GenSDF: Two-Stage Learning of Generalizable Signed Distance Functions

Anonymous Author(s)
Affiliation
Address
email

Abstract

We investigate the generalization capabilities of neural signed distance functions (SDFs) for learning 3D object representations for unseen and unlabeled point clouds. Existing methods can fit SDFs to a handful of object classes and boast fine detail or fast inference speeds, but do not generalize well to unseen shapes. We introduce a two-stage semi-supervised meta-learning approach that transfers shape priors from labeled to unlabeled data to reconstruct unseen object categories. The first stage uses an episodic training scheme to simulate training on unlabeled data and meta-learns initial shape priors. The second stage then introduces unlabeled data with *disjoint classes* in a semi-supervised scheme to diversify these priors and achieve generalization. We assess our method on both synthetic data and real collected point clouds. Experimental results and analysis validate that our approach outperforms existing neural SDF methods and is capable of robust zero-shot inference on 100+ unseen classes.

1 Introduction

Learned 3D representations can form the core of shape recovery [1], robotic manipulation [2], scene understanding [3], and content generation [4] tasks. While simple to interpret, high-quality explicit 3D representations can come with a slew of hidden costs. Conventional voxel grids without acceleration suffer from cubical memory requirements [5, 6, 7], dense point clouds contain large amounts of redundant information make querying local structures computationally expensive [8, 9], and direct optimization of mesh representations is often infeasible [10]. As an alternative, works such as [11, 12, 13] propose *implicit* learned representations; networks trained to functionally approximate some underlying 3D geometry. These methods can offer lower memory requirements and faster training and inference speeds [14] than conventional explicit representation methods [5, 8].

Neural signed distance functions (SDFs) [11] are a subset of this work which implicitly model the distance from a queried location to the nearest point on a shape's surface – negative inside the shape, positive outside, zero at the surface. Existing methods are capable of approximating diverse synthetic geometry, including multiple objects [11, 15, 16] and varying levels of detail for single objects [17, 18]. However, most successful methods are fully supervised–they require access to ground truth signed distance values to fit to an object. As ordinary point clouds lack these values, a fully-supervised approach cannot learn from such unlabeled data.

To replace the need for any annotated data, fully unsupervised approaches [19, 20, 21] operate directly on point clouds. These methods leverage pseudo-labels and proxies for ground truth signed distance values, for example the unsigned distance to the nearest point in the point cloud. Unfortunately,

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

existing unsupervised methods can only reconstruct a handful of shapes, and break down when we introduce just three or four additional additional object categories. As a result, these approaches can also not take advantage of large unlabeled point cloud datasets [22, 23, 24] and their real-world applications are limited. Our proposed method lifts this limitation, and learns from large-scale unlabeled data, successfully reconstructing an order of magnitude more categories in inference than seen during training.

To achieve this we propose a semi-supervised learning approach. Existing semi-supervised methods use labeled and unlabeled data of the same class to improve intra-class performance for tasks such as 3D reconstruction from images [25, 26, 27] and scene reconstruction from point clouds [3, 28]. Although they perform well on seen data, these methods are trained on labeled and unlabeled data of the same class, and so do not by themselves generalize well to unseen classes. We propose a novel semi-supervised meta-learning learning approach that learns geometric relations between labeled and unlabeled data of non-overlapping categories. One which learns an SDF that generalizes to a diverse set of unseen, out-of-distribution geometries.

We train in two stages. In the first meta-learning stage, we learn shape priors from the labeled set. Different from existing supervised methods, we introduce an episodic training scheme where we split data into subsets with disjoint classes to learn a generalized shape representation. One part of this split is treated as a labeled supervised set of data, with ground truth signed distance penalties, while the other is used to emulate unlabeled data samples, with only self-supervised losses. We validate that this first stage provides a generic shape prior which favors expressiveness at the cost of reconstruction quality. In the second, semi-supervised stage, we introduce a new fully disjoint set of unlabeled data to the training scheme. Successful training hinges on our novel self-supervised sign prediction loss, which diverges from the unsigned convention of previous works [19, 20, 21]. This second stage allows the model to ingest large amounts of unlabeled data to diversify the shape model learned in the first stage, which significantly increases the quality of the signed distance predictions over a wide set of unseen shape classes.

We evaluate our method on Acronym [29], a synthetic dataset, and YCB [23], a collection of real-world point clouds. We validate the generalization capabilities of our approach, and find that it outperforms state-of-the-art learned SDF methods and is capable of robust zero-shot inference [30] on 100+ unseen classes and scanned point clouds. We will release code and trained models.

2 Related Work

Learning Implicit Surface Representations. Implicit neural representations form the backbone of many modern 3D reconstruction and view synthesis works [11, 12, 13, 31, 32]. They use neural networks as *universal approximators* [33] to directly learn a mapping between spatial or spatio-temporal [34] coordinates and scene attributes, which offers a fully-differentiable and space-efficient [28] way to represent 3D geometry. Park et al. [11] first experimented with mapping coordinates to signed distance values – a neural signed distance function (SDF) – whereby an object is implicitly represented by the distances between 3D coordinate queries their closest surface point.

With labeled data – when all 3D query points are annotated with ground truth signed distance values – predictions of signed distance values can be supervised directly [11, 15]. Existing methods [11, 15] require test-time fine-tuning even for a single class, prohibiting real-time inference or prediction on unlabeled data, or overfit to single objects [17, 18]. Occupancy-based methods [16, 35, 36] can learn a rudimentary shape prior, but still struggle to reconstruct fine features in unseen shapes. The proposed semi-supervised method learns a robust shape prior, and substantially outperforms fully supervised methods when tested on a set of over one hundred unseen categories.

With only unlabeled data – un-annotated point clouds – approaches try to generate proxies for signed distance values for supervision. Ma et al. [19] look at the unsigned distance to the nearest point in the point cloud as a proxy, and Atzmon and Lipman [20] propose a class of loss functions and geometric initializations such that these unsigned distances converge to signed values. We propose a signed

nearest neighbor loss and demonstrate that our approach, unlike existing unsupervised methods, is able to reconstruct thin object structures and converge on a dataset with large object variance.

Semi-supervised Surface Representations. Traditional semi-supervised methods rely on learning from datasets where each category contains labeled and unlabeled samples. During inference they rely on pseudo-labels [37, 38], or employ ensemble techniques to enforce consistency [39]. These methods cannot be directly applied to target data from unseen categories.

For 3D representation tasks, existing methods [25, 26, 27] use semi-supervision to reconstruct geometries from images or point clouds [3, 28]. Elich et al. [25] pretrain their model with labeled data to learn 3D representations, to recover known shapes from RGBD data. Li et al. [3] use labeled data as dense boundary values for semi-supervised SDF learning, and fit road scenes from point clouds. While these approaches improve predictions on seen data, they cannot generalize well to unseen classes, even with test-time optimization [3]. Recently, person re-identification and domain adaptation methods perform semi-supervision on non-overlapping labeled and unlabeled classes and have shown to generalize better [40, 41, 42]. This has not been attempted for learning SDFs, and our method introduces a meta-learning approach to learn shape priors, where we emulate training on unlabeled data with pseudo-labels.

Generalization. The ability to learn generalizable 3D representations is necessary for real-world applications, as real objects exhibit significantly more diversity than any dataset. In learning implicit functions, one approach [43, 44] is to break down shape features to find common properties, but this also requires prior knowledge about target shapes which might not be available.

We propose a meta-learning approach that does not make assumptions or require prior information about target data. Meta-learning has been extensively studied in few-shot learning [45, 46], domain generalization [47], and image retrieval [39], 3D reconstruction from 2D or 2.5D images [48, 49, 50], and point cloud classification and segmentation tasks [51, 52, 53]. However, meta-learning has not been extensively investigated in the context of SDFs. The most relevant method is MetaSDF [15], albeit it is fully supervised and exhibits large time and memory complexity due to inner loop gradient descent steps. Furthermore, MetaSDF overfits to single objects and requires ground truth signed distance values during test time. The proposed episodic training approach avoids this costly inner loop, and learns a unified network that only requires a raw input point cloud for reconstruction.

3 Semi-supervised Meta-Learning of Implicit Functions

We next formalize the problem we are solving in Sec. 3.1, then describe the proposed learning method and training objectives in Sections 3.2 and 3.3.

3.1 Problem Formulation

Conditional Signed Distance Functions We are interested in learning a generalized representation for shapes. We learn a signed distance function (SDF), which is a continuous function that represents the surface of a shape approximated by the zero level-set of a neural network. Given a raw input point cloud $P = \{p_i \in \mathbb{R}^3\}_{i=1}^N$ with N points and a 3D query point $x \in \mathbb{R}^3$, we learn a conditional SDF $\Phi : \mathbb{R}^3 \times \{p_i \in \mathbb{R}^3\}_{i=1}^N \to \mathbb{R}$ such that the function Φ predicts the signed distance value for a 3D coordinate conditioned on a sparse point cloud. Then Φ is a unified function that can be directly applied to any target point cloud.

$$x, P \mapsto \Phi(x, P) = s, \tag{1}$$

where s denotes the predicted signed distance value between x and the shape described by P.

The surface boundary of a shape described by P is its zero-level set $S_0(\Phi(P))$, which can be expressed as

$$S_0(\Phi(P)) = \{z \in \mathbb{R}^3 \mid \Phi(z, P) = 0\}. \tag{2}$$

Thus, given a trained Φ, we can visualize the surface of some P by drawing its zero-level set.

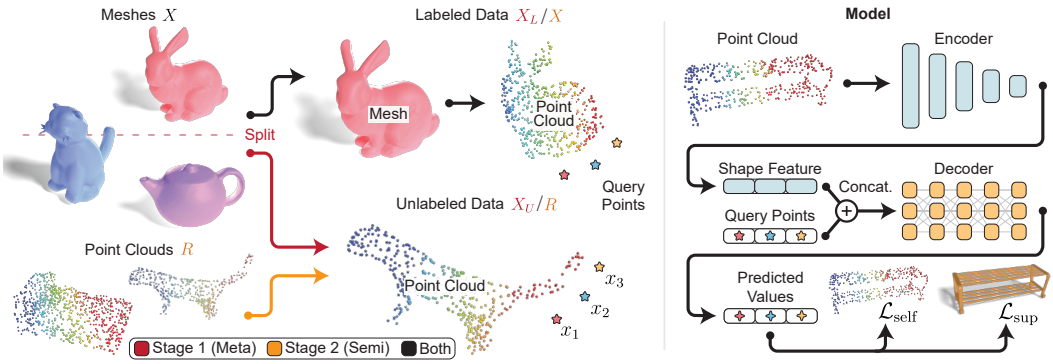


Figure 1: Our two-stage learning pipeline. The first stage (**red**) episodically splits the dataset into two subsets of disjoint categories of emulated labeled and unlabeled data, trained with combined supervised $\mathcal{L}_{\text{sup}}$ and self-supervised $\mathcal{L}_{\text{self}}$ loss. The second stage (**orange**) introduces real unlabeled data with non-overlapping classes in a semi-supervised fashion to improve model generalization.

Problem Setting We assume both labeled and unlabeled datasets are available, denoted X and R, respectively. $X = \{P_X, \text{SDF}\}$, where P_X is a set of point clouds and $\text{SDF}(\cdot)$ denotes the ground-truth SDF operator that is defined for all query points $x \in \mathbb{R}^3$. $R = \{P_R\}$, where P_R is a set of point clouds belonging to classes disjoint with those of P_X. Note that each point cloud is a set of 3D coordinates that describe on the surface boundary of an object, and contains no additional information such as normals.

To learn a generalizable conditional SDF, we propose a two-stage semi-supervised meta-learning algorithm. In the first stage, we aim to learn shape priors using X. As illustrated in Figure 1, we propose to train our model using an episodic scheme [47], where every f epochs we randomly split X into two subsets with *disjoint* categories. In the second stage, as shown in Figure 1, we continue to train the pretrained weights from the first stage and train the model on X and R in a semi-supervised fashion. We obtain a unified neural network Φ that learns from one dataset and can generalize to represent SDFs of objects in unseen categories during test time; i.e., zero-shot inference [30].

3.2 Stage 1: Meta-Learning a Shape Prior

Our eventual goal is to train on labeled and unlabeled data simultaneously, so we mimic this goal in a meta-learning approach as we learn shape priors from X. Every f epochs, we split X into two subsets X_L and X_U with disjoint categories. We train X_L with a supervised loss $\mathcal{L}_{\text{sup}}$, similar to typical supervised approaches on labeled data. We take away ground-truth signed distance values from X_U making it "pseudo-unlabeled", and train it with our self-supervised loss $\mathcal{L}_{\text{self}}$. Our training objective is to minimize

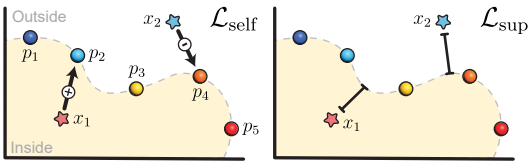


Figure 2: Our self-supervised (**left**) loss uses the nearest neighbor of a query point to approximate *signed* distance. The supervised loss (**right**) instead penalizes based on exact signed distance.

$$\mathcal{L}_{\text{meta}} = \mathcal{L}_{\text{sup}}(X_L) + \lambda_m \mathcal{L}_{\text{self}}(X_U) \tag{3}$$

where λ_m controls the strength of the self supervised loss $\mathcal{L}_{\text{self}}$. Compared to only training with a supervised loss, this meta-learning approach intends to accomplish two goals. First, the model learns to backpropagate with our self-supervised loss, which we need for training unlabeled data in the second stage. Second, episodic training on non-overlapping categories forces the model to develop generalized representations as it cannot overfit to the data. We empirically show in our ablation study that our method has stronger representation power for both seen and unseen classes. We provide an example of a training step in Alg. 1 and a diagram in Fig. 1. Next, we explain our loss functions in more detail.

Algorithm 1 Stage 1: Meta-Learning a Shape Prior

preprocess: a labeled set X
initialize: learning rate η, split frequency f, hyperparameter λ_m, model Φ with parameters θ
for e **in range**(num_epochs):
    if $(e \texttt{ mod } f) == 0$:
        $X_L, X_U \leftarrow \texttt{split}(X)$
    $\mathcal{L}_{\text{sup}} = \frac{1}{|X_L|} \sum_{(x,\text{SDF}) \in X_L} \|\Phi(x;\theta^e), \text{SDF}(x)\|_1$ ▷ Eq. (4)
    $\mathcal{L}_{\text{self}} = \frac{1}{|X_U|} \sum_{(x,t) \in X_U} \|(x \pm \frac{x-t}{\|x-t\|} \cdot \Phi(x;\theta^e)),\ \ t\|_2^2$ ▷ Eq. (7), simplified
    $\mathcal{L}_{\text{meta}} = \mathcal{L}_{\text{sup}} + \lambda_m \mathcal{L}_{\text{self}}$ ▷ Eq. (3)
    $\theta^{e+1} \leftarrow \theta^e - \eta \nabla \mathcal{L}_{\text{meta}}$
return θ

Supervised Loss Component In the supervised setting, we have the ground truth SDF operator; i.e., we know the signed distance values for all query points x. Therefore, the loss function is simply

$$\mathcal{L}_{\text{sup}} = \frac{1}{K} \sum_{k \in K} \|\Phi(x_k) - \text{SDF}(x_k)\|_1, \tag{4}$$

where $\| \cdot \|_1$ is the L1 loss.

Self-supervised Loss Component for Unlabeled Training Samples For each point $x \in \mathbb{R}^3$, we use its closest point $p \in P$ to approximate its projection to the surface, defined as

$$t = \underset{p \in P}{\text{argmin}} \|x - p\|_2. \tag{5}$$

In the literature [54], it is common to use

$$\hat{t} = x - \nabla\Phi(x)\Phi(x) \tag{6}$$

to approximate the closest point t on the object surface to query point x given some trained SDF function Φ for tasks such as collision detection. We build on this insight: we want to predict $\hat{t}$ to approximate t on the point cloud by using the predicted signed distance value. However, we find that directly using Eq. (6) leads to highly inaccurate sign predictions. There are no explicit penalties for predicting the wrong sign and when the query point is close to the surface, an ℓ_2 loss between $\hat{t}$ and t can be very low even when the sign is predicted incorrectly. For a surface boundary to be accurately estimated, there must be accurate sign changes between 3D coordinates.

Thus, we substitute $\nabla\Phi(x)$ with the direction vector between t and x. Our formulation has an explicit penalty for incorrect sign predictions: the vector would point in the direction opposite to the point cloud, guaranteeing a larger error. Furthermore, we can leverage sign predictions from labeled data to directly supervise the sign. This is not possible in the original formulation since $\nabla\Phi(x)\Phi(x)$ is calculated as a unit that points in the direction of maximum distance change. We denote our approach self-supervised because the predictions of the signs are used as labels. Our training objective is to minimize the distance between $\hat{t}$ and t. Furthermore, we add an additional ℓ_1 loss by leveraging the point clouds which have a distance value of 0. We find that this helps with thinner structures. Our final loss function is

$$\mathcal{L}_{\text{self}} = \frac{1}{K} \sum_{k \in K} \|\hat{t}_k - t_k\|_2^2 + \lambda_p \frac{1}{N} \sum_{p \in P_{\text{unlab}}} \|\Phi(p_n)\|_1, \tag{7}$$

where λ_p determines the weight of the point cloud predictions and

$$\hat{t} = \begin{cases} x - \dfrac{x-t}{\|x-t\|}\ \Phi(x) & \text{if } \Phi(x) \geq 0, \\ \\ x + \dfrac{x-t}{\|x-t\|}\ \Phi(x) & \text{if } \Phi(x) < 0. \end{cases} \tag{8}$$

Previous work [19] has predicted t by directly calculating Eq. (6) during training, but their resulting sign predictions are substantially less accurate than ours for the reasons we discuss. We validate this with the results shown in Tab. 1. We note that a point cloud P is sampled from a surface and does not represent the continuous boundary. Thus, t is the nearest neighbor that exists in P but not necessarily the true one that exists in the target object we want to reconstruct. Therefore, unlike using labeled data, there are inaccuracies when using P as a proxy. As a result previous unsupervised work [19] that use P as a proxy are not robust when training on multiple classes. Our semi-supervised method lifts this constraint in the training so that these inaccuracies do not build up.

3.3 Stage 2: Semi-supervision with Unlabeled Training Samples

In our second stage, visualized in Fig. 1, we train our conditional SDF from the meta-learning stage on both labeled and unlabeled data in a semi-supervised fashion. As a result of the learned priors from the previous stage, this stage of training remains robust even with large amounts of unlabeled data. We show in our experiments (see Sec. 4.4 for an ablation study) that without meta-learning priors, semi-supervision is not as robust. Furthermore, continuing to train with labeled data provides strong supervision to ensure inaccuracies from unlabeled data do not build up. Given labeled set X and unlabeled set R, our training objective is to minimize

$$\mathcal{L}_{\text{semi}} = \mathcal{L}_{\text{sup}}(X) + \lambda_s \mathcal{L}_{\text{self}}(R) \tag{9}$$

where λ_s controls the strength of the self supervised loss $\mathcal{L}_{\text{self}}$.

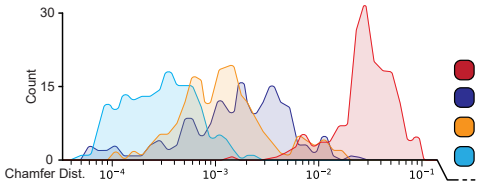
4 Experiments

In this section, we analyze and validate the proposed representation method for seen and unseen object categories. We begin with a discussion of implementation details and experimental setup.

Implementation For our experiments, we borrow the encoder architecture from [16] which is a modified PointNet [8] that uses plane projection to learn local geometric features, and parallel UNets [55] to aggregate spatial information. For the decoder we use an 8-layer multi-layer perceptron with 512-dimensional hidden layers, similar to the architecture from [11, 15, 19, 20]. We emphasize, that our two-stage training pipeline is designed to be architecture-agnostic, and we provide results for alternate encoders and decoders in the supplement. For Eqs. (3) and (9) we the set the loss coefficients to $\lambda_m = \lambda_s = 0.1$, and $\lambda_p = 0.01$ in Eq. (7). We provide a full architecture description and additional training details in the supplemental document.

Experimental Setup We source training and evaluation data from Acronym [29] and YCB [23]. Acronym is a post-processed subset of the popular ShapeNet [56] dataset which contains 8,872 clean, synthetic 3D models in 262 shape categories. In contrast to ShapeNet, all of Acronym's meshes are *watertight*, which ensures that they have well-defined SDFs. We use the 20 largest classes (2,995 meshes) as a labeled dataset X for all fully-supervised methods, including our meta-learning first stage. The next 76 classes (3,408 meshes) form our unlabeled dataset R for semi-supervised training. The remaining 166 classes (1,525 meshes) are withheld for testing. Preprocessing details can be found in the supplemental document. The YCB dataset [23] contains point clouds from 3D scans of common real-world objects and does not contain ground truth meshes. We use these point clouds as additional unlabeled test data to assess SDF predictions on out-of-distribution objects.

Evaluation Given a point cloud P that describes the surface of an object, the proposed method returns an implicit signed distance function $\Phi(P)$. We construct a cube with each dimension ranging from -1 to 1, then discretize it into a set of grids with resolution 256^3. We run Marching Cubes [57] at each grid point to reconstruct a mesh surface for qualitative assessment. For quantitative evaluation we measure the Chamfer Distance [8] between 30,000 randomly sampled points on the surfaces of the reconstructed and ground truth meshes.



	Seen Classes	Unseen Classes
DeepSDF	18.26/10.87	33.19/29.74
MetaSDF	4.269/1.536	2.834/1.746
ConvOccNet	1.348/0.715	2.333/1.032
Proposed	**0.351/0.166**	**0.407/0.250**

Figure 3: Mean/median Chamfer Distance (CD) of 20 seen and 166 unseen classes (**right**, scale is 10^{-4}). For our method we report numbers prior to any test-time refinement. We include a histogram plot for visualizing the per-class CD distribution of the unseen classes (**left**, log scale).

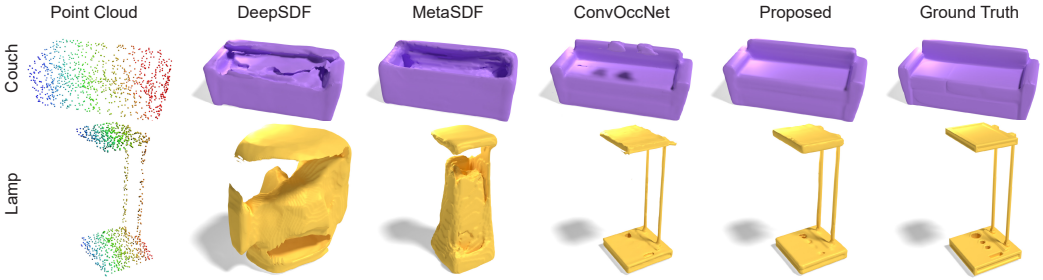


Figure 4: Reconstruction results for seen objects – objects in the train set. DeepSDF and MetaSDF struggle to recover thin structures and non-convex surfaces due to their lack of learned shape priors.

Experiments We conduct experiments to investigate the generalizability and representational power of our method. Specifically, we analyze the effects of reconstructing seen and unseen objects (Sec. 4.1), self-supervision on unlabeled data (Sec. 4.2), performance on real-world point cloud scans (Sec. 4.3), and ablation experiments (Sec. 4.4).

4.1 Reconstructing Seen and Unseen Shapes

In this subsection, we compare our method to existing approaches: DeepSDF [11], MetaSDF [15], and Convolutional Occupancy Networks (ConvOccNet) [16] on both seen and unseen shapes.[1] For reconstructing seen data, we evaluate results on the 20 shared classes (see the experimental setup paragraph in Sec. 4). For testing on unseen classes, we use the remaining 166 categories that are *disjoint* with any class used for training. To improve visual results, we apply refinement by fitting the model on the point cloud for a few iterations. Specifically, we employ our self-supervised loss without using any additional information beyond 5,000 raw input points. In contrast, DeepSDF [11] and MetaSDF [15] use ground truth signed distance values for refinement. We include specific refinement details in our supplement. Since ConvOccNet [16] does not perform refinement, we report quantitative results of the proposed method without refinement for a fair comparison.

Seen Categories. Fig. 4 shows reconstruction results on seen shapes. We observe that existing methods and our approach are capable of reconstructing seen data. However, DeepSDF and MetaSDF fail to reconstruct thin structures such as the lamp poles due to the lack of a shape prior. Quantitatively, reported in the middle column of Fig. 3, even though we use the same encoder as ConvOccNet [16] to learn shape priors, we achieve a lower CD by an order of magnitude.

Unseen Categories. When tested on unseen categories, as shown in Fig. 5, only our proposed method is able to reconstruct unseen categories, and accurately represents fine details such as narrow gaps in the Shelf example. Quantitatively, as shown in the right column of the table in Fig. 3, the proposed method achieves better performance. Notably, the gap between the CD of seen and unseen classes is lower in our case. While the CD of most methods double between seen and unseen classes, there is only a marginal increase for ours.

[1] Approaches such as [17, 58, 18] are designed to overfit to single objects, which significantly deviates from the focus of this paper, so we do not include comparisons with these methods.

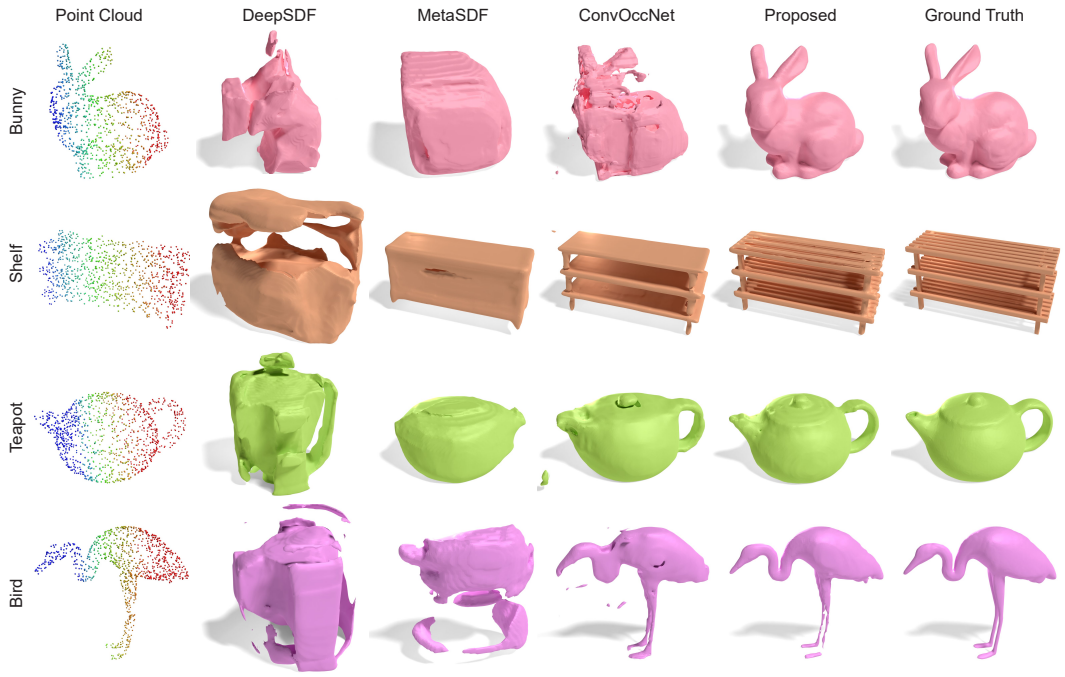


Figure 5: Reconstruction results for *unseen categories* – never included during training. Only our method recovers fine detail in unseen categories, such as the gaps in *Shelf*'s surfaces and the neck of *Bird*. Compared to ConvOccNet our approach also produces fewer artifacts in empty space.

4.2 Comparison to Unsupervised Methods on Unlabeled Data

A self/unsupervised method is essential for our semi-supervised stage to train on unlabeled data and for test-time refinement on only the raw input point cloud. In this subsection, we compare to existing unsupervised baselines: NeuralPull [19] and SAL [20]. For a fair comparison, we use only our self-supervised loss in Eq. (7) to train directly on unlabeled data and do not perform test-time refinement (*self* in Tab. 1).

Table 1: Mean/median Chamfer Distance for self- and un-supervised methods when trained on 5 objects from the same *QueenBed* class and 352 objects from 4 classes (*QueenBed*, *Plant*, *TrashBin*, *DeskLamp*). We report pre-refinement numbers for our approach, scale is 10^{-4}.

	Queen Beds	Four Classes
SAL	1.786 / 1.869	65.55 / 49.38
NeuralPull	1.745 / 1.502	2.419 / 1.812
Proposed (self)	0.146 / 0.139	0.218 / 0.184
Proposed	**0.081** / **0.077**	**0.089** / **0.061**

Tab. 1 validates that when training a single Queen Bed class, our explicit sign penalty allows our self-supervised method to reconstruct sharper details from irregular surfaces. On the other hand, existing methods [20, 19] produce low losses for close approximations to the absolute values of the signed distances, rather than accurate sign predictions, resulting in loss of details where there is abrupt change in the surface (see supplement for visualizations). All non-supervised methods we test on, including ours, degrade when training degrade quickly as the number of meshes increases.

The proposed method with semi-supervision scales well as we increase the number of classes. Thus, our method can tackle large-scale operations and take advantage of the availability of large amounts of unlabeled data. We also show in our supplement that using our self-supervised variant from this section leads to significantly more robust training.

4.3 Reconstruction of Real-World Raw Point Clouds

Next, we test our model on the YCB [23] dataset, which contains scanned point clouds and 2D images but no ground truth meshes. YCB is a different domain from Acronym [29] and we perform no

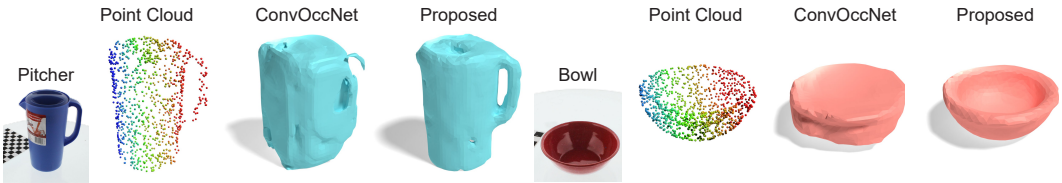


Figure 6: Reconstruction results for real scanned data from unseen categories in YCB [23]. This is a difficult setting, as the data has a different canonicalization, scaling, and distribution. Our proposed method is able to recover structures such as *Pitcher*'s handle and the inside of *Bowl*.

scaling or normalization of the input point cloud to thoroughly test performance of out-of-distribution data. We omit results from DeepSDF [11] and MetaSDF [15] since they use ground truth signed distance values during test time. As shown in Fig. 6, ConvOccNet [16] fails to reconstruct input data with a different scale and loses detail such as the handle of the pitcher. The proposed method, in contrast, is able to approximate the 3D shape with detail and results in plausible shape geometry, illustrating the ability of handling in-the-wild and out-of-distribution data. In the future, adding more diverse unlabeled data to the training of the proposed model, including different domains and scaling, may further boost the generalizability to any target data.

4.4 Ablation Experiments

In this subsection, we analyze individual components of our two-stage learning. We show quantitative results in Tab. 2. *Only meta* refers here to the meta-learning first stage. This outperforms existing fully supervised methods (see Fig. 3) for both seen and unseen data, while using the same amount of data and no additional manual annotations. The favorable results validate the generalization capabilities of our episodic training scheme. Our semi-supervision stage further improves reconstruction quality by introducing more shapes without requiring annotations.

Table 2: Mean/median Chamfer Distance (CD) for various method ablations on 20 classes of seen data and 166 classes of unseen data. We report pre-refinement numbers for our approach, scale is 10^{-4}.

	Seen Classes	Unseen Classes
Proposed	**0.351/0.166**	**0.407/0.250**
Only meta	0.503/0.218	0.515/0.297
No split	0.695/0.468	1.164/1.367
Only semi	0.821/0.510	0.889/0.612

No split refers to training on the labeled set without episodic training and resplitting. This setting is similar to that of ConvOccNet [16] with the only difference being the choice of decoder. Our decoder produces a lower CD because it does not produce artifacts in empty space. The CD for seen classes is similar to that of our meta-learning stage, but falls behind when reconstructing unseen classes. *Only semi* refers to only training the proposed semi-supervision stage from scratch without transferring the learned representation from the meta-learning stage. Even though this stage has access to both labeled and unlabeled data, it has a higher CD compared to our final two-stage approach. This is because errors in the proxies of unlabeled data accumulate and result in training that is less robust. This indicates that our meta-learning stage indeed builds a strong foundational prior that ensures the second stage semi-supervised training is robust.

5 Conclusion

In this work, we devise a two-stage semi-supervised meta-learning approach that achieves strong generalization to unseen object categories. This method is able to fit to any target point cloud without additional annotations and outperforms existing SDF methods in both seen and unseen categories. Our method is not without limitations, which we plan to address in the future. Similar to existing SDF representations [16], we do not achieve real-time inference due to large number of model parameters. Learning from large-scale multi-object point clouds, with abundant datasets for automotive scenes [59, 60] is also an exciting future direction. The proposed approach may also serve as a representation for inverse rendering methods that tackle vision tasks in an analysis-by-synthesis approach.

References

[1] Matthew Berger, Andrea Tagliasacchi, Lee M. Seversky, Pierre Alliez, Gaël Guennebaud, Joshua A. Levine, Andrei Sharf, and Claudio T. Silva. A survey of surface reconstruction from point clouds. *Comput. Graph. Forum*, 36(1):301–329, jan 2017. 1

[2] Danny Driess, Jung-Su Ha, Marc Toussaint, and Russ Tedrake. Learning models as functionals of signed-distance fields for manipulation planning. In *5th Annual Conference on Robot Learning*, 2021. 1

[3] Pengfei Li, Yongliang Shi, Tianyu Liu, Hao Zhao, Guyue Zhou, and Ya-Qin Zhang. Semi-supervised implicit scene completion from sparse lidar, 2021. 1, 2, 3

[4] Roy Or-El, Xuan Luo, Mengyi Shan, Eli Shechtman, Jeong Joon Park, and Ira Kemelmacher-Shlizerman. Stylesdf: High-resolution 3d-consistent image and geometry generation. *arXiv e-prints*, pages arXiv–2112, 2021. 1

[5] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 1

[6] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *Proceedings of the IEEE international conference on computer vision*, pages 2088–2096, 2017. 1

[7] Lingjie Liu, Jiatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 15651–15663. Curran Associates, Inc., 2020. 1

[8] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1, 6

[9] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017. 1

[10] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan Russell, and Mathieu Aubry. AtlasNet: A Papier-Mâché Approach to Learning 3D Surface Generation. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018. 1

[11] 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. 1, 2, 6, 7, 9

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

[13] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1, 2

[14] Thomas Davies, Derek Nowrouzezahrai, and Alec Jacobson. On the effectiveness of weight-encoded neural implicit 3d shapes. *arXiv preprint arXiv:2009.09808*, 2020. 1

[15] Vincent Sitzmann, Eric R. Chan, Richard Tucker, Noah Snavely, and Gordon Wetzstein. Metasdf: Meta-learning signed distance functions. In *Proc. NeurIPS*, 2020. 1, 2, 3, 6, 7, 9

[16] Songyou Peng, Michael Niemeyer, Lars Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *European Conference on Computer Vision*, pages 523–540. Springer, 2020. 1, 2, 6, 7, 9, 15

[17] David B Lindell, Dave Van Veen, Jeong Joon Park, and Gordon Wetzstein. Bacon: Band-limited coordinate networks for multiscale scene representation. *arXiv preprint arXiv:2112.04645*, 2021. 1, 2, 7

[18] Towaki Takikawa, Joey Litalien, Kangxue Yin, Karsten Kreis, Charles Loop, Derek Nowrouzezahrai, Alec Jacobson, Morgan McGuire, and Sanja Fidler. Neural geometric level of detail: Real-time rendering with implicit 3d shapes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11358–11367, 2021. 1, 2, 7

[19] Baorui Ma, Zhizhong Han, Yu-Shen Liu, and Matthias Zwicker. Neural-pull: Learning signed distance function from point clouds by learning to pull space onto surface. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 7246–7257. PMLR, 18–24 Jul 2021. 1, 2, 6, 8, 15

[20] Matan Atzmon and Yaron Lipman. Sal: Sign agnostic learning of shapes from raw data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2565–2574, 2020. 1, 2, 6, 8

[21] Matan Atzmon and Yaron Lipman. Sald: Sign agnostic learning with derivatives. *arXiv preprint arXiv:2006.05400*, 2020. 1, 2

[22] Timo Hackel, N. Savinov, L. Ladicky, Jan D. Wegner, K. Schindler, and M. Pollefeys. SEMANTIC3D.NET: A new large-scale point cloud classification benchmark. In *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume IV-1-W1, pages 91–98, 2017. 2

[23] Berk Calli, Arjun Singh, Aaron Walsman, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 international conference on advanced robotics (ICAR)*, pages 510–517. IEEE, 2015. 2, 6, 8, 9, 15

[24] Mikaela Angelina Uy, Quang-Hieu Pham, Binh-Son Hua, Duc Thanh Nguyen, and Sai-Kit Yeung. Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In *International Conference on Computer Vision (ICCV)*, 2019. 2

[25] Cathrin Elich, Martin R. Oswald, Marc Pollefeys, and Joerg Stueckler. Semi-supervised learning of multi-object 3d scene representations, 2021. 2, 3

[26] Atul Kanaujia, Cristian Sminchisescu, and Dimitris Metaxas. Semi-supervised hierarchical models for 3d human pose reconstruction. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 2, 3

[27] Jingtan Piao, Chen Qian, and Hongsheng Li. Semi-supervised monocular 3d face reconstruction with end-to-end shape-preserved domain transfer. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9397–9406, 2019. 2, 3

[28] Vincent Sitzmann, Julien N.P. Martel, Alexander W. Bergman, David B. Lindell, and Gordon Wetzstein. Implicit neural representations with periodic activation functions. In *arXiv*, 2020. 2, 3

[29] Clemens Eppner, Arsalan Mousavian, and Dieter Fox. ACRONYM: A large-scale grasp dataset based on simulation. In *Under Review at ICRA 2021*, 2020. 2, 6, 8

[30] Yongqin Xian, H. Christoph Lampert, Bernt Schiele, and Zeynep Akata. Zero-shot learning - a comprehensive evaluation of the good, the bad and the ugly. *TPAMI*, 2018. 2, 4

[31] Ilya Chugunov, Yuxuan Zhang, Zhihao Xia, Cecilia Zhang, Jiawen Chen, and Felix Heide. The implicit values of a good hand shake: Handheld multi-frame neural depth refinement. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 2

[32] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7210–7219, 2021. 2

[33] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. Multilayer feedforward networks are universal approximators. *Neural networks*, 2(5):359–366, 1989. 2

[34] Keunhong Park, Utkarsh Sinha, Jonathan T Barron, Sofien Bouaziz, Dan B Goldman, Steven M Seitz, and Ricardo Martin-Brualla. Nerfies: Deformable neural radiance fields. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5865–5874, 2021. 2

[35] 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

[36] 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

[37] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for large-scale search. *IEEE transactions on pattern analysis and machine intelligence*, 34(12):2393–2406, 2012. 3

[38] Yu Wu, Yutian Lin, Xuanyi Dong, Yan Yan, Wei Bian, and Yi Yang. Progressive learning for person re-identification with one example. *IEEE Transactions on Image Processing*, 28(6):2872–2881, 2019. 3

[39] Guan'an Wang, Qinghao Hu, Jian Cheng, and Zengguang Hou. Semi-supervised generative adversarial hashing for image retrieval. In *Proceedings of the European conference on computer vision (ECCV)*, pages 469–485, 2018. 3

[40] Yun-Chun Chen, Chao-Te Chou, and Yu-Chiang Frank Wang. Learning to learn in a semi-supervised fashion. In *European Conference on Computer Vision*, pages 460–478. Springer, 2020. 3

[41] Xiaomeng Xin, Jinjun Wang, Ruji Xie, Sanping Zhou, Wenli Huang, and Nanning Zheng. Semi-supervised person re-identification using multi-view clustering. *Pattern Recognition*, 88:285–297, 2019. 3

[42] Xiaomeng Xin, Xindi Wu, Yuechen Wang, and Jinjun Wang. Deep self-paced learning for semi-supervised person re-identification using multi-view self-paced clustering. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2631–2635. IEEE, 2019. 3

[43] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Carsten Stoll, and Christian Theobalt. PatchNets: Patch-Based Generalizable Deep Implicit 3D Shape Representations. *European Conference on Computer Vision (ECCV)*, 2020. 3

[44] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas Funkhouser. Local deep implicit functions for 3d shape. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4857–4866, 2020. 3

[45] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017. 3

[46] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018. 3

[47] Yogesh Balaji, Swami Sankaranarayanan, and Rama Chellappa. Metareg: Towards domain generalization using meta-regularization. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. 3, 4

[48] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Joshua B Tenenbaum, William T Freeman, and Jiajun Wu. Learning to Reconstruct Shapes From Unseen Classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018. 3

[49] B. Wallace and B. Hariharan. Few-shot generalization for single-image 3d reconstruction via priors. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3817–3826, Los Alamitos, CA, USA, nov 2019. IEEE Computer Society. 3

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

[51] Ali Cheraghian, Shafin Rahman, and Lars Petersson. Zero-shot learning of 3d point cloud objects. In *International Conference on Machine Vision Applications (MVA)*, 2019. 3

[52] Charu Sharma and Manohar Kaul. Self-supervised few-shot learning on point clouds. *Advances in Neural Information Processing Systems*, 33:7212–7221, 2020. 3

[53] Na Zhao, Tat-Seng Chua, and Gim Hee Lee. Few-shot 3d point cloud semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021. 3

[54] Miles Macklin, Kenny Erleben, Matthias Müller, Nuttapong Chentanez, Stefan Jeschke, and Zach Corse. Local optimization for robust signed distance field collision. In *Proc. ACM Comput. Graph. Interact. Tech. 3, 1*, 2020. 5

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing. 6

[56] 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. 6

[57] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, page 163–169, New York, NY, USA, 1987. Association for Computing Machinery. 6

[58] Julien N. P. Martel, David B. Lindell, Connor Z. Lin, Eric R. Chan, Marco Monteiro, and Gordon Wetzstein. Acorn: Adaptive coordinate networks for neural scene representation. *ACM Trans. Graph. (SIGGRAPH)*, 40(4), 2021. 7

[59] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3354–3361. IEEE, 2012. 9

[60] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 9

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Experiments (Sec. 4) and Conclusion (Sec. 5).
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not believe that our work has potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] We will release code and pretrained models. Full reproduction details are in supplement.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Implementations details can be found in supplement.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] Provided in supplement.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We will release code.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] We use datasets that are publicly available. References are provided.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A] We did not use human subjects.
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

A Appendix

In this Appendix, we provide additional experimental results based on reviewer comments. In the next revision we will integrate these additional results into our main text and supplementary material.

A.1 Additional YCB Results

We provide additional reconstructions for samples from the YCB [23] dataset in Fig. 7 after recentering and coordinate normalization. YCB point clouds are obtained by fusing real RGB-D images, which result in measurements with different canonicalization and distribution from our the simulated data we use for training. ConvOccNet [16] produces irregular surfaces and holes in the reconstruction, while our method generalizes to these real-world measurements and produces smooth surfaces and detailed structures such as *Pitcher*'s handle and spout holes, and the inner curves of the *Mustard Bottle*.

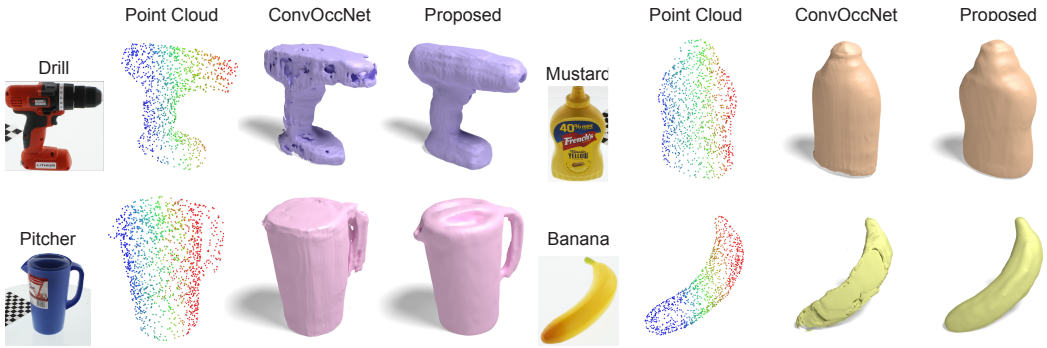


Figure 7: Additional reconstruction results for real scanned data from unseen categories in YCB [23] after recentering and normalization. Note the reconstruction of fine object features such as the *Pitcher* handle by our proposed method.

A.2 Detailed Comparison of Our Self-Supervised Loss vs NeuralPull

An unsupervised loss is essential for training with unlabeled data. In this subsection, we dive into a deeper comparison between our self-supervised loss and the approach described in NeuralPull [19].

First, we report accuracies of sign predictions between NeuralPull and our self-supervised method. For this experiment, we train on five point clouds from the QueenBed class (same setup as Sec. 5 of our supplementary), and run inference on 130,000 points near the surface of each point cloud (130,000 * 5 = 650,000 total). While both methods produced low ℓ_2 losses, in Tab. 4 our method achieves higher accuracy in sign prediction.

Table 3: Confusion matrices for NeuralPull and our self-supervised loss sign predictions on five point clouds of the QueenBed class. Of the five point clouds, there are a total of 650,000 points to predict, with 449,650 being positive and 200,350 being negative.

NeuralPull	Pred Positive	Pred Negative
449,650 Pos	37,779 (0.974)	11,871 (0.026)
200,350 Neg	37,986 (0.190)	162,363 (0.810)

Our SSL	Pred Positive	Pred Negative
449,650 Pos	449,245 (0.999)	405 (0.001)
200,350 Neg	2,765 (0.014)	197,585 (0.986)

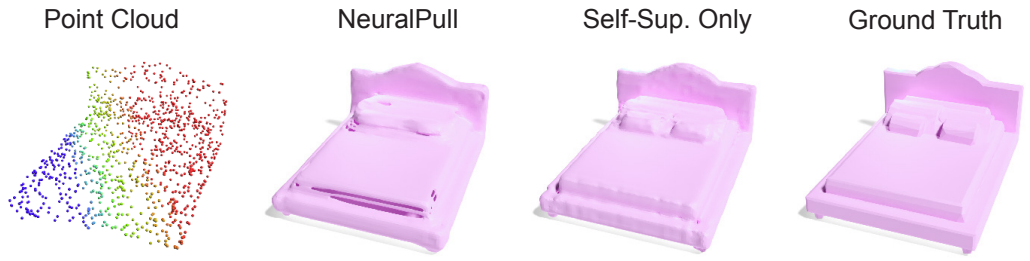


Figure 8: Comparison of self/unsupervised methods. Our self-supervised loss better penalizes incorrect signs and leads to sharper details from irregular surfaces such as pillows. All results are without test-time refinement.

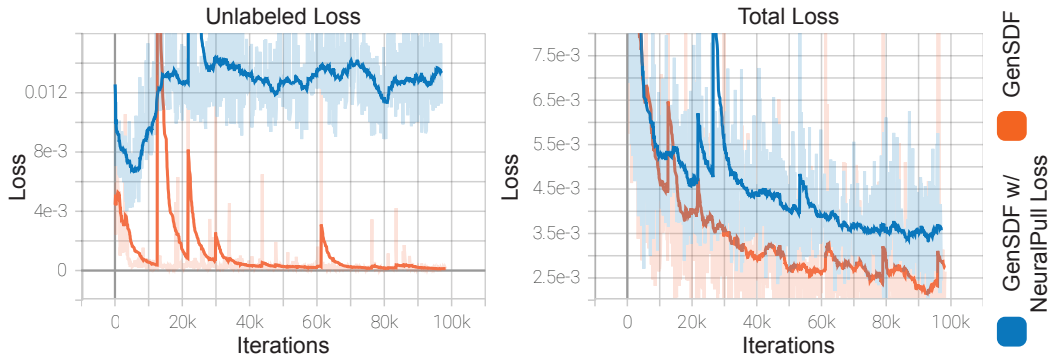


Figure 9: Training curves of GenSDF with our self-supervised loss vs GenSDF with NeuralPull's loss. (Left) Only logging the self-supervised loss vs NeuralPull's loss. (Right) Curve of the total loss.

We show visualizations in Fig. 8. Due to the more accurate sign predictions, our reconstruction results have sharper edges and details, while NeuralPull reconstructions often appear overly smooth and contain significantly more holes.

Next, we train the meta-learning first stage of our GenSDF model but swap out our self-supervised loss for NeuralPull's loss. We train on six classes from our labeled dataset. As we can see in Fig. 9, with NeuralPull's loss training converges at a local minima. We also provide quantitative results in Tab. 4. To visually understand the extremely high CD of NeuralPull, we visualize a reconstruction result in Fig. 10. Sign predictions for points inside the surface are mostly incorrect and a reasonable surface boundary is not formed. We manually inspected dozens of samples and confirm that all reconstructions share this problem.

Table 4: Comparing our self-supervised loss and NeuralPull's loss when trained with GenSDF. Mean/median Chamfer Distance (CD) of 166 unseen classes, scale is 10^{-4}.

	GenSDF (meta stage)	GenSDF w/ NeuralPull loss (meta stage)
Chamfer Distance	0.972 / 0.657	359.6 / 361.0

NeuralPull's inaccuracies build up as we increase the number of meshes, and its training objective conflicts with that of the supervised loss, which has accurate sign predictions. Thus, the model is unable to converge at the appropriate signed distance function.

A.3 Understanding the Effectiveness of Our Meta-learning Training

We provide two experiments to understand the effectiveness and limitations of our meta-learning approach.

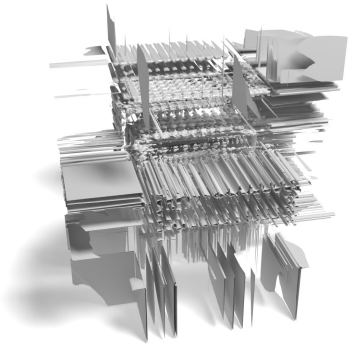


Figure 10: Reconstruction after training with GenSDF with NeuralPull's loss. Sign predictions for points inside the surface are mostly inaccurate and a reasonable surface boundary is not formed. All reconstructed samples we inspected share this problem regardless of the input point cloud.

Training on diverse vs similar classes We train two models on our meta-learning first stage. The first trains on the following six diverse shape classes, with a mix of small, large, planar, convex, and concave surfaces: 'Bench', 'QueenBed', 'EndTable', 'FloorLamp', 'Monitor', 'PottedPlant' The second trains on six classes that are semantically and geometrically similar, large planar shapes: 'EndTable', 'AccentTable', 'CoffeeTable', 'DiningTable', 'RoundTable', 'Table.' Then we evaluate on 166 unseen classes. The first model outperforms the second in reconstruction quality, validating that the diversity in the training set is more important than the raw number of classes. We do note that the difference in CD is noticeable but not very significant. This validates the effectiveness of our meta-learning approach: even with less diverse data, our model can still produce generalized priors. We provide quantitative results in Tab. 5 and training curves in Fig. 11.

Figure 11: Training curves of training on six diverse and six similar classes. When training on the latter, training loss decreases more because the model **overfits** on data. However, Tab. 5 verifies that training on diverse classes produces a prior that **generalizes** better to unseen classes.

Table 5: Results of training on six diverse and six similar classes. Mean/median Chamfer Distance (CD) of 166 unseen classes, scale is 10^{-4}.

	Six Diverse Classes	Six Similar Classes
CD	0.972 / 0.657	1.321 / 0.934

Training only with the supervised loss We train two separate models for a prior. Both train on six classes: 'Bench', 'QueenBed', 'EndTable', 'FloorLamp', 'Monitor', 'PottedPlant,' with a total of 600 meshes. The first model is trained using our two-stage learning approach. The second model is

trained with only the supervised loss, but introduces dropout (20%) and small amounts of Gaussian noise perturbations to the point clouds to reduce overfitting.

Then, we use both priors to train our second stage in exactly the same fashion. With our prior, our model performs significantly better when reconstructing unseen classes. We provide quantitative results on 166 unseen classes in Tab. 6. We highlight that after training stage 1, both methods perform on par on unseen classes. However, our prior that emulates semi-supervised training is able to significantly outperform in the second stage, where we introduce true unlabeled data samples.

Table 6: Comparison of our approach to a prior trained only with the supervised loss. Mean/median Chamfer Distance (CD) of 166 unseen classes, scale is 10^{-4}.

	Proposed	Supervised Prior (Augmentation + Dropout)
After stage 1	0.972 / 0.657	1.027 / 0.697
After stage 2	0.420 / 0.316	0.851 / 0.568