EFEM: *E*quivariant Neural *F*ield *E*xpectation *M*aximization for 3D Object Segmentation Without Scene Supervision

Jiahui Lei[1] Congyue Deng[2] Karl Schmeckpeper[1] Leonidas Guibas[2] Kostas Daniilidis[1]
[1] University of Pennsylvania [2] Stanford University
{leijh, karls, kostas}@cis.upenn.edu, {congyue, guibas}@cs.stanford.edu

Abstract

*We introduce Equivariant Neural Field Expectation Maximization (**EFEM**), a simple, effective, and robust geometric algorithm that can segment objects in 3D scenes without annotations or training on scenes. We achieve such unsupervised segmentation by exploiting single object shape priors. We make two novel steps in that direction. First, we introduce equivariant shape representations to this problem to eliminate the complexity induced by the variation in object configuration. Second, we propose a novel EM algorithm that can iteratively refine segmentation masks using the equivariant shape prior. We collect a novel real dataset Chairs and Mugs that contains various object configurations and novel scenes in order to verify the effectiveness and robustness of our method. Experimental results demonstrate that our method achieves consistent and robust performance across different scenes where the (weakly) supervised methods may fail. Code and data available at* https://www.cis.upenn.edu/~leijh/projects/efem

1. Introduction

Learning how to decompose 3D scenes into object instances is a fundamental problem in visual perception systems. Past developments in 3D computer vision have made huge strides on this problem by training neural networks on 3D scene datasets with segmentation masks [55, 63, 67]. However, these works heavily rely on large labeled datasets [3, 15] that require laborious 3D annotation based on special expertise. Few recent papers alleviate this problem by reducing the need to either sparse point labeling [24, 60] or bounding boxes [12].

In this work, we follow an object-centric approach inspired by the Gestalt school of perception that captures an object as a whole shape [32, 47] invariant to its pose and scale [31]. A holistic approach builds up a prior for each object category, that then enables object recognition in different complex scenes with varying configurations. Directly learning object-centric priors instead of analyzing each 3D scene inspires a more efficient way of learning instance segmentation: both a mug on the table and a mug in the dishwasher are mugs, and one does not have to learn to segment out a mug in all possible environmental contexts if we have a unified shape concept for mugs. Such object-centric recognition facilitates a robust scene analysis for autonomous systems in many interactive real-world environments with a diversity of object configurations: Imagine a scenario where a robot is doing the dishes in the kitchen. Dirty bowls are piled in the sink and the robot is cleaning them and placing them into a cabinet. Objects of the same category appear in the scene repeatedly under different configurations (piles, neat lines in the cabinet). What is even more challenging is that even within this one single task (doing dishes) the scene configuration can drastically change when objects are moved. We show that such scenarios cannot be addressed by the state-of-the-art strongly or weakly supervised methods that struggle under such scene configuration variations.

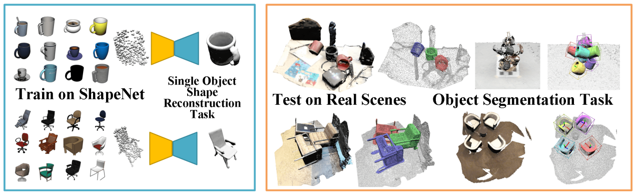

Figure 1. We present EFEM, an unsupervised 3D object segmentation method applicable to real-world scenes (results on the right) by only training on ShapeNet single object reconstruction.

In this paper, we introduce a method that can segment 3D object instances from 3D static scenes by learning priors of single object shapes (ShapeNet [4]) without using any scene-level labels. Two main challenges arise when we remove the scene-level annotation. First, objects in the scene can have a different position, rotation, and scale than the canonical poses where the single object shapes were trained. Second, the shape encoder which is trained on object-level input cannot be directly applied to the scene observations unless the object masks are known. We address the first challenge by introducing equivariance to this problem. By learning a shape prior that is equivariant to the similitude group SIM(3), the composition of a rotation, a translation, and a uniform scaling in 3D (Sec. 3.1), we address the complexity induced by the SIM(3) composition of objects. For the second challenge, we introduce a simple and effective iterative algorithm, Equivariant neural Field Expectation Maximization (**EFEM**), that refines the object segmentation mask, by alternately iterating between mask updating and shape reconstruction (Sec. 3.2). The above two steps enable us to directly exploit the learned single instance shape prior to perform segmentation in real-world scenes. We collected and annotated a novel real-world test set (240 scenes) (Sec. 4.4) that contains diverse object configurations and novel scenes to evaluate the generalizability and robustness to novel object instances and object configuration changes. Experiments on both synthetic data (Sec. 4.3) and our novel real dataset (Sec. 4.4) give us an insight to the effectiveness of the method. Compared to weakly supervised methods, when the testing scene setup is similar to the training setup, our method has a small performance gap to the (weakly) supervised baselines. However, when the testing scenes are drawn from novel object configurations, our method consistently outperforms the (weakly) supervised baselines.

Our paper makes the following novel contributions to the 3D scene segmentation problem: (1) a simple and effective iterative EM algorithm that can segment objects from the scenes using only single object shape priors. (2) addressing the diversity of object composition in 3D scenes by combining representations equivariant to rotation, translation, and scaling of the objects. (3) an unsupervised pipeline for 3D instance segmentation that works in real-world data and can generalize to novel setups. (4) a novel real-world test set **Chairs and Mugs** that contains diverse object configurations and scenes.

2. Related Work

3D Instance segmentation. Point cloud instance segmentation has been a long-existing challenge in 3D vision even before the existence of deep learning. Early interests have been focusing on 3D object retrieval in scenes where RANSAC and generalized Hough voting were the most prominent paradigms [54]. In recent years, with the proliferation of large synthetic [17, 68] or real [3, 15, 22, 25] datasets with rich annotations, learning-based methods have shown great success on this task.

Supervised methods. Most supervised methods fall into two categories: top-down methods that locate objects first and then predict the refined segmentation mask [23, 76], and bottom-up methods learning to group points into object proposals. Regarding the grouping approach, a variety of algorithms have been explored, such as point-pair similarity matrices [64, 80], mean-shift clustering [35], graph-based grouping [20], cluster growing [29], hierarchical aggregation [7, 40], or adversarial methods [78]. Later works [62, 63] combine top-down and bottom-up approaches and achieve impressive results. Recently, transformers and attention mechanisms have also been introduced to this problem [55]. [46] also incorporates neural implicit representations and simultaneously performs segmentation and shape reconstruction. Despite their requirements for laborious data annotation, we will also show that supervised methods heavily rely on the correlation between training and test scenes, and even the state-of-the-art supervised methods struggle to generalize to novel scene configurations or to changes in background patterns.

Unsupervised or weakly supervised methods. Many attempts have been made to learn instance segmentation with limited annotations. A number of works leverage pre-extracted features from scenes via representation learning such as graph attentions [36] or contrastive learning [13, 24, 70, 77] to facilitate weakly-supervised training with fewer point labels. Other approaches directly propagate sparse annotations to dense point labels by learning point affinity graphs [60] or by bounding box voting [12]. These methods are usually the most scalable to large scenes, but their point features are scene-dependent, and we will show that similarly to fully-supervised learning methods, weakly-supervised methods have difficulties in generalizing to scenes with different configurations. When confronting dynamic scenes, one can exploit temporal self-consistency [58] or scene pair constraints [26, 77] but such constraints introduce additional assumptions about the scenes. Most related to our work are the retrieval-based methods leveraging object priors [39, 72]. Traditional retrieval methods are either limited to one given object template [72] or need to solve a discrete combinatorial optimization problem searching for the target template in the object category [39]. We resolve these issues by learning an implicit shape prior which can be optimized continuously in the feature space.

Implicit object priors. First introduced in [10, 45, 49], neural implicit representations (also known as neural fields [71]) parameterize 3D shapes as level sets of neural networks. They not only show strong capabilities of capturing geometric details within limited capacity [28, 50, 57, 59] but also avoid shape discretization that introduces sampling noises. When representing a collection of objects with a shared implicit network, its bottleneck layer naturally forms a latent embedding of the objects, which can serve as a shape prior for many downstream tasks such as shape generation [10, 27, 44], reconstruction [41], completion [11, 45, 49], computing correspondences [33, 37, 43, 56], and part decomposition [9]. We refer the readers to [71] for a comprehensive review. Our method takes advantage of the recent work in neural fields to learn a deep shape prior.

Equivariant point cloud networks. Equivariant networks are designed to preserve transformation coherence between the input and latent representations. With well-developed theories [1, 14, 34, 65, 74], equivariant networks have a variety of designs on pointclouds [2, 16, 19, 30, 51, 61], which benefits many downstream tasks such as robotic applications [18, 21, 52, 56, 66, 75], 3D reconstruction [5, 6], and object pose estimation/canonicalization [38, 42, 48, 53, 81]. Unlike these object-level works, we focus on leveraging equivariance object features in scene understandings. Most related to us is [79] which also applies object equivariance to scenes. But they perform supervised 3D object bounding box detection while we predict dense instance masks. In this work, we employ the vector neurons [2, 6, 16] to build our equivariant shape prior.

3. Method

Now we introduce our method for unsupervised object segmentation in 3D scenes. At the training stage, we learn an object-level shape prior (Sec. 3.1, Fig. 2 top left) utilizing a collection of synthetic models from an object category (e.g. all chairs in ShapeNet [4]). At inference time, we are given a scene point cloud $X_{N\times 6}$ of N points with coordinates as well as normal vectors with an unknown number of novel instances from the object category, and our task is to predict their instance segmentation masks. We will introduce a simple and novel iterative algorithm for predicting instance segmentation masks, starting from a single-object proposal phase (Sec. 3.2, Fig. 2 top right) and followed by a multiple-object joint proposal phase (Sec. 3.3, Fig. 2 bottom). As by-products, our model also outputs implicit surface reconstructions, poses, and bounding boxes.

3.1. SIM(3) Equivariant Shape Priors

Most synthetic datasets have their objects manually aligned to canonical poses and unit scales, yet the SIM(3) transformations (translations, rotations, and scales) must be considered when applying the shape priors to real-world scenes. To this end, we constructed a SIM(3)-equivariant SDF encoder-decoder following the paradigms of prior work [16, 45] (Fig. 2 left top).

Point cloud encoder. Given an object point cloud $P_{N_O\times 3}$ with N_O points, it is first encoded by a SIM(3)-equivariant encoder Φ (yellow block Fig. 2 left) constructed with Vector Neurons (VN) [2, 8, 16] into a latent embedding $\Theta = \Phi(P)$ (orange block Fig. 2 middle). More concretely, the input point cloud P is first subtracted by its centroid $\bar{P}$ for translation equivariance, followed by a backbone network providing a global vector-channeled embedding F, which is scale- and rotation-equivariant and translation invariant. F is then mapped to the shape implict code Θ comprising four components $(\Theta_\mathrm{R}, \Theta_\mathrm{inv}, \Theta_\mathrm{c}, \Theta_\mathrm{s})$. The backbone network is a rotation-equivariant VN Point-Transformer [2] with additional scale equivariance enforced by channel-wise normalizations as in [8]. We denote the modified linear layer with scale invariance [8] as $\widehat{\mathrm{VN}}$. The four components of Θ are computed from F with four heads separately:

1. A vector-channeled rotation equivariant latent code $\Theta_\mathrm{R} = \widehat{\mathrm{VN}}_R(F)$.
2. A scalar-channeled invariant latent code $\Theta_\mathrm{inv} = \langle \widehat{\mathrm{VN}}_I(F), \Theta_\mathrm{R}\rangle$ computed with inner product.
3. A scalar Θ_s by taking the average of F's per-channel norm which explicitly encodes the object scale.
4. A centroid correction vector predicting the offset between the centroid of points and the actual object center, which could be different due to the partiality and noises of the point could: $\Theta_\mathrm{c} = \mathrm{VN}_C(F) + \bar{P}$, where VN_C has 1 output vector channel.

Network architecture details can be found in the supplementary. For any transformation $g = (s, R, t) \in \mathrm{SIM}(3)$ with scale s, rotation R, and translation t, its action on Θ and the equivariance of the encoder Φ can be written as:

$$g \circ \Theta = (\Theta_\mathrm{R} R, \Theta_\mathrm{inv}, s\Theta_\mathrm{c} R + t, s\Theta_\mathrm{s}) = \Phi(sPR + t). \quad (1)$$

SDF decoder. Give a query position $x \in \mathbb{R}^3$, its SDF value $\hat{v}(x)$ is predicted as

$$\hat{v}(x) = \Psi(x; \Theta) = \Psi(\Theta_\mathrm{inv}, \langle \Theta_\mathrm{R}, \tilde{x}\rangle), \quad (2)$$

where $\tilde{x} = (x - \Theta_\mathrm{c})/\Theta_\mathrm{s}$ is the canonicalized coordinate of x with center Θ_c and scale Θ_s, and Ψ is an MLP as in [16] that decodes the concatenation of the invariant feature Θ_inv and the channel-wise inner product between Θ_R and $\tilde{x}$.

Training. The implicit shape prior is trained with the standard L2 loss for the query points sampled around each object. To avoid arbitrary prediction of Θ_c and Θ_s that may make training unstable in early epochs, we regularize Θ_c to stay around zero and Θ_s to stay around one. To help the network's generalization to real-world scenarios with partial observations, clutters, and sensor noises, we further augment the input object point clouds with partial depths and content-wise augmentations. Additional details of these augmentations are provided in the supplementary. We will next introduce how to exploit this learned shape-prior network which only takes instance-level inputs in scene-level point cloud segmentation.

3.2. Iterative Algorithm for Single Proposal

Algorithm 1 Single Proposal Estimation

```
Data: Scene Point Cloud $X_{N\times 6}$; trained encoder $\Phi$ and
      decoder $\Psi$ with frozen weight
Initialize $W_0$
while not reach max step do
    M-step: Sample $P_t$ by Eq. 3 from $W_{t-1}$ and forward
      encoder $\Phi$ to update $\Theta_t$.
    E-step: Evaluate decoder $\Psi$ and update $W_t$ by Eq. 7.
end
Extract mesh and compute absolute pose as a byproduct.
Compute confidence $C$ in Eq. 10 and output mask.
```

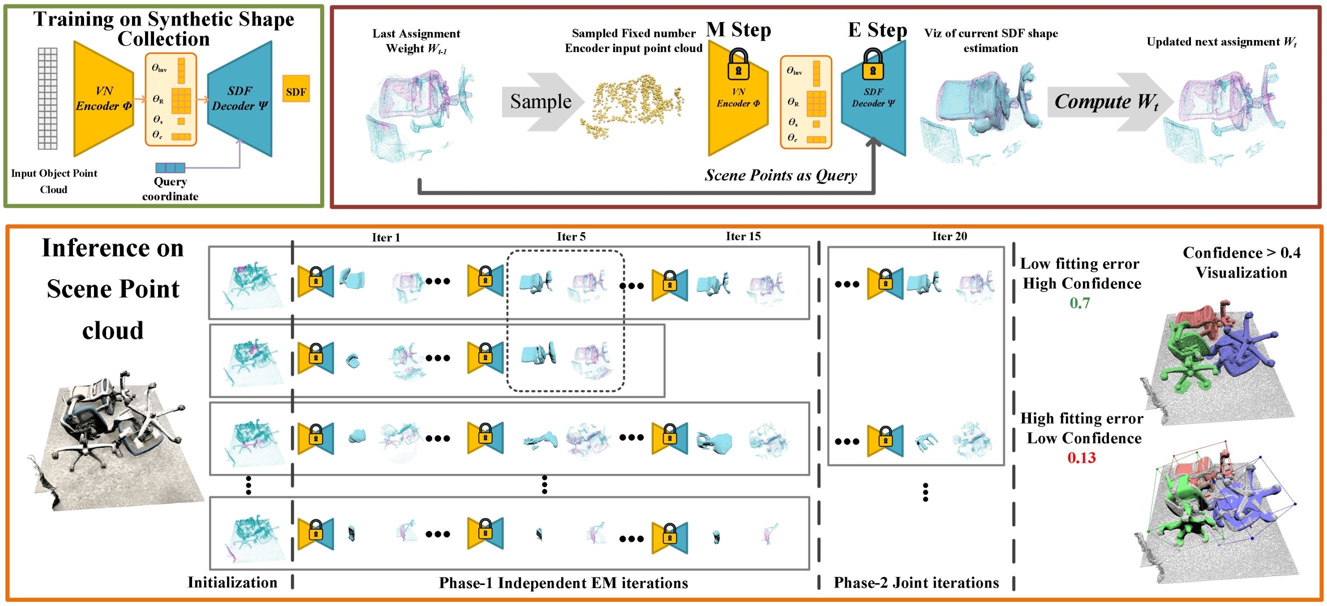
Training on Synthetic Shape Collection
VN Encoder Φ
θ_{inv}
θ_R
θ_s
θ_c
SDF Decoder Ψ
SDF
Input Object Point Cloud
Query coordinate
Last Assignment Weight W_{t-1}
Sample
Sampled Fixed number Encoder input point cloud
M Step
VN Encoder Φ
E Step
SDF Decoder Ψ
Viz of current SDF shape estimation
Compute W_t
Updated next assignment W_t
Scene Points as Query
Inference on Scene Point cloud
Iter 1
Iter 5
Iter 15
Iter 20
Low fitting error High Confidence 0.7
Confidence > 0.4 Visualization
High fitting error Low Confidence 0.13
Initialization
Phase-1 Independent EM iterations
Phase-2 Joint iterations

Figure 2. **Overview**: **Left top**: the single object SDF Encoder-Decoder (Sec. 3.1) is trained on the shape collection. Once trained, the network's weights are frozen.**Top right** Single EM step on scene observations (Sec. 3.2): given the last estimation of the object mask W_{t-1}, a set of points is sampled from the full scene point cloud and then passed to the shape encoder Φ to produce the current estimation of the shape embedding Θ. Based on the Θ, all the scene points are queried again through the decoder Ψ to generate a new assignment mask W_t. **Bottom** Object segmentation pipeline: starting from random crop initializations from the left, the above EM step is applied to each proposal (each row) in parallel to refine their masks. In the early steps (Phase-1 Sec. 3.2), all the proposals run independently but in the second phase, multiple proposals can be jointly optimized (Sec. 3.3). Note that we eliminate the duplicated (the second) or unreasonably sized (the fourth) proposals during optimization. Finally, the remaining proposals and their confidences are output.

As shown in Fig 2, the key component of our full algorithm is the single proposal processing (right top) since the full algorithm is constructed by many single proposals being processed in parallel. Each proposal is represented by a soft assignment mask W over all scene points, where the continuous value on each point $W[i] \in [0, 1]$ indicates how likely this point belongs to the object that the proposal is representing. In the early single proposal iteration steps (Phase-1), each proposal aims to fit its W to one object independently. In later iterations, we optimize the proposals jointly, as discussed in Section 3.3.

Initialization. The single proposal algorithm starts from the initial assignment W_0 which is drawn from a random ball or cylinder cropping of the scene. The radius of the crop is set to be similar to the average size of the target class. All points inside the crop are set to have $W[i] = 1.0$ while all other points are set to have $W[i] = 0.0$.

M-Step: Estimating shape embeddings. We treat the learned decoder with fixed weight as a parametric model of shape categories driven by the parameter Θ. In each iteration of our algorithm, a new Θ_t is estimated from the last assignment W_{t-1}. Note that the learned shape prior encoder Φ only accepts a fixed number N_O of points as input, which is always far less than the number of points in the scene observation. Therefore at each iteration, we first sample a fixed number N_O of points from the scene point cloud $X_{N\times6}$ based on the last assignment estimation W_{t-1}

$$P_t = \mathcal{M}(W_{t-1}, X_{N\times6}), \tag{3}$$

where the sampling operations $\mathcal{M}$ contain two steps. First, a sample is drawn from a Bernoulli distribution with positive probability $W_{t-1}[i]$ to determine if the point $X[i]$ belongs to the object's foreground. Second, for all points that are marked as foreground, we globally apply multinomial sampling based on their $W[i]$ to find N_O sampled points. Finally, we produce the new shape estimation by passing the sampled point cloud through Φ so $\Theta_t = \Phi(P_t)$. This method of updating the shape parameter can be interpreted as the M-step of an EM algorithm, which computes better distribution parameters based on the last assignment via weighted maximum likelihood.

Fitting error. Given the Θ estimation, the fitting error of one observed point $\mathbf{x} = [x_{\text{obs}}, n_{\text{obs}}]$ (x_{obs} denotes position and n_{obs} denotes normal) in the scene point cloud $X_{N\times6}$ is:

$$e_D(\mathbf{x}, \Theta) = |\Psi(x_{\text{obs}}; \Theta)| \tag{4}$$

$$e_N(\mathbf{x}, \Theta) = \text{acos}\left(\frac{n_{\text{obs}}^T \nabla_x \Psi(x_{\text{obs}}; \Theta)}{\|\nabla_x \Psi(x_{\text{obs}}; \Theta)\|}\right) \tag{5}$$

$$E(\mathbf{x}, \Theta) = \alpha_D e_D(\mathbf{x}, \Theta) + \alpha_N e_N(\mathbf{x}, \Theta) \tag{6}$$

which measures the observed point distance to the zero level set of the SDF and the normal consistency between the observed point and the decoded SDF gradient. The hyperparameters α_D and α_N control the importance of the distance and angle terms respectively.

E-Step: Updating point assignments. After updating the shape parameters Θ_t, we update the assignment weight W_t by querying the decoder Ψ for all the points in the scene point cloud X and computing the error in Eq. 6. The new assignment is designed to be updated as:

$$W_t[i] = \frac{e^{-E(X[i],\Theta_t)}}{e^{-E(X[i],\Theta_t)} + \Omega}, \tag{7}$$

where Ω is a constant hyperparameter that gives every point some probability to be in the background. This step can be interpreted as the E-step in an EM algorithm.

Termination and confidence score. When the initialization is not near an object instance of the target class, the shape prior tends to never fit the input observations, so that the error E is large everywhere and the weight W is always small. We terminate the proposals that have less than a predefined threshold of small-error points at each iteration. After the last iteration, we use Marching Cubes to extract a mesh $(\mathcal{V}, \mathcal{E})$ for each proposal and the mesh serves as the byproduct of our method. Our algorithm also produces a pose estimation with respect to the training shape collection via Procrustes registration [81] on Θ_{SO3} between the observed objects and the member of the training set with the most similar Θ (see Supp.). We also remove proposals with meshes outside a predefined reasonable range of scales.

One advantage of utilizing our shape prior is that we can explicitly compute the confidence score from the shape reconstruction. Specifically, we compute two scores by measuring the fitting errors: (1.) **Observation fitting score**: a good fitting should have all the encoder input points located on the decoded SDF zero level set. We measure the proportion of the encoder input point cloud which has a small distance and angle error:

$$S_1 = \frac{1}{N_O} \left| \left\{ \mathbf{x} \in P_T \,\middle|\, \begin{array}{l} e_D(\mathbf{x}, \Theta) < \delta_D, \\ e_N(\mathbf{x}, \Theta) < \delta_N \end{array} \right\} \right|, \tag{8}$$

where the δ_D, δ_N are the thresholds. (2.) **Reconstruction coverage score**: Since we recognize objects by their shapes, we should be less confident in detections where the observed points only cover a small portion of the extracted mesh. For every vertex $\mathbf{v} = [x_{\text{recon}}, n_{\text{recon}}]$ from the extracted mesh, we find its nearest neighbour $\mathbf{x} = [x_{\text{obs}}, n_{\text{obs}}]$ in the observed scene point cloud $X_{N\times 6}$ and measure their distance error as $e'_D(\mathbf{v}) = \|x_{\text{obs}} - x_{\text{recon}}\|_2$ and their orientation error as $e'_N(\mathbf{v}) = \text{acos}(n_{\text{obs}}^T n_{\text{recon}})$. This gives a combined coverage score of:

$$S_2 = \frac{1}{|\mathcal{V}|} \left| \{ \mathbf{v} \in \mathcal{V} | e'_D(\mathbf{v}) < \delta_D, e'_N(\mathbf{v}) < \delta_N \} \right|. \tag{9}$$

We use S_1 as the main confidence measurement and S_2 for avoiding poorly observed cases, so the final confidence score C is:

$$C = S_1 * \max(1.0, S_2/\delta_C), \tag{10}$$

where $\delta_C \in [0.0, 1.0]$ is a threshold controlling the importance of the coverage score. Note how the above confidence values evaluate the quality of the output shown on the right in Fig. 2, which enables the user to select the balance between recall and precision during inference.

Finally, for each point in the scene point cloud $X_{N\times 6}$, we check whether the distance error in Eq. 4 and Eq. 5 is smaller than the output threshold and mark points with small errors as in the foreground. Note that since the observation is noisy and the learned shape prior is not perfect, the output error thresholds can be larger than the ones used in Eq. 8 and Eq.9.

3.3. Multiple Proposals

Since the EM algorithm outputs can be affected by its initialization and we do not know the number of objects in the scene, we initialize a large number of proposals randomly spread across the entire scene to cover all possible objects. We observe that many proposals will quickly converge to similar positions during the early iterations. Therefore, at each iteration, we remove duplicated proposals and only keep the one with the highest fitting score S_1 defined in Eq. 8. Duplication is determined by computing the overlap of W between different proposals (see Supp. for details).

As shown in Fig. 2, we further divide the iterations into two phases. In the first phase, since the early shape estimation is not converged to a reasonable place, we let each proposal run fully independently. When we enter the second phase, mesh extraction and pose estimation will be first applied to remove proposals with unreasonable scales. Then, we optimize all proposals globally by updating the joint assignment weight with:

$$W_t^{(k)}[i] = \frac{S_1^{(k)} e^{-E(X[i],\Theta_t^{(k)})}}{\sum_j S_1^{(j)} e^{-E(X[i],\Theta_t^{(j)})} + \Omega}, \tag{11}$$

where k is the index of current active proposals and $S_1^{(k)}$ is the current fitting score in Eq. 8, which increases the assignment weight for more confident proposals. During the last iterations in Phase-2, we also remove the proposals that are largely contained by other proposals to simplify the decomposition of the scene, following a similar methodology to duplication removal. Additional details of this process are available in the supplemental.

4. Experiments

We focus our experiments on answering four main questions: First, can our method successfully segment the objects of interest from the scene? Second, how does our

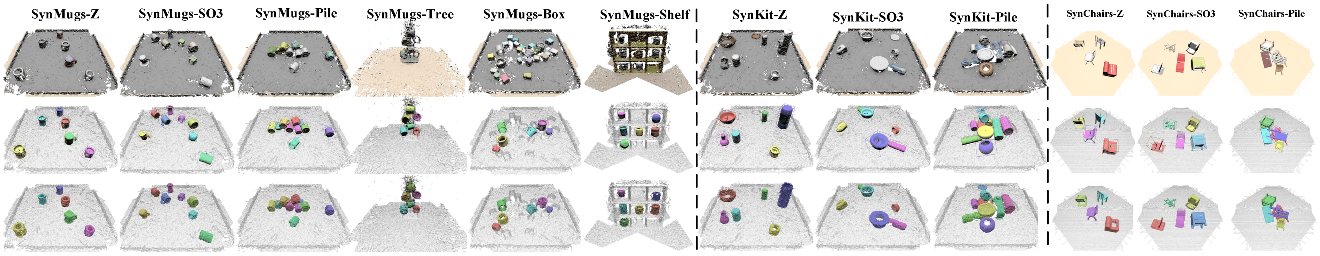

Figure 3. **Synthetic scenes: Top**: the mesh reconstruction as input, note how our data has a realistic simulated depth pattern; **Middle**: visualization of the estimated shapes, poses and bounding boxes as the byproducts of our method. **Bottom**: Our segmentation prediction.

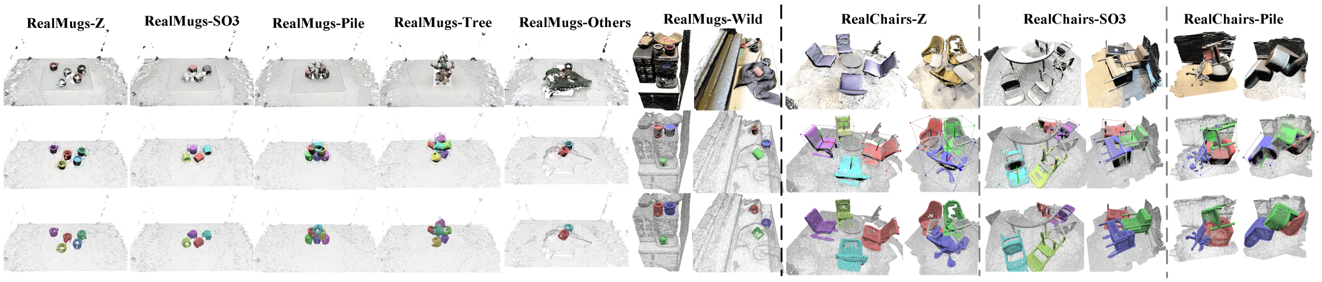

Figure 4. **Chairs and Mugs** real testset: the three rows are in the same format as Fig. 3.

method compare to existing baselines when different training data distributions are accessible? Third, how does each approach generalize to the different testing environments, including the real-world scenes and the out-of-distribution configurations? And fourth, how do the different components of our model contribute to its performance? To answer these questions, we perform a series of baseline comparisons as well as ablation studies both on synthetic and real-world scenes.

4.1. Baselines

To the best of the authors' knowledge, fully unsupervised instance segmentation (not foreground-background segmentation) in static scenes remains unexplored in the deep learning literature. Therefore, we compare our method with supervised and weakly supervised methods. SoftGroup (CVPR22) [63] is the current SoTA for 3D instance segmentation methods that are trained with ground truth instance masks. Box2Mask (ECCV22) [12] is a recent weakly supervised method, which only needs ground truth instance bounding boxes as supervision while retaining competitive performance. The closest weakly supervised method to ours is ContrastiveSceneContext (CVPR21) [24], which can be trained with only a few point labels. To conduct fair comparisons and reduce the sim2real gap, all baselines are fully trained on the corresponding training set with positions and normals as input while not using colors.

4.2. Experimental Setup

We focus our experiments on scenes that contain objects humans or robots can interact with. These objects are more interesting targets for segmentation as their configurations can change dramatically in real-world applications, such as robotics or AR/VR. We experiment with three object categories that frequently appear in the presence of clutter and in diverse configurations in real-world data: **Mugs**, **Kitchen containers**, and **Chairs**. Since there is no available dataset that contains these objects with rich configuration changes (not just sitting upright on a flat surface), and all baselines need to be trained with ground truth, we gather simulated data to train the baselines, then evaluate on simulated and real test scenes containing completely unseen object instances.

Three types of scenes are used for training baselines on each of the object classes. Take the mugs scene as an example (Fig. 3 Left top): In the **Z** scenes, all instances are upright and not in contact with each other. In **SO(3)**, the objects can have a random orientation but are still not in contact with each other. **Pile** is a much more challenging setting where the objects can touch each other, can take any orientation, and can form piles. We simulate 500 scenes for training, 50 scenes for validation and 100 scenes for testing in each of the setups. For each baseline we train three models, one on the data from each of the three scene types.

We train our equivariant shape prior (Sec. 3.1) on the corresponding categories from ShapeNet [4] and freeze their weights once trained. Note that all the following results are generated from the same trained shape prior for each category, and all the objects in our testing scenes never appear in the training set of the shape prior. We evaluate each model on all of the scenes and report results in Tab. 1, Tab. 2 and Tab. 3. The gray cells indicate results from models that were trained on less difficult datasets than they were evaluated on, with the underlined number indicating the best

Mugs	Testing	Z		SO3		Pile		Tree		Box		Shelf		(R) Z		(R) SO3		(R) Pile		(R) Tree		(R) Others		(R) Wild	
Training	Metrics	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50
Scene Z	CSC (100)	6.0	22.0	1.9	7.6	0.1	0.5	0.0	0.0	0.0	0.2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.2	0.9
	CSC (200)	78.7	98.1	62.7	83.3	8.9	17.4	0.6	2.0	2.0	5.0	0.8	3.1	72.1	99.8	9.5	21.4	0.3	1.6	0.6	2.8	6.7	16.0	2.0	6.1
	Box2Mask	96.9	**100**	92.1	99.3	36.7	27.6	12.3	45.3	10.0	24.0	8.6	14.6	98.1	**100**	93.1	**100**	5.6	19.1	8.4	28.8	18.4	39.5	18.1	27.9
	SoftGroup	**100**	**100**	96.5	98.5	21.9	27.6	0.4	1.8	1.6	3.7	9.2	17.2	93.7	97.6	97.3	**100**	0.0	0.0	0.9	3.4	4.3	9.1	18.3	32.9
Scene SO3	CSC (100)	1.7	8.6	1.4	6.5	0.1	0.6	0.0	0.0	0.0	0.1	0.1	0.6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.9
	CSC (200)	70.0	99.3	72.7	95.7	22.3	42.2	10.2	24.6	5.4	12.3	5.2	20.0	68.4	**100**	59.8	**100**	1.0	3.7	4.8	20.1	6.9	19.7	3.9	11.8
	Box2Mask	95.8	**100**	94.7	**100**	53.7	80.1	30.2	80.6	26.5	40.1	13.4	33.0	**100**	**100**	95.3	**100**	12.7	34.3	19.6	49.2	27.1	50.6	50.3	74.8
	SoftGroup	**100**	**100**	99.6	99.8	43.9	51.9	10.0	18.7	15.3	21.3	20.8	30.9	99.2	**100**	98.2	**100**	2.8	4.8	10.4	21.7	6.5	13.3	18.3	30.7
Scene Pile	CSC (100)	0.0	0.3	0.0	0.3	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7	1.8
	CSC (200)	53.3	88.4	50.2	77.7	29.5	61.3	29.4	66.2	7.6	19.8	10.3	34.1	13.0	45.0	22.7	65.2	5.3	30.1	1.5	8.0	3.5	15.4	2.8	8.5
	Box2Mask	96.0	**100**	94.7	**100**	78.7	**99.6**	55.2	94.8	42.7	52.9	26.3	54.6	**100**	**100**	95.8	**100**	72.3	**98.0**	55.8	**98.1**	56.0	79.2	45.1	71.4
	SoftGroup	99.5	**100**	**99.7**	**100**	**89.0**	93.2	42.3	72.8	23.4	25.5	28.8	39.5	99.6	**100**	**99.7**	**100**	**74.7**	86.3	53.4	83.0	50.9	66.5	48.1	65.9
ShapeNet	EFEM	78.4	99.8	79.3	99.8	68.2	96.8	**68.8**	**99.0**	**59.9**	**77.0**	**48.7**	**72.4**	87.2	**100**	87.3	**100**	63.1	94.0	**69.6**	95.4	**69.7**	**89.4**	**54.1**	**82.3**

Table 1. Results (%) on SynMugs (left) and RealMugs (right with label R). A50 corresponds to mAP50. The full table including mAP25 is available in the supplemental. The grey cells highlight that the testing scenes are out of the training distribution and the bold number means the best among all methods while the underlined number means the best within grey cells.

	Testing	(C) Z		(C) SO3		(C) Pile		(K) Z		(K) SO3		(K) Pile	
Training	Metrics	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50	AP	A50
Scene Z	CSC (100)	66.2	83.1	56.6	80.2	6.1	15.2	12.7	29.9	7.5	14.5	2.3	5.9
	CSC (200)	77.7	91.0	63.7	85.1	10.5	22.7	63.0	80.9	37.9	49.2	12.2	23.2
	Box2Mask	**99.9**	**100**	95.8	98.9	19.3	41.2	89.7	97.3	69.6	87.0	41.4	68.9
	SoftGroup	99.8	**100**	94.8	98.5	24.1	36.5	93.6	96.8	94.0	99.3	52.3	63.5
Scene SO3	CSC (100)	74.9	81.4	77.3	82.4	17.7	31.0	8.5	21.7	11.3	23.8	2.0	5.4
	CSC (200)	84.3	86.6	85.4	88.5	20.6	35.3	27.8	48.9	53.9	77.5	18.1	38.9
	Box2Mask	99.7	**100**	99.1	99.5	50.5	80.3	85.3	96.2	91.8	98.5	52.9	76.5
	SoftGroup	99.6	**100**	99.1	**100**	55.1	68.2	89.3	93.9	**96.9**	**99.3**	56.2	67.2
Scene Pile	CSC (100)	67.0	74.1	74.6	82.1	47.2	63.8	3.4	11.2	2.6	7.1	1.4	4.5
	CSC (200)	71.7	76.3	88.4	92.4	67.4	81.6	26.5	53.8	44.8	73.6	28.8	60.4
	Box2Mask	99.6	99.7	**99.5**	99.7	78.6	96.9	87.2	**97.9**	92.1	98.6	74.8	**94.6**
	SoftGroup	99.4	**100**	98.9	**100**	**95.0**	**97.0**	**93.7**	97.3	96.0	98.8	**84.6**	91.3
ShapeNet	EFEM	93.1	99.2	86.1	97.4	75.3	88.0	69.4	83.4	67.6	83.1	60.1	78.9

Table 2. SynChairs (left) and SynKit (right), same format as Tab. 1

RealChairs	Testing	R(Z)			R(SO3)			R(Pile)		
Training	Metrics	AP	AP50	AP25	AP	AP50	AP25	AP	AP50	AP25
Scene Z	CSC (100)	0.5	1.4	9.4	1.3	2.8	10.8	0.0	0.0	7.2
	CSC (200)	0.8	1.2	11.0	0.3	0.6	9.8	0.0	0.2	3.1
	Box2Mask	0.0	0.3	20.2	0.3	1.9	27.2	0.0	0.1	11.8
	SoftGroup	5.1	7.7	27.0	2.8	6.7	21.9	0.0	0.0	3.7
Scene SO3	CSC (100)	3.6	5.3	18.2	1.4	2.5	13.3	0.0	0.1	4.5
	CSC (200)	1.2	2.6	10.4	1.8	4.9	20.2	0.1	0.5	9.1
	Box2Mask	1.5	5.9	48.4	5.1	19.9	56.7	0.4	1.8	28.2
	SoftGroup	13.8	17.2	27.4	7.9	16.2	30.3	0.1	0.1	6.1
Scene Pile	CSC (100)	5.2	7.1	9.6	3.2	7.4	12.3	0.1	0.3	2.9
	CSC (200)	8.3	11.7	23.1	13.9	23.2	33.6	1.0	2.6	13.1
	Box2Mask	1.6	6.5	37.8	9.4	22.0	65.3	1.3	4.8	46.6
	SoftGroup	20.6	27.8	36.1	14.4	24.4	42.1	0.4	1.5	8.0
Scannet	CSC (100)	36.8	49.8	62.0	5.3	13.1	18.8	3.9	10.8	18.1
	CSC (200)	50.0	74.3	77.8	6.2	13.1	18.6	5.5	13.4	27.1
	Box2Mask	**84.1**	**99.4**	**99.4**	16.2	33.7	41.2	14.3	29.8	50.7
	SoftGroup	74.0	81.1	87.5	22.3	39.6	48.6	11.5	18.0	36.1
ShapeNet	EFEM	51.0	78.3	85.0	**51.2**	**75.7**	**87.9**	**34.4**	**55.3**	**74.8**

Table 3. Results on RealChairs, same format as Tab. 1

performing inside gray cells. The metric for evaluating the segmentation is the commonly used mAP [15].

4.3. Results on synthetic data

We first evaluate the performance on simulated data, and we found that existing baselines work very well inside the training distribution if trained with enough supervisory signals. However, the weakly supervised method CSC [24] has a significant performance drop when the number of supervision points decreases from 200 to 100. This performance drop becomes increasingly severe when the training set distribution becomes more complex, demonstrating the difficulty of unsupervised segmentation. With zero scene-level supervision, our method performs well with a small gap in performance to the (weakly) supervised methods.

When the baselines are trained on **Z** but tested on **SO3**, the baselines do not show a large drop in performance, potentially because both **Z** and **SO3** scenes have no objects in contact with each other. However, when tested on **Pile**, we see that baselines trained on **Z** or **SO3** perform differently, and are both worse than the ones trained on **Pile**, indicating a failure to generalize to clutter. Our method outperforms all the baselines that are not trained on **Pile** configurations.

We additionally generate 50 scenes each from three new and more difficult scene setups for testing on **Mugs** as shown in Fig. 3. In the **Tree** scenes mugs are hanging on a holder tree and distributed vertically. The **Box** scenes include cubes to simulate objects that have not been seen during training. In the **Shelf** scenes, the mugs are put on a shelf that is only visible from one side. In all three of these scene setups, our method is able to outperform all baselines as the baselines are unable to generalize to configurations of mugs that are significantly outside of the training distribution.

4.4. Results on real data

We additionally evaluate the performance of our model on real data. To the best of the authors' knowledge, there is no existing real dataset that contains interactable objects in diverse configurations for 3D instance segmentation. Therefore we collect a test set **Chairs and Mugs** that contains the reconstruction of 240 real scenes with object instance mask annotations to test our method in the real world.

RealMugs: As shown in Fig. 4 we replicate the **Z, SO3, Pile** and **Tree** setups in the real world. The scene is captured by 4 calibrated realsense D455 cameras mounted on the corners of the table. We further introduce two new setups that cover hard-to-simulate scenes. The **Others** setup contains random objects that a manipulator may encounter, including cloth, toys, paper bags, wires and tools. These objects are added into the tabletop scene, contacting and occluding the mugs. The **Wild** setup contains crops of real-world indoor scans from labs, kitchens, and teaching buildings. Unlike [73], our scenes are not restricted to only showing the

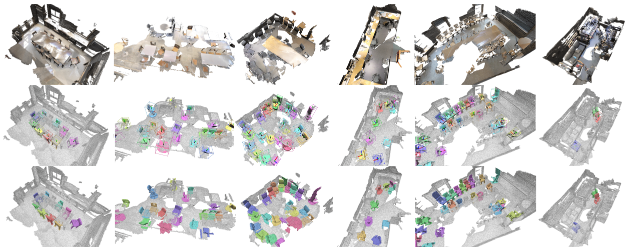

Figure 5. ScanNet qualitative results, the same format as Fig. 3

.

table and upright mugs, but include diverse configurations, backgrounds, and distractors. These scans are captured by an iPad with a lidar scanner. Since the **Z, SO3** and **Pile** setups are easy to simulate realistically, we only collect 10 scenes per setup. We collect 50 scenes for each of the **Tree**, **Others** and **Wild** setups.

We test all the baselines trained on the synthetic dataset and our method trained on ShapeNet directly on these real scenes. Quantitative results are shown in Tab. 1. Since our simulator includes advanced techniques of active light simulation in the physical engine [69] and we do not use the colours as input, the sim2real gap for the baseline methods is minimal in these controlled setups. We can draw similar conclusions for **Z**, **SO3**, **Pile** and **Tree** setups as with the synthetic experiments. Our method retains reasonable performance on **Others** scenes due to its awareness of the object shape. When tested on the **Wild** real-world scenes, our method also performs the best, which demonstrates the strong generalizability of our method.

RealChairs: We also collect data of chairs in the real world following the **Z**, **SO3** and **Pile** setups. Although we have plenty of real-world scan datasets like [15], none of them include diverse configuration changes of chairs. Therefore we collected and annotated a small test set with 20 scenes per setup as shown in Fig. 4. We train the baselines in simulation and test them in the real world. Additionally, we take all the baselines' official model weights from being trained on the real world ScanNet dataset [15] to evaluate on our test set. The results are shown in Tab. 3. There is a larger sim2real gap for the baselines with the chairs data than with the mugs data because of less realistic depth simulation and difficulties aligning the scale between Shapenet chairs and real-world chairs. In contrast, the baselines trained on ScanNet work very well on the **Z** setup, which aligns best with the ScanNet dataset. However, they have a significant drop in performance when the testing distribution shifts to the **SO3** and **Pile** setups. In contrast, our method retains reasonable performance on real world chairs across different setups. We will release our Chairs and Mugs dataset to provide more opportunities to study robustness, generalizability and equivariance for scene and object understanding in the real world.

We also show our effectiveness on ScanNet [15], where the indoor scene scan can span the whole room. Qualitative results are shown in Fig. 5 and the AP metrics for the chairs category of our method on the validation/test set are $AP = 24.6/20.2$, $AP50 = 50.8/39.0$ and $AP25 = 61.3/48.3$ where in comparison the weakly supervised method CSC [24] trained on 200 points labels achieves $AP50 = 62.9/61.1$ (See Suppl. for a table). One main reason for our performance drop on ScanNet is that ScanNet has many partially observed chairs, which are hard to be recognized via shape. We leave future explorations to fill this gap between our unsupervised method and the (weakly) supervised ones.

4.5. Ablations

Ablation	AP	AP50	AP25
Full	**0.601**	**0.789**	**0.807**
No Phase 2	0.548	0.740	0.756
No Normal	0.203	0.302	0.337

Table 4. Ablations

We verify the effectiveness of our design by removing the phase-2 joint iterations and removing the usage of the normals. When not using phase-2 (Sec. 3.3), we let the phase-1 independent iterations run more steps to keep the total number of iterations constant. When removing the normals, all error computing, assignment weight updating, and confidence scoring will only take the distance error term in to account while ignoring the normal term. We compare our full model with the ablated models on SynKit **Pile** setups. The quantitative results are shown in Tab. 4, which illustrates that both components contribute to our model's overall performance. More ablation studies can be found in our supplementary.

5. Conclusions

We present EFEM, a method for 3D instance segmentation that is trained only on shape datasets. Without requiring any real or simulated scene data, our method can generalize to complex, real world scenes better than existing methods that also require more supervision.

Limitations and future work. Although we show an encouraging step towards unsupervised 3D instance segmentation by generalizing knowledge directly from a shape collection to a scene, several weaknesses remain. First, our method has a performance drop when the object is significantly occluded. This is due to our method recognizing objects only via their shape, rather than also reasoning about color or other features. Second our current running speed is slow (∼1min per scene) due to the large number of proposals the initialization requires.

Acknowledgements. The authors appreciate the support of the following grants: NSF FRR 2220868, NSF IIS-RI 2212433, NSF TRIPODS 1934960, ARL DCIST CRA W911NF-17-2-0181, ARO MURI W911NF-20-1-0080, ONR N00014-22-1-2677 awarded to UPenn; Toyota University 2.0 program and Vannevar Bush Faculty Fellowship awarded to Stanford University.

References

[1] Jimmy Aronsson. Homogeneous vector bundles and g-equivariant convolutional neural networks. *Sampling Theory, Signal Processing, and Data Analysis*, 20(2):1–35, 2022. 3

[2] Serge Assaad, Carlton Downey, Rami Al-Rfou, Nigamaa Nayakanti, and Ben Sapp. Vn-transformer: Rotation-equivariant attention for vector neurons. *arXiv preprint arXiv:2206.04176*, 2022. 3

[3] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *International Conference on 3D Vision (3DV)*, 2017. 1, 2

[4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 3, 6

[5] Evangelos Chatzipantazis, Stefanos Pertigkiozoglou, Edgar Dobriban, and Kostas Daniilidis. Se (3)-equivariant attention networks for shape reconstruction in function space. *arXiv preprint arXiv:2204.02394*, 2022. 3

[6] Haiwei Chen, Shichen Liu, Weikai Chen, Hao Li, and Randall Hill. Equivariant point network for 3d point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14514–14523, 2021. 3

[7] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021. 2

[8] Yunlu Chen, Basura Fernando, Hakan Bilen, Matthias Nießner, and Efstratios Gavves. 3d equivariant graph implicit functions. *ECCV*, 2022. 3

[9] Zhiqin Chen, Kangxue Yin, Matthew Fisher, Siddhartha Chaudhuri, and Hao Zhang. Bae-net: Branched autoencoder for shape co-segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8490–8499, 2019. 2

[10] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 2

[11] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6981, 2020. 2

[12] Julian Chibane, Francis Engelmann, Tuan Anh Tran, and Gerard Pons-Moll. Box2mask: Weakly supervised 3d semantic instance segmentation using bounding boxes. In *European Conference on Computer Vision*, pages 681–699. Springer, 2022. 1, 2, 6

[13] Ruihang Chu, Xiaoqing Ye, Zhengzhe Liu, Xiao Tan, Xiaojuan Qi, Chi-Wing Fu, and Jiaya Jia. Twist: Two-way inter-label self-training for semi-supervised 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1100–1109, 2022. 2

[14] Taco Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. *arXiv preprint arXiv:1811.02017*, 2018. 3

[15] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5828–5839, 2017. 1, 2, 7, 8

[16] Congyue Deng, Or Litany, Yueqi Duan, Adrien Poulenard, Andrea Tagliasacchi, and Leonidas J Guibas. Vector neurons: A general framework for so (3)-equivariant networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12200–12209, 2021. 3

[17] Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3d-front: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10933–10942, 2021. 2

[18] Jiahui Fu, Yilun Du, Kurran Singh, Joshua B Tenenbaum, and John J Leonard. Robust change detection based on neural descriptor fields. *arXiv preprint arXiv:2208.01014*, 2022. 3

[19] Fabian Fuchs, Daniel Worrall, Volker Fischer, and Max Welling. Se (3)-transformers: 3d roto-translation equivariant attention networks. *Advances in Neural Information Processing Systems*, 33:1970–1981, 2020. 3

[20] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2940–2949, 2020. 2

[21] Carolina Higuera, Siyuan Dong, Byron Boots, and Mustafa Mukadam. Neural contact fields: Tracking extrinsic contact with tactile sensing. *arXiv preprint arXiv:2210.09297*, 2022. 3

[22] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2019. 2

[23] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4421–4430, 2019. 2

[24] Ji Hou, Benjamin Graham, Matthias Nießner, and Saining Xie. Exploring data-efficient 3d scene understanding with contrastive scene contexts. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15587–15597, 2021. 1, 2, 6, 7, 8

[25] Binh-Son Hua, Quang-Hieu Pham, Duc Thanh Nguyen, Minh-Khoi Tran, Lap-Fai Yu, and Sai-Kit Yeung. Scenenn: A scene meshes dataset with annotations. In *2016 fourth international conference on 3D vision (3DV)*, pages 92–101. Ieee, 2016. 2

[26] Jiahui Huang, He Wang, Tolga Birdal, Minhyuk Sung, Federica Arrigoni, Shi-Min Hu, and Leonidas J Guibas. Multibodysync: Multi-body segmentation and motion estimation via 3d scan synchronization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern*

Recognition, pages 7108–7118, 2021. 2

[27] Moritz Ibing, Isaak Lim, and Leif Kobbelt. 3d shape generation with grid-based implicit functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13559–13568, 2021. 2

[28] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 2

[29] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and Pattern recognition*, pages 4867–4876, 2020. 2

[30] Oren Katzir, Dani Lischinski, and Daniel Cohen-Or. Shape-pose disentanglement using se (3)-equivariant vector neurons. *arXiv preprint arXiv:2204.01159*, 2022. 3

[31] David G Kendall. A survey of the statistical theory of shape. *Statistical Science*, 4(2):87–99, 1989. 1

[32] Kurt Koffka. *Principles of Gestalt Psychology*. Lund Humphries, 1935. 1

[33] Amit Pal Singh Kohli, Vincent Sitzmann, and Gordon Wetzstein. Semantic implicit neural scene representations with semi-supervised training. In *2020 International Conference on 3D Vision (3DV)*, pages 423–433. IEEE, 2020. 2

[34] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In *International Conference on Machine Learning*, pages 2747–2755. PMLR, 2018. 3

[35] Jean Lahoud, Bernard Ghanem, Marc Pollefeys, and Martin R Oswald. 3d instance segmentation via multi-task metric learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9256–9266, 2019. 2

[36] Min Seok Lee, Seok Woo Yang, and Sung Won Han. Gaia: Graphical information gain based attention network for weakly supervised point cloud semantic segmentation. *arXiv preprint arXiv:2210.01558*, 2022. 2

[37] Jiahui Lei and Kostas Daniilidis. Cadex: Learning canonical deformation coordinate space for dynamic surface representation via neural homeomorphism. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6624–6634, 2022. 2

[38] Xiaolong Li, Yijia Weng, Li Yi, Leonidas Guibas, A Lynn Abbott, Shuran Song, and He Wang. Leveraging se(3) equivariance for self-supervised category-level object pose estimation. *Thirty-Fifth Conference on Neural Information Processing Systems*, 2021. 3

[39] Yangyan Li, Angela Dai, Leonidas Guibas, and Matthias Nießner. Database-assisted object retrieval for real-time 3d reconstruction. In *Computer graphics forum*, volume 34, pages 435–446. Wiley Online Library, 2015. 2

[40] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021. 2

[41] Chen-Hsuan Lin, Chaoyang Wang, and Simon Lucey. Sdf-srn: Learning signed distance 3d object reconstruction from static images. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 2

[42] Cheng-Wei Lin, Tung-I Chen, Hsin-Ying Lee, Wen-Chin Chen, and Winston H Hsu. Coarse-to-fine point cloud registration with se (3)-equivariant representations. *arXiv preprint arXiv:2210.02045*, 2022. 3

[43] Feng Liu and Xiaoming Liu. Learning implicit functions for topology-varying dense 3d shape correspondence. *Advances in Neural Information Processing Systems*, 33:4823–4834, 2020. 2

[44] Zhengzhe Liu, Yi Wang, Xiaojuan Qi, and Chi-Wing Fu. Towards implicit text-guided 3d shape generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17896–17906, 2022. 2

[45] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 2, 3

[46] Yinyu Nie, Ji Hou, Xiaoguang Han, and Matthias Nießner. Rfd-net: Point scene understanding by semantic instance reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4608–4618, 2021. 2

[47] Stephen E. Palmer. *Vision science: Photons to phenomenology*. The MIT Press, 1999. 1

[48] Haoran Pan, Jun Zhou, Yuanpeng Liu, Xuequan Lu, Weiming Wang, Xuefeng Yan, and Mingqiang Wei. So (3)-pose: So (3)-equivariance learning for 6d object pose estimation. *arXiv preprint arXiv:2208.08338*, 2022. 3

[49] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 2

[50] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 523–540. Springer, 2020. 2

[51] Adrien Poulenard and Leonidas J Guibas. A functional approach to rotation equivariant non-linearities for tensor field networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13174–13183, 2021. 3

[52] Hyunwoo Ryu, Jeong-Hoon Lee, Hong-in Lee, and Jongeun Choi. Equivariant descriptor fields: Se (3)-equivariant energy-based models for end-to-end visual robotic manipulation learning. *arXiv preprint arXiv:2206.08321*, 2022. 3

[53] Rahul Sajnani, Adrien Poulenard, Jivitesh Jain, Radhika Dua, Leonidas J. Guibas, and Srinath Sridhar. Condor: Self-supervised canonicalization of 3d pose for partial shapes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022. 3

[54] Ruwen Schnabel, Roland Wahl, and Reinhard Klein. Efficient ransac for point-cloud shape detection. In *Computer graphics forum*, volume 26, pages 214–226. Wiley Online Library, 2007. 2

[55] Jonas Schult, Francis Engelmann, Alexander Hermans, Or

Litany, Siyu Tang, and Bastian Leibe. Mask3d for 3d semantic instance segmentation. *arXiv preprint arXiv:2210.03105*, 2022. 1, 2

[56] Anthony Simeonov, Yilun Du, Andrea Tagliasacchi, Joshua B Tenenbaum, Alberto Rodriguez, Pulkit Agrawal, and Vincent Sitzmann. Neural descriptor fields: Se (3)-equivariant object representations for manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6394–6400. IEEE, 2022. 2, 3

[57] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. *Advances in Neural Information Processing Systems*, 33:7462–7473, 2020. 2

[58] Ziyang Song and Bo Yang. Ogc: Unsupervised 3d object segmentation from rigid dynamics of point clouds. *arXiv preprint arXiv:2210.04458*, 2022. 2

[59] Matthew Tancik, Pratul P. Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan T. Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *NeurIPS*, 2020. 2

[60] Linghua Tang, Le Hui, and Jin Xie. Learning inter-superpoint affinity for weakly supervised 3d instance segmentation. *arXiv preprint arXiv:2210.05534*, 2022. 1, 2

[61] Nathaniel Thomas, Tess Smidt, Steven Kearnes, Lusann Yang, Li Li, Kai Kohlhoff, and Patrick Riley. Tensor field networks: Rotation-and translation-equivariant neural networks for 3d point clouds. *arXiv preprint arXiv:1802.08219*, 2018. 3

[62] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, Junyeong Kim, and Chang D Yoo. Softgroup++: Scalable 3d instance segmentation with octree pyramid grouping. *arXiv preprint arXiv:2209.08263*, 2022. 2

[63] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022. 1, 2, 6

[64] Weiyue Wang, Ronald Yu, Qiangui Huang, and Ulrich Neumann. Sgpn: Similarity group proposal network for 3d point cloud instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2569–2578, 2018. 2

[65] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling. Coordinate independent convolutional networks–isometry and gauge equivariant convolutions on riemannian manifolds. *arXiv preprint arXiv:2106.06020*, 2021. 3

[66] Thomas Weng, David Held, Franziska Meier, and Mustafa Mukadam. Neural grasp distance fields for robot manipulation. *arXiv preprint arXiv:2211.02647*, 2022. 3

[67] Yizheng Wu, Min Shi, Shuaiyuan Du, Hao Lu, Zhiguo Cao, and Weicai Zhong. 3d instances as 1d kernels. In *European Conference on Computer Vision*, pages 235–252. Springer, 2022. 1

[68] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *arXiv preprint arXiv:1801.02209*, 2018. 2

[69] Fanbo Xiang, Yuzhe Qin, Kaichun Mo, Yikuan Xia, Hao Zhu, Fangchen Liu, Minghua Liu, Hanxiao Jiang, Yifu Yuan, He Wang, et al. Sapien: A simulated part-based interactive environment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11097–11107, 2020. 8

[70] Saining Xie, Jiatao Gu, Demi Guo, Charles R Qi, Leonidas Guibas, and Or Litany. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding. In *European conference on computer vision*, pages 574–591. Springer, 2020. 2

[71] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. Neural fields in visual computing and beyond. *Computer Graphics Forum*, 2022. 2

[72] Zhexin Xie, Peidong Liang, Jin Tao, Liang Zeng, Ziyang Zhao, Xiang Cheng, Jianhuan Zhang, and Chentao Zhang. An improved supervoxel clustering algorithm of 3d point clouds for the localization of industrial robots. *Electronics*, 11(10):1612, 2022. 2

[73] Mutian Xu, Pei Chen, Haolin Liu, and Xiaoguang Han. To-scene: A large-scale dataset for understanding 3d tabletop scenes. *arXiv preprint arXiv:2203.09440*, 2022. 7

[74] Yinshuang Xu, Jiahui Lei, Edgar Dobriban, and Kostas Daniilidis. Unified fourier-based kernel and nonlinearity design for equivariant networks on homogeneous spaces. In *International Conference on Machine Learning*, pages 24596–24614. PMLR, 2022. 3

[75] Zhengrong Xue, Zhecheng Yuan, Jiashun Wang, Xueqian Wang, Yang Gao, and Huazhe Xu. Useek: Unsupervised se (3)-equivariant 3d keypoints for generalizable manipulation. *arXiv preprint arXiv:2209.13864*, 2022. 3

[76] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. *Advances in neural information processing systems*, 32, 2019. 2

[77] Cheng-Kun Yang, Yung-Yu Chuang, and Yen-Yu Lin. Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7335–7344, 2021. 2

[78] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019. 2

[79] Hong-Xing Yu, Jiajun Wu, and Li Yi. Rotationally equivariant 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1456–1464, 2022. 3

[80] Biao Zhang and Peter Wonka. Point cloud instance segmentation using probabilistic embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8883–8892, 2021. 2

[81] Minghan Zhu, Maani Ghaffari, and Huei Peng. Correspondence-free point cloud registration with so (3)-equivariant implicit shape representations. In *Conference on Robot Learning*, pages 1412–1422. PMLR, 2022. 3, 5