SuperFlex: Deformable Superquadrics for Point Cloud Decomposition

Gabriel Tavernini[1*] Elisabetta Fedele[1*] Tiago Novello[2,3]
Leonidas Guibas[2] Marc Pollefeys[1] Francis Engelmann[4]

[1]ETH Zurich [2]Stanford University [3]IMPA [4]USI Lugano

Abstract. Superquadrics have proven to provide a compact, geometrically meaningful representation for 3D objects. However, existing methods suffer from limited reconstruction accuracy, are restricted to rigid primitives, and lack robustness to partial point clouds. In this work, we present SuperFlex, an enhanced framework that expands the expressive power and applicability of superquadric decompositions. First, we introduce a novel loss formulation which significantly improves reconstruction accuracy. Second, we include bending and tapering deformations, enabling high-fidelity representation of curved and asymmetric geometries. Finally, we leverage these high-quality decompositions as supervision to train a model that is robust to partial real-world point clouds. Experiments demonstrate substantial improvements in reconstruction accuracy over both optimization- and learning-based baselines while maintaining a highly compact primitive representation. Project page: https://superflex3d.github.io.

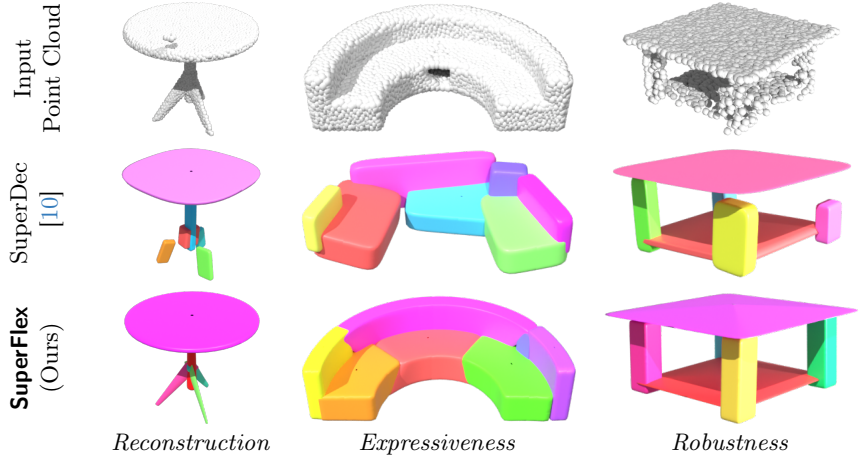

Fig. 1: 3D Object Decomposition with Deformable Superquadrics. Given a 3D point cloud of an object, our SuperFlex decomposes it into geometrically accurate *superquadric* primitives. Our loss significantly improves reconstruction quality *(left)*, while *bending* and *tapering* further increase the expressiveness of superquadrics *(center)*. We also show robustness to real-world partial point clouds *(right)*.

1 Introduction

Obtaining compact, interpretable, and geometrically meaningful representation of 3D objects remains a fundamental challenge in computer vision. While polygon meshes, point clouds, and implicit representations offer high geometric fidelity, they often lack the structural abstraction required for high-level reasoning in robotics and scene understanding. Superquadrics [2] offer a compact alternative: they can represent a wide range of shapes, from ellipsoids to cylinders and boxes, using only a small set of explicit parameters.

Existing superquadric decomposition methods have traditionally been divided into learning- and optimization-based. Learning-based approaches are often limited to specific object categories [24, 32], while optimization-based methods [17, 18] struggle to jointly address part segmentation and primitive fitting, making them sensitive to local minima and hyperparameter tuning. Recently, SuperDec [10] demonstrated the potential of combining these paradigms by using a transformer-based model to segment inputs into parts and predict their corresponding superquadric parameters.

Despite these advances, fundamental limitations remain. First, SuperDec primarily relies on Chamfer-distance supervision, which can lead to gaps between primitives and suboptimal shape fitting. Second, it predicts only *rigid* superquadrics, which are inherently convex and symmetric, limiting their ability to faithfully represent bent and tapered object parts. Finally, it is only trained to unsupervisely decompose complete point clouds, lacking robustness to occlusions and partial data common in real-world scenes. To address these challenges, we introduce SuperFlex, a framework for deformable superquadrics decomposition and robust primitive prediction. SuperFlex as shown in Fig. 1 improves both reconstruction fidelity and robustness to partial observations.

In summary, our contributions are the following:

- **Improved feed-forward reconstruction.** We introduce a novel training loss that significantly improves the reconstruction accuracy of feed-forward superquadric decomposition networks.
- **Deformable superquadrics.** We integrate bending and tapering deformations into a learning-based superquadric decomposition, substantially increasing the expressive power of individual primitives while maintaining a compact parametric representation.
- **Robust feed-forward prediction.** We use our high-quality superquadric decompositions as supervision to train a robust predictor that learns to output complete primitive structure even from partial observations.

We compare SuperFlex against existing methods on objects from ShapeNet, a standard object-level benchmark, and on the Aria Synthetic Environment for scene-level evaluation with partial point clouds. Our results demonstrate that deformable superquadrics provide substantially higher reconstruction accuracy while maintaining a similarly compact representation, and enable robust primitive prediction from partial observations.

2 Related Work

The task of decomposing complex 3D shapes into elementary primitives is a long-standing problem. It requires jointly solving two coupled tasks: (1) segmenting the input into parts that can each be represented by a single primitive, and (2) fitting a primitive to each part. Traditionally, this problem was formulated as a per-instance optimization problem and solved independently for every input shape. More recently, self-supervised learning-based approaches have shown that neural networks can jointly solve these tasks while learning reusable geometric representations across objects, leading to improved decomposition quality and significantly faster inference.

Optimization-based methods originate from the superquadric fitting literature [13]. Recent works have addressed several limitations of classical fitting algorithms. EMS [17] introduced a probabilistic formulation that improves robustness to noise and outliers. However, to obtain multiple primitives, it recursively fits a dominant superquadric and subsequently processes the residual clusters, limiting its ability to represent objects with non-hierarchical structures. Marching Primitives [18] and Light-SQ [31] instead leverage signed distance fields (SDFs) to improve fitting accuracy. While effective, their reliance on accurate SDFs limits their applicability to real-world settings, where only partial point clouds are typically available. More recently, SuperFrusta [11] extended this optimization paradigm to a more expressive primitive representation, enabling accurate reconstruction of a broader range of geometries through differentiable optimization. Despite their strong reconstruction accuracy, optimization-based methods cannot learn reusable shape priors, are susceptible to local minima, and require expensive optimization at inference time.

Learning-based methods predict primitive parameters directly from point clouds [23], images [22], or rendered depth maps [12]. They are typically trained in a self-supervised manner using reconstruction objectives. Early learning-based methods showed that, with appropriate reconstruction losses, neural networks can decompose shapes into a small set of primitives for specific object categories. Tulsiani *et al.* [28] introduced a CNN-based approach for cuboid decomposition, later extended to more expressive primitives such as superquadrics [24]. CSA [32] further improved interpretability by using stronger point encoders and jointly predicting primitive parameters and part segmentations. However, these approaches often rely on category-specific training and global shape features, which limits generalization to out-of-category objects and complex real-world scenes. SuperDec [10] alleviates these issues with a category-agnostic set predictor and sparsity regularization. SuperFlex builds on this line by introducing deformable superquadrics and a novel loss function to significantly improve the reconstruction fidelity.

3 Preliminaries

Superquadrics are a family of shapes introduced by Barr *et al.* [2] and widely adopted in computer vision, graphics, and robotics [9, 13, 20, 22, 24, 26, 30, 34]. In canonical pose, a superquadric is fully described by five parameters: three scale values $\mathbf{s} = (s_x, s_y, s_z)$ and two shape parameters $\boldsymbol{\epsilon} = (\epsilon_1, \epsilon_2)$. The implicit equation of a superquadric, defining its surface, is given by:

$$q_{\text{can}}(\mathbf{x}; \mathbf{s}, \boldsymbol{\epsilon}) := \left(\left(\frac{x}{s_x} \right)^{\frac{2}{\epsilon_2}} + \left(\frac{y}{s_y} \right)^{\frac{2}{\epsilon_2}} \right)^{\frac{\epsilon_2}{\epsilon_1}} + \left(\frac{z}{s_z} \right)^{\frac{2}{\epsilon_1}} = 1 \,. \tag{1}$$

Extending this representation to a global coordinate system requires an additional rigid transformation consisting of translation $\mathbf{t} \in \mathbb{R}^3$ and rotation $R \in SO(3)$, resulting in a total of 11 parameters per superquadric. The implicit representation of the superquadrics then becomes $q_{\text{can}}\left(R^{-1}(\mathbf{x} - \mathbf{t}); \mathbf{s}, \boldsymbol{\epsilon}\right) = 1$. While this formulation provides a compact representation, real-world objects frequently exhibit *tapered* and *bent* geometries. Next, we describe how to handle these cases by extending the canonical model q_{can} with such deformations.

Parametric Deformations. The expressiveness of superquadrics can be further extended by introducing global shape deformations [3,13,27]. A deformation $\mathbf{D}$ explicitly maps surface points $\mathbf{x}$ of the undeformed shape to their deformed counterparts $\mathbf{X}$, i.e., $\mathbf{X} = \mathbf{D}(\mathbf{x})$. In this work, we consider two such deformations, *tapering* $\boldsymbol{\tau}$ and *bending* $\boldsymbol{\beta}$ (see Fig. 2). Tapering progressively narrows or widens a shape along a specified axis, while bending warps a straight line into a circular arc, whose curvature is defined by k. The bending orientation is determined by a specified axis and an angle α. Further details are provided in the Appendix.
In this work, we found single-axis tapering to be sufficient, using $\boldsymbol{\tau} = (\tau_x, \tau_y)$ to control tapering along the x- and y-directions with respect to the z-axis. For bending, we use all three axes, leading to $\boldsymbol{\beta} = (\boldsymbol{\beta}_x, \boldsymbol{\beta}_y, \boldsymbol{\beta}_z)$, which comprises the parameter pairs (k_x, α_x), (k_y, α_y), and (k_z, α_z). In total, the deformation model introduces eight additional parameters: two for tapering and six for bending.
Given a 3D point $\mathbf{x}$, the implicit function of a *deformable superquadric*, parameterized by the 19 coefficients $\Theta = \{\mathbf{t}, R, \mathbf{s}, \boldsymbol{\epsilon}, \boldsymbol{\tau}, \boldsymbol{\beta}\}$, can be evaluated by first

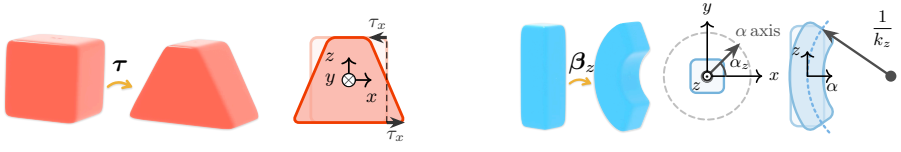

Fig. 2: Illustration of superquadric deformations: *tapering (red)* and *bending (blue)*. Both deformations are shown along a single axis (the z-axis). For tapering along z, the parameters are τ_x (shown) and τ_y (not shown). For bending, we select a bending angle α_z around the z axis, and bend to the curvature defined by k_z.

applying to $\mathbf{x}$ the inverse rigid transformation, followed by inverse bending with respect to the z-, x-, and y-axes and then inverse tapering. We denote this composition of inverse deformation operators by $\mathbf{D}^{-1}_{\boldsymbol{\tau},\boldsymbol{\beta}}$. Finally for all the surface points $\mathbf{x}$ of a deformed superquadric, the following equations holds true:

$$q(\mathbf{x};\Theta) = q_{\text{can}}\Big(\mathbf{D}^{-1}_{\boldsymbol{\tau},\boldsymbol{\beta}}\big(R^{-1}(\mathbf{x}-\mathbf{t})\big);\mathbf{s},\boldsymbol{\epsilon}\Big) = 1. \tag{2}$$

Radial Distance. To fit superquadrics to observed surface points, we need a distance from a point $\mathbf{x}$ to the superquadric surface. The true Euclidean signed distance function (SDF) has no closed-form solution for superquadrics and is expensive to compute during training. Instead, we use the *radial distance*: the distance from $\mathbf{x}$ to where the ray from the superquadric's center through $\mathbf{x}$ hits the surface, which can be computed explicitly as follows [29]:

$$d(\mathbf{x};\Theta) = \|R^{-1}(\mathbf{x}-\mathbf{t})\|\left(1 - q(\mathbf{x};\Theta)^{-\frac{\epsilon_1}{2}}\right). \tag{3}$$

4 Method

We aim to decompose object point clouds into a set of deformable superquadric primitives. To achieve this, we first present a novel reconstruction loss, which we use both to supervise a feed-forward model (Sec. 4.1) and to refine its predictions through test-time optimization (Sec. 4.2). However, real-world point clouds are rarely complete: sensor noise and occlusions typically result in partial observations. To address this, we show how the high-quality decompositions obtained on full point clouds can serve as supervision to train a model that is robust to partial, real-world point clouds (Sec. 4.3).

4.1 Feed-forward Neural Network

Model architecture. Fig. 3 illustrates the SuperFlex model, a Transformer-based architecture closely following the design of SuperDec [10]. Given an input point cloud $\mathcal{P} \in \mathbb{R}^{N\times 3}$, we first extract point features $\mathcal{F}_{\text{PC}} \in \mathbb{R}^{N\times D}$ using a point cloud encoder $\mathcal{E}$ (PVCNN [19]). In parallel, we initialize P superquadric tokens $\mathcal{F}_{\text{SQ}} \in \mathbb{R}^{P\times D}$ using sinusoidal positional encodings. A transformer decoder $\mathcal{D}$ iteratively refines these queries through self-attention among the superquadric tokens $\mathcal{F}_{\text{SQ}}$, cross-attention with the point features $\mathcal{F}_{\text{PC}}$, and feed-forward layers for L times. A *regression head* $\mathcal{Q}$ maps the refined tokens to the parameters Θ of P deformable superquadrics and their existence probabilities $\mathbf{e}$:

$$f(\mathcal{P}) = \mathcal{Q}\big(\mathcal{D}(\mathcal{F}_{\text{SQ}},\mathcal{F}_{\text{PC}})\big) = \{(\Theta_p,\mathbf{e}_p)\}_{p=1}^{P}, \tag{4}$$

where $\Theta_p \in \mathbb{R}^{19}$ denotes the geometric parameters of the p-th deformable superquadric and $\mathbf{e}_p \in [0,1]$ denotes its existence probability.
Finally, a *segmentation head* predicts a soft assignment matrix $W \in \mathbb{R}^{N\times P}$ as $W = \sigma\big(\mathcal{F}_{\text{PC}} \cdot \phi(\mathcal{F}_{\text{SQ}})^\top\big)$, where $\phi(\mathcal{F}_{\text{SQ}}) \in \mathbb{R}^{P\times D}$ is a learned projection of the refined superquadric queries and σ is the softmax which for each point defines a probability distribution over the P superquadrics.

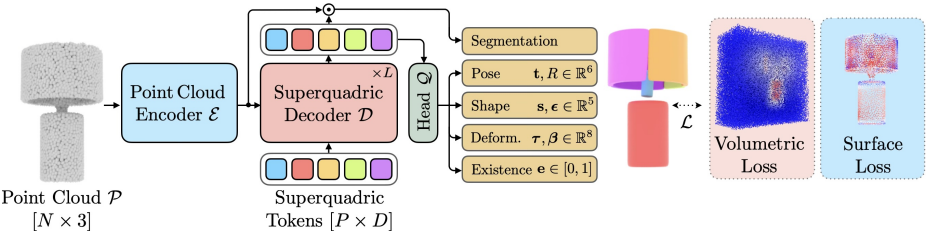

Fig. 3: Illustration of the SuperFlex model. Given an input point cloud, SuperFlex decomposes the object into a set of superquadric primitives, each defined by its pose $(\mathbf{t}, R)$, shape $(\mathbf{s}, \boldsymbol{\epsilon})$, and deformations $(\boldsymbol{\tau}, \boldsymbol{\beta})$. The model is trained via self-supervised joint volumetric and surface losses. A subsequent (optional) object-specific optimization can further improve the superquadric decomposition quality using the same losses.

Loss function. To train the model f, recent methods such as SuperDec [10] rely primarily on the Chamfer distance between the input point cloud and points sampled from the superquadric surfaces. Although effective for coarse alignment, this loss biases the optimization towards overly rounded primitives, limiting the model's ability to capture sharp geometric details.

To address this limitation, we introduce an alternative loss that combines a differentiable Intersection-over-Union loss $\mathcal{L}_{\text{IoU}}$ with a signed distance function (SDF) loss $\mathcal{L}_{\text{SDF}}$. The former encourages the union of superquadrics to better align with the overall shape, resulting in fewer holes in the reconstruction. The latter provides geometry-aware gradients within a narrow band around the surface, enabling the recovery of fine details. We also add the regularization term $\mathcal{L}_{\text{reg}}$, described below, and define the overall training objective as:

$$\mathcal{L} = \lambda_{\text{IoU}}\mathcal{L}_{\text{IoU}} + \lambda_{\text{SDF}}\mathcal{L}_{\text{SDF}} + \mathcal{L}_{\text{reg}}\,. \tag{5}$$

The Volumetric Loss $\mathcal{L}_{IoU}$ encourages the predicted set of superquadrics to accurately reconstruct the target geometry. For a point $\mathbf{x} \in \mathbb{R}^3$, we define the occupancy of each predicted superquadric Θ_p using a soft indicator function [8]:

$$\mathcal{I}_p(\mathbf{x}) = \sigma\left(-\frac{d(\mathbf{x};\Theta_p)}{\tau_{\text{IoU}}}\right), \tag{6}$$

where $d(\mathbf{x};\Theta_p)$ is the radial distance to the p-th superquadric surface (Eq. 3), σ is the sigmoid function, and τ_{IoU} is a temperature parameter. Thus, the global occupancy field associated with the predicted P primitives $f(\mathcal{P})$ is defined by:

$$\hat{\mathcal{O}}(\mathbf{x}) = 1 - \prod_{p=1}^{P}\left(1 - \mathbf{e}_p \cdot \mathcal{I}_p(\mathbf{x})\right). \tag{7}$$

The final $\mathcal{L}_{\text{IoU}}$ loss penalizes the discrepancy between the predicted occupancy $\hat{\mathcal{O}}(\mathbf{x})$ and the ground-truth occupancy $\mathcal{O}(\mathbf{x})$:

$$\mathcal{L}_{\text{IoU}} = -\log\left(\frac{\sum_{\mathbf{x}\in\mathcal{V}} \hat{\mathcal{O}}(\mathbf{x})\,\mathcal{O}(\mathbf{x})}{\sum_{\mathbf{x}\in\mathcal{V}}\left(\hat{\mathcal{O}}(\mathbf{x}) + \mathcal{O}(\mathbf{x}) - \hat{\mathcal{O}}(\mathbf{x})\,\mathcal{O}(\mathbf{x})\right)}\right). \tag{8}$$

where $\mathcal{V}$ is a set of points sampled uniformly in a unit cube containing the object.

The Surface Loss $\mathcal{L}_{SDF}$ additionally encourages local surface alignment by minimizing the distance between the input point cloud $\mathcal{P} = \{\mathbf{x}_i\}_{i=1}^{N}$ and the surfaces of the P predicted superquadrics:

$$\mathcal{L}_{\text{SDF}} = \frac{1}{N}\sum_{i=1}^{N}\sum_{p=1}^{P} w_{ip}\,\psi\Big(d(\mathbf{x}_i;\Theta_p)\Big). \tag{9}$$

Here, w_{ip} is the predicted segmentation assignment weight that controls the influence of primitive p on point $\mathbf{x}_i$. To improve stability and robustness to outliers, we pass the distances $d(\mathbf{x}_i;\Theta_p)$ through an activation function ψ, described in more detail in the Appendix.

Regularization Losses. We further regularize the primitive prediction with

$$\mathcal{L}_{\text{reg}} = \lambda_o\mathcal{L}_{\text{overlap}} + \lambda_s\mathcal{L}_{\text{sparsity}} + \lambda_e\mathcal{L}_{\text{exist}}\,. \tag{10}$$

The overlap loss $\mathcal{L}_{\text{overlap}}$ reduces spatial redundancy by penalizing regions where multiple primitives explain the same volume. Following [8], we aggregate the occupancy indicators $\mathcal{I}_p(\mathbf{x})$ and penalize values above one:

$$\mathcal{L}_{\text{overlap}} = \mathbb{E}_{\mathbf{x}}\left[\text{ReLU}\left(\sum_{p=1}^{P}\mathcal{I}_p(\mathbf{x}) - 1\right)\right]. \tag{11}$$

For compactness and primitive selection, we use the same sparsity and existence regularizers as SuperDec [10]. Specifically, $\mathcal{L}_{\text{sparsity}}$ penalizes the 0.5-norm of the primitive usage $m_p = \frac{1}{N}\sum_{i=1}^{N} w_{ip}$, encouraging parsimonious decompositions. The existence loss $\mathcal{L}_{\text{exist}}$ supervises the predicted existence probability $\mathbf{e}_p$ using binary cross-entropy, with a self-generated target $\hat{\mathbf{e}}_p = \mathbb{I}(m_p > \epsilon_{\mathbf{e}})$ obtained by thresholding the usage metric. This keeps existence predictions consistent with each primitive's actual geometric contribution.

4.2 Test-time Optimization

To further refine individual object reconstructions, we perform additional test-time optimization of the primitive parameters Θ. This refinement uses the same reconstruction loss as at training time, but differs in two aspects: (1) we only optimize primitives predicted as existent, and (2) we no longer rely on the assignment matrix for SDF weighting. We detail these two modifications below.

Active Primitive Selection. To simplify the optimization landscape, we treat the existence probabilities as a binary mask. We select a set of active primitives $\mathcal{M}_{\text{active}}$ by thresholding the network's predictions at $\mathbf{e}_p > 0.5$. During this refinement phase we only optimize the geometric parameters Θ_p of the active set.

Global Distance and Union Representation. Unlike the training phase, where the model relies on a predicted assignment matrix $w_k(\mathbf{x})$ for $\mathcal{L}_{\text{SDF}}$, the test-time optimization treats the primitives as a single global shape. To improve the gradient flow during optimization, we replace the hard minimum with a LogSumExp approximation [8]. To evaluate the volumetric consistency $\mathcal{L}_{\text{IoU}}$ during optimization, we bypass the probabilistic aggregation of individual primitives. Instead, we compute a single global occupancy probability $\hat{\mathcal{O}}(\mathbf{x})$ by directly applying the soft indicator function to the unified distance field. This formulation ensures that the IoU penalty is driven purely by the boundary of the active set $\mathcal{M}_{\text{active}}$, maintaining differentiability while approximating the geometric union of the primitives. The effect of test-time optimization is shown in Tab. 2.

4.3 Robustness to Occlusions

Up to this point, we assumed complete object point clouds. In practice, however, real-world scans are typically noisy and partial due to occlusions. To enable structural completion from partial inputs, we use the decompositions predicted by our test-time-optimized model on complete point clouds (Sec. 4.2) as pseudo-ground truth, and finetune SuperFlex to predict these same decompositions from partially observed versions of the same point clouds.

Training Objective. In a deterministic setting, direct regression of superquadric parameters is ill-posed due to the inherent symmetries of the representation (*e.g.*, axis permutations and sign flips), which produce conflicting gradients during training. To address this, we frame the task as a set-prediction problem. We employ *Hungarian matching* to find the optimal one-to-one assignment between the predicted set of superquadrics $\{\hat{\Theta}_p\}$ and the ground truth set $\{\Theta_p\}$. After matching, we minimize a geometric loss based on the Chamfer Distance (CD) [4] only for the matched pairs where the ground-truth primitive exists:

$$\mathcal{L}_{\text{geom}} = \sum_{p=1}^{M} \text{CD}\left(\mathcal{S}(\hat{\Theta}_p), \mathcal{S}(\Theta_p)\right), \tag{12}$$

where $\mathcal{S}(\Theta)$ denotes a set of points sampled from the surface of superquadric Θ. This formulation makes the loss invariant to parametric permutations and directly emphasizes surface alignment. To regularize training, we also include the volumetric loss $\mathcal{L}_{\text{IoU}}$ (Eq. 8), computed from the full-object occupancy. Furthermore, we introduce an auxiliary geometric loss $\mathcal{L}_{\text{rigid}}$ which is computed as in Eq. 12, but without considering tapering and bending deformations. This acts as a shape prior, encouraging the model to first capture the coarse rigid structure before refining the complex deformations. The final loss is therefore defined as $\mathcal{L}_{\text{sup}} = \mathcal{L}_{\text{geom}} + \lambda_{\text{IoU}}\mathcal{L}_{\text{IoU}} + \mathcal{L}_{\text{rigid}}$.

5 Evaluation

We first compare SuperFlex with previous learning- and optimization-based approaches for single-object decomposition (Sec. 5.1). Then, starting from the predictions of SuperFlex, we evaluate the effect of additional test-time optimization (Sec. 5.2). Finally, we demonstrate the robustness of our finetuned variant of SuperFlex on real-world point clouds (Sec. 5.3).

5.1 Comparison with State-of-the-art Methods

Dataset. We evaluate on the ShapeNet dataset [5] which is widely used in prior primitive-decomposition work, enabling direct comparison with existing baselines. We use the 13 classes and train/val/test splits defined by Choy *et al*. [6], and for each object, we sample 4096 points using Farthest Point Sampling (FPS) [25]. All objects are pre-aligned to a canonical orientation.

Methods in Comparison. We compare SuperFlex against several state-of-the-art primitive-based decomposition methods. Among the *learning-based* approaches, we consider SQ [24], the seminal method for neural superquadric prediction; CSA [32], which represents objects using cuboids with a consistency-aware assignment; and SuperDec [10], a recent feed-forward approach for superquadric decomposition. These methods achieve fast inference but often struggle to accurately capture complex local geometry. We also compare against *aoptimization-based* methods, including EMS [17], which focuses on robust primitive estimation, and Marching Primitives [18], which performs more extensive optimization to achieve high IoU at the expense of substantially higher runtime and a larger number of primitives. We evaluate SuperFlex in two variants: a rigid version using standard superquadrics and the full version, which additionally incorporates tapering and bending deformations.

Training Details. Following SuperDec, all learning-based methods are trained jointly on the 13 ShapeNet categories. We train our network for 1000 epochs starting from SuperDec's ShapeNet checkpoint, with hyperparameters $P = 16$, $N = 4096$, $D = 128$, $L = 3$, $\epsilon_{\mathbf{e}} = 24$, $\tau_{\text{IoU}} = 0.001$, $\lambda_{\text{IoU}} = 0.2$, $\lambda_{\text{SDF}} = 3.2$, $\lambda_o = 5$, $\lambda_s = 1.26$, $\lambda_e = 0.01$. We use a learning rate of 3×10^{-4} for standard model training, and reduce it to 1×10^{-4} for the robust fine-tuning. The model is trained on 4 GPUs on an NVIDIA GH200 GPU Node with a batch size of 32.

Metrics. We compute different metrics to compare *reconstruction accuracy*, *compactness*, and *speed* of different methods. We evaluate reconstruction accuracy using L1 and L2 Chamfer distances, Intersection over Union (IoU), and F-score [16]. We evaluate compactness in terms of average number of active primitives (# Prim.).

Method	Prim. Type	T&B	IoU↑	F [16]↑	L1↓	L2↓	# Prim.↓	Runtime↓
SQ [24]	Superquadrics	✗	0.27	0.13	3.67	0.29	10	0.0065 s
CSA [32]	Cuboids	✗	0.42	0.14	3.52	0.30	7.85	**0.004** s
EMS [17]	Superquadrics	✗	0.40	0.29	4.40	1.06	5.67	7.82 s
March.Prim. [18]	Superquadrics	✗	0.56	0.18	2.08	0.10	24.10	163.41 s
SuperDec [10]	Superquadrics	✗	0.59	0.28	1.77	0.050	5.79	0.0075 s
SuperFlex (ours)	Superquadrics	✗	0.70	0.35	1.75	0.049	6.04	0.0075 s
SuperFlex (ours)	Superquadrics	✓	**0.72**	**0.37**	**1.54**	**0.043**	**5.64**	0.0082 s

Table 1: Quantitative results on ShapeNet. We compare our model to other baselines approaches on ShapeNet. T&B indicates whether tapering and bending deformations are enabled. L1 and L2 scores are multiplied by 10^2.

Quantitative Evaluation. We report quantitative results in Tab. 1. SuperFlex significantly outperforms both learning-based and optimization-based methods, particularly in terms of IoU. The only method that achieves comparable IoU to our base model is Marching Primitives [18]. However, it requires over 4× more primitives and is approximately 10,000× slower. We consider IoU to be the most representative metric for evaluating primitive-based decompositions, as it directly measures how well the union of the predicted primitives matches the occupancy of the ground truth shape.

Qualitative Evaluation. We present qualitative results in Fig. 4. Compared to Marching Primitives [18], SuperFlex achieves comparable reconstruction quality while using substantially fewer primitives. Compared to SuperDec [10], it produces similarly compact decompositions while capturing the underlying geometry much more accurately. In particular, our method better represents rounded surfaces (first three columns) and avoids the large gaps between neighboring primitives often observed in prior work (last three columns).
The results of SuperFlex without tapering and bending demonstrate that our loss function alone substantially improves reconstruction quality while retaining the representational constraints of rigid superquadrics. Enabling tapering and bending further improves the reconstruction of asymmetric shapes and non-convex parts, respectively. Additional qualitative results and comparisons with more baselines are provided in the Appendix.

5.2 Test-time Optimization

To evaluate our refinement method, we perform test-time optimization using Adam [15] for 1000 iterations per object, using a temperature $\tau_{\text{opt}} = 0.01$. Tab. 2 shows that our lightweight refinement stage, while substantially faster than prior optimization-based methods, further improves the predictions of the feed-forward model, yielding a relative improvement of 22%. Fig. 5 qualitatively illustrates how the refinement produces more accurate primitive fits.

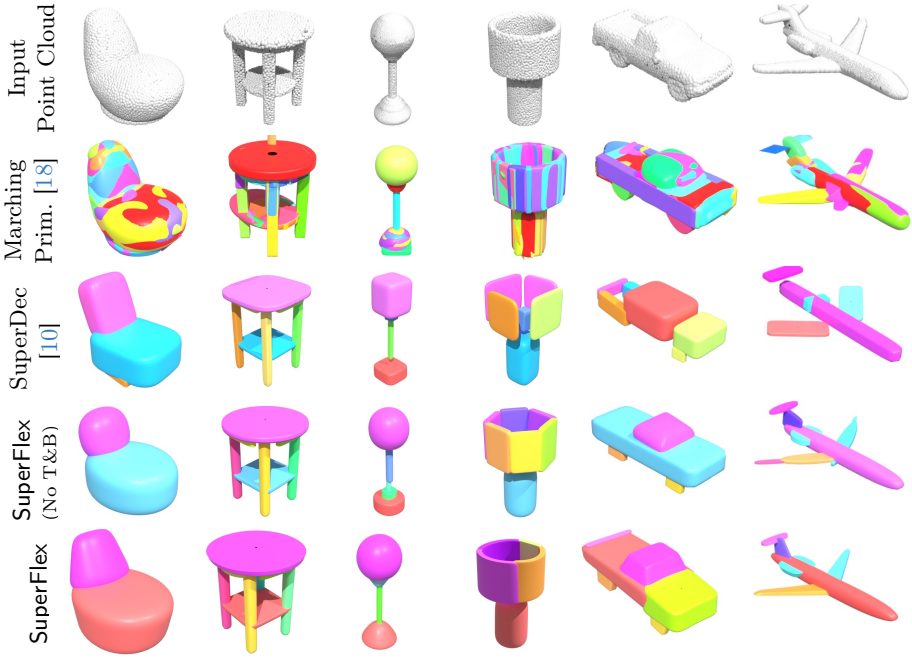

Fig. 4: Qualitative Results on ShapeNet. *Top row:* input point clouds. *Below:* the outputs of baselines and our methods. Different colors indicate different primitives. For SuperFlex, we compare variants with and without tapering and bending (T&B) deformation parameters.

Test-time Opt.	IoU↑	F [16]↑	L1↓	L2↓	# Primitives↓	Runtime↓
✗	0.72	0.37	1.54	0.043	**5.64**	**0.0082** s
✓	**0.87**	**0.49**	**1.32**	**0.036**	**5.64**	5 s

Table 2: Test-time Optimization on ShapeNet. We evaluate the impact of applying test-time optimization to our feed-forward predictions. The first row reports the original feed-forward predictions, while the second row reports the refined predictions after test-time optimization. Chamfer distances (L1 and L2) are multiplied by 10^2.

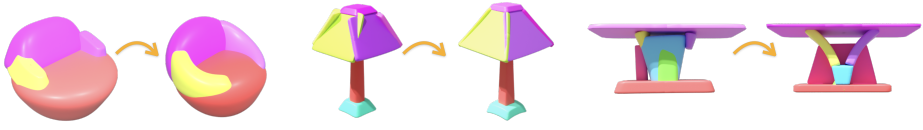

Fig. 5: Qualitative Test-time Optimization Results. SuperFlex refinement fits curved parts with a compact set of flexible superquadric primitives.

Fig. 6: Qualitative Results on ASE. *Top row:* partial input point clouds. *Below:* the outputs of our standard SuperFlex and its robust fine-tuned variant.

5.3 Partial Point Clouds

Datasets. We evaluate reconstruction under realistic sensing conditions using both quantitative and qualitative experiments. For quantitative evaluation, we use the Aria Synthetic Environments (ASE) dataset [21], while qualitative results on real-world scans are shown on ScanNet++ [33]. ASE provides complete ground truth geometry for every object, enabling accurate evaluation across all metrics. The dataset consists of complex multi-room indoor scenes populated with furniture models from the Amazon Berkeley Objects (ABO) dataset [7]. We use the provided depth maps and instance segmentation masks to extract object point clouds exhibiting realistic viewpoint-dependent incompleteness, including self-occlusions, partial visibility, and occlusions caused by surrounding objects. These partial point clouds are provided as input to our model, and predictions are evaluated against the ground truth 3D meshes from ABO.

Training Details. We train a robust variant of SuperFlex to predict deformable superquadrics from partially observed point clouds (see Sec. 4.3). During training, we simulate realistic scanning artifacts by applying *Hidden Point Removal* [14] to model viewpoint-dependent visibility, and random spherical masking to mimic occlusions from foreground objects. To further reduce the domain gap between these synthetic augmentations and real-world occlusions, we perform a brief fine-tuning stage of 100 epochs on partial point clouds from the ASE dataset. This adaptation is especially helpful when large portions of an object are occluded.

Quantitative Evaluation. Results are shown in Tab. 3. Fine-tuning with occlusion augmentations more than doubles the IoU over the unsupervised SuperFlex, showing that pseudo-ground-truth superquadrics provide an effective supervision signal for partial observations. Fine-tuning on ASE further improves performance, highlighting the benefit of adapting to the target domain.

Method	Occ.	ASE	IoU↑	F [16]↑	L1↓	L2↓	# Prim.↓
SQ [24]	✗	✗	0.13	0.10	8.62	3.00	10.0
CSA [32]	✗	✗	0.13	0.11	6.81	2.34	8.66
EMS [17]	✗	✗	0.20	**0.19**	10.96	6.05	4.47
SuperDec [10]	✗	✗	0.20	0.14	5.98	2.45	6.0
SuperFlex	✗	✗	0.20	0.16	5.85	2.09	6.3
SuperFlex	✓	✗	0.48	0.16	4.49	1.62	4.9
SuperFlex	✓	✓	**0.54**	0.18	**3.98**	**1.39**	**4.3**

Table 3: Quantitative results on partial object point clouds from ASE [1]. Occ. denotes fine-tuning with occlusion augmentations on ShapeNet, while ASE denotes fine-tuning on partial object point clouds from ASE.

Fig. 7: Robustness on real-world point clouds. We show results from two ScanNet++ scenes where objects are decomposed into sets of superquadrics by our robust SuperFlex variant. Different colors indicate different object instances. A rendering of the original scene is shown in the top-left corner.

Qualitative Evaluation Figure 6 shows results on objects from the Aria Synthetic Environments (ASE) dataset [21], where partial point clouds are extracted from depth and instance masks with realistic occlusions and missing observations. Despite severe incompleteness, our occlusion-aware model reconstructs complete surfaces, demonstrating that the learned priors capture global object structure. Figure 7 shows results on real-world ScanNet++ scenes, where objects are segmented using ground truth instance masks and decomposed into superquadrics with our method. These results demonstrate robustness in cluttered environments and generalization beyond synthetic data.

5.4 Ablations

Losses. We conduct an ablation study to evaluate the contribution of each term in our loss function. The results, summarized in Table 4, demonstrate the impact of our volumetric and surface objectives on reconstruction quality and decompo-

Method	IoU↑	F [16]↑	L1↓	L2↓	# Prim.↓	Overlap↓
$\mathcal{L}_{\text{SuperDec}}$ [10]	0.59	0.29	1.75	0.050	5.78	4%
$\mathcal{L}_{\text{IoU}}$	**0.74**	**0.40**	1.76	0.124	5.94	32%
$\mathcal{L}_{\text{IoU}} + \mathcal{L}_{\text{SDF}}$	0.72	0.37	1.54	**0.042**	5.80	14%
$\mathcal{L}_{\text{IoU}} + \mathcal{L}_{\text{SDF}} + \mathcal{L}_{\text{overlap}}$	0.72	0.37	**1.54**	0.043	**5.64**	**8%**

Table 4: Loss Ablation Study. We ablate the contribution of each term in our loss function. We report baseline performances training our model with the reconstruction loss from SuperDec (first row) and the improvements obtained by substituting it with our loss terms, which we incrementally introduce (following rows). We keep $\mathcal{L}_{\text{sparsity}}$ and $\mathcal{L}_{\text{exist}}$ fixed in all experiments, as they are essential to the model architecture.

Tapering	Bending	IoU↑	F [16]↑	L1↓	L2↓
✗	✗	0.81	0.44	1.46	0.041
✓	✗	0.83	0.45	1.40	0.038
✗	✓	0.84	0.46	1.39	0.037
✓	✓	**0.85**	**0.47**	**1.35**	**0.036**

Table 5: Ablation Study on Bending and Tapering. We evaluate the impact of tapering and bending deformations. We start from undeformed feed-forward predictions and we evaluate different configurations. Results are computed on ShapeNet, starting from undeformed predictions.

sition parsimony. We will explain the results line by line. Firstly, replacing SuperDec's [10] Chamfer loss with our differentiable $\mathcal{L}_{\text{IoU}}$ improves reconstruction quality, increasing IoU ($0.59 \rightarrow 0.74$) and F-score ($0.29 \rightarrow 0.40$). However, this comes at the cost of increased geometric redundancy, reflected by higher overlap (32%) and L2 error. This behavior arises because the volumetric objective primarily enforces global occupancy consistency, without explicitly constraining surface precision or discouraging multiple primitives from explaining the same region. As a result, thin or detailed structures are penalized less strongly than with point-based distance metrics. Adding $\mathcal{L}_{\text{SDF}}$ introduces surface-level supervision that improves local geometric accuracy, substantially reducing L2 error ($0.124 \rightarrow 0.042$). By aligning primitives with signed distance constraints, the model better captures fine-grained surface structure and exhibits reduced redundancy, lowering overlap from 32% to 14%, while preserving strong volumetric performance. Finally, $\mathcal{L}_{\text{overlap}}$ minimizes spatial redundancy, reducing overlap to 8% and the average primitive count to 5.64 without sacrificing reconstruction fidelity. These results confirm that combining volumetric, surface, and regularization terms is essential for achieving high-fidelity yet parsimonious 3D decompositions.

Deformations To validate the impact of our refinement procedure and the additional expressivity introduced by the extended superquadric parameterization, we perform an ablation study on the ShapeNet test split. Starting from the

undeformed feedforward initializations, we refine the decompositions using different deformation parameterizations. Specifically, we analyze the contribution of tapering and bending during the optimization stage (Table 5). Introducing tapering alone leads to a modest improvement in reconstruction quality; similarly, enabling 3-axis bending results in a further gain across all metrics, indicating that these deformations capture geometric variations not well represented by the base superquadric formulation. When both deformations are enabled, the model achieves its best performance, increasing IoU from 0.81 to 0.85 while consistently reducing reconstruction errors. Although the quantitative gains, particularly for tapering, appear incremental, qualitative inspection (see Fig. 4 and Fig. 5) reveals that these deformations are crucial for capturing primitives with gradually varying cross-sections or curved profiles. These local refinements are often highly effective visually, even if they are not the primary drivers of global metrics such as IoU. Overall, the study confirms that deformation parameters substantially improve the expressivity of superquadric primitives during refinement.

6 Conclusion

We presented SuperFlex, a method for superquadric-based 3D reconstruction that introduces tapering and bending deformations to improve reconstruction fidelity. Our approach combines a novel joint volumetric and surface loss with an efficient refinement stage, enabling accurate primitive decompositions while remaining substantially faster than prior optimization-based methods. Leveraging these high-quality decompositions as pseudo-ground truth, we fine-tune a model variant that predicts complete object structures from noisy and partial real-world observations. Extensive experiments demonstrate that SuperFlex preserves the compactness of parametric representations while overcoming key limitations of prior superquadric-based approaches, making it an effective and scalable representation for 3D scene understanding.

Acknowledgments: This work was supported by a SwissAI Grant for Small Projects and a GPU grant from NVIDIA. Elisabetta Fedele is supported by the ETH AI Center doctoral fellowship and by the Swiss National Science Foundation (SNSF) Advanced Grant 216260 (*Beyond Frozen Worlds: Capturing Functional 3D Digital Twins from the Real World*).

References

1. Avetisyan, A., Xie, C., Howard-Jenkins, H., Yang, T.Y., Aroudj, S., Patra, S., Zhang, F., Frost, D., Holland, L., Orme, C., et al.: SceneScript: Reconstructing Scenes with an Autoregressive Structured Language Model. In: European Conference on Computer Vision (ECCV) (2024) 13
2. Barr, A.H.: Superquadrics and Angle-Preserving Transformations. IEEE Computer Graphics and Applications (1981) 2, 4

3. Barr, A.H.: Global and local deformations of solid primitives. ACM SIGGRAPH Computer Graphics (1984) 4
4. Brazil, G., Kumar, A., Straub, J., Ravi, N., Johnson, J., Gkioxari, G.: Omni3D: A Large Benchmark and Model for 3D Object Detection in the Wild. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 8
5. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., et al.: ShapeNet: An Information-rich 3D Model Repository. arXiv preprint arXiv:1512.03012 (2015) 9
6. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European Conference on Computer Vision (ECCV) (2016) 9
7. Collins, J., Goel, S., Deng, K., Luthra, A., Xu, L., Gundogdu, E., Zhang, X., Vicente, T.F.Y., Dideriksen, T., Arora, H., et al.: ABO: Dataset and Benchmarks for Real-world 3D Object Understanding. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 12
8. Deng, B., Genova, K., Yazdani, S., Bouaziz, S., Hinton, G., Tagliasacchi, A.: CvxNet: Learnable Convex Decomposition. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2020) 6, 7, 8
9. Fedele, E., Engelmann, F., Huang, I., Litany, O., Pollefeys, M., Guibas, L.: SpaceControl: Introducing Test-Time Spatial Control to 3D Generative Modeling. International Conference on Learning Representations (ICLR) (2026) 4
10. Fedele, E., Sun, B., Guibas, L., Pollefeys, M., Engelmann, F.: SuperDec: 3D Scene Decomposition with Superquadric Primitives. In: International Conference on Computer Vision (ICCV) (2025) 1, 2, 3, 5, 6, 7, 9, 10, 11, 13, 14
11. Ganeshan, A., Gadelha, M., Groueix, T., Chen, Z., Chaudhuri, S., Kim, V., Yifan, W., Ritchie, D.: Residual Primitive Fitting of 3D Shapes with SuperFrusta. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2026) 3
12. Genova, K., Cole, F., Vlasic, D., Sarna, A., Freeman, W.T., Funkhouser, T.: Learning shape templates with structured implicit functions. In: International Conference on Computer Vision (ICCV) (2019) 3
13. Jaklic, A., Leonardis, A., Solina, F.: Segmentation and Recovery of Superquadrics, vol. 20. Springer Science & Business Media (2000) 3, 4
14. Katz, S., Tal, A., Basri, R.: Direct visibility of point sets. In: ACM SIGGRAPH 2007 (2007) 12
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization (2017) 10
16. Knapitsch, A., Park, J., Zhou, Q.Y., Koltun, V.: Tanks and temples: benchmarking large-scale scene reconstruction. ACM Trans. Graph. (2017) 9, 10, 11, 13, 14
17. Liu, W., Wu, Y., Ruan, S., Chirikjian, G.S.: Robust and Accurate Superquadric Recovery: A Probabilistic Approach. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2022) 2, 3, 9, 10, 13
18. Liu, W., Wu, Y., Ruan, S., Chirikjian, G.S.: Marching-primitives: Shape abstraction from signed distance function. In: International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 8771–8780 (2023) 2, 3, 9, 10, 11
19. Liu, Z., Tang, H., Lin, Y., Han, S.: Point-voxel cnn for efficient 3d deep learning. International Conference on Neural Information Processing Systems (NeurIPS) (2019) 5
20. Monnier, T., Austin, J., Kanazawa, A., Efros, A., Aubry, M.: Differentiable blocks world: Qualitative 3d decomposition by rendering primitives. International Conference on Neural Information Processing Systems (NeurIPS) (2023) 4

21. Pan, X., Charron, N., Yang, Y., Peters, S., Whelan, T., Kong, C., Parkhi, O., Newcombe, R., Ren, Y.C.: Aria digital twin: A new benchmark dataset for egocentric 3d machine perception. In: International Conference on Computer Vision (ICCV) (2023) 12, 13
22. Paschalidou, D., Gool, L.V., Geiger, A.: Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In: International Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1060–1070 (2020) 3, 4
23. Paschalidou, D., Katharopoulos, A., Geiger, A., Fidler, S.: Neural parts: Learning expressive 3d shape abstractions with invertible neural networks. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2021) 3
24. Paschalidou, D., Ulusoy, A.O., Geiger, A.: Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2019) 2, 3, 4, 9, 10, 13
25. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: International Conference on Neural Information Processing Systems (NeurIPS) (2017) 9
26. Sella, E., Phung, H., Amiel, N., Litany, O., Patashnik, O., Averbuch-Elor, H.: Prox-e: Fine-grained 3d shape editing via primitive-based abstractions. ACM SIGGRAPH 2026 Conference Papers (2026) 4
27. Solina, F., Bajcsy, R.: Recovery of parametric models from range images: The case for superquadrics with global deformations. IEEE transactions on pattern analysis and machine intelligence (1990) 4
28. Tulsiani, S., Su, H., Guibas, L.J., Efros, A.A., Malik, J.: Learning shape abstractions by assembling volumetric primitives. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2017) 3
29. Van Dop, E.R., Regtien, P.P.: Fitting undeformed superquadrics to range data: improving model recovery and classification. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (1998) 5
30. Vezzani, G., Pattacini, U., Natale, L.: A Grasping Approach Based on Superquadric Models. In: International Conference on Robotics and Automation (ICRA) (2017) 4
31. Wang, Y., Chen, W., Hu, Z., Zhang, R., Yin, Y., Wu, R., Luo, K., Qian, S., Ma, Y., Li, H., et al.: Light-SQ: Structure-aware Shape Abstraction with Superquadrics for Generated Meshes. In: ACM SIGGRAPH Asia 2025 Conference Papers (2025) 3
32. Yang, K., Chen, X.: Unsupervised learning for cuboid shape abstraction via joint segmentation from point clouds. ACM Transactions On Graphics (TOG) (2021) 2, 3, 9, 10, 13
33. Yeshwanth, C., Liu, Y.C., Nießner, M., Dai, A.: Scannet++: A high-fidelity dataset of 3d indoor scenes. In: International Conference on Computer Vision and Pattern Recognition (CVPR) (2023) 12
34. Zhao, H., Brunke, L., Lagerquist, O., Zhou, S., Schoellig, A.P.: Sq-cbf: Signed distance functions for numerically stable superquadric-based safety filtering. arXiv preprint arXiv:2602.11049 (2026) 4