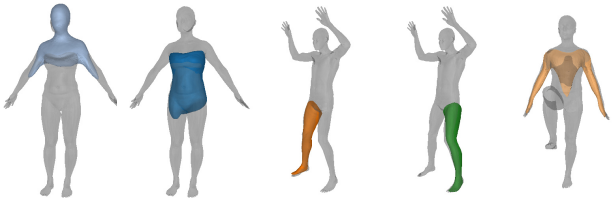
Neural Parts: Learning Expressive 3D Shape Abstractions with Invertible Neural Networks

Despoina Paschalidou[1,5,6] Angelos Katharopoulos[3,4] Andreas Geiger[1,2,5] Sanja Fidler[6,7,8]
[1]Max Planck Institute for Intelligent Systems Tübingen [2]University of Tübingen
[3]Idiap Research Institute, Switzerland [4]École Polytechique Fédérale de Lausanne (EPFL)
[5]Max Planck ETH Center for Learning Systems [6]NVIDIA [7]University of Toronto [8]Vector Institute
{firstname.lastname}@tue.mpg.de angelos.katharopoulos@idiap.ch sfidler@nvidia.com

Abstract

Impressive progress in 3D shape extraction led to representations that can capture object geometries with high fidelity. In parallel, primitive-based methods seek to represent objects as semantically consistent part arrangements. However, due to the simplicity of existing primitive representations, these methods fail to accurately reconstruct 3D shapes using a small number of primitives/parts. We address the trade-off between reconstruction quality and number of parts with Neural Parts, a novel 3D primitive representation that defines primitives using an Invertible Neural Network (INN) which implements homeomorphic mappings between a sphere and the target object. The INN allows us to compute the inverse mapping of the homeomorphism, which in turn, enables the efficient computation of both the implicit surface function of a primitive and its mesh, without any additional post-processing. Our model learns to parse 3D objects into semantically consistent part arrangements without any part-level supervision. Evaluations on ShapeNet, D-FAUST and FreiHAND demonstrate that our primitives can capture complex geometries and thus simultaneously achieve geometrically accurate as well as interpretable reconstructions using an order of magnitude fewer primitives than state-of-the-art shape abstraction methods.

(a) Convexes



(b) Neural Parts

Figure 1: **Expressive Primitives.** We address the trade-off between reconstruction quality and sparsity (i.e. number of parts) in primitive-based methods. Prior work [83, 61, 63, 16] has considered convex shapes as primitives, which due to their simplicity, require a large number of parts to accurately represent complex shapes. This results in less interpretable shape abstractions (i.e. primitives are not identifiable parts e.g. legs, arms etc.). In this work, we propose Neural Parts, a novel 3D primitive representation that can represent arbitrarily complex genus-zero shapes and thus yields geometrically more accurate and semantically more meaningful parts compared to simpler primitives.

1. Introduction

Recovering the geometry of a 3D shape from a single RGB image is a fundamental task in computer vision and graphics. Existing shape reconstruction models utilize a neural network to learn a parametric function that maps the input image into a mesh [47, 33, 41, 84, 89, 59], a point-cloud [24, 66, 1, 40, 80, 89], a voxel grid [8, 15, 26, 68, 70, 77, 87] or an implicit surface [51, 13, 60, 73, 88, 52]. An alternative line of research focuses on compact low-dimensional representations that reconstruct 3D objects by decomposing them into simpler parts, called primitives [83, 61, 16, 30]. Primitive-based representations seek to infer *semantically consistent part arrangements* across different object instances and provide a more interpretable alternative, compared to representations that only focus on capturing the global object geometry. Primitives are particularly useful for various applications where the notion of

parts is necessary, such as shape editing, physics-based applications, graphics simulation etc.

Existing primitive-based methods rely on simple shapes for decomposing complex 3D objects into parts. In the early days of computer vision, researchers explored various shape primitives such as 3D polyhedral shapes [71], generalized cylinders [6] and geons [5] for representing 3D geometries. More recently, the primitive paradigm has been revisited in the context of deep learning and [83, 61, 36, 16] have demonstrated the ability of neural networks to learn part-level geometries using 3D cuboids [83, 56, 92], superquadrics [61], spheres [36] or convexes [16]. Due to their simple parametrization, these primitives have limited expressivity and cannot capture complex geometries. Therefore, existing part-based methods require a large number of primitives for extracting geometrically accurate reconstructions. However, using more primitives comes at the expense of the interpetability of the reconstruction.

To address this, we devise Neural Parts, a novel 3D primitive representation that is more *expressive* and *interpretable*, in comparison to alternatives that are limited to convex shapes. We argue that a primitive should be a non trivial genus-zero shape with well defined implicit and explicit representations. These characteristics allow us to efficiently combine primitives and accurately represent arbitrarily complex geometries. To this end, we pose the task of primitive learning as the task of learning *a family of homeomorphic mappings* between the 3D space of a simple genus-zero shape (e.g. sphere, cube, ellipsoid) and the 3D space of the target object. We implement this mapping using an Invertible Neural Network (INN) [19]. Being able to map 3D points in both directions allows us to efficiently compute the explicit representation of each primitive, namely its tesselation as well as the implicit representation, i.e. the relative position of a point wrt. the primitive's surface. In contrast to prior work [83, 61, 16, 63] that directly predict the primitive parameters (i.e. centroids and sizes for cuboids and superquadrics and hyperplanes for convexes), we employ the INN to fully define each primitive. Note that while a homeomorphism preserves the genus of the shape it does not constrain it in any other way. As a result, while existing primitives are constrained to a specific family of shapes (e.g. ellipsoids), our primitives can capture arbitrarily complicated genus-zero shapes (see Fig. 1). We demonstrate that Neural Parts can be learned in an unsupervised fashion (i.e. without any primitive annotations), directly from unstructured 3D point clouds by ensuring that the assembly of predicted primitives accurately reconstructs the target.

In summary, we make the following **contributions**: We propose the first model that defines primitives as a homeomorphic mapping between two topological spaces through conditioning an INN on an image. Since the homeomorphism does not impose any constraints on the primitive shape, our model effectively decouples geometric accuracy from parsimony and as a result captures complex geometries with an order of magnitude fewer primitives. Experiments on ShapeNet [9], D-FAUST [7] and FreiHAND [91] demonstrate that our model can parse objects into more expressive and semantically meaningful shape abstractions compared to models that rely on simpler primitives. Code and data is available at https://paschalidoud.github.io/neural_parts.

2. Related Work

Learning-based 3D reconstruction approaches can be categorized based on the type of their output representation to: depth-based [42, 37, 62, 21], voxel-based [15, 86, 26, 68], point-based [24, 66, 1], mesh-based [47, 33, 41], implicit-based [51, 13, 60] and primitive-based [83, 56, 61]. Here, we primarily focus on primitive-based methods that are more relevant to our work. Since our formulation is independent of a specific INN implementation, a thorough discussion of INNs is beyond the scope of this paper, thus we refer the reader to [2] for a detailed overview.

3D Representations: Voxels [8, 15, 86, 26, 68, 77, 87] naturally capture the 3D geometry by discretizing the shape into a regular grid. While several efficient space partitioning techniques [50, 70, 79, 35, 69] have been proposed to address their high memory and computation requirements, their application is still limited. A promising new direction explored learning a deformation of the grid itself to better capture geometric details [27]. Pointclouds [24, 66, 1, 40, 80, 89] are more memory efficient but lack surface connectivity, thus post-processing is necessary for generating the final mesh. Most mesh-based methods [47, 33, 41, 84, 89, 31, 59, 10, 16, 90] naturally yield smooth reconstructions but either require a deformable template mesh [84] or represent the geometry as an atlas of multiple mappings [33, 17, 49]. To address these limitations, implicit models [51, 13, 60, 73, 88, 52, 54, 57, 3, 85, 29, 12, 55, 4, 32, 74, 39, 14, 65, 67, 58] have recently gained popularity. These methods represent a 3D shape as the level-set of a distance or occupancy field implemented as a neural network, that takes a context vector and a query point and predicts either a signed distance value [60, 52, 4, 32, 78] or a binary occupancy value [51, 13] for the query point. While these methods result in accurate reconstructions, they lack interpretability as they do not consider the part-based object structure. Instead, in this work, we focus on part-based representations and showcase that our model simultaneously yields both geometrically accurate and interpretable reconstructions. Furthermore, in contrast to implicit models, that require expensive iso-surfacing operations (i.e. marching cubes) to extract a mesh, our model directly predicts a high resolution mesh for each part, without any post-processing.

Structured-based Representations: Our work falls into the category of shape abstraction techniques. This line of research seeks to decompose 3D shapes into semantically meaningful simpler parts using either supervision in terms of the primitive parameters [92, 56, 44, 53, 28, 45, 76, 46] or without any part-level annotations [83, 61, 18, 63, 30, 16, 43]. Neural Parts perform primitive-based learning in an unsupervised manner. Traditional primitives include cuboids [83, 56, 92, 44, 53, 22], superquadrics [61, 63], convexes [16, 11, 25], CSG trees [75] or shape programms [23, 81, 48]. Due to the simplicity of the shapes of traditional primitives, the reconstruction quality of existing part-based methods is coupled with the number of primitives, namely a larger number of primitives results in more accurate reconstructions (see Fig. 5). However, these reconstructions are less parsimonious and the constituent primitives often lack a semantic interpretation (i.e. are not recognizable parts). Instead, Neural Parts are not restricted to convex shapes and can capture complex geometries with a few primitives. In recent work, [29] propose a 3D representation that decomposes space into a structured set of implicit functions [30]. However, extracting a single part from their prediction is not possible. This is not the case for our model.

Shape Deformations: Deforming a single genus-zero shape into more complicated shapes with graph convolutions [84], MLPs [33, 82] and Neural ODEs [34] has demonstrated impressive results. Groueix et al. [33] were among the first to employ an MLP to implement a homeomorphism between a sphere and a complicated shape. Note that since the deformation is implemented via an MLP, computing the inverse mapping becomes infeasible. Very recently, [34] proposed to learn the deformation of a single ellipsoid using several Neural ODEs. While [34] propose an invertible model, it does not consider any part decomposition or latent object structure. Instead, we use the inverse mapping of the homeomorphism to define the predicted shape as the union of primitives, by discarding points from a part's surface that are internal to any other part. Thus our model learns to combine multiple genus-zero primitives and is able to reconstruct shapes of arbitrary genus. In addition, we formulate our optimization objective using both the forward and the inverse mapping of the homeomorphism, which in turn allows us to utilize both volumetric and surface information during training. This facilitates imposing additional constraints on the predicted primitives e.g. normal consistency and parsimony and improves performance.

3. Method

Given an input image we seek to learn a representation with M primitives that best describes the target object. We define our primitives via a *deformation between shapes* that is parametrized as a *learned homeomorphism* implemented with an Invertible Neural Network (INN). Using an INN allows us to efficiently compute the implicit and explicit representation of the predicted shape and impose various constraints on the predicted parts. In particular, for each primitive, we seek to learn a homeomorphism between the 3D space of a simple genus-zero shape and the 3D space of the target object, such that the deformed shape matches a part of the target object. Due to its simple implicit surface definition and tesselation, we employ a sphere as our genus-zero shape. We refer to the 3D space of the sphere as *latent space* and to the 3D space of the target as *primitive space*.

In Sec. 3.1, we present the explicit and implicit representation of our primitives as a homeomorphism of a sphere. Subsequently, in Sec. 3.2, we present our novel architecture for predicting multiple primitives using homeomorphisms conditioned on the input image. Finally, in Sec. 3.3, we formulate our optimization objective.

3.1. Primitives as Homeomorphic Mappings

A homeomorphism is a continuous map between two topological spaces Y and X that preserves all topological properties. Intuitively a homeomorphism is a continuous stretching and bending of Y into a new space X. In our 3D topology, a homeomorphism $\phi_{\boldsymbol{\theta}} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is

$$\mathbf{x} = \phi_{\boldsymbol{\theta}}(\mathbf{y}) \text{ and } \mathbf{y} = \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}) \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are 3D points in X and Y and $\phi_{\boldsymbol{\theta}} : Y \rightarrow X$, $\phi_{\boldsymbol{\theta}}^{-1} : X \rightarrow Y$ are continuous bijections. In our setting, Y and X correspond to the latent and the primitive space respectively. Using the explicit and implicit representation of a sphere with radius r, positioned at $(0, 0, 0)$ and the homeomorphic mapping from (1), we can now define the implicit and explicit representation of a single primitive.

Explicit Representation: The explicit representation of a primitive, parametrized as a mesh with vertices $\mathcal{V}_p$ and faces $\mathcal{F}_p$, can be obtained by applying the homeomorphism on the sphere vertices $\mathcal{V}$ and faces $\mathcal{F}$ as follows:

$$\begin{aligned} \mathcal{V}_p &= \{\phi_{\boldsymbol{\theta}}(\mathbf{v}_j),\ \forall\ \mathbf{v}_j \in \mathcal{V}\} \\ \mathcal{F}_p &= \mathcal{F}. \end{aligned} \tag{2}$$

Note that applying $\phi_{\boldsymbol{\theta}}$ on the sphere vertices $\mathcal{V}$ alters their location in the primitive space, while the vertex arrangements (i.e. faces) remain unchanged. Furthermore, since our primitives are defined as a deformation of a sphere mesh of arbitrarily high resolution, we can also obtain primitive meshes of arbitrary resolutions without any post-processing, e.g. marching-cubes.

Implicit Representation: The implicit representation of a primitive can be derived by applying the inverse homeomorphic mapping on a 3D point $\mathbf{x}$ as follows

$$g(\mathbf{x}) = \|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})\|_2 - r. \tag{3}$$

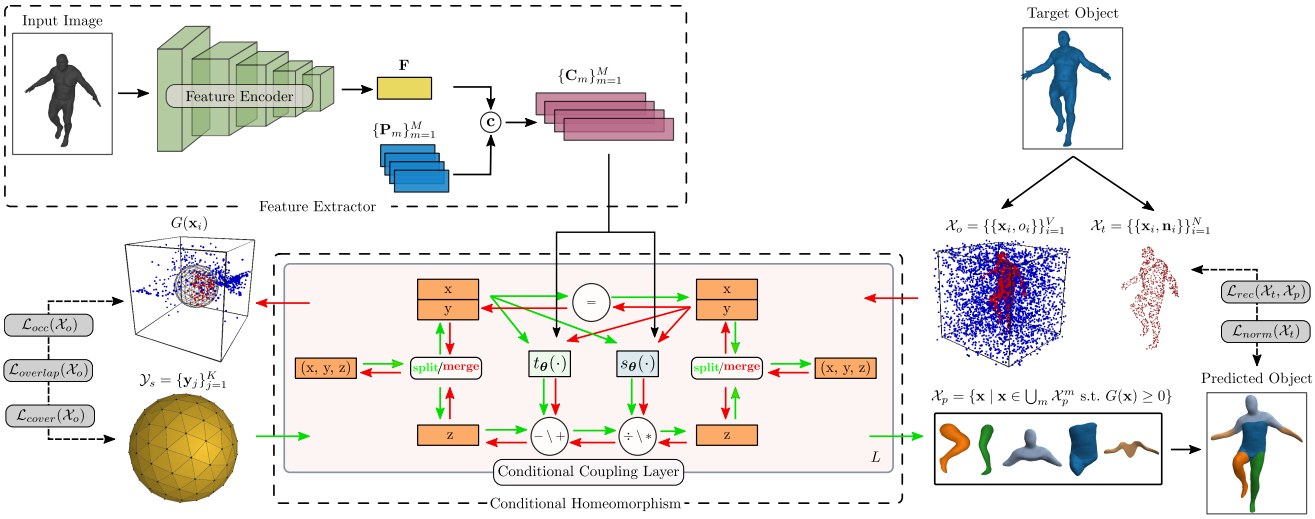


Figure 2: **Method Overview.** Our model comprises two main components: A *Feature Extractor* which maps an input image into a per-primitive shape embedding and a *Conditional Homeomorphism* that deforms a sphere into M primitives and vice-versa. First, the *feature encoder* maps the input to a global feature representation $\mathbf{F}$. Then, for every primitive m, $\mathbf{F}$ is concatenated with a learnable primitive embedding $\mathbf{P}_m$ to generate the shape embedding $\mathbf{C}_m$ for this primitive. The Conditional Homeomorphism $\phi_{\boldsymbol{\theta}}(\cdot; \mathbf{C}_m)$ is implemented by a stack of L conditional coupling layers. Applying the *forward mapping* on a set of points $\mathcal{Y}_s$, randomly sampled on the surface of the sphere, generates points on the surface of the m-th primitive $\mathcal{X}_p^m$ (9). Using the *inverse mapping* $\phi_{\boldsymbol{\theta}}^{-1}(\cdot; \mathbf{C}_m)$, allows us to compute whether any point in 3D space lies inside or outside a primitive (7). We train our model using both *surface* ($\mathcal{L}_{rec}, \mathcal{L}_{norm}$) and *occupancy* ($\mathcal{L}_{occ}$) losses to simultaneously capture fine object details and volumetric characteristics of the target object. The use of the *inverse mapping* allows us to impose additional constraints (e.g. discouraging inter-penetration) on the predicted primitives ($\mathcal{L}_{overlap}, \mathcal{L}_{cover}$).

To evaluate the relative position of a point $\mathbf{x}$ wrt. the primitive surface it suffices to evaluate whether $\phi_{\theta}^{-1}(\mathbf{x})$ lies inside, outside or on the surface of the sphere. Namely, points that are internal to the sphere are also inside the primitive and points that are outside the sphere are also outside the primitive surface. Note that computing $g(\mathbf{x})$ in (3) is only possible because $\phi_{\boldsymbol{\theta}}(\mathbf{x})$ is implemented with an INN.

Multiple Primitives: The homeomorphism in (1) implements a single deformation. However, we seek to predict multiple primitives (i.e. deformations) conditioned on the input. Hence, we define a conditional homeomorphism as:

$$\mathbf{x} = \phi_{\boldsymbol{\theta}}(\mathbf{y}; \mathbf{C}_m) \text{ and } \mathbf{y} = \phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m) \tag{4}$$

where $\mathbf{C}_m$ is the shape embedding for the m-th primitive and is predicted from the input. Note that for different shape embeddings a different homeomorphism is defined.

3.2. Network Architecture

Our architecture comprises two main components: (i) the *feature extractor* that maps the input to a vector of per-primitive shape embeddings $\{\mathbf{C}_m\}_{m=1}^M$ and (ii) the *conditional homeomorphism* that learns a homeomorphic mapping conditioned on the shape embedding. The overall architecture is illustrated in Fig. 2.

Feature Extractor: The first part of the feature extractor module is a ResNet-18 [38] that extracts a feature representation $\mathbf{F} \in \mathbb{R}^D$ from the input image. Subsequently, for every primitive m, $\mathbf{F}$ is concatenated with a learnable primitive embedding $\mathbf{P}_m \in \mathbb{R}^D$ to derive a shape embedding $\mathbf{C}_m \in \mathbb{R}^{2D}$ for this primitive.

Conditional Homeomorphism: We implement the INN using a Real NVP [20] due to its simple formulation. A Real NVP models a bijective mapping by stacking a sequence of simple bijective transformation functions. For each bijection, typically referred to as *affine coupling layer*, given an input 3D point (x_i, y_i, z_i), the output point (x_o, y_o, z_o) is

$$\begin{aligned} x_o &= x_i \\ y_o &= y_i \\ z_o &= z_i \exp\left(s_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right) + t_{\boldsymbol{\theta}}\left(x_i, y_i\right) \end{aligned} \tag{5}$$

where $s_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$ and $t_{\boldsymbol{\theta}} : \mathbb{R}^2 \rightarrow \mathbb{R}$ are *scale* and *translation* functions implemented with two arbitrarily complicated networks. Namely, in each bijection, the input is split into two, (x_i, y_i) and z_i. The first part remains unchanged and the second undergoes an affine transformation with $s_{\boldsymbol{\theta}}(\cdot)$ and $t_{\boldsymbol{\theta}}(\cdot)$. We follow [20] and we enforce that consecutive affine coupling layers scale and translate different input dimensions. Namely, we alternate the splitting

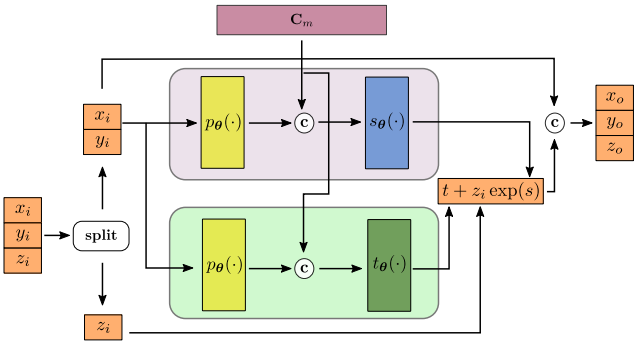


Figure 3: **Conditional Coupling Layer.** Pictorial representation of (6). The input point (x_i, y_i, z_i) is passed into a *coupling layer* that scales and translates one dimension of the input based on the other two and the per-primitive shape embedding $\mathbf{C}_m$. The scale factor s and the translation amount t are predicted by two MLPs, $s_{\boldsymbol{\theta}}(\cdot)$ and $t_{\boldsymbol{\theta}}(\cdot)$. $p_{\boldsymbol{\theta}}(\cdot)$ is another MLP that increases the dimensionality of the input point before it is concatenated with $\mathbf{C}_m$.

between the dimensions of the input randomly.

However, the original Real NVP cannot be directly applied in our setting as it does not consider a shape embedding. To address this, we augment the affine coupling layer as follows: we first map (x_i, y_i) into a higher dimensional feature vector using a mapping, $p_{\boldsymbol{\theta}}(\cdot)$, implemented as an MLP. This is done to increase the relative importance of the input point before concatenating it with the high-dimensional shape embedding $\mathbf{C}_m$. The *conditional affine coupling layer* becomes

$$\begin{aligned} x_o &= x_i \\ y_o &= y_i \\ z_o &= z_i \exp\left(s_{\boldsymbol{\theta}}\left(\left[\mathbf{C}_m; p_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right]\right)\right) \\ &\quad + t_{\boldsymbol{\theta}}\left(\left[\mathbf{C}_m; p_{\boldsymbol{\theta}}\left(x_i, y_i\right)\right]\right) \end{aligned} \tag{6}$$

where $[\cdot;\cdot]$ denotes concatenation. A graphical representation of our conditional coupling layer is provided in Fig. 3.

3.3. Training

Due to the lack of primitive annotations, we train our model by minimizing the geometric distance between the target and the predicted shape. In the following, we define the implicit and explicit representation of the predicted shape as the union of M primitives.

The implicit surface of the m-th primitive can be derived from (3), by applying the inverse homeomorphic mapping on points in $3D$ space as follows

$$g^m(\mathbf{x}) = \left\|\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x}; \mathbf{C}_m)\right\|_2 - r, \ \forall\ \mathbf{x} \in \mathbb{R}^3 \tag{7}$$

The implicit surface representation of the predicted object is defined as the union of all per-primitive implicit functions

$$G(\mathbf{x}) = \min_{m \in 0 \ldots M} g^m(\mathbf{x}), \tag{8}$$

namely a point is inside the predicted shape if it is inside at least one primitive.

Similarly, the explicit representation of the m-th primitive is a set of points on its surface, let it be $\mathcal{X}_p^m$, that are generated by applying the forward homeomorphic mapping on points on the sphere surface $\mathcal{Y}_s$ in the latent space

$$\mathcal{X}_p^m = \{\phi_{\boldsymbol{\theta}}(\mathbf{y}_j; \mathbf{C}_m),\ \forall\ \mathbf{y}_j \in \mathcal{Y}_s\}. \tag{9}$$

To generate points on the surface of the predicted shape $\mathcal{X}_p$, we first need to generate points on the surface of each primitive and then discard the ones that are inside any other primitive. From (8) this can be expressed as follows:

$$\mathcal{X}_p = \{\mathbf{x} \mid \mathbf{x} \in \bigcup_m \mathcal{X}_p^m \text{ s.t. } G(\mathbf{x}) \geq 0\}. \tag{10}$$

Loss Functions: Our loss $\mathcal{L}$ seeks to minimize the geometric distance between the target and the predicted shape and is composed of five loss terms:

$$\begin{aligned} \mathcal{L} = \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) + \mathcal{L}_{occ}(\mathcal{X}_o) + \mathcal{L}_{norm}(\mathcal{X}_t) \\ + \mathcal{L}_{overlap}(\mathcal{X}_o) + \mathcal{L}_{cover}(\mathcal{X}_o) \end{aligned} \tag{11}$$

where $\mathcal{X}_t = \{\{\mathbf{x}_i, \mathbf{n}_i\}\}_{i=1}^N$ comprises surface samples of the target shape and the corresponding normals, and $\mathcal{X}_o = \{\{\mathbf{x}_i, o_i\}\}_{i=1}^V$ denotes a set of occupancy pairs, where $\mathbf{x}_i$ corresponds to the location of the i-th point and o_i denotes whether $\mathbf{x}_i$ lies inside $(o_i = 1)$ or outside $(o_i = 0)$ the target. Note that our optimization objective comprises both occupancy (13) and surface losses (12)+(14), since they model complementary characteristics of the target object e.g. the surface loss attends to fine details that may have small volume, whereas the occupancy loss more efficiently models empty space. We empirically observe that using both significantly improves reconstruction (see Sec. 4.3).

Reconstruction Loss: We measure the surface reconstruction quality using a bidirectional Chamfer loss between the points $\mathcal{X}_p$ on the surface of the predicted shape and the points on the target object $\mathcal{X}_t$ as follows:

$$\begin{aligned} \mathcal{L}_{rec}(\mathcal{X}_t, \mathcal{X}_p) = &\frac{1}{|\mathcal{X}_t|} \sum_{\mathbf{x}_i \in \mathcal{X}_t} \min_{\mathbf{x}_j \in \mathcal{X}_p} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 + \\ &\frac{1}{|\mathcal{X}_p|} \sum_{\mathbf{x}_j \in \mathcal{X}_p} \min_{\mathbf{x}_i \in \mathcal{X}_t} \|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \end{aligned} \tag{12}$$

The first term of (12) measures the average distance of all ground truth points to the closest predicted points and the second term measures the average distance of all predicted points to the closest ground-truth points.

Occupancy Loss: The occupancy loss ensures that the volume of the predicted shape matches the volume of the target. Intuitively, we want to ensure that the free and the occupied space of the predicted and the target object coincide.

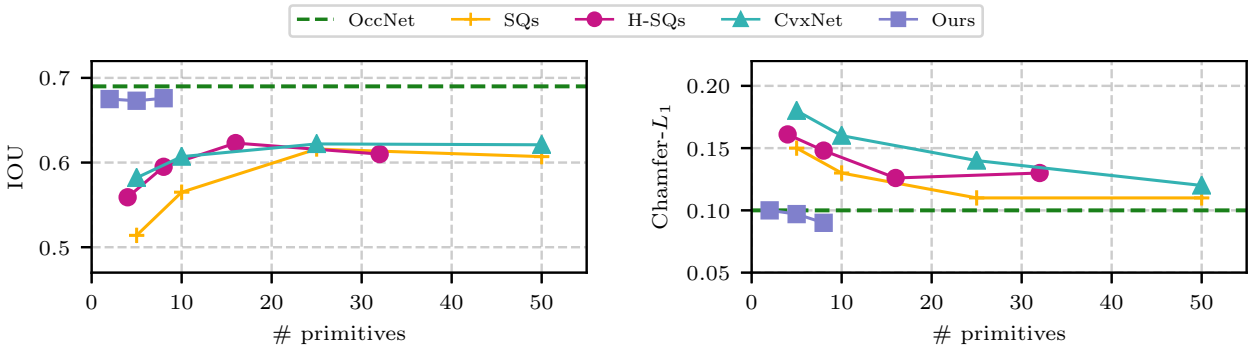


Figure 4: **Trade-off Reconstruction Quality and # Primitives**. We evaluate the reconstruction quality of primitive-based methods on the D-FAUST test set for different number of primitives. Neural Parts (**purple**) outperform CvxNet (**turquoise**), SQs (**orange**) and H-SQs (**magenta**) in terms of both IoU (↑) and Chamfer-L1 (↓) for any primitive configuration, even when using as little as 2 primitives. In addition, we show that our reconstructions are competitive to OccNet (dashed) which does not provide a primitive-based representation and requires expensive post-processing for extracting surface meshes.

To this end, we convert the implicit surface of the predicted shape from (8) to an indicator function and compute the binary cross-entropy loss over all volume samples $\mathcal{X}_o$

$$\mathcal{L}_{occ}(\mathcal{X}_o) = \sum_{(\mathbf{x},o)\in\mathcal{X}_o} \mathcal{L}_{ce}\left(\sigma\left(\frac{-G(\mathbf{x})}{\tau}\right), o\right). \tag{13}$$

$\mathcal{L}_{ce}(\cdot,\cdot)$ is the cross-entropy loss, $\sigma(\cdot)$ is the sigmoid function and τ is a temperature hyperparameter that defines the sharpness of the boundary of the indicator function. Note that $\sigma\left(\frac{-G(\mathbf{x})}{\tau}\right)$ is 1 when $\mathbf{x}$ is inside the predicted shape and 0 otherwise.

Normal Consistency Loss: The normal consistency loss ensures that the orientation of the normals of the predicted shape will be aligned with the normals of the target. We penalize misalignments between the predicted and the target normals by minimizing the cosine distance as follows

$$\mathcal{L}_{norm}(\mathcal{X}_t) = \frac{1}{|\mathcal{X}_t|} \sum_{(\mathbf{x},\mathbf{n})\in\mathcal{X}_t} \left(1 - \left\langle \frac{\nabla_{\mathbf{x}} G(\mathbf{x})}{\|\nabla_{\mathbf{x}} G(\mathbf{x})\|_2}, n \right\rangle\right) \tag{14}$$

where $\langle\cdot,\cdot\rangle$ is the dot product. Note that the surface normal of the predicted shape for a point $\mathbf{x}$ is simply the gradient of the implicit surface wrt. to point $\mathbf{x}$ and can be efficiently computed with automatic differentiation.

Overlapping Loss: To encourage semantically meaningful shape abstractions, (i.e. primitives represent different object parts), we introduce a non-overlapping loss that penalizes any point in space that is internal to more than λ primitives

$$\mathcal{L}_{overlap}(\mathcal{X}_o) = \frac{1}{|\mathcal{X}_o|} \max\left(0, \sum_{m=1}^{M} \sigma\left(\frac{-g^m(\mathbf{x})}{\tau}\right) - \lambda\right) \tag{15}$$

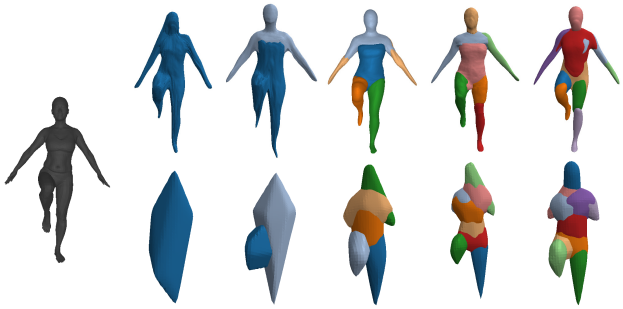


Figure 5: **Human body modelling**. We visualize the target mesh and the predicted primitives with Neural Parts (first row) and CvxNet (second) using $1, 2, 5, 8$ and 10 primitives.

Coverage Loss: The coverage loss makes sure that all primitives cover parts of the predicted shape. In practice, it prevents degenerate primitive arrangements, where some primitives are very small and do not contribute to the reconstruction. We implement this loss by encouraging that each primitive contains at least k points of the target object:

$$\mathcal{L}_{cover}(\mathcal{X}_o) = \sum_{m=1}^{M} \sum_{\mathbf{x}\in\mathcal{N}_k^m} \max\left(0, g^m(\mathbf{x})\right). \tag{16}$$

Here $\mathcal{N}_k^m \subset \{(\mathbf{x},o) \in \mathcal{X}_o | o = 1\}$ contains the k points with the minimum distance from the m-th primitive.

4. Experimental Evaluation

Datasets: We evaluate our model on D-FAUST [7], FreiHAND [91] and ShapeNet [9]. For ShapeNet [9], we perform category specific training using the same image renderings and train/test splits as [15]. For D-FAUST [7], we follow the experimental evaluation proposed in [63] and for FreiHAND [91], we select the first 5000 hand poses

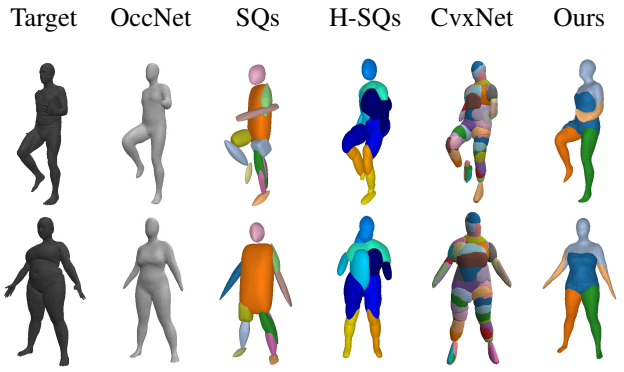


Figure 6: **Single Image 3D Reconstruction on D-FAUST**. The input image is shown on the first column and the rest contain predictions of all methods: OccNet (second), primitive-based predictions with superquadrics (third and fourth) and convexes (fifth) and ours with 5 primitives (last).

and generate meshes using the provided MANO parameters [72]. Details regarding data preprocessing as well as additional results are provided in the supplementary.

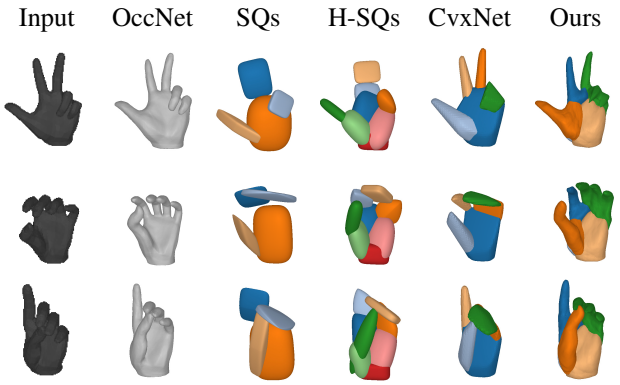
Baselines: We compare against various primitive-based methods: SQs [61] that employ superquadrics, CvxNet [16] that use smooth convexes and H-SQs [63] that consider a hierarchical, geometrically more accurate decomposition of parts using superquadrics. Finally, we also report results for OccNet [51], a state-of-the-art implicit-based method that does not reason about object parts.

4.1. Representation Power

In this experiment, we train our model and our baselines on D-FAUST for different number of primitives and measure their reconstruction quality wrt. IoU and Chamfer-L_1 distance. In particular, we train our model for $2, 5$ and 8 primitives, CvxNet [16] and SQs [61] for $5, 10, 25$ and 50 primitives and H-SQs [63] for $4, 8, 16$ and 32 primitives. We observe that our model achieves more accurate reconstructions for any given number of primitives and is competitive to OccNet that does not reason about parts. In particular, our model achieves 67.3% IOU with only 5 primitives, whereas CvxNet, with 10 times more primitives, achieves 62% (see Fig. 4). In Fig. 5, we provide a qualitative comparison of reconstructions with different number of parts and we observe that Neural Parts accurately capture the human limbs with as little as one or two primitives, whereas CvxNet cannot capture the arms even with 10 primitives.

4.2. Reconstruction Accuracy

Dynamic FAUST: In this experiment, we compare CvxNet and SQs with 50 primitives and H-SQs for a maximum number of 32 primitives to Neural Parts with 5 primitives to showcase that our model can capture the human body's geometry using an order of magnitude less primitives. Qualitative results from the predicted primitives using our model and the baselines are summarized in Fig. 6. Note that OccNet is not directly comparable with part-based methods, however, we include it in our analysis as a typical representative of powerful implicit shape extraction techniques. We observe that while all methods roughly capture the human pose, Neural Parts result in a part assembly that is very close to the target object. While CvxNet with 50 primitives yield fairly accurate reconstructions, the final representation lacks any part-level semantic interpretation. The quantitative evaluation of this experiment is provided in Fig. 4.

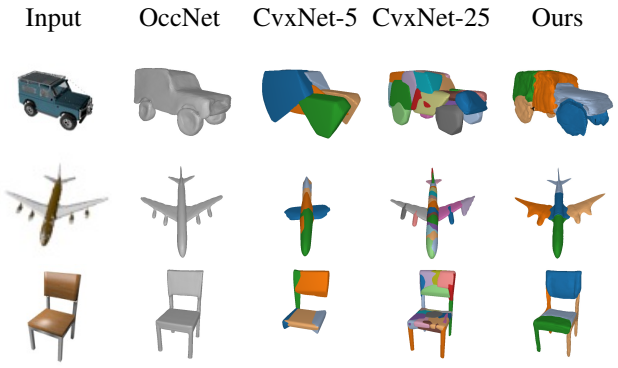


	OccNet	SQs	H-SQs	CvxNet	Ours
IoU	0.891	0.693	0.768	0.832	**0.879**
Chamfer-L_1	0.038	0.093	0.077	0.059	**0.057**

Figure 7: **Single Image 3D Reconstruction on FreiHAND**. We compare our model with OccNet, SQs and CvxNet with 5 primitives and H-SQs with 8 primitives. Our model outperforms all primitive based methods in terms of both IoU (↑) and Chamfer-L1 (↓) distance.

FreiHAND: Similarly, we train our model, CvxNet and SQs with 5 and H-SQs with 8 primitives on the FreiHAND dataset and we observe that Neural Parts yield more geometrically accurate reconstructions that faithfully capture fine details, i.e. the position of the thumb, (see Fig. 7). In contrast, CvxNet, H-SQs, SQs focus primarily on the structure of the predicted shape and miss out fine details.

ShapeNet: We train our model with 5 primitives and CvxNet with 5 and 25 primitives on cars, planes and chairs and observe that our model results in more accurate reconstructions than CvxNet with both 5 and 25 primitives (see Fig. 8). When increasing the number of primitives to 25, CvxNet improves in terms of reconstruction quality but the predicted primitives lack semantic interpretation. For the case of chairs and planes, CvxNet with 25 primitives accurately capture the object's geometry, but when we reduce the primitives to 5 entire object parts are missing.



IoU	OccNet	CvxNet - 5	CvxNet - 25	Ours
cars	0.763	0.650	0.666	**0.697**
planes	0.451	0.425	0.448	**0.454**
chairs	0.432	0.364	0.392	**0.412**

Figure 8: **Single Image 3D Reconstruction on ShapeNet**. We compare Neural Parts to OccNet and CvxNet with 5 and 25 primitives. Our model yields semantic and more accurate reconstructions with $5\times$ less primitives.

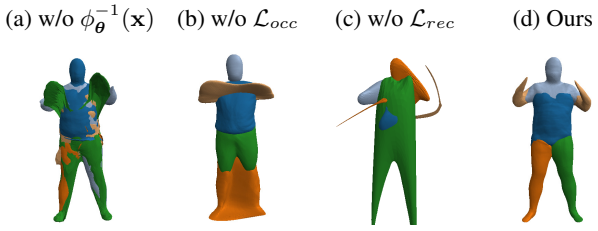
4.3. Ablation Study

In this section, we ablate various components of our model to evaluate their impact on the single-view 3D reconstruction task on D-FAUST with 5 primitives.

Invertibility: To investigate the impact of the INN, we train our model without using the inverse mapping, $\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$. As a result, it is not possible to either compute the union of parts or enforce additional constraints on the predicted primitives i.e. we only train with the loss of (12), without discarding internal points as in (9). From Fig. 9a it becomes evident that each primitive seeks to cover the entire shape thus resulting in redundant and non semantic primitives. In addition, the lack of an inverse mapping prevents us from using any occupancy loss, which results in poor modelling of empty space (i.e. the hands are connected with the body).

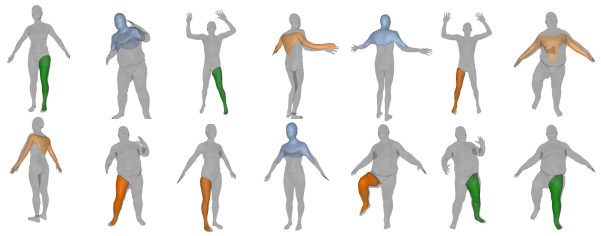
Occupancy and Surface Losses: We compare the performance of our model using either the occupancy (13) or the surface loss (12). The reconstruction without $\mathcal{L}_{occ}$ (Fig. 9b) accurately captures the geometry of the human but fails to efficiently model empty space (i.e. legs are connected). Similarly, the reconstruction without $\mathcal{L}_{rec}$ (Fig. 9c) results in degenerate primitives with small volume that are "pushed" far away from the human body.

Semantic Consistency: We now investigate the ability of our model to decompose 3D objects into semantically consistent parts. Similar to [16], we use 5 representative vertex indices provided by SMPL-X [64] (i.e. thumbs, toes and nose) and compute the classification accuracy of those points when using the label of the closest primitive. We compare with CvxNet with 5 and 50 primitives and note that



	w/o $\phi_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$	w/o $\mathcal{L}_{occ}$	w/o $\mathcal{L}_{rec}$	Ours
IoU	0.639	0.642	0.643	0.673
Chamfer-L_1	0.119	0.125	0.150	0.09

Figure 9: **Ablation Study.** We ablate the INN as well as the volumetric and the surface loss and show their impact both quantitatively and qualitatively.



	L-thumb	R-thumb	L-toe	R-toe	Nose
CvxNet-5	61.1%	67.1%	98.2%	91.2%	98%
CvxNet-50	29%	37%	56.3%	58.1%	52%
Ours	91.9%	88.2%	99.8%	92.5%	100%

Figure 10: **Semantic Consistency.** We report the classification accuracy of semantic vertices on the human body using the label of the closest primitive. Our predicted primitives are consistently used for representing the same human part.

Neural Parts are more semantically consistent (see Fig. 10).

5. Conclusion

We consider this paper a step towards bridging the gap between interpretable and geometrically accurate primitive-based representations. Our experiments demonstrate that our model yields geometrically accurate and semantically meaningful shape abstractions. In addition, we show that Neural Parts outperform existing methods, that rely on simpler shapes, both in terms of accuracy and semanticness. In future work, we plan to investigate learning primitive-based decompositions without any 3D supervision, but using differentiable rendering techniques. In addition, we would like to experiment with more complex shapes than spheres, to further improve the expressivity of our primitives.

Acknowledgments

This research was supported by the Max Planck ETH Center for Learning Systems and an NVIDIA research gift.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas J. Guibas. Learning representations and generative models for 3d point clouds. In *Proc. of the International Conf. on Machine learning (ICML)*, 2018. 1, 2

[2] Lynton Ardizzone, Jakob Kruse, Carsten Rother, and Ullrich Köthe. Analyzing inverse problems with invertible neural networks. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 2

[3] Matan Atzmon, Niv Haim, Lior Yariv, Ofer Israelov, Haggai Maron, and Yaron Lipman. Controlling neural level sets. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[4] Matan Atzmon and Yaron Lipman. SAL: sign agnostic learning of shapes from raw data. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2562–2571, 2020. 2

[5] Irving Biederman. Human image understanding: Recent research and a theory. *Computer Vision, Graphics, and Image Processing*, 1986. 2

[6] I Binford. Visual perception by computer. In *IEEE Conference of Systems and Control*, 1971. 2

[7] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: registering human bodies in motion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 6

[8] André Brock, Theodore Lim, James M. Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv.org*, 1608.04236, 2016. 1, 2

[9] 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. *arXiv.org*, 1512.03012, 2015. 2, 6

[10] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[11] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. Bsp-net: Generating compact meshes via binary space partitioning. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 42–51, 2020. 3

[12] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. BAE-NET: branched autoencoder for shape co-segmentation. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[13] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[14] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[15] Christopher Bongsoo Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2016. 1, 2, 6

[16] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey Hinton, and Andrea Tagliasacchi. Cvxnets: Learnable convex decomposition. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 7, 8

[17] Zhantao Deng, Jan Bednařík, Mathieu Salzmann, and Pascal Fua. Better patch stitching for parametric surface reconstruction. 2020. 2

[18] Theo Deprelle, Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. Learning elementary structures for 3d shape generation and matching. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 3

[19] Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: non-linear independent components estimation. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2015. 2

[20] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2017. 4

[21] Simon Donne and Andreas Geiger. Learning non-volumetric depth fusion using successive reprojections. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 2

[22] Anastasia Dubrovina, Fei Xia, Panos Achlioptas, Mira Shalah, Raphaël Groscot, and Leonidas J. Guibas. Composite shape modeling via latent space factorization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 8139–8148. IEEE, 2019. 3

[23] Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Joshua B. Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *Advances in Neural Information Processing Systems (NIPS)*, 2018. 3

[24] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[25] Matheus Gadelha, Giorgio Gori, Duygu Ceylan, Radomír Mech, Nathan Carr, Tamy Boubekeur, Rui Wang, and Subhransu Maji. Learning generative models of shape handles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 3

[26] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2017. 1, 2

[27] Jun Gao, Wenzheng Chen, Tommy Xiang, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Learning deformable tetrahedral meshes for 3d reconstruction. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[28] Lin Gao, Jie Yang, Tong Wu, Yu-Jie Yuan, Hongbo Fu, Yu-Kun Lai, and Hao Zhang. SDM-NET: deep generative network for structured deformable mesh. In *ACM SIGGRAPH Conference and Exhibition on Computer Graphics and Interactive Techniques in Asia (SIGGRAPH Asia)*, 2019. 3

[29] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A. Funkhouser. Local deep implicit functions for 3d shape. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2, 3

[30] Kyle Genova, Forrester Cole, Daniel Vlasic, Aaron Sarna, William T Freeman, and Thomas Funkhouser. Learning shape templates with structured implicit functions. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 3

[31] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh R-CNN. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[32] Amos Gropp, Lior Yariv, Niv Haim, Matan Atzmon, and Yaron Lipman. Implicit geometric regularization for learning shapes. In *Proc. of the International Conf. on Machine learning (ICML)*, 2020. 2

[33] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. AtlasNet: A papier-mâché approach to learning 3d surface generation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2, 3

[34] Kunal Gupta and Manmohan Chandraker. Neural mesh flow: 3d manifold mesh generationvia diffeomorphic flows. *arXiv.org*, 2020. 3

[35] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. *arXiv.org*, 1704.00710, 2017. 2

[36] Zekun Hao, Hadar Averbuch-Elor, Noah Snavely, and Serge J. Belongie. Dualsdf: Semantic shape manipulation using a two-level representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[37] Wilfried Hartmann, Silvano Galliani, Michal Havlena, Luc Van Gool, and Konrad Schindler. Learned multi-patch similarity. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[39] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas Funkhouser. Local implicit grid representations for 3d scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[40] Li Jiang, Shaoshuai Shi, Xiaojuan Qi, and Jiaya Jia. GAL: geometric adversarial loss for single-view 3d-object reconstruction. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1, 2

[41] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1, 2

[42] Abhishek Kar, Christian Häne, and Jitendra Malik. Learning a multi-view stereo machine. In *Advances in Neural Information Processing Systems (NIPS)*, 2017. 2

[43] Yuki Kawana, Yusuke Mukuta, and Tatsuya Harada. Neural star domain as primitive representation. In *Advances in Neural Information Processing Systems (NIPS)*, 2020. 3

[44] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao (Richard) Zhang, and Leonidas J. Guibas. GRASS: generative recursive autoencoders for shape structures. *ACM Trans. on Graphics*, 36(4), 2017. 3

[45] Lingxiao Li, Minhyuk Sung, Anastasia Dubrovina, Li Yi, and Leonidas Guibas. Supervised fitting of geometric primitives to 3d point clouds. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 3

[46] Yichen Li, Kaichun Mo, Lin Shao, Minhyuk Sung, and Leonidas J. Guibas. Learning 3d part assembly from a single image. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 3

[47] Yiyi Liao, Simon Donne, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[48] Yunchao Liu, Zheng Wu, Daniel Ritchie, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Learning to describe scenes with programs. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 3

[49] Qianli Ma, Shunsuke Saito, Jinlong Yang, Siyu Tang, and Michael J. Black. SCALE: Modeling clothed humans with a surface codec of articulated local elements. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[50] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2015. 2

[51] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 7

[52] Mateusz Michalkiewicz, Jhony K Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders Eriksson. Implicit surface representations as layers in neural networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[53] Kaichun Mo, Paul Guerrero, Li Yi, Hao Su, Peter Wonka, Niloy Mitra, and Leonidas Guibas. Structurenet: Hierarchical graph networks for 3d shape generation. In *ACM Trans. on Graphics*, 2019. 3

[54] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Occupancy flow: 4d reconstruction by learning particle dynamics. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[55] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[56] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single RGB image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[57] Michael Oechsle, Lars Mescheder, Michael Niemeyer, Thilo Strauss, and Andreas Geiger. Texture fields: Learning texture representations in function space. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2

[58] Michael Oechsle, Michael Niemeyer, Christian Reiser, Lars Mescheder, Thilo Strauss, and Andreas Geiger. Learning implicit surface light fields. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2020. 2

[59] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. Deep mesh reconstruction from single RGB images via topology modification networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[60] Jeong Joon Park, Peter Florence, Julian Straub, Richard A. Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2

[61] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 3, 7

[62] Despoina Paschalidou, Ali Osman Ulusoy, Carolin Schmitt, Luc van Gool, and Andreas Geiger. Raynet: Learning volumetric 3d reconstruction with ray potentials. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2

[63] Despoina Paschalidou, Luc van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3, 6, 7

[64] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed A. A. Osman, Dimitrios Tzionas, and Michael J. Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019. 8

[65] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 2

[66] Charles R 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 (NIPS)*, 2017. 1, 2

[67] Edoardo Remelli, Artem Lukoianov, Stephan R. Richter, Benoît Guillard, Timur M. Bagautdinov, Pierre Baqué, and Pascal Fua. Meshsdf: Differentiable iso-surface extraction. 2020. 2

[68] Danilo Jimenez Rezende, S. M. Ali Eslami, Shakir Mohamed, Peter Battaglia, Max Jaderberg, and Nicolas Heess. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 1, 2

[69] Gernot Riegler, Ali Osman Ulusoy, Horst Bischof, and Andreas Geiger. OctNetFusion: Learning depth fusion from data. In *Proc. of the International Conf. on 3D Vision (3DV)*, 2017. 2

[70] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2

[71] Lawrence G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 2

[72] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *ACM Trans. Gr.*, 36(6):245:1–245:17, 2017. 7

[73] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Angjoo Kanazawa, and Hao Li. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[74] Shunsuke Saito, Tomas Simon, Jason M. Saragih, and Hanbyul Joo. Pifuhd: Multi-level pixel-aligned implicit function for high-resolution 3d human digitization. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[75] Gopal Sharma, Rishabh Goyal, Difan Liu, Evangelos Kalogerakis, and Subhransu Maji. Csgnet: Neural shape parser for constructive solid geometry. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[76] Gopal Sharma, Difan Liu, Subhransu Maji, Evangelos Kalogerakis, Siddhartha Chaudhuri, and Radomír Mech. Parsenet: A parametric surface fitting network for 3d point clouds. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020. 3

[77] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1, 2

[78] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3D shapes. *arXiv preprint arXiv:2101.10994*, 2021. 2

[79] M. Tatarchenko, A. Dosovitskiy, and T. Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2

[80] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[81] Yonglong Tian, Andrew Luo, Xingyuan Sun, Kevin Ellis, William T Freeman, Joshua B Tenenbaum, and Jiajun Wu. Learning to infer and execute 3d shape programs. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2019. 3

[82] Shubham Tulsiani, Nilesh Kulkarni, and Abhinav Gupta. Implicit mesh reconstruction from unannotated image collections. *arXiv.org*, 2020. 3

[83] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3

[84] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018. 1, 2, 3

[85] WeiyueXu Wang, Qiangeng Xu, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 2

[86] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 2

[87] Haozhe Xie, Hongxun Yao, Xiaoshuai Sun, Shangchen Zhou, and Shengping Zhang. Pix2vox: Context-aware 3d reconstruction from single and multi-view images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[88] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. DISN: deep implicit surface network for high-quality single-view 3d reconstruction. In *Advances in Neural Information Processing Systems (NIPS)*, 2019. 1, 2

[89] Guandao Yang, Xun Huang, Zekun Hao, Ming-Yu Liu, Serge J. Belongie, and Bharath Hariharan. Pointflow: 3d point cloud generation with continuous normalizing flows. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 1, 2

[90] Xiuming Zhang, Sean Ryan Fanello, Yun-Ta Tsai, Tiancheng Sun, Tianfan Xue, Rohit Pandey, Sergio Orts-Escolano, Philip L. Davidson, Christoph Rhemann, Paul E. Debevec, Jonathan T. Barron, Ravi Ramamoorthi, and William T. Freeman. Neural light transport for relighting and view synthesis. *arXiv.org*, 2008.03806, 2020. 2

[91] Christian Zimmermann, Duygu Ceylan, Jimei Yang, Bryan Russell, Max Argus, and Thomas Brox. Freihand: A dataset for markerless capture of hand pose and shape from single rgb images. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019. 2, 6

[92] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2017. 2, 3