

DECOLLAGE: 3D Detailization by Controllable, Localized, and Learned Geometry Enhancement

Qimin Chen^{1,2}, Zhiqin Chen², Vladimir G. Kim², Noam Aigerman³,
Hao Zhang^{1,4}, and Siddhartha Chaudhuri²

¹ Simon Fraser University

² Adobe Research

³ University of Montreal

⁴ Amazon

Abstract. We present a 3D modeling method which enables end-users to refine or *detailize* 3D shapes using machine learning, expanding the capabilities of AI-assisted 3D content creation. Given a coarse voxel shape (e.g., one produced with a simple box extrusion tool or via generative modeling), a user can directly “paint” desired target styles representing compelling geometric details, from input exemplar shapes, over different regions of the coarse shape. These regions are then up-sampled into high-resolution geometries which adhere with the painted styles. To achieve such controllable and localized 3D detailization, we build on top of a Pyramid GAN by making it *masking-aware*. We devise novel structural losses and priors to ensure that our method preserves both desired coarse structures and fine-grained features even if the painted styles are borrowed from diverse sources, e.g., different semantic parts and even different shape categories. Through extensive experiments, we show that our ability to localize details enables novel interactive creative workflows and applications. Our experiments further demonstrate that in comparison to prior techniques built on global detailization, our method generates structure-preserving, high-resolution stylized geometries with more coherent shape details and style transitions.

Keywords: 3D detailization · Controllable 3D generation · Generative adversarial network · High-resolution geometry

1 Introduction

Customized 3D content is becoming more widely available, driven by rapid advances in generative AI and increasing demand from computer games, AR/VR, and e-commerce. Recently, deep generative models based on diffusion and vision-language models have made significant waves in improving the accessibility (e.g., via text prompting) and ingenuity of generated content, as well as enabling zero-shot learning. However, while effective at creating coarse content, the latest methods along these fronts, e.g., [15, 28, 41, 47], still lack the ability to generate and precisely control high-quality *geometric details*. Also, their slow speed remains a roadblock to integrating them into artists’ conventional workflows.

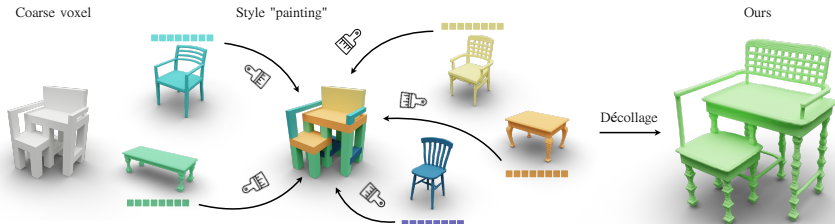


Fig. 1: D collage is an art form created by “cutting/removing pieces of an original image”¹. When “painting” a style exemplar with geometric details over a region of a coarse shape, coarse surfaces are removed to unveil a *detailed* version to mimic the exemplar. We show an out-of-distribution chair-like shape detailedized via *style mixing*, where five exemplars “d collage” the coarse voxels.

In this paper, we propose a learning-based method that enables novice users to add geometric details to a coarse 3D shape by selecting regions on it and assigning them the *styles* of exemplar shapes with compelling geometric details. Each region is upsampled and *detailed* by a neural network to replicate the corresponding exemplar’s detail style, while preserving the overall structure of the coarse input *content* shape. In general, when detailizing multiple regions using separate and possibly diverse style exemplars, i.e., “style mixing,” as shown in Figure 1, the goal of our region-specific, *localized* 3D detailization is to produce structure-preserving and globally coherent results in terms of shape details and part connections, across feature scales and shape categories. Style mixing from different shape categories offers additional design freedom which can boost the creative potential of the generated shapes without compromising structural validity and functionality of the input content shapes, as shown in Figure 1.

Enabling localized style control via exemplar shapes is a natural content creation paradigm, which addresses both the questions of *which* detail to generate and *where* to generate. However, the problem is technically challenging, especially with style mixing. Even when both the content and style shapes happen to be semantically segmented, decoupled assignment of details to target areas inevitably leads to structural inconsistencies, especially over joint regions, as the details may not trivially mix. Besides requiring special treatment to ensure coherent part connections, the network also needs to have a global understanding of the whole shape while applying detail locally. Prior methods for conditional detailization [5, 6] are designed to deal with a single style exemplar and do not perform well when there is significant structural dissimilarity to the content shape. In addition, many style configurations may have never been observed during training, leading to out-of-distribution failures: see Figure 2.

To address the above challenges, our method leverages a *hierarchical* backbone architecture for generative adversarial learning, i.e., a Pyramid GAN [12, 23, 49, 51]. This enables our network to capture both global structures using coarse-level reasoning, as well as local geometric details at finer levels. Accordingly, our

¹ <https://en.wikipedia.org/wiki/Decollage>

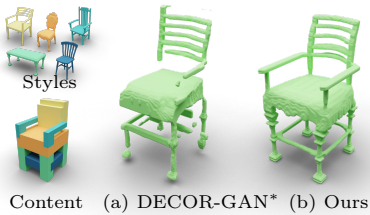


Fig. 2: DECOR-GAN* [6] (a) with naïve local controllability generates disconnected structures and floating pieces. Our method DECOLLAGE (b) fares much better in preserving global structure and generating local geometric details.

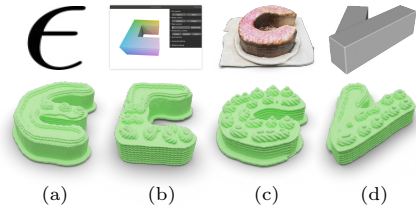


Fig. 3: Application: detailing shapes from various sources including (a) extruding 2D profiles; (b) coarse voxels created via an interactive user interface (see supplementary); (c) shapes generated by text-to-3D model; (d) simple CAD primitives.

network is trained with both a global discriminator and a local, style-conditioned one, where we employ an *adaptive α weighting* to adjust the importance of the discriminators depending on proximity to style transition regions. In addition, we propose novel network losses to encourage structure preservation during up-sampling. Specifically, we ensure that the coarse shape is preserved under resampling at different resolutions by both downsampling and upsampling it. Finally, since local style control requires the coarse targets to be partitioned into regions to guide the generation, we propose several data augmentation mechanisms to generate segmented coarse targets from detailed sources during self-supervised training, by randomly changing their part structures, scales, and orientations.

In summary, our work offers the first method for *interactive, controllable, and localized* geometry detail generation, unlike existing shape detailization works [5, 6] which only offer global style control. On the technical front, the adaptive α weighting between discriminators has not been used in prior works, and is essential to our interactive workflow. Although Pyramid GAN has been heavily explored before, it alone is insufficient to tackle the challenge of style mixing. To this end, our key contribution is the use of novel structure-preserving losses tailored for the Pyramid architecture with discriminator adaptivity to overcome incoherent structures. Once trained, our network enables novel *interactive* 3D modeling, allowing both structure editing by end-users and automated per-region detailization, as shown in Figure 3. This application also showcases the versatility of our modeling paradigm through DECOLLAGE, in terms of 3D content shapes.

We conduct experiments to show that our approach performs significantly better than relevant baselines on 3D detailization by borrowing details across different categories of shapes. We further demonstrate that our method outperforms prior works on tasks they were designed to handle, i.e., a standard example-based detailization with a *single* style, from the same category [6]. Lastly, we showcase applications of our method to enable various workflows, such as creating detailed shapes from coarse labeled blocks, and detailizing coarse generated shapes by painting style labels.

2 Related work

While our approach might appear to tackle a similar problem to 3D voxel up-sampling [7, 9, 10, 40, 42, 45], however, a critical distinction is that it aims to generate new features and details. We thus review prior 3D generative models and shape detailization techniques.

3D generative models. Various 3D generative models have been introduced for point clouds [2, 53], voxels [11, 50], neural implicit functions [8, 34, 39], neural radiance fields [28, 36, 41], and hybrid representations [15, 19, 26]. These approaches are predominantly empowered by variational autoencoders (VAEs) [25], generative adversarial networks (GANs) [16], or diffusion probabilistic models [18, 46].

Despite significant progress in this field, very few works offer controllable and interactive 3D shape generation for modeling applications. Notably, Point-E [38], Shap-E [21], and One-2-3-45 [31] can generate a 3D model from text or single image inputs, with processing times ranging from 30 seconds to over a minute. DECOR-GAN [6] and ShaDDR [5] can efficiently produce a detailed 3D shape from coarse voxel inputs, taking less than 1 to 2 seconds. Although ShaDDR provides interactivity during generation, it only offers global style control. Our work builds upon DECOR-GAN to deliver the first *controllable* and *localized* interactive modeling experience, while introducing new formulations and technical contributions that enable local style control and improve robustness in handling coarser and out-of-distribution inputs.

3D shape detailization. Apart from classic methods that apply displacement maps or volumetric textures [22, 37] on the surfaces to represent geometric details, recent methods have been proposed to perform local geometric operations on a coarse mesh to synthesize surface details. Berkiten et al. [3] employs metric learning to transfer displacement maps from a high-quality 3D mesh to a coarse mesh. Hertz et al. [17] learns geometric texture from a single reference 3D mesh and is able to apply the learned texture to a new shape. Neural Subdivision [29] also learns local geometric features and is able to transfer them via mesh subdivision. Leveraging differentiable rendering of meshes, Paparazzi [30] and Text2Mesh [35] can generate geometric details on the mesh surface conditioned on the style of a reference image or input text, respectively. 3DStyleNet [52] transfers geometric and texture styles from one shape to another. However, none of the aforementioned methods is capable of altering the topology of the coarse mesh, therefore restricting the range of geometric styles they can synthesize.

Other methods aim to synthesize geometric details by replicating local patches from a reference shape, thereby overcoming the topology constraint. Inspired by image quilting [13], mesh quilting [55] can detailize the surface of a coarse mesh by copying, deforming, and stitching local patches of a given geometric texture patch. SketchPatch [14] adopts a PatchGAN [20] discriminator to mimic the local style of a reference image in order to stylize plain solid-lined sketches. DECOR-GAN [6] similarly utilizes PatchGAN for generating detailed voxel shapes from input coarse voxels, with the geometric style of the generated shape copied from

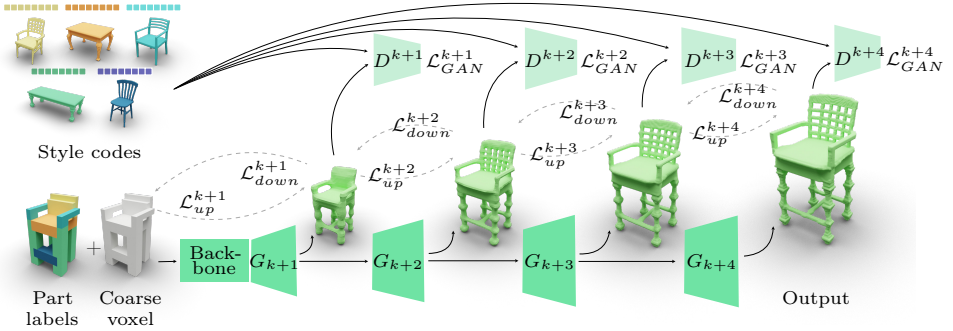


Fig. 4: Network architecture. Conditioned on a set of style codes associated with each segmented part, the network upsamples the coarse content voxel with part labels into detailed geometries in multiple resolutions. For each upsampling level j , the discriminator enforces the local patches of each part in the upsampled geometry to be plausible with respect to the styles they are conditioned on. The structure-preserving losses \mathcal{L}_{down}^j and \mathcal{L}_{up}^j enforce the structure of the output to be consistent with the input.

one detailed reference voxel model. ShaDDR [5] improves the generated geometry of [6] by leveraging a 2-level hierarchical GAN, and introduces texture generation. DMTET [44] can detailize a coarse voxel shape into a detailed mesh through differentiable marching tetrahedra. While the ability to generate arbitrary topology is desirable, it also comes with drawbacks. For instance, DECOR-GAN can synthesize impressive patterns, yet it is prone to producing disconnected parts and redundant floating pieces; see Figure 2. Our method effectively addresses this issue via novel structure-preserving losses and adaptive α weighting of style and global discriminators.

Hierarchical GANs. The pyramid structure of our network architecture draws inspiration from hierarchical GANs [12, 24, 48, 54], a structure widely employed in various tasks, including image [43] and 3D [23, 49, 51] generation. Multi-level generation based on different scales has been applied outside the scope of GANs to facilitate generation of structurally diverse shapes [27]. We apply hierarchical GAN and devise novel loss functions to generate shapes that are both globally plausible and locally detailed.

3 Method

Our method requires only a few (e.g., 16) detailed “style” shapes $\mathcal{S} = \{s_1, \dots, s_N\}$ with varying geometric styles as training data, where each style shape s_i is a $2^K \times 2^K \times 2^K$ high-resolution voxel grid. We also assume all the shapes are co-segmented into meaningful parts, so that each voxel $v \in s_i$ is associated with a part label $P(v)$. After training, given a $2^k \times 2^k \times 2^k$ coarse voxel grid c where each voxel $v \in c$ is associated with a part label $P(v)$ and a style $S(v) \in \{1, \dots, N\}$, our model can generate a high-resolution, detailed shape that has the overall structure of the input c , while the local geometric detail in the region corresponded

to a input voxel v follows the style exhibited in $s_{S(v)}$. In our experiments, we use $k = 4$ and $K = 8$.

In this section, we first introduce our network architecture - a pyramid GAN with $(K - k)$ levels, in Sec. 3.1. Next, in Sec. 3.2, we apply masked adversarial losses to ensure that the generated geometry follows the styles specified by the input voxels, while an adaptive α weighting scheme is developed to promote proper connectivity in style transition regions. We also devise two novel loss functions tailored for our pyramidal network architecture to preserve the structure of the input coarse voxels. Our method requires coarse voxels c with diverse structures and part segmentation during training. Therefore, in Sec. 3.3, we propose a data augmentation technique to take full advantage of the N segmented style shapes in \mathcal{S} and use them to synthesis the coarse voxels for our training.

3.1 Network architecture

Our network is shown in Figure 4. For each style shape s_i , we associate it with an optimizable 8-dimensional latent code z_i to represent its geometric style; see Figure 4 top-left. The input to our model is a coarse voxel grid c of resolution 2^k , where each voxel v contains its binary occupancy $O(v)$, part label $P(v)$ and a latent code $z_{S(v)}$ corresponding to its designated style $S(v)$; see in Figure 4 bottom-left. Our model contains a backbone and a pyramid of generator networks G^j that upsample the input into occupancy voxels of size 2^j , where $j \in \{k + 1, \dots, K\}$. Each G^j doubles the size of its input; see Figure 4 bottom. Both the backbone and the generators are 3D convolutional neural networks (CNNs).

Correspondingly, a pyramid of 3D CNN PatchGAN [20] discriminators D^j are employed to ensure that the shape generated at each resolution level is plausible; see Figure 4 top-right. Each D^j inputs a generated occupancy grid at 2^j resolution and outputs a voxel grid of the same resolution, while each output voxel has $N + 1$ channels. The first N channels of an voxel v in the discriminator output represent the likelihood that the local patch covered by v 's receptive field is from one of the N style shapes, thus the first N channels are *style-specific* discriminators. The $(N + 1)$ -th channel can be viewed as a *global* discriminator that evaluates the likelihood of the patch being plausible for a 3D shape. We denote the style-specific discriminators as D_i^j for level j and style i , and the global discriminator as D_*^j .

A main advantage of pyramid GANs is that we can have different receptive fields at different levels, so that the coarse levels only focus on generating coherent structures, while the fine levels pay more attention to generating plausible details. This provides better generalizability to inputs with drastically different structures compared to the training style shapes. We set the discriminator receptive fields to 7^3 and 9^3 for the first two levels and 18^3 for the rest.

3.2 Loss functions

Reconstruction loss. Given a high-resolution style shape s_i , we downsample it to a lower resolution and use it as an input coarse shape. We can directly apply a

reconstruction loss as we have the ground truth (GT). Denote the downsampled shape at resolution 2^j as s_i^j , for each style i and resolution level j , we have

$$\mathcal{L}_{recon} = \mathbb{E}_v(G^j(s_i^k)[v] - O(s_i^j[v]))^2, \quad (1)$$

where v iterates the indices of all voxels, $G^j(\cdot)$ is the output voxel grid of the generator, and $s[v]$ queries the voxel in s at index v . Note that the style $S(s_i^k[v]) = i$.

Adversarial loss. We do not have the GT detailed shape for an arbitrary coarse shape c . Thus we resort to the discriminators to supervise shape generation. In addition, the geometric style of the generated shape should respect the designated styles $S(v)$ in the input voxels. Hence, we have the following adversarial loss to train the generators, which is a masked version of the LSGAN [33] loss,

$$\mathcal{L}_{GAN} = \mathbb{E}_v((D_*^j(G^j(c))[v] - 1)^2 + \alpha \cdot (D_{S(c[v])}^j(G^j(c))[v] - 1)^2), \quad (2)$$

where $D^j(\cdot)$ is the output voxel grid of the discriminator, and α is a parameter to control the influence of the style-specific discriminators. Losses to train the discriminators can be found in the supplementary.

Adaptive α weighting. Setting α to a larger value can make the generated shape more stylistic in the region near v with respect to the style $S(c[v])$, yet it can make the region less plausible and less coherent with other regions, since the influence of the global discriminator has been tuned down. On the other hand, setting α to a small value lets the generator generate structurally coherent but style-less shapes. Our observation is that the regions near the transition boundary where two parts of different styles meet are the most problematic, since the geometries in these regions are unlikely to be observed in our training examples with single styles. Therefore, we develop a novel strategy to set α adaptively for each voxel: if a voxel is near a transition boundary (within 2 voxels), we will set a small α for that voxel, e.g., $\alpha_1 = 0.1$; otherwise, we set a larger α , e.g., $\alpha_2 = 0.5$.

Structure-preserving losses. Finally, to make the generated shape respect the structures presented in the input coarse voxels, we propose two novel structure-preserving losses. The downsampling loss ensures that if we downsample the generated shape at level $j + 1$ from resolution 2^{j+1} to 2^j , the result agrees with the generated shape at level j , or the input shape if $j = k$.

$$\mathcal{L}_{down} = \|\phi_{\downarrow}(G^{j+1}(c)) - \mathcal{X}(G^j(c))\|_2^2, \quad (3)$$

where ϕ_{\downarrow} is the max-pooling operator that downsamples the input by a factor of 2; \mathcal{X} is the stop-gradient operator to prevent the generated shape in level $j + 1$ from negatively affecting the generated shape in level j . Similarly, we have an upsampling loss to ensure the upsampled result of level j agrees with the shape at level $j + 1$.

$$\mathcal{L}_{up} = \|G^{j+1}(c) - \mathcal{X}(\phi_{\uparrow}(G^j(c)))\|_2^2, \quad (4)$$

where ϕ_{\uparrow} upsamples the input by a factor of 2 via nearest neighbor.

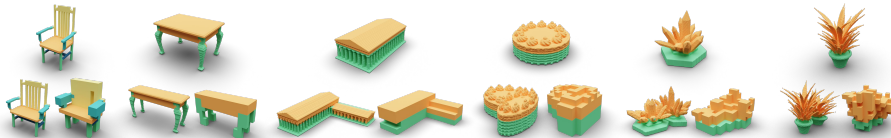


Fig. 5: Augmentation examples of different categories. For each category, we show the original style shape in the first row, the corresponding augmented style shape in the second row left, and downsampled as coarse shapes for training in the second row right.

The final loss is a sum of the above loss terms:

$$\mathcal{L} = \mathcal{L}_{recon} + \mathcal{L}_{GAN} + \gamma_1 \cdot \mathcal{L}_{down} + \gamma_2 \cdot \mathcal{L}_{up}. \quad (5)$$

We set $\gamma_1 = \gamma_2 = 10$ in our experiments.

3.3 Data augmentation

From few detailed and segmented style shapes, we synthesize an arbitrary number of coarse voxels with diverse structures and part segmentation for training, as shown in Figure 5. To create a coarse voxel, we choose a random style shape s_i and randomly scale it in x, y, and z directions. For some categories such as plant, we further perform random rotation and combine multiple shapes to have more geometric variations. We downsample the augmented shapes into 2^k resolution as coarse shapes for training. Note that all the style shapes are co-segmented, therefore we also obtain segmentation in the resulting coarse voxels. At this point, we can randomly assign different styles to different segmented parts.

3.4 Implementation details

Similar to DECOR-GAN [6], we apply a Gaussian filter with $\sigma = 1$ on each style shape to convert its binary occupancy voxels into a smoother and more continuous scalar grid for the sake of better optimization. Training our model takes about 30 hours on a single NVIDIA 3090Ti GPU for $k = 4$ and $K = 8$. After training, generating a detailed shape only takes less than a second. We extract the mesh surfaces using Marching Cubes [32].

4 Experiments

In this section, we first evaluate our proposed method in single-category single-style shape detailization and compare with other detailization methods in Sec. 4.1. Next, we demonstrate that our method can generate novel 3D shapes with better localized style control in multi-category style mixing in Sec. 4.2, and perform ablation study in Sec. 4.3. We also show several applications including interactive editing in Sec. 4.4.

Datasets. We conduct experiments on six shape categories: 16 chairs, 16 tables, and 5 plants from ShapeNet [4]; and 5 buildings, 3 cakes, and 3 crystals from 3D Warehouse [1] under CC-BY 4.0. For each style shape, we obtain binary occupancy voxels and manually annotate the part labels for training. We segment each chair into five parts: back, seat, armrest, leg, and stretcher. Tables are labeled into two parts: tabletop and legs. A plant is labeled into pot and leaf. A building is labeled into the roof and main body. Cakes and crystals are segmented into bottom and top parts. All the training style shapes can be found in the supplementary. While increasing the number of style shapes is possible, we aim to demonstrate that our proposed method has *robust* generalizability to coarse voxels with drastically different and complex structures, even when trained on a very limited number of style shapes.

Evaluation metrics. For quantitative evaluation, we follow DECOR-GAN [6] and adopt the following metrics. *Strict-IOU* is to measure the Intersection over Union (IOU) between the downsampled output voxels and the input voxels, to evaluate how well the generated shape respects the structures in the input shape. *Loose-IOU* is a relaxed version of Strict-IOU to compute the proportion of occupied voxels in the input that are also occupied in the downsampled output. *LP-IOU* and *LP-F-score* are to measure the percentage of local patches in the generated shape that are “similar” (according to IOU or F-score) to at least one local patch in the style shapes. Higher LP-IOU and LP-F-score indicate that the local details of the generated shapes are more similar to the local details of the style shapes, thus the generated shapes are deemed to be more locally plausible. *Cls-score* is to evaluate the overall plausibility of the generated shapes by training a classifier network to distinguish between the rendered images of the generated shapes and those of the real shapes and recording the mean classification accuracy. More details can be found in the supplementary.

4.1 Single-category detailization

We first qualitatively and quantitatively compare our method with DECOR-GAN [6] and ShaDDR’s geometry generator [5] on single-category single-style shape detailization. We use the data in DECOR-GAN and train individual models for different categories for fair comparisons. We report the results on chair category due to page limit; other categories can be found in the supplementary.

We show quantitative comparison in Table 1 and qualitative results in Figure 6. DECOR-GAN and ShaDDR are likely to generate disconnected parts, especially in the joint regions, e.g., where armrests meet seats or backs. Their results in Figure 6 frequently show fragmented or disconnected parts. In contrast, our method can produce significantly higher-quality upsampled geometry with better connectivity. Moreover, our generated shapes better preserve the structures in the input voxels. An example is indicated by arrows in Figure 6, where the armrest of our generated chair closely follows the shape of its coarse content voxels, while the results of other methods fail to follow. This is also reflected by



Fig. 6: Single-category detailization on chair category. We show the input content shapes on the left and style shapes on top. Please zoom in to observe the details.

our higher Strict- and Loose-IOU in Table 1. Our method also generates better local details, as reflected by higher LP-IOU and LP-F-score.

Table 1: Quantitative results of single category detailization on the chair category.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-IOU \uparrow	LP-F-score \uparrow	Cls-score \downarrow
DECOR-GAN	0.581	0.753	0.517	0.906	0.533
ShaDDR	0.596	0.760	0.563	0.907	0.527
Ours (Pyramid full)	0.748	0.908	0.591	0.914	0.506

Table 2: Quantitative results of multi-category detailization on chair and table.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-IOU \uparrow	LP-F-score \uparrow	Cls-score \downarrow
DECOR-GAN*	0.609	0.839	0.509	0.961	0.567
ShaDDR*	0.611	0.854	0.496	0.933	0.549
Ours <i>w/o</i> part labels	0.747	0.900	0.508	0.969	0.533
Ours <i>w</i> part labels	0.750	0.906	0.513	0.977	0.518

4.2 Multi-category style mixing

We train one single model with $k = 4$ and $K = 8$ for the chair and table categories and another with $k = 4$ and $K = 8$ for the plant, building, cake and crystal categories. Since the original DECOR-GAN and ShaDDR are not able to perform style mixing during geometry detailization, we modify the training procedure of DECOR-GAN and ShaDDR by assigning different styles to different regions of the input coarse voxels for fair comparisons. We denote these two baselines as DECOR-GAN* and ShaDDR*.

We show the style mixing results in Figure 7 and 8. DECOR-GAN* and ShaDDR* fail to produce coherent and connected structures for out-of-distribution content shapes with different style combinations. Our method not only demonstrates improved generalization to novel coarse content shapes, but also produces better geometric details with smooth style transition. Our method also outperforms other methods quantitatively, as shown in Table 2. More qualitative results can be found in the supplementary material.

4.3 Ablation study

In this section, we validate the effectiveness of our pyramid GAN structure and structure-preserving losses. In Table 3 and Figure 9, we compare several vari-

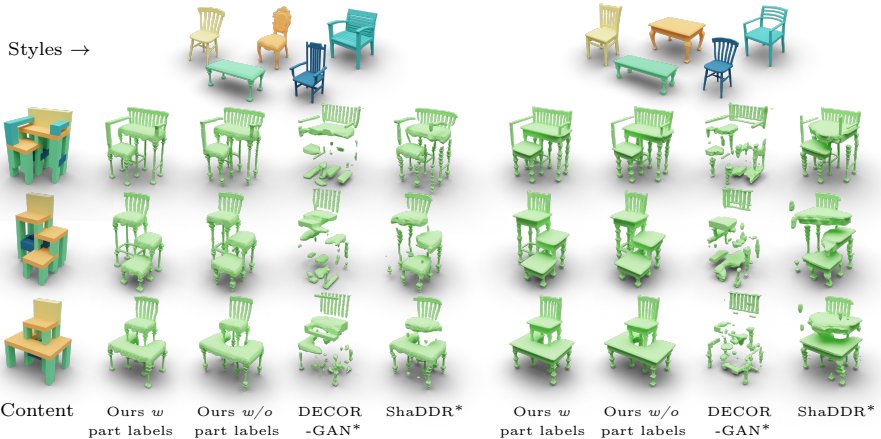


Fig. 7: Multi-category style mixing results on chair and table categories. We show the input coarse voxels with style labels on the left. The corresponding style shapes for the colored style labels are shown on top. Please zoom in to observe the details.

ations of our proposed method in single-category single-style setting. (1) *Pyramid vanilla*, in which we only use a pyramid GAN structure without structure-preserving losses, i.e. $\gamma_1 = \gamma_2 = 0$ in Equation 5. In this setting, we adopt DECOR-GAN’s design for preserving the content structure, which are non-differentiable masks applied to the generated voxels to cut off all the voxels outside the valid region defined by the coarse input voxels. We observe connectivity issues as well as floating pieces in the generated shape, as shown in Figure 9 (a). (2) *Pyramid w \mathcal{L}_{up}* , in which we remove the masks and add the upsampling loss to Pyramid vanilla, i.e., $\gamma_1 = 0, \gamma_2 = 10$. The upsampling loss can better preserve the overall structure, while the floating pieces remain, as shown in Figure 9 (b). (3) *Pyramid w \mathcal{L}_{down}* , in which we remove the masks and add the downsampling loss to Pyramid vanilla, i.e., $\gamma_1 = 10, \gamma_2 = 0$. The downsampling loss can also help preserve the overall structure, and it effectively eliminates the floating pieces. Yet it tends to miss thin structures, as shown in Figure 9 (c). (4) Our proposed method, *Pyramid full*, where $\gamma_1 = \gamma_2 = 10.0$. In this setting, the generated shapes exhibit better connectivity and better adherence to the global structure, as shown in Figure 9 (d). The quantitative results in Table 3 also show that our full model has the best performance.

We also perform a more thorough ablation study of each component in the multi-category style mixing setting, as shown in Table 4 and Figure 10 (left). Note that adaptive α is only used in the multi-category style mixing setting. By leveraging Pyramid architecture ((a) vs. (b)), the overall structure of the output is significantly improved. Both \mathcal{L}_{down} and \mathcal{L}_{up} further help *refine* the overall structure where \mathcal{L}_{down} can effectively eliminate the floating pieces and \mathcal{L}_{up} can improve thin structures ((c) vs. (d) and (g) vs. (h)). This conclusion is also consistent with the ablation study in the single-category detailization setting. The adaptive α weighting can effectively improve the boundary *transition* where

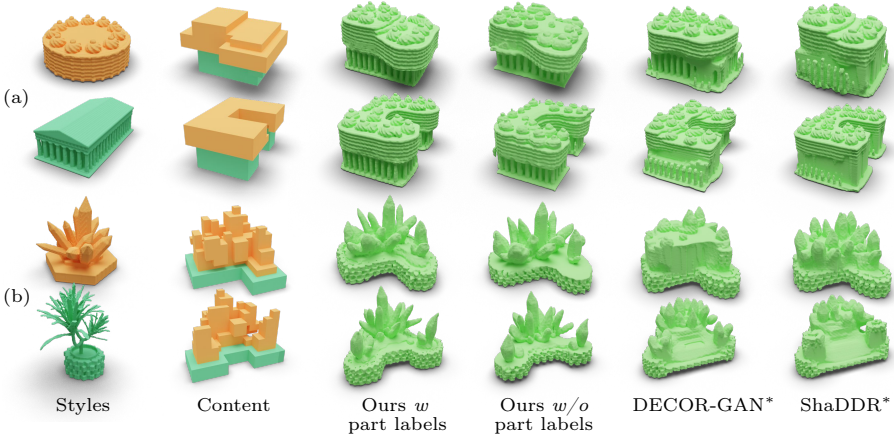


Fig. 8: Multi-category style mixing results on plant, building, cake, and crystal categories. For each group, e.g., (a), we show two style shapes in the first column, coarse input shapes with style labels in the second column, and results in the remaining columns. Please zoom in to observe the local geometric details.

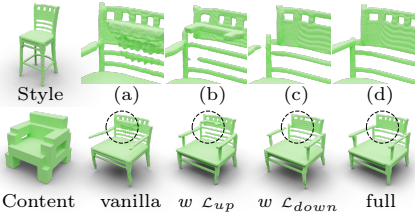


Fig. 9: Qualitative ablation study of the proposed structure-preserving losses in the single-category setting.

Table 3: Quantitative ablation study of the proposed structure-preserving losses in the single-category setting.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-score \uparrow	LP-F-score \uparrow	Cls-score \downarrow
(a). Pyramid vanilla	0.578	0.819	0.508	0.868	0.593
(b). Pyramid $w \mathcal{L}_{up}$	0.705	0.873	0.547	0.883	0.550
(c). Pyramid $w \mathcal{L}_{down}$	0.730	0.889	0.564	0.892	0.558
(d). Pyramid full	0.748	0.908	0.591	0.914	0.506

two different styles meet, e.g. the armrest is well-connected to the seat and the stretcher is well-connected to the leg ((c) vs. (g), (d) vs. (h) and (f) vs. (i)).

In addition, we perform an ablation study on the adaptive α weighting scheme described in Sec. 3.2 with different α values. Figure 10 (right) shows qualitative results of setting the parameter α_1 to 0.1, 0.3 and 0.5 for voxels near the transition boundary and α_2 to 0.5 for the rest regions. Setting $\alpha_1 = \alpha_2 = 0.5$ can be considered equivalent to not adaptively adjusting the α . By using a smaller α_1 , the region where two different styles meet has a smoother style transition, e.g. the armrest is well-connected to the seat in the last column of Figure 10 (right).

We further stress test our method by removing part labels from the input to the network. That is, the users need not specify which part of each exemplar should supply details: the network needs to automatically decide this based on the geometry of the selected region on the coarse shape. Therefore, no co-segmentation is needed in this setting, and we only use per-shape segmentation to assign styles to each local region of the coarse input shape during training.

Table 4: Quantitative ablation studies of Pyramid structures (P), structure-preserving losses (\mathcal{L}_{down} and \mathcal{L}_{up}) and adaptive α weighting (adp- α) in the multi-category chair and table style mixing setting.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-IOU \uparrow	LP-F-score \uparrow	Cls-score \downarrow
(a). P (\times) (DECOR-GAN*)	0.621	0.849	0.261	0.939	0.638
(b). P (\checkmark), \mathcal{L}_{down} (\times), \mathcal{L}_{up} (\times), adp- α (\times)	0.661	0.880	0.260	0.941	0.645
(c). P (\checkmark), \mathcal{L}_{down} (\checkmark), \mathcal{L}_{up} (\times), adp- α (\times)	0.713	0.894	0.274	0.949	0.586
(d). P (\checkmark), \mathcal{L}_{down} (\times), \mathcal{L}_{up} (\checkmark), adp- α (\times)	0.705	0.889	0.271	0.952	0.592
(e). P (\checkmark), \mathcal{L}_{down} (\times), \mathcal{L}_{up} (\times), adp- α (\checkmark)	0.688	0.882	0.263	0.955	0.568
(f). P (\checkmark), \mathcal{L}_{down} (\checkmark), \mathcal{L}_{up} (\checkmark), adp- α (\times)	0.753	0.905	0.281	0.963	0.533
(g). P (\checkmark), \mathcal{L}_{down} (\checkmark), \mathcal{L}_{up} (\times), adp- α (\checkmark)	0.721	0.904	0.279	0.960	0.529
(h). P (\checkmark), \mathcal{L}_{down} (\times), \mathcal{L}_{up} (\checkmark), adp- α (\checkmark)	0.724	0.898	0.277	0.952	0.547
(i). P (\checkmark), \mathcal{L}_{down} (\checkmark), \mathcal{L}_{up} (\checkmark), adp- α (\checkmark)	0.761	0.913	0.282	0.968	0.527

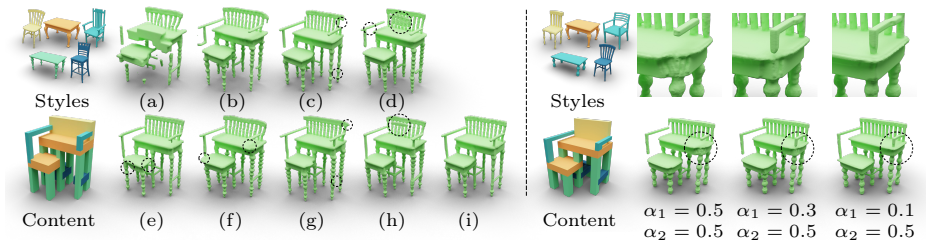


Fig. 10: (Left) Qualitative ablation studies of Pyramid structures (P), structure-preserving losses (\mathcal{L}_{down} and \mathcal{L}_{up}) and adaptive α weighting (adp- α) in the multi-category chair and table style mixing setting. The configurations of different models used in (a-i) can be found in Table 4. Zoom in to observe the details. **(Right)** Qualitative ablation study of different adaptive α values. The input coarse voxel and style shapes are shown on the left. The zoom-ins are shown on the top. α_1 is set for voxels near the transition boundary and α_2 for the rest.

Figure 11 visually compares inputs *with* vs. *without* part labels. Even without part labels, our method can generate reasonable results with only slightly worse local details, which may be attributed to the network leveraging certain capabilities to identify selected regions. This is also reflected by slightly lower LP-IOU and LP-F-score compared to input *with* part labels in Table 2.

4.4 Application

Our method can be applied to detailize shapes from various sources, as shown in Figure 3. (a) We can detailize coarse shapes that are easily obtainable by extruding 2D profiles, such as fonts. (b) We develop an interface for users to model coarse voxels and assign style labels interactively. Please see our supplementary video for a real-time, interactive demo. (c) We ran an off-the-shelf text-to-3D model to obtain a “C-shaped cake”. Then we remove the bottom plane of the generated shape and apply our method on its downsampled voxels to obtain a detailed cake. (d) We can also detailize shapes that are created using simple primitives such as two boxes. For each shape, we apply the styles of two cakes at different regions, which may not correspond to semantic parts. Note that we offer

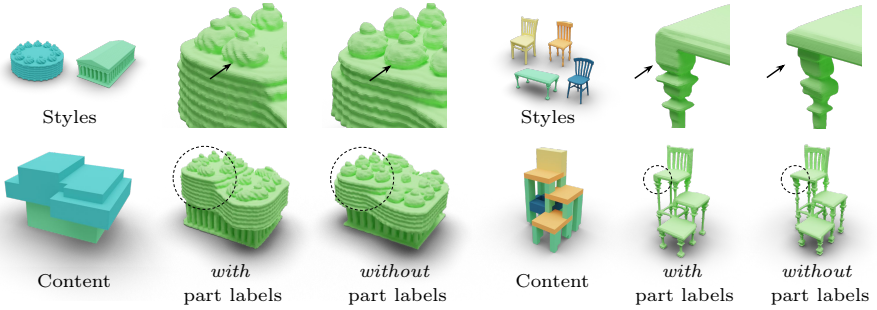


Fig. 11: Ablation study of input *with* vs. *without* part labels on building-cake and chair-table style mixing. By leveraging part labels as additional input, the network can generate more details with more natural connections between different regions.

the first method for interactive *controllable* and *localized* detail generation, unlike ShaDDR [5] which only offers global style control. This may enable creative modeling, such as the building and plant mixing in Figure 12 (a).

5 Conclusion, discussion, and future work

We present the first exemplar-based generative model for 3D detailization *which* offers local control, interactivity, and generalizability to out-of-distribution coarse structures. Our novel structure-preserving losses, along with the global discriminator and spatially adaptive style adjustment, lead to clear improvements over current detailization methods and enable coherent style transition even when mixing diverse exemplars.

To generate better local geometric details, we currently assume that a meaningful co-segmentation is available for all style shapes. This may limit the detailization to only coarse-level structures and also prevent style transfer which breaks the semantic barrier. Our method adopts the occupancy voxel representation for 3D shapes and relies on 3D CNNs to perform upsampling, which can limit the resolution of the final generated shapes. For example, our maximum output resolution is 256^3 , which may not be sufficient to represent finer geometric details, such as the tips of the crystals in Figure 8 (b). In our experiments, we found mixing styles for significantly different geometries, e.g., the failure case of mixing chair and plant in Figure 12 (b), can often lead to undesirable artifacts. This is due to the significant differences in the *local* structure of the coarse shape and the style shapes. For example, the chair back cannot be detailed into a plant because such a shape does not exist in either the content or the style shapes. Mixing these styles requires a deeper understanding of semantics and aesthetics.

As for future work, we would like to transfer non-homogeneous shape details onto non-homogenous coarse structures. The use of diffusion models for voxel

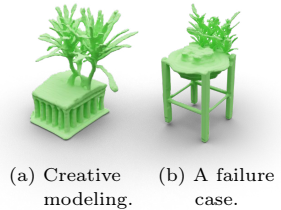


Fig. 12: Examples of creative modeling and a failure case.

upsampling and the integration with large language and vision-language models for geometry and texture detailization are both worth exploring.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. 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

1. 3d warehouse. <https://3dwarehouse.sketchup.com/>
2. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: ICLR. pp. 40–49 (2018)
3. Berkiten, S., Halber, M., Solomon, J., Ma, C., Li, H., Rusinkiewicz, S.: Learning detail transfer based on geometric features. In: Computer Graphics Forum. vol. 36, pp. 361–373. Wiley Online Library (2017)
4. Chang, A.X., Funkhouser, T., Guibas, L., Harrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: Shapenet: An information-rich 3d model repository. arXiv preprint arXiv:1512.03012 (2015)
5. Chen, Q., Chen, Z., Zhou, H., Zhang, H.: ShaDDR: interactive example-based geometry and texture generation via 3D shape detailization and differentiable rendering. In: ACM SIGGRAPH Asia (2023)
6. Chen, Z., Kim, V.G., Fisher, M., Aigerman, N., Zhang, H., Chaudhuri, S.: Decoran: 3d shape detailization by conditional refinement. In: CVPR. pp. 15740–15749 (2021)
7. Chen, Z., Tagliasacchi, A., Funkhouser, T., Zhang, H.: Neural dual contouring. ACM Transactions on Graphics (Special Issue of SIGGRAPH) **41**(4) (2022)
8. Chen, Z., Zhang, H.: Learning implicit fields for generative shape modeling. In: CVPR. pp. 5939–5948 (2019)
9. Chen, Z., Zhang, H.: Neural marching cubes. ACM Transactions on Graphics (Special Issue of SIGGRAPH Asia) **40**(6) (2021)
10. Chibane, J., Alldieck, T., Pons-Moll, G.: Implicit functions in feature space for 3d shape reconstruction and completion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 6970–6981 (2020)
11. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: ECCV. pp. 628–644 (2016)
12. Denton, E.L., Chintala, S., Fergus, R., et al.: Deep generative image models using a laplacian pyramid of adversarial networks. Advances in neural information processing systems **28** (2015)
13. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of Annual Conference on Computer Graphics and Interactive Techniques. pp. 341–346 (2001)
14. Fish, N., Perry, L., Bermano, A., Cohen-Or, D.: Sketchpatch: Sketch stylization via seamless patch-level synthesis. ACM Transactions on Graphics **39**(6), 1–14 (2020)
15. Gao, J., Shen, T., Wang, Z., Chen, W., Yin, K., Li, D., Litany, O., Gojcic, Z., Fidler, S.: Get3d: A generative model of high quality 3d textured shapes learned from images. In: NeurIPS (2022)

16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. *Communications of the ACM* **63**(11), 139–144 (2020)
17. Hertz, A., Hanocka, R., Giryes, R., Cohen-Or, D.: Deep geometric texture synthesis. *ACM Transactions on Graphics (TOG)* **39**(4), 108–1 (2020)
18. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *NeurIPS* **33**, 6840–6851 (2020)
19. Hui, K.H., Li, R., Hu, J., Fu, C.W.: Neural wavelet-domain diffusion for 3d shape generation. In: *SIGGRAPH Asia Conference Papers*. pp. 1–9 (2022)
20. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: *CVPR*. pp. 1125–1134 (2017)
21. Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463* (2023)
22. Kajiya, J.T., Kay, T.L.: Rendering fur with three dimensional textures. *ACM SIGGRAPH Computer Graphics* **23**(3), 271–280 (1989)
23. Karnewar, A., Ritschel, T., Wang, O., Mitra, N.: 3inGAN: Learning a 3D generative model from images of a self-similar scene. In: *3DV* (2022)
24. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: *International Conference on Learning Representations* (2018)
25. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013)
26. Li, M., Duan, Y., Zhou, J., Lu, J.: Diffusion-sdf: Text-to-shape via voxelized diffusion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 12642–12651 (2023)
27. Li, W., Chen, X., Wang, J., Chen, B.: Patch-based 3d natural scene generation from a single example. In: *CVPR*. pp. 16762–16772 (2023)
28. Lin, C.H., Gao, J., Tang, L., Takikawa, T., Zeng, X., Huang, X., Kreis, K., Fidler, S., Liu, M.Y., Lin, T.Y.: Magic3d: High-resolution text-to-3d content creation. In: *CVPR* (2023)
29. Liu, H.T.D., Kim, V.G., Chaudhuri, S., Aigerman, N., Jacobson, A.: Neural subdivision. *ACM Transactions on Graphics (TOG)* **39**(4), 124–1 (2020)
30. Liu, H.T.D., Tao, M., Jacobson, A.: Paparazzi: surface editing by way of multi-view image processing. *ACM Trans. Graph.* **37**(6), 221–1 (2018)
31. Liu, M., Xu, C., Jin, H., Chen, L., Varma, M., Xu, Z., Su, H.: One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. In: *Neural Information Processing Systems* (2023)
32. Lorensen, W.E., Cline, H.E.: Marching cubes: A high resolution 3d surface construction algorithm. In: *SIGGRAPH* (1987)
33. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: *ICCV* (2017)
34. Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A.: Occupancy networks: Learning 3d reconstruction in function space. In: *CVPR*. pp. 4460–4470 (2019)
35. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: *CVPR*. pp. 13492–13502 (2022)
36. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM* **65**(1), 99–106 (2021)

37. Neyret, F.: Modeling, animating, and rendering complex scenes using volumetric textures. *IEEE Transactions on Visualization and Computer Graphics* **4**(1), 55–70 (1998)
38. Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022)
39. Park, J.J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S.: DeepSDF: Learning continuous signed distance functions for shape representation. In: *CVPR*. pp. 165–174 (2019)
40. Peng, S., Niemeyer, M., Mescheder, L., Pollefeys, M., Geiger, A.: Convolutional occupancy networks. In: *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III* 16. pp. 523–540. Springer (2020)
41. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: DreamFusion: Text-to-3D using 2D diffusion. *arXiv preprint arXiv:2209.14988* (2022)
42. Sellán, S., Batty, C., Stein, O.: Reach for the spheres: Tangency-aware surface reconstruction of sdfs. In: *ACM SIGGRAPH Asia* (2023)
43. Shaham, T.R., Dekel, T., Michaeli, T.: Singan: Learning a generative model from a single natural image. In: *ICCV*. pp. 4570–4580 (2019)
44. Shen, T., Gao, J., Yin, K., Liu, M.Y., Fidler, S.: Deep marching tetrahedra: a hybrid representation for high-resolution 3D shape synthesis. In: *NeurIPS* (2021)
45. Siddiqui, Y., Thies, J., Ma, F., Shan, Q., Nießner, M., Dai, A.: RetrievalFUSE: Neural 3D scene reconstruction with a database. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 12568–12577 (2021)
46. Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., Ganguli, S.: Deep unsupervised learning using nonequilibrium thermodynamics. In: *ICML*. pp. 2256–2265. PMLR (2015)
47. Tang, J., Wang, T., Zhang, B., Zhang, T., Yi, R., Ma, L., Chen, D.: Make-it-3D: High-fidelity 3D creation from a single image with diffusion prior (2023)
48. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional GANs. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 8798–8807 (2018)
49. Wang, Y., Chen, X., Chen, B.: Singrav: Learning a generative radiance volume from a single natural scene. *arXiv preprint arXiv:2210.01202* (2022)
50. Wu, J., Zhang, C., Xue, T., Freeman, B., Tenenbaum, J.: Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *NeurIPS* **29** (2016)
51. Wu, R., Zheng, C.: Learning to generate 3D shapes from a single example. *ACM Transactions on Graphics* **41**(6), 1–19 (2022)
52. Yin, K., Gao, J., Shugrina, M., Khamis, S., Fidler, S.: 3dstylenet: Creating 3D shapes with geometric and texture style variations. In: *ICCV* (2021)
53. Zeng, X., Wahdat, A., Williams, F., Gojcic, Z., Litany, O., Fidler, S., Kreis, K.: Lion: Latent point diffusion models for 3D shape generation. In: *NeurIPS* (2022)
54. Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., Metaxas, D.N.: StackGAN: Text to photo-realistic image synthesis with stacked generative adversarial networks. In: *Proceedings of the IEEE international conference on computer vision*. pp. 5907–5915 (2017)
55. Zhou, K., Huang, X., Wang, X., Tong, Y., Desbrun, M., Guo, B., Shum, H.Y.: Mesh quilting for geometric texture synthesis. In: *ACM SIGGRAPH 2006 Papers*, pp. 690–697 (2006)

DECOLLAGE: 3D Detailization by Controllable, Localized, and Learned Geometry Enhancement

(Supplementary Material)

A Data and code

We show the style shapes for training chair and table style mixing in Figure 14, and plant, building, cake and crystal in Figure 15. We also provide the ready-to-use data and code in the supplementary. Code will be released upon acceptance.

B Loss function

We follow the notations defined in the main paper and provide the discriminator loss.

Discriminator loss. For any style shape s and any coarse shape c with designated styles $S(v)$, the discriminator loss is the sum of the global discriminator D_*^j 's loss and the style-specific discriminators D_i^j 's loss:

$$L_{D^j} = L_{D_*^j} + L_{D_{style}^j} \quad (6)$$

where

$$L_{D_*^j} = \mathbb{E}_v((D_*^j(s^j)[v] - 1)^2 + (D_*^j(G^j(c))[v])^2) \quad (7)$$

$$L_{D_{style}^j} = \mathbb{E}_v((D_{S(s[v])}^j(s^j)[v] - 1)^2 + (D_{S(c[v])}^j(G^j(c))[v])^2) \quad (8)$$

C Evaluation metrics

We use the following metrics from DECOR-GAN [6] to quantitatively evaluate the quality of the generated shapes on both single-category detailization and multi-category style mixing tasks.

Strict-IOU and Loose-IOU. Strict-IOU is to measure the Intersection over Union (IOU) between the downsampled output voxels and the input voxels, to evaluate how well the generated shape respects the structures in the input shape. Loose-IOU is a relaxed version of Strict-IOU to compute the proportion of occupied voxels in the input that are also occupied in the downsampled output.

LP-IOU and LP-F-score. LP-IOU and LP-F-score are to measure the percentage of local patches in the generated shape that are ‘‘similar’’ (according to IOU or F-score) to at least one local patch in the style shapes. Higher LP-IOU and LP-F-score indicate that the local details of the generated shapes are more

similar to the local details of the style shapes, thus the generated shapes are deemed to be more locally plausible. We mark the two patches as “similar” if the IOU (F-score) is above 0.95. To reduce the computational complexity, we sample 12^3 patches in a voxel model, a size slightly smaller than the receptive field of the discriminator. Additionally, We only sample surface patches with at least one occupied voxel and one unoccupied voxel in their central 2^3 areas to avoid sampling featureless patches located far inside or outside the shape. We sample 1000 patches from each testing shape and compare them with all potential patches in the detailed shapes.

Cls-score. Cls-score is to evaluate the overall plausibility of the generated shapes by training a classifier network to distinguish between the rendered images of the generated shapes and those of the real shapes and recording the mean classification accuracy. We train a ResNet using high-resolution voxels (from which content shapes are downsampled) as real samples and our generated shapes as fake samples. For each sample, we randomly render 24 images with a resolution of 256^2 . The images are randomly cropped to 10 small patches with a resolution of 64^2 for training.

Evaluation details. For IOU and LP, we evaluate 320 generated shapes (20 contents \times 16 styles) since they are computationally expensive. For Cls-score, we evaluate 1600 generated shapes (100 contents \times 16 styles). For multi-category style mixing, we generate 16 style sets, each containing 5 random style combinations.

D More results on single category detailization

We show qualitative and quantitative results on the building category in Figure 16 and Table 5, and the plant category in Figure 16 and Table 6.

Table 5: Quantitative of single-category detailization on the building category.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-IOU \uparrow	LP-F-score \uparrow	Cls-score \downarrow
DECOR-GAN	0.693	0.973	0.429	0.662	0.598
ShaDDR	0.601	0.957	0.425	0.633	0.633
Ours	0.732	0.987	0.442	0.648	0.592

Table 6: Quantitative of single-category detailization on the plant category.

	Strict-IOU \uparrow	Loose-IOU \uparrow	LP-IOU \uparrow	LP-F-score \uparrow	Cls-score \downarrow
DECOR-GAN	0.417	0.728	0.385	0.769	0.648
ShaDDR	0.217	0.636	0.220	0.522	0.673
Ours	0.419	0.757	0.358	0.771	0.629

E More results on multi-category style mixing

We show additional qualitative results on the chair and table style mixing in Figure 17 and 18, and plant, building, cake and crystal in Figure 19 and 20.

F More analysis of the generative capability

Our network design and losses (e.g., structure preservation) train the model to pick reasonable/plausible details to generate for each part regardless of its spatial location, as unreasonable/implausible details will lead to sub-optimal losses. This contributes to the fact that most of our results exhibit transfers between structurally matched parts, which, under normal circumstances, would also represent parts that share similar spatial locations (e.g., chair back to back).

That said, our generative capability is certainly not constrained by the relative spatial positions of the parts. Please see Fig. 13 for additional examples. Specifically, Figs. 13 (a-c) show that our model can generate armrests in various positions, even on just *one side* (b). Fig. 13 (d) shows a result of applying the style of tabletop to the chair back and (e) shows a chair back style detailedized both in the middle of and in front of the seat, even between chair legs.

Even when part labels are removed from the input, our method is still able to generate reasonable geometry that respects both the style shape and the structure of the coarse voxel shape, as shown in Fig. 13 (f).

Note that changing style guidance from one to another *within* the training style shapes does not require retraining. On the other hand, the model needs to be retrained to include new style shapes that are unseen during training.

G GUI demo

After the user assigns styles to each region, our method takes less than a second ($\sim 0.3s$) to generate style-mixing results. We provide a video of our GUI demo powered by the network *with* part labels version in the supplementary.

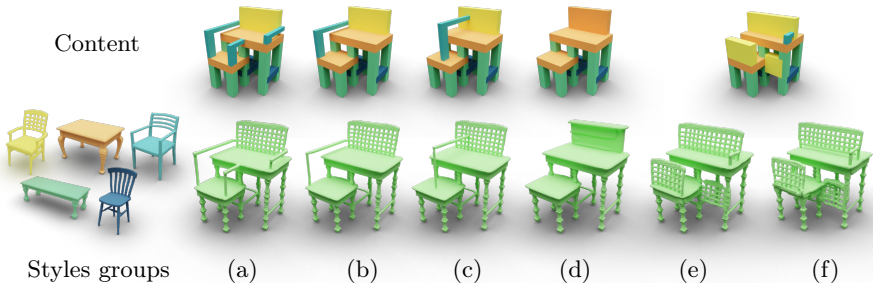


Fig. 13: (a)-(e) *with* part labels, (f) *without* part labels. Please zoom in to observe the local geometric details.



Fig. 14: The segmented training style shapes of chair and table categories. We segment the chair into five parts: back, seat, leg, armrest and stretcher and the table into two parts: tabletop and leg.

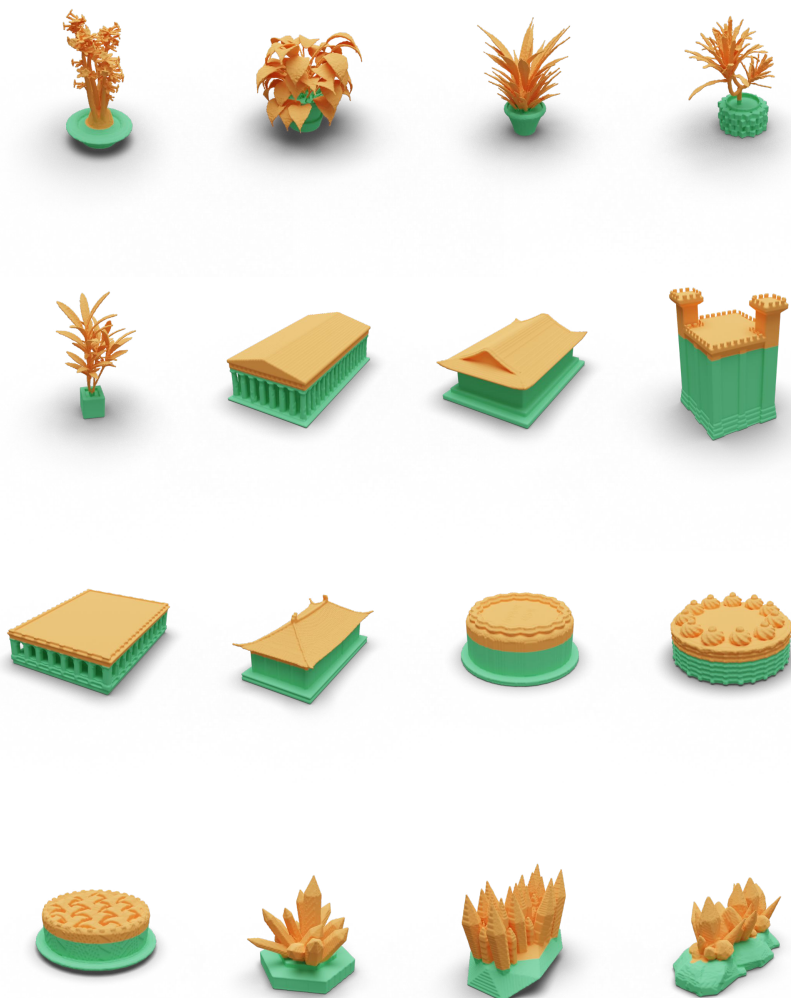


Fig. 15: The segmented training style shapes of plant, building, cake and crystal categories. We segment the plant into two parts: pot and leaves, the building into two parts: main body and roof, the cake and crystal into two parts: bottom and top.

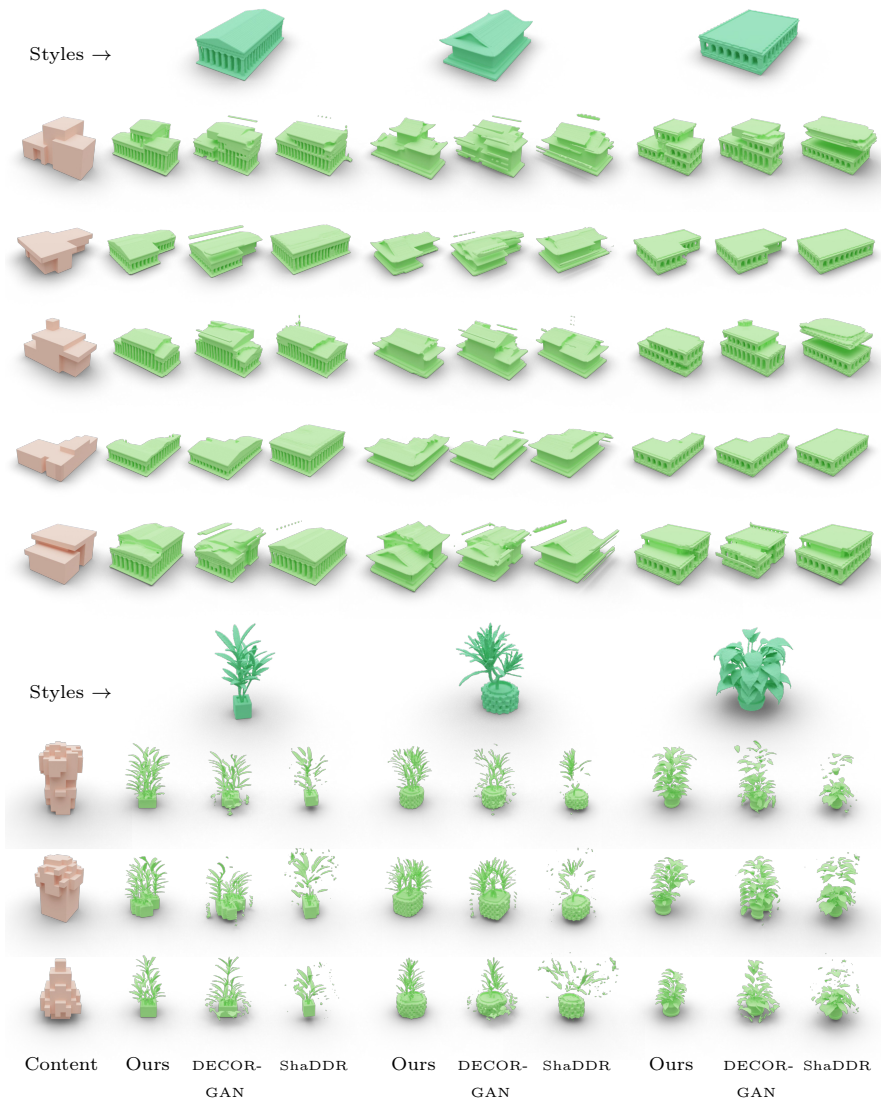


Fig. 16: Single-category detailization on building and plant categories. For each category, we show the input content shapes on the left and style shapes on top.

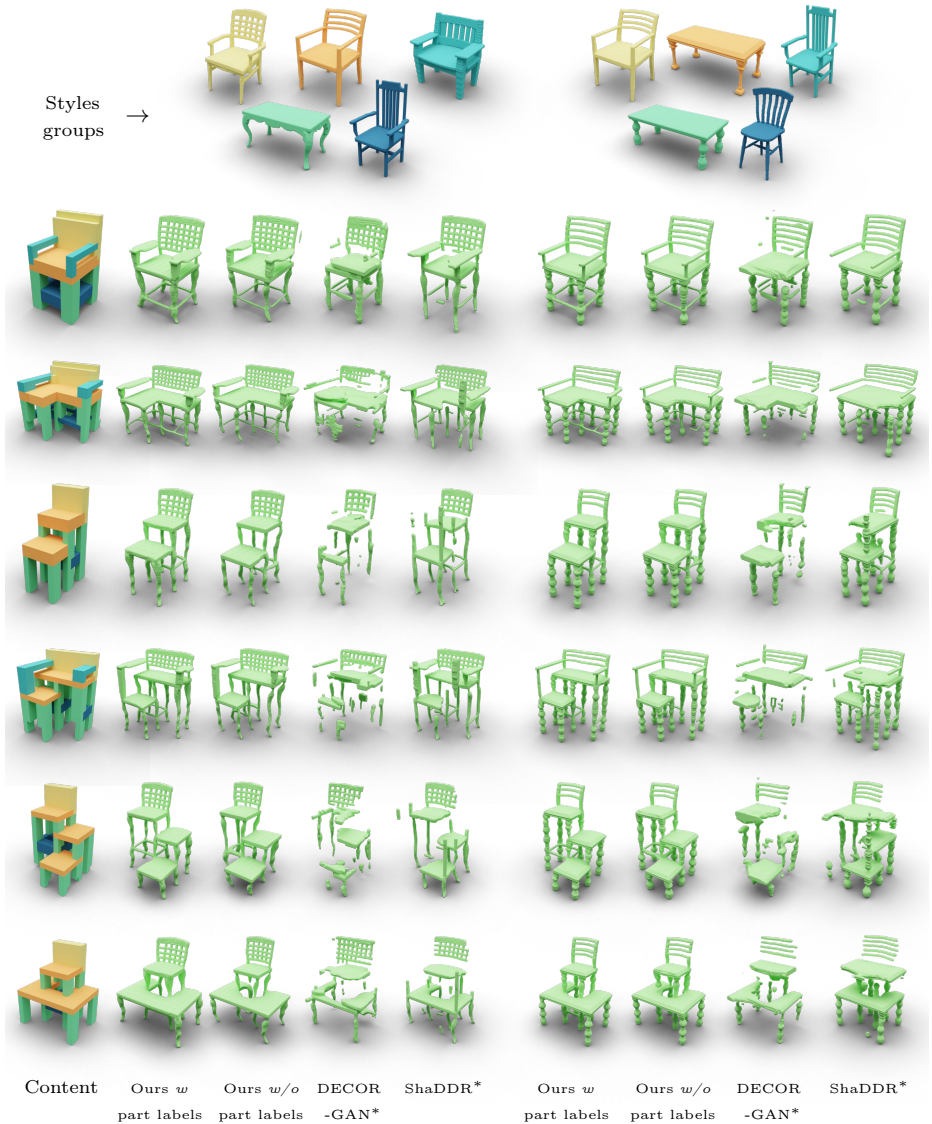


Fig. 17: Multi-category style mixing results on chair and table categories. We show the input coarse voxels with style labels on the left. The corresponding style shapes for the colored style labels are shown on top. Please zoom in to observe the local geometric details.

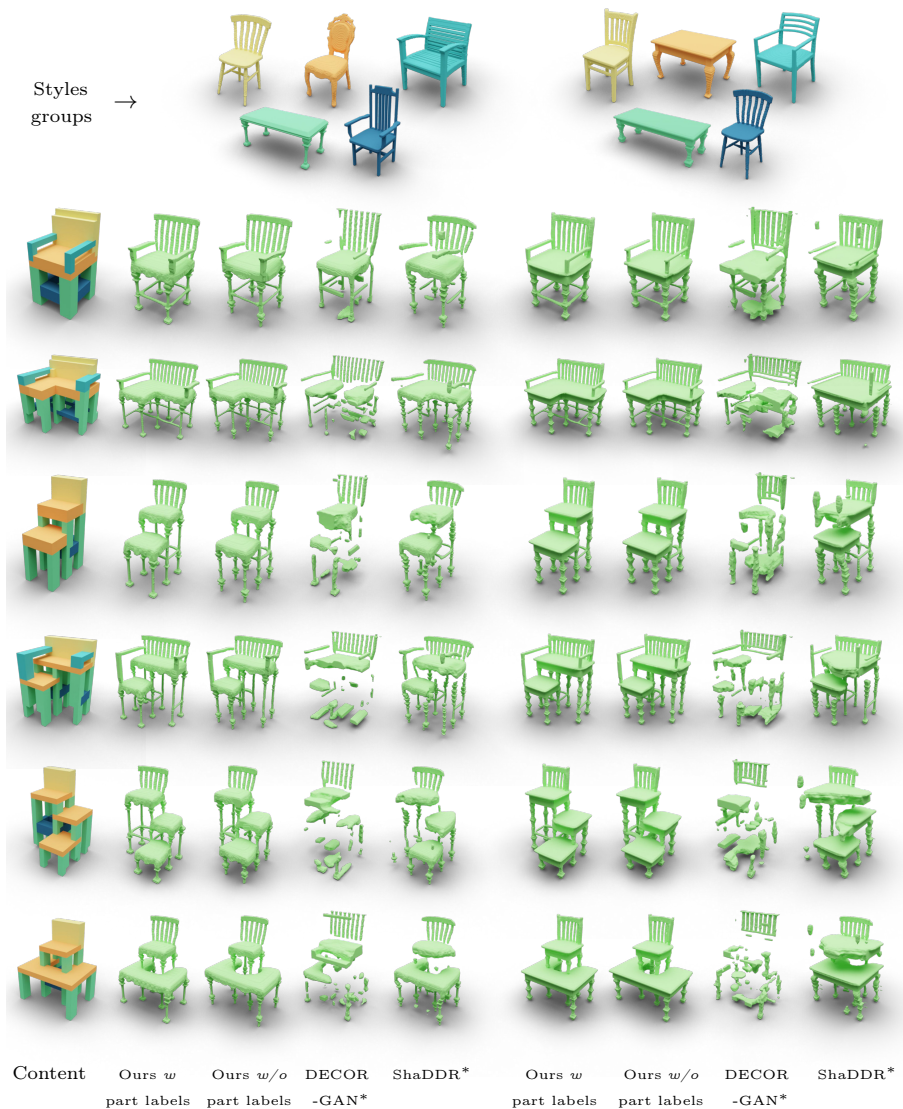


Fig. 18: Multi-category style mixing results on chair and table categories. We show the input coarse voxels with style labels on the left. The corresponding style shapes for the colored style labels are shown on top. Please zoom in to observe the local geometric details.

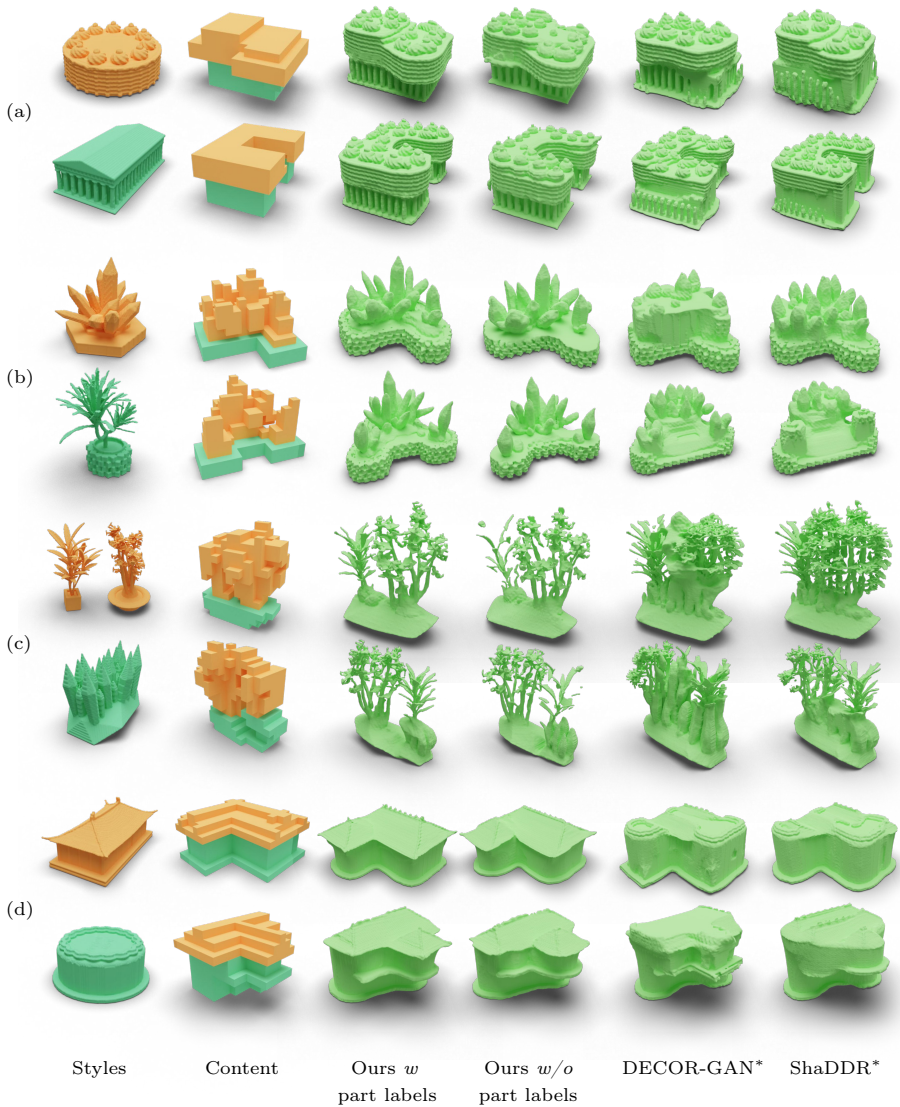


Fig. 19: Multi-category style mixing results on plant, building, cake, and crystal categories. For each group, e.g., (a), we show two style shapes in the first column, coarse input shapes with style labels in the second column, and results in the remaining columns. Please zoom in to observe the local geometric details.

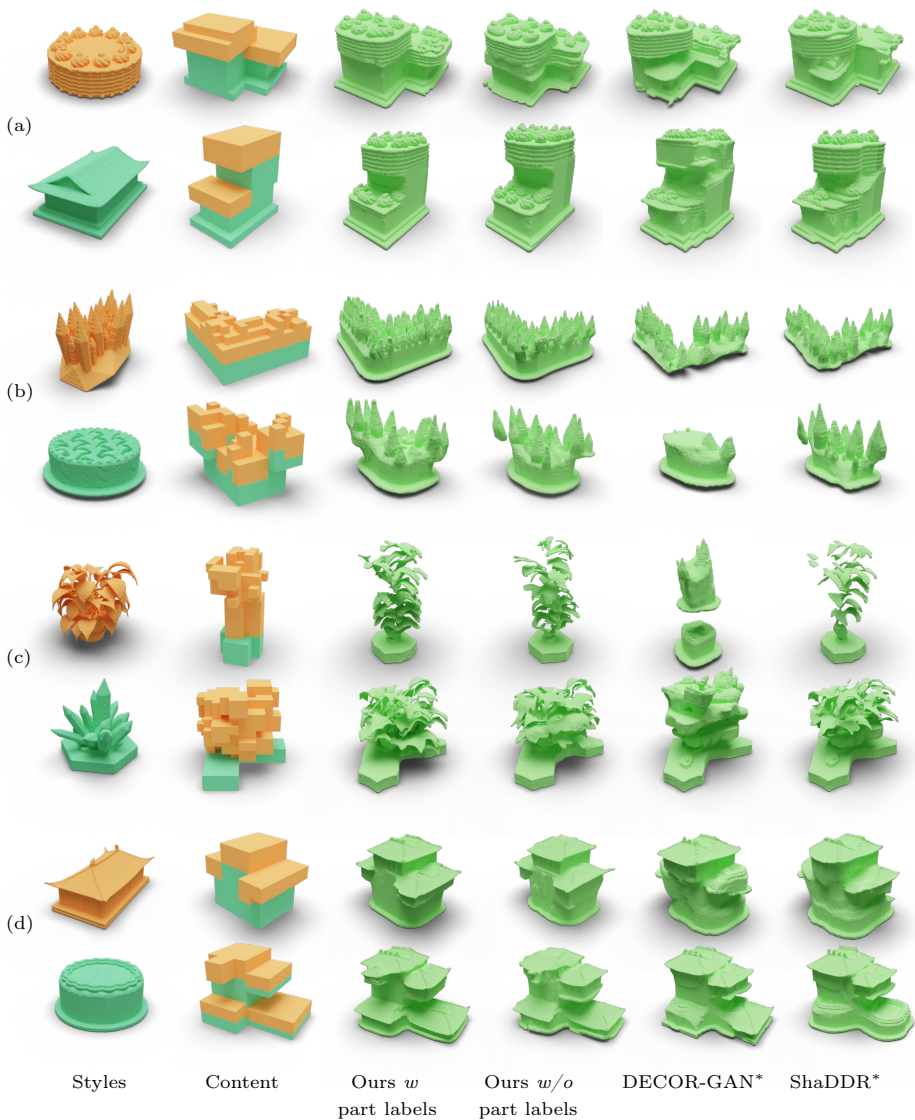


Fig. 20: Multi-category style mixing results on plant, building, cake, and crystal categories. For each group, e.g., (a), we show two style shapes in the first column, coarse input shapes with style labels in the second column, and results in the remaining columns. Please zoom in to observe the local geometric details.