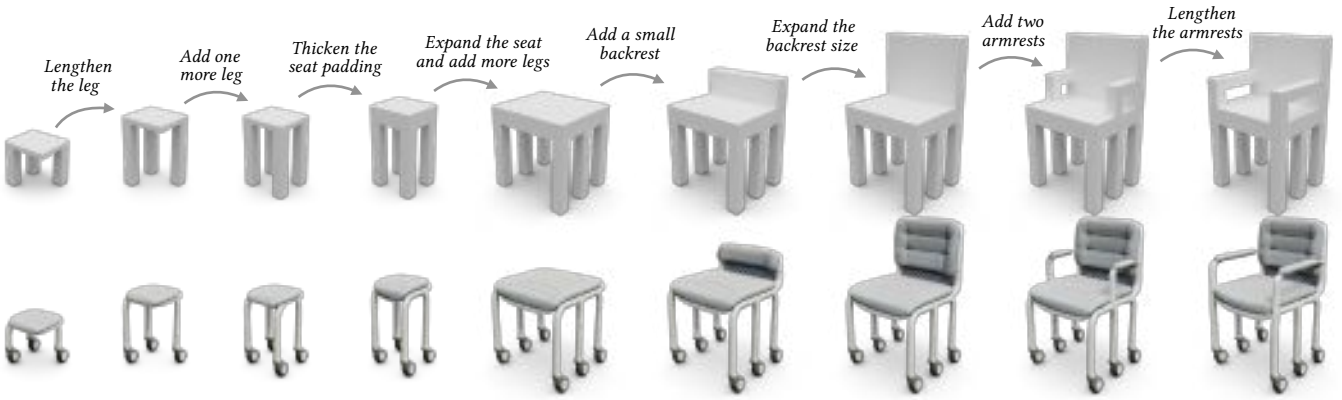


Fig. 13. Example of procedural editing. After training the model with the text prompt *"an office chair with wheels and thick padding"*, the detailization of each edit takes **less than one second**. Our method demonstrates strong robustness to minor modifications and local edits.

Fig. 14. Qualitative comparison of text-guided detailization with input coarse voxels control. Note that even with a simple structure, as shown in the last column, existing methods fail to produce plausible results when given a creative text prompt.

ART-DECO: Arbitrary Text Guidance for 3D Detailizer Construction

(Supplementary Material)

A DATA, TEXT PROMPTS AND CODE

We evaluate our model in seven shape categories: chair, table, couch, bed, building, animal, and cake. We collect 1,505 chairs, 1,292 tables, 300 couches and 300 beds from ShapeNet [Chang et al. 2015], 32 buildings and 16 cakes from 3D Warehouse [Trimble Inc. 2014] under CC-BY 4.0, and 2,082 animals from Objaverse [Deitke et al. 2023]. We voxelize the 3D shapes as our training set. For building and cake, we perform data augmentation to obtain 1,200 voxel grids for each category as our training set.

We use the following text prompts for text-guided detailization.

- *"a farmhouse chair with a cross-back design and a natural wood finish"*
- *"a Queen Anne chair with a leather back in a light, neutral tone and a cushioned seat"*
- *"a metal chair with a leather back and a cushioned seat"*
- *"a chair with a backrest shaped like a large maple leaf"*
- *"a Scandinavian-style chair with a clean design and soft fabric padding"*
- *"an office-style chair with wheels and a thick seat padding"*
- *"a rustic wooden chair with a rough texture and a simple, handcrafted look"*
- *"a Victorian chair with elegant curves and velvet upholstery"*
- *"a traditional Japanese palace with tiled roofs and wooden walls"*
- *"a Gothic church with spires, detailed carvings, and flying buttresses"*
- *"a Russian cathedral with domes and tall spires"*
- *"a large stone castle with round towers and pointed arches"*
- *"an Italian basilica with domed roof and arched windows"*
- *"a castle with cone-shaped tower roofs and brick walls"*
- *"a stone cathedral with a tall rose window and flying arches"*
- *"an old German cathedral with timber framing and steep roof"*
- *"a Gothic church with a bell tower and steep tiled roof"*
- *"a medieval castle with towers, battlements, and stone walls"*
- *"a Byzantine cathedral with tall domes, mosaics"*
- *"a table with a metal frame and a simple wood surface"*
- *"a luxury marble table with a stone surface"*
- *"a stone table with a heavy base and rough texture"*
- *"a folding table with a plastic surface and metal frame"*
- *"a classic Queen Anne table with a smooth and muted finish"*
- *"a mid-century modern table with clean lines and angled legs"*
- *"an industrial style table with a metal frame and a thick wood top"*
- *"a cute golden retriever"*
- *"a cute bulldog"*
- *"a cute smooth-coated Labrador retriever"*
- *"a cute Shiba Inu"*
- *"a wild coyote"*
- *"a wild grizzly bear"*
- *"a large Bengal tiger"*
- *"a robot dog"*
- *"a cake with chocolate dripping down the sides"*

We use the following prompts for cross-category detailization.

- *"a classic furniture piece made of polished wood with subtle details"*
- *"an old Queen Anne style furniture in a light, neutral tone"*
- *"a stylish furniture piece featuring leather and a plush surface in a natural tone"*

We will provide the ready-to-use data and code upon publication.

B NETWORK ARCHITECTURE

For the density upsampling network, we use 5 layers of 3D convolution to extract the features of the input coarse voxel grid, followed by 2 layers of transposed 3D convolution for upsampling. Each upsampling layer doubles the input resolution. For the albedo upsampling network, we use the same network architecture as the density upsampling network, except that the final upsampling layer outputs three channels to represent RGB values.

We use a learning rate of 10^{-4}, a batch size of 1 and Adam optimizer for all experiments.

C EVALUATION METRICS

We use Render-FID and CLIP score to quantitatively evaluate the quality of the generated shapes from the rendering perspective. We also use Strict-IoU and Loose-IoU to quantitatively evaluate the structure consistency of the generated shapes. For the chair category, we use the test set from DECOLLAGE. For building and cake categories, we randomly augment voxels to create the test set.

Generating a single shape using CLAY's commercial product, Rodin, takes approximately 5 to 10 minutes, while Coin3D requires about 25 minutes to optimize one shape, which makes both approaches impractical for large-scale automatic evaluation. Therefore, we randomly select 8 text prompts. For each text prompt, we randomly sample 10 coarse voxels from the corresponding test set to compute the metrics.

Render-FID. For each generated detailed shape, we uniformly render 4 views at 0, 90, 180, and 270 degrees. For each text prompt, we first augment it with view-dependent description by appending *front view, side view, back view* to the end of the prompt. For each augmented text prompt, We then use Stable Diffusion 2.1 with *stabilityai/stable-diffusion-2-1-base* to generate 4 different images. We compute the FID between the rendered images of the generated shapes and the images generated from Stable Diffusion.

CLIP score. For each generated detailed shape, we uniformly render 24 views by rotating the camera around the object with fixed poses. We then compute the cosine similarity between the CLIP embeddings of the input text prompt and the rendered images of the generated shape. We average the score over all the views. We compute the CLIP score using *openai/clip-vit-large-patch14* version of CLIP model.

$$\text{CLIPScore} = max(100 \cdot cos(E_1, E_2), 0) \tag{4}$$

Strict-IoU. Given the density field of the generated shape v_d and the corresponding input coarse voxel grid v, we first voxelize the density field using a threshold value of 30. We then downsample it to the same resolution as the input coarse voxel grid, which we define as v'. We compute the Strict-IoU as follows:

$$\text{Strict-IoU} = \frac{||v \ \& \ v'||_1}{||v \mid v'||_1} \tag{5}$$

Loose-IoU. Loose-IoU is the relaxed version of Strict-IoU and we define it as follows:

$$\text{Loose-IoU} = \frac{||v \ \& \ v'||_1}{||v||_1} \tag{6}$$

D COIN3D IMPLEMENTATION

We use the teddy bear example included in the officially released code to demonstrate that Coin3D can work on simple structures. Figure 15 shows the intermediate results and the final output generated by the officially released code. We also show the intermediate results and the final output of Coin3D when applied to the complex structure in Figure 16. Coin3D performs well on simple structures but struggles to handle more complex ones.

E SHADDR IMPLEMENTATION

Since our method generates the detailed shape is in radiance field representation, for a fair comparison, we modify one of the ablation settings in ShaDDR to also produce a radiance field, ensuring a fair comparison. More specifically, we modify the supervision of ShaDDR's texture branch from a 2D discriminator to a 3D discriminator, which can directly discriminate the output albedo field against the albedo field of the style shape. We provide the detailed shapes generated by our method, which are used as styles for training ShaDDR in Figure 17.

F ADDITIONAL QUANTITATIVE COMPARISONS

Table 3 to 10 show the quantitative comparisons of randomly selected text prompts in our experiments.

G ADDITIONAL QUANTITATIVE ABLATIONS

Table 11 to 18 show the quantitative ablation of using regularization loss with different λ_{reg} on randomly selected text prompts.

H ADDITIONAL QUALITATIVE RESULTS

Figure 18 shows additional results of cross-category detailization.

Table 3. Quantitative comparison of structural guided geneartion on text prompt *"a farmhouse chair with a cross-back design and a natural wood finish"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	218.634	17.462	0.502	0.633
Coin3D	202.395	20.413	0.479	0.628
CLAY	178.604	23.634	**0.546**	0.647
Ours	**176.069**	**24.758**	0.510	**0.651**

Table 4. Quantitative comparison of structural guided geneartion on text prompt *"a Queen Anne chair with a leather back in a light, neutral tone and a cushioned seat"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	177.382	25.834	0.572	0.619
Coin3D	218.345	19.371	0.617	0.694
CLAY	185.332	24.182	0.673	**0.761**
Ours	**163.286**	**28.737**	**0.682**	0.751

Table 5. Quantitative comparison of structural guided geneartion on text prompt *"a metal chair with a leather back and a cushioned seat"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	194.472	20.273	0.512	0.601
Coin3D	191.389	20.812	0.496	0.598
CLAY	177.283	23.226	0.524	0.618
Ours	**173.923**	**23.239**	**0.578**	**0.682**

Table 6. Quantitative comparison of structural guided geneartion on text prompt *"a traditional Japanese palace with tiled roofs and wooden walls"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	221.312	20.283	0.671	0.769
Coin3D	218.126	21.283	0.643	0.737
CLAY	172.284	25.283	**0.684**	**0.792**
Ours	**165.835**	**27.264**	0.671	0.763

Table 7. Quantitative comparison of structural guided geneartion on text prompt *"a Gothic church with spires, detailed carvings, and flying buttresses"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	206.925	18.275	**0.753**	**0.815**
Coin3D	192.268	20.178	0.639	0.751
CLAY	**170.237**	23.825	0.692	0.763
Ours	171.235	**25.836**	0.729	0.797

Table 8. Quantitative comparison of structural guided geneartion on text prompt *"a Russian cathedral with domes and tall spires"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	211.935	17.237	**0.733**	**0.827**
Coin3D	194.126	19.175	0.621	0.756
CLAY	**170.003**	21.528	0.711	0.819
Ours	172.946	**23.730**	0.712	0.803

Table 9. Quantitative comparison of structural guided geneartion on text prompt *"a cake with chocolate dripping down the sides"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	201.274	16.238	0.651	0.779
Coin3D	197.271	18.274	0.529	0.636
CLAY	**172.194**	24.482	0.612	0.728
Ours	178.836	**24.624**	**0.658**	**0.785**

Table 10. Quantitative comparison of structural guided geneartion on text prompt *"a chair with a backrest shaped like a large maple leaf"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
ShaDDR	182.307	25.437	0.566	0.673
Coin3D	193.561	22.491	0.510	0.627
CLAY	177.284	25.381	0.564	0.668
Ours	**171.491**	**30.184**	**0.571**	**0.682**

Table 11. Ablation results on text prompt *"a farmhouse chair with a cross-back design and a natural wood finish"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	176.304	23.634	0.483	0.627
$\lambda_{mask} = 10^2$	177.936	24.237	0.489	0.631
$\lambda_{mask} = 10^3$	176.382	23.532	0.507	0.648
$\lambda_{mask} = 10^4$	**176.069**	**24.758**	**0.510**	**0.651**

Table 12. Ablation results on text prompt *"a Queen Anne chair with a leather back in a light, neutral tone and a cushioned seat"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	**163.022**	28.281	0.652	0.729
$\lambda_{mask} = 10^2$	163.381	27.769	0.658	0.730
$\lambda_{mask} = 10^3$	163.836	28.172	0.676	0.747
$\lambda_{mask} = 10^4$	163.286	**28.737**	**0.682**	**0.751**

Table 13. Ablation results on text prompt *"a metal chair with a leather back and a cushioned seat"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	174.197	**23.625**	0.518	0.633
$\lambda_{mask} = 10^2$	173.823	23.116	0.522	0.637
$\lambda_{mask} = 10^3$	**173.468**	23.361	0.569	0.663
$\lambda_{mask} = 10^4$	173.923	23.239	**0.578**	**0.682**

Table 14. Ablation results on text prompt *"a traditional Japanese palace with tiled roofs and wooden walls"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	165.963	26.927	0.611	0.706
$\lambda_{mask} = 10^2$	165.782	26.694	0.609	0.694
$\lambda_{mask} = 10^3$	**165.196**	26.379	0.652	0.757
$\lambda_{mask} = 10^4$	165.835	**27.264**	**0.671**	**0.763**

Table 15. Ablation results on text prompt *"a Gothic church with spires, detailed carvings, and flying buttresses"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	171.397	**25.913**	0.658	0.746
$\lambda_{mask} = 10^2$	**170.731**	25.612	0.633	0.739
$\lambda_{mask} = 10^3$	171.169	25.081	0.689	0.784
$\lambda_{mask} = 10^4$	171.235	25.836	**0.729**	**0.797**

Table 16. Ablation results on text prompt *"a Russian cathedral with domes and tall spires"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	**171.291**	23.172	0.612	0.728
$\lambda_{mask} = 10^2$	172.735	23.561	0.633	0.747
$\lambda_{mask} = 10^3$	173.013	23.597	0.684	0.793
$\lambda_{mask} = 10^4$	172.946	**23.730**	**0.712**	**0.803**

Table 17. Ablation results on text prompt *"a cake with chocolate dripping down the sides"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	**177.304**	23.812	0.609	0.716
$\lambda_{mask} = 10^2$	177.782	24.018	0.603	0.701
$\lambda_{mask} = 10^3$	178.423	24.184	0.643	0.769
$\lambda_{mask} = 10^4$	178.836	**24.624**	**0.658**	**0.785**

Table 18. Ablation results on text prompt *"a chair with a backrest shaped like a large maple leaf"*.

	Render-FID ↓	CLIP score ↑	Strict-IoU ↑	Loose-IoU ↑
$\lambda_{mask} = 0$	171.622	**30.383**	0.533	0.659
$\lambda_{mask} = 10^2$	171.293	29.962	0.526	0.642
$\lambda_{mask} = 10^3$	**170.932**	29.075	0.566	0.663
$\lambda_{mask} = 10^4$	171.491	30.184	**0.571**	**0.682**

(a) Rendering of the input coarse proxy

(b) Single-view image

(c) Multi-view images

(d) Rendering of NeuS reconstruction

Fig. 15. Coin3D reproduction experiment with teddy bear coarse proxy and text prompt *"a lovely teddy bear"*. We show (a) the rendering of the input coarse proxy, (b) the single-view image generated by 2D ControlNet conditioned on the text prompt *a lovely teddy bear*, (c) the generated multi-view images and (d) the rendered images of NeuS reconstruction.

(a) Rendering of the input coarse proxy

(b) Single-view image

(c) Multi-view images

(d) Rendering of NeuS reconstruction

Fig. 16. Coin3D applied to a complex chair structure with text prompt *"a chair with a backrest shaped like a large maple leaf"*. We show (a) the rendering of the input coarse proxy, (b) the single-view image generated by 2D ControlNet conditioned on the text prompt *a lovely teddy bear*, (c) the generated multi-view images and (d) the rendered images of NeuS reconstruction.

Fig. 17. We show the detailed shapes generated by our method, which are used as styles shapes for training ShaDDR. These detailed shapes are of good quality and suitable for training ShaDDR.

Fig. 18. Additional results of cross-category detailization. We show a collection of coarse voxels from the chair, table, couch, and bed classes on the top and the detailed shapes on the bottom. Our method can generate structurally varying shapes spanning multiple furniture categories with a consistent style.