Latent Uncertainty-Aware Multi-View SDF Scan Completion

Faezeh Zakeri Lukas Ruppert Raphael Braun
Hendrik P.A. Lensch
Computer Graphics, University of Tübingen, Germany
{faezeh-sadat.zakeri, lukas.ruppert, raphael.braun, hendrik.lensch}@uni-tuebingen.de

Abstract

Imperfect reconstructions arising from occlusions, shadows, reflections, and other factors during 3D scanning often result in incomplete sections of the scanned object, with missing parts scattered randomly across its surface. We introduce an uncertainty-aware signed distance field (SDF) latent transformer that leverages uncertainty to identify and reconstruct missing parts based on the global shape of the incomplete scanned object and the immediate neighborhood of the affected regions. To our knowledge, we are the first to utilize uncertainties for SDF shape completion in the latent space. Our model has been trained on the entire Objaverse 1.0 dataset and demonstrates that our uncertainty-aware SDF completion method significantly outperforms previous works both numerically and visually. Code will be published at github.com/cgtuebingen/ua3dscancomp.

1. Introduction

3D scan completion involves predicting missing sections of objects based on their visible parts. In real-world scenarios, 3D scanners like depth cameras or LiDAR often cannot capture an object in its entirety due to factors such as occlusions, sensor limitations, or surface properties like reflectivity. Our method directly works on the resulting partial scans, combines all available views and fills in missing geometry. For many applications, including autonomous driving, robotics, and virtual reality, where high-quality 3D shapes need to be derived directly from scanning devices, 3D scan completion can play a crucial role [63].

In this work, we present a novel method for shape completion on real-world 3D scans. As the input to our method, we use a set of one or more aligned partial meshes one might retrieve from 3D scanners with depth cameras or through visual odometry. Since 3D scans are inherently noisy and rarely observe the entire object, occluded areas need to be filled in even after combining all observations.

In our method, we estimate the uncertainty inherent in all individual observations and leverage it for effective shape completion in the form of (latent) signed distance fields (SDFs). Before shape completion, we combine the individual inputs and estimated uncertainties into a joint representation, which we then complete using an uncertainty-aware latent-space SDF transformer. This transformer operates on a sequence of 3D patches and may produce seams between them. We present a seam removal model, which utilizes the SDF's Eikonal property to remove any such seams.

Our main contributions are: 1. An uncertainty model for SDFs generated from partial meshes which is used not only for combining partial inputs but is also an essential part in the following shape completion, where it serves as a clear indicator for missing information. 2. A GPU-based on-the-fly partial-view generation pipeline, simulating the 3D scanning process, enabling training on randomly generated inputs from huge datasets such as Objaverse 1.0 [15] (≈8TB) without major preprocessing or extra storage. 3. Fast uncertainty-based shape completion using a single forward pass through a patch-based latent SDF transformer, and a seam removal model that fixes inconsistencies between predicted SDF patches using the Eikonal property.

To the best of our knowledge, we are the first latent-space shape completion method leveraging uncertainty and also the first to train on partial views from the entire Objaverse 1.0 [15] dataset, thanks to our efficient pipeline. Our method compares favorably to other shape completion methods on ShapeNet [4]. To demonstrate the real-world performance of our model, we scanned several 3D objects with a handheld 3D scanner and show shape completion results from 3 partial views in Figure 1.

2. Related Work

Learning-based approaches have achieved impressive results in reconstructing 3D shapes from partial inputs using output representation types such as voxels, meshes, point clouds, and implicit functions. Point clouds are a more common representation for shape completion [51] but need heavy post-processing due to sparsity and noise [20, 22, 23]. Meshes are more efficient but pose topological challenges like self-intersections and non-manifold structures [19].

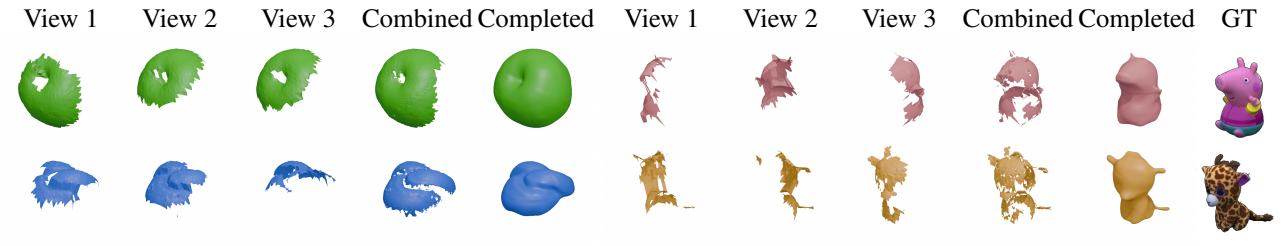

Figure 1. Partial meshes from individual camera views of our own real-world 3D scans combined and completed by our method.

Voxel grids extend 2D pixels to 3D grids, offering simplicity and regularity. Each grid cell holds an occupancy or signed distance value. SDFs represent shapes as continuous functions, with the zero-level set marking the surface and the sign indicating in- and outside areas. We review recent shape-completion approaches, roughly categorized by type.

2.1. Point-based Completion

Most of the object-level point cloud completion leverages autoencoders as backbones, employing either PointNet-based modules to capture global features, as seen in methods like PointNet [36], PointNet++ [37], PCN [55], FoldingNet [52], TopNet [43], PF-Net [21], GRNet [50], and MSN [26]. Others are using a point transformer [45] to focus on local structures as seen in more pioneer works of PointTransformer [61] and PoinTr [53] methods. To gain from both local and global features, DPPCC [60] aggregates various features to separately represent both the known and missing parts. SnowflakeNet [49] addresses the decoding task using a SkipTransformer-based architecture that iteratively splits and refines low-resolution points. SeedFormer [62] addresses sparsity by completing a sparse set of patch seeds derived from an incomplete input and upscales these seeds into a full point cloud. The seeds consist of sparse 3D positions that are enhanced by a transformer, which incorporates semantic information. LAKe-Net [41] identifies aligned key points to create a surface skeleton mesh, which assists in generating a complete point cloud. AdaPoinTr [54] tackles discontinuity issues and demonstrates increased robustness against noisy input. AnchorFormer [6] extracts global features and identifies key "anchor" points, which are combined with a subset of input points and subsequently upsampled into a dense mesh. VRPCN [33] represents the shape from partial input as probability distributions, which are sampled and refined through a hierarchical encoder-decoder network. LION [57] introduces a flexible latent diffusion model for point clouds, using a VAE and Point-Voxel CNNs to map point cloud features into latent space.

2.2. Occupancy-based Completion

Occupancy Networks [29] is a classic encoder-decoder approach that encodes a partial input point cloud into a global feature vector z, which is then used by a neural implicit decoder to predict the complete geometry. In their approach, 3D shapes are represented via their occupancy. These methods allow querying at any arbitrary 3D location to determine whether a given point lies inside or outside the represented object. Using the MISE algorithm, meshes can be efficiently reconstructed from this representation [29]. Convolutional Occupancy Networks [35] expands the Occupancy Networks to scene-scale sizes by leveraging local feature refinement on a latent 3D grid with learnable 3D convolutions. IF-Nets [8] use multi-scale feature grids to encode 3D shapes globally and locally and predict the occupancy at the feature space. Few-Shot [30] propose learning class-specific global shape priors directly from data to capture multi-scale information about the 3D shape and account for intra-class variability by virtue of an implicit compositional structure. ShapeFormer [51] efficiently encodes shapes into a sparse grid of vector-quantized deep implicit functions (VQDIF). It completes partial inputs using an autoregressive transformer that processes a sequence of location and patch index pairs. 3DILG [58] employs an irregular grid of latent codes for shape representation and utilizes an autoregressive transformer to complete partial inputs. 3DShape2VecSet [59] generates multiple global vectors for shape representation, generated by cross-attention between input points and sampled query points. The (variationally) auto-encoded shape is subsequently completed by latent space diffusion. The output point cloud is retrieved from the completed global vectors by cross-attention against a dense query point cloud. Wu and Miao [46] map both partial and complete shapes to the same sequence and train exclusively on complete point clouds. Also, instead of utilizing a large number of grids to represent shapes and autoregressively decoding both coordinates and latents, their approach gradually generates shapes from 24 structural patch features, applying autoregression only to feature tokens.

2.3. SDF-based Completion

3D-EPN [13] proposed a 3D CNN encoder to complete low resolution partial 3D scans using semantic information. DeepSDF [34] introduced a technique for learning continuous signed distance functions (SDFs) to represent 3D shapes, jointly optimizing compressed latent SDF rep-

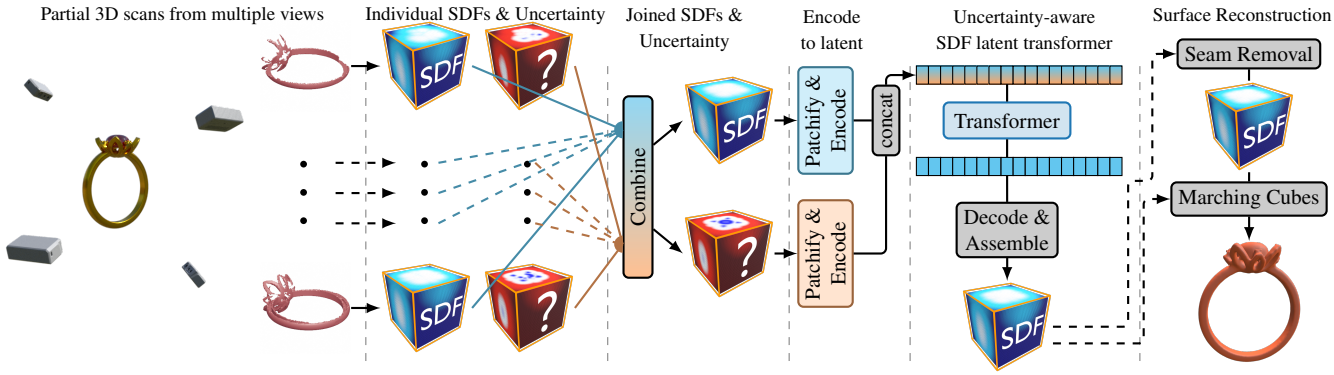

Figure 2. We present a method for multi-view scan completion. Given one or more views, we compute singed distance fields (SDFs) and uncertainty estimates (Section 3.1), which we combine into a single input (Section 3.2) for shape completion by our uncertainty-aware SDF latent transformer (Section 3.3). The completed shape is predicted as a sequence of 3D patches. We optionally remove potential seams between the patches (Section 3.4) and extract the completed object from the SDF using Marching Cubes [27].

resentations along with an coordinate-MLP-style decoder to reconstruct the full SDF from these latent codes. Relying on a single latent vector for shape representation limits their resolution. AutoSDF [31] allows for higher resolution, using patches. They use an autoregressive transformer to generate shapes from a subset of patches or inputs extracted from other modalities. Their level of detail is limited by vector quantization. SDFusion [7] and DiffComplete [10] utilize latent diffusion models for shape generation and completion. DiffComplete [10] supports the combination of multiple partial inputs. PatchComplete [38], and Wu et al. [47] operate on 32^3 truncated SDFs. Both methods use attention against learned multi-resolution shape priors. POC-SLT [56] employs a patch-wise variational autoencoder (P-VAE), trained to represent arbitrary SDF patches. In contrast to other methods, they operate on 128^3 SDFs and encode them into patch-wise latent codes without quantization. Their shape completion is formulated as a masked autoencoding (MAE) problem. Given SDF patches need to contain complete information, partial occlusions cannot be handled. We utilize their P-VAE to autoencode SDF patches, but use uncertainty to allow for shape completion from arbitrarily occluded inputs.

2.4. Uncertainty

Physical measurements always come with some level of noise and uncertainty. Especially in 3D space, objects are rarely observed from all angles, leaving entirely unknown occluded regions. Explicitly modeling the uncertainty in these regions allows predictions to rely primarily on known-good measurements. Eldesokey et al. [16] enhance 3D depth completion by utilizing uncertainty in normalized CNNs. Similarly, Cao et al. [3] employ a deep ensemble method to address uncertainty in LiDAR-based object completion. Truncated SDF fusion methods [11, 32], model uncertainty from the camera's perspective: Surfaces which are observed at an angle or at large distances contribute less to the merged result and any SDF measurements with distances beyond a truncation threshold are discarded entirely. We do not assume the presence of camera information and instead derive our uncertainty measure directly from the partial mesh. We combine partial views using this uncertainty and also estimate the combined uncertainty to help with the following shape completion.

3. Method

Our pipeline for 3D shape completion from partial 3D scans consists of four main steps (see Figure 2): First, we estimate the uncertainty of partial 3D meshes generated by a 3D scanner (Section 3.1). Second, we combine the partial observations into a joint SDF and uncertainty representation (Section 3.2). Third, this joint representation is divided into regular patches and encoded into a sequence of SDF and uncertainty latent codes, which is then completed by a transformer into a latent sequence of completed SDF patches (Section 3.3). Those are decoded and assembled back into a 128^3 SDF grid. Finally, potential seams between patches are removed by a convolutional model (Section 3.4).

3.1. Modeling Uncertainty

During 3D scanning, one or more cameras capture multiple views of the to-be-scanned object. For each view, a depth map describes the distance of each pixel from the camera. By projecting this depth map back into the scene, one can extract a partial mesh describing the visible parts of the scanned object. Our method expects that these partial meshes are already registered with each other, e.g., through bundle adjustment [44] or ICP [5]. From each of these partial meshes, we compute a 128^3 SDF s, and uncertainty field u, e.g. Figure 3.

Given only partial information, the partial SDFs neces-

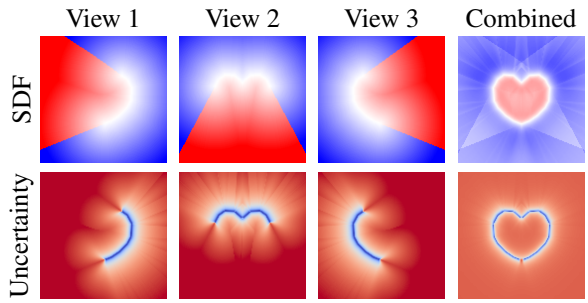

Figure 3. 2D example of partial and combined SDFs and uncertainty from three camera views (1: right, 2: top, 3: left). In the SDF, blue areas are outside, white areas are close to the surface, and red areas are inside. In the uncertainty, blue areas are close to 0 (certain) and red values represent high uncertainty. Here, the uncertainty-based combination already recovers the complete heart shape and predicts low uncertainty for its surface.

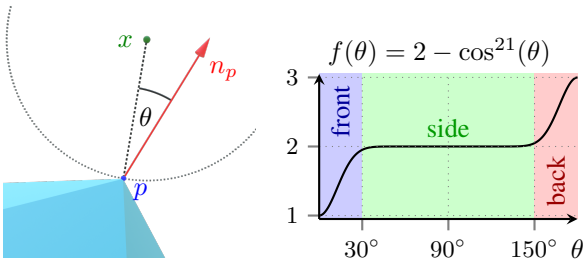

Figure 4. Observability term for a point x on the SDF grid. The angle θ is defined between the surface normal n_p at the closest point p on the partial mesh and the direction from p to x.

sarily contain errors in unobserved areas. Any single observation from a camera only ever sees one side of an object and thus only produces surface patches and not a closed, watertight mesh. Anything behind those surface patches will be classified as inside indefinitely, despite the limited thickness of the surface.

Real-world 3D scans often consist of hundreds of individual views. In order to efficiently judge the uncertainty of all views, we designed a quick-to-evaluate closed-form estimate, derived from the following principles.

Squared Distance. The only SDF values we can truly be certain of are those close to the surface. The further we move from the surface, the more likely it becomes that there is another unobserved closest surface which should instead define the SDF value. Therefore, our main term for modeling uncertainty is the squared distance between the SDF grid point x and the closest surface point p: $d^2(x, p)$.

Observability. In absence of a camera pose, we assume that areas in **front** of a surface are likely to represent empty space between the camera and the object. On the other hand, on the unobserved **sides** and especially **behind** surfaces, we cannot rely as much on the SDF information extracted from the mesh. Here, we assign higher uncertainty. We define the observability factor as $2 - \cos^{21}\theta$ (Figure 4, right), where θ is the angle between the smooth surface surface normal n_p at the closest surface point p and the direction towards the SDF grid position x (Figure 4, left). I.e.,

$$\cos\theta = \frac{n_p \cdot (x - p)}{||n_p|| \cdot ||x - p||}. \tag{1}$$

The exponent is chosen such that the **front** and **back** are in the $< 30°$ and $> 150°$ range, with the **sides** in between.

Tangentiality. Without a closed surface, the SDF's boundary between inside and outside infinitely extends tangentially from the outer edges of a partial mesh. Here, the sign flips abruptly from positive to negative. To prevent this artifact from affecting the combined SDF, we assign high uncertainty to this tangential region, where $\cos\theta \approx 0$.

Uncertainty Estimate. Our uncertainty estimate combines these three parts as follows:

$$u_x = \underbrace{d^2(x,p)}_{\text{sqr. distance}} \cdot \underbrace{(2 - \cos^{21}\theta)}_{\text{observability}} + \underbrace{0.5 \cdot (1 - |\cos\theta|)^2}_{\text{tangentiality}} \tag{2}$$

The uncertainty estimate is computed purely from the partial mesh and does not rely on any information about the camera location or view frustum, which may not be available. Low uncertainty can only be achieved close to a surface and in front of it where the camera did not observe any closer (occluding) surface. Behind surfaces and to the sides, uncertainty becomes high, as these areas are likely to contain other closest surfaces defining the correct SDF value which could not be observed in the current view.

3.2. Combining Partial Views

We combine the partial SDF values $\{s_{x,1}, \ldots, s_{x,N}\}$ and associated uncertainty values $\{u_{x,1}, \ldots, u_{x,N}\}$ from N (potentially hundreds) input views in an inverse-variance fashion, where our uncertainty estimates are treated as a substitute for variance. The combined SDF values s_x and uncertainty values u_x are therefore computed as:

$$s_x = \frac{1}{W_x}\sum_{\mathrm{i}=1}^{N} s_{x,\mathrm{i}} w_{x,\mathrm{i}}, \quad u_x = \frac{1}{W_x}\sum_{\mathrm{i}=1}^{N} \left(s_{x,\mathrm{i}}^2 w_{x,\mathrm{i}}\right) - s_x^2, \\ \text{where } w_{x,\mathrm{i}} = \frac{1}{u_{x,\mathrm{i}} + \varepsilon}, \quad \text{and} \quad W_x = \sum_{\mathrm{i}=1}^{N} w_{x,\mathrm{i}} \tag{3}$$

For numerical stability, we compute the weighted mean and variance incrementally [18]. This also allows us to combine

arbitrarily many partial views into one combined input for further shape completion by the transformer (Section 3.3).

The SDF domain is well-suited for combining information from multiple views: Combining them naturally interpolates between noisy inputs or slightly offset input views. However, since the SDFs are computed from partial observations, large parts of them contain incorrect information and may wrongly classify occluded areas as inside. Besides our inverse-uncertainty weighted combination, we introduce a Bayesian approach to deal with completely unobserved areas: We initialize the incremental mean and variance (SDF and uncertainty) with a small prior which predicts the entire space as being slightly outside ($s_0 = -0.03$) with a high uncertainty ($u_0 = 1$) and a low weight ($w_0 = N \cdot 0.1$). Any certain enough observation can now mark areas as inside, but areas lacking any clear observation will default to being classified as outside.

While the combined SDF values still cleanly separate inside and outside by the zero-level set, they otherwise no longer represent a true SDF where each value accurately describes the distance to the closest surface. Therefore, we generate a mesh from the combined SDF values via Marching Cubes [27] and then compute a new, consistent SDF from that for further processing.

3.3. Shape Completion

Given a 128^3 SDF and corresponding uncertainty estimates from the combined partial views (Section 3.2), we transform the input into a sequence of latent codes, and perform latent-space shape completion using a transformer encoder.

3.3.1. Encoding and Decoding SDFs and Uncertainty

To encode the SDF, we divide the SDF into 64 patches of size 32^3 and apply the patchwise variational autoencoder (P-VAE) by Zakeri et al. [56], to independently encode them into latent codes of dimension 4096. The result of shape completion will again be a sequence of latent SDF patches, which we also decode using the P-VAE [56] and then assemble the patches back into a full 128^3 SDF result.

The P-VAE [56] is a convolutional model, allowing for fast encoding an decoding. We use the pre-trained model by Zakeri et al. [56], which was trained on millions of SDF patches extracted randomly from ShapeNet [4] meshes and was shown to generalize well to different datasets.

We globally encode the uncertainty with a CNN and patchify the output into 64 latent codes of dimension 4096, matching the SDF patches. The CNN consists of 2 sequential convolutional blocks, which each consist of a 3D convolution layer (first 5×5×5, stride 4, then 3×3×3, stride 2), a 3D batch norm, and a ReLU activation. Unlike the pre-trained P-VAE, which we only use for inference, the uncertainty CNN is trained alongside our shape completion model.

3.3.2. Uncertainty-Aware SDF Latent Transformer

We use a standard PyTorch [1] transformer encoder implementation with 18 layers, an internal latent dimension of 2048 and 16 heads to perform latent space shape completion. Its inputs are compressed latent tokens of dimensions 2048 representing SDF patches, uncertainties, and 3D positional encoding [40].

For processing, the encoded SDF and uncertainty patches are first concatenated into a sequence of tokens with dimensions $[B, 64, 2 \cdot 4096]$. We further concatenate a 3D positional encoding [40] of the same shape to the latent vectors and then map the tokens down to a latent dimension of size 2048 with a linear layer, to match the transformer's internal dimensionality. After shape completion by the transformer, the resulting sequence of latent codes is mapped up to a sequence of latent SDF patches with latent dimension 4096 by another linear layer.

3.3.3. On-the-Fly Training Data Generation

In order to efficiently train on datasets as large as Objaverse 1.0 [15] at non-trivial resolutions, we generate most of our input data on-the-fly, on the GPU. For each object, we only load vertices and faces and select a random subset of preset camera views. For each camera view, we rasterize a depth image using nvdiffrast [25] to simulate a 3D scanner and generate a partial mesh from the result. From these partial meshes, still on the GPU, we compute SDF and uncertainty grids using custom CUDA code. We then incrementally combine the per-view SDF and uncertainty on the CPU (to preserve GPU memory) and send the combined result from the data loader over to the main process for training. This allows us to augment the dataset to practically infinitely many different combinations of partial views while only having to load and store the compact mesh data per sample. We can also freely adapt the task difficulty by selecting fewer or more partial views per input.

Despite this memory-efficient approach, our training data for Objaverse 1.0 [15] still consists of almost 8 TiB. Per object, that is approximately 10.4 MiB of mesh data, 1MiB for the ground truth latent codes, and some small meta data for identifying the object.

3.3.4. Data Augmentation - Camera Selection

We make use of our ability to generate partial inputs on the fly to train our model on a wide range of different inputs: For every sample, we first choose uniformly from three different sampling strategies: Random, continuous, or stratified. We pre-define 20 possible camera views using a spherical Fibonacci [28] point set for uniform coverage. During the first 10 epochs of training, each method samples 1-20 of these views for training. In the last 4 epochs, for fine-tuning on sparser inputs, we restrict this to 1-4 views.

With random sampling, views are chosen independently at random, simulating the kind of input one might obtain by

taking individual (depth) images with a handheld camera.

In the continuous setting, a contiguous range of views is selected with only the starting point chosen at random. This simulates moving a 3D scanner around an object and typically leaves large parts of the object unobserved, since the given views are clustered closely together.

With stratified sampling, we achieve the opposite effect: The object is covered evenly, with only slight variations of the camera positions chosen per stratum. This would be the result of a 3D scan with a dense camera array. Here, only smaller gaps left by occlusions need to be filled in.

3.3.5. Losses and Training Schedule

The completed result is compared in latent space against the ground truth SDF patches using an L^1 loss. The ground truth is generated by merging 100 views from a spherical Fibonacci [28] point set.

We train the uncertainty encoder CNN alongside the SDF Latent Transformer from scratch, details can be found in Appendix 9. Unlike Zakeri et al. [56], we train the transformer in a pure sequence-to-sequence setting, not as a masked autoencoder (MAE).

For visualization and metric evaluation, the predicted latent codes are decoded into a 128^3 SDF grid and a mesh is extracted using Marching Cubes [27].

3.4. Seam Removal

Since shape completion operates on patches, the predicted output patches do not necessarily align perfectly. In order to always produce a seamless 128^3 SDF result, we train a small CNN to fix potential seams. The CNN consists of only two convolutional layers, separated by ReLU. The first layer performs a $3\times3\times3$ convolution, internally mapping the input to 8 channels. The second layer performs a $5\times5\times5$ convolution, directly predicting the filtered 128^3 SDF. The seam-removal CNN is intentionally kept small, such that it performs its task of filtering the shape completion output without the ability to overfit on any particular features.

To stay consistent with the given input SDF s_{in}, we start with an L^1 loss against the output SDF s. To produce seamless SDFs, we add a loss based on the Eikonal property: The gradient ∇s of an SDF s always has magnitude 1, since it measures the distance to the closest surface. We estimate these SDF gradients using central differences and use the following weighted combination of losses for training:

$$\mathcal{L}_{\text{Consistency}} = \frac{1}{N}\sum_{i=1}^{N} |s - s_{\text{in}}|\,, \tag{4}$$

$$\mathcal{L}_{\text{Eikonal}} = \frac{1}{N}\sum_{i=1}^{N} \left(\|\nabla s\| - 1\right)^2, \tag{5}$$

$$\mathcal{L}_{\text{SR}} = \mathcal{L}_{\text{Consistency}} + 0.05 \cdot \mathcal{L}_{\text{Eikonal}}. \tag{6}$$

4. Evaluation

We evaluate our shape completion pipeline on the Objaverse 1.0 [15] and ShapeNet [4] datasets. We also show qualitative shape completion results on several real-world scans we captured using a handheld 3D scanner. We further demonstrate the effectiveness of the proposed uncertainty measure and seam removal in two ablation studies.

We have trained our main model on Objaverse 1.0 [15], processed and filtered in several steps. For the full training details, see Appendix 7. Reported Chamfer Distances (CD_{L1} and CD_{L2}) are multiplied by 100. We define all metrics used for evaluation in Appendix 13.

4.1. Synthetic Shape Completion

We show synthetic shape completion results given 1-4 input views on Objaverse 1.0 [15] in Figure 5 and provide quantitative results in Table 1. Visually, we see plausible shape completions starting from the first view, even given particularly sparse inputs. Quantitatively, the best results are generally achieved using 3 views without seam removal. Since there is currently no other work performing partial scan completion on all of Objaverse 1.0 [15], and it would be infeasible to re-train them on this large-scale dataset, there are no related works compared in this table.

In terms of runtime, it takes 35.6ms to run the inference on the combined SDF and uncertainty on a single RTX 4090, including encoding and decoding. The computation and combination of SDFs and uncertainties runs in parallel during data loading and can be offloaded to a different GPU.

To relate the performance of our method to previous work, we evaluate our method on ShapeNet [4] in Table 2. We follow the evaluation procedure of PatchComplete [38] and use the data from their paper for comparison. Visual shape completion results, including comparisons to results by PatchComplete [38], can be found in Figure 6. For this task, we fine-tuned our method on the training split of ShapeNet [4] defined by PatchComplete [38] (Appendix 9).

Our numerical results significantly outperform previous state-of-the-art methods and our visual results show plausible shape completions, even from sparse inputs such as just the short side of a bag.

4.2. Imperfect Poses

Since we do not assume any knowledge of the camera pose in our pipeline, we have analyzed how our model responds to randomly rotated input views which one may encounter when aligning multiple input views against each other.

The results are shown in Section 10 in the supplemental. In general, the larger the rotation, the worse the results become. However, given at least 3 views, larger rotations lead to high uncertainty in overlapping areas, allowing our model to somewhat compensate for the alignment error.

Table 1. Shape completion on Objaverse 1.0 [15]. In the single-view setting, our method produces state-of-the-art results. Adding more views further improves our model's performance. (Since DiffComplete [10] does not provide a pre-trained checkpoint, we could not evaluate their method.) As a baseline, the uncertainty-combined SDF (Sec. 3.2, input to our shape completion) is also evaluated.

Model	Task	IoU↑	F_1↑	CD_{L2} ↓	HD↓	CD_{L1} ↓	NC↑	IN↓	CMP↑
PatchComplete [38] (zero-shot)	1 view	0.413	0.239	5.968	0.598	18.253	-	0.892	0.373
Combined SDF	1 view	0.138	0.230	27.769	1.130	44.256	0.509	0.842	0.180
Ours	1 view	**0.524**	**0.506**	**4.415**	**0.452**	**11.573**	0.766	**0.812**	**0.756**
Ours + seam removal	1 view	0.515	0.471	5.493	0.516	13.331	**0.774**	**0.812**	0.755
Ours (no uncertainty)	1 view	0.413	0.364	6.574	0.556	16.678	0.710	0.881	0.623
Combined SDF	2 views	0.279	0.360	16.408	0.989	29.234	0.564	0.811	0.310
Ours	2 views	0.601	0.589	4.107	0.411	9.919	0.796	0.771	0.829
Ours + seam removal	2 views	0.591	0.550	5.426	0.486	11.910	0.802	0.773	0.828
Ours (no uncertainty)	2 views	0.495	0.451	5.091	0.484	13.297	0.748	0.839	0.685
Combined SDF	3 views	0.389	0.457	9.416	0.831	20.181	0.616	0.770	0.418
Ours	3 views	0.636	0.625	**3.882**	0.399	9.350	0.806	0.749	0.857
Ours + seam removal	3 views	0.625	0.587	5.217	0.474	11.283	0.812	0.753	0.856
Ours (no uncertainty)	3 views	0.537	0.497	4.970	0.469	12.404	0.765	0.815	0.742
Combined SDF	4 views	0.449	0.514	7.194	0.744	17.160	0.643	0.736	0.480
Ours	4 views	**0.653**	**0.644**	3.915	**0.395**	**9.108**	0.812	**0.738**	**0.872**
Ours + seam removal	4 views	0.642	0.606	5.314	0.470	11.099	**0.816**	0.743	0.870
Ours (no uncertainty)	4 views	0.564	0.528	4.660	0.449	11.586	0.772	0.802	0.769

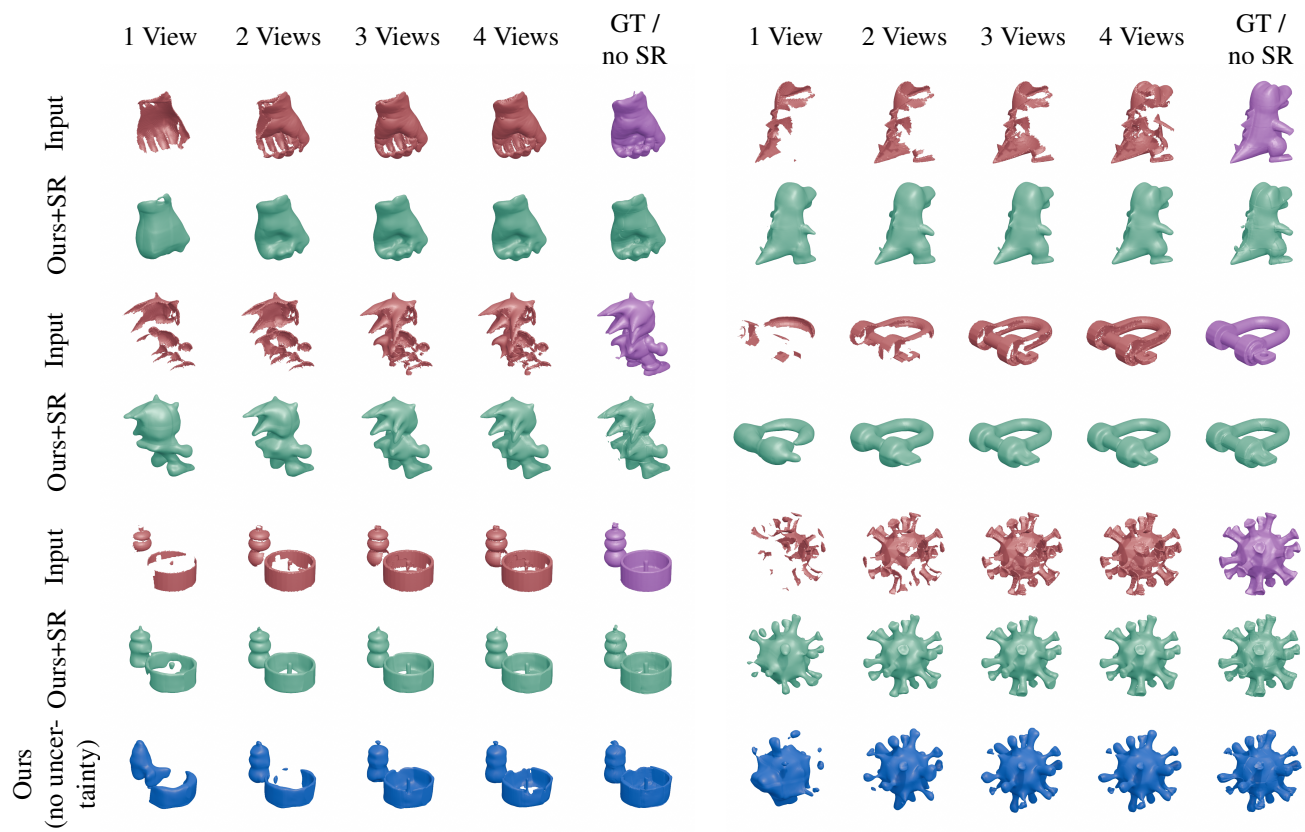

Figure 5. Partial scan completion on Objaverse 1.0 [15]. Partial inputs generated from 1-4 views. We show our results generally with seam removal "Ours+SR". We also show the result before seam removal in the "no SR" column. As an ablation, without uncertainty information, the model struggles to complete inputs, especially in the single-view case. Best viewed zoomed-in.

Table 2. Single-view shape completion on ShapeNet [4]: Average across unseen categories (bag, lamp, bathtub, bed, basket, printer, laptop, bench). Data of other methods are taken from DiffComplete [10]. Per-class results shown in Appendix 8.

Method	IoU↑	CD_{L1} ↓
3D-EPN [13]	0.594	5.58
Few-Shot [30]	0.403	9.58
IF-Nets [8]	0.581	5.29
AutoSDF [31]	0.452	5.86
ConvONet [35]	0.601	5.26
PatchComplete [38]	0.654	4.27
DiffComplete [10]	0.675	4.10
Ours	**0.684**	**3.04**
Ours + seam removal	0.649	4.34
Ours (zero-shot)	0.637	3.23

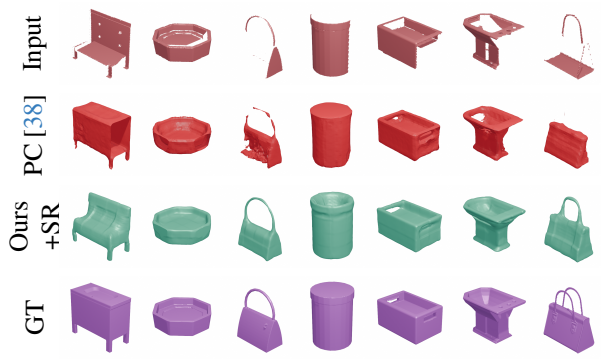

Figure 6. Single-view shape completion on ShapeNet [4] objects. Our method produces fewer objectionable artifacts than PatchComplete [38].

4.3. Real-World Shape Completion

To evaluate our method on real-world 3D scans, we scanned several objects with a commercial handheld 3D scanner. The individual views, combined partial inputs and completed results are shown in Figure 1. The corresponding meshes are included in the supplemental.

4.4. Ablation: Uncertainty and Seam Removal

To evaluate the effectiveness of our model with and without seam removal and uncertainty, we evaluate three different configurations for Objaverse 1.0 [15] in Table 1.

The base model without seam removal "Ours" produces the best numeric results. The addition of seam removal "Ours + seam removal" results in visually more pleasing results, but negatively affects performance on the metrics. Besides smoothing over the seams, the filtering process also tends to slightly smooth out wanted details and tends to produce minimally slimmer objects. This not only removes visually objectionable "floaters" but also some thin surfaces in the completion result, which can drastically increase the closest distance from nearby points in the ground truth.

In the "Ours (no uncertainty)" configuration, we completely remove the uncertainty information from the input to the shape completion transformer. Without access to the uncertainty information, the performance drops significantly. This validates that our combined uncertainty estimates are essential for generating high-quality shape completions.

5. Limitations and Future Work

In future work, we would like to address shape completion on entire 3D scenes. In existing datasets [9, 12], the individual camera views are typically already combined into a single mesh. This prevents our method from assessing the uncertainty, which we derive from the surfaces observed by individual camera views. Another complexity in scene completion is having to remove geometry from neighboring objects. Our model is trained to complete all visible surfaces into plausible shapes, it does not attempt to remove any of the given geometry. Neither does it perform class-based shape generation, merely conditioned on the input. Failure cases are shown in Supplemental 11.

To cleanly separate out the geometry of individual objects, one might already apply segmentation models [14, 24] on the partial views, to exclude neighboring geometry early-on. One might also try to extend the model with a more concrete understanding of object classes and parts, e.g. by training it with secondary heads performing classification or even segmentation as side tasks. This may improve its ability to extend missing parts which are not observed by any of the partial views. Previous work [39] has shown that shape completion trained on canonical views can produce better reconstructions of known object classes, but struggles to complete novel objects or objects in different orientations. Currently, due to the random orientations of objects in the Objaverse 1.0 [15] dataset, our model tends more towards generality than to specialize on shape completion of known classes. Enhancing our model with semantic information might combine the best of both worlds. We expand on this idea for future work in Supplemental 12.

6. Conclusion

We presented a novel method for partial scan completion from arbitrarily many input views which heavily benefits from uncertainty as a core part of the shape completion process. Especially in ambiguous, few-view scenarios, our uncertainty model guides the shape completion towards high-quality results. Our model generalizes well to a wide variety of complex objects, and is designed with efficiency in mind. As a result, our method is the first shape completion method that can handle the entire Objaverse 1.0 [15] dataset.

Acknowledgment

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG) – EXC number 2064/1 – Project number 390727645 and SFB 1233, TP 2, Project number 276693517, and by the IMPRS-IS.

References

[1] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, Bin Bao, Peter Bell, David Berard, Evgeni Burovski, Geeta Chauhan, Anjali Chourdia, Will Constable, Alban Desmaison, Zachary DeVito, Elias Ellison, Will Feng, Jiong Gong, Michael Gschwind, Brian Hirsh, Sherlock Huang, Kshiteej Kalambarkar, Laurent Kirsch, Michael Lazos, Mario Lezcano, Yanbo Liang, Jason Liang, Yinghai Lu, CK Luk, Bert Maher, Yunjie Pan, Christian Puhrsch, Matthias Reso, Mark Saroufim, Marcos Yukio Siraichi, Helen Suk, Michael Suo, Phil Tillet, Eikan Wang, Xiaodong Wang, William Wen, Shunting Zhang, Xu Zhao, Keren Zhou, Richard Zou, Ajit Mathews, Gregory Chanan, Peng Wu, and Soumith Chintala. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, 2024. 5

[2] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Niessner. Scan2cad: Learning cad model alignment in rgb-d scans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3, 4

[3] Anh-Quan Cao, Angela Dai, and Raoul de Charette. Pasco: Urban 3d panoptic scene completion with uncertainty awareness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14554–14564, 2024. 3

[4] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015. 1, 5, 6, 8, 2, 3

[5] Yang Chen and Gérard Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992. Range Image Understanding. 3

[6] Zhikai Chen, Fuchen Long, Zhaofan Qiu, Ting Yao, Wengang Zhou, Jiebo Luo, and Tao Mei. Anchorformer: Point cloud completion from discriminative nodes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13581–13590, 2023. 2

[7] 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 (CVPR)*, pages 4456–4465, 2023. 3

[8] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 8

[9] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 8

[10] Ruihang Chu, Enze Xie, Shentong Mo, Zhenguo Li, Matthias Niessner, Chi-Wing Fu, and Jiaya Jia. Diffcomplete: Diffusion-based generative 3d shape completion. In *Advances in Neural Information Processing Systems*, pages 75951–75966. Curran Associates, Inc., 2023. 3, 7, 8, 2

[11] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques*, page 303–312, New York, NY, USA, 1996. Association for Computing Machinery. 3

[12] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017. 8, 3, 4

[13] Angela Dai, Charles Ruizhongtai Qi, and Matthias Niessner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 8

[14] Angela Dai, Daniel Ritchie, Martin Bokeloh, Scott Reed, Jürgen Sturm, and Matthias Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4578–4587, 2018. 8

[15] 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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13142–13153, 2023. 1, 5, 6, 7, 8, 2, 3

[16] Abdelrahman Eldesokey, Michael Felsberg, Karl Holmquist, and Michael Persson. Uncertainty-aware cnns for depth completion: Uncertainty from beginning to end. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12014–12023, 2020. 3

[17] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[18] Tony Finch. Incremental calculation of weighted mean and variance. *University of Cambridge*, 4(11-5):41–42, 2009. 4

[19] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018. 1

[20] Zhiyang Huang, Nathan Carr, and Tao Ju. Variational implicit point set surfaces. *ACM Trans. Graph.*, 38(4), 2019. 1

[21] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[22] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Trans. Graph.*, 32(3), 2013. 1

[23] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, page 61–70, Goslar, DEU, 2006. Eurographics Association. 1

[24] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 8

[25] Samuli Laine, Janne Hellsten, Tero Karras, Yeongho Seol, Jaakko Lehtinen, and Timo Aila. Modular primitives for high-performance differentiable rendering. *ACM Transactions on Graphics*, 39(6), 2020. 5

[26] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. *CoRR*, abs/1912.00280, 2019. 2

[27] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 3, 5, 6, 4

[28] R. Marques, C. Bouville, M. Ribardière, L. P. Santos, and K. Bouatouch. Spherical fibonacci point sets for illumination integrals. *Computer Graphics Forum*, 32(8):134–143, 2013. 5, 6, 1

[29] 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 (CVPR)*, 2019. 2, 5

[30] Mateusz Michalkiewicz, Sarah Parisot, Stavros Tsogkas, Mahsa Baktashmotlagh, Anders Eriksson, and Eugene Belilovsky. Few-shot single-view 3-d object reconstruction with compositional priors. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV*, page 614–630, Berlin, Heidelberg, 2020. Springer-Verlag. 2, 8

[31] Paritosh Mittal, Y. Cheng, Maneesh Singh, and Shubham Tulsiani. Autosdf: Shape priors for 3d completion, reconstruction and generation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 306–315, 2022. 3, 8, 2, 4

[32] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pages 127–136, 2011. 3

[33] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8524–8533, 2021. 2

[34] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[35] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision – ECCV 2020*, pages 523–540, Cham, 2020. Springer. 2, 8

[36] Charles R. Qi, Hao Su, Kaichun Mo, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[37] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2017. 2

[38] Yuchen Rao, Yinyu Nie, and Angela Dai. Patchcomplete: Learning multi-resolution patch priors for 3d shape completion on unseen categories. In *Advances in Neural Information Processing Systems*, pages 34436–34450. Curran Associates, Inc., 2022. 3, 6, 7, 8, 1, 2

[39] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 8

[40] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *Advances in Neural Information Processing Systems*, pages 7462–7473. Curran Associates, Inc., 2020. 5

[41] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: Topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1726–1735, 2022. 2

[42] Maxim Tatarchenko, Stephan R. Richter, Rene Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 4

[43] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 383–392, 2019. 2

[44] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment — a modern synthesis. In *Vision Algorithms: Theory and Practice*, pages 298–372, Berlin, Heidelberg, 2000. Springer Berlin Heidelberg. 3

[45] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia

Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[46] Hang Wu and Yubin Miao. Voting-based patch sequence autoregression network for adaptive point cloud completion. *Comput. Graph.*, 118(C):111–122, 2024. 2

[47] Lintai Wu, Junhui Hou, Lin Song, and Yong Xu. 3d shape completion on unseen categories: A weakly-supervised approach. *IEEE transactions on visualization and computer graphics*, PP, 2024. 3

[48] Rundi Wu, Xuelin Chen, Yixin Zhuang, and Baoquan Chen. Multimodal shape completion via conditional generative adversarial networks. In *Computer Vision – ECCV 2020*, Cham, 2020. Springer International Publishing. 4

[49] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5499–5509, 2021. 2

[50] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *Computer Vision – ECCV 2020*, pages 365–381, Cham, 2020. Springer International Publishing. 2

[51] Xingguang Yan, Liqiang Lin, Niloy J Mitra, Dani Lischinski, Daniel Cohen-Or, and Hui Huang. Shapeformer: Transformer-based shape completion via sparse representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6239–6249, 2022. 1, 2

[52] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[53] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12498–12507, 2021. 2

[54] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(12):14114–14130, 2023. 2

[55] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737, 2018. 2

[56] Faezeh Zakeri, Raphael Braun, Lukas Ruppert, and Hendrik P.A. Lensch. POC-SLT: Partial Object Completion with SDF Latent Transformers. *Proceedings of the Conference on Robots and Vision*, 2025. https://crv.pubpub.org/pub/yanc7d1w. 3, 5, 6

[57] Xiaohui Zeng, Arash Vahdat, Francis Williams, Zan Gojcic, Or Litany, Sanja Fidler, and Karsten Kreis. Lion: Latent point diffusion models for 3d shape generation. In *Advances in Neural Information Processing Systems*, pages 10021–10039. Curran Associates, Inc., 2022. 2

[58] Biao Zhang, Matthias Nießner, and Peter Wonka. 3dilg: Irregular latent grids for 3d generative modeling. In *Advances in Neural Information Processing Systems 35 - 36th Conference on Neural Information Processing Systems, NeurIPS 2022*. Neural information processing systems foundation, 2022. Publisher Copyright: © 2022 Neural information processing systems foundation. All rights reserved.; 36th Conference on Neural Information Processing Systems, NeurIPS 2022 ; Conference date: 28-11-2022 Through 09-12-2022. 2

[59] Biao Zhang, Jiapeng Tang, Matthias Nießner, and Peter Wonka. 3dshape2vecset: A 3d shape representation for neural fields and generative diffusion models. *ACM Trans. Graph.*, 42(4), 2023. 2

[60] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 512–528. Springer, 2020. 2

[61] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021. 2

[62] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. In *Computer Vision – ECCV 2022*, pages 416–432, Cham, 2022. Springer Nature Switzerland. 2

[63] Zihan Zhu, Songyou Peng, Viktor Larsson, Weiwei Xu, Hujun Bao, Zhaopeng Cui, Martin R Oswald, and Marc Pollefeys. Nice-slam: Neural implicit scalable encoding for slam. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12786–12796, 2022. 1