SPGen: Spherical Projection as Consistent and Flexible Representation for Single Image 3D Shape Generation

JINGDONG ZHANG, Texas A&M University, USA
WEIKAI CHEN*, LightSpeed Studios, USA
YUAN LIU, Hong Kong University of Science and Technology, China
JIONGHAO WANG, Texas A&M University, USA
ZHENGMING YU, Texas A&M University, USA
ZHUOWEN SHEN, Texas A&M University, USA
BO YANG, Waymo, USA
WENPING WANG, Texas A&M University, USA
XIN LI, Texas A&M University, USA

Fig. 1. We present SPGen, a powerful generative model creating consistent 3D shapes with flexible topology from single view images in seconds.

*Corresponding author.

Authors' Contact Information: Jingdong Zhang, Texas A&M University, College Station, Texas, USA, jdzhang@tamu.edu; Weikai Chen, LightSpeed Studios, USA, weikaichen@global.tencent.com; Yuan Liu, Hong Kong University of Science and Technology, Hong Kong, China, yuanly@ust.hk; Jionghao Wang, Texas A&M University, College Station, Texas, USA, jionghao@tamu.edu; Zhengming Yu, Texas A&M University, College Station, Texas, USA, yuzhengming@tamu.edu; Zhuowen Shen, Texas A&M University, College Station, Texas, USA, mickshen@tamu.edu; Bo Yang, Waymo, USA, yangbo@waymo.com; Wenping Wang, Texas A&M University, College Station, Texas, USA, wenping@tamu.edu; Xin Li, Texas A&M University, College Station, Texas, USA, xinli@tamu.edu.

SA Conference Papers '25, Hong Kong, Hong Kong

Existing single-view 3D generative models typically adopt multiview diffusion priors to reconstruct object surfaces, yet they remain prone to inter-view inconsistencies and are unable to faithfully represent complex internal structure or nontrivial topologies. In particular, we encode geometry information by projecting it onto a bounding sphere and unwrapping it into a compact and structural multi-layer 2D Spherical Projection (SP) representation. Operating solely in the image domain, SPGen offers three key advantages simultaneously: (1) *Consistency.* The injective SP mapping encodes surface geometry with a single viewpoint which naturally eliminates view inconsistency and ambiguity; (2) *Flexibility.* Multi-layer SP maps represent nested

ACM ISBN 979-8-4007-2137-3/2025/12
https://doi.org/10.1145/3757377.3763959

internal structures and support direct lifting to watertight or open 3D surfaces; (3) *Efficiency.* The image-domain formulation allows the direct inheritance of powerful 2D diffusion priors and enables efficient finetuning with limited computational resources. Extensive experiments demonstrate that SPGen significantly outperforms existing baselines in geometric quality and computational efficiency.

Additional Key Words and Phrases: 3D Shape Generation, Spherical Projection, Representation Learning, Stable Diffusion

ACM Reference Format:
Jingdong Zhang, Weikai Chen, Yuan Liu, Jionghao Wang, Zhengming Yu, Zhuowen Shen, Bo Yang, Wenping Wang, and Xin Li. 2025. SPGen: Spherical Projection as Consistent and Flexible Representation for Single Image 3D Shape Generation. In *SIGGRAPH Asia 2025 Conference Papers (SA Conference Papers '25), December 15–18, 2025, Hong Kong, Hong Kong*. ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3757377.3763959

1 Introduction

High-quality 3D asset generation is critical for applications spanning from AR/VR, robotics, industrial design to digital content creation. Conventional modeling workflows are often manual, time-consuming, and require specialized expertise. Recent progress in deep generative models [Goodfellow et al. 2014; Ho et al. 2020; Kingma 2013; Rombach et al. 2022; Van Den Oord et al. 2017] has substantially advanced the automation and accessibility of this process. By harnessing large-scale datasets and strong priors from pretrained networks, these models[Chen et al. 2024c; Hong et al. 2023; Lin et al. 2023; Liu et al. 2023b; Long et al. 2024; Zhang et al. 2024a] enable the synthesis of high-fidelity, semantically meaningful 3D geometry from limited inputs, such as a single image or text, thus facilitating scalable and efficient 3D content creation.

Existing 3D generative methods can be broadly categorized based on their intermediate representations. Geometry-based methods [Alliegro et al. 2023; Chen et al. 2024a,c; Hao et al. 2024; Hong et al. 2023; Li et al. 2023b; Siddiqui et al. 2024; Tochilkin et al. 2024; Wu et al. 2024b; Yu et al. 2024; Zhang et al. 2024a] directly synthesize 3D structures such as point clouds, signed distance fields (SDFs), or explicit mesh faces by using diffusion models, causal transformers, or large reconstruction networks. While effective, unlike the abundance of image or language data, these methods are constrained by the scarcity and noisiness of 3D data, which hinders scalability and requires heavy data preprocessing.

To mitigate these challenges, image-based approaches [Elizarov et al. 2024; Liu et al. 2023b,a; Long et al. 2024; Richter and Roth 2018; Xu et al. 2024a; Yan et al. 2024b; Zhang et al. 2018] generate 3D content via intermediate multiview images, geometry image, uv atlas or spherical projection, from which geometry is recovered using differentiable rendering, feed-forward networks, etc. Though these approaches can leverage powerful pretrained 2D priors, they are suffering from several limitations respectively, multiview images are usually lack of strict view consistency and geometric coherence, geometry images and uv atlas are limited by non-unique cuttings and mappings, which burden the model with extensive boundary stitching and hamper scalable training, while simple spherical projection is suffering from severe self-occlusions. These issues either degrade the qualities of restored geometry or limit scalable training on large-scale datasets.

To address the limitations of existing 3D generative methods, we propose SPGen, a novel scalable framework that generates high-quality meshes based on *multi-layer Spherical Projection* (SP) maps. Concretely, given a normalized 3D object, we enclose it within a unit sphere and cast rays from the origin outward through each point on the spherical surface. For each ray, the intersection information, such as depth, is recorded at the corresponding point on the sphere. The sphere is subsequently projected onto a 2D image plane, forming what we term a Spherical Projection (SP) map. For general objects exhibiting self-occlusion or nested internal geometry, we trace multiple intersections along each ray and store them sequentially in multi-layer SP maps, thereby capturing fine-grained spatial structure beyond the external surface. This SP-based formulation underpins our image-centric generation pipeline and confers three key advantages simultaneously. (1) *Consistency.* The SP map is a naturally view-consistent representation encoding 360-degree geometry. The mapping of valid pixel to surface point is an injective function, which eliminates potential view conflicts. (2) *Flexibility.* Multi-layer SP maps enables faithful representation of geometries with varying resolution and topology. Notably, both watertight and open surfaces, as well as layered internal structures, can be directly reconstructed from multi-layer SP maps. (3) *Efficiency.* As SP maps are structured 2D representations, we can finetune powerful pretrained diffusion backbones such as SDXL [Podell et al. 2023] with limited resource consumptions, and meanwhile inherit strong prior knowledge including locality, semantic, implicit symmetry and repetition patterns instead of learning from scratch.

Moreover, we introduce tailored components to address the challenges when applying general generative pipelines specifically on SP maps. We observe that errors of SP maps during training mainly gather at geometric boundaries, thus we propose to adopt a composition of geometry regularization at the boundaries to enhance the SP map qualities. For multi-layer SP generation, we leverage layer-wise self-attention to enforce alignment between interior and exterior layers. After denoising, we perform unprojections from SP maps to 3D spaces to obtain a dense 3D point cloud, followed by mesh extraction via a lightweight feed-forward network. SPGen can generate 3D meshes with high geometric quality in seconds. Experimental results demonstrate that SPGen achieves superior performance compared to prior methods, despite requiring significantly less training overhead.

In summary, our contributions are three-fold:

- We propose to use multi-layer Spherical Projection (SP) maps as compact and structural representations for 3D shape generation. The SP maps encode the whole surface geometry through injective mappings, enabling view-consistent and topology-flexible geometry reconstruction.
- We present a novel generation pipeline SPGen. By efficiently finetuning the powerful image diffusion model with limited resources to inherit rich prior knowledge, and incorporating specifically designed geometry regularization and layer-wise self-attention, SPGen could generate high-quality 3D meshes in seconds via single image conditioning.
- Our proposed method achieves state-of-the-art performance on public benchmarks with significantly lower training overhead, indicating the effectiveness and efficiency of SPGen.

2 Related Works

2.1 3D Shape Creation with Geometric Representations

Creating high-quality 3D shapes from single view input has received a lot of attention recently [Alliegro et al. 2023; Chen et al. 2024a,c; Hao et al. 2024; Hong et al. 2023; Hu et al. 2024; Li et al. 2023b; Lin et al. 2023; Liu et al. 2023b; Long et al. 2024; Poole et al. 2022; Richter and Roth 2018; Siddiqui et al. 2024; Tang et al. 2025, 2023; Tochilkin et al. 2024; Wang et al. 2025a,b,c; Wu et al. 2024a, 2018; Xu et al. 2024a; Yu et al. 2024; Zhang et al. 2024a, 2018]. Based on the representation adopted, these works can be roughly divided into two groups, geometry-based and image-based. The former usually takes point clouds, Signed Distance Fields (SDF) fields, or directly uses mesh faces as surface representations. For point clouds, Point-E [Nichol et al. 2022] tries to denoise point cloud directly by a transformer-based diffusion model, CLAY [Zhang et al. 2024a], Direct3D [Wu et al. 2024b] further enhanced this idea by applying an autoencoder to compress the point clouds and acquire compact and representative latents, which makes it easier to scale-up the model. Taking SDF as a shape representation, SDF-StyleGAN [Zheng et al. 2022] and SDF-3DGAN [Jiang et al. 2023] extend Generative Adversarial Network (GAN) from 2D images to 3D shapes. LAS-Diffusion [Zheng et al. 2023] introduces a two-stage pipeline generating refined SDFs from coarse voxel occupancy fields. SurfD [Yu et al. 2024] proposed to generate Unsigned Distance Fields (UDF) to represent open surfaces. Different from denoising SDF [Li et al. 2023a; Yariv et al. 2024; Zheng et al. 2023], Large Reconstruction Models (LRM) [Hong et al. 2023; Li et al. 2023b; Tochilkin et al. 2024] directly learn SDF from single or multiple input images by feed-forward transformers. Another bunch of works directly take mesh faces as representations, PolyDiff [Alliegro et al. 2023] trains a diffusion model to denoise mesh faces while [Chen et al. 2024a,c; Siddiqui et al. 2024] encodes each face as a token and predicts the next token autoregressively to predict the whole mesh. Despite the achievements of geometry-based methods, these methods have higher requirements for the quality and form of data, for example, autoregressive methods constrain the number of mesh faces and SDF methods need to calculate SDFs for water-tight objects only, which requires more complex data preprocessing processes and limits the scaling-up of these methods.

2.2 3D Shape Creation from Image-based Representations

To solve the shortcomings of geometry-based representations, and utilize the strong priors stored in powerful 2D generative models trained with billions of data, another bunch of works takes images as the representation to record the geometry. Matryoshka Network [Richter and Roth 2018] proposes to mark the space between each "entry–exit" depth pair in its six axis-aligned stacks as occupied, fuses the results into a voxel volume to restore both external and internal of an object completely, while Genre [Zhang et al. 2018] represents the outer surface by a single-layer spherical depth map, and takes the reconstruction from single image as a spherical map inpainting task. However, their restored geometry qualities are limited by the resolution bottleneck of voxel grids during surface extraction. Different from them, another bunch of works take advantage of geometry images [Elizarov et al. 2024; Gu et al. 2002] or uv atlas [Yan et al. 2024b] to unfold the surface geometries to image charts, these representations require non-unique cuttings and mappings, which burden the model with extensive boundary stitching and hamper scalable dataset preparation.

Recently, multiview-based representations are also quickly thriving, DreamFusion [Poole et al. 2022] proposes Score Distillation Sampling (SDS) to distill from image diffusion models and extract surface geometry by differentiable rendering [Guédon and Lepetit 2024; Huang et al. 2024; Kerbl et al. 2023; Laine et al. 2020; Mildenhall et al. 2021; Shen et al. 2024; Wang et al. 2021; Yariv et al. 2021] without training on 3D datasets, and [Lin et al. 2023; Tang et al. 2023] further improves the results. However, these methods are computationally expensive and usually suffer from blurred details. Some works focus on generating multi-view consistent images from one input image and therefore reconstruct the geometry [Liu et al. 2024, 2023b,a; Long et al. 2024; Shi et al. 2023; Xu et al. 2024a], among them, Zero123 [Liu et al. 2023b] firstly tries to incorporate camera as conditions and SyncDreamer [Liu et al. 2023a] introduces spatial attention to align views and extract surfaces by [Wang et al. 2021]. Wonder3D [Long et al. 2024] enhances the performance by introducing cross-domain diffusion for both RGB images and normal maps, and Zero-1-to-G [Meng et al. 2025] extends it to incorporate multiple Gaussian attributes. [Tang et al. 2025; Wang et al. 2025c; Wu et al. 2024a; Xu et al. 2024a,b] are combinations of both multiview generation process and feed-forward reconstruction process, while CRM [Wang et al. 2025c] focuses on the strong connections between canonical views and triplane features, [Tang et al. 2025; Xu et al. 2024b] focus on feed-forward Gaussians reconstruction, and [Wu et al. 2024a] focuses on multi-view depth generation. However, these methods mentioned above heavily rely on the quality of multi-view image generation, and strict constraints of view consistency are not guaranteed by solely applying cross-attention as a soft constraint, consequently degrading the geometry quality when inconsistencies occur. Besides, multi-view images fail to represent objects with severe self-occlusions or internal layers. Differently, we propose to use multi-layer Spherical Projection to record the whole object geometry and serve as a consistent and coherent representation.

3 Methodology

The detailed design of our proposed SPGen pipeline is shown in Fig 2. Firstly, we extract the SP maps from object meshes. After the SP maps are prepared, we finetune the image AutoEncoder on SP maps to obtain compact latents. Then we finetune the latent diffusion model to generate multi-layer SP maps. Finally, the SP maps are used to reconstruct high-quality shapes by Poisson reconstruction or UDF reconstruction to represent water-tight or open surfaces.

3.1 Spherical Projection

Current methods taking advantage of pretrained image generative models usually adopt multiview images as representations, since multiview images comprehensively cover the exterior of the object and project surface points to 2D domain. However, these projections are not simple injective functions due to the overlapping of adjacent views, which consequently leads to ambiguity during the generation process. Since the object surface is usually a complex 2D manifold, we try to find an injective projection that maps it

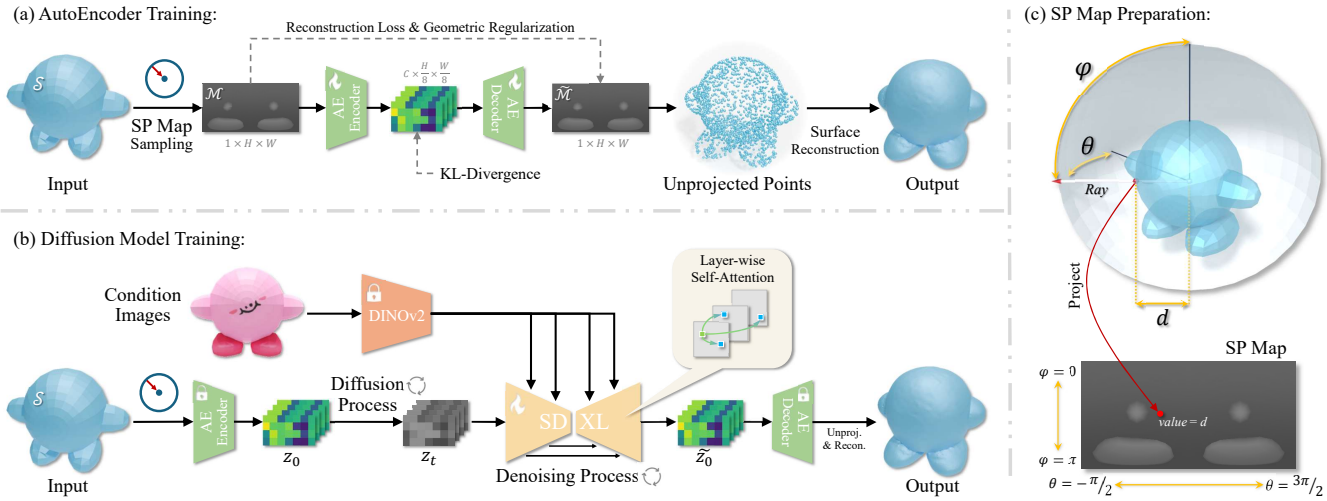

Fig. 2. **Illustration of SPGen.** (a) The AutoEncoder (AE) Training Pipeline: we train the AutoEncoder on projected Spherical Projection (SP) maps. We introduce geometry regularization to aid the reconstruction process, and KL divergence is applied to regularize the latent distribution. After the SP map is reconstructed, we unproject points into 3D space and extract the surface. (b) The Diffusion Model Training Pipeline: we use the finetuned AutoEncoder to produce latent and train the denoise UNet with conditioning image embeddings from DINOv2, except from the standard self-attention guiding the diffusion process, we also introduce layer-wise self-attention among multi-layer SP maps. The denoised latent is fed into AE decoder and produces the final mesh output similarly. (c) The detailed preparation of one layer of SP Map: the object mesh is normalized and placed at the center of a union sphere. We cast rays from the origin and record the depth value d on the SP map parameterized by azimuth angle θ and polar angle φ.

to a structural 2D domain that could be handled by image generative models. Therefore, we propose Spherical Projection (SP) as the shape representation.

Adopting SP maps as a panorama for scene-level generation is a common solution [Feng et al. 2023; Hara et al. 2022; Lu et al. 2024; Wang et al. 2023; Yan et al. 2024a]. Differently, we are introducing SP maps as the geometry representations for shape generation. As shown in Fig. 2 (c), we firstly cast a ray from the origin along the radial direction with azimuth angle θ and polar angle φ. When the ray intersects with a surface point $\mathcal{P} \in \mathbb{R}^3$, we calculate the distance that the ray travels as $d = \|\mathcal{P}\|_2$. In this simple way, we can use equirectangular projection [Hara et al. 2022] $\mathrm{F}(\mathcal{P}) : \mathbb{R}^3 \rightarrow \mathbb{R}^2$, to map a 3D point to the 2D domain parameterized by θ and φ. By recording the corresponding d at each point (θ, φ), the geometry is recorded on the SP map. To acquire the original point position, we simply perform the conversion from spherical coordinates to Euclidean coordinates:

$$\mathcal{P} = \mathrm{F}^{-1}(\theta, \phi) = \begin{bmatrix} \sin\phi\cos\theta \\ \sin\phi\sin\theta \\ \cos\phi \end{bmatrix} d. \tag{1}$$

For rays that intersect with the mesh surface more than once, we record all the intersection positions, and project them onto SP maps reversely, i.e., starting from the outermost intersection, until we reach the maximum recording depth or there are no more intersections. In this manner, we not only solve the self-occlusion issue, but also empower the SP maps to represent the inner structures of a complex object. As shown in Algo. 1, assume that we want to prepare k layers of SP maps $\{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^k\}$ for mesh surface $\mathcal{S}$, for each ray $\mathcal{R}_i$, we need to perform ray-mesh intersection penetrating all layers and record the intersection points $\{\mathcal{P}_i^0, \mathcal{P}_i^1, \ldots, \mathcal{P}_i^k\}$.

ALGORITHM 1: Multi-layer SP Map Preparation

Input: The object surface $\mathcal{S}$
Output: k-layer SP maps $\mathfrak{M} = \{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^k\}$ storing depth
Initialize SP map layers as empty;
Initialize rays $\mathfrak{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n\}$, uniformly sample (θ, ϕ);
for *each point* $\mathcal{R}_i \in \mathfrak{R}$ **do**
  $\{\mathcal{P}_i^0, \mathcal{P}_i^1, \ldots, \mathcal{P}_i^k\}$ = **RayMeshIntersection**$(\mathcal{R}_i, \mathcal{S}, k)$;
  $step = 0$;
  for j *in range*(k) **do**
    if P_i^{k-j} *is not NULL* **then**
      $\mathcal{M}^{step} = d = \|P_i^{k-j}\|_2$;
      $step$ += 1;
    end
  end
end

If there is no more intersections after layer j', where $j' \leq k$, then $\mathcal{P}_i^{j'}, \ldots, \mathcal{P}_i^k$ are set as *NULL*. Then we loop through each layer of SP map $\mathcal{M}^j$ reversely and record the depth values $d = \|P_i^{k-j}\|_2$ of valid points on the map.

After the SP maps are prepared, we finetune the AutoEncoder and Diffusion Model to generate 3D shapes based on them, which will be explained in detail in the following sections.

3.2 Generation Pipeline

3.2.1 Preliminaries. First, we are going to introduce our training pipeline. Our generation pipeline is built upon SDXL [Podell et al. 2023], leveraging the strong prior from pretraining. Specifically, this stable diffusion pipeline is composed of a set of Kullback–Leibler

divergence regularized (KL-regularized) Autoencoder $\Psi_{\mathcal{E}}, \Psi_{\mathcal{D}}$ and a large-scale denoising UNet Θ. The AutoEncoder compresses high-resolution input $\mathcal{M}$ to compact latent z_0, and this process is optimized by jointly minimizing reconstruction error and regularizing latent distribution:

$$\begin{aligned} z_0 &\sim \mathcal{Q}\left(z \mid \mathcal{M}\right), \\ L_{recon} &= \mathbb{E}_{\mathcal{M}}\Big[\left\| \mathcal{M} - \Psi_{\mathcal{D}}\left(\Psi_{\mathcal{E}}\left(\mathcal{M}\right)\right) \right\| \Big] \\ &\quad + \lambda \cdot \mathbb{E}_{\mathcal{M}}\Big[D_{\mathrm{KL}}\left(\mathcal{Q}\left(z \mid \mathcal{M}\right) \| \mathcal{N}(0, I)\right) \Big], \end{aligned} \tag{2}$$

where $\mathcal{Q}$ is the output distribution of $\Psi_{\mathcal{E}}$ and λ is the control coefficient of regularization strength. After $\Psi_{\mathcal{E}}, \Psi_{\mathcal{D}}$ are trained, z_0 is generated accordingly and a noise scheduler gradually adds Gaussian noise to it over T time steps:

$$z_t = \sqrt{\alpha_t} z_0 + \sqrt{1-\alpha_t}\epsilon, \quad \epsilon \sim \mathcal{N}(0, I), \tag{3}$$

where z_t is the noisy version at time step $t < T$, and α_t is the noise schedule coefficient. Then the denoise UNet is trained to parameterize the reverse diffusion process by predicting the noise $\epsilon_\Theta(z_t, t)$ of time step t. Finally, the optimization process is achieved by minimizing the mean squared error (MSE) between the predicted noise and the actual noise:

$$L_{diff} = \mathbb{E}_{z_0,\epsilon,t}\Big[\left\| \epsilon - \epsilon_\Theta\left(z_t, t\right) \right\|^2 \Big]. \tag{4}$$

After training, synthesis data is generated by reversing the noise-adding process in latent space, and the denoised latent is fed to $\Psi_{\mathcal{D}}$ to produce final results.

However, since the diffusion model is only trained on RGB data for image synthesis, which mainly focuses on better perceptual quality and lacks the generalization ability on Sp depth maps, solely inference on pretrained pipeline results in poor geometry quality. Thus, we propose to finetune the whole pipeline and we will illustrate the details in the following sections.

3.2.2 Layer-wise Self Attention. Applying attention to associate multiple predicting objectives is a popular technique in recent works [Fu et al. 2025; Long et al. 2024; Shi et al. 2023; Ye and Xu 2022; Zhang et al. 2025]. In our pipeline, we need to constrain the generated layers to have reasonable relative positions in space, and self-intersections or floating artifacts are therefore avoided.

Considering the intermediate hidden states $\{m^1, m^2, \ldots, m^k\}, m^j \in \mathbb{R}^{C\times h\times w}$ generated by UNet parameters Θ that corresponding with SP maps $\{\mathcal{M}^1, \mathcal{M}^2, \ldots, \mathcal{M}^k\}$. We first flatten them by $\mathrm{Flat}(m^j) : \mathbb{R}^{C\times h\times w} \to \mathbb{R}^{C\times(hw)}$ and concatenate them along the spatial dimension:

$$\bar{m} = \mathrm{Concat}\left(\left[\mathrm{Flat}(m^1), \ldots, \mathrm{Flat}(m^k)\right], dim = -1\right), \tag{5}$$

where $\mathrm{Concat}(\cdot)$ represents the concatenation operation. Followingly, we perform the standard self-attention on $\bar{m}$:

$$\mathrm{Attention}\left(Q, K, V\right) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{C_a}}\right) \cdot V, \tag{6}$$

where Q, K and V are produced by linear projections from $\bar{m}$, and C_a represents the projected dimension for attention. In this way, layer relations are modeled, leading to a more accurate geometry during the denoising process.

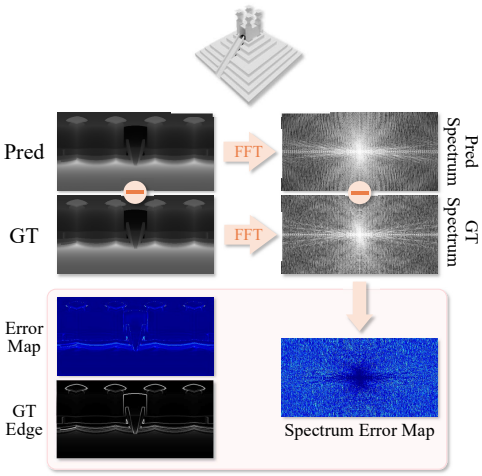

Fig. 3. **Visualization of error distributions.** We visualize the edge map, spectrum map, and corresponding error maps of prediction and groundtruth (GT). The error distribution is highly aligned with the edge distribution, indicating a large amount of error falls on areas with large image gradients. This observation is consistent with the phenomenon in the spectral domain, in which high-frequency component dominates the error distribution.

3.2.3 Geometry Regularization. In Sec. 3.2.1, we analyzed the necessity of fine-tuning the diffusion model. However, simply transplanting the methods of training image diffusion is suboptimal since these methods (such as perceptual loss [Johnson et al. 2016]) do not contribute to better geometry quality. Among the pipeline components, finetuning the AutoEncoder Ψ is the most tricky part, since the input and output are SP maps, which have very different distributions from the original RGB image, on the contrary, the KL regularization forces the latent distribution to be closer to the original standard normal distribution, which poses a great challenge to the training process.

During training, we observed that if we only apply L1 distance as the reconstruction loss, the reconstructed results are not satisfying with blurred details and noisy surfaces. We study the output by analyzing error maps. As shown in Fig. 3, we observe that the error in the spatial domain is concentrated on the edge of the SP Map, which is the high-frequency component of an image. This concentration of error occurs because standard reconstruction losses, like L1 distance, average the error over all pixels. Since edges and details are a small fraction of total pixels, their error has minimal impact on the overall loss. The model therefore prioritizes optimizing large, smooth areas, causing the high-frequency details crucial for accurate geometry to become blurred or misplaced. Thus, we further visualize the spectrum produced by the Fast Fourier Transform (FFT), which consistently shows that the error mainly exists in the four corners of the spectrum, which is where the high-frequency components exist, while the center where the low-frequency components are located is darker, indicating that the error is smaller. This is consistent with the conclusion in [Jiang et al. 2021].

Inspired by this observation, we propose two regularization losses to enhance the geometry quality. First, we directly strengthen supervision of image boundaries, since the pixel L1 loss will tend to punish overall deviation, while the pixel ratio of the edge is too small to be

fully supervised, we extract a hard boundary mask $\mathcal{B}$ with margin by Sobel operator and dilatation operator: $\mathcal{B} = \text{Dilate}\,(\text{Sobel}(\mathcal{M}))$, then we use $\mathcal{B}$ to mask out the boundary pixels and impose greater punishment on them individually:

$$L_{edge} = \mathbb{E}_{\mathcal{M}}\Big[\mu\mathcal{B}\cdot\|\mathcal{M}-\Psi\,(\mathcal{M})\| + (1-\mu)(1-\mathcal{B})\cdot\|\mathcal{M}-\Psi\,(\mathcal{M})\|\Big], \tag{7}$$

where $\Psi = (\Psi_{\mathcal{D}} \circ \Psi_{\mathcal{E}})$, and μ is a control coefficient of the strength, by applying a larger weight μ to the loss calculated within this masked region $\mathcal{B}$, we compel the AutoEncoder to pay closer attention to these critical areas. This targeted penalty prevents the model from smearing details across edges and results in a more precise reconstruction of sharp geometric contours and object silhouettes.

Second, we enhance the high-frequency component from the spectral domain. We first perform Fast Fourier Transform (FFT) to both prediction and groundtruth, denoting as $\mathcal{M}_s = \text{FFT}(\mathcal{M})$, $\tilde{\mathcal{M}}_s = \text{FFT}(\Psi(\mathcal{M}))$, decomposing the SP map into its constituent frequencies. And then impose a high pass filter $\mathcal{H}$ on the spectrum, which is a circular mask that covers the central area where low-frequency components are located, and only allows high-frequency components to compute loss, we separately calculate the L1 distance on principal value of argument and modulus respectively:

$$L_{spec} = \mathbb{E}_{\mathcal{M}}\Big[\mathcal{H}\cdot\Big\|\text{Arg}(\mathcal{M}_s)-\text{Arg}(\tilde{\mathcal{M}}_s)\Big\| + \zeta\mathcal{H}\cdot\Big\|\|\mathcal{M}_s\|_2-\|\tilde{\mathcal{M}}_s\|_2\Big\|\Big], \tag{8}$$

where ζ is also a coefficient. By separately penalizing discrepancies in both the phase and magnitude of these frequencies, the model significantly reduces surface noise and improves the crispness of the final reconstructed geometry. With L_{recon} from Eq. 2, the total loss for training the AutoEncoder can be written as:

$$L = L_{recon} + \alpha L_{edge} + \beta L_{spec}. \tag{9}$$

3.2.4 Surface Extraction. After the SP maps are generated, we unproject points to 3D space by Eq. 1 and extract the surface from the point cloud. For watertight objects, we simply perform Poisson reconstruction. We sample oriented point clouds from groundtruth meshes and use them to train a light-weight 3D-Unet as a normal estimator, which predicts a unit normal vector for each point. On this basis, the gradient field is calculated and the surface is reconstructed. For open surfaces, we follow [Yu et al. 2024], use their pretrained point-cloud-to-UDF AutoEncoder to predict the UDF in space, and the implicit surface is extracted by MeshUDF [Guillard et al. 2022]. The reconstruction process is exceptionally fast while remaining cost-efficient, with high-quality geometry generated from SPGen, we can obtain accurate and detailed meshes suitable for downstream tasks such as editing, rendering, simulation, etc.

4 Experiments

4.1 Experimental Setting

4.1.1 Dataset. We refer to the criteria in [Chen et al. 2024c; Long et al. 2024] to filter the Objaverse dataset [Deitke et al. 2023] by removing low-quality or scene-level meshes and acquire around 160k objects as our whole training split. Before training, we follow [Long et al. 2024] to render the multi-view image for reference. We also picked 1993 objects out of the training indices as our validation split on Objaverse. We normalize the scales of object meshes to the range of $[-0.5, 0.5]$, and translate objects to the origin, where the sphere-center is also fixed for scanning SP maps. We then render 4 layers of SP maps per object with the resolution 256×512 since we empirically discover 4 layers of SP maps are adequate to cover almost all surface points. For evaluation, we follow prior works to use the Google Scanned Objects (GSO) dataset [Downs et al. 2022] and we randomly choose 30 shapes consisting of common objects used in daily life. We use the same protocol to render one image as the evaluation input. We further exploit the Deepfashion3D dataset [Zhu et al. 2020] to illustrate our model capacity on open-surface objects, we follow the setting in [Yu et al. 2024] to divide the train and test splits. For data from Deepfashion3D, we render 3 layers of SP maps per object with the resolution 256×512, and we use the same protocol as [Yu et al. 2024] to generate sketch for each input view.

Method	Latency	CD.↓	Vol. IoU↑	F-Sco. (%)↑
Point-E	~25s	0.0690	0.1953	52.23
Shape-E	~20s	0.0418	0.2785	64.83
Wonder3D	~10min	0.0398	0.2930	68.82
CRM	~18s	0.0264	0.3374	74.43
OpenLRM	~15s	0.0344	0.3770	71.50
LGM	~40s	0.0212	0.4220	78.41
InstantMesh	~35s	0.0120	0.4310	88.84
Ours	6-10s	**0.0051**	**0.5407**	**95.57**

Table 1. **Quantitative comparison on GSO.** Our specific inference time (latency) depends on how many steps we use for denoising and adopting the normal estimator with different sizes.

Method	CD.↓	Vol. IoU↑	F-Sco. (%)↑
Wonder3D	0.0223	0.3370	73.52
OpenLRM	0.0237	0.3680	78.25
LGM	0.0244	0.3110	71.60
InstantMesh	0.0314	0.2890	68.72
SurfD	0.0136	0.3860	82.31
Ours (rgb)	0.0099	0.4200	87.16
Ours (sketch)	**0.0092**	**0.4480**	**89.35**

Table 2. **Quantitative comparison on DeepFashion3d.**

4.1.2 Metrics. To evaluate the geometry quality of our method, we report Chamfer Distance (CD), Volume IoU and F-Score (with a threshold of 0.1) between the reconstructed mesh and groundtruth mesh. Since the generated meshes are usually placed at different angles, we follow [Huang et al. 2025] and perform brute-force search in rotations to align each predicted mesh with the groundtruth mesh before centering and scaling all meshes to $[-1, 1]$.

4.1.3 Baselines. We compare to state-of-the-art 3D creation models with single view image (or sketch) as condition, including geometry-based representations Point-E [Nichol et al. 2022], Shape-E [Jun and Nichol 2023], OpenLRM [He and Wang 2023] (an open-sourced implementation of LRM [Hong et al. 2023]) and SurfD [Yu et al. 2024]. Multi-view based methods Wonder3D [Long et al. 2024], CRM [Wang et al. 2025c], LGM [Tang et al. 2025] and InstantMesh [Xu et al. 2024a]. We use their official code implementations and pretrained weights.

4.1.4 Implementation Details. We firstly modify the input and output channel of the AutoEncoder from 3 to 1 to fit the single-channel

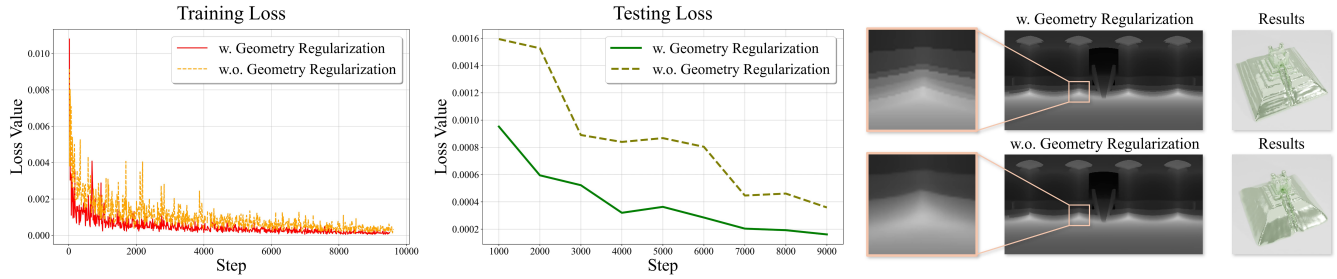

Fig. 4. **Analysis on geometry regularization.** On the left, we compare the loss curve on training and testing splits, and on the right we visualize the geometry details on SP maps and reconstructed mesh surfaces.

Resolution	32		64		128		256	
	CD.↓	Storage↓	CD.↓	Storage↓	CD.↓	Storage↓	CD.↓	Storage↓
Matryoshka	7.59	10	2.43	25	1.43	78	0.95	261
UV Mapping	6.28	8	2.29	32	1.16	128	0.88	512
Ours	2.66	5	1.58	16	0.96	56	0.85	194

Table 3. **Quantitative comparisons with image-based methods Matryoshka [Richter and Roth 2018] and UV Mapping [Yan et al. 2024b]** in terms of representation capacity (Chamfer Distance, CD. $\times 10^{-3}$) and storage efficiency (KB) by reconstructing the ground-truth meshes under different image resolutions.

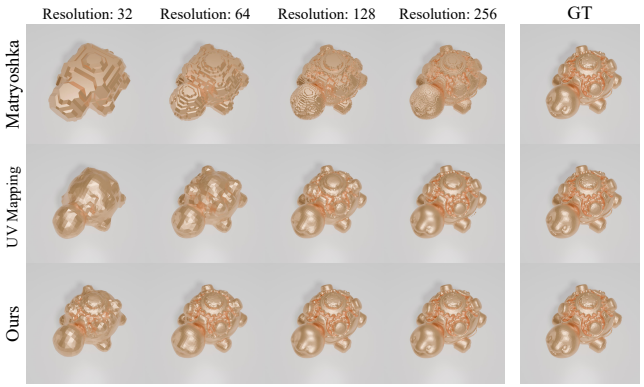

Fig. 5. **Visualization reconstruction quality of Matryoshka [Richter and Roth 2018] and UV Mapping [Yan et al. 2024b].** We reconstruct the ground-truth under different image resolutions, the comparison shows that our SP maps maintain better geometry details and surface qualities.

Method	CD.↓	Vol. IoU↑	F-Sco. (%)↑
Ours	0.0051	0.5407	95.57
w.o. LSA	0.0436	0.2568	60.75
w.o. finetuning AE	0.0610	0.2072	52.04
w.o. finetuning UNet	0.1742	0.1034	27.42

Table 4. **Ablations on Layer-wise Self-Attention (LSA) and finetuning.**

depth input, and then finetune with our proposed geometry regularization for 40k iterations with a total batchsize 64 on all training data from Objaverse. The initial learning rate is set as 1×10^{-4} with a cosine annealing learning rate scheduler. After the AutoEncoder is finetuned, we offline generate SP map latents and use them to finetune the denoise UNet. We train the UNet for 80k iterations with a total batchsize 80 per shape on all data. The initial learning rate is set as 1×10^{-5} with warm up for $100 steps$ and annealing learning rate scheduler, and we use DDIM [Song et al. 2020] and Euler Ancestral Discrete [Karras et al. 2022] noise scheduler for training and inference respectively. We follow [Wang et al. 2023] and apply circular padding to the SP maps. After the finetuning on Objaverse, we continue to finetune the model on the deepfashion3D training split since the SP maps have different numbers of layers in which exist potential gaps. We train with both sketch and RGB images as conditions to obtain versatile generation abilities. All of our fine-tuning procedure requires only two GPUs, each equipped with 18,176 CUDA cores, 142 streaming multiprocessors, and 48 GB of VRAM, running for approximately seven days—demonstrating greater computational efficiency compared to prior works.

4.2 Compare with SOTA Methods

We compare our SPGen with other SOTA works on GSO, Objaverse validation split and Deepfashion3d test split. For GSO and Objaverse validation, we picked the front left view of the object as the condition images. For Deepfashion3d, we follow [Yu et al. 2024] and pick the front view of sketch or RGB image as conditions.

4.2.1 Qualitative Results. As shown in Fig. 8, our generated shapes yield consistent and smooth geometry, this is due to the adoption of SP maps as the coherent representation, in contrast, multiview-based methods such as Wonder3D fail when view inconsistency occurs. Moreover, our method handles shapes with relatively complex topologies, such as porous parts or thin-walled cup structures, while methods relying on SDF for surface extraction fail to accurately generate these structures. It is worth mentioning that our method has good symmetry. When one side of the symmetrical object is occluded, our method accurately restores the unseen geometry.

We also conduct visual comparisons on DeepFashion3D in Fig. 9. For general single image shape creation methods, we adopt front-view RGB image as the condition. For SurfD, we adopt sketch as a condition since they focus on the sketch-to-shape setting. While our method can adopt either RGB or sketch as input in this more challenging setting, no existing SDF-based methods can accurately describe open surfaces. SurfD with the point cloud UDF diffusion model beats those SDF-based methods, and we achieve better results based on UDF, indicating our capability of representing and generating open-surface objects.

Additionally, we compare with image-based representations including nested depth maps from Matryoshka [Richter and Roth

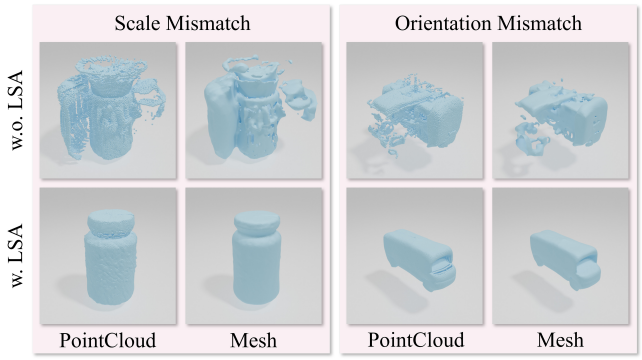

Fig. 6. **Visualization on the effect of Layer-wise Self-Attention (LSA).** Without LSA, the predicted layers show scale or orientation mismatch.

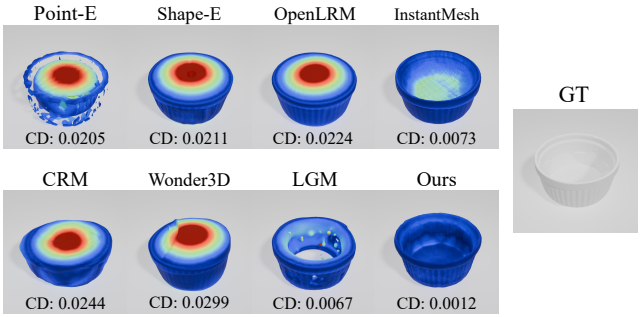

Fig. 7. **Qualitative comparison demonstrating the correlation between Chamfer Distance (CD) and geometric quality.** Lower Chamfer Distance (CD) values correspond directly to higher-fidelity 3D reconstructions with fewer visual artifacts.

2018] and UV Mapping from [Yan et al. 2024b]. We compare the surface quality and geometry details under different image resolutions and as shown in Fig. 5, our SP map yields significantly better surface quality with fewer jagged artifacts compared with Matryoshka, and richer geometry details (e.g. the gear area) under the same resolution compared with UV Mappings.

4.2.2 Quantitative Results. We use the aforementioned metrics to evaluate the accuracy of generated shapes on both GSO and DeepFashion3d datasets. As shown in Table 1 and Table 2, our method surpasses other SOTA works on all three metrics, and we consume relatively low latency during inference. On GSO dataset, we achieve +57.5%, +25.4%, and +7.6% relative gain compared with the best performing InstantMesh [Xu et al. 2024a] in CD., Volume IoU and F-Score respectively, and on DeepFashion3D, we achieve +32.4%, +16.1%, and +8.6% relative gain on the three metrics over SurfD [Yu et al. 2024], indicating the effectiveness of our method. We also compare with image-based methods in terms of representation capacity and storage efficiency by reconstructing the ground-truth meshes under different image resolutions. As shown in Table 3, our SP maps achieve higher reconstruction accuracies and require less storage compared with other two methods under different resolutions, especially at lower resolutions. Since all of these involved 2D image-based representations could be incorporated with the same generative pipeline, our SP Map representation, superior in both reconstruction quality and compactness, provides the generative model with a more accurate and efficient target, allowing higher potential for high-fidelity shape generation.

4.3 Ablation Study

4.3.1 Study on Geometry Regularization. In Sec. 3.2.3, we claimed the importance of applying our proposed geometry regularization to encourage high-quality surface reconstruction and detail preserving during the AutoEncoder training process. Here we conduct ablations to validate the effectiveness. As shown in Fig. 4, we randomly select a small subset with 10k samples from our training split, and train the AutoEncoder on it with or without geometry regularization. Applying geometry regularization greatly speeds up convergence and reduces the training loss from 3.05×10^{-4} to 1.47×10^{-4}, testing loss from 3.59×10^{-4} to 1.61×10^{-4} by 9k iterations. As shown in the right part, after applying geometry regularization, the high-frequency details are greatly enhanced on both SP maps and reconstructed mesh surfaces.

4.3.2 Study on Layer-wise Self-Attention. In Sec. 3.2.2, we claim that the Layer-wise Self-Attention (LSA) is proposed to learn the spatial relations of different SP layers. To validate its effect, we removed the LSA layers in the denoise UNet, the results are shown in Table 4 and Fig. 6, removing LSA resulting in the mismatch of either layer scale or orientation, and leading to poor mesh quality.

4.3.3 Study on Finetuning. As we mentioned in Sec. 3.2.1, we finetuned the whole pipeline since the pretrained weights are only generalized on RGB images. In Table. 4, we analyze the role of finetuning the pipeline, without finetuning either component of the pipeline, the performance drops significantly, therefore, it is necessary to finetune the whole pipeline on all SP maps.

4.3.4 Significance of Chamfer Distance Improvements. We use Chamfer Distance (CD) to quantify the geometric accuracy of the reconstructed meshes. We emphasize that minor numerical reductions in CD correspond to significant and visually perceptible improvements in mesh quality as shown in Fig. 7. A lower CD score directly reflects an enhanced ability to capture correct object structures, maintain surface smoothness, and eliminate artifacts. Thus, the incremental CD gains reported in our experiments represent meaningful progress towards generating high-fidelity 3D geometry.

5 Conclusions

We propose SPGen, a novel framework for high-quality 3D shape generation using multi-layer Spherical Projection (SP) as a structural representation. SPGen effectively addresses three key challenges in current 3D creation models: view inconsistency, limited representation capability and low efficiency. By projecting 3D mesh surfaces onto a unit sphere and unfolding them into 2D SP maps, our method ensures geometric consistency through the injective mapping from SP maps to 3D surfaces and flexibly captures complex topologies, including internal layers and open surfaces. SPGen incorporates our proposed geometry regularization and layer-wise self-attention to enhance geometry quality. Extensive experiments demonstrate that SPGen outperforms existing methods in geometric accuracy while maintaining low computational cost and latency, making it a robust and efficient solution for 3D shape generation.

References

Antonio Alliegro, Yawar Siddiqui, Tatiana Tommasi, and Matthias Nießner. 2023. Polydiff: Generating 3d polygonal meshes with diffusion models. *arXiv preprint arXiv:2312.11417* (2023).

D Blender Online Community. 2018. Blender—A 3D modelling and rendering package. *Blender Foundation* (2018).

Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. 2021. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9650–9660.

Changyou Chen, Han Ding, Bunyamin Sisman, Yi Xu, Ouye Xie, Benjamin Z Yao, Son Dinh Tran, and Belinda Zeng. 2024b. Diffusion models for multi-task generative modeling. In *The Twelfth International Conference on Learning Representations*.

Sijin Chen, Xin Chen, Anqi Pang, Xianfang Zeng, Wei Cheng, Yijun Fu, Fukun Yin, Yanru Wang, Zhibin Wang, Chi Zhang, et al. 2024a. MeshXL: Neural Coordinate Field for Generative 3D Foundation Models. *arXiv preprint arXiv:2405.20853* (2024).

Yiwen Chen, Tong He, Di Huang, Weicai Ye, Sijin Chen, Jiaxiang Tang, Xin Chen, Zhongang Cai, Lei Yang, Gang Yu, et al. 2024c. MeshAnything: Artist-Created Mesh Generation with Autoregressive Transformers. *arXiv preprint arXiv:2406.10163* (2024).

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

Alexey Dosovitskiy. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. 2022. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2553–2560.

Slava Elizarov, Ciara Rowles, and Simon Donné. 2024. Geometry Image Diffusion: Fast and Data-Efficient Text-to-3D with Image-Based Surface Representation. *arXiv preprint arXiv:2409.03718* (2024).

Mengyang Feng, Jinlin Liu, Miaomiao Cui, and Xuansong Xie. 2023. Diffusion360: Seamless 360 degree panoramic image generation based on diffusion models. *arXiv preprint arXiv:2311.13141* (2023).

Xiao Fu, Wei Yin, Mu Hu, Kaixuan Wang, Yuexin Ma, Ping Tan, Shaojie Shen, Dahua Lin, and Xiaoxiao Long. 2025. Geowizard: Unleashing the diffusion priors for 3d geometry estimation from a single image. In *European Conference on Computer Vision*. Springer, 241–258.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. *Advances in neural information processing systems* 27 (2014).

Xianfeng Gu, Steven J Gortler, and Hugues Hoppe. 2002. Geometry images. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 355–361.

Antoine Guédon and Vincent Lepetit. 2024. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5354–5363.

Benoit Guillard, Federico Stella, and Pascal Fua. 2022. Meshudf: Fast and differentiable meshing of unsigned distance field networks. In *European Conference on Computer Vision*. Springer, 576–592.

Zekun Hao, David W Romero, Tsung-Yi Lin, and Ming-Yu Liu. 2024. Meshtron: High-Fidelity, Artist-Like 3D Mesh Generation at Scale. *arXiv preprint arXiv:2412.09548* (2024).

Takayuki Hara, Yusuke Mukuta, and Tatsuya Harada. 2022. Spherical image generation from a few normal-field-of-view images by considering scene symmetry. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (2022), 6339–6353.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

Xianglong He, Zi-Xin Zou, Chia-Hao Chen, Yuan-Chen Guo, Ding Liang, Chun Yuan, Wanli Ouyang, Yan-Pei Cao, and Yangguang Li. 2025. SparseFlex: High-Resolution and Arbitrary-Topology 3D Shape Modeling. *arXiv preprint arXiv:2503.21732* (2025).

Zexin He and Tengfei Wang. 2023. Openlrm: Open-source large reconstruction models.

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

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. 2023. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400* (2023).

Tao Hu, Wenhang Ge, Yuyang Zhao, and Gim Hee Lee. 2024. X-Ray: A Sequential 3D Representation for Generation. *arXiv preprint arXiv:2404.14329* (2024).

Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2024. 2d gaussian splatting for geometrically accurate radiance fields. In *ACM SIGGRAPH 2024 conference papers*. 1–11.

Zixuan Huang, Mark Boss, Aaryaman Vasishta, James M Rehg, and Varun Jampani. 2025. SPAR3D: Stable Point-Aware Reconstruction of 3D Objects from Single Images. *arXiv preprint arXiv:2501.04689* (2025).

Liming Jiang, Bo Dai, Wayne Wu, and Chen Change Loy. 2021. Focal frequency loss for image reconstruction and synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*. 13919–13929.

Lutao Jiang, Ruyi Ji, and Libo Zhang. 2023. Sdf-3dgan: A 3d object generative method based on implicit signed distance function. *arXiv preprint arXiv:2303.06821* (2023).

Justin Johnson, Alexandre Alahi, and Li Fei-Fei. 2016. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*. Springer, 694–711.

Heewoo Jun and Alex Nichol. 2023. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463* (2023).

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. 2022. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems* 35 (2022), 26565–26577.

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

Diederik P Kingma. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. 2020. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics (ToG)* 39, 6 (2020), 1–14.

Jiahao Li, Hao Tan, Kai Zhang, Zexiang Xu, Fujun Luan, Yinghao Xu, Yicong Hong, Kalyan Sunkavalli, Greg Shakhnarovich, and Sai Bi. 2023b. Instant3d: Fast text-to-3d with sparse-view generation and large reconstruction model. *arXiv preprint arXiv:2311.06214* (2023).

Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. 2023a. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 12642–12651.

Yangguang Li, Zi-Xin Zou, Zexiang Liu, Dehu Wang, Yuan Liang, Zhipeng Yu, Xingchao Liu, Yuan-Chen Guo, Ding Liang, Wanli Ouyang, et al. 2025. TripoSG: High-Fidelity 3D Shape Synthesis using Large-Scale Rectified Flow Models. *arXiv preprint arXiv:2502.06608* (2025).

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

Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. 2024. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems* 36 (2024).

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

Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023a. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453* (2023).

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

Zhuqiang Lu, Kun Hu, Chaoyue Wang, Lei Bai, and Zhiyong Wang. 2024. Autoregressive Omni-Aware Outpainting for Open-Vocabulary 360-Degree Image Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 14211–14219.

Xuyi Meng, Chen Wang, Jiahui Lei, Kostas Daniilidis, Jiatao Gu, and Lingjie Liu. 2025. Zero-1-to-G: Taming Pretrained 2D Diffusion Model for Direct 3D Generation. *arXiv preprint arXiv:2501.05427* (2025).

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

Tomas Möller and Ben Trumbore. 1997. Fast, minimum storage ray-triangle intersection. *Journal of graphics tools* 2, 1 (1997), 21–28.

Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. 2022. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751* (2022).

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193* (2023).

William Peebles and Saining Xie. 2023. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*. 4195–4205.

Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. 2020. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 523–540.

John Phillips, Julieta Martinez, Ioan Andrei Bârsan, Sergio Casas, Abbas Sadat, and Raquel Urtasun. 2021. Deep multi-task learning for joint localization, perception, and prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4679–4689.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. 2023. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952* (2023).

Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

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

Stephan R Richter and Stefan Roth. 2018. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1936–1944.

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

Zhuowen Shen, Yuan Liu, Zhang Chen, Zhong Li, Jiepeng Wang, Yongqing Liang, Zhengming Yu, Jingdong Zhang, Yi Xu, Scott Schaefer, et al. 2024. SolidGS: Consolidating Gaussian Surfel Splatting for Sparse-View Surface Reconstruction. *arXiv preprint arXiv:2412.15400* (2024).

Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2023. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023).

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

Jiaming Song, Chenlin Meng, and Stefano Ermon. 2020. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502* (2020).

Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. 2025. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. In *European Conference on Computer Vision*. Springer, 1–18.

Jiaxiang Tang, Jiawei Ren, Hang Zhou, Ziwei Liu, and Gang Zeng. 2023. Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. *arXiv preprint arXiv:2309.16653* (2023).

Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. 2024. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151* (2024).

Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. *Advances in neural information processing systems* 30 (2017).

Simon Vandenhende, Stamatios Georgoulis, Wouter Van Gansbeke, Marc Proesmans, Dengxin Dai, and Luc Van Gool. 2021. Multi-task learning for dense prediction tasks: A survey. *IEEE transactions on pattern analysis and machine intelligence* 44, 7 (2021), 3614–3633.

Simon Vandenhende, Stamatios Georgoulis, and Luc Van Gool. 2020. Mti-net: Multi-scale task interaction networks for multi-task learning. In *European conference on computer vision*. Springer, 527–543.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

Jionghao Wang, Ziyu Chen, Jun Ling, Rong Xie, and Li Song. 2023. 360-degree panorama generation from few unregistered nfov images. *arXiv preprint arXiv:2308.14686* (2023).

Jionghao Wang, Cheng Lin, Yuan Liu, Rui Xu, Zhiyang Dou, Xiaoxiao Long, Haoxiang Guo, Taku Komura, Wenping Wang, and Xin Li. 2025a. PDT: Point Distribution Transformation with Diffusion Models. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–11.

Jionghao Wang, Yuan Liu, Zhiyang Dou, Zhengming Yu, Yongqing Liang, Cheng Lin, Rong Xie, Li Song, Xin Li, and Wenping Wang. 2025b. Disentangled clothed avatar generation from text descriptions. In *European Conference on Computer Vision*. Springer, 381–401.

Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. 2021. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. *arXiv preprint arXiv:2106.10689* (2021).

Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2025c. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision*. Springer, 57–74.

Haoyu Wu, Meher Gitika Karumuri, Chuhang Zou, Seungbae Bang, Yuelong Li, Dimitris Samaras, and Sunil Hadap. 2024a. Direct and explicit 3D generation from a single image. *arXiv preprint arXiv:2411.10947* (2024).

Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. 2018. Learning shape priors for single-view 3d completion and reconstruction. In *Proceedings of the European conference on computer vision (ECCV)*. 646–662.

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

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3d latents for scalable and versatile 3d generation. *arXiv preprint arXiv:2412.01506* (2024).

Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024a. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

Yinghao Xu, Zifan Shi, Wang Yifan, Hansheng Chen, Ceyuan Yang, Sida Peng, Yujun Shen, and Gordon Wetzstein. 2024b. Grm: Large gaussian reconstruction model for efficient 3d reconstruction and generation. *arXiv preprint arXiv:2403.14621* (2024).

Kun Yan, Lei Ji, Chenfei Wu, Jian Liang, Ming Zhou, Nan Duan, and Shuai Ma. 2024a. HORIZON: High-Resolution Semantically Controlled Panorama Synthesis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 6431–6439.

Xingguang Yan, Han-Hung Lee, Ziyu Wan, and Angel X Chang. 2024b. An object is worth 64x64 pixels: Generating 3d object via image diffusion. *arXiv preprint arXiv:2408.03178* (2024).

Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. 2021. Volume rendering of neural implicit surfaces. *Advances in Neural Information Processing Systems* 34 (2021), 4805–4815.

Lior Yariv, Omri Puny, Oran Gafni, and Yaron Lipman. 2024. Mosaic-sdf for 3d generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4630–4639.

Hanrong Ye and Dan Xu. 2022. Inverted pyramid multi-task transformer for dense scene understanding. In *European Conference on Computer Vision*. Springer, 514–530.

Zhengming Yu, Zhiyang Dou, Xiaoxiao Long, Cheng Lin, Zekun Li, Yuan Liu, Norman Müller, Taku Komura, Marc Habermann, Christian Theobalt, et al. 2024. Surf-D: Generating High-Quality Surfaces of Arbitrary Topologies Using Diffusion Models. In *European Conference on Computer Vision*. Springer, 419–438.

Zhengming Yu, Tianye Li, Jingxiang Sun, Omer Shapira, Seonwook Park, Michael Stengel, Matthew Chan, Xin Li, Wenping Wang, Koki Nagano, et al. 2025. GAIA: Generative Animatable Interactive Avatars with Expression-conditioned Gaussians. In *Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference Conference Papers*. 1–10.

Biao Zhang, Jiapeng Tang, Matthias Niessner, and Peter Wonka. 2023. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Transactions On Graphics (TOG)* 42, 4 (2023), 1–16.

Jingdong Zhang, Jiayuan Fan, Peng Ye, Bo Zhang, Hancheng Ye, Baopu Li, Yancheng Cai, and Tao Chen. 2025. BridgeNet: Comprehensive and Effective Feature Interactions via Bridge Feature for Multi-Task Dense Predictions. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2025).

Jingdong Zhang, Hanrong Ye, Xin Li, Wenping Wang, and Dan Xu. 2024b. Multi-task label discovery via hierarchical task tokens for partially annotated dense predictions. *arXiv preprint arXiv:2411.18823* (2024).

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

Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. 2018. Learning to reconstruct shapes from unseen classes. *Advances in neural information processing systems* 31 (2018).

Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. 2025. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202* (2025).

Xinyang Zheng, Yang Liu, Pengshuai Wang, and Xin Tong. 2022. SDF-StyleGAN: Implicit SDF-Based StyleGAN for 3D Shape Generation. In *Computer Graphics Forum*, Vol. 41. Wiley Online Library, 52–63.

Xin-Yang Zheng, Hao Pan, Peng-Shuai Wang, Xin Tong, Yang Liu, and Heung-Yeung Shum. 2023. Locally attentional sdf diffusion for controllable 3d shape generation. *ACM Transactions on Graphics (ToG)* 42, 4 (2023), 1–13.

Heming Zhu, Yu Cao, Hang Jin, Weikai Chen, Dong Du, Zhangye Wang, Shuguang Cui, and Xiaoguang Han. 2020. Deep fashion3d: A dataset and benchmark for 3d garment reconstruction from single images. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*. Springer, 512–530.

Fig. 8. **Qualitative comparison with SOTA works**. Our SPGen yields highly accurate surface geometry with better topologies.

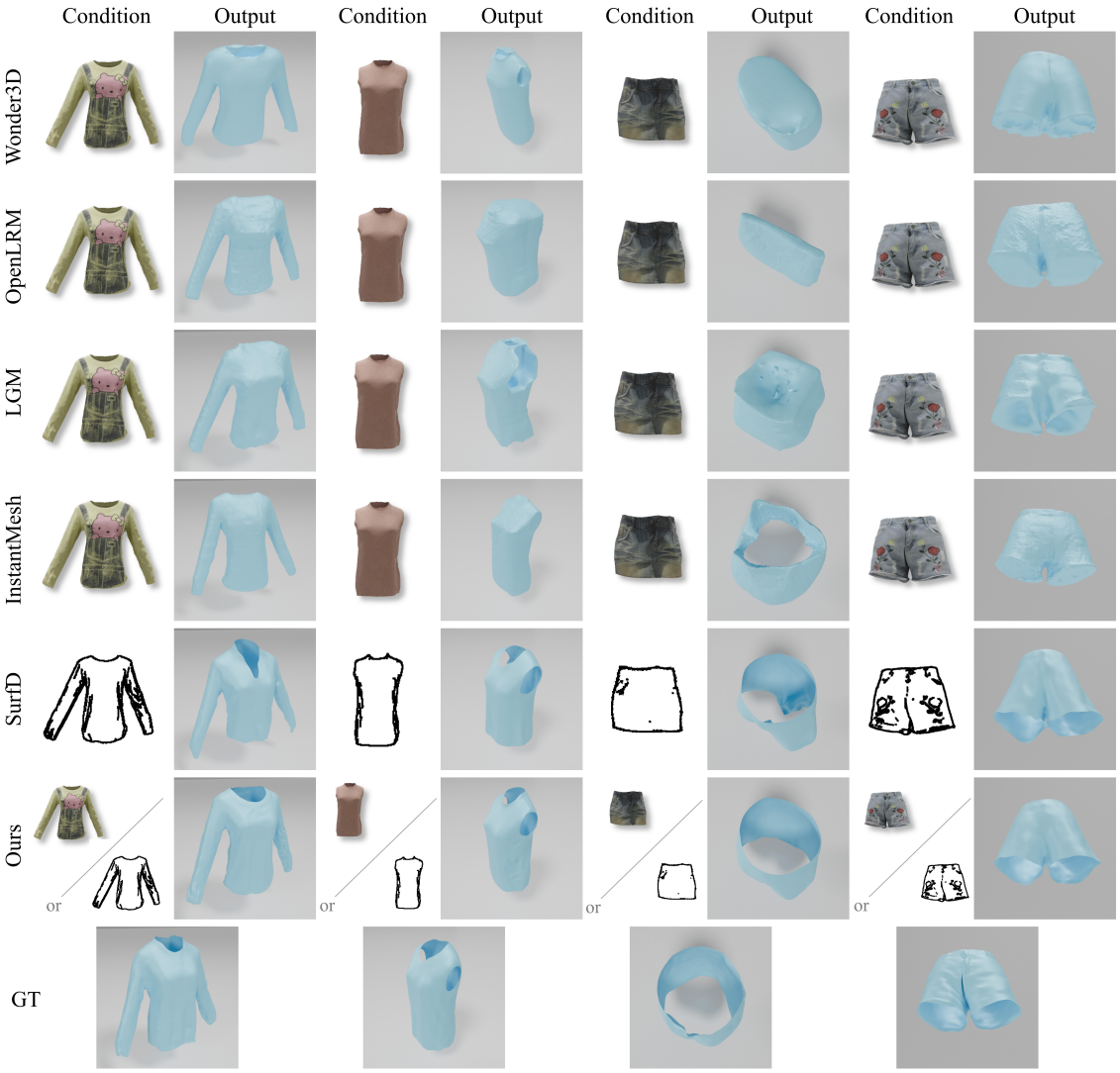

Fig. 9. **Qualitative comparison with SOTA works on DeepFashion3d test split.** Our SPGen can handle either sketch or RGB images individually as the condition. Since the geometric visual effects of the results conditioning by two different conditions are similar, we only show the effect of using RGB as the condition here.

Supplementary

In this supplementary material, we will discuss: i) More details of designs and implementations. ii) More comparisons with image-based geometry representations and other generative pipelines. iii) More ablation studies. iv) Downstream application scenarios. v) Video results showcase.

A Detail Design and Implementation

A.1 Single Image Conditioned Denoising

We further claim the usage of single image conditioning and the corresponding visual encoder in this section. To control the denoising process, conditions are usually applied by cross-attention [Vaswani 2017], to the UNet layers. For image conditions, usually a pre-trained vision encoder is applied to embed the condition image into high dimension feature space, and then perform cross-attention with hidden states of UNet. These vision encoders are trained by

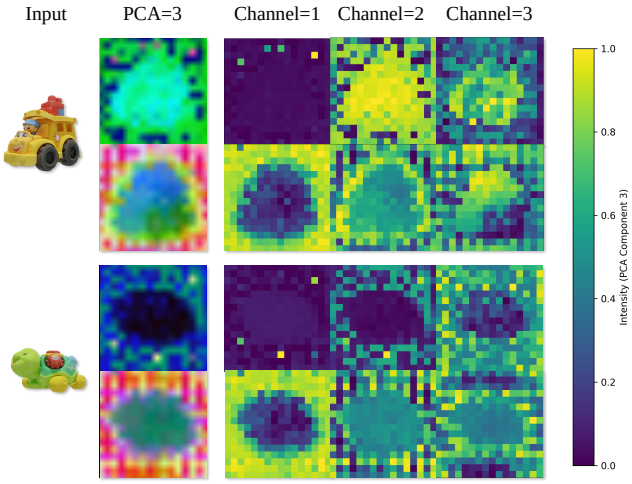

Fig. 10. **Visualization of the visual embedding distributions.** We visualize the final layer token maps of CLIP and DINOv2 via PCA and show the first three main components. The primary component of CLIP feature contains almost no spatial information, and for other major components, the quality of the feature is also lower than DINOv2.

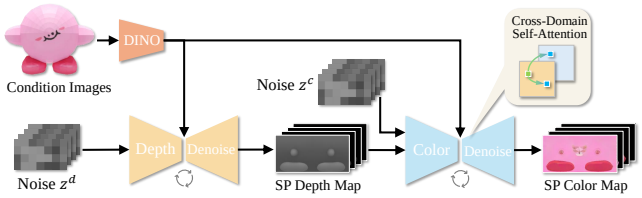

Fig. 11. **Illustration of the depth and color SP maps generation pipeline.** We adopt a cascade pipeline to generate SP depth maps with single-view image condition first, and then use both the generated SP depth map and the single-view image to condition the SP color map generation. The cross-domain self-attention is applied to guarantee the color and depth are matched on the generated maps. Note that we also adopt the latent diffusion pattern for both stages, and the autoencoder is omitted.

contrastive learning [He et al. 2020; Radford et al. 2021] or self-distillation [Caron et al. 2021; Oquab et al. 2023] to gain the ability to capture image semantics and structures. Current shape generation baselines usually adopt either or both kinds of vision encoders to guide denoising.

We investigate the quality of the visual embeddings produced by the most commonly used contrastive learning CLIP model and self-distillation DINOv2 model. These models use the same ViT [Dosovitskiy 2020] backbone so we take the output token map from the final layer and use principal component analysis (PCA) to reduce the channel dimension to 3. As shown in Fig. 10, the primary component from CLIP contains almost no spatial information, since the contrastive learning process focuses more on matching the global semantics of image-text pairs. Differently, DINOv2 adopts various data augmentations to enhance the capturing of image-level details, therefore gaining more refined token maps. For our goal, we target generating 3D shapes that are more similar to the condition image, which requires more image-level details to perform cross-attention and fertilize the 3D shape generation process. Thus, we take DINOv2 as our vision-encoder for diffusion pipeline.

A.2 Spherical Projected Texture Map

Our way of generating Spherical Projection maps to record depth as the geometry can also be extended to different surface attributes, e.g. texture, normal vector, curvature, etc. We show the capability of SPGen on texture generation by recording the vertex colors of each intersection point on an SP color map to represent the surface texture (corresponding with the SP depth map representing the geometry). The map preparation process is the same as we explained in Sec. 3.1. To generate the SP color map, we implement an extra diffusion pipeline conditioned by both the single-view image and SP depth map. As shown in Fig. 11, the whole pipeline is based on a cascade structure, in the depth denoise stage, we use the same diffusion model as in Fig. 2, after the SP depth maps are generated, in the color denoise stage, we use both the SP depth latents and single-view image embeddings to jointly condition another identical denoising UNet.

As discussed in [Long et al. 2024; Meng et al. 2025; Wu et al. 2024a], multi-attribute images like RGB, depth, normal can be generated in a shared denoising UNet by domain switcher [Long et al. 2024; Meng et al. 2025] or extra domain-specific branches [Wu et al. 2024a], and bunches of works have proven the benefits of learning multiple domains together leading to the joint promotions [Chen et al. 2024b; Long et al. 2024; Phillips et al. 2021; Radford et al. 2021; Vandenhende et al. 2021, 2020; Wu et al. 2024a; Zhang et al. 2025, 2024b]. However, these methods do not apply to SP maps, as the RGB and depth domains on SP maps are significantly distinct. Though domain gaps also remain among different attributes in perspective projected images, they still share the basic image-level structures, i.e., contours, shapes, etc. In contrast, different attributes on SP maps can result in completely distinct map distributions, e.g., a sphere with complex texture or a complex shape with pure color. To avoid negative transfer of knowledge among these domains, we adopt a decoupled pipeline that denoising the shape and texture separately based on their corresponding SP maps. Specifically, following [Shi et al. 2023], after the SP depth latents z^d are generated, we add noise to them in the set steps t to simulate the noisy distribution z_t^d, and then feed the noisy depth latent into the color denoising UNet to record the intermediate hidden status. Afterwards, we sample pure noise from a uniform distribution and also feed it into the same UNet, with conducting Cross-Domain Self-Attention with the pre-saved depth hidden status. Thus, the learning of the texture map could be bundled to the corresponding shape. We also apply the embeddings from the single-image condition to provide the visible texture information to the network by cross-attention. Finally, after the denoising and generating z^c, we use a specifically finetuned VAE decoder to restore the SP color maps.

Due to the resource limitation, we conduct a small-scale experiment on a split of 2k Objaverse training data, as shown in Fig. 12,

Method	CD.↓	Vol. IoU↑	F-Sco. (%)↑
Wonder3D	0.0246	0.3618	74.86
CRM	0.0172	0.3945	79.28
OpenLRM	0.0136	0.4512	82.42
LGM	0.0203	0.3756	77.59
InstantMesh	0.0157	0.4108	82.62
Ours	**0.0061**	**0.5527**	**94.74**

Table 5. **Quantitative comparison on Objaverse validation split.**

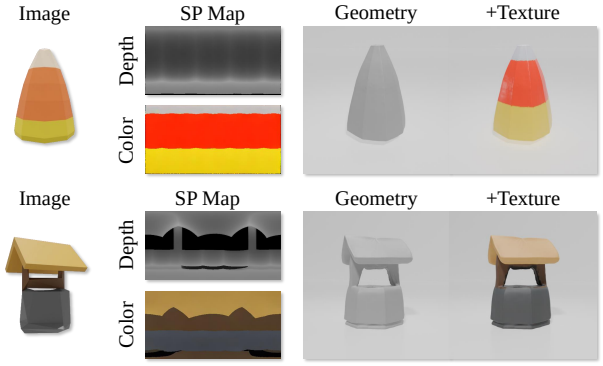

Fig. 12. **Visualization of the textured shape generation.**

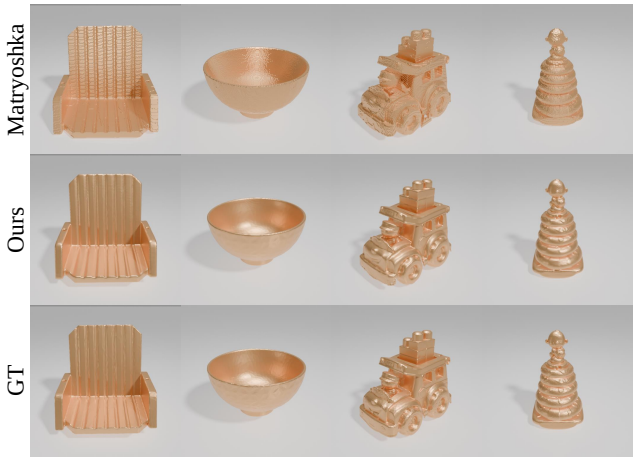

Fig. 13. **Visualization reconstruction quality of Matryoshka [Richter and Roth 2018] and our SP maps.** We reconstruct the ground-truth directly with both methods, the comparison shows that our SP maps maintains better surface details compared with matryoshka.

the results shows the feasibility of generating textured mesh via Spherical Projection maps.

A.3 Implementation Details

A.3.1 Dataset Preparation. We refer to the criteria in [Chen et al. 2024c; Long et al. 2024] to filter the Objaverse dataset by removing low-quality or scene-level meshes and acquire around 160k objects as our whole training split. We also picked 1993 objects out of the training indices as our validation split on Objaverse. Before training, we follow [Long et al. 2024] and render 13 views per object with resolution 512×512 including orthographic and other oblique perspectives by Blender [Blender Online Community 2018]. We normalize the scales of object meshes to the range of $[-0.5, 0.5]$ without normalizing the orientations to maintain diversity, and translate objects to the origin. We also fix the sphere-center at the origin which is used to scan SP maps. The azimuth range for SP map is $\theta \in [-\pi/2, 3\pi/2)$, and the polar range is $\varphi \in [0, \pi)$.

A.3.2 Model Details. We use the VAE and UNet from SDXL [Podell et al. 2023], and load the pretrained weight of both to conduct finetuning. For VAE reconstruction, we apply L1 loss and our geometry regularization on the reconstructed maps, and a KL-divergence regularization with the weight of 10^{-8} to ensure the latent space won't shift away from the uniform distribution during the finetuning. For the denoising Unet, we apply LSA layers and adopt L2 loss in the latent space. After the SP maps are generated, we unproject the points on map into the 3D space, and use a normal estimator to predict normal vector for each point. The normal estimator is a ConvONet [Peng et al. 2020], we scale it up to 192 hidden dimensions, and the 3D-UNet encoder contains 5-levels with 128 feature dimensions. And the model is also trained on all our training split by sampling 25600 oriented points on each mesh object.

A.4 Limitations

A.4.1 Faces Parallel to the Ray Direction. A theoretical ambiguity may arise when surface faces are exactly parallel to the sphere radius (i.e., orthogonal to the spherical surface), as in the flat boundary of a hemisphere. In such cases, rays shot from the sphere center may intersect the face tangentially or fail to yield a unique depth, leading to potential instability. Our implementation is based on the Möller–Trumbore algorithm [Möller and Trumbore 1997] and explicitly detects near-parallel ray-face configurations by recording the angle between the ray and face normal. When the angle falls below a small threshold, the intersection is discarded to avoid numeral unstable cases. Empirically, such events are usually rare—fewer across the dataset, and in pathological cases with large flat regions aligned with the radial direction, we have to apply a small random perturbation to the ray origin or surface to ensure numerical stability.

A.4.2 Distortions at the Polar Areas. Equirectangular projection of the sphere introduces non-uniform sampling, especially near the polar regions, which may cause distortion, which could possibly lead to inaccurate geometry details in the polar areas. However, because SP maps have no severe degeneration or flipped mapping, these distortions remain acceptable and errors on 3D surfaces are controllable. We conduct analysis in the ablation study.

Method	GPU	Training Time	Iterations	Data Amount	Latency	CD. ↓	Vol. IoU ↑	F-Score (%)↑
CLAY	256 *A800* (10 TB)	~2 *weeks*	–	527k	~15s	0.0046	0.6355	96.95
Trellis	64 *A100* (2.5 TB)	–	400k	500k	~40s	0.0030	0.6495	98.35
Hunyuan3D-2	–	–	–	–	~15s	0.0028	0.7440	98.43
TripoSG	160 *A100* (6.25 TB)	~3 *weeks*	700k	2m	~50s	0.0030	0.7381	99.08
TripoSF	64 *A100* (2.5 TB)	-	-	400k	-	-	-	-
Ours	2 GPUs (0.09 TB)	~1 *week*	80k	160k	6–10s	0.0034	0.6208	98.28

Table 6. **Quantitative comparison with large foundation 3D generative pipelines.**

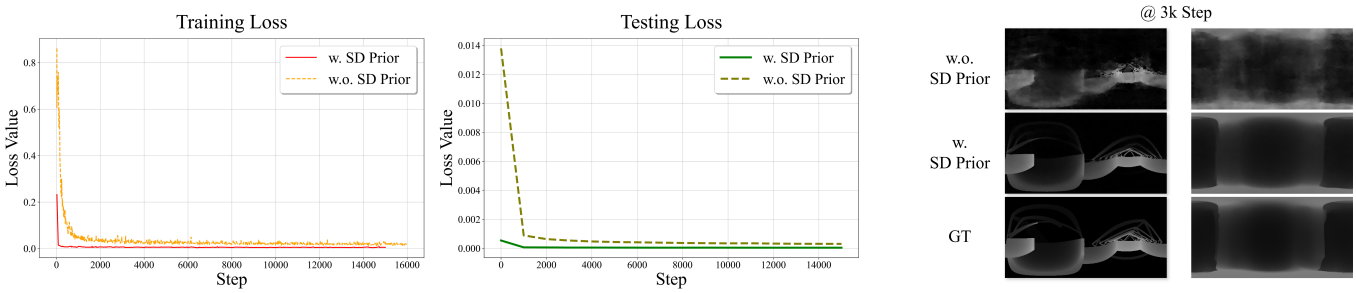

Fig. 14. **Analysis on SD priors.** On the left, we compare the loss curve on training and testing splits, and on the right we visualize comparison of SP depth maps at step 3k with and without SD priors.

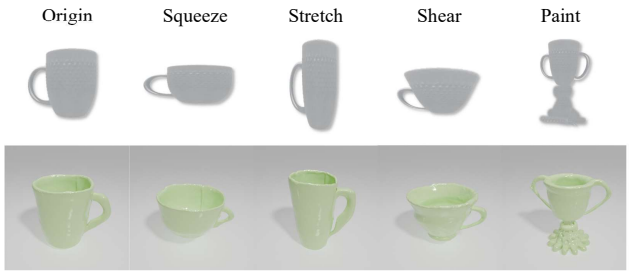

Fig. 15. **Visualization of the shape editing via image control.** We try to edit the original image and SPGen is able to generate corresponding shapes accurately.

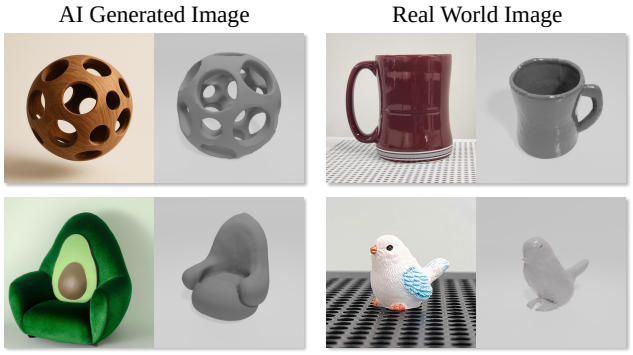

Fig. 16. **Real-world Evaluation.**

B Comparisons and Discussions

B.1 Comparisons on Objaverse Validation Split

As we mentioned in Sec. 4.1.1, we set a small validation split on Objaverse with 1993 objects to indicating the status of model convergence. Note that since different works are adopting different data filtering strategy on Objaverse, so our validation data could be the training data of other works. We conduct quantitative comparisons with our baselines on this split. We randomly choose 20 shapes consisting of common objects used in daily life, and the results are shown in Table 5. Our SPGen also achieves consistent gain on all three metrics compared with other works.

B.2 Comparisons with Image-based Geometry Representations

B.2.1 Comparison with Zero-1-to-G. Zero-1-to-G [Meng et al. 2025] is an extended multi-view diffusion method from [Long et al. 2024] incorporating multiple Gaussian attributes, whereas SPGen differently uses consistent spherical projections and directly encodes surface geometry. Zero-1-to-G cannot guarantee the strict view-consistency and relies on an extra differentiable-rendering stage for surface extraction.

B.2.2 Comparison with Geometry Image and UV Mapping. Geometry Image [Elizarov et al. 2024; Gu et al. 2002] is a representation that unfolds a mesh surface onto a single regular 2D grid so each pixel stores the surface's (x, y, z) coordinates, while UV Mapping [Yan et al. 2024b] assigns 2D (u, v) coordinates to a parametrized mesh surface to accurately wrapped over it. Both representations unfold the 3D surface onto a structural 2D domain which makes it possible to leverage the strong priors from pretrained image generative models. However, these representations are different from SP maps

in: 1) These methods have non-unique geometry mapping of the same object, which hinders the construction of consistent large-scale datasets and scalable model training; 2) Both of them require extensive cutting to unfold mesh surface, especially on objects with genus > 0, which potentially leads to more burdens for model to learn the boundary relations, and possibly leads to higher border errors with unstitched patches (refer to Fig.5 in [Yan et al. 2024b] and Fig.7 in [Elizarov et al. 2024]). In contrast, SP maps leverage fixed mappings and cuts, which ensure the building a standardized and scalable generative training pipeline.

B.2.3 Comparison with Matryoshka Network. Matryoshka Network [Richter and Roth 2018] encodes a shape by predicting six axis-aligned stacks of nested depth images (one for each $\pm X, \pm Y, \pm Z$ direction) that are fused into a fixed-resolution voxel grid before polygonization. This design can also effectively record 3D shapes in structural 2D images and takes advantage of image processing networks such as 2D CNNs. However, compared with SP maps, it is still limited in (i) the voxel–resolution bottleneck: though the nested depth images are N^2 complexity, the surface detail is capped by the N^3 volume, whereas SPGen reconstructs geometry directly from high-resolution spherical-projection maps whose memory grows only quadratically, enabling lower budgets in reconstructing the geometries; and (ii) the cross-view or cross-layer consistency: since no explicit cross-attentions among views or layers to ensure the consistency, though the final depth fusion step will store a unique shape, the depth stacks may still contain inconsistency (e.g., entry/exit order inversion along a ray, misaligned strip boundaries), which leads to possible holes, thinned walls, or jagged artifacts despite on the restored shape. In contrast, SP map is a naturally view consistent injective function plus layer-wise self-attention to eliminate potential conflicts.

We also conduct ground-truth surface reconstruction experiments on both Matryoshka and SP maps, we still use 256 as the resolution for SP maps, and 4 layers in depth. To align with our setting, we use 256^3 of spatial resolution for Matryoshka, and also set maximum 4 layers. Note that 256^3 voxel is relatively expensive during surface extraction since the original setting of Matryoshka in single-view reconstruction is 32^3. As shown in Fig. 13, our reconstruction results achieve higher quality with more details, while Matryoshka results are suffering from jagged artifacts and coarse surfaces.

B.2.4 Comparison with GenRe. GenRe [Zhang et al. 2018] adopts a cascaded, multi-stage pipeline that first predicts a depth map from an RGB image, projects it to a partial spherical map, then completes this map using a feed-forward 2D inpainting network, and finally projects the result to a voxel grid for refinement. This pipeline has key limitations that our end-to-end framework effectively addresses i) Representation capability: GenRe uses a single-layer spherical map that lacks internal structure representation and relies on post voxel-based refinement, leading to resolution bottleneck. In contrast, our multi-layer SP maps efficiently capture complex topologies directly in the 2D domain. ii) External dependencies: GenRe relies on predefined camera parameters and a separate depth estimato, which introduces error accumulation. Our framework is fully self-contained and free from such dependencies.

B.3 Comparisons with More 3D Generative Pipelines

Recently, there are bunches of scalable 3D generative models trained on large-scale datasets yielding strong generalization ability and robustness on generating high-quality 3D meshes. These works adopt 3DShape2VecSet [Zhang et al. 2023] ([Li et al. 2025; Zhang et al. 2024a; Zhao et al. 2025]) or Sparse Voxel ([He et al. 2025; Xiang et al. 2024]) as the geometry representation, and use signed distance functions or occupancy field as implicit surfaces. These works usually adapt scalable DiT [Peebles and Xie 2023] and train on $\sim$ 500k or more data. As shown in Table 6, we compare to these works on randomly selected GSO and Objaverse-validation data. Since CLAY is a closed-source work, we use their API Rodin Gen-1. Our SPGen only consumes less than 5% of the training resources to achieve relatively competitive performance with faster inference speed, indicating the compactness and effectiveness.

C Ablation Studies

C.1 Study on Pretrained SD Priors

Recent studies have demonstrated that SD models are highly adaptable and can improve performance on various 2D representations beyond RGB, including panoramic images [Wang et al. 2023], depth and Gaussian feature maps [Meng et al. 2025; Wu et al. 2024a; Yu et al. 2025], normal maps [Long et al. 2024], etc. These works show that SD priors are beneficial in bridging domain gaps and performing generalization. Similar to [Wang et al. 2023; Wu et al. 2024a], our SP map is unfolded surface depth image with panoramic-style distortion, retaining image-level local structures and spatial patterns, which can be naturally benefited from SD priors. We also conduct ablation studies on the training convergence with and without SD priors. As shown in Fig. 14, the model with SD priors achieves significantly faster convergence and lower loss (4.37×10^{-5} *v.s.* 3.08×10^{-4} at 15k step on test data), further validating the effectiveness of SD priors post-adaptation. Additionally, we also show the generated SP depth map at training step 3k, with SD Prior, the result has significantly better quality with less noise.

C.2 Study on Border Consistency

We follow [Wang et al. 2023] and apply circular padding in the azimuth direction to ensure the SP maps align well at the border and the rotational invariance. The circular padding encourages the generative model to learn consistent predictions across the entire SP map. We calculate the absolute-relative-error between the azimuth borders, which is only 0.23%, which proves the consistent learning effect on borders.

C.3 Study on the Number of SP Map Layers

To ensure representation comprehensiveness, we evaluated reconstructed IoU on 160K Objavese objects using different numbers of SP map layers. The results are:

$\{1 : 92.0\%, 2 : 98.7\%, 3 : 99.8\%, 4 : 99.9\%, 5 : 99.9\%\}$,

indicating that using less than 4 layers yields incomplete reconstructions, while additional layers provide negligible improvement. Thus, we select 4 layers as they effectively model complex objects while balancing representation completeness and computational efficiency.

C.4 Study on the Distortions.

We empirically measure the absolute-relative-error across surface regions and observe only marginal variation: 0.20% in polar areas, 0.25% near the equator, and an average of 0.22%. This validates that the model handles projection distortion effectively, and that geometric reconstruction from SP maps remains robust even in the presence of polar distortions.

C.5 Real-World Evaluation

We evaluated our SPGen with real-world data to verify its generalization ability and robustness. As shown in Fig. 16, we use AI-generated images and daily photos as conditions, our method achieves good geometric quality and restores the conditional image well.

D Downstream Applications

Our generation pipeline can be used for a lot of downstream tasks such as editing, rendering, animation, etc. In this section, we test the editing ability of shapes via the single image control. As shown in Fig. 15, we perform several editing ways including squeeze, stretch, shear, and direct paint on it, our SPGen shows stable and controllable transformation according to the change of input view, indicating the power of robust generation and generalization.

E Video Results

More detailed evaluations and comparisons can be found in the attached video.

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009