PLSR: Progressive and Localized Super-Resolution of 3D Objects via Localized Latent Voxel Diffusion

Yuxin Liu[1], Minshan Xie[2], Jiawen Liang[3], Runsong Zhu[1]*✉, Chi-Wing Fu[1], and Tien-Tsin Wong[4]

[1] The Chinese University of Hong Kong, Hong Kong SAR, China
[2] Guangdong University of Technology, China
[3] City University of Hong Kong, Hong Kong SAR, China
[4] Monash University, Australia
yxliu22@cse.cuhk.edu.hk, zhurunsong@gmail.com

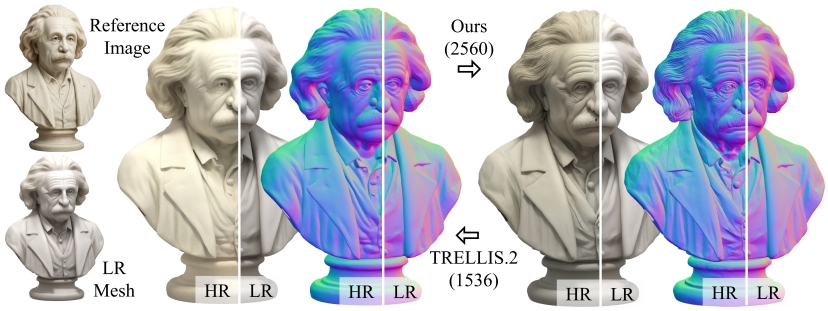

Fig. 1: We propose PLSR, a method that tackles the high-resolution mesh generation problem through **P**rogressive and **L**ocalized **S**uper-**R**esolution. Our method provides extra resolution increase over state-of-the-art 3D generation methods (e.g., TRELLIS.2). Zoom in to see the fine details.

Abstract. High-resolution 3D asset generation is vital in various 3D applications. Existing state-of-the-art diffusion-based models remain constrained by fixed resolutions, limiting their ability to produce details. In this paper, we tackle the challenge of generating more detailed, higher-resolution 3D objects by introducing a 3D super-resolution (SR) framework built on existing 3D generative foundation models. To this end, we design **PLSR**, a progressive and localized super-resolution solution to achieve this goal effectively and memory efficiently. Technically, given a coarse geometry from a pretrained 3D generator, we decompose the global SR task into localized sub-tasks via an **associative input decomposition** scheme, adapt a flow-based 3D generator into a **localized super-resolution model** through low-cost finetuning, and unify

* Corresponding author

them in an iterative **patch-wise denoising pipeline** for seamless high-resolution output. Experiments on challenging objects show that our approach is able to generate 3D details with new strong fine-detail fidelity while significantly reducing the computational cost, offering a new and practical solution for high-resolution 3D asset generation.

Keywords: 3D asset generation · super-resolution · progressive · localized · diffusion-based 3D models

1 Introduction

High-resolution 3D geometric models are foundational to numerous industry applications, including computer games, TV/movie production, computer-aided design and fast prototyping. Traditionally, creating these models has been time-consuming and requires substantial expertise, and hence limited to trained professionals. Recent advances in generative artificial intelligence have lowered this barrier by enabling the automatic generation of 3D assets from images.

Diffusion and flow-based models have demonstrated superior performance in generating high-quality 3D content, emerging as a powerful and general solution for 3D asset generation. In particular, recent methods such as XCube [39], Craftsman, Hunyuan3D [61], and TRELLIS.2 [53] introduce various techniques to enhance the scalability. While significant improvement has been achieved, these models are still typically trained at a fixed resolution in voxel grids or latent spaces, due to the cubic growth in memory and compute costs. On the other hand, users' desired level of details for 3D assets can vary arbitrarily.

In this paper, we tackle the challenge of generating more detailed, higher-resolution 3D objects by introducing a 3D super-resolution (SR) framework built on existing 3D generative foundation models. Specifically, a pretrained 3D generator is first employed to produce a coarse global mesh, which is then refined by a dedicated SR model designed to synthesize fine-grained details. Benefiting from this SR setting, our solution naturally inherits the structural priors from the object-level shape to generate challenging fine details, making the learning process more effective.

To do so, we propose a **progressive** and **localized** SR (PLSR) framework to generate details with two goals in mind. First, the super-resolution must primarily rely on local information rather than global information, so that we can adopt a divide-and-conquer strategy to decompose the global SR task into multiple independently processed patches, yielding better memory efficiency for both training and inference. Second, we want this localized training setting to learn an image-conditioned mapping between LR and HR patches, so that it can be independent of the absolute object resolution. This allows the SR model to be compatible with objects at any resolution, which is essential for progressive super-resolution.

Specifically, given a coarse global geometry produced by a pretrained 3D generator, we use an **associative input decomposition** scheme to decompose global SR task into localized sub-tasks with aligned local conditions, and adapt

an object-level flow-based 3D generator into a **localized super-resolution model** through low-cost finetuning, which is responsible for processing localized SR tasks. Finally, we design an iterative **patch-wise denoising pipeline** where global latents are decomposed, processed and stitched per denoising iteration, achieving seamless high-resolution SR using localized processing.

Experiments across various challenging test cases demonstrate that our method consistently improves fine-detail fidelity under a practical single-GPU setting. By leveraging localized conditioning and progressive refinement, our framework super-resolves coarse 3D objects in the structured latent [53] domain, achieving higher resolutions with only low-cost fine-tuning and offering a practical solution for scalable 3D super-resolution. Our contributions are summarized as follows:

- **Resource-efficient Super-Resolution (SR) Framework**: We introduce a novel SR framework that achieves substantial resolution enhancement without requiring multi-resolution HR training data or multi-GPU compute, making high-quality 3D generation more accessible.
- **Patch-wise refinement model**: We propose a model design that enables progressive and localized super-resolution without assuming the whole-object generation as in existing methods.
- **Fine-detail generation**: Experiments on challenging cases demonstrate the superior performance of PLSR to generate fine details.

2 Related Work

2.1 3D Content Generation

3D content generation is a long-standing challenge in computer graphics and vision [15, 19, 60]. Early progress was limited by the scale and diversity of classic repositories such as ShapeNet [6]. To address data scarcity, zero-shot pipelines [20, 33] optimized text-prompted 3D representations using 2D language–image priors like CLIP [38]. With the advances of 2D diffusion models [40], score distillation sampling (SDS) became the dominant paradigm for text-to-3D generation [35], inspiring variants such as ProlificDreamer [48] and DreamGaussian [46]. But SDS-based pipelines remain iterative and suffer from multi-view inconsistency.

The release of large-scale datasets such as Objaverse [11] and Objaverse-XL [10] enabled feed-forward 3D generation. Leveraging multi-view supervision, LRM [17] introduced an end-to-end regression framework that predicts triplane NeRF representations [5] from a single image, while Instant3D [25] adopts a two-stage pipeline that first synthesizes a "view-compiled" image grid from text prompts and then reconstructs 3D geometry. Recent works improve efficiency and fidelity through Gaussian splatting (LGM [45]), convolutional backbones (CRM [49]), and enhanced geometric priors (InstantMesh [55]).

To improve generation quality while maintaining efficiency, viewpoint-conditioned multi-view diffusion has emerged as a key technique. These models synthesize consistent multi-view images conditioned on text or single-view inputs, which are then lifted into 3D through reconstruction. Zero123++ [42] models joint

distributions of canonical views by tiling them into a single frame, while SyncDreamer [29] enforces cross-view coherence via synchronized noise prediction and intermediate 3D feature volumes. Wonder3D [31] employs cross-domain diffusion to generate multi-view normals and colors, followed by explicit geometry extraction through normal fusion. Era3D [26] improves scalability and camera prior alignment with diffusion-based camera regression and efficient epipolar attention. Unique3D [50] further advances multi-view generation and upscaling with a multi-stage surface reconstruction pipeline.

Beyond 2D lifting, another promising direction is the development of native 3D diffusion and large-scale generative models that learn distributions directly over 3D latents or geometries. Direct3D [51] introduces a triplane-latent VAE and diffusion transformer for image-conditioned generation without SDS. CLAY [57] scales to 1.5B parameters for geometry and PBR textures, while industrial systems such as Hunyuan3D 2.0 [61] integrate flow-based geometry diffusion with texture synthesis and production-ready editing tools. Large-scale pipelines like TRELLIS [54] and TRELLIS.2 [53] further advance scalability by combining hierarchical latent spaces with multi-resolution diffusion, enabling high-fidelity asset generation for professional content creation, with TRELLIS.2 supporting up to 1536^3 through training-free upscaling. Representation-centric works also aim for high-resolution outputs. Specifically, recent work XCube [39] supports sparse voxel grids up to 1024^3, and TetraDiffusion [22] uses tetrahedral partitions for efficient mesh generation.

Very recently, a concurrent work (*i.e.*, LATTICE [24]) introduced VoxSet for high-resolution decoding, achieving increased resolution by training on a large-scale dataset at a substantial computational cost. However, most approaches still require multi-scale high-resolution supervision or heavy multi-GPU training, motivating cost-effective test-time resolution scaling such as localized patch refinement that reuses object-level priors without multi-scale retraining.

2.2 Super-Resolution for 3D Content/Geometry

Rather than synthesizing textured meshes from scratch, another line of work enhances geometric surfaces of existing 3D models. Analogous to image super-resolution, point-cloud upsampling aims to densify sparse inputs uniformly on the underlying surface [32,36,37,56], but generally does not introduce novel fine-scale geometry. Exemplar-based detail transfer can map local patterns from a high-quality source mesh to a target [2], and recent work extends such paradigms to coarse voxel refinement for few-shot detailization [7]. Text-guided mesh editing frameworks enable semantic manipulation but struggle with precise fine-grained control [14]. PBR material generation often estimates normal/bump maps that can be exploited as refinement signals [47].

Diffusion-based SR approaches aim to recover high-frequency details directly. SDF-Diffusion [43] adopts a two-stage pipeline, first generating a low-resolution signed distance field (SDF) and then applying a patch-based SR model trained on paired samples. For radiance-field or Gaussian-based scenes, 3DSR [8] leverages off-the-shelf 2D diffusion SR within a 3D Gaussian-splat representation

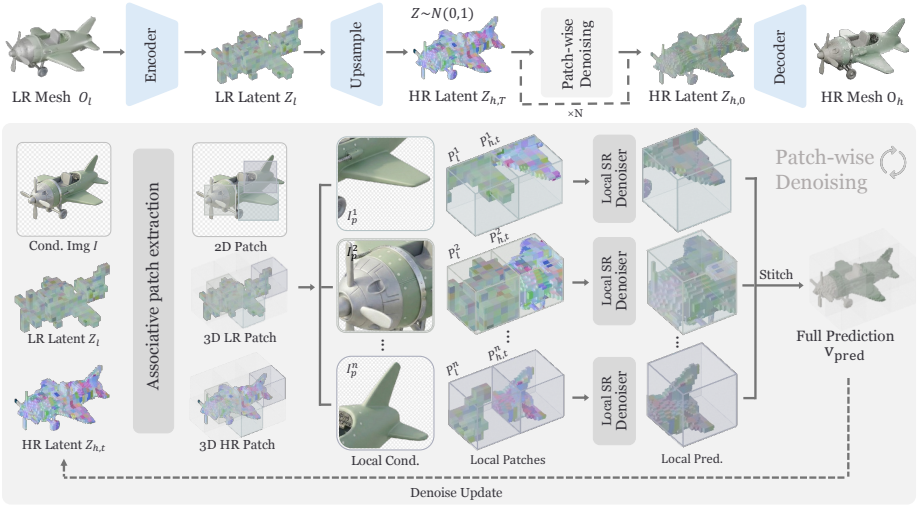

Fig. 2: Overview of our PLSR. Starting from a coarse mesh O_l, we encode to obtain LR condition Z_l, upsample to obtain the sparse voxel skeleton of HR latent, and initialize with Gaussian noise to form $Z_{h,T}$. Rather than denoising Z_h globally, we split Z_h into fixed-size patches and crop Z_l and image I associatively to obtain a triplet, $(P_{h,t}, P_l, I_p)$. During iterative patch-wise denoising, our model iterates through the input tuples and predicts for each individual patch. The per-patch outputs are aggregated into a global prediction v_{pred} for denoising update. Once Z_h is fully denoised, it can be decoded to an output mesh or be used as new LR condition Z_l for the next SR round. Particularly, our localized SR process can be performed progressively to achieve higher resolution.

to enhance resolution while maintaining cross-view consistency. Mesh-oriented pipelines, e.g. SuperCarver [58], refine multi-view normal maps via diffusion and update geometry through inverse rendering, achieving up to 16× surface-detail enhancement with texture alignment. However, these methods either require high-resolution supervision or are tailored to mesh or radiance-field settings.

Existing SR methods improve fidelity but typically depend on high-resolution training or specialized architectures. In contrast, we propose a novel localized SR framework, providing a scalable solution for high-resolution detail under limited computational resources.

3 Method

Figure 2 illustrates our **progressive** and **localized** super-resolution pipeline. Our goal is to generate a high-resolution textured mesh $\mathcal{O}_h$ from a coarse input mesh $\mathcal{O}_l$ conditioned on one or more reference images I with known camera parameters in a memory-efficient way. We first introduce the basic concepts in Sec. 3.1, associative input decomposition in Sec. 3.2, our localized SR model in Sec. 3.3, model training in Sec. 3.4, and inference pipeline in Sec. 3.5.

3.1 Preliminaries: Structured Latents and Flow-Based Generation

Structured sparse latent. Following [53], we represent a 3D object by a compact VAE latent Z defined on a sparse voxel support $\mathcal{S}_r \subset \mathbb{Z}^3$ at resolution r. Each active voxel index in $\mathcal{S}_r$ holds a d-dimensional feature, yielding $Z \in \mathbb{R}^{|\mathcal{S}_r| \times d}$. A pretrained encoder/decoder maps between meshes $\mathcal{O}$ and such sparse latents. In practice, TRELLIS.2 converts meshes into high-resolution (1024^3) intermediate o-voxel representation, and encodes the o-voxel representation with a convolutional VAE encoder to provide $\times 16$ compression of the voxel representation along each axis, into a compact structured latent. The *structured latent* itself takes a negligible amount of memory even at high-resolution. Its voxel structure also naturally supports easy patch operations, and stitching via weighted average, which we leverage for localized SR.

Flow model: velocity and sampling. We generate (or refine) a latent Z on a prescribed support $\mathcal{S}_r$ using a flow model [27,28] that predicts a time-dependent *velocity* field $v_\theta(Z_t, t \mid \mathcal{C})$, where Z_t denotes the latent at time $t \in [0, 1]$ and $\mathcal{C}$ collects conditioning signals (e.g., the LR latent Z_l, image crops I_p). Intuitively, v_θ defines how to *move* the current latent to become slightly closer to the target data at time t. Starting from Gaussian noise $Z_1 \sim \mathcal{N}(0, I)$ on $\mathcal{S}_r$, we integrate the ODE

$$\frac{dZ_t}{dt} \;=\; v_\theta(Z_t, t \mid \mathcal{C}), \tag{1}$$

from t=1 to t=0 to obtain a clean latent Z_0, which is then decoded to a mesh $\mathcal{O}$. In practice we use a discrete solver with step size Δt, yielding the update

$$Z_{t-\Delta t} \;=\; Z_t \;+\; \Delta t \cdot v_\theta(Z_t, t \mid \mathcal{C}), \tag{2}$$

where each step can be interpreted as a small *denoising* move guided by the predicted velocity. In later sections, we instantiate $\mathcal{C}$ with (Z_l, I) and apply Eq. (2) in a patch-wise manner.

3.2 Associative Input Decomposition

Different from a typical image-guided 3D super-resolution setup where all inputs are global, our localized super-resolution requires local input triplets (noisy high-resolution latent patch P_h, low-resolution latent patch P_l, and corresponding local reference image I_p) to perform image-conditioned local super-resolution. This aligns with the core idea of learning relative super-resolution independent of absolute object resolution, as it encourages the model to learn its super-resolution mostly on local information while preventing it from relying on global cues to form its outputs.

However, the local conditioning image has to be defined carefully and in a consistent way such that a similar image distribution would apply to both training and inference to minimize the gap. Considering users typically provide only global reference images as conditions in practice, the local conditioning

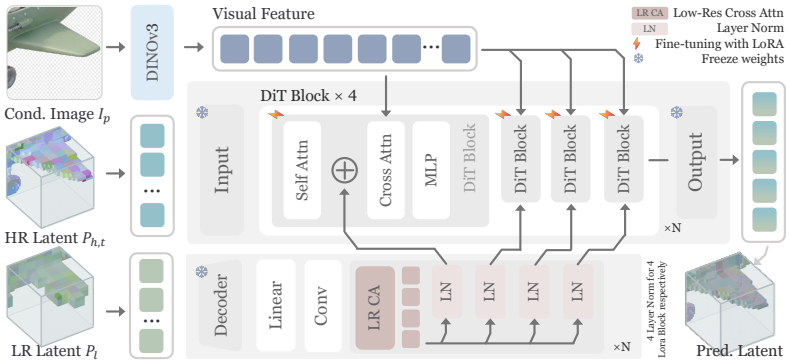

Fig. 3: Architecture of our proposed localized SR model.

image must be easily obtainable, given the global image. Therefore, we choose to use image crops from a global reference image centered around the target 3D patch as a suitable proxy of local reference image of the patch, as a crop can contain most necessary information to generate local details with minimal irrelevant information.

To obtain the triplet, we associatively extract the patches to ensure their alignment. Given a patch bounding box in 3D space, we first locate and crop the partial structured latent P_h and P_l from the global latent grids. For patch image condition, we project the patch bounding box onto the conditioning image using known camera parameters, thereby extracting localized image features. By cropping the conditioning image with this patch bounding box, we effectively obtain a close-up shot of the patch currently being refined, providing both localized and fine-grained features necessary for refinement. This patch extraction method is used in both training and inference for consistency.

As our base flow model embeds the positions of active voxels of the structured latent into features to ensure positional-awareness during generation, we choose to enforce both LR and HR patch to use voxel coordinates relative to the patch origin instead of the global coordinates in both training and inference, preventing the model from relying on global positional cues for prediction.

3.3 Localized Super-Resolution Model

Basic architecture. We base our model on the pretrained sparse latent DiT model proposed in TRELLIS.2 [53]. The original DiT model consists of a sequence of transformer blocks with image cross attention, AdaLN-single modulation and Rotary Position Embedding. Given noisy structured latent Z_t, the flow model predicts velocity v conditioned on DINOv3 image features F_I extracted from I and timestep t. We incorporate an LR conditioning branch into the model to take additional LR conditioning features extracted from LR latent patch using the first two decoder blocks of the pretrained VAE decoder, denoted as $\mathcal{D}_2$.

The last layer of $\mathcal{D}_2$ upsamples the LR feature by $\times 2$, aligning it with $P_{h,t}$. The model architecture is shown in Fig. 3. Formally, our model v_θ predicts velocity:

$$v = v_\theta(P_{h,t}, \mathcal{D}_2(P_l), F_{I_p}, t), \tag{3}$$

where $P_{h,t}$ denotes noisy high-resolution latent patch at time t; P_l denotes its corresponding low-resolution latent patch; I_p denotes the corresponding image patch obtained by randomly extracting patches from training data associatively.

Lightweight LR Attention for Flexible Conditioning. Naturally, channel-wise concatenating the condition to input is a simple-yet-effective way to achieve spatially aligned conditioning. While our LR feature $\mathcal{D}_2(P_l)$ is at the same resolution as input P_h, we observed that they exhibit slight voxel coordinate discrepancies, probably due to slight shape changes that occur during remeshing and simplification used to degrade LR mesh. This small discrepancy makes channel concatenation strategies ineffective due to the strict spatial alignment requirement. We instead choose to introduce a lightweight LR attention layer to overcome the mismatch problem. These layers compute attention between each HR voxel and its neighboring LR cells within a small radius, enabling flexible information transfer without resorting to heavy global attention. To keep the LR conditioning branch lightweight, we include a dedicated LR attention layer only in a subset of transformer blocks of the original flow model, and reuse the resulting features across subsequent blocks until a new attention computation is performed. To ensure the compatibility with the feature statistics of each block, we apply layer normalization with element-wise affine transformation when reusing LR features. The LR features are added element-wise to transformer feature after self-attention layer. In this way, the trainable parameter in our fine-tuning is substantially reduced, while providing necessary LR signal for robust SR.

3.4 Model Training

To adapt the model to the new task, we adopt LoRA [18] fine-tuning on all linear layers except the input and output layer, avoiding catastrophic forgetting or unstable optimization when fine-tuned on limited dataset and resources. To speed up learning, we reuse the weights from original self-attention layer to initialize our LR lightweight attention layer.

We train our model using standard Conditional Flow Matching objective:

$$\mathcal{L}_{CFM}(\theta) = \mathbb{E}_{t,P_{h,0},\epsilon}||v_\theta(P_h(t), \mathcal{D}_2(P_l), F_{I_p}, t) - (\epsilon - P_{h,0})||_2^2 \tag{4}$$

During training, $P_h(t) = (1-t)P_{h,0} + t\epsilon$ is randomly sampled by injecting Gaussian noise ϵ into $P_{h,0}$ based on random timestep t. A core benefit of this patch formulation lies in its memory and data efficiency during training. On one hand, encoding meshes at high-resolution (i.e., 96^3 latent resolution and beyond) is extremely resource-intensive or even infeasible due to GPU memory constraints. On the other hand, high-quality datasets are not widely accessible. We instead train on abundant P_h data that can be diversely sampled from moderate-resolution

data, bypassing both limitations. To be specific, we encode textured meshes at the same global resolution as in TRELLIS.2 [53] (64^3 latent resolution) and define our model's output patch size to be 32^3, covering half of the volume along each axis.

3.5 Patch-wise Super-Resolution Pipeline

Our pipeline performs patch-based super-resolution in the latent domain, leveraging its compactness and smoothness to keep the full latent representation in memory. The voxel space is partitioned into local patches, which are processed sequentially and later stitched together via weighted feature aggregation. Benefiting from the iterative denoising nature of flow models, we place our split-refine-and-stitch process inside each denoising iteration, so that a consensus prediction will be formed by weighted aggregation of per-patch predictions, enabling mutual awareness of neighboring patches, and providing a unified basis during the next denoising iteration. Compared to fully independent processing and one-time stitching over the output, this iterative aggregation yields better coherence, aligning with the observations in earlier works on image and texture synthesis [1, 30].

Initialization and Conditioning. The pipeline begins with a low-resolution sparse latent (half the target resolution), obtained either from coarse mesh encoding or from previous SR outputs. We first use the pretrained decoder to upsample the LR condition into sparse features at the same resolution as the target output to perform resolution-aligned conditioning. Due to the fact that compressed latents operate at 1/16 resolution per-axis to the raw voxel representation, it is in most cases safe to assume that the voxel skeleton extracted from the coarse object shares an indistinguishable structure compared to its HR counterpart. Thus, for inference, we could reuse the above-mentioned voxel indices to initialize noisy HR latent for denoising.

Patch Processing and Image Conditioning. At each denoising iteration, the 3D space is partitioned into equal-sized windows that cover the entire latent grid, with partial overlaps for neighbor aggregation, and we use associative patch extraction to obtain triplets for each window. When multiple conditioning images are provided by the user, we employ a visibility-based strategy to select one informative image for each denoising step. Specifically, we render the patch voxel structure with a voxel renderer from each camera, and apply depth testing to discard pixels where the target patch is occluded by surrounding regions, yielding an occlusion-aware visibility score for each image. Rather than always choosing the image with maximum visibility, we sample images weighted by visibility at each denoising step, thereby balancing local relevance with global information coverage. For patches occluded from all views, the DINOv3 image features are zeroed-out to indicate no visual cues, falling back to LR 3D latents for stability.

Aggregation and Global Coherence. After predicting per-patch velocity with the flow model, all patches are aggregated at their overlapping region by weighted averaging. As patch-based predictions tend to be less reliable at patch boundaries due to an incomplete context, we weigh patch higher near the center and lower at their boundaries during merging. The aggregated global velocity is used to update the denoising process, i.e., to compute $Z_{h,t-1}$ at the next time step. This $Z_{h,t-1}$ then serves as the input noisy latent in the subsequent iteration, ensuring that patch-wise processing in the next iteration starts from a unified input state.

Progressive Super-Resolution. Our method supports multi-round SR in latent space by feeding the output latent into the next round of SR as low-resolution latent. Each round of super-resolution increases the resolution by a factor of two along each axis, and super-resolving for multiple rounds supports $\times 4$ or more super-resolution relative to the initial LR resolution.

Practical Workflow. In practical image-to-3D workflows where a reference image is present, we can invoke an existing 3D generator to obtain the coarse mesh, align the user-provided reference image with the mesh, and refine based on the estimated camera parameter to boost resolution. For mesh-only input scenarios, users may render the object from known camera views and obtain detailed conditioning images through image editing tools, manual painting, or both.

Selective Super-Resolution. Due to the localized nature of our model, it can also be used for selective super-resolution to refine only in user-specified regions for efficiency. However, due to page limitation, we defer the discussion of this application to the supplementary materials.

4 Experiments

4.1 Experimental Setting

Implementation Details. We train on 30K assets from Objaverse-XL [10], filtered by aesthetic score [41] ≥ 6.5. Each asset is encoded into geometry and texture latents using the TRELLIS.2 VAE at a latent resolution of 64^3 (corresponding to 1024^3 o-voxels [53]). A low-resolution version is obtained by simplifying and re-encoding the same asset at 32^3 resolution. We render 24 viewpoints per asset in Blender 3.3 LTS [3] with randomized FOV settings at 2048^2 resolution to incorporate fine visual details. We sample 3D patches using our associative extraction strategy, producing HR/LR latent pairs of size $32^3/16^3$ along with local image crops at 1024^2 resolution. All parameters of the pretrained TRELLIS.2-4B shape and texture flow DiT are frozen, and we insert LoRA [18] adapters into all linear layers except the input/output layers. Geometry and texture models

are trained using flow-matching loss with classifier-free guidance (drop rate 0.3). A light regularization loss ($\lambda = 0.005$) discourages trivial opening in the image-modulation path. Optimization uses AdamW ($\text{lr} = 1 \times 10^{-5}$, weight decay $= 0.1$) with EMA (0.999). Random color jitter, geometric warp, and local dropout augmentations simulate inconsistencies in patch-aligned conditioning images. Both geometry and texture SR models are trained for 2 epochs, with 2 objects per batch and up to 3 patches per object, requiring approximately 50 GPU-hours per model. During inference, we encode the coarse object at initial resolution 640^3 and progressively super-resolve it to resolutions of 2560^3 by conducting twice the SR operation. Besides, to use a user-specified image as the condition for inference, our method uses MapAnything [23] to predict the camera parameters relative to the coarse object. All experiments are conducted on a single NVIDIA A100 GPU with 80 GB memory and implemented using PyTorch. Refer to the supplementary material for runtime and memory on average use case.

Datasets and Evaluation Metrics. We evaluate our method under two complementary evaluation settings: a pseudo-ground-truth test set for objective metrics, and an open-world challenging test set for VLM and human evaluation.

Pseudo-GT evaluation. Following the evaluation protocol of TRELLIS.2, we collect around 90 complex and detail-rich 3D assets from the Sketchfab staff-picked list [44] as pseudo ground truth. For each asset, we render the original high-resolution object as the image condition, and construct the low-resolution input by remeshing the geometry and baking low-resolution textures in Blender 5.1.1 [4], removing high-frequency geometry and texture details. This setting allows us to evaluate generative refinement ability using both 2D render-space and 3D geometry-space metrics. For render-space perceptual similarity, realism, and detail richness, we render each generated asset from four preset views and compute LPIPS [59], FID [16], and the High Frequency Ratio (HFR), defined as $\text{HFR}(X) = \frac{\text{mean}(|X - \text{Gaussian}(X)|)}{\text{mean}(|X|) + \epsilon}$. For 3D geometric fidelity, we report Chamfer Distance (CD) and F1 Score in normalized space with a threshold of 10^{-6}. Point clouds for 3D metrics are sampled only from visible outer surfaces, obtained by unprojecting depth maps, with 1M points per asset.

Open-world evaluation. We further assemble an open-world test set containing around 70 challenging objects, *e.g.*, characters, animals, mechanical assets, props, and miniature scenes. Each sample includes a reference image depicting a challenging object and a low-resolution textured mesh generated by a pretrained 3D generator [53,61]. Since no ground-truth high-resolution mesh is available in this setting, following GPTEval3D [52], we adopt a VLM-based pairwise evaluation using GPT-5.2 [34]. The VLM refers to the reference image and a pair of method outputs rendered at 1024^2 resolution, using normal renders for geometry evaluation and colored renders for texture evaluation. It compares the outputs along four aspects: detail richness (*Detail Rich*), alignment with the reference image (*Align w/ Ref.*), artifact-freeness (*Artifact Free*), and overall quality (*Overall*

Table 1: Objective pseudo-GT evaluation. CD/HFR are scaled as indicated.

Obj. Geometry Metrics (CD$\times 10^3$, HFR$\times 10^2$)

Method	LPIPS↓	FID↓	HFR↑	CD↓	F1↑
TR.2(1536)	0.122	60.37	0.981	0.0247	0.059
TR.2(2560)	0.122	63.19	0.923	0.0225	0.036
UltraShape	0.167	72.03	0.876	1.3982	0.109
DetailGen3D	0.187	93.35	0.645	0.4600	0.083
Ours(2560)	**0.109**	**55.89**	**1.088**	**0.0139**	**0.145**

Obj. Texture Metrics (HFR$\times 10^2$)

Method	LPIPS↓	FID↓	HFR↑
TR.2(1536)	0.150	70.20	1.501
TR.2(2560)	0.158	72.77	1.482
MVPaint	0.154	77.18	1.052
Ours(2560)	**0.125**	**59.89**	**1.646**

Table 2: Open-world evaluation. ELO [13] ratings for geometry and texture performance are reported in separate tables for clarity.

Geometry ELO ↑

Method	Detail Rich	Align w/ Ref.	Artifact Free	**Overall Quality**
TR.2(1536)	1775	1774	**1736**	1790
TR.2(2560)	1627	1592	1492	1572
UltraShape	1246	1385	1551	1356
DetailGen3D	869	927	1111	925
Ours(2560)	**1982**	**1821**	1608	**1855**

Texture ELO ↑

Method	Detail Rich	Align w/ Ref.	Artifact Free	**Overall Quality**
TR.2(1536)	1553	**1621**	**1624**	1599
TR.2(2560)	1494	1426	1441	1416
MVPaint	1197	1369	1358	1350
Ours(2560)	**1756**	1582	1575	**1633**

Quality). Each pairwise comparison is repeated three times to reduce stochasticity, and final rankings are computed using an ELO [13] estimator. Details on data list/identifiers, VLM prompts and additional human evaluations on the same dimensions are included in the supplementary materials.

Baselines. We compare with TRELLIS.2 [53], UltraShape [21], DetailGen3D [12], and MVPaint UVR [9]. SuperCarver [58] and LATTICE [24] are not included because their official code/checkpoints are unavailable, making controlled comparisons infeasible. We instead include UltraShape as an open-source implementation of LATTICE for geometry refinement. For TRELLIS.2, we use its self-cascading refinement pipeline and evaluate it at both its default resolution (1536^3) and our output resolution (2560^3). Both TRELLIS.2 and UltraShape initialize their voxel latent support from the input coarse mesh. DetailGen3D serves as an image-guided geometry detailization baseline, while MVPaint UVR is used for texture refinement. Geometry- or texture-only baselines are evaluated only on applicable metrics.

4.2 Results

Qualitative Comparison. We provide the visual comparisons between our method and two of the most competitive baselines (*i.e.*, TRELLIS.2 and UltraShape) in Fig. 4. Remaining baselines are compared in the supplementary materials. The results clearly demonstrate that our method consistently reconstructs sharper fine-grained geometry and richer surface appearance, while maintaining global coherence and avoiding patch seams.

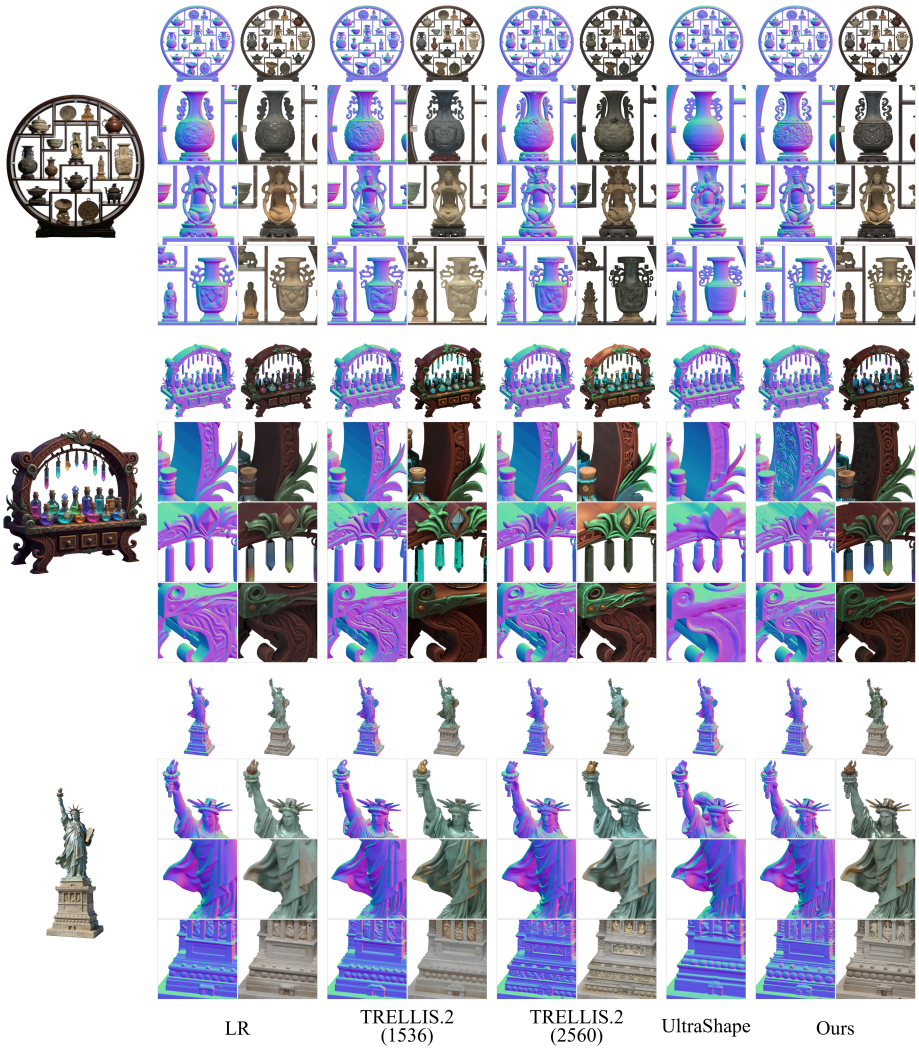

Fig. 4: Qualitative comparison on complex shape. Our method recovers fine details and avoids seams, while baselines show limited detail refinement ability. Zoom-in recommended.

Quantitative Comparison. On the pseudo-GT test set (Tab. 1), our method achieves the best results across all reported geometry and texture metrics. The higher HFR, together with lower LPIPS and FID, suggests that our method recovers perceptually meaningful high-frequency details rather than merely amplifying noise. The lower CD and higher F1 further indicate better fidelity to the pseudo-GT geometry.

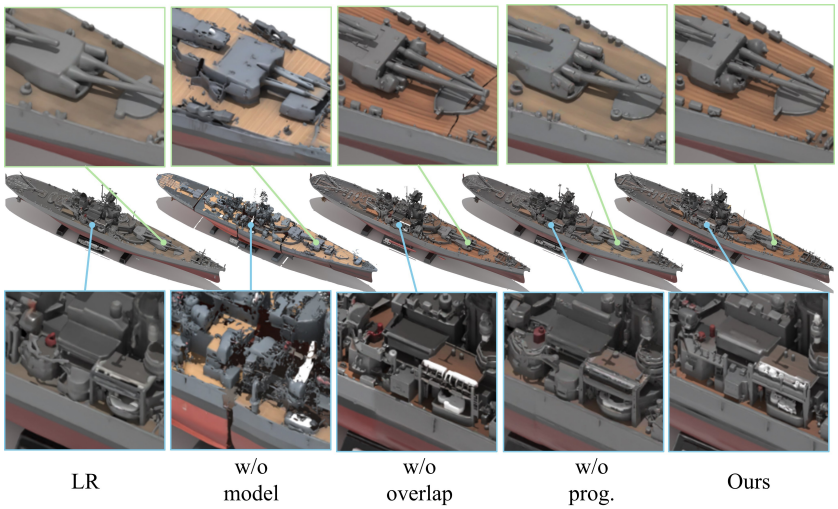

Fig. 5: Qualitative ablation study. Given LR cues as anchors, our trained model suppresses most seams even without overlaps, and with overlapping applied, the seams are fully removed.

Table 3: Quantitative ablation study. We compare our method in a patch-based inference setting with different alternatives (*i.e.*, TRELLIS.2 base model with our pipeline (w/o model), our method without overlapped inference (w/o overlap), our method without progressive SR (w/o prog.).

Ablation Geometry ELO ↑					**Ablation Texture ELO ↑**				
Method	Detail Rich	Align w/ Ref.	Artifact Free	**Overall Quality**	Method	Detail Rich	Align w/ Ref.	Artifact Free	**Overall Quality**
w/o model	1291	1281	1209	1281	w/o model	1288	1270	1269	1271
w/o overlap	<u>1611</u>	<u>1559</u>	1550	<u>1574</u>	w/o overlap	<u>1624</u>	<u>1606</u>	1537	<u>1609</u>
w/o prog.	1424	1552	**1665**	1537	w/o prog.	1441	1509	<u>1595</u>	1475
Ours	**1672**	**1606**	<u>1575</u>	**1605**	Ours	**1645**	**1613**	**1597**	**1643**

Table 2 further reports VLM-based ELO ratings on the open-world test set. Our method achieves the highest scores in detail richness, image alignment, and overall quality for geometry refinement, and in detail richness and overall quality for texture refinement. It also ranks second on the remaining criteria, including artifact-freeness and texture alignment. These results show that our local super-resolution not only scales textured meshes to a substantially higher resolution, *i.e.*, $\times 2.5$ of the TRELLIS.2 base resolution, but also does so in a perceptually meaningful and image-faithful way.

We also observe that directly increasing the TRELLIS.2 resolution to 2560^3 leads to degraded performance, likely due to limited generalization beyond its training resolution, further highlighting the value of progressive local super-resolution for extreme-resolution refinement.

4.3 Ablation Study

We conduct ablations via VLM-based ratings to verify the effectiveness of each component by comparing our full method with three variants. The first variant uses the original TRELLIS.2 flow model without our modification and training for local SR prediction (w/o model). The second variant conducts single-stage refinement without multi-round SR (w/o progressive). The third removes overlaps between local patches to evaluate the effectiveness of our synchronized denoising pipeline (w/o overlap). Both the qualitative results in Fig. 5 and quantitative results in Tab. 3 demonstrate that our local SR model achieved robust refinement based on LR anchors and inference-time overlapping further suppresses seam artifacts. Moreover, non-progressive refinement achieved limited detailization, showing that the progressive refinement helps enhance multi-scale details.

Effect of conditioning views. We further evaluate the effect of using multiple conditioning views on the pseudo-GT test set. Increasing the number of conditioning views from one to four improves normal/color LPIPS from 0.109/0.125 to 0.103/0.119, showing that additional views provide better coverage and reduce ambiguity in local refinement. Together with the qualitative ablation in the supplement, these results suggest that reference images provide useful instance-specific cues complementary to the LR 3D latent, while regions with limited visibility still rely more on the LR latent condition.

5 Limitation

Our method relies on overall accurate camera parameters and conditioning image quality. Severe misalignment or unrealistic conditioning images may introduce inconsistencies. Additionally, the coarse mesh must be structurally adequate, since our SR model is not designed to correct large geometric errors. Also, transfer to a different sparse voxel latent representation requires retraining the adapter.

6 Conclusion

We presented a resource-efficient, progressive and localized SR framework for high-resolution 3D asset generation without high-resolution retraining or multi-GPU compute. By using structured sparse latents as a compact intermediate representation, PLSR decomposes the global mesh super-resolution into local conditional denoising tasks and progressively synthesizes fine geometry and texture details. Experiments demonstrate consistent improvements over strong baselines in geometric and texture detail, suggesting that localized latent SR is a practical direction for scalable high-resolution 3D asset generation and refinement.

Acknowledgements.

This research was supported in part by the Australian Government through the Australian Research Council (ARC) Discovery Project DP260100218.

References

1. Bar-Tal, O., Yariv, L., Lipman, Y., Dekel, T.: Multidiffusion: fusing diffusion paths for controlled image generation. In: Proceedings of the 40th International Conference on Machine Learning. pp. 1737–1752 (2023)
2. 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)
3. Blender Foundation: Blender 3.3. `https://developer.blender.org/docs/release_notes/3.3/` (2022), version 3.3 LTS, accessed June 2026
4. Blender Foundation: Blender 5.1.1. `https://developer.blender.org/docs/release_notes/5.1/corrective_releases/` (2026), version 5.1.1, accessed June 2026
5. Chan, E.R., Lin, C.Z., Chan, M.A., Nagano, K., Pan, B., De Mello, S., Gallo, O., Guibas, L.J., Tremblay, J., Khamis, S., et al.: Efficient geometry-aware 3d generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 16123–16133 (2022)
6. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, 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)
7. Chen, Q., Chen, Z., Kim, V.G., Aigerman, N., Zhang, H., Chaudhuri, S.: Decollage: 3d detailization by controllable, localized, and learned geometry enhancement. In: European Conference on Computer Vision. pp. 110–127. Springer (2024)
8. Chen, Y.T., Liao, T.H., Guo, P., Schwing, A., Huang, J.B.: Bridging diffusion models and 3d representations: A 3d consistent super-resolution framework. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13481–13490 (2025)
9. Cheng, W., Mu, J., Zeng, X., Chen, X., Pang, A., Zhang, C., Wang, Z., Fu, B., Yu, G., Liu, Z., et al.: Mvpaint: Synchronized multi-view diffusion for painting anything 3d. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 585–594 (2025)
10. Deitke, M., Liu, R., Wallingford, M., Ngo, H., Michel, O., Kusupati, A., Fan, A., Laforte, C., Voleti, V., Gadre, S.Y., et al.: Objaverse-xl: A universe of 10m+ 3d objects. Advances in Neural Information Processing Systems **36**, 35799–35813 (2023)
11. Deitke, M., Schwenk, D., Salvador, J., Weihs, L., Michel, O., VanderBilt, E., Schmidt, L., Ehsani, K., Kembhavi, A., Farhadi, A.: Objaverse: A universe of annotated 3d objects. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13142–13153 (2023)
12. Deng, K., Guo, Y.C., Sun, J., Zou, Z.X., Li, Y., Cai, X., Cao, Y.P., Liu, Y., Liang, D.: Detailgen3d: Generative 3d geometry enhancement via data-dependent flow. In: 2026 International Conference on 3D Vision (3DV). pp. 1–31. IEEE (2026)
13. Elo, A.E.: The proposed uscf rating system, its development, theory, and applications. Chess life **22**(8), 242–247 (1967)
14. Gao, W., Aigerman, N., Groueix, T., Kim, V., Hanocka, R.: Textdeformer: Geometry manipulation using text guidance. In: ACM SIGGRAPH 2023 conference proceedings. pp. 1–11 (2023)
15. Han, Y., Yu, T., Yu, X., Xu, D., Zheng, B., Dai, Z., Yang, C., Wang, Y., Dai, Q.: Super-nerf: View-consistent detail generation for nerf super-resolution. IEEE Transactions on Visualization and Computer Graphics **31**(9), 6053–6066 (2024)

16. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local nash equilibrium. Advances in neural information processing systems **30** (2017)
17. Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: Lrm: Large reconstruction model for single image to 3d. In: International Conference on Learning Representations. vol. 2024, pp. 50678–50702 (2024)
18. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W., et al.: Lora: Low-rank adaptation of large language models. In: International Conference on Learning Representations (2022)
19. Huang, Y., Zou, S., Liu, X., Xu, K.: Part-aware shape generation with latent 3d diffusion of neural voxel fields. IEEE Transactions on Visualization and Computer Graphics **31**(10), 8057–8069 (2025)
20. Jain, A., Mildenhall, B., Barron, J.T., Abbeel, P., Poole, B.: Zero-shot text-guided object generation with dream fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 867–876 (2022)
21. Jia, T., Yan, D., Hao, D., Li, Y., Zhang, K., He, X., Li, L., Chen, J., Jiang, L., Yin, Q., Quan, L., Chen, Y.C., Yuan, L.: Ultrashape 1.0: High-fidelity 3d shape generation via scalable geometric refinement. arxiv preprint arXiv:2512.21185 (2025)
22. Kalischek, N., Peters, T., Wegner, J.D., Schindler, K.: Tetradiffusion: Tetrahedral diffusion models for 3d shape generation. In: European Conference on Computer Vision. pp. 357–373. Springer (2024)
23. Keetha, N., Müller, N., Schönberger, J., Porzi, L., Zhang, Y., Fischer, T., Knapitsch, A., Zauss, D., Weber, E., Antunes, N., et al.: Mapanything: Universal feed-forward metric 3d reconstruction; map-anything. github. io. In: 2026 International Conference on 3D Vision (3DV). pp. 499–509. IEEE (2026)
24. Lai, Z., Zhao, Y., Zhao, Z., Liu, H., Lin, Q., Huang, J., Guo, C., Yue, X.: Lattice: Democratize high-fidelity 3d generation at scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19982–19992 (2026)
25. Li, J., Tan, H., Zhang, K., Xu, Z., Luan, F., Xu, Y., Hong, Y., Sunkavalli, K., Shakhnarovich, G., Bi, S.: Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. In: International conference on learning representations. vol. 2024, pp. 21896–21920 (2024)
26. Li, P., Liu, Y., Long, X., Zhang, F., Lin, C., Li, M., Qi, X., Zhang, S., Xue, W., Luo, W., et al.: Era3d: High-resolution multiview diffusion using efficient row-wise attention. Advances in Neural Information Processing Systems **37**, 55975–56000 (2024)
27. Lipman, Y., Chen, R.T., Ben-Hamu, H., Nickel, M., Le, M.: Flow matching for generative modeling. In: International Conference on Learning Representations (2023)
28. Liu, X., Gong, C., Liu, Q.: Flow straight and fast: Learning to generate and transfer data with rectified flow. In: International Conference on Learning Representations (2023)
29. Liu, Y., Lin, C., Zeng, Z., Long, X., Liu, L., Komura, T., Wang, W.: Syncdreamer: Generating multiview-consistent images from a single-view image. In: International conference on learning representations. vol. 2024, pp. 27676–27697 (2024)
30. Liu, Y., Xie, M., Liu, H., Wong, T.T.: Text-guided texturing by synchronized multi-view diffusion. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)
31. Long, X., Guo, Y.C., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., et al.: Wonder3d: Single image to 3d using cross-

domain diffusion. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 9970–9980 (2024)
32. Mao, A., Du, Z., Hou, J., Duan, Y., Liu, Y.j., He, Y.: Pu-flow: A point cloud upsampling network with normalizing flows. IEEE Transactions on Visualization and Computer Graphics **29**(12), 4964–4977 (2022)
33. Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13492–13502 (2022)
34. OpenAI: GPT-5.2 Model. `https://developers.openai.com/api/docs/models/gpt-5.2` (2025), model version: gpt-5.2-2025-12-11; accessed June 2026
35. Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. In: International Conference on Learning Representations (2023)
36. Qian, Y., Hou, J., Kwong, S., He, Y.: Pugeo-net: A geometry-centric network for 3d point cloud upsampling. In: European conference on computer vision. pp. 752–769. Springer (2020)
37. Qian, Y., Hou, J., Kwong, S., He, Y.: Deep magnification-flexible upsampling over 3d point clouds. IEEE Transactions on Image Processing **30**, 8354–8367 (2021)
38. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PmLR (2021)
39. Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4209–4219 (2024)
40. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
41. Schuhmann, C.: improved-aesthetic-predictor: Clip+mlp aesthetic score predictor. `https://github.com/christophschuhmann/improved-aesthetic-predictor` (2026), `https://github.com/christophschuhmann/improved-aesthetic-predictor`, gitHub repository, commit or version should be specified if a precise snapshot is required
42. Shi, R., Chen, H., Zhang, Z., Liu, M., Xu, C., Wei, X., Chen, L., Zeng, C., Su, H.: Zero123++: a single image to consistent multi-view diffusion base model. arXiv preprint arXiv:2310.15110 (2023)
43. Shim, J., Kang, C., Joo, K.: Diffusion-based signed distance fields for 3d shape generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20887–20897 (2023)
44. Sketchfab: Staff picks. `https://sketchfab.com/3d-models/staffpicks` (2026), accessed June 2026
45. Tang, J., Chen, Z., Chen, X., Wang, T., Zeng, G., Liu, Z.: Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In: European Conference on Computer Vision. pp. 1–18. Springer (2024)
46. Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. In: International Conference on Learning Representations. vol. 2024, pp. 33879–33896 (2024)
47. Wang, Y., Xu, X., Ma, L., Wang, H., Dai, B.: Boosting 3d object generation through pbr materials. In: SIGGRAPH Asia 2024 Conference Papers. pp. 1–11 (2024)

48. Wang, Z., Lu, C., Wang, Y., Bao, F., Li, C., Su, H., Zhu, J.: Prolificdreamer: High-fidelity and diverse text-to-3d generation with variational score distillation. Advances in neural information processing systems **36**, 8406–8441 (2023)
49. Wang, Z., Wang, Y., Chen, Y., Xiang, C., Chen, S., Yu, D., Li, C., Su, H., Zhu, J.: Crm: Single image to 3d textured mesh with convolutional reconstruction model. In: European conference on computer vision. pp. 57–74. Springer (2024)
50. Wu, K., Liu, F., Cai, Z., Yan, R., Wang, H., Hu, Y., Duan, Y., Ma, K.: Unique3d: High-quality and efficient 3d mesh generation from a single image. Advances in Neural Information Processing Systems **37**, 125116–125141 (2024)
51. Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. Advances in Neural Information Processing Systems **37**, 121859–121881 (2024)
52. Wu, T., Yang, G., Li, Z., Zhang, K., Liu, Z., Guibas, L., Lin, D., Wetzstein, G.: Gpt-4v (ision) is a human-aligned evaluator for text-to-3d generation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 22227–22238 (2024)
53. Xiang, J., Chen, X., Xu, S., Wang, R., Lv, Z., Deng, Y., Zhu, H., Dong, Y., Zhao, H., Yuan, N.J., et al.: Native and compact structured latents for 3d generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14419–14429 (2026)
54. Xiang, J., Lv, Z., Xu, S., Deng, Y., Wang, R., Zhang, B., Chen, D., Tong, X., Yang, J.: Structured 3d latents for scalable and versatile 3d generation. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 21469–21480 (2025)
55. Xu, J., Cheng, W., Gao, Y., Wang, X., Gao, S., Shan, Y.: Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. arXiv preprint arXiv:2404.07191 (2024)
56. Yu, L., Li, X., Fu, C.W., Cohen-Or, D., Heng, P.A.: Pu-net: Point cloud upsampling network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2790–2799 (2018)
57. Zhang, L., Wang, Z., Zhang, Q., Qiu, Q., Pang, A., Jiang, H., Yang, W., Xu, L., Yu, J.: Clay: A controllable large-scale generative model for creating high-quality 3d assets. ACM Transactions on Graphics (TOG) **43**(4), 1–20 (2024)
58. Zhang, Q., Jian, X., Zhang, X., Wang, W., Hou, J.: Supercarver: Texture-consistent 3d geometry super-resolution for high-fidelity surface detail generation. IEEE Transactions on Visualization and Computer Graphics (2026)
59. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 586–595 (2018)
60. Zhang, W., Yue, Y., Pan, H., Chen, Z., Wang, C., Pfister, H., Wang, W.: Marching windows: Scalable mesh generation for volumetric data with multiple materials. IEEE transactions on visualization and computer graphics **30**(3), 1728–1742 (2022)
61. Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., et al.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. arXiv preprint arXiv:2501.12202 (2025)