SUPERDEC: 3D Scene Decomposition with Superquadric Primitives

Elisabetta Fedele[1,2] Boyang Sun[1] Leonidas Guibas[2] Marc Pollefeys[1,3] Francis Engelmann[2]

[1]ETH Zurich [2]Stanford University [3]Microsoft

Abstract

We present SUPERDEC*, an approach for creating compact 3D scene representations via decomposition into superquadric primitives. While most recent works leverage geometric primitives to obtain photorealistic 3D scene representations, we propose to leverage them to obtain a compact yet expressive representation. We propose to solve the problem locally on individual objects and leverage the capabilities of instance segmentation methods to scale our solution to full 3D scenes. In doing that, we design a new architecture which efficiently decompose point clouds of arbitrary objects in a compact set of superquadrics. We train our architecture on ShapeNet and we prove its generalization capabilities on object instances extracted from the ScanNet++ dataset as well as on full Replica scenes. Finally, we show how a compact representation based on superquadrics can be useful for a diverse range of downstream applications, including robotic tasks and controllable visual content generation and editing. Our project page is:* https://super-dec.github.io.

1. Introduction

3D scene representations are essential for computer vision and robotics, serving as the foundation for tasks such as 3D scene understanding [33, 47], scene generation [41, 42], functional reasoning [19, 57], and scene interaction [9, 13, 21, 38]. Recent work [18] employs 3D Gaussians as geometric primitives to achieve high-quality photorealistic reconstructions. However, these representations are typically memory-intensive. In contrast, we aim for a lightweight yet geometrically accurate 3D scene representation by decomposing the input point cloud into a compact set of explicit primitives, namely superquadrics (Fig. 1).

Representations for 3D *scenes* include well established formats such as point clouds, meshes, signed distance functions or voxel grids – each offering different trade-off between geometric details, computational cost, resolution, performance, interpretability, and editability. Recently, multi-view approaches like Neural Radiance Fields (NeRF) [29] and Gaussian Splatting (GS) [18] have gained popularity as 3D scene representations. These methods optimize photometric losses to ensure that their implicit (NeRF) or explicit (GS) underlying representations align with observed images. While these representations excel in photorealism, none of them provides explicit control over compactness, often resulting in large, non-modular scene encodings – which are not suitable for tasks requiring explicit spatial reasoning.

Figure 1. **3D Scene Decomposition with Superquadrics.** Given a 3D point cloud of an arbitrary scene, SUPERDEC decomposes all scene objects into a compact set of superquadric primitives.

While optimizing compactness on a scene level remains a challenging task, many approaches have shown that geometric primitives as cuboids [49, 55] or superquadrics [1, 31, 32] enable compact and interpretable decompositions of *individual objects*. Overall, these methods are either learning-based [32, 55], prioritizing speed at the expense of accuracy, or optimization-based [1, 24, 31], achieving better accuracy but incurring higher computation times. While these methods can be accurate for specific object shapes, both approaches struggle to generalize across datasets containing diverse shapes; the former requires category-specific training, while the latter relies on hand-crafted heuristics that limit scalability in unconstrained settings.

Motivated by the abstraction capabilities of geometric primitives for individual objects categories, we propose to represent complex 3D scenes as a compact set of superquadrics.

To this end, we learn general object-level shape priors to optimize compactness and leverage an off-the-shelf 3D instance segmentation method, Mask3D [40], to scale our approach to full 3D scenes.

We choose superquadrics as building block as they offer more accurate shape modeling than cuboids while incurring minimal additional parameter costs (9 vs. 11, including 6-DoF pose parameters). To obtain a model which is able to generalize across different shapes, we draw inspiration from the literature of supervised segmentation and we look at the problem from the perspective of unsupervised geometric-based segmentation, using local point-based features to iteratively refine the predicted geometric primitives. We train our model on ShapeNet [4] and we evaluate it on three challenging and diverse 3D datasets: ShapeNet [4], ScanNet++ [56], and Replica [44]. On the 3D object dataset ShapeNet, our approach achieves a L2 error 6 times smaller compared to prior state-of-the-art work [32] while requiring only half the numbers of primitives. On ScanNet++ and Replica we demonstrate that our approach works well in the real-world scene-level setting, even if trained only on ShapeNet [4]. Finally, we demonstrate the practical use of our method as scene representation for robotic tasks including path planning and object grasping, as well as an editable 3D scene representation for controllable image generation.

In summary, our contributions are the following:

1) We introduce SUPERDEC, a novel method for decomposing 3D scenes using superquadric primitives.

2) SUPERDEC achieves state-of-the-art object decomposition scores on ShapeNet trained jointly on multiple classes.

3) We demonstrate the effectiveness of 3D superquadric scene representations for robotic tasks and controllable generative content creation.

2. Related Work

Learning-based methods have shown that neural networks, when equipped with suitable reconstruction losses, can directly predict geometric primitive parameters to decompose point clouds into a minimal set of primitives for specific object categories. Tulsiani [49] introduced a CNN-based method for cuboid decomposition, which was later extended to more expressive primitives such as superquadrics by Paschalidou *et al*. [32]. CSA [55] further enhanced interpretability by employing a stronger point encoder and jointly predicting cuboid parameters and part segmentations. However, these methods remain constrained by their reliance on category-specific training. We attribute this limitation to their model design, which encodes only global shape features, sufficient for intra-category generalization but ineffective for decomposing out-of-category objects.

Optimization-based methods largely originate from the literature on superquadric fitting. EMS [24] revisited this line of work by introducing a probabilistic formulation that enables the decomposition of arbitrary objects into multiple superquadrics. Given an input point cloud, the method first fits a superquadric to the main structure and identifies unfitted outlier clusters, which are then recursively processed in a hierarchical fashion up to a predefined depth level. However, as noted in their paper and confirmed by our experiments, this approach implicitly assumes that objects exhibit a hierarchical geometric structure, limiting its applicability to many real-world objects such as tables and chairs. Other methods, such as Marching Primitives [25], require Signed Distance Functions (SDFs) as input, which are generally unavailable in real-world scenes. More fundamentally, since these approaches optimize from scratch for each object, they cannot leverage generalizable point features or learned shape priors, both of which are critical for abstraction and robustness under partial observations, a common challenge in practical 3D capture scenarios.

Scene-level decomposition. With the emergence of 3DGS [18], an increasing number of works have explored representing 3D scenes using various geometric primitives [14, 15]. While heuristics can control the number of Gaussians, achieving truly compact representations remains challenging. DBW [31] addresses this by fitting a small set of meshed superquadrics to 3D scenes, building on the principles of 3DGS. Given a set of scene images, it performs test-time optimization with a photoconsistency loss and renders the primitives using a differentiable rasterizer such as SoftRas [22]. To model the environment, DBW adds a meshed ground plane and a meshed icosphere for the background. However, it is restricted to scenes with fewer than 10 primitives and requires objects to be aligned to a ground plane. Furthermore, the optimization is computationally expensive, taking around three hours even for simple DTU [17] scenes. In contrast, our method differs significantly in terms of input requirements, generality of application, and computational efficiency.

Superquadrics are a parametric family of shapes introduced by Barr *et al*. [2] in 1981 and have since been widely adopted in both computer vision and graphics [7, 34, 43]. Their popularity stems from their ability to represent a diverse range of shapes with a highly compact parameterization. A superquadric in its canonical pose is defined by just five parameters: (s_x, s_y, s_z) for the scales along the three principal semi-axes and (ϵ_1, ϵ_2) for the shape-defining exponents. Given those parameters, their surface is described by the implicit equation:

$$f(\mathbf{x}) = \left(\left(\frac{x}{s_x} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{s_y} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{s_z} \right)^{\frac{2}{\epsilon_1}} = 1 \,. \quad (1)$$

Extending this representation to a global coordinate system requires 6 additional parameters (3 for translation and

3 for rotation), resulting in a total of 11 parameters. Another key property of superquadrics is the ability to compute the radial distance from any point in 3D space to the superquadric surface, *i.e.*, the distance between a point and the superquadric's surface along the line connecting that point to the center of the superquadric. Specifically, given a point $\mathbf{x} \in \mathbb{R}^3$, its radial distance to the surface of a canonically oriented superquadric is defined as:

$$d_r = |\mathbf{x}| \cdot |1 - f(\mathbf{x})^{-\epsilon_1/2}| \,, \tag{2}$$

where $f(\mathbf{x})$ is given in Eq. 1. We refer the reader to [51] for the derivation and to [16] for a more comprehensive overview on superquadrics.

3. Method

Our ultimate goal is a 3D scene decomposition using superquadric primitives. To this end, we primarily focus on single-object decomposition and then show how our method, combined with 3D instance segmentation [40], can be applied to full 3D scenes. We detail the single-object approach in Sec. 3.1 and its extension to full scenes in Sec. 3.2.

3.1. Single Object Decomposition

Fig. 2 illustrates our model for single-object decomposition. It consists of two main components: a self-supervised feed-forward neural network that jointly predicts superquadric parameters and a segmentation matrix associating points to superquadrics, followed by a lightweight Levenberg–Marquardt (LM) optimization [20, 28].

3.1.1. Feed-forward Neural Network

Our deep learning model draws inspiration from recent fully-supervised Transformer-based [52] segmentation models [5, 6, 40]. These models iteratively decode a sequence of queries, each representing a segmentation mask, by cross-attending to input pixels or points. In our case, the queries represent superquadrics. Next, we show how such an architecture can be adapted to *unsupervisedly* segment superquadrics, instead of *supervisedly* segment objects.

Model Details. Given an input point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, where each of the N points has a 3D coordinate, we first extract rich point features $\mathcal{F}_{PC} \in \mathbb{R}^{N\times H}$ using the PVCNN [26] point encoder. At the same time, we initialize the superquadrics features $\mathcal{F}_{SQ} \in \mathbb{R}^{P\times H}$ with sinusoidal positional encodings. We feed these features in a Transformer decoder [52] which leverages self-attention, cross-attention and feed-forward layers to refine them.

Once refined, the superquadric features $\mathcal{F}_{SQ}$ and the point features $\mathcal{F}_{PC}$ are fed into two prediction heads: The *segmentation head* takes as input $\mathcal{F}_{SQ}$ and $\mathcal{F}_{PC}$ and predicts a soft assignment matrix $M \in \mathbb{R}^{N\times P}$ associating points to superquadrics and whose elements are defined as:

$$m_{ij} = \sigma\big(\phi(\mathcal{F}_{PC}) \cdot \mathcal{F}_{SQ}\big) \,, \tag{3}$$

where $\phi(\mathcal{F}_{PC}) \in \mathbb{R}^{N\times H}$ is a learned projection of the point features to match the dimensionality of the superquadric features, and σ is the softmax function. The second head, the *superquadric head*, takes the superquadric features $\mathcal{F}_{SQ}$ as input and predicts 12 parameters for each superquadric: 11 encoding its 5-DoF shape and 6-DoF pose, and one modeling its existence probability α, enabling a variable number of superquadrics per object.

Losses. We train our model in a self-supervised manner, without requiring any ground truth annotation. Specifically, the total loss is defined as:

$$\mathcal{L} = \mathcal{L}_{rec} + \lambda_{par}\mathcal{L}_{par} \ + \lambda_{exist}\mathcal{L}_{exist} \,, \tag{4}$$

where $\mathcal{L}_{rec}$ is the reconstruction loss aligning the predicted superquadrics to the input point cloud $\mathcal{P}$, $\mathcal{L}_{par}$ is the parsimony loss encouraging a small number of primitives, $\mathcal{L}_{exist}$ is the existence loss, and λ_{par}, λ_{exist} are weighting coefficients. The reconstruction loss $\mathcal{L}_{rec}$ consists of three terms:

$$\mathcal{L}_{rec} = \mathcal{L}_{\mathcal{P}\to SQ} + \mathcal{L}_{SQ\to\mathcal{P}} + \mathcal{L}_{N} \,. \tag{5}$$

The first two terms correspond to the bi-directional Chamfer distance between the input point cloud and the superquadric surfaces, while the third term serves as a regularizer incorporating normal information to improve convergence during training. To compute the Chamfer distance, we approximate each superquadric surface by uniformly sampling S points, following the method of Pilu *et al.* [35]. Denoting by $d(\mathbf{x}_i, \mathbf{x}'_{js})$ the euclidean distance between the i-th point in the input point cloud and the s-th point sampled on the surface of the j-th superquadric, we define $\mathcal{L}_{\mathcal{P}\to SQ}$ as:

$$\mathcal{L}_{\mathcal{P}\to SQ} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{P} m_{ij} \min_{s\in[S]} d(\mathbf{x}_i, \mathbf{x}'_{js}) \,, \tag{6}$$

and $\mathcal{L}_{SQ\to\mathcal{P}}$ as:

$$\mathcal{L}_{SQ\to\mathcal{P}} = \frac{1}{S\sum_{j=1}^{P}\alpha_j}\sum_{j=1}^{P}\alpha_j\sum_{s=1}^{S}\min_{i\in[N]} d(\mathbf{x}_i, \mathbf{x}'_{js}) \,. \tag{7}$$

The last term of Eq. 5, *i.e.*, $\mathcal{L}_N$ is defined as the reconstruction loss from Yang *et al.* [55], and is used to incorporate normal information during training which leads to accelerated convergence. Additionally, since we seek not only accuracy but also compactness, we introduce a parsimony loss to encourage the use of fewer primitives. To do that, we optimize the 0.5-norm of $m_j \coloneqq \sum_{i=1}^{N}\frac{m_{ij}}{N}$ and define the parsimony loss as:

$$\mathcal{L}_{par} = \left(\frac{1}{P}\sum_{j=1}^{P}\frac{\sqrt{m_j}}{P}\right)^2 \,. \tag{8}$$

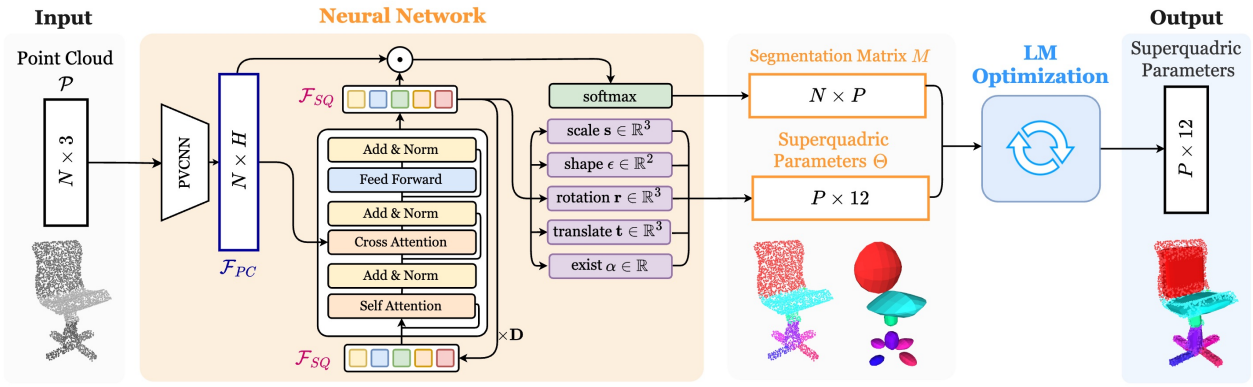

Figure 2. **Illustration of the SUPERDEC Model.** Given a point cloud of an object with N points, a Transformer-based neural network predicts parameters for P superquadrics, as well as a soft segmentation matrix that assigns points to superquadrics. The predicted parameters include the 11 superquadric parameters and an objectness score. These predictions provide an effective initialization for the subsequent Levenberg–Marquardt (LM) optimization, which refines the superquadrics.

Lastly, we employ an existence loss $\mathcal{L}_{exist}$ which uses the predicted segmentation as a teacher for the linear head in charge of predicting the existence probability. More specifically, given a threshold ϵ_{exist}, we define the ground-truth existence of the jth superquadric as $\hat{\alpha}_j := m_j > \epsilon_{exist}$ and define $\mathcal{L}_{exist}$ as:

$$\mathcal{L}_{exist} = \sum_{j=1}^{P} \frac{BCE(\alpha_j, \hat{\alpha}_j)}{P} , \tag{9}$$

where BCE is the binary cross entropy and α_j is the predicted existence probability for the jth superquadric.

3.1.2. Optimization

Our optimization module takes as input the predicted soft segmentation matrix M as well as the superquadric parameters Θ, and further refines the superquadric parameters using the Levenberg-Marquardt (LM) [20, 28] algorithm. Specifically, given a point cloud of N points, it iteratively refines the parameters Θ_j of the jth superquadric, by computing two sets of residuals: The first set of residuals r_{ij} with $i \in [1, N]$ and $j \in [1, P]$ is defined as:

$$r_{ij} = m_{ij} \tilde{d}_j(\mathbf{x}_i) , \tag{10}$$

where $\tilde{d}_j(\mathbf{x}_i)$ denotes the radial distance of point $\mathbf{x}_i$ from the jth superquadric, computed according to Eq. 2. The second set of residuals is used for normalization and is obtained by sampling a set of K points $\mathbf{p_1}, \ldots, \mathbf{p_K}$ on the surface of the given superquadric and then computing the distance of each of them from the point cloud. Specifically, for $i \in [N+1, N+K]$ and $j \in [1, P]$ we compute r_{ij} as:

$$r_{ij} = \min_k ||\mathbf{p_{i-N}} - \Pi_j(\mathbf{x_k})||_2, \quad \text{with } k \in [1, N] . \tag{11}$$

3.2. Decomposition of Full 3D Scenes

After training on single objects, extending SUPERDEC to full 3D scenes is straightforward. Given a scene-level point cloud, we extract 3D object instance masks using Mask3D [40]. Each object is centered and uniformly rescaled to the unit sphere. We then predict the superquadric primitives for each object individually using our model. We found our model trained on ShapeNet [4] to generalize well on real-world 3D scenes from ScanNet++ [56] and Replica [44] without additional fine-tuning.

4. Experiments

We first compare our SUPERDEC with previous state-of-the-art methods on individual objects and full 3D scenes (Sec. 4.1). We then demonstrate the usefulness of our representation on down-stream applications for robotics and controllable image generation (Sec. 4.2). Finally, in Sec. 4.3, we present additional analyses on part segmentation and the implicit learning of shape categories, followed by a study of the compactness–accuracy trade-off and runtime.

4.1. Comparing with State-of-the-art Methods

Datasets. We compare on three different datasets: *ShapeNet* [4]: We use the 13 and train-val-test splits as defined in Choy *et al*. [8]. For each object, we randomly sample 4096 points using Farthest Point Sampling (FPS) [36]. All objects are pre-aligned in a canonical orientation. ShapeNet is a widely used dataset and is therefore well-suited for comparison with existing baselines.
ScanNet++ [56]: We further evaluate our model on real-world object scans from the ScanNet++ validation set. Each object is extracted using ground truth mask annotations, and 4,096 points are sampled per object. In contrast to ShapeNet, these object point clouds are noisier, partially ob-

Model	Primitive Type	Segmentation	*In-category* L1 ↓	*In-category* L2 ↓	*In-category* #Prim.↓	*Out-of-category* L1 ↓	*Out-of-category* L2 ↓	*Out-of-category* #Prim.↓
EMS (Liu *et al.*) [24]	Superquadrics	✗	5.771	1.345	5.68	5.410	1.211	5.68
CSA (Yang *et al.*) [55]	Cuboids	✓	5.157	0.527	9.21	4.897	0.427	11.75
SQ (Paschalidou *et al.*) [32]	Superquadrics	✗	3.668	0.279	10	4.193	0.354	9
SUPERDEC (Ours)	Superquadrics	✓	**1.698**	**0.051**	**5.8**	**1.847**	**0.061**	**5.26**

Table 1. **Quantitative Results on ShapeNet [4].** We show scores for in-category and out-of-category experiments and are scaled by 10^3.

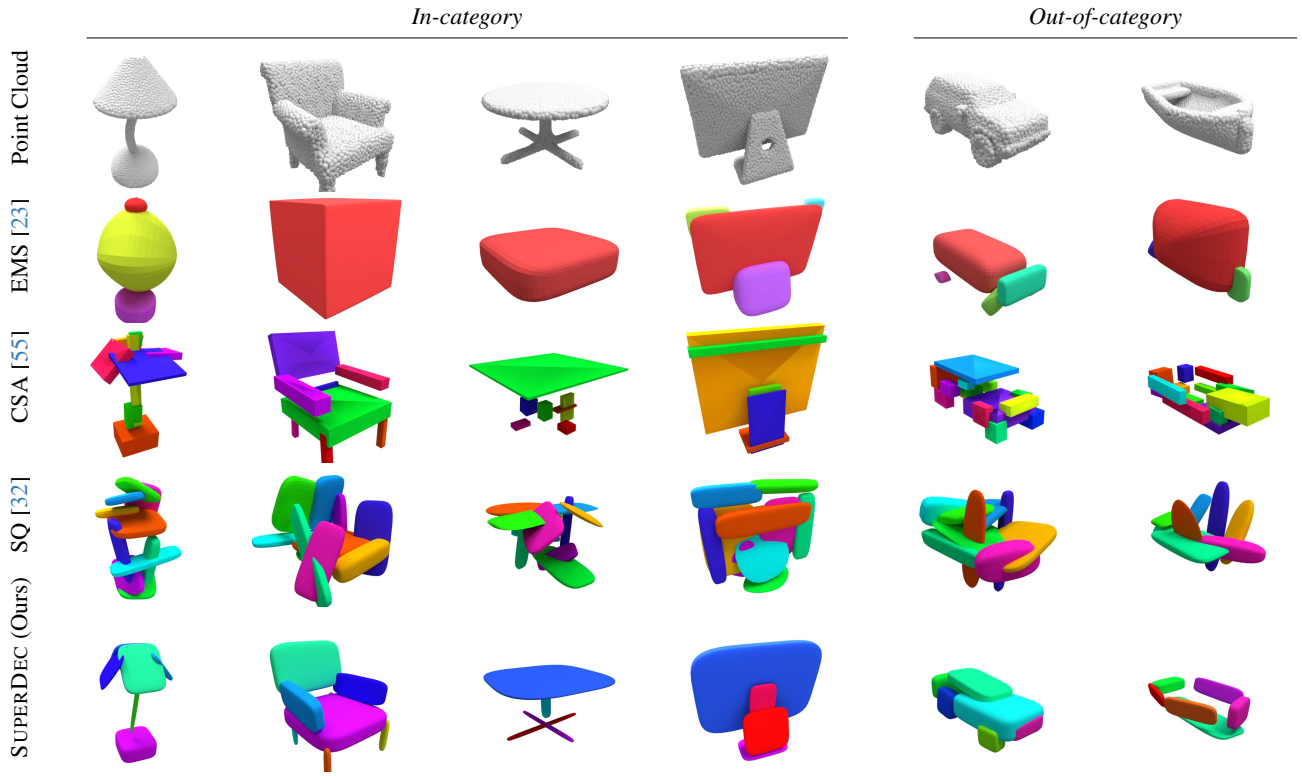

Figure 3. **Qualitative Results on ShapeNet [4].** We show results on test samples for in-category *(four first columns)* classes and out-of-category classes *(two last columns)*. The latter were not seen during training and illustrate how well models generalize to novel classes.

served, and subject to random orientation and translation, providing a more realistic and challenging evaluation setting for our method.

Replica [44]: Lastly, we present qualitative results on full 3D scenes from Replica. Object instances are extracted using the pre-trained 3D instance segmentation model Mask3D [40], allowing us to demonstrate our approach in a fully realistic setting without relying on ground truth annotations.

Methods in comparison. We compare to a wide range of prior works using both cuboids and superquadrics. *SQ* [32] is a learning-based approach for object-level decomposition using superquadrics; it takes a voxel grid as input and predicts superquadric primitives via a CNN. *CSA* [55] is another learning-based method but uses cuboids as geometric primitives. It takes a point cloud as input and predicts cuboid parameters from a global latent code. Lastly, *EMS* [24] is an optimization-based approach that decomposes objects by hierarchically fitting superquadrics to parts of a point cloud until a maximum depth is reached.

Training Details. Our goal is to develop a general-purpose, class-agnostic model capable of representing arbitrary objects as superquadrics. Existing methods typically train separate models for each object class, assuming that all classes are known in advance and that sufficient training data is available for each. These assumptions, however, often fail in real-world scenarios. To address this, we jointly train a single model on all 13 ShapeNet classes using the publicly available code of prior methods, moving towards a more realistic *class-agnostic* solution. In our model we set the following hyper-parameters $P=16$, $S=4096$, $K=25$, $\lambda_{exist}=0.01$, $\lambda_{par}=0.06$.

Model	L1 ↓	L2 ↓	# Prim.↓
EMS (Liu *et al.*) [24]	5.51	2.11	4.25
SUPERDEC (Ours)	1.37	0.07	5.41

Table 2. **Object-Level Evaluation on ScanNet++** [56]

Metrics. We assess *reconstruction accuracy* using L1 and L2 Chamfer distances and *compactness* by the average number of geometric primitives.

4.1.1. Results on ShapeNet

We show scores in Tab. 1 and qualitative results in Fig. 3. To evaluate both accuracy and generalization, we conduct two experiments: *in-category* and *out-of-category*. In the *in-category* setting, all learning-based methods are jointly trained on the 13 classes of the ShapeNet training set and evaluated on the corresponding test set. In the *out-of-category* setting, models are trained on half of the categories (*airplane*, *bench*, *chair*, *lamp*, *rifle*, *table*) and tested on the remaining ones (*car*, *sofa*, *loudspeaker*, *cabinet*, *display*, *telephone*, *watercraft*). Our SUPERDEC model significantly outperforms both learned and non-learned baselines. Compared to learned baselines, we reduce the L2 loss by a factor of six while using nearly half the number of primitives, supporting our hypothesis that leveraging local point features improves 3D decomposition in both accuracy and compactness. Compared to the non-learned baseline, we predict a similar number of primitives but achieve an L2 loss approximately 20 times smaller, validating the benefit of learning shape priors to avoid local minima that often hinder purely optimization-based approaches.

4.1.2. Quantitative Results on ScanNet++ Instances

In this section, models are evaluated on real-world, out-of-category objects, which appear in arbitrary orientations and often exhibit incomplete point clouds due to reconstruction artifacts. Tab. 2 shows the quantitative results. Despite never being trained on real-world objects, our method outperforms the optimization baseline by a large margin, achieving a 30-fold reduction in L2 loss.

4.1.3. Qualitative Results on Full Replica Scenes

Lastly, we qualitatively evaluate our pipeline on full 3D scenes from Replica, where our object-level model is applied on top of class-agnostic instance segmentation predictions from Mask3D [40]. As shown in Fig. 1, our method effectively reconstructs object shapes, even under noisy segmentation masks and geometries that differ substantially from those seen during training.

4.2. Down-stream Applications

Next, we show the versatility of the SUPERDEC representation for downstream applications, including robotics tasks such as path planning and object grasping (Sec. 4.2.1), and controllable image generation (Sec. 4.2.2).

Method	Time (ms)	Suc. (%)	Mem. (MB)
Occupancy	0.056	**100.00**	0.873
PointCloud	0.063	89.57	19.286
Voxels	**0.030**	98.78	0.101
Cuboids [37]	0.120	61.23	**0.024**
SUPERDEC	0.150	91.71	0.042

Table 3. **Path Planning Results.** Values are averaged over 15 ScanNet++ scenes.

Figure 4. **Grasping Result.** Visualization of computed grasp poses for a milk bottle, some flowers, a side table, and a plant.

4.2.1. Robotics

Path planning seeks to compute a collision-free shortest path between a given start and end point in 3D space, enabling efficient robot navigation. Although essential for traversing large environments, it typically demands storing large-scale 3D representations. Here, we assess whether our compact representation can perform this task effectively while reducing memory requirements. We conduct experiments on 15 ScanNet++ [56] scenes, comparing SUPERDEC to common 3D representations, including dense occupancy grids, point clouds, voxel grids, and cuboids [37]. As shown in Tab. 4, SUPERDEC not only reduces memory consumption compared to traditional representations but also achieves a higher success rate than dense point clouds. Further details about experiment setup, metrics and analysis are provided in the Appendix.

Object Grasping enables robots to grasp real-world objects by computing suitable grasping poses. Existing methods fall into two categories, each with complementary limitations. Geometry-based approaches [3, 12, 30] require precise 3D object models, which are often unavailable in real-world scenarios. Learning-based approaches [27, 46, 54] operate directly on raw sensor data but tend to be biased towards training data, which typically consists of tabletop scenes with small, convex, or low-genus objects [11, 45]. To overcome these limitations, SuperQ-GRASP [48] explored decomposing objects into explicit primitives. However, its reliance on Marching Primitives [25] to obtain superquadrics from the object's Signed Distance Function (SDF) makes it unsuitable for most real-world cases

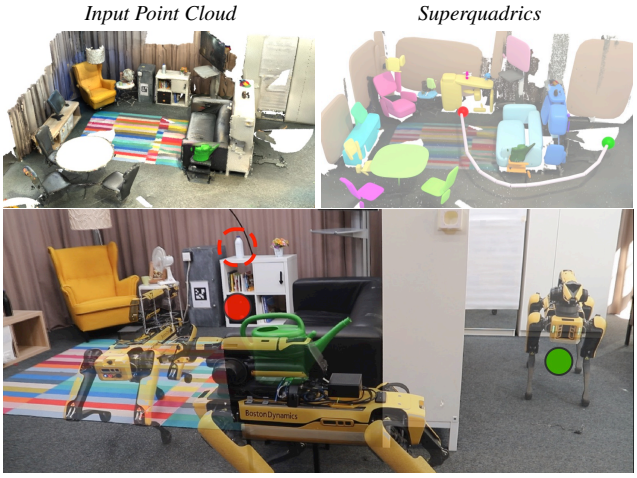

Figure 5. **Real-world robot experiment.** The top row shows the input scan (*left*) and the representation from SUPERDEC with the computed path and grasping pose (*right*). The bottom row illustrates the robot following the planned path. We denote the starting point of the path with a green sphere, and the target location with a red sphere. The target object (a milk bottle) is circled in red.

where only point clouds are available. In contrast, our approach directly processes point clouds of entire scenes and, when combined with the class-agnostic segmentations from Mask3D [40], extracts superquadrics for all objects. Given the superquadric parameters, we employ a superquadric-based geometric method [53] to compute grasping poses for selected objects. Fig. 4 shows predicted grasping poses on objects from a real-world 3D scan of a room. In practice, our method eliminates the need for data-driven grasping models while remaining adaptable to diverse object shapes and producing high-quality grasping poses.

Real-world Experiment. Finally, we demonstrate the real-world applicability of our superquadric-based representation by deploying it on a legged robot (Boston Dynamics Spot) equipped with an arm, supporting both motion planning and object grasping in an indoor environment. We scan a scene using a 3D scanning application on an iPad, extract a dense point cloud, and run SUPERDEC on it. Given the robot's starting position and a specified target object (a milk bottle), we compute both the path and the grasping pose as described earlier (see Fig. 4), enabling the robot to approach and successfully grasp the object. Fig. 5 shows the computed representation, the planned trajectory, the grasping pose, and a frame from the real-world demonstration. This experiment suggests that integrating SUPERDEC with open-vocabulary segmentation methods such as OpenMask3D [47] could allow robots to navigate to and grasp arbitrary objects specified via natural-language prompts.

4.2.2. Controllable Generation and Editing

We investigate how the SUPERDEC representation can be directly used to introduce joint spatial and semantic control in text-to-image diffusion models [39]. Specifically, we generate images by conditioning a ControlNet [58] on depth maps rendered from the superquadrics extracted from Replica [44] scenes. Qualitative results are shown in Fig. 6 and Fig. 7. Fig. 6 demonstrates spatial control: by *moving*, *duplicating*, or *removing* superquadrics corresponding to a plant, we coherently influence the generated images. Fig. 7 highlights semantic control: we can vary the room's style while preserving its semantic and geometric structure, and observe that object semantics naturally emerge from the spatial arrangement of superquadrics without explicit conditioning – *e.g.*, pillows appear on couches, and a plant is placed on a central table, reflecting plausible real-world arrangements.

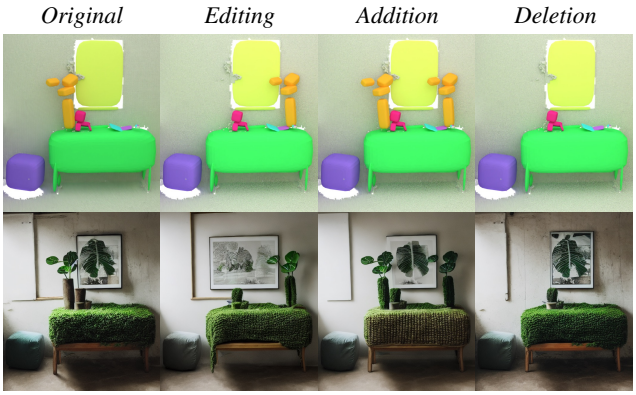

Figure 6. ***Spatial* control using SUPERDEC.** Top row shows superquadrics generated by SUPERDEC, bottom row shows generated images using the prompt *'A corner of a room with a plant'*.

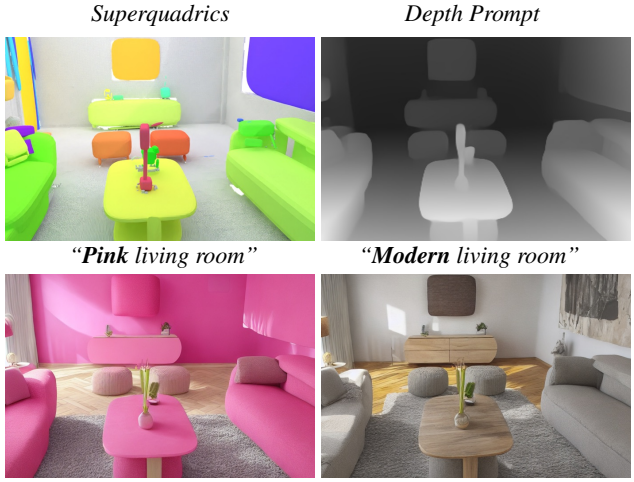

Figure 7. ***Semantic* control using SUPERDEC.** *Top row*: superquadrics created by our SUPERDEC, and depth map to prompt the generation of text-to-image diffusion model. *Bottom row*: generations with two different textual prompts.

4.3. Analysis Experiments

Unsupervised part segmentation. Besides superquadric parameters, our method also predicts a segmentation matrix

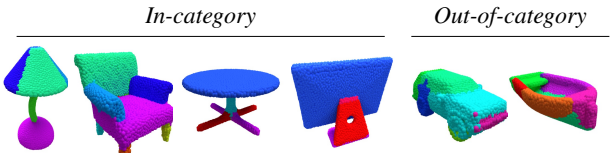

Figure 8. **Qualitative Results on ShapeNet [4] segmentation.** We show the resulting segmentation matrices on test samples for in-category *(four first columns)* classes and out-of-category classes *(two last columns)*. The latter were not seen during training.

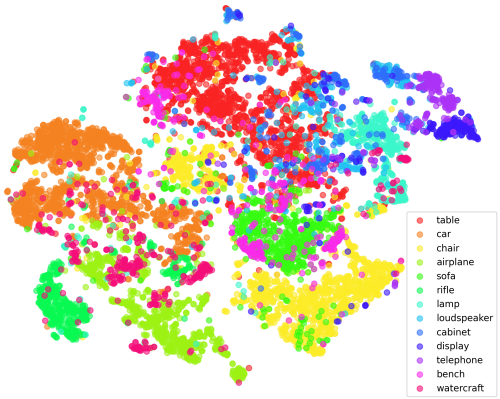

Figure 9. **t-SNE Visualization of Primitive Embeddings** across different ShapeNet classes.

which segments the initial point cloud into parts that are fitted to the predicted superquadrics. In Fig. 8, we visualize the predicted segmentation masks for the same examples shown in Fig. 3. We observe that segmentation masks, especially in the *in-category* experiments, appear very sharp. This suggests that our method, especially if trained at a larger scale, can be leveraged for different applications as geometry-based part segmentation or as pretraining for supervised semantic part segmentation.

What does our network learn? Since our network performs unsupervised part segmentation, we analyze the features learned by the Transformer decoder across object classes. Inspired by BERT [10]'s `[CLS]` token, we append a learnable embedding to the sequence of embedded superquadrics; although never explicitly decoded, this embedding is refined through self- and cross-attention. After training, we extract and visualize these embeddings using t-SNE [50] for ShapeNet [4] categories (Fig. 9). We observe that categories with consistent shapes, such as *chairs*, *airplanes*, and *cars*, form clear clusters, while categories with high intra-class variability, such as *watercraft*, are more dispersed. This indicates that our model organizes objects by geometric structure without requiring class annotations.

How fast is our method? Our model is highly parallelizable, allowing multiple objects to be batched and processed simultaneously in a single forward pass. On an RTX 4090 (24 GB), we can process up to 256 objects in parallel. On average, the forward pass takes 0.13 s for a complete Replica [44] scene, 3D instance segmentation with Mask3D [40] requires 0.3 s, and each LM optimization step takes less than 1 s, see supplementary for more details.

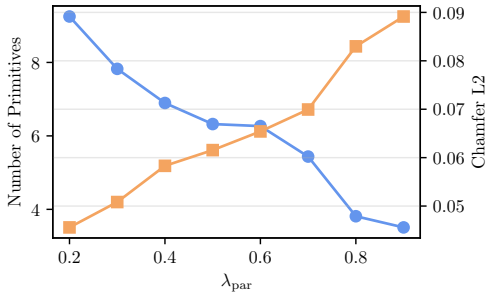

Figure 10. **Compactness vs. reconstruction accuracy tradeoff.** We run experiment for different values of the parsimony weight λ_{par} (x-axis) and we visualize the resulting number of primitives (y-axis, *left*) and the L2 Chamfer distance (y-axis, *right*).

Compactness–Accuracy Trade-off. The hyperparameter λ_{par} controls the trade-off between reconstruction accuracy and representation compactness (see Eq. 4). We evaluate this trade-off quantitatively by first training the model with $\lambda_{par} = 0.1$ for 500 epochs, followed by fine-tuning for 100 epochs with varying λ_{par} values. Fig. 10 shows the impact of λ_{par} on Chamfer distance and the average number of predicted primitives. By adjusting λ_{par}, the model can smoothly balance compactness and accuracy, allowing for easy fine-tuning to meet target reconstruction quality. In our experiments, we use $\lambda_{par} = 0.6$, which approximately corresponds to the intersection point of the two curves.

5. Conclusion

We proposed SUPERDEC, a method for deriving compact yet expressive 3D scene representations based on simple geometric primitives – specifically, superquadrics. Our model outperforms prior primitive-based methods and generalizes well to out-of-category classes. We further demonstrated the potential of the resulting 3D scene representation for various applications in robotics, and as a geometric prompt for diffusion-based image generation. While this is only a first step towards more compact, geometry-aware 3D scene representations, we anticipate broader applications and expect to see further research in this direction.

Acknowledgemnts. Elisabetta Fedele is supported by the ETH AI Center doctoral fellowship, by the Swiss National Science Foundation (SNSF) Advanced Grant 216260 (*Beyond Frozen Worlds: Capturing Functional 3D Digital Twins from the Real World*), and an SNSF Mobility Grant. Francis Engelmann is supported by an SNSF Post-Doc.mobilty grant. We also gratefully acknowledge *GPU donations* from NVIDIA.

References

[1] Stephan Alaniz, Massimiliano Mancini, and Zeynep Akata. Iterative superquadric recomposition of 3d objects from multiple viewsd. In *International Conference on Computer Vision (ICCV)*, 2023. 1

[2] Alan H. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1981. 2

[3] Junhao Cai, Jingcheng Su, Zida Zhou, Hui Cheng, Qifeng Chen, and Michael Yu Wang. Volumetric-based contact point detection for 7-dof grasping. In *Conference on Robot Learning (CoRL)*, 2022. 6

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing 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. Technical report, 2015. 2, 4, 5, 8

[5] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention Mask Transformer for Universal Image Segmentation. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 3

[6] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-Pixel Classification is Not All You Need for Semantic Segmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2021. 3

[7] Laurent Chevalier, Fabrice Jaillet, and Atilla Baskurt. Segmentation and superquadric modeling of 3d objects. In *International Conference in Central Europe on Computer Graphics and Visualization*, 2003. 2

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction. In *European Conference on Computer Vision (ECCV)*, 2016. 4

[9] Alexandros Delitzas, Ayca Takmaz, Federico Tombari, Robert Sumner, Marc Pollefeys, and Francis Engelmann. SceneFun3D: Fine-Grained Functionality and Affordance Understanding in 3D Scenes. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies.*, 2019. 8

[11] Hao-Shu Fang, Chenxi Wang, Hongjie Fang, Minghao Gou, Jirong Liu, Hengxu Yan, Wenhai Liu, Yichen Xie, and Cewu Lu. Anygrasp: Robust and efficient grasp perception in spatial and temporal domains. *IEEE Transactions on Robotics*, 2023. 6

[12] B. Faverjon and J. Ponce. On Computing Two-finger Force-closure Grasps of Curved 2D Objects. In *International Conference on Robotics and Automation (ICRA)*, 1991. 6

[13] Qiao Gu, Ali Kuwajerwala, Sacha Morin, Krishna Murthy Jatavallabhula, Bipasha Sen, Aditya Agarwal, Corban Rivera, William Paul, Kirsty Ellis, Rama Chellappa, et al. Conceptgraphs: Open-vocabulary 3d scene graphs for perception and planning. In *International Conference on Robotics and Automation (ICRA)*, 2024. 1

[14] Abdullah Hamdi, Luke Melas-Kyriazi, Jinjie Mai, Guocheng Qian, Ruoshi Liu, Carl Vondrick, Bernard Ghanem, and Andrea Vedaldi. Ges: Generalized exponential splatting for efficient radiance field rendering. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 2

[15] Jan Held, Renaud Vandeghen, Abdullah Hamdi, Adrien Deli'ege, Anthony Cioppa, Silvio Giancola, Andrea Vedaldi, Bernard Ghanem, and Marc Van Droogenbroeck. 3D convex splatting: Radiance field rendering with 3D smooth convexes. *ArXiv*, 2024. 2

[16] Ales Jaklic, Ales Leonardis, and Franc Solina. *Segmentation and recovery of superquadrics*. Springer Science & Business Media, 2000. 3

[17] Rasmus Ramsbøl Jensen, A. Dahl, George Vogiatzis, Engil Tola, and Henrik Aanæs. Large scale multi-view stereopsis evaluation. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. 2

[18] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkuehler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions On Graphics (TOG)*, 2023. 1, 2

[19] Sebastian Koch, Narunas Vaskevicius, Mirco Colosi, Pedro Hermosilla, and Timo Ropinski. Open3DSG: Open-vocabulary 3D Scene Graphs from Point Clouds with Queryable Objects and Open-set Relationships. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[20] Kenneth Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly of Applied Mathematics*, 1944. 3, 4

[21] Lei Li and Angela Dai. Genzi: Zero-shot 3d human-scene interaction generation. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[22] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. *International Conference on Computer Vision (ICCV)*, 2019. 2

[23] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S. Chirikjian. Robust and accurate superquadric recovery: a probabilistic approach. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5

[24] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Robust and Accurate Superquadric Recovery: A Probabilistic Approach. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 1, 2, 5, 6

[25] Weixiao Liu, Yuwei Wu, Sipu Ruan, and Gregory S Chirikjian. Marching-primitives: Shape abstraction from signed distance function. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 2, 6

[26] Zhijian Liu, Haotian Tang, Yujun Lin, and Song Han. Point-Voxel CNN for Efficient 3D Deep Learning. In *Interna-*

tional Conference on Neural Information Processing Systems (NeurIPS), 2019. 3

[27] Jeffrey Mahler, Matthew Matl, Vishal Satish, Michael Danielczuk, Bill DeRose, Stephen McKinley, and Ken Goldberg. Learning ambidextrous robot grasping policies. *Science Robotics*, 2019. 6

[28] Donald W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of The Society for Industrial and Applied Mathematics*, 1963. 3, 4

[29] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *European Conference on Computer Vision (ECCV)*, 2020. 1

[30] A.T. Miller, S. Knoop, H.I. Christensen, and P.K. Allen. Automatic Grasp Planning Using Shape Primitives. In *International Conference on Robotics and Automation (ICRA)*, 2003. 6

[31] Tom Monnier, Jake Austin, Angjoo Kanazawa, Alexei A. Efros, and Mathieu Aubry. Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 1, 2

[32] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1, 2, 5

[33] Songyou Peng, Kyle Genova, Chiyu Max Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. OpenScene: 3D Scene Understanding with Open Vocabularies. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1

[34] Alex P Pentland. Parts: structured descriptions of shape. In *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, 1986. 2

[35] Maurizio Pilu and Robert B Fisher. Equal-distance sampling of superellipse models. In *British Machine Vision Conference (BMVC)*, 1995. 3

[36] Charles R Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017. 4

[37] Michaël Ramamonjisoa, Sinisa Stekovic, and Vincent Lepetit. MonteBoxFinder: Detecting and Filtering Primitives to Fit a Noisy Point Cloud. In *European Conference on Computer Vision (ECCV)*, 2022. 6, 3

[38] Aaron Ray, Christopher Bradley, Luca Carlone, and Nicholas Roy. Task and motion planning in hierarchical 3d scene graphs. *arXiv preprint arXiv:2403.08094*, 2024. 1

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 7

[40] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. *International Conference on Robotics and Automation (ICRA)*, 2022. 2, 3, 4, 5, 6, 7, 8

[41] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. Controlroom3d: Room generation using semantic proxy rooms. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024. 1

[42] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199*, 2024. 1

[43] Franc Solina and Ruzena Bajcsy. Recovery of parametric models from range images: The case for superquadrics with global deformations. *Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1990. 2

[44] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J. Engel, Raul Mur-Artal, Carl Yuheng Ren, Shobhit Verma, Anton Clarkson, Ming Yan, Brian Budge, Yajie Yan, Xiaqing Pan, June Yon, Yuyang Zou, Kimberly Leon, Nigel Carter, Jesus Briales, Tyler Gillingham, Elias Mueggler, Luis Pesqueira, Manolis Savva, Dhruv Batra, Hauke Malte Strasdat, Renzo De Nardi, Michael Goesele, S. Lovegrove, and Richard A. Newcombe. The replica dataset: A digital replica of indoor spaces. *ArXiv*, 2019. 2, 4, 5, 7, 8

[45] Matan Sudry, Tom Jurgenson, Aviv Tamar, and Erez Karpas. Hierarchical planning for rope manipulation using knot theory and a learned inverse model. In *Conference on Robot Learning*, 2023. 6

[46] Martin Sundermeyer, Arsalan Mousavian, Rudolph Triebel, and Dieter Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *International Conference on Robotics and Automation (ICRA)*, 2021. 6

[47] Ayça Takmaz, Elisabetta Fedele, Robert W. Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. OpenMask3D: Open-Vocabulary 3D Instance Segmentation. In *International Conference on Neural Information Processing Systems (NeurIPS)*, 2023. 1, 7

[48] Xun Tu and Karthik Desingh. Superq-grasp: Superquadrics-based grasp pose estimation on larger objects for mobile-manipulation. *arXiv*, 2024. 6

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

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of machine learning research*, 2008. 8

[51] Erik Roeland van Dop and Paulus P.L. Regtien. Fitting undeformed superquadrics to range data: improving model recovery and classification. *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 1998. 3

[52] Ashish Vaswani, Noam M. Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Inter-*

national Conference on Neural Information Processing Systems (NeurIPS), 2017. 3

[53] Giulia Vezzani, Ugo Pattacini, and Lorenzo Natale. A grasping approach based on superquadric models. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1579–1586. IEEE, 2017. 7

[54] Chenxi Wang, Hao-Shu Fang, Minghao Gou, Hongjie Fang, Jin Gao, and Cewu Lu. Graspness Discovery in Clutters for Fast and Accurate Grasp Detection. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 6

[55] Kaizhi Yang and Xuejin Chen. Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. *ACM Transactions On Graphics (TOG)*, 2021. 1, 2, 3, 5

[56] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. *International Conference on Computer Vision (ICCV)*, 2023. 2, 4, 6, 1, 3

[57] Chenyangguang Zhang, Alexandros Delitzas, Fangjinhua Wang, Ruida Zhang, Xiangyang Ji, Marc Pollefeys, and Francis Engelmann. Open-Vocabulary Functional 3D Scene Graphs for Real-World Indoor Spaces. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025. 1

[58] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *International Conference on Computer Vision (ICCV)*, 2023. 7