SPAR3D: Stable Point-Aware Reconstruction of 3D Objects from Single Images

Zixuan Huang[1,2*] Mark Boss[1] Aaryaman Vasishta[1] James M. Rehg[2] Varun Jampani[1]
[1]Stability AI, [2]UIUC

Figure 1. We present SPAR3D, a state-of-the-art 3D reconstructor that reconstructs high-quality 3D meshes from single-view images. SPAR3D enjoys a fast reconstruction speed at 0.7 seconds and supports interactive user edits.

Abstract

We study the problem of single-image 3D object reconstruction. Recent works have diverged into two directions: regression-based modeling and generative modeling. Regression methods efficiently infer visible surfaces, but struggle with occluded regions. Generative methods handle uncertain regions better by modeling distributions, but are computationally expensive and the generation is often misaligned with visible surfaces. In this paper, we present SPAR3D, a novel two-stage approach aiming to take the best of both directions. The first stage of SPAR3D generates sparse 3D point clouds using a lightweight point diffusion model, which has a fast sampling speed. The second stage uses both the sampled point cloud and the input image to create highly detailed meshes. Our two-stage design enables a probabilistic modeling of the ill-posed single-image 3D task, while maintaining high computational efficiency and great output fidelity. Using point clouds as an intermediate representation further allows for interactive user edits. Evaluated on diverse datasets, SPAR3D demonstrates superior performance over previous state-of-the-art methods, at an inference speed of 0.7 seconds.

[*]Work done at Stability AI.

1. Introduction

Reconstructing 3D objects from monocular images is a fundamental problem in computer vision. An efficient reconstruction system opens up a wide range of applications, including augmented reality, filmmaking, and manufacturing. Monocular 3D reconstruction is also a complex inverse problem: while the visible surface can be estimated from shading, predicting the occluded surface necessitates a strong 3D object prior. Our field has seen a divergence in two different directions: feedforward regression [3, 11, 20, 25, 26, 28, 38, 55, 56, 61–64, 67, 68, 71] and diffusion-based generation [7, 9, 10, 27, 30, 32–36, 40, 48–50, 70, 73]. Despite the significant progress made in both directions, each has fundamental limitations.

Regression-based models are highly effective in adhering to the visible surface in the image, and the inference speed is typically fast. However, they makes the oversimplified assumption of bijective mapping between images and 3D. This assumption introduces ambiguity in the learning objective, leading to poorly estimated surfaces and textures in occluded regions. On the other hand, diffusion-based approaches are generative and do not predict the statistical mean. However, their iterative sampling at inference time is computationally inefficient when modeling high-resolution 3D. Additionally, previous studies such as [28] indicate that diffusion-generated 3D models exhibit worse alignment to the surface visible in the input image. How can we take the best of both worlds while avoiding their limitations?

In light of this, we propose SPAR3D, which breaks the 3D reconstruction process down into two stages: the point sampling stage and the meshing stage. The point sampling stage uses diffusion models to generate sparse point clouds, followed by the meshing stage transforming point clouds into highly detailed meshes. Our main idea is to offload the uncertainty modeling to the point sampling stage, where the low resolution of the point clouds allows rapid iterative sampling. The subsequent meshing stage leverages the local image features to transform the point cloud into a detailed mesh of high output fidelity. Reducing the meshing uncertainty with point clouds further facilitates unsupervised learning of inverse rendering, which reduces the baked-in lighting in the textures. Our two-stage design enables SPAR3D to significantly outperform previous regressive methods, while preserving high computational efficiency and fidelity to input observation.

A key design choice of our method is the usage of point clouds to connect the two stages. To ensure fast reconstruction, our intermediate representation needs to be lightweight so it can be efficiently generated. On the other hand, it should provide enough guidance to the meshing stage. This inspires us to use point clouds, which are perhaps the most computationally efficient 3D representation because all information bits are used to represent the surface. Moreover, the lack of connectivity, typically considered as the drawback of point clouds, now turns into an advantage with our two-stage approach for editing purposes. When the back surface does not align with user expectations, local edits can be easily made on the low-resolution point clouds without worrying about topologies (see Fig. 1 bottom). Feeding edited point clouds into the meshing stage produces better meshes tailored towards user requirements.

Our experiments demonstrate the superiority of SPAR3D over previous state-of-the-art methods, with solid quantitative and qualitative results on various data sources. SPAR3D also exhibits a strong generalization ability to in-the-wild images and AI-generated images. With a total inference time below 0.7 seconds, SPAR3D is not only efficient but also allows for easy user-driven edits, offering a practical solution to the task of monocular 3D reconstruction. We hope that this is a meaningful step towards scalable generation of high-quality 3D assets.

2. Related Work

Feedforward 3D reconstruction methods address the problem of 3D object reconstruction by learning a feedforward model in a regression-based manner. Earlier works [11, 20, 26, 38, 61, 64, 68] in this field typically predict only the geometry and train on small datasets [6, 52], which limits their generalization ability. Recently, larger 3D datasets [12, 44] have been collected, unlocking the potential to train feedforward 3D models at scale [25, 28, 63]. These models exhibit great generalization ability to unseen images, and excel at producing reconstructions that tightly align with the observed cues in the input image. In particular, LRM [25] and follow-up works [3, 55, 56, 62, 67, 71] show that properly designed large transformer models can be trained using only rendering losses to capture object geometry and texture in great detail. Despite the high fidelity and computational efficiency of these models, the oversimplified bijective assumption in these regressive approaches results in oversmoothed unseen surfaces. Multi-view diffusion models [35, 36, 47, 48] have been considered as a remedy for this, where additional viewpoints are synthesized as input to the feedforward model [55, 62, 67]. However, the inconsistency across viewpoints often leads to significant artifacts on the reconstructed surfaces, and the computational efficiency of these approaches is severely affected by the slow multi-view generation process. Our model also aims to overcome the learning ambiguity in regressive approaches, but our point sampling approach is inherently 3D-consistent and computationally efficient, and further allows easy user edits.

Generative 3D modeling learns the image-conditioned distribution of 3D assets instead of a deterministic mapping. Early 3D generative works use GAN [5, 17, 29, 57],

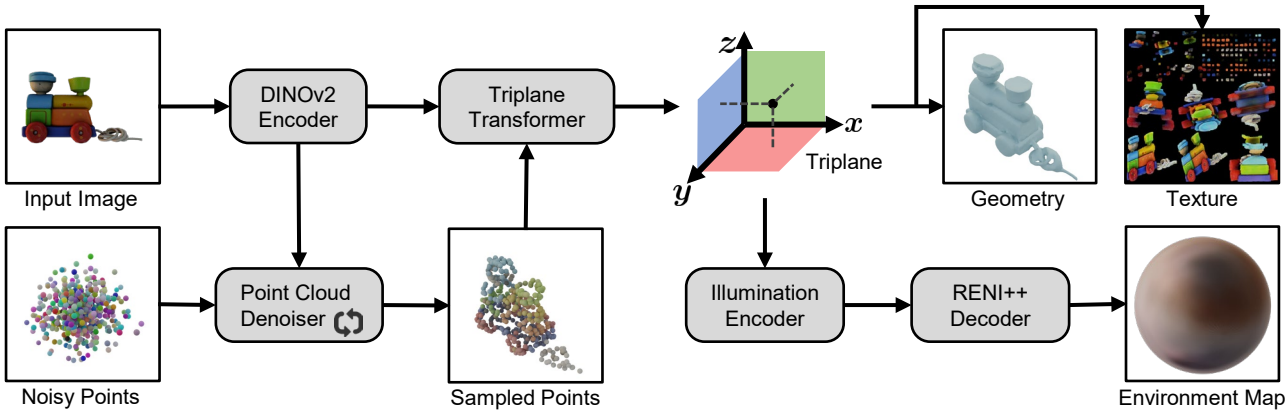

Figure 2. **SPAR3D Overview.** Conditioned on the input image, SPAR3D first leverages a point diffusion model to generate a sparse point cloud. The triplane transformer then uses the sampled point cloud and image features to produce high-resolution triplane features. The triplane features are then queried to reconstruct the geometry, texture, and illumination of the object in the image.

normalizing flow [31, 69] or VAE [18, 39, 66] as the generative framework. Inspired by the success of 2D diffusion models [15, 45], 3D diffusion models [7, 9, 10, 27, 30, 32–36, 40, 48–50, 70, 73] have also been extensively explored in recent works. Despite the advantage of probabilistic modeling that avoids over-smoothed results, diffusion-based 3D generation has two drawbacks: 1) not aligning well with input observations, and 2) having low inference speed at high resolution. Our work inherits the advantage of probabilistic modeling, while avoiding the drawbacks by using diffusion to generate only sparse point clouds.

Optimization-based single-view 3D leverages 2D generative priors to recover 3D from single-view images. These works [13, 21, 37, 54] rely on SDS-type loss [43, 60] and generate 3D assets by optimizing for each object image separately. These methods achieve promising results without large-scale annotation. However, the lack of a strong explicit 3D prior makes the optimization process inefficient and prone to local minima.

3. Method

SPAR3D Overview. Given the input image $I \in \mathbb{R}^{3\times h\times w}$, our method produces a 3D mesh with PBR materials, including albedo, metallic, roughness and surface normals. The main goal of our work is to develop a model that enjoys the benefits of distribution learning through diffusion models, while not suffering from the low output fidelity and computational inefficiency. To this end, we design a two-stage model that consists of the point sampling stage and the meshing stage (see Fig. 2). At the point sampling stage, a point diffusion model learns the conditional distribution of point clouds given the input image. This stage is computationally efficient given the low resolution of the point clouds. The regression-based meshing stage transforms the sampled point cloud into a highly detailed mesh that aligns with the visible surface. The reduced uncertainty with point sampling further facilitates the learning of materials and illumination in an unsupervised manner during the meshing stage. This reduces baked-in lighting artifacts and results in better modeling of specular surfaces. Finally, by using sparse point clouds as the intermediate representation, SPAR3D enables human editing in the loop.

3.1. Point Sampling Stage

Overview. The point sampling stage produces a sparse point cloud as the input to the meshing stage. The core of the point sampling stage is a point diffusion model, which generates point clouds $\boldsymbol{p}_0 \in \mathbb{R}^{n\times 6}$ conditioned on the input image I. The six channels include three XYZ channels and three RGB channels. In our work, the resolution of the point cloud n is set to 512.

Point Diffusion Framework. Our diffusion framework is based on DDPM [24], which consists of two processes: 1) the forward process which adds noise to the original point cloud, and 2) the backward process where the denoiser learns to remove the noise. At timestep $t \in [0, T]$, the diffusion process combines Gaussian noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$ with a point cloud $\boldsymbol{p}_0$ as

$$\boldsymbol{p}_t = \sqrt{\bar{\alpha}_t}\boldsymbol{p}_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \tag{1}$$

where $\bar{\alpha}_t$ denotes the noise schedule. We use the sigmoid noise schedule proposed in [8], combined with input scaling and the renormalization trick. The denoiser $\boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t; \boldsymbol{c})$ then learns to recover the noise from $\boldsymbol{p}_t$ and is supervised by

$$L_{simple}(\theta) = \mathbb{E}_{t,\boldsymbol{p}_0,\boldsymbol{\epsilon}}\|\boldsymbol{\epsilon} - \boldsymbol{\epsilon}_\theta(\boldsymbol{p}_t, t; \boldsymbol{c})\|_2^2. \tag{2}$$

Here $\boldsymbol{c}$ denotes the image condition tokens. During inference, we use the DDIM sampler [51] to generate point cloud samples. Samples generated directly often align poorly with the condition, hence we use the classifier-free guidance (CFG) [23] to improve sampling fidelity.

Denoiser Design. We use a transformer denoiser similar to Point-E [40], where the noisy point cloud $\boldsymbol{p}_t \in \mathbb{R}^{n\times 6}$ is linearly mapped to a set of point tokens $\boldsymbol{x} \in \mathbb{R}^{n\times d}$. We use DINOv2 [41] to encode the input image I as conditioning tokens $\boldsymbol{c} \in \mathbb{R}^{c\times d}$. The conditions and the point tokens are then concatenated together as input to the transformer, which predicts the added noise on each point.

Albedo Point clouds. In the meshing stage, we estimate the materials and lighting alongside the geometry. However, this decomposition is inherently ambiguous because there are countless combinations of lighting and albedo that can explain the same input image. It is challenging to learn this highly uncertain decomposition during the regressive meshing stage alone. We therefore reduce the uncertainty at the point sampling stage, by directly generating albedo point clouds with diffusion models. Sampling albedo point clouds as input to the meshing stage drastically reduces the ambiguity of inverse rendering and stabilizes the decomposition learning.

3.2. Meshing Stage

Overview. The meshing stage produces a textured mesh from the input image and the point cloud. The backbone of our meshing model is a large triplane transformer, which predicts triplane features from the image and point cloud conditions. We estimate the geometry, texture and lighting of the current object from the triplane, and metallic/roughness from the image features. The geometry and materials are fed into our differentiable renderer during training, so that we can apply rendering loss to supervise our model.

Triplane Transformer. Our triplane transformer consists of three submodules: the point cloud encoder, the image encoder, and the transformer backbone. We use a simple transformer encoder to encode the point cloud as a set of point tokens. Given the low resolution of the point clouds, each point can be directly mapped to a single token. Our image encoder is DINOv2 [41], which produces local image embeddings. Our triplane transformer follows a similar design to PointInfinity [27] and SF3D [3], which produces triplanes at high resolution of 384×384 by using a computationally-detached two-stream design.

Surface Estimation. To estimate the geometry, the triplanes are queried with a shallow MLP to produce density values. Similar to [3, 62, 67], we convert the implicit density field to explicit surface using differentiable Marching Tetrahedron (DMTet) [46]. We additionally use two MLP heads to predict vertex offsets and surface normals together with density. These two attributes reduce the artifacts introduced by the Marching Tetrahedron and lead to locally smoother surfaces.

Material and Illumination Estimation. We perform inverse rendering and jointly estimate materials (albedo, metallic and roughness) and illumination alongside the geometry. The task is highly ill-posed and Neural-PIL [2] showed that an illumination prior can reduce the ambiguity. We build our illumination estimator upon the learning-based illumination prior from RENI++ [19]. RENI++ is originally an unconditional generative model for HDR illumination generation. We learn an encoder to map triplane features into the latent space of RENI++. This allows us to estimate the environment illumination in the input image. The albedo is estimated from triplane similar to geometry, where a shallow MLP predicts the albedo value for each 3D location. For metallic and roughness, we follow SF3D [3] and learn to estimate them with a probabilistic approach via a Beta prior. We find that the CLIP encoder used in SF3D is unstable when the object size changes. We therefore replace their CLIP encoder with AlphaCLIP [53] to alleviate this issue using foreground object masks.

Differentiable Rendering. We implement a differentiable renderer that renders images based on the predicted environment map, PBR materials and geometry surface (see Fig. 3). We use a differentiable mesh rasterizer and add a differentiable shader. Specifically, we leverage the standard simplified Disney PBR model [4] in our shader. As we use RENI++ to reconstruct environment maps, we need to explicitly integrate the incoming radiance. Here, we opt to use the Monte Carlo Integration. Given the low sample counts we can computationally afford during training, we rely on Multiple Importance Sampling (MIS) with the balanced heuristic [58] to reduce integration variance. Additionally, to better model the self-occlusion which has been typically ignored in prior works, we implement a visibility test for better shadow modeling. We take inspiration from real-time graphics and model the visibility test as a screen-space method using the depth map from our rasterizer. An overview of this test is shown in Fig. 4. Specifically, we ray-march a short distance (0.25) in 6 steps for all proposed sample directions from MIS, and project the position back to image space. If the current ray depth is farther away than the sampled value from the depth map, then the ray is marked as shadowed.

Loss Function. Our main loss function is the rendering loss that compares renderings from novel views to the grountruth (GT) images. Specifically, our rendering loss is a linear combination of 1) the L2 distance between the rendered and GT images, 2) the perceptual distance between the rendered and GT images measured by LPIPS [72], and 3) the L2 distance between the rendered opacity and the GT foreground mask. Apart from the rendering loss, we also follow SF3D and apply the mesh and shading regularization that regularizes the surface smoothness and the inverse rendering respectively.

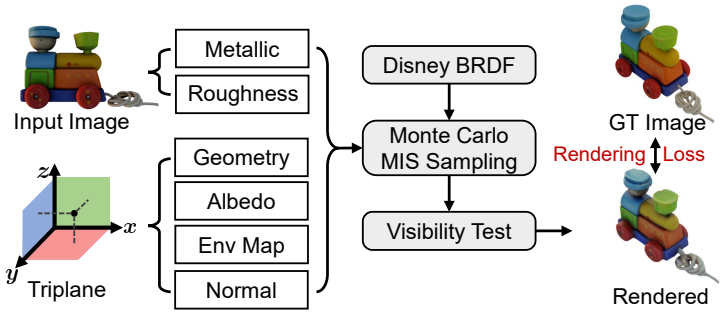

Figure 3. **Our Differentiable Renderer.** We estimate geometry, albedo, lighting, and normal maps from the triplane and metallic/roughness values from the image. We rasterize and interpolate these values as input to our shader (omitted here for simplicity). Our shader uses the Disney BRDF [4] and performs Monte Carlo integration. We further perform visibility testing to improve shadow modeling. Finally, we compare the rendered image with the GT image and minimize the rendering loss.

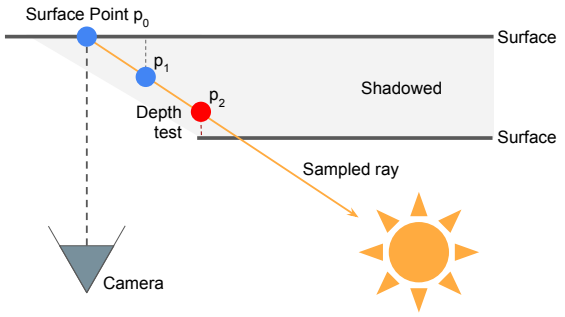

Figure 4. **Shadow Modeling.** We perform visibility testing in screen-space by marching along sampled rays. If any point along the ray has a ray depth which is farther away than the depth map, we consider the entire ray as shadowed.

3.3. Interactive Editing

A unique advantage of our two-stage design is that it naturally supports interactive editing of unseen regions in our produced mesh. In most circumstances, the visible surface is determined by the input image and remains highly accurate, while the unseen surface is mainly based on the sampled point cloud, which might not align with user intention. In this case, editing the unseen surface of the mesh is feasible by altering the point cloud. Point clouds are perhaps one of the most flexible 3D representation for editing purposes because there are no topology constraints. Given the low resolution of our point clouds, editing the point cloud is fairly efficient and intuitive. Users can easily delete, duplicate, stretch or recolor points in the point cloud. Our efficient meshing model is able to produce the adjusted mesh in 0.3 seconds, which makes this process fairly interactive.

3.4. Implementation Details

Point Sampling Stage. Our point diffusion model has 16 transformer blocks in total. Each transformer block consists of two Layer Normalization layer, one Multi-Head Attention (MHA) layer and one MLP. We use a feature dimension of 1024 and 16 attention heads in each MHA layer. With many emissive objects in our dataset, albedo can be visually distinct from the input image and hard to learn. Therefore, instead of directly generating albedo point clouds in the point sampling stage, we learn to generate white-lit point clouds as a proxy target.

Meshing Stage. Our triplane transformer consists of 4 two-stream blocks [27]. Each two-stream block consists of three self-attentions and two cross-attentions. The main computation is carried out using 3,072 latent tokens, each with a feature dimension of 1024. The MHA includes 16 attention heads. The point cloud encoder is a vanilla transformer with 12 layers and 512 feature dimension, and the image encoder is DINOv2-large. We use a tetrahedra resolution of 160 for DMTet. In our differentiable shader, we follow Hasselgren *et al*. [22] and use a detached biased sampling scheme. We sample based on the specular lobe (GGX [59]), the 2D piecewise-linear distribution of the environment map luminance and the hemispherical distribution. Specifically, we include 6 samples from the GGX lobe, 6 samples from the 2D piecewise-linear distribution of the luminance, and 4 samples from the hemispherical distribution. The main body of the shader is implemented in PyTorch, while the screen-space shadowing and the 2D piecewise-linear distribution computation for the environment map are implemented as custom CUDA kernels for efficiency. The training of our meshing stage includes multiple phases, where we increase the rendering resolution and decrease the batch size at later training phases. We use GT point clouds as input when training the meshing model. The curation of our training data follows TripoSR [56].

4. Experiments

4.1. Evaluation

Datasets. We used two datasets for evaluation, GSO [16] and OmniObject3D [65]. We follow TripoSR [56] and remove simple box or cylindrical objects to avoid bias on simple geometries. Each of the evaluation sets consists of around 250 objects. We render the objects with diverse azimuth angles at different elevations, with randomly sampled HDRI environment maps. We also vary the focal length of the camera to create more diverse test cases.

Metrics. To evaluate the geometry quality of the reconstructed meshes, we use follow prior works [56, 67] and use Chamfer Distance (CD) and F-score (FS) as our evaluation metrics. CD measures the alignment between two point clouds and is defined as the average of accuracy and

Method	CD↓	FS@0.1↑	FS@0.2↑	FS@0.5↑	PSNR↑	SSIM↑	LPIPS↓	Time (s)↓
Shap-E [30]	0.204	0.359	0.638	0.922	15.3	0.802	0.205	3.1
LN3Diff [30]	0.174	0.422	0.703	0.949	17.1	0.819	0.169	5.1
LGM [55]	0.196	0.356	0.635	0.936	17.0	0.818	0.184	41.0
CRM [62]	0.161	0.437	0.735	0.961	17.5	0.830	0.169	7.4
TripoSR [56]	0.145	0.501	0.784	0.968	18.5	0.837	0.151	**0.2**
InstantMesh [67]	0.135	0.545	0.812	0.971	18.1	0.838	0.146	36.1
SF3D [3]	0.137	0.540	0.806	0.970	18.0	**0.839**	0.145	0.3
SPAR3D (ours)	**0.120**	**0.584**	**0.850**	**0.983**	**18.6**	0.836	**0.139**	0.7

Table 1. **Quantitative Comparisons on GSO [16].** SPAR3D performs favorably to other state-of-the-art methods.

Method	CD↓	FS@0.1↑	FS@0.2↑	FS@0.5↑	PSNR↑	SSIM↑	LPIPS↓	Time (s)↓
Shap-E [30]	0.212	0.349	0.624	0.909	14.8	0.8006	0.205	3.1
LN3Diff [30]	0.160	0.480	0.744	0.957	16.7	0.819	0.161	5.0
LGM [55]	0.200	0.366	0.638	0.924	16.1	0.810	0.188	42.0
CRM [62]	0.155	0.482	0.765	0.962	17.0	0.828	0.162	7.0
TripoSR [56]	0.144	0.537	0.785	0.963	**18.0**	0.835	0.147	**0.2**
InstantMesh [67]	0.145	0.546	0.790	0.962	17.2	0.832	0.150	34.7
SF3D [3]	0.138	0.554	0.800	0.967	17.4	**0.836**	0.145	0.3
SPAR3D (ours)	**0.122**	**0.587**	**0.845**	**0.978**	17.9	0.832	**0.140**	0.7

Table 2. **Quantitative Comparisons on OmniObject3D [65].** SPAR3D performs favorably to other state-of-the-art methods.

completeness:

$$d(S_1, S_2) = \frac{1}{2|S_1|} \sum_{x \in S_1} \min_{y \in S_2} \|x-y\|_2 + \frac{1}{2|S_2|} \sum_{y \in S_2} \min_{x \in S_1} \|x-y\|_2 \tag{3}$$

FS evaluates point cloud alignment by calculating the F-score with a predefined threshold. Predicted points that lie within the distance threshold are considered as correct predictions. A higher FS means better alignment between the reconstructed shape and the groundtruth. To evaluate the texture quality, we compute standard image metrics, including PSNR, SSIM and LPIPS, between images rendered from the predicted mesh and the groundtruth images.

Protocol. To calculate the metrics that are comparable across methods, the meshes need to lie in the same coordinate system. To this end, we perform brute-force search in rotations to align each predicted mesh with the groundtruth mesh. Both the prediction and the groundtruth are normalized before the brute-force alignment, and the alignment is further refined with ICP.

Baselines. We compare SPAR3D with other efficient methods for single-view 3D generation or reconstruction [3, 55, 56, 62, 67]. We use the official implementation for all baselines, and we evaluate the produced meshes under the same protocol. Specifically, we compare against TripoSR [56], LGM [55], CRM [62], InstantMesh [67], LN3Diff [32], Shap-E [30] and SF3D [3]. Among these baselines, TripoSR and SF3D are pure regression-based approaches; LGM, CRM and InstantMesh use multiview diffusion to generate pseudo multi-view images; LN3Diff and Shap-E are purely diffusion-based 3D generative models.

4.2. Main Results

Quantitative Comparison. We compare SPAR3D to other baselines on GSO and Omniobject3D quantitatively. As shown in Tab. 1 and Tab. 2, SPAR3D outperforms all other regressive or generative baselines significantly across most metrics on both datasets. For SSIM, we observe that SPAR3D is slightly worse than the strongest baseline for this metric. We find that this relates to the Monte Carlo noise from our shader. SPAR3D is also among the fastest reconstruction models with an inference speed of 0.7 seconds per object, which is significantly faster than 3D or multi-view diffusion-based approaches.

Qualitative Results. We show qualitative results of different methods in Fig. 5. The reconstructed meshes from pure regression-based approaches such as SF3D or TripoSR align with the input image well, but the backside is often less accurate and over-smoothed. Multi-view diffusion-based methods such as LGM, CRM and InstantMesh show more details on the backside. However, the inconsistency in the synthesized views leads to clear artifacts and overall worse results. Pure generative approaches such as Shap-E and LN3Diff are able to produce sharp surfaces in their generation. However, many details are erroneous hallucinations that do not accurately follow the input images, and the visible surfaces are often reconstructed incorrectly. Compared to prior art, the meshes produced by SPAR3D not only faithfully resemble the input image, but also exhibit well-generated occluded parts with reasonable details.

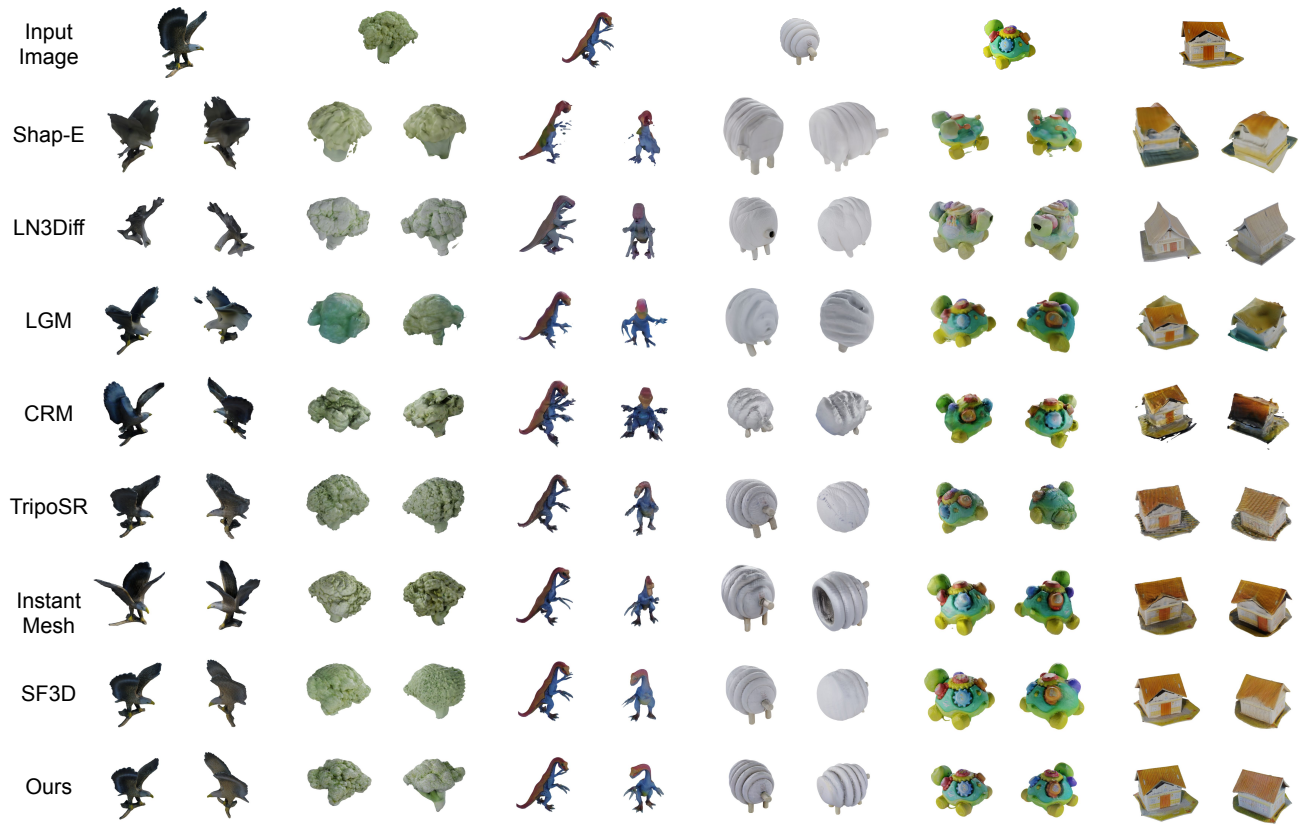

Figure 5. **Qualitative Comparison.** We compare SPAR3D to other state-of-the-art methods visually. SPAR3D not only aligns better with the visible surfaces from images, but also generates higher-quality geometries and textures for the occluded surfaces.

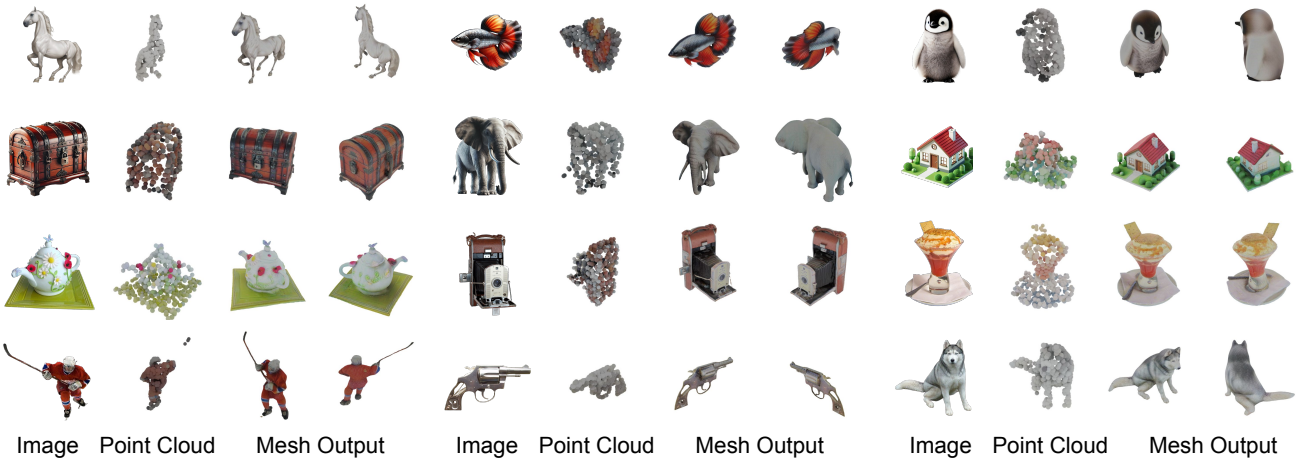

Figure 6. **Generalization Results.** We show qualitative results of SPAR3D on in-the-wild images from 2D generative models (top 2 rows) and ImageNet (bottom 2 rows). The reconstructed meshes exhibit accurate geometric structures with great textures, demonstrating a strong generalization performance of SPAR3D.

In Fig. 6, we further show qualitative results of SPAR3D on in-the-wild images. The images are either generated using SDXL [42]/Dall-E 3 [1] or from the validation set of ImageNet [14]. The high quality of the reconstructed meshes demonstrates a strong generalization performance of SPAR3D.

4.3. Editing Results

The usage of explicit point clouds as an intermediate representation enables interactive editing of the generated meshes. Users can easily alter the unseen surface of the mesh by manipulating the point cloud. In Fig. 7, we show a

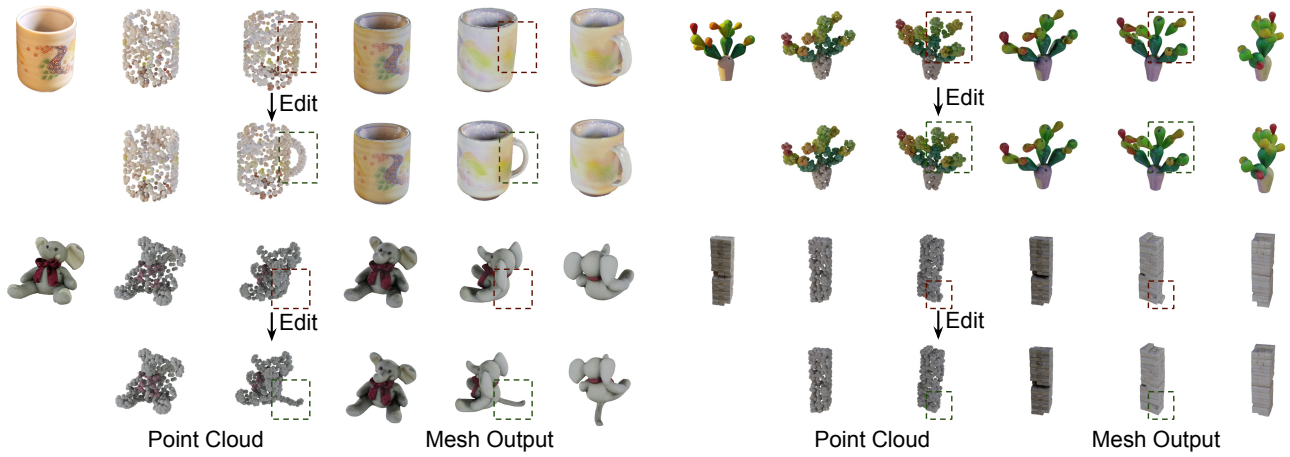

Figure 7. **Editing Results.** We show qualitative examples of interactive editing with SPAR3D. On the left two examples, we add a handle to the mug and a tail to the elephant doll by duplicating existing points. On the right two examples, we move or delete points to fix imperfections and to improve local details on the mesh. All the edits are performed in Blender within a minute.

few editing examples with SPAR3D, by either adding major object parts to the reconstruction, or improving undesirable generated details.

4.4. Ablation

We ablate the key idea of SPAR3D, the point sampling stage, which can be seen as an addition to standard regression approaches. We consider a variant of our model (SPAR3D w/o Point), where we remove the point sampling stage and make SPAR3D a full regressive model. We compare this variant with our full model on both GSO and Omniobject3D. As shown in Tab. 3, our full SPAR3D significantly outperforms the regressive variant, which validates the effectiveness of our design.

Method	CD↓	FS@0.1↑	PSNR↑	LPIPS↓
SPAR3D w/o Point	0.136	0.506	18.5	0.146
SPAR3D	**0.120**	**0.584**	**18.6**	**0.139**
SPAR3D w/o Point	0.140	0.509	17.8	0.146
SPAR3D	**0.122**	**0.587**	**17.9**	**0.140**

Table 3. **Ablation Study on GSO (top 2 rows) and Omniobject3D (bottom 2 rows).** Removing the point sampling stage leads to significant performance drop.

4.5. Analysis

We further design experiments to understand how SPAR3D works. Our key assumption when designing SPAR3D is that the two-stage design effectively separates the uncertain part (back-surface modeling) and the deterministic part (visible surface modeling) of the monocular 3D reconstruction problem. Ideally, the meshing stage should mainly rely on the input image for reconstructing the visible surface, while relying on the point cloud to generate the back surface. To see whether this is true, we design an experiment where we artificially use point clouds that conflict with the input image. In Fig. 8, we feed the input image of a squirrel and

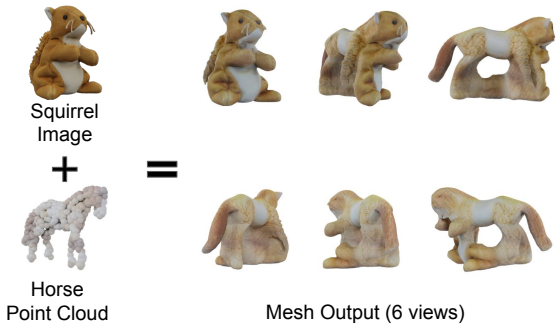

Figure 8. **Generated Mesh with Conflicting Cues.** Under conflicting cues from images and point clouds, our model reconstructs the visible surface based on the image, while generating the backside surface based on the point cloud.

the point cloud of a horse to the meshing model. As shown in the figure, the reconstructed mesh indeed aligns with the squirrel image well on the visible surface, while the back surface mainly adheres to the point cloud. This result validates our assumption.

5. Conclusion

We present SPAR3D, a simple yet effective approach for single-view 3D reconstruction. The core of our model is a two-stage design based on point sampling. We first generate a sparse point cloud via point diffusion, and then reconstruct a highly detailed mesh from both the point cloud and the image. This design enables us to take the best of regression-based and generative modeling. Evaluated on standard benchmarks and in-the-wild images, SPAR3D significantly outperforms previous state-of-the-art methods with a fast inference speed. We will release our model upon publication, and we hope our effort is useful for future research towards scalable generation of high-quality 3D content.

References

[1] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science. https://cdn. openai. com/papers/dall-e-3. pdf*, 2(3):8, 2023. 7

[2] Mark Boss, Varun Jampani, Raphael Braun, Ce Liu, Jonathan T. Barron, and Hendrik P.A. Lensch. Neural-pil: Neural pre-integrated lighting for reflectance decomposition. *NeurIPS*, 2021. 4

[3] Mark Boss, Zixuan Huang, Aaryaman Vasishta, and Varun Jampani. Sf3d: Stable fast 3d mesh reconstruction with uv-unwrapping and illumination disentanglement. *arXiv preprint*, 2024. 2, 4, 6, 1

[4] Brent Burley. Physically-based shading at disney. *ACM Transactions on Graphics (SIGGRAPH)*, 2012. 4, 5

[5] Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 364–381. Springer, 2020. 2

[6] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 2

[7] Hansheng Chen, Jiatao Gu, Anpei Chen, Wei Tian, Zhuowen Tu, Lingjie Liu, and Hao Su. Single-stage diffusion nerf: A unified approach to 3d generation and reconstruction. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2416–2425, 2023. 2, 3

[8] Ting Chen. On the importance of noise scheduling for diffusion models. *arXiv preprint arXiv:2301.10972*, 2023. 3

[9] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander G Schwing, and Liang-Yan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4456–4465, 2023. 2, 3

[10] Gene Chou, Yuval Bahat, and Felix Heide. Diffusion-sdf: Conditional generative modeling of signed distance functions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2262–2272, 2023. 2, 3

[11] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 2

[12] Matt Deitke, Dustin Schwenk, Jordi Salvador, Luca Weihs, Oscar Michel, Eli VanderBilt, Ludwig Schmidt, Kiana Ehsani, Aniruddha Kembhavi, and Ali Farhadi. Objaverse: A universe of annotated 3d objects. *arXiv preprint arXiv:2212.08051*, 2022. 2

[13] Congyue Deng, Chiyu Jiang, Charles R Qi, Xinchen Yan, Yin Zhou, Leonidas Guibas, Dragomir Anguelov, et al. Nerdi: Single-view nerf synthesis with language-guided diffusion as general image priors. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 20637–20647, 2023. 3

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021. 3

[16] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 5, 6

[17] Jun Gao, Tianchang Shen, Zian Wang, Wenzheng Chen, Kangxue Yin, Daiqing Li, Or Litany, Zan Gojcic, and Sanja Fidler. Get3d: A generative model of high quality 3d textured shapes learned from images. *Advances In Neural Information Processing Systems*, 35:31841–31854, 2022. 2

[18] Lin Gao, Tong Wu, Yu-Jie Yuan, Ming-Xian Lin, Yu-Kun Lai, and Hao Zhang. Tm-net: Deep generative networks for textured meshes. *ACM Transactions on Graphics (TOG)*, 40 (6):1–15, 2021. 3

[19] James AD Gardner, Bernhard Egger, and William AP Smith. Reni++ a rotation-equivariant, scale-invariant, natural illumination prior. *arXiv preprint arXiv:2311.09361*, 2023. 4

[20] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[21] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. In *International Conference on Machine Learning*, pages 11808–11826. PMLR, 2023. 3

[22] Jon Hasselgren, Nikolai Hofmann, and Jacob Munkberg. Shape, Light, and Material Decomposition from Images using Monte Carlo Rendering and Denoising. *Adv. Neural Inform. Process. Syst. (NeurIPS)*, 2022. 5

[23] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022. 3

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020. 3

[25] Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*, 2023. 2

[26] Zixuan Huang, Varun Jampani, Anh Thai, Yuanzhen Li, Stefan Stojanov, and James M Rehg. Shapeclipper: Scalable 3d shape learning from single-view images via geometric and clip-based consistency. In *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2023. 2

[27] Zixuan Huang, Justin Johnson, Shoubhik Debnath, James M Rehg, and Chao-Yuan Wu. Pointinfinity: Resolution-invariant point diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10050–10060, 2024. 2, 3, 4, 5

[28] Zixuan Huang, Stefan Stojanov, Anh Thai, Varun Jampani, and James M Rehg. Zeroshape: Regression-based zero-shot shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10061–10071, 2024. 2

[29] Le Hui, Rui Xu, Jin Xie, Jianjun Qian, and Jian Yang. Progressive point cloud deconvolution generation network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 397–413. Springer, 2020. 2

[30] Heewoo Jun and Alex Nichol. Shap-e: Generating conditional 3d implicit functions. *arXiv preprint arXiv:2305.02463*, 2023. 2, 3, 6

[31] Roman Klokov, Edmond Boyer, and Jakob Verbeek. Discrete point flow networks for efficient point cloud generation. In *European Conference on Computer Vision*, pages 694–710. Springer, 2020. 3

[32] Yushi Lan, Fangzhou Hong, Shuai Yang, Shangchen Zhou, Xuyi Meng, Bo Dai, Xingang Pan, and Chen Change Loy. Ln3diff: Scalable latent neural fields diffusion for speedy 3d generation. *arXiv preprint arXiv:2403.12019*, 2024. 2, 3, 6

[33] Muheng Li, Yueqi Duan, Jie Zhou, and Jiwen Lu. Diffusion-sdf: Text-to-shape via voxelized diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12642–12651, 2023.

[34] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. *arXiv preprint arXiv:2311.07885*, 2023.

[35] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9298–9309, 2023. 2

[36] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453*, 2023. 2, 3

[37] Luke Melas-Kyriazi, Iro Laina, Christian Rupprecht, and Andrea Vedaldi. Realfusion: 360deg reconstruction of any object from a single image. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8446–8455, 2023. 3

[38] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2

[39] Paritosh Mittal, Yen-Chi Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 306–315, 2022. 3

[40] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2, 3, 4

[41] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 4

[42] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 7

[43] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. In *The Eleventh International Conference on Learning Representations*, 2023. 3

[44] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common objects in 3d: Large-scale learning and evaluation of real-life 3d category reconstruction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10901–10911, 2021. 2

[45] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022. 3

[46] Tianchang Shen, Jun Gao, Kangxue Yin, Ming-Yu Liu, and Sanja Fidler. Deep marching tetrahedra: a hybrid representation for high-resolution 3d shape synthesis. *Advances in Neural Information Processing Systems*, 34:6087–6101, 2021. 4

[47] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. Zero123++: a single image to consistent multi-view diffusion base model, 2023. 2

[48] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512*, 2023. 2, 3

[49] Jaehyeok Shim, Changwoo Kang, and Kyungdon Joo. Diffusion-based signed distance fields for 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20887–20897, 2023.

[50] J Ryan Shue, Eric Ryan Chan, Ryan Po, Zachary Ankner, Jiajun Wu, and Gordon Wetzstein. 3d neural field generation using triplane diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20875–20886, 2023. 2, 3

[51] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. 3

[52] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2974–2983, 2018. 2

[53] Zeyi Sun, Ye Fang, Tong Wu, Pan Zhang, Yuhang Zang, Shu Kong, Yuanjun Xiong, Dahua Lin, and Jiaqi Wang. Alphaclip: A clip model focusing on wherever you want, 2023. 4

[54] Junshu Tang, Tengfei Wang, Bo Zhang, Ting Zhang, Ran Yi, Lizhuang Ma, and Dong Chen. Make-it-3d: High-fidelity 3d creation from a single image with diffusion prior. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 22819–22829, 2023. 3

[55] Jiaxiang Tang, Zhaoxi Chen, Xiaokang Chen, Tengfei Wang, Gang Zeng, and Ziwei Liu. Lgm: Large multi-view gaussian model for high-resolution 3d content creation. *arXiv preprint arXiv:2402.05054*, 2024. 2, 6

[56] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, , Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2, 5, 6

[57] Diego Valsesia, Giulia Fracastoro, and Enrico Magli. Learning localized generative models for 3d point clouds via graph convolution. In *International conference on learning representations*, 2018. 2

[58] Eric Veach. *Robust Monte Carlo Methods for Light Transport Simulation*. PhD thesis, Stanford University, 1997. 4

[59] Bruce Walter, Stephen R. Marschner, Hongsong Li, and Kenneth E. Torrance. Microfacet models for refraction through rough surfaces. *Eurographics Symposium on Rendering*, 2007. 5

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

[61] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European conference on computer vision (ECCV)*, pages 52–67, 2018. 2

[62] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. Crm: Single image to 3d textured mesh with convolutional reconstruction model. *arXiv preprint arXiv:2403.05034*, 2024. 2, 4, 6

[63] Chao-Yuan Wu, Justin Johnson, Jitendra Malik, Christoph Feichtenhofer, and Georgia Gkioxari. Multiview compressive coding for 3d reconstruction. *arXiv preprint arXiv:2301.08247*, 2023. 2

[64] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. *Advances in neural information processing systems*, 30, 2017. 2

[65] Tong Wu, Jiarui Zhang, Xiao Fu, Yuxin Wang, Jiawei Ren, Liang Pan, Wayne Wu, Lei Yang, Jiaqi Wang, Chen Qian, et al. Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. *arXiv preprint arXiv:2301.07525*, 2023. 5, 6

[66] Zhijie Wu, Xiang Wang, Di Lin, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Sagnet: Structure-aware generative network for 3d-shape modeling. *ACM Transactions on Graphics (TOG)*, 38(4):1–14, 2019. 3

[67] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191*, 2024. 2, 4, 5, 6

[68] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *Advances in neural information processing systems*, 32, 2019. 2

[69] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4541–4550, 2019. 3

[70] Lior Yariv, Omri Puny, Oran Gafni, and Yaron Lipman. Mosaic-sdf for 3d generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4630–4639, 2024. 2, 3

[71] Kai Zhang, Sai Bi, Hao Tan, Yuanbo Xiangli, Nanxuan Zhao, Kalyan Sunkavalli, and Zexiang Xu. Gs-lrm: Large reconstruction model for 3d gaussian splatting. *arXiv preprint arXiv:2404.19702*, 2024. 2

[72] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 4

[73] Zibo Zhao, Wen Liu, Xin Chen, Xianfang Zeng, Rui Wang, Pei Cheng, BIN FU, Tao Chen, Gang YU, and Shenghua Gao. Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. 2, 3

Figure 9. **Decomposition and Relighting Results.** We show decomposed albedo and relighting results of SPAR3D in comparison with SF3D. The albedo estimated by SPAR3D has less baked-in lighting compared with SF3D and results in better relighting outcomes.

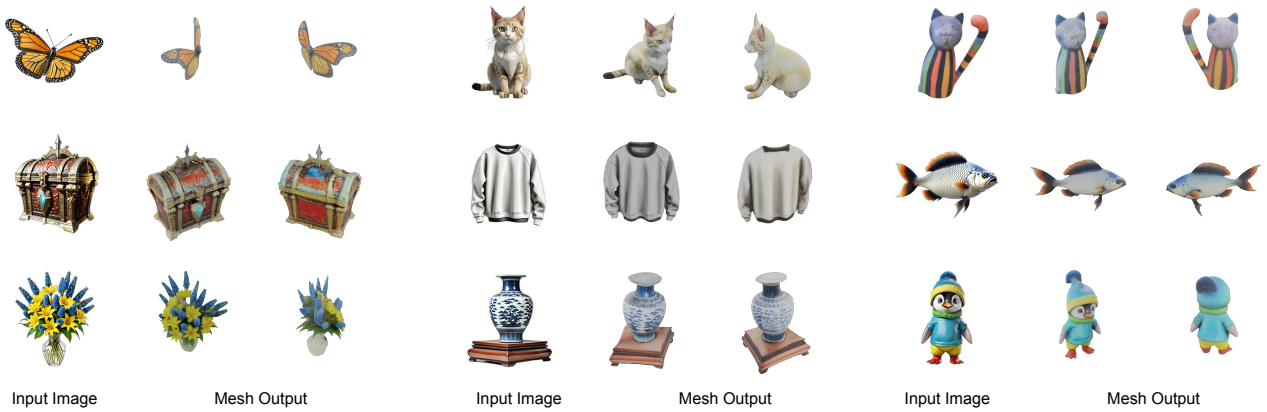

Figure 10. **Additional In-the-wild Results.** We show additional results of SPAR3D on in-the-wild images. The reconstructed meshes achieve high fidelity and exhibit great surface details.

This appendix is structured as follows: in Appendix A we discuss the limitations of our approach; in Appendix B we provide two additional illustrations of our architecture; in Appendix C we show decomposition and relighting results of our model in comparison with SF3D [3]; in Appendix D we present additional in-the-wild results.

A. Limitations

The main limitations of SPAR3D are twofold. First, the point clouds generated during the point sampling stage occasionally exhibit artifacts, such as small surface spikes or detached parts. While these imperfections can typically be remedied through SPAR3D's editing capabilities with minimal effort (see Fig. 7 in the main paper), exploring more principled solutions (e.g. improving the denoiser design or diffusion samplers) could further enhance the utility and robustness of our method.

Second, although SPAR3D learns material decomposition during training, the accuracy of these decompositions can sometimes be suboptimal. This limitation is primarily due to the inherent ambiguity of inverse rendering from a single image, especially when learned in an unsupervised manner. Unsupervised decomposition learning is useful given the scarcity of 3D assets containing high-quality Physically Based Rendering (PBR) materials and is scalable to real-world multi-view datasets. However, investigating semi-supervised learning techniques may offer a pathway to more plausible material estimations in future work.

B. Additional Illustrations of our Architecture

We show additional illustrations of our point cloud denoiser and our meshing model in Fig. 11 and Fig. 12. We hope these illustrations facilitate a better understanding of our architecture.

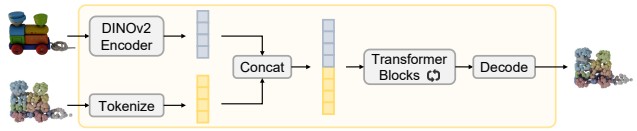

Figure 11. **Point Cloud Denoiser Architecture.** We illustrate the architecture of our point cloud denoiser. The point cloud denoiser takes the noisy point cloud and the image as input, and produces a denoised point cloud. The image and the noisy point cloud are encoded as latent vectors and concatenated together. The concatenated latent vectors are processed by a set of transformer blocks and decoded as the denoised point cloud.

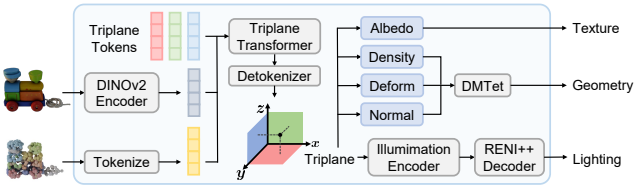

Figure 12. **Meshing Model Architecture.** We illustrate the architecture of our meshing model, which takes the point cloud and the image as input, and produces a textured mesh and an environment map as output. Specifically, the meshing model first encodes the image and the point cloud as latent vectors. The learnable triplane tokens are then processed by the triplane transformer conditioned on the latent vectors. We query the triplane with MLPs to obtain albedo, density, vertex deformation and surface normal, which are converted to a textured mesh using DMTet. The triplane also produces an environment map using the illumination prior from RENI++. The metallic and roughness values are estimated from the image directly and are omitted here for simplicity.

C. Decomposition Results

We show decomposition and relighting results of SPAR3D in comparison with SF3D, which is a full regressive method. As shown in Fig. 9, our estimated albedo often has less baked-in lighting artifacts compared with SF3D, which improves the quality of relighting under different illumination conditions.

D. Additional In-the-wild Results

We present additional reconstruction results on in-the-wild images. In Fig. 10, we show the reconstructions of SPAR3D on images from 3D-Arena (Ebert, 2024). On this data source, SPAR3D also achieves high reconstruction quality. This further validates the strong generalization ability of SPAR3D.