Spurfies: Sparse-view Surface Reconstruction using Local Geometry Priors

Kevin Raj[1] Christopher Wewer[1] Raza Yunus[1] Eddy Ilg[2] Jan Eric Lenssen[1]
[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany
[2]Saarland University, Saarland Informatrics Campus, Germany
{kraj,jlenssen}@mpi-inf.mpg.de

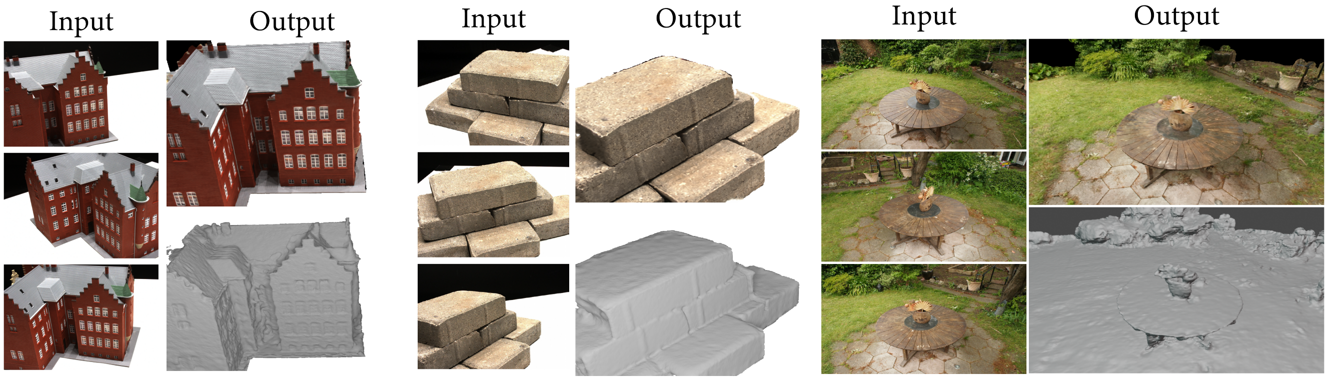

Figure 1. Spurfies leverages synthetic data to learn local surface priors for surface reconstruction from few images. Our model significantly outperforms previous methods and can be applied to both bounded (DTU dataset, left) and unbounded scenes (Mip-NeRF360, right).

Abstract

We introduce Spurfies[1], a novel method for sparse-view surface reconstruction that disentangles appearance and geometry information to utilize local geometry priors trained on synthetic data. Recent research heavily focuses on 3D reconstruction using dense multi-view setups, typically requiring hundreds of images. However, these methods often struggle with few-view scenarios. Existing sparse-view reconstruction techniques often rely on multi-view stereo networks that need to learn joint priors for geometry and appearance from a large amount of data. In contrast, we introduce a neural point representation that disentangles geometry and appearance to train a local geometry prior using a subset of the synthetic ShapeNet dataset only. During inference, we utilize this surface prior as additional constraint for surface and appearance reconstruction from sparse input views via differentiable volume rendering, restricting the space of possible solutions. We validate the effectiveness of our method on the DTU dataset and demonstrate that it outperforms previous state of the art by 35% *in surface quality while achieving competitive novel view synthesis quality. Moreover, in contrast to previous works, our method can be applied to larger, unbounded scenes, such as Mip-NeRF360.*

1. Introduction

3D surface reconstruction just from a small number of 2D input images has been an important goal in computer vision in the recent years. While we entered an era in which we can get high-quality reconstructions from a large number of carefully captured images through recent advances in SDF-based neural fields [35, 38] or surfels in the form of 2D Gaussians [15], we still lack the capabilities to perform this task from a sparse set of views. The key challenge in this setup is the under-constrained nature of the optimization problem at hand: there are simply too many potential combinations of geometry and appearance that would satisfy the sparse set of given image observations.

The usual way to solve such under-constrained tasks is to introduce regularization or priors into the formulation. Typical approaches include feature consistency [16] and cues from monocular depth [50], multi-view stereo [34, 42], or structure from motion [11]. More recently, learned priors have been introduced, e.g. from pre-trained 2D diffusion models [43]. They have been shown to clearly improve re-

[1] geometric-rl.mpi-inf.mpg.de/spurfies/

construction quality from a few observations, being able to hallucinate missing appearance details. However, a clear limitation of diffusion-based approaches is that they require large amounts of training data in order to generalize to the diverse domain of appearances that 3D scenes can have.

In this work, we make the key observation that the space of local surface geometry is much less diverse than that of surface appearance and make the following three key assumptions: (1) we argue that much less training data is needed to learn a useful surface geometry prior than to learn a prior about appearance, (2) that synthetic data can serve well as training data for geometry as the domain gap between real and synthetic data for geometry is smaller, and (3) that a geometry prior already serves as a useful constraint to appearance reconstruction from only a few observations. To verify, we design an architecture that explicitly disentangles geometry from appearance modeling, while still jointly optimizing both during the reconstruction process. With this architecture, we are able to pre-train the geometry part of our network on synthetic data and enable it to learn the space of common surface structures. During test-time optimization, the pre-trained geometry branch then provides regularization for the joint SDF and appearance reconstruction of the scene.

As neural field representation we opt for a point-based neural field [46], with points initialized by DUSt3R [37], which we extend to model signed distance functions (SDFs) instead of density. On the neural point cloud we store separate features for SDFs and appearance in local neighborhoods. The representation is rendered via volume rendering by aggregating and processing neighboring features for sample points along the ray. Signed distance and radiance are predicted by a pre-trained geometry and a per-scene appearance decoder, respectively. We show that Spurfies

- outperforms previous methods in sparse-view surface reconstruction quality by a large margin,
- reaches state-of-the-art novel view synthesis quality, and
- can be applied to larger, unbounded scenes,

by relying on recent advances in point reconstruction [37] and a local surface prior trained purely on synthetic data.

2. Related Work

2.1. Surface Reconstruction from Images

Dense Views Since NeRF [26] revolutionized inverse graphics by parameterizing implicit radiance fields as neural networks trained via differentiable volume rendering, there have been many approaches to directly reconstruct 3D surfaces in a similar manner. NeuS [35] and VolSDF [47] reformulate the density of radiance fields based on a signed distance function to implicitly learn a surface representation via inverse rendering. Follow-up works like MonoSDF [50], Geo-Neus [11], RegSDF [51], and NeuRIS [34] leverage depth and normal estimation, sparse geometry from structure from motion (SfM), and photometric consistency constraints in multi-view stereo for additional regularization. Another line of research [21, 38, 44, 48, 54] focuses on accelerating training and rendering of neural surface representations by incorporating explicit data structures like multi-resolution hash grids [27]. To this end, following the breakthrough of fully explicit 3D Gaussian Splatting [19], SuGaR [14], 2D GS [15], and DN-Splatter [33] try to tame Gaussians for accurate mesh reconstruction. While high-fidelity surface reconstruction is now possible with training in the matter of minutes, all these approaches assume dense captures of the scene in the form of a large number of input images.

Sparse Views To reduce the number of input views and therefore enable applications to more real-world settings, geometric priors can further constrain the space of possible reconstructions. Approaches for generalizable surface reconstruction [23, 24, 31, 42, 45] learn such priors on additional training scenes. Specifically, SparseNeuS [24] fuses 2D features into a cascade of coarse-to-fine voxel grids for end-to-end SDF prediction from sparse input views. These predictions can be used as initialization for per-scene fine-tuning. VolRecon [31] and ReTR [23] leverage transformer architectures for rendering conditioned on similar fused voxel features. C2F2NeUS [45] replaces imprecise, regular voxel grids with view-based cascade cost frustums. S-VolSDF [42] combines VolSDF [47] with CasMVSNet [13] by supervising the former with probability volumes from the latter, which in turn produces depth hypotheses used for sampling in a finer MVS stage. All of these approaches train a surface prior on image data, limiting generalization to scenes close to the training distribution.

NeuSurf [16] proposes a two-stage approach by first training an unsigned distance field given sparse point clouds from SfM [55], which is used as a coarse geometric constraint for fitting a fine signed distance field in the second stage. However, it relies on global representations resulting in limited efficiency, scalability, and level of detail. In contrast to that, our method trains a local geometric prior on general, synthetic, texture-less meshes, requiring little data for generalization. We achieve this by decomposing our architecture into geometry and appearance branches build on a point-based representation. Given the recent breakthrough of DUSt3R [37] for stereo 3D point cloud reconstruction via large-scale (pre-)training of transformer architectures, point-based representations render an elegant solution to obtain watertight surfaces.

2.2. Point-based 3D Representations

Point-based 3D representations [2, 19, 30, 46] equip point clouds obtained from RGB-D sensors, MVS, or SfM pipelines with additional learnable parameters. PointNeRF [46] decodes these features with small MLPs to represent a radiance field fitted on dense input views. Subsequent works employ similar representations for relighting and deformation [22], single-view reconstruction [39], and object generation [32]. 3D GS [19] and follow-ups for surface [14, 15, 33], generalizable [6, 8, 40], or non-rigid [9, 25, 41] reconstruction use optimizable Gaussian parameters and spherical harmonic coefficients for view-dependent RGB, enabling much faster rasterization-based rendering without any neural components. However, the explicit nature is hardly compatible with learning of neural priors without loosing the efficiency advantages. Therefore, our architecture is based on neural point clouds with individual geometry and appearance decoding branches for data-efficient pre-training of a local surface prior only.

3. Prerequisites

Given a sparse set of $N = 3$ RGB images $\mathcal{I} = \{\mathbf{I}_i \,|\, i \in 1,\ldots,N\}$, known camera poses $\mathcal{V} = \{\mathbf{v}_i \,|\, i \in 1,\ldots,N\}$ and a point-cloud $\mathcal{P} = \{\mathbf{p}_i \,|\, i \in 1,\ldots,M\}$ that we obtain from DUSt3R [37], we aim to reconstruct a surface $\mathcal{S}$. In the following, we denote all predictions from our model with the $\hat{}$-symbol. We choose a continuous implicit surface representation using a signed distance field $\hat{s}(\mathbf{x})$ and use the approach from VolSDF [47] by attaching a volumetric radiance field with density $\hat{\sigma}(\mathbf{x})$ and radiance $\hat{r}(\mathbf{x})$ to leverage its graceful optimization properties via backpropagation and gradient descent. The density is derived from the signed distance via:

$$\hat{\sigma}(\mathbf{x}) = \begin{cases} \frac{1}{2}\exp\left(\frac{\hat{s}(\mathbf{x})}{\beta}\right)\cdot\alpha & \text{if } \hat{s}(\mathbf{x}) \leq 0 \\ \left(1 - \frac{1}{2}\exp\left(-\frac{\hat{s}(\mathbf{x})}{\beta}\right)\right)\cdot\alpha & \text{if } \hat{s}(\mathbf{x}) > 0\,, \end{cases} \tag{1}$$

where α and β are learnable parameters that control the sharpness of the surface. For a given ray $\mathbf{x}(t) = \mathbf{o} + t\cdot\mathbf{d}$, we use importance sampling to sample points with intervals Δ_i to obtain density $\hat{\sigma}_i$ and radiance $\hat{r}_i$. We then compute the final pixel colors $\hat{C}_{u,v}$ for pixel (u, v) via transmittance computation and ray accumulation from NeRF [26, 47].

4. Spurfies

As the combination of signed distance fields with volumetric rendering introduced in the last paragraph has shown to provide high-quality surface reconstructions from many observations, the question is how to enable sparse-view reconstruction by representing $\hat{s}(\mathbf{x})$ and $\hat{r}(\mathbf{x})$ with a data prior that was obtained from a training dataset. We introduce our technique that we name *Spurfies* (Figure 2). It extends volumetric rendering by two key components: **(1) A neural point representation** (Section 4.1), where each point $\mathbf{p}_i$ is associated with geometry and appearance latent codes $\mathbf{f}_i = (\mathbf{f}_i^g, \mathbf{f}_i^a)$ that enable local processing and effective disentanglement of both modalities. **(2) A local geometry prior** (Section 4.2) to mitigate the shape-radiance ambiguity inherent in sparse-view reconstruction [53]. We argue that there is less variety in geometry than in appearance such that disentanglement in combination with a strong geometry prior is key. Unlike previous works that rely on multi-view stereo data during training, it is sufficient to obtain such a local prior from a synthetic object dataset (ShapeNet [5]).

4.1. Neural Point Representation

We use a distributed representation that is applicable to whole scenes by attaching disentangled latent codes $\mathbf{f}_i^g, \mathbf{f}_i^a$ to the points $\mathbf{p}_i$ to form a neural point cloud. To regress $\hat{s}(\mathbf{x})$ and $\hat{r}(\mathbf{x})$ at a query location $\mathbf{x}$, we use K neural points $\mathbf{p}_k$ within a local neighborhood of radius R.

Local Processing. We map the point features to the relative query location:

$$\mathbf{f}_{kx}^g = G_{\text{LP}}(\mathbf{f}_k^g, \mathbf{x} - \mathbf{p}_k)\,, \tag{2}$$

$$\mathbf{f}_{kx}^a = A_{\text{LP}}(\mathbf{f}_k^a, \mathbf{x} - \mathbf{p}_k)\,, \tag{3}$$

where G_{LP} and A_{LP} are two small MLPs. The resulting intermediate latent codes $\mathbf{f}_{kx}^g$ and $\mathbf{f}_{kx}^a$ encode local geometry and appearance. The use of relative positions $\mathbf{x} - \mathbf{p}_k$ makes our architecture local and translation invariant, a property that allows application on larger scenes [46].

Signed Distance Regression. We then directly map the geometry features to signed distance values and interpolate them at the query location via:

$$\hat{s}(\mathbf{x}) = \frac{\sum_{k=1}^K w_k \cdot \hat{s}_{kx}}{\sum_{k=1}^K w_k}, \quad \hat{s}_{kx} = G_{\text{REG}}(\mathbf{f}_{kx}^g), \tag{4}$$

where G_{REG} is another MLP and the weighting function $w_k = e^{-\lambda\|\mathbf{x}-\mathbf{p}_k\|_2^2}$ is modelled using radial basis functions (RBFs). We use Gaussians as the distance kernel with variance λ, which controls the influence of the neighboring neural points $\mathbf{p}_k$. In addition, the RBF weights are bounded between $[0, 1]$ providing stability during the training process.

Radiance Regression. For radiance prediction, we first interpolate the local scene appearance encoded by neighboring points and subsequently regress the radiance with a small MLP A_{REG}:

$$\hat{r}(\mathbf{x}, \mathbf{d}) = A_{\text{REG}}(\mathbf{f}_x^a, \mathbf{d}), \quad \mathbf{f}_x^a = \frac{\sum_{k=1}^K w_k \cdot \mathbf{f}_{kx}^a}{\sum_{k=1}^K w_k}\,, \tag{5}$$

with view direction $\mathbf{d}$, following previous work [46]. The interpolation weights w_k are computed using the same RBF

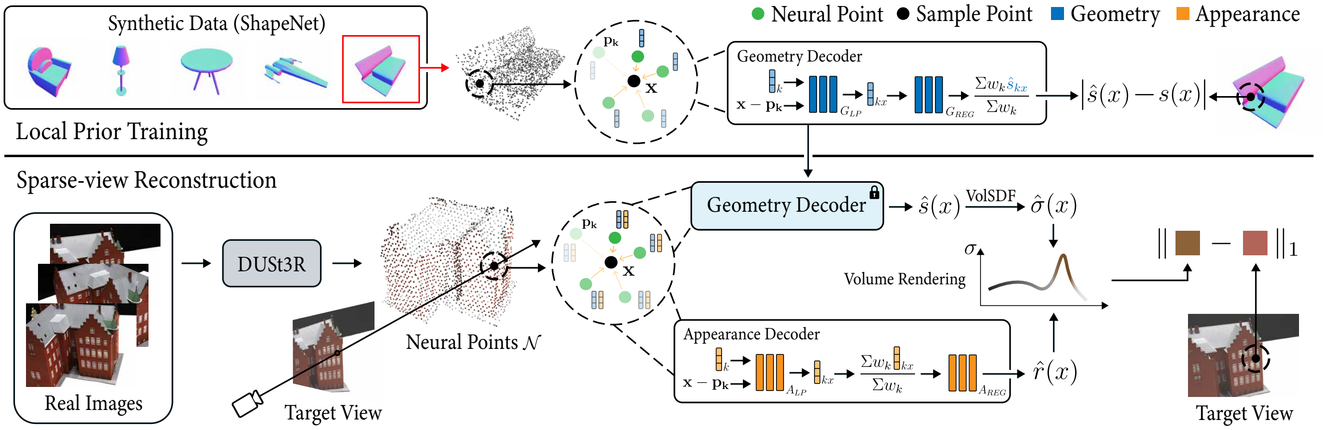

Figure 2. **Method overview:** 1) **Preprocess:** given a sparse set of input views, we make use of DUSt3R [37] to predict points $\mathcal{P}$. **Representation:** The points serve as basis for a neural point representation that stores disentangled features $\mathbf{f}^a$, $\mathbf{f}^g$ for geometry and appearance on each point. **Local Prior** (top): We learn a local geometry prior G_{LP} & G_{REG} over a subset of shapes from the synthetic ShapeNet dataset [5] by optimizing to predict ground truth SDF. 3) **Spurfies** (bottom): We make use of the prior for surface reconstruction from sparse images, only optimizing the latent codes $\mathbf{f}^a$, $\mathbf{f}^g$ and the color MLPs A_{LP} & A_{REG} to reconstruct images via volume rendering.

kernel as in the SDF regression, ensuring smooth blending of local appearance information. Finally, $\hat{s}(\mathbf{x})$ and $\hat{r}(\mathbf{x})$ are rendered via volumetric rendering (c.f. Sec. 3).

4.2. Local Priors

To address the shape-radiance ambiguity [53] inherent in sparse-view settings, we propose a novel approach that diverges from traditional multi-view stereo (MVS) based methods. While previous sparse-view reconstruction techniques [24, 31] often rely on MVS networks with cost volumes and use 3D CNNs to regress SDF and color, we leverage the disentangled nature of our proposed neural point representation. Notably, in our approach the geometry and appearance branches are completely separate. This enables us to learn generalizable local geometry priors in the networks G_{LP} and G_{REG} on a synthetic dataset with available ground-truth geometry. During inference, we keep the geometry networks frozen and only optimize the geometry latent codes together with the full appearance branch.

4.2.1 Training

To train the geometry network effectively, we require a set of query points $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^{S}$ with corresponding ground-truth signed distances $s(\mathbf{x}_i)$. For each training mesh, we also sample a neural point-cloud $\mathcal{N} = \{(\mathbf{p}_i, \mathbf{f}_i^g)\}_{i=1}^{\text{M}}$. It consists of global point coordinates and randomly initialized geometry latent codes for each point.

We follow the procedure outlined in [4, 29] to sample the query points and evaluate the ground-truth SDF. To enhance the network's robustness to noise, we add Gaussian noise to the neural point locations with variance 0.005. Our training dataset is a small subset of ShapeNet [5], consisting of five object classes: sofas, chairs, planes, tables, and lamps. The local prior is trained on a total of 50 objects, with 10 objects randomly selected from each class.

This training approach ensures that our geometry network learns a robust and generalizable representation of local shapes using SDF ground truth as opposed to using incomplete depth maps or unreliable inverse renderings, which require much more diverse training data and are significantly more expensive. A key contribution is to show that a small and accurate synthetic dataset is sufficient to learn a prior that generalizes to general, large scale scenes.

4.2.2 Loss Functions

We train the local prior using the following loss functions.

SDF Loss. We take the absolute difference between the ground-truth $s(\mathbf{x}_i)$ and the predicted SDF value $\hat{s}(\mathbf{x}_i)$ [27]. We further weight the loss closer to the surface by scaling with the inverse of the ground-truth distance and add a small number to prevent division by zero:

$$\mathcal{L}_{\text{SDF}} = \sum_{i=1}^{S} \frac{|s(\mathbf{x}_i) - \hat{s}(\mathbf{x}_i)|}{|s(\mathbf{x}_i)| + \epsilon}. \tag{6}$$

Eikonal Loss. We use the Eikonal loss to regularize the Signed Distance Function (SDF) [12] and ensure the properties of a true distance field:

$$\mathcal{L}_{\text{Eik}} = \sum_{i=1}^{S} \left(\|\nabla \hat{s}(\mathbf{x}_i)\|_2 - 1\right)^2. \tag{7}$$

Scan ID	**21**	**24**	**34**	**37**	**38**	**40**	**82**	**106**	**110**	**114**	**118**	**Mean CD**
Points2Surf [10]	3.73	2.85	2.55	5.13	3.85	2.41	2.30	3.95	3.33	2.37	2.84	3.21
DUSt3R [37] + Poiss. [18]	3.31	2.12	2.00	4.25	3.28	2.59	2.48	4.28	3.85	2.68	3.32	3.11
CAP-UDF [55]	2.72	1.53	1.45	4.05	2.78	1.81	4.22	3.51	3.83	2.24	3.65	2.89
NeuS [35]	4.52	3.33	3.03	4.77	1.87	4.35	1.89	4.18	5.46	1.09	2.40	3.36
VolSDF [47]	4.54	2.61	1.51	4.05	1.27	3.58	3.48	2.62	2.79	0.52	1.10	2.56
SuGar [14]	2.71	2.04	2.14	4.01	2.90	2.45	4.68	3.82	3.28	2.44	2.66	3.01
SparseNeus$_{ft}$ [24]	3.73	4.48	3.28	5.21	3.29	4.21	3.30	2.73	3.39	1.40	2.46	3.41
VolRecon [31]	3.05	3.30	2.27	4.36	2.51	3.24	3.30	3.10	3.58	1.86	3.68	3.11
S-VolSDF [42]	3.18	2.95	2.19	3.40	2.30	2.69	2.69	1.60	1.48	1.21	1.16	2.26
NeuSurf [16]	3.22	2.42	1.38	2.61	1.72	3.46	2.68	1.44	2.42	**0.61**	**0.87**	2.08
Ours	**2.36**	**1.12**	**0.83**	**2.39**	**1.14**	**1.55**	**1.67**	**1.26**	**1.14**	**0.61**	0.94	**1.36**

Table 1. **Quantitative mesh reconstruction comparison** based on Chamfer Distance (mm) ↓ on the DTU dataset. On average, we improve on the previous best method by **35%** (**1.36** vs. 2.08 CD). We also outperform various baselines that work on points only.

	PSNR ↑	**SSIM ↑**	**LPIPS ↓**
IBRNet$_{ft}$ [36]	15.71	0.75	0.29
MVSNeRF [7]	18.37	0.81	0.25
VolSDF [47]	14.18	0.62	0.35
S-VolSDF [42]	19.67	0.71	0.30
NeuSurf [16]	18.95	0.76	0.26
SuGaR [14]	**21.32**	**0.83**	0.24
Ours	20.78	0.80	**0.20**

Table 2. **Quantitative NVS comparison on DTU**. Our method reaches state-of-the-art novel view synthesis quality by being on par with SuGaR [14]. Note that SuGaR achieves much lower surface quality (c.f. Tab. 1) and shows artifacts in qualitative results (c.f. Fig. 4) in the given sparse-view setting.

Total Variation Loss. We find that regularizing the geometry latent codes, such that nearby neural points are closer in latent space, is beneficial for surface reconstruction. Thus, we introduce a total variation loss:

$$\mathcal{L}_{\text{TV}} = \sum_{i=1}^{M} \sum_{k \in \mathcal{K}(i)} \frac{\|\mathbf{f}_i^g - \mathbf{f}_k^g\|_1}{\|\mathbf{p}_i - \mathbf{p}_k\|_2} , \tag{8}$$

where $\mathcal{K}(i)$ is the local neighborhood of neural points around the point with index i. This results in smooth changes to the regressed local surface (c.f. Sec. 5.2).

Training Objective. In total, we find the optimal parameters of G_{LP}, G_{REG}, and $\mathbf{f}_i^g$ by minimizing:

$$\mathcal{L}_{\text{Prior}} = \mathcal{L}_{\text{SDF}} + \lambda_{\text{TV}} \cdot \mathcal{L}_{\text{TV}} + \lambda_{\text{Eik}} \cdot \mathcal{L}_{\text{Eik}} , \tag{9}$$

where the factors λ weigh the individual loss terms.

4.3. Sparse-view Reconstruction

We incorporate the knowledge learned from the local prior to the volume rendering pipeline. We achieve this by keeping the geometry network fixed and only optimizing latent codes and the appearance network using the following established rendering losses for sparse-view reconstruction. For details about loss functions from previous works and the individual weights we refer to the appendix.

Rendering Loss. To ensure that our reconstructed model accurately reproduces the input images, we employ a rendering loss. It is defined as the L_1 distance between the rendered and ground-truth images:

$$\mathcal{L}_{\text{Ren}} = \sum_{i,u,v \in \mathcal{I}} \|\hat{C}_{i,u,v} - C_{i,u,v}\|_1 ,$$

where $\hat{C}_{i,u,v}$ is the pixel color of pixel (u, v) of image i, and $C_{i,u,v}$ the respective ground truth value.

Feature Consistency Loss. We use a feature consistency loss $\mathcal{L}_{\text{FC}}$ from NeuSurf [16]. It minimizes the differences in VisMVSNet [52] features between the projections of surface points onto different views. This encourages consistency across views and support reconstructing finer surface details. This loss minimizes the differences in feature space between the projections of each surface point onto different views. By doing so, we encourage consistency across views and support reconstructing finer surface details.

Pseudo SDF Loss. We adapt a pseudo SDF loss $\mathcal{L}_{\text{Pseu}}$ from previous work [11] for our approach. It is minimized when the neural points $\mathbf{p}_i$ lie on the zero level set of the predicted SDF $\hat{s}$, i.e., on the surface.

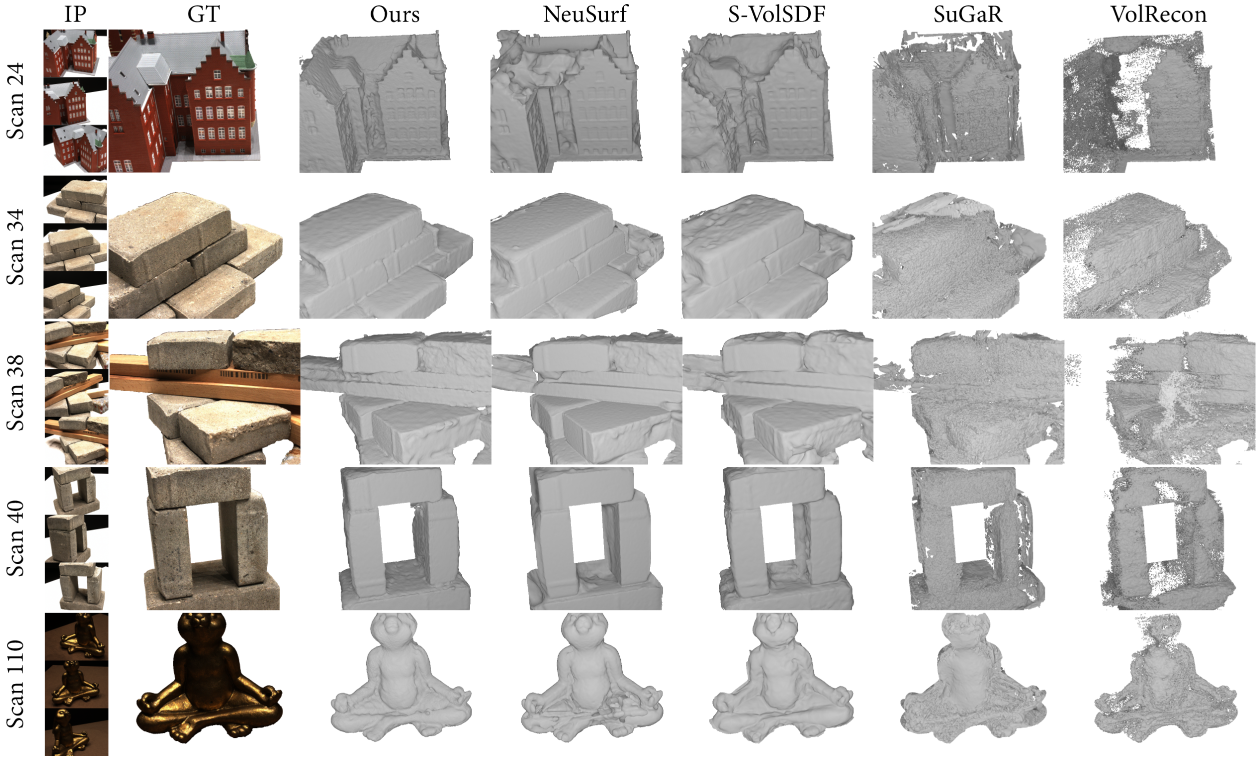

Figure 3. **Qualitative mesh reconstruction comparison on DTU**. Compared to previous state-of-the-art sparse-view methods, our reconstruction demonstrates superior completeness in regions with less view overlap. Our closest competitor is NeuSurf, which also reconstructs high quality surfaces on the object-centric DTU scenes. However, it fails to generalize to larger scenes (c.f. Fig. 5).

Optimization Objective. In total, we find the optimal parameters of A_{LP}, A_{REG}, $\mathbf{f}_i^g$, and $\mathbf{f}_i^a$ by minimizing:

$$\mathcal{L}_{\mathrm{Rec}} = \mathcal{L}_{\mathrm{Ren}} + \lambda_{\mathrm{FC}} \cdot \mathcal{L}_{\mathrm{FC}} + \lambda_{\mathrm{Pseu}} \cdot \mathcal{L}_{\mathrm{Pseu}} + \lambda_{\mathrm{TV}} \cdot \mathcal{L}_{\mathrm{TV}}, \quad (10)$$

obtaining the final scene reconstruction.

5. Experiments

In this section, we compare Spurfies against previous methods for sparse-view surface reconstruction. We begin by introducing the used datasets, baselines and metrics, before presenting reconstruction results in Sec. 5.1 and an ablation study with additional feature analysis in Sec. 5.2.

Datasets. In our evaluation process, we utilize the DTU dataset [17], a comprehensive collection of object scans featuring 49 images per scan, each with a resolution of 576×768. The dataset provides point clouds, camera intrinsics, and poses, making it ideal for assessing sparse-view reconstruction methods. We observe that various approaches in the literature differ in their selection of input views for evaluation. Adopting the strategy employed by PixelNeRF [49], we take views 22, 25, and 28 from each scan as input, leaving 25 test views. The setup represents a dispersed view distribution with low overlap. For evaluation, we follow previous work S-VolSDF [42], using the same set of scans from the DTU dataset for our approach and all baseline methods. We also evaluate on Mip-NeRF360 [3], a dataset of larger, unbounded scenes, captured from dense views. Since none of the previous works evaluates sparse-view reconstruction on Mip-NeRF360, we randomly choose a subset of three views to test. Note that none of our test scenes are in the training data of DUSt3R [37].

Baselines. We compare our work against multiple state-of-the-art sparse-view reconstruction techniques: SparseNeuS [24], VolRecon [31], NeuSurf [16] and S-VolSDF [42]. These methods leverage MVS priors predominantly trained on multi-view datasets such as DTU [17], ensuring they operate within a similar data distribution. Additionally, we compare our approach with another point-based (dense-view) reconstruction method SuGaR [14] that employs Gaussians as the underlying 3D representation. To ensure a fair comparison, we initialize SuGaR [14] with point clouds obtained using DUSt3R [37], which improves its results. We also evaluate against surface reconstruction baselines directly from point clouds such as Poisson reconstruction [18], Points2Surf [10] and CAP-UDF [55], directly applied to the DUSt3R [37]

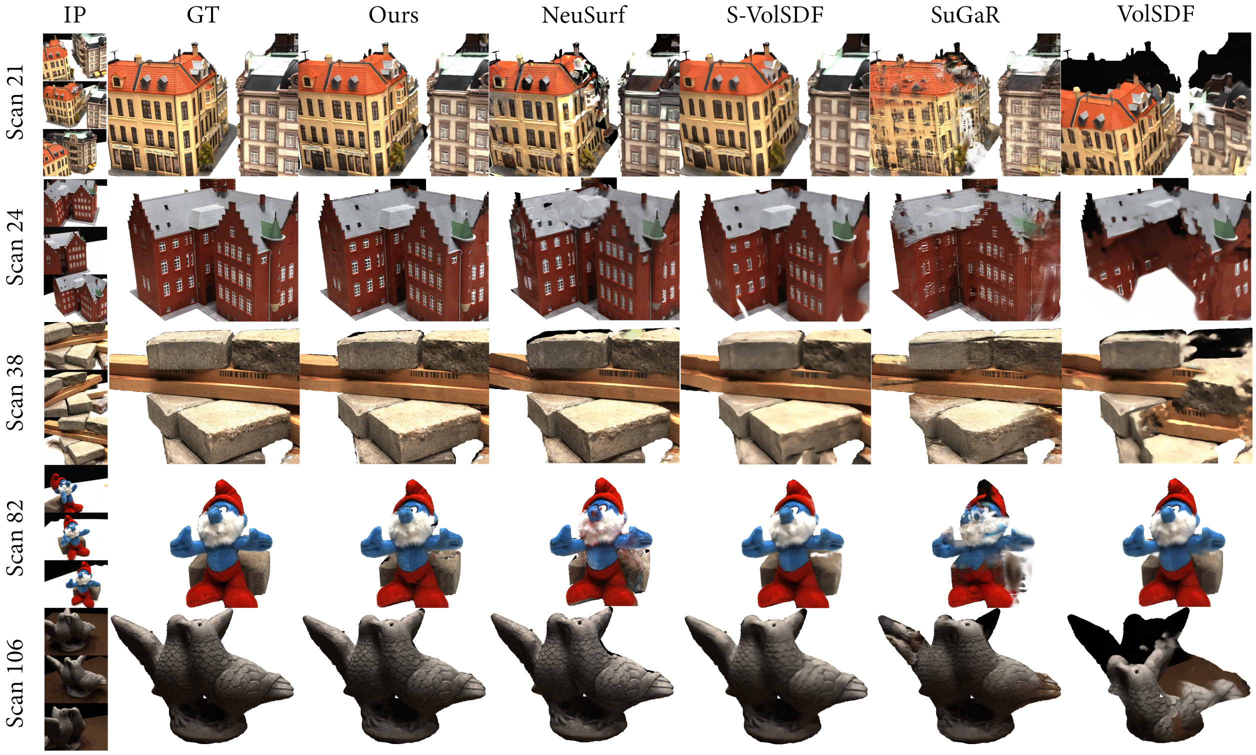

Figure 4. **Qualitative NVS comparison on DTU**. Our method demonstrates superior novel view synthesis quality relative to previous techniques. In comparison to NeuSurf [16] and S-VolSDF [42], our method produces sharper details. The Gaussian reconstruction method SuGaR [14] and volume rendering approach VolSDF [47] are not designed for sparse-view reconstruction and, therefore, show typical artifacts.

point clouds. Further, we compare against dense volume rendering methods such as VolSDF [47] and NeuS [35].

Metrics. We evaluate our technique against the baselines using both mesh reconstruction and novel view synthesis (NVS) metrics. For mesh quality assessment, we employ the Chamfer Distance (CD), which measures the accuracy of the reconstructed mesh by calculating the average distance between points sampled from the mesh and the corresponding points in the ground-truth point cloud. For NVS, we utilize Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and Learned Perceptual Image Patch Similarity (LPIPS).

Implementation Details. For the neural point representation, we built upon codebases from previous work [39, 46]. Our training process for the local prior utilizes a batch size of 5. Each batch instance comprises 40,000 randomly sampled query points, equally distributed w.r.t. positive and negative signed distances, along with 2,000 neural points. We train the geometry networks and the latent codes for 5,000 epochs. The total training time for the local prior is approximately 8 hours, utilizing a single A100 GPU.

For sparse-view reconstruction, we optimize the parameters for 100,000 iterations using the Adam optimizer [20] on a single A100 GPU, which takes 3.5 hrs to converge depending upon the number of neural points. We refer to the supplementary material for more details.

5.1. Reconstruction Results

Our method demonstrates state-of-the-art quantitative results (Table 1) in comparison to previous sparse-view and point-based reconstruction techniques. Qualitative comparisons are presented in Figure 3 and the supplement. Our reconstructions exhibit superior completeness and absence of floaters, resulting in enhanced reconstruction quality. Notably, while previous works utilize DTU for prior training and inference on the same data distribution, our prior is trained exclusively on synthetic data (Section 4.2). Leveraging our local property, we evaluate our method on unbounded scenes from Mip-NeRF360. Figure 5 presents qualitative mesh comparison results with NeuSurf [16], illustrating that our method achieves significantly more accurate reconstructions. In the absence of ground-truth point clouds, our evaluation is limited to qualitative comparisons with more results in the supplementary material.

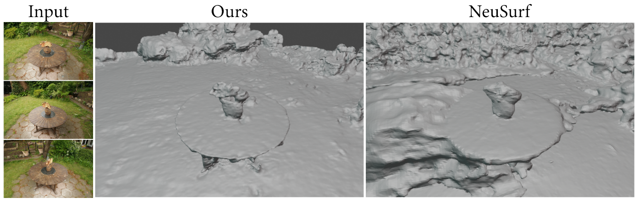

Figure 5. **Qualitative mesh reconstruction comparison on Mip-NeRF360 [3].** While NeuSurf [16] also produces good results on object-centric scenes it fails on larger, unbounded scenes. In contrast, our local prior generalizes well to Mip-NeRF360. Please refer to the appendix for additional results.

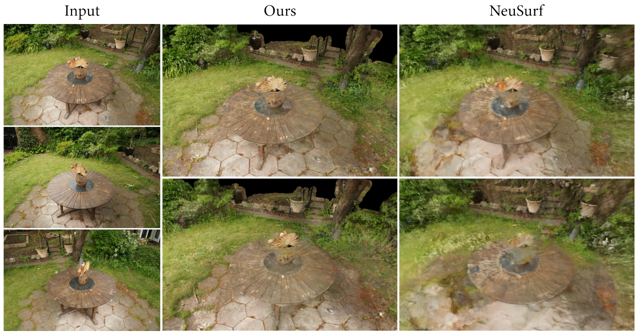

Figure 6. **Qualitative NVS comparison on Mip-NeRF360 [3].** Our geometry prior acts as effective regularization for appearance resulting in robust fitting of sparse views. Novel views from NeuSurf [16] quickly deteriorate when moving away from training views, indicating overfitting.

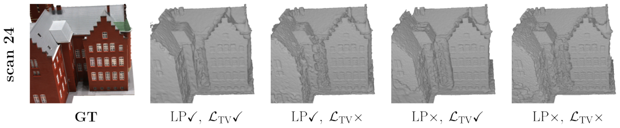
scan 24
GT
LP✓, $\mathcal{L}_{\text{TV}}$✓
LP✓, $\mathcal{L}_{\text{TV}}$×
LP×, $\mathcal{L}_{\text{TV}}$✓
LP×, $\mathcal{L}_{\text{TV}}$×

Figure 7. **Qualitative ablation** of our local prior (LP) and total variation $\mathcal{L}_{\text{TV}}$ regularization of geometry latent codes. It can be seen that both improve surface quality.

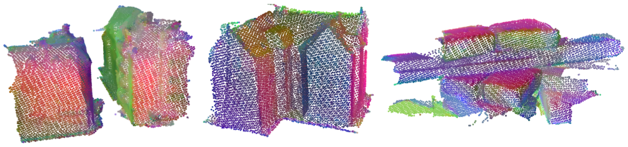

Figure 8. **Learned geometry latent codes** visualized via PCA. We observe similar features for points with same surface orientations.

Quantitative and qualitative results for novel view synthesis are presented in Table 2 and Figure 4, respectively. Our method achieves state-of-the-art NVS results among volume rendering-based techniques. The results are comparable to the Gaussian splatting-based SuGaR [14], despite SuGaR's inferior surface reconstruction quality.

Local Prior	$\mathcal{L}_{\text{TV}}$	**Mean CD** ↓
×	×	2.09
×	✓	1.91
✓	×	1.59
✓	✓	**1.36**

Table 3. **Quantitative ablation** of the proposed local prior and total variation regularization of geometry latent codes. Both contribute strongly to the reconstruction quality.

5.2. Ablation Study

We study the effect of our proposed local prior and the total variation regularization $\mathcal{L}_{\text{TV}}$ of the geometry latent codes with quantitative results in Table 3 and qualitative results in Figure 7. With the proposed point-based representation and regularizing losses from prior works (cf. Sec. 4.3), Spurfies matches the performance of the state-of-the-art baseline NeuSurf [16]. Leveraging the local prior obtained by training on synthetic data significantly enhances the level of detail in reconstructions. The total variation regularization on top mainly removes noise in under-constrained regions resulting in clean meshes.

Analysis. In Figure 8, we visualize the learned geometry latent codes using Principal Component Analysis (PCA). It reveals a correlation between similar orientations and their corresponding geometry codes. By clustering the points based on their learnt latent codes we can also extract parts that are similarly oriented (c.f. supplementary material).

6. Conclusion

We introduced Spurfies, a novel approach for sparse-view surface reconstruction that uses a distributed neural representation on point clouds and completely disentangles geometry and appearance. Our work shows that using a small subset of accurate synthetic ShapeNet data is sufficient to learn a high-quality local surface prior. We evaluated our method on the DTU and Mip-NeRF360 datasets, demonstrating state-of-the-art performance in mesh reconstruction from sparse views on DTU and that our method does generalize to large scenes. Our results indicate that Spurfies effectively mitigates the shape-radiance ambiguity inherent in sparse-view settings, producing high-quality surface reconstructions with sparse input views. Our work provides an important insight for the future direction of the field, which is that the geometry prior is most important and can effectively be modeled by disentangling it from appearance and by training it on synthetic data.

References

[1] Henrik Aanæs, Rasmus Ramsbøl Jensen, George Vogiatzis, Engin Tola, and Anders Bjorholm Dahl. Large-scale data for multiple-view stereopsis. *International Journal of Computer Vision*, 120:153–168, 2016. 3

[2] Kara-Ali Aliev, Artem Sevastopolsky, Maria Kolos, Dmitry Ulyanov, and Victor Lempitsky. Neural point-based graphics. In *ECCV*, 2020. 2

[3] Jonathan T Barron, Ben Mildenhall, Dor Verbin, Pratul P Srinivasan, and Peter Hedman. Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In *CVPR*, 2022. 6, 8, 3, 4, 5

[4] Rohan Chabra et al. Deep local shapes: Learning local sdf priors for detailed 3d reconstruction. In *ECCV*, 2020. 4

[5] 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*, 2015. 3, 4, 5

[6] David Charatan, Sizhe Li, Andrea Tagliasacchi, and Vincent Sitzmann. pixelsplat: 3d gaussian splats from image pairs for scalable generalizable 3d reconstruction. In *CVPR*, 2024. 3

[7] Anpei Chen, Zexiang Xu, Fuqiang Zhao, Xiaoshuai Zhang, Fanbo Xiang, Jingyi Yu, and Hao Su. Mvsnerf: Fast generalizable radiance field reconstruction from multi-view stereo. In *CVPR*, 2021. 5

[8] Yuedong Chen, Haofei Xu, Chuanxia Zheng, Bohan Zhuang, Marc Pollefeys, Andreas Geiger, Tat-Jen Cham, and Jianfei Cai. Mvsplat: Efficient 3d gaussian splatting from sparse multi-view images. In *ECCV*, 2024. 3

[9] Devikalyan Das, Christopher Wewer, Raza Yunus, Eddy Ilg, and Jan Eric Lenssen. Neural parametric gaussians for monocular non-rigid object reconstruction. In *CVPR*, 2024. 3

[10] Philipp Erler, Paul Guerrero, Stefan Ohrhallinger, Niloy J Mitra, and Michael Wimmer. Points2surf learning implicit surfaces from point clouds. In *ECCV*, 2020. 5, 6, 3

[11] Qiancheng Fu, Qingshan Xu, Yew-Soon Ong, and Wenbing Tao. Geo-neus: Geometry-consistent neural implicit surfaces learning for multi-view reconstruction. In *NeurIPS*, 2022. 1, 2, 5

[12] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Machine Learning and Systems*, 2020. 4

[13] Xiaodong Gu, Zhiwen Fan, Siyu Zhu, Zuozhuo Dai, Feitong Tan, and Ping Tan. Cascade cost volume for high-resolution multi-view stereo and stereo matching. In *CVPR*, 2020. 2

[14] Antoine Guédon and Vincent Lepetit. Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In *CVPR*, 2024. 2, 3, 5, 6, 7, 8

[15] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, 2024. 1, 2, 3

[16] Han Huang, Yulun Wu, Junsheng Zhou, Ge Gao, Ming Gu, and Yu-Shen Liu. Neusurf: On-surface priors for neural surface reconstruction from sparse input views. In *AAAI*, 2024. 1, 2, 5, 6, 7, 8, 3

[17] Rasmus Jensen, Anders Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. In *CVPR*, 2014. 6, 3

[18] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, 2006. 5, 6

[19] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. In *ACM TOG*, 2023. 2, 3

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv*, 2014. 7, 3, 4

[21] Zhaoshuo Li, Thomas Müller, Alex Evans, Russell H Taylor, Mathias Unberath, Ming-Yu Liu, and Chen-Hsuan Lin. Neuralangelo: High-fidelity neural surface reconstruction. In *CVPR*, 2023. 2

[22] Ruofan Liang, Jiahao Zhang, Haoda Li, Chen Yang, Yushi Guan, and Nandita Vijaykumar. SPIDR: SDF-based Neural Point Fields for Illumination and Deformation. In *arXiv*, 2022. 3

[23] Yixun Liang, Hao He, and Yingcong Chen. Retr: Modeling rendering via transformer for generalizable neural surface reconstruction. In *NeurIPS*, 2023. 2

[24] Xiaoxiao Long, Cheng Lin, Peng Wang, Taku Komura, and Wenping Wang. Sparseneus: Fast generalizable neural surface reconstruction from sparse views. In *ECCV*, 2022. 2, 4, 5, 6

[25] Jonathon Luiten, Georgios Kopanas, Bastian Leibe, and Deva Ramanan. Dynamic 3d gaussians: Tracking by persistent dynamic view synthesis. In *3DV*, 2024. 3

[26] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3

[27] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. In *ACM TOG*, 2022. 2, 4

[28] Michael Niemeyer, Jonathan T Barron, Ben Mildenhall, Mehdi SM Sajjadi, Andreas Geiger, and Noha Radwan. Regnerf: Regularizing neural radiance fields for view synthesis from sparse inputs. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5480–5490, 2022. 3

[29] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019. 4

[30] Ruslan Rakhimov, Andrei-Timotei Ardelean, Victor Lempitsky, and Evgeny Burnaev. NPBG++: Accelerating neural point-based graphics. In *CVPR*, 2022. 2

[31] Yufan Ren, Tong Zhang, Marc Pollefeys, Sabine Süsstrunk, and Fangjinhua Wang. Volrecon: Volume rendering of signed ray distance functions for generalizable multi-view reconstruction. In *CVPR*, 2023. 2, 4, 5, 6

[32] Philipp Schröppel, Christopher Wewer, Jan Eric Lenssen, Eddy Ilg, and Thomas Brox. Neural point cloud diffusion for disentangled 3d shape and appearance generation. In *CVPR*, 2024. 3

[33] Matias Turkulainen, Xuqian Ren, Iaroslav Melekhov, Otto Seiskari, Esa Rahtu, and Juho Kannala. Dn-splatter: Depth and normal priors for gaussian splatting and meshing. In *arXiv*, 2024. 2, 3

[34] Jiepeng Wang, Peng Wang, Xiaoxiao Long, Christian Theobalt, Taku Komura, Lingjie Liu, and Wenping Wang. Neuris: Neural reconstruction of indoor scenes using normal priors. In *ECCV*, 2022. 1, 2

[35] Peng Wang, Lingjie Liu, Yuan Liu, Christian Theobalt, Taku Komura, and Wenping Wang. Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction. In *NeurIPS*, 2021. 1, 2, 5, 7

[36] Qianqian Wang, Zhicheng Wang, Kyle Genova, Pratul Srinivasan, Howard Zhou, Jonathan T. Barron, Ricardo Martin-Brualla, Noah Snavely, and Thomas Funkhouser. Ibrnet: Learning multi-view image-based rendering. In *CVPR*, 2021. 5

[37] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *arXiv*, 2023. 2, 3, 4, 5, 6

[38] Yiming Wang, Qin Han, Marc Habermann, Kostas Daniilidis, Christian Theobalt, and Lingjie Liu. Neus2: Fast learning of neural implicit surfaces for multi-view reconstruction. In *ICCV*, 2023. 1, 2

[39] Christopher Wewer, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. SimNP: Learning self-similarity priors between neural points. In *ICCV*, 2023. 3, 7

[40] Christopher Wewer, Kevin Raj, Eddy Ilg, Bernt Schiele, and Jan Eric Lenssen. latentsplat: Autoencoding variational gaussians for fast generalizable 3d reconstruction. In *ECCV*, 2024. 3

[41] Guanjun Wu, Taoran Yi, Jiemin Fang, Lingxi Xie, Xiaopeng Zhang, Wei Wei, Wenyu Liu, Qi Tian, and Xinggang Wang. 4d gaussian splatting for real-time dynamic scene rendering. In *CVPR*, 2024. 3

[42] Haoyu Wu, Alexandros Graikos, and Dimitris Samaras. S-volsdf: Sparse multi-view stereo regularization of neural implicit surfaces. In *ICCV*, 2023. 1, 2, 5, 6, 7, 3

[43] Rundi Wu, Ben Mildenhall, Philipp Henzler, Keunhong Park, Ruiqi Gao, Daniel Watson, Pratul P. Srinivasan, Dor Verbin, Jonathan T. Barron, Ben Poole, and Aleksander Holynski. Reconfusion: 3d reconstruction with diffusion priors. In *CVPR*, 2024. 1

[44] Tong Wu, Jiaqi Wang, Xingang Pan, Xudong Xu, Christian Theobalt, Ziwei Liu, and Dahua Lin. Voxurf: Voxel-based efficient and accurate neural surface reconstruction. In *ICLR*, 2023. 2

[45] Luoyuan Xu, Tao Guan, Yuesong Wang, Wenkai Liu, Zhaojie Zeng, Junle Wang, and Wei Yang. C2f2neus: Cascade cost frustum fusion for high fidelity and generalizable neural surface reconstruction. In *ICCV*, 2023. 2

[46] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *CVPR*, 2022. 2, 3, 7

[47] Lior Yariv, Jiatao Gu, Yoni Kasten, and Yaron Lipman. Volume rendering of neural implicit surfaces. In *NeurIPS*, 2021. 2, 3, 5, 7

[48] Lior Yariv, Peter Hedman, Christian Reiser, Dor Verbin, Pratul P. Srinivasan, Richard Szeliski, Jonathan T. Barron, and Ben Mildenhall. Bakedsdf: Meshing neural sdfs for real-time view synthesis. In *ACM TOG*, 2023. 2

[49] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. pixelnerf: Neural radiance fields from one or few images. In *CVPR*, 2021. 6

[50] Zehao Yu, Songyou Peng, Michael Niemeyer, Torsten Sattler, and Andreas Geiger. Monosdf: Exploring monocular geometric cues for neural implicit surface reconstruction. In *NeurIPS*, 2022. 1, 2

[51] Jingyang Zhang, Yao Yao, Shiwei Li, Tian Fang, David McKinnon, Yanghai Tsin, and Long Quan. Critical regularizations for neural surface reconstruction in the wild. In *CVPR*, 2022. 2

[52] Jingyang Zhang, Shiwei Li, Zixin Luo, Tian Fang, and Yao Yao. Vis-mvsnet: Visibility-aware multi-view stereo network. In *ICCV*, 2023. 5, 2

[53] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun. Nerf++: Analyzing and improving neural radiance fields. In *arXiv*, 2020. 3, 4

[54] Fuqiang Zhao, Yuheng Jiang, Kaixin Yao, Jiakai Zhang, Liao Wang, Haizhao Dai, Yuhui Zhong, Yingliang Zhang, Minye Wu, Lan Xu, and Jingyi Yu. Human performance modeling and rendering via neural animated mesh. In *ACM TOG*, 2022. 2

[55] Junsheng Zhou, Baorui Ma, Shujuan Li, Yu-Shen Liu, Yi Fang, and Zhizhong Han. Cap-udf: Learning unsigned distance functions progressively from raw point clouds with consistency-aware field optimization. In *IEEE TPAMI*, 2024. 2, 5, 6, 3

Spurfies: Sparse-view Surface Reconstruction using Local Geometry Priors

Supplementary Material

A. Architectures

Our method employs four Multi-Layer Perceptrons (MLPs) and two sets of learnable latent codes. The MLPs are: A_{LP}, G_{LP}, used for local processing, signed distance regressor G_{REG}, and Radiance regressor A_{REG}. The latent codes are for color f^a and geometry f^g. Here are the details of these components:

Latent codes: The color latent codes $f^a \in \mathbb{R}^{64}$ and the geometry latent codes $f^g \in \mathbb{R}^{32}$ are both initialized from a normal distribution with variance $1\mathrm{e}{-4}$.

Radiance Local Processing A_{LP}: This MLP comprises four linear layers with intermediate dimensions of 128. It takes as input the color latent codes f^a and the relative distance of query points to the neural points, with positional encoding using 6 frequencies.

Raidance Regression A_{REG}: processes the aggregated color latent codes along with the view direction (without positional encoding). It consists of three linear layers with dimensions [259, 128, 3].

Geometry Local Processing G_{LP}: This MLP also has four linear layers with intermediate dimensions of 128. It processes the geometry latent codes f^g and the relative distance of query points to the neural points, without any positional encoding. This MLP is frozen after learning the local geometry prior and remains unchanged during sparse view surface reconstruction.

Signed Distance Regression G_{REG}: This consists of a single frozen linear layer that maps the processed geometry latent codes to an SDF value.

B. Loss Functions

Feature Consistency Loss $\mathcal{L}_{\mathrm{FC}}$ [16]: First, we estimate a set of surface points $\hat{\mathcal{P}} = \{\mathbf{p} | \mathbf{p} = \mathbf{x}_{i,u,v}(t^*)\}$ by finding zero-crossings t^* along each ray $\mathbf{x}_{i,u,v}(t)$ that are computed using linear interpolation between adjacent samples:

$$t^* = \frac{\hat{s}(\mathbf{x}(t_j))t_{j+1} - \hat{s}(\mathbf{x}(t_{j+1}))t_j}{\hat{s}(\mathbf{r}(t_j)) - \hat{s}(\mathbf{x}(t_{j+1}))} \,. \qquad (11)$$

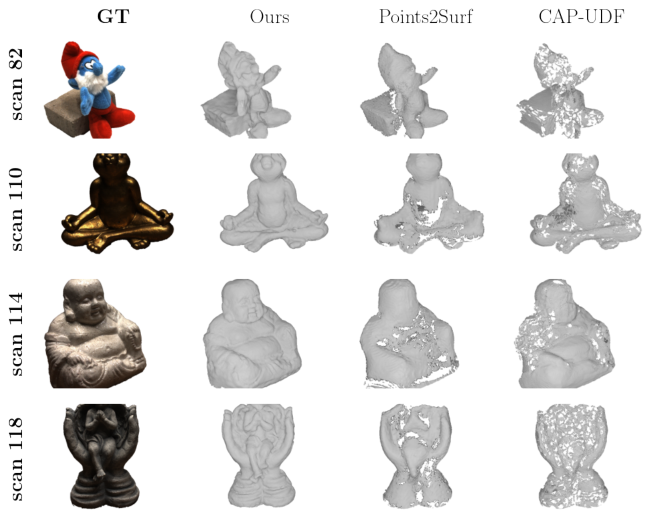

Figure 1. **Qualitative comparison** of mesh reconstruction with the point-based mesh reconstruction methods. In contrast to our approach, point-based mesh reconstruction methods often show missing areas, even when initialized with DUST3R [37] point clouds.

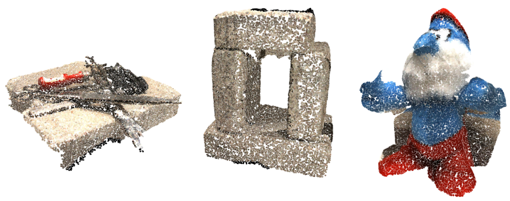

Figure 2. Sampled points from the reconstructed mesh on few scans from DTU dataset.

where t_j is estimated using $\hat{s}(\mathbf{x}(t_j)) \cdot \hat{s}(\mathbf{x}(t_{j+1})) < 0$. We then define the photo-consistency loss as:

$$\mathcal{L}_{\mathrm{FC}} = \frac{1}{|\hat{\mathcal{P}}||\mathcal{I}|} \sum_{p_i \in \hat{\mathcal{P}}} \sum_{\pi_j \in \Pi} \|f_\phi(\pi_j(p_i)) - f_\phi(\pi_0(p_i))\|_1, \qquad (12)$$

where Π is the set of the projection matrices for the images $\mathcal{I}$, with $\mathcal{I}_0$ being the reference view, and f_ϕ computed with VisMVSNet [52].

Pseudo loss $\mathcal{L}_{\mathrm{pseu}}$: We estimate surface points using rendering weights [11]. This approach ensures that the estimated points have a SDF value close to zero, effectively lying on the surface. We compute the estimated surface point location t^* along a ray $\mathbf{x}(t)$ as a weighted average of sample positions:

$$t^* = \sum_i \frac{w_i \cdot t_i}{\sum w_i}, \tag{13}$$

where w_i are the rendering weights and t_i are the sample depths along the ray. Using these estimated surface points, we introduce the pseudo ground-truth loss:

$$\mathcal{L}_{\text{Pseu}} = \frac{1}{N} \sum ||\hat{s}(\mathbf{x}(t^*))|| \,. \tag{14}$$

C. Implementation Details

Training: For sparse view reconstruction, we optimize a composite loss function:

$$L_{total} = \mathcal{L}_{\text{ren}} + \lambda_{\text{fc}} \cdot \mathcal{L}_{\text{fc}} + \lambda_{\text{pseu}} \cdot \mathcal{L}_{\text{pseu}} + \lambda_{\text{TV}} \cdot \mathcal{L}_{\text{TV}}, \tag{15}$$

where $\lambda_{\text{fc}} = 0.5$, $\lambda_{\text{pseu}} = 0.5$, and $\lambda_{\text{TV}} = 0.01$. We train the model for 100,000 iterations using the Adam optimizer [20] on a single A100 GPU.

We implemented efficient querying of K neural neighbors implemented using a GPU-accelerated VoxelGrid approach [39, 46]. The VoxelGrid parameters are configured as follows:

```
voxel_size = (0.025, 0.025, 0.025)
    ↪ % Voxel size for each dimension
voxel_scale = (2, 2, 2)
    ↪ % Voxel scale for each dimension
kernel_size = (3, 3, 3)
    ↪ % Range of voxels searched for
    ↪ neighbors
max_points_per_voxel = 26
    ↪ % Maximum number of points stored in
    ↪ a voxel
max_occ_voxels_per_example = 20000
    ↪ % Maximum number of occupied voxels
    ↪ per point cloud
ranges = (-1.0, -1.0, -1.0, 1.0, 1.0, 1.0)
    ↪ % Maximum ranges the VoxelGrid spans
```

The voxel size is set to match the average distance between neural points, ensuring an appropriate spatial distribution. Each voxel is limited to containing a maximum of 26 points, balancing between spatial resolution and computational efficiency.

We set K = 8 for neighbor queries, aiming to have, on average, 8 queried neural points for every ray-marched query point. This configuration strikes a balance between capturing sufficient local information and maintaining computational efficiency.

Datasets: For evaluation on the DTU [17] dataset, we adhere to the split established by S-VolSDF [42]. This protocol excludes scans from the training set of multi-view stereo methods, utilizing only those in the test/validation splits. Additionally, we follow the standard protocol employed by [1, 28, 47] for the quantitative evaluation.

Since, there are no previous sparse view method tested on Mip-NeRF 360 [3], we randomly select three input views for all qualitative evaluations. The lack of ground-truth point clouds precludes Chamfer Distance (CD) evaluation, limiting our analysis to qualitative results. Our evaluation encompasses four scenes from Mip-NeRF 360, with corresponding view IDs as follows: Garden: (DSC08116, DSC08121, DSC08140) Kitchen: (DSCF0683, DSCF0700, DSCF0716) Treehill: (_DSC9004, _DSC9005, _DSC9006) Stump: (_DSC9307, _DSC9313, _DSC9328).

Results: Our method demonstrates superior performance in mesh reconstruction compared to point-based techniques such as Points2Surf [10] and CAP-UDF [55]. As shown in Figure 1, even when using points obtained from Dust3R [37], these alternative methods often produce reconstructions with holes and fail to capture fine details. In contrast, our approach achieves more complete and detailed reconstructions, highlighting the effectiveness of our neural point-based representation and learned local geometry prior.

Additional reconstruction and Novel View Synthesis (NVS) results on the Mip-NeRF 360 dataset are presented in Figure 3 in comparison with NeuSurf [16] and S-VolSDF [42]. We also show points sampled from the reconstructed mesh Figure 2.

D. Local Prior

Data: To train our local prior, we design a setup that emulates volume rendering conditions. We sample two distinct sets of points:

1) **Query points** $\mathcal{X} = \{(\mathrm{x}_i, \rho_i)\}_{i=1}^{\mathrm{N}}$, sampled in close proximity to the mesh surface. These points are generated using two different variances (0.05 and 0.001) to simulate the ray-marching process in volume rendering. On average, we sample $\mathrm{N} = 500k$ query points for every mesh.

2) **Neural points** $\mathcal{N} = \{(\mathrm{p}_j, \mathrm{f}_j^g)\}_{j=1}^{\mathrm{M}}$, which represent the underlying structure of our reconstruction. To ensure density-agnostic learning during local prior training, we employ farthest point sampling on the mesh surface, maintaining an average inter-point distance of 0.025. The number of neural points, M, varies based on the mesh size, which is normalized to fit within a unit cube. During inference for sparse view reconstruction, we subsample the neural points to match the same density as training.

Our training data is from five classes in the ShapeNet [5] dataset: sofas, chairs, planes, tables, and lamps. To enhance robustness against noise, we add Gaussian noise with a variance of 0.01 to the neural points.

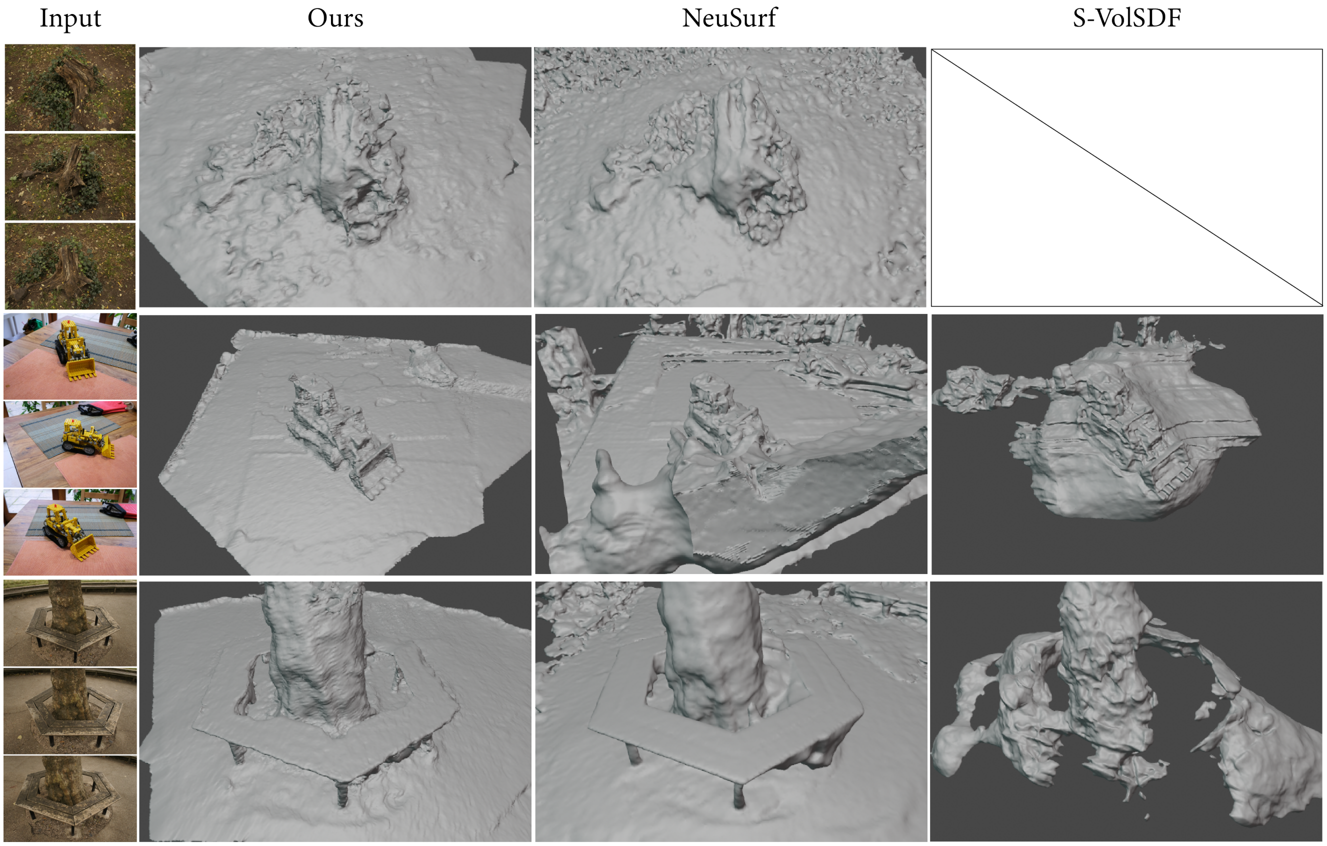

Figure 3. **Qualitative mesh reconstruction on Mip-NeRF 360 [3]**. Compared to previous sparse view methods, we can achieve much better reconstruction on larger, unbounded scenes. S-VolSDF completely failed on the stump scene.

Training: To train the local prior, we employ a combination of loss functions:

$$\mathcal{L}_{\text{prior}} = \mathcal{L}_{\text{SDF}} + \lambda_{\text{TV}} \cdot \mathcal{L}_{\text{TV}} + \lambda_{\text{eik}} \cdot \mathcal{L}_{\text{eik}} \quad (16)$$

where $\lambda_{\text{TV}} = 1\text{e}{-2}$ and $\lambda_{\text{eik}} = 1\text{e}{-3}$.

Our training process utilizes a batch size of 5. Each batch instance comprises 40,000 randomly sampled query points, equally distributed between positive and negative SDF samples, along with 2,000 neural points. These neural points are padded with points outside the unit cube to ensure consistent batch size.

We train the geometry MLP and the latent codes for 5,000 epochs. For the latent codes, we implement a cosine annealing learning rate schedule, starting at $1\text{e}{-2}$ and gradually decreasing to $3\text{e}{-4}$. The MLP is trained with a constant learning rate of $3\text{e}{-4}$. We use the Adam [20] optimizer throughout the training process. Instead of $K = 8$ during sparse view reconstruction, we set $K = 4$ neighbors during the local prior training.

The total training time for the local prior is approximately 8 hours, utilizing a single A100 GPU. This comprehensive training approach ensures that our local prior effectively captures the geometric properties of diverse shapes, enabling robust sparse-view reconstruction.

Results: We show some quantitative results in Figure 5 of surface reconstruction on ShapeNet [5] dataset and the Stanford bunny. These results are shown for unseen objects after training the prior. The geometry MLP is frozen and only the geometry latent codes are optimized. We achieve quality mesh reconstruction with high surface details.

We extend our analysis to demonstrate the potential of our optimized geometry latent codes for point cloud clustering. Figure 6 illustrates this capability using the Stanford bunny model, where we present six distinct clusters derived from these codes.

The clustering results reveal an property of our optimized geometry latent codes. These codes appear to capture and encode local surface orientations effectively. As a result, the clustering process groups points with similar local geometric characteristics together. This suggests that our method not only reconstructs the surface accurately but also learns a meaningful representation of local surface properties. Specifically, we observe that: 1) Points belonging to the same cluster tend to have similar surface normals or curvature properties. 2) Transitions between clusters are generally smooth, indicating a continuous representation of geometric features which is achieved using $\mathcal{L}_{\mathrm{TV}}$.

This clustering capability demonstrates an additional utility of our approach beyond surface reconstruction. It suggests potential applications in shape analysis, feature detection, and semantic segmentation of 3D models.

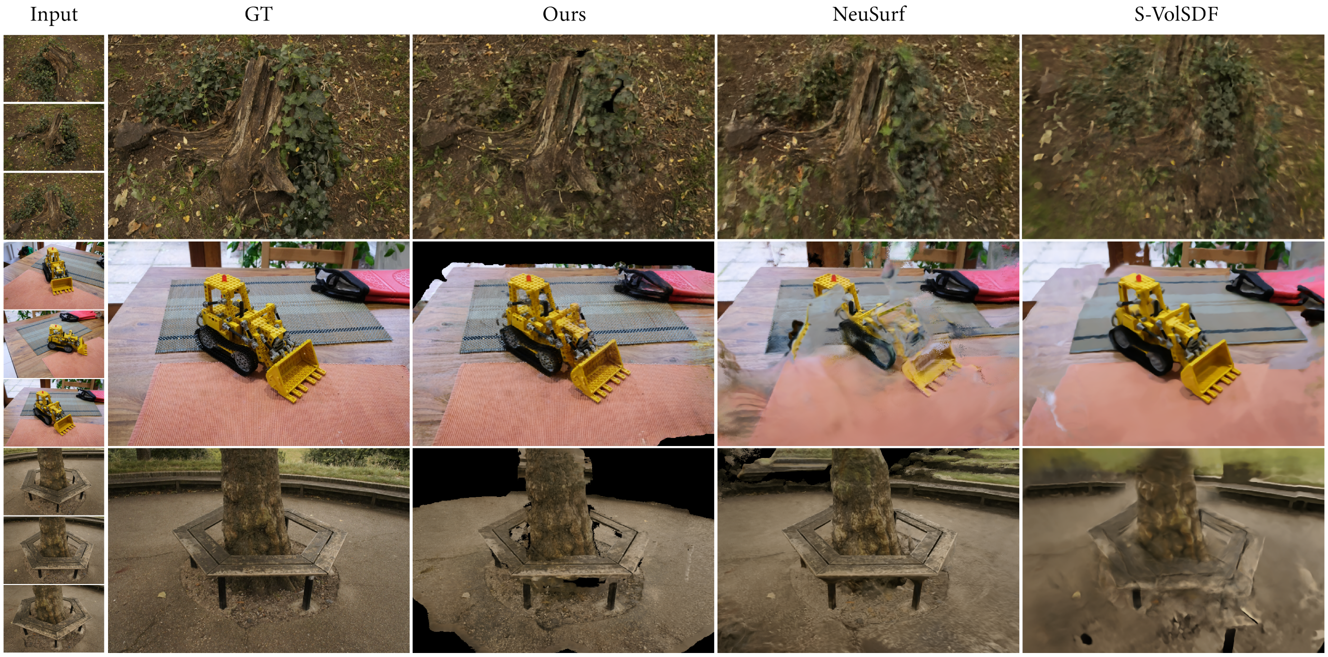

Figure 4. **Qualitative NVS on Mip-NeRF 360 [3].** Spurfies can synthesize novel views in higher quality than previous sparse-view methods.

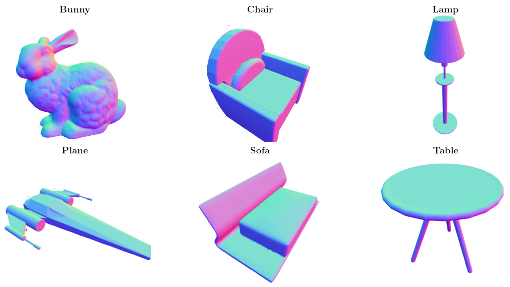

Figure 5. **Mesh reconstruction** results on a few unseen objects from ShapeNet [5] and the Stanford bunny.

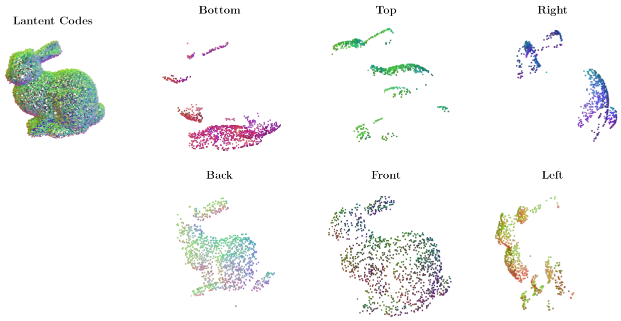

Figure 6. Clustering of optimized geometry latent codes based on six orientations. The geometry latent codes add local descriptive information to point clouds.