This CVPR 2021 paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Fostering Generalization in Single-view 3D Reconstruction by Learning a Hierarchy of Local and Global Shape Priors

Jan Bechtold[1,2] Maxim Tatarchenko[1] Volker Fischer[1] Thomas Brox[2]

[1]Bosch Center for Artificial Intelligence [2]University of Freiburg

Abstract

Single-view 3D object reconstruction has seen much progress, yet methods still struggle generalizing to novel shapes unseen during training. Common approaches predominantly rely on learned global shape priors and, hence, disregard detailed local observations. In this work, we address this issue by learning a hierarchy of priors at different levels of locality from ground truth input depth maps. We argue that exploiting local priors allows our method to efficiently use input observations, thus improving generalization in visible areas of novel shapes. At the same time, the combination of local and global priors enables meaningful hallucination of unobserved parts resulting in consistent 3D shapes. We show that the hierarchical approach generalizes much better than the global approach. It generalizes not only between different instances of a class but also across classes and to unseen arrangements of objects.

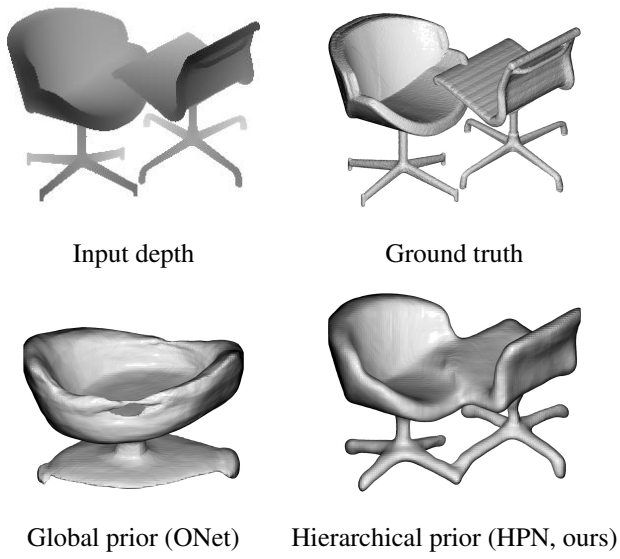

Figure 1. We employ a hierarchical shape prior to enable recombination of partial shapes observed during training. This significantly improves generalization compared to conventional global shape priors.

1. Introduction

The usual problem setting of single-view 3D reconstruction assumes an input image with a single dominant object, where the geometry of both the visible and the invisible part of this object shall be reconstructed. For the invisible parts, reconstruction must rely on shape priors, which can be based on the object class, symmetry, or smoothness. The geometry of the visible parts can be obtained, at least partially, from sensing data (e.g., depth, texture, shading).

Most existing approaches are encoder-decoder networks [7, 10, 12, 19, 27, 30, 34] and have been shown to barely generalize to novel shape categories [38]. Only few works have targeted generalization explicitly [3, 32, 38]. They argue that, for better generalization, the problem should be split into two parts: (1) prediction of a geometric representation of the visible parts from a single RGB image and (2) prediction of the final shape from the geometric representation. In this paper, we focus on the prediction of the object shape and assume the ground truth depth map to be already given as input. This reflects the argument that an intermediate depth map helps generalization [38] and should make the reconstruction of the visible parts almost trivial.

Surprisingly, however, existing approaches fail to generalize even in the visible areas, despite the perfect input. Consider the example in Fig. 1: ONet [19] trained on single chairs uses its learned prior to reconstruct the shape for an input with two chairs. Although the required shape prior (chairs) has been seen during training, the approach cannot use this knowledge to explain the clean observation of two chairs (Fig. 1 top left), which leads to an unresolved competition between observation and prior (Fig. 1 bottom left). This reveals a general problem of existing approaches: not only do they not generalize to new object classes, they even do not generalize to new combinations of the same training

classes. Even if we would train these networks on pairs of chairs, they must see all possible configurations of pairs – a combinatorial explosion.

In this paper, we propose to foster the recombination of previously seen partial shapes by a hierarchical approach. It consists of two main building blocks: (1) a local reconstruction module that reconstructs the shape at a certain level of locality (Fig. 2), and (2) fusion of the beliefs from various levels of locality (Fig. 3). The reconstruction module is effectively an implicit surface network (e.g. ONet) which performs shape estimation from patches of the input image. If the patch size covers the whole image, it comes down to the original global surface network. Intuitively, instead of reconstructing the full shape with a single prediction effort, local versions of the network learn to estimate geometry of individual object parts and put those together to obtain the whole shape. Since similar shape parts are likely to repeat between different categories, this strategy offers effective recombination of parts from various training samples and, hence, much better generalization potential.

Since local patches have a limited view of the overall shape, the reconstructed global shape may not look consistent, especially in large occluded areas. Therefore, we combine multiple patch sizes (including the global one based on the full image) to form a hierarchy of such local networks. The combination is possible by simple averaging of the logit outputs.

We demonstrate the intriguing effect of the new hierarchical reconstruction concept on various generalization tasks derived from the ShapeNet [5] dataset. This includes tasks that require inter-class generalization and generalization from single to multiple objects. The results show the huge effect of the ability to recombine parts, which is missing in all previous learning-based reconstruction approaches. This ability also improves the data efficiency: in contrast to existing global methods, the performance of our local networks does not noticeably degrade even when training on as little as 1% of the original data. Since the choice of the base reconstruction module is flexible, the hierarchy of local networks acts as a working principle that can be applied to enhance the generalization of effectively any method based on implicit functions. We refer to this as Hierarchical Prior Network (HPN).

2. Related Work

3D representations. A large portion of single-view 3D reconstruction research has dealt with developing methods that operate on different 3D representations. Those include voxels [7], octrees [30], patch-based [12] or deformable [34] meshes, point clouds [10], nested depth maps [27] and implicit functions [19, 11]. All these pipelines effectively follow the same design: a 2D encoder which compresses the input image into a single global latent vector and a 3D decoder which regresses the output 3D representation from it.

3D parts. Multiple works reconstruct the output shape as a collection of 3D parts which can come in form of cuboids [17, 21, 33, 40], superquadrics [22, 23], convex elements [9] or actual semantic parts [14, 36]. All these approaches use parts solely as an alternative 3D representation and do not provide a mechanism for attending to local patches of the input image. This is different for our method: we directly consider the relationship between local input patches and their 3D counterparts. Note also that we do not make any assumptions about shape parts being semantically meaningful, which makes our approach general and prevents the need for having semantic annotations similar to [20].

Generalization. Only few methods explicitly touch the matter of generalization to shape categories unseen during training. Shin *et al*. [29] and Tatarchenko *et al*. [31] analyze the conventional setup and conclude that working in the viewer-centered mode is a necessary (though not sufficient) condition for generalization. Zhang *et al*. [38], Wu *et al*. [35] and Thai *et al*. [32] propose to predict intermediate geometric representations in the pipeline and show that this improves generalization. In our work, we use a similar setting but further simplify it by starting from a ground truth depth map. Surprisingly, we find that even then the actual generalization achieved by existing methods is still limited. Thai *et al*. [32] show that using three-degree-of-freedom camera poses and SDFs as a 3D representation, while keeping the architecture from [19], helps generalize to a new dataset.

Local encoding. Several existing works proposed to include local encoding modules into the pipeline. Xu *et al*. [37] combine local and global features with the aim of improving the reconstruction details. However, their method is not forced to use local information and could in principle ignore it, plus they never explicitly target the generalization setting. For a special case of reconstructing human clothing, Saito *et al*. [28] propose to align local per-pixel features to the global shape context, thus explicitly leveraging the 2D-3D relationship. Peng *et al*. [25] combine a local encoder with an implicit function decoder for a task of point-cloud-based surface reconstruction. Similarly, multiple works [1, 13, 4] target a setting where surfaces are locally reconstructed from sparse multi-view observations. Similar in spirit to our approach, Chibane *et al*. [6] propose to extract a hierarchy of features for solving several 3D-to-3D tasks. Bautista *et al*. [3] locally assign features and 3D points in order to get a more expressive intermediate shape representation. Most similar to ours is the work from Genova *et al*. [11]. Local Gaussian regions of the input depth map are encoded and decoded independently. The global 3D shape results from the sum of

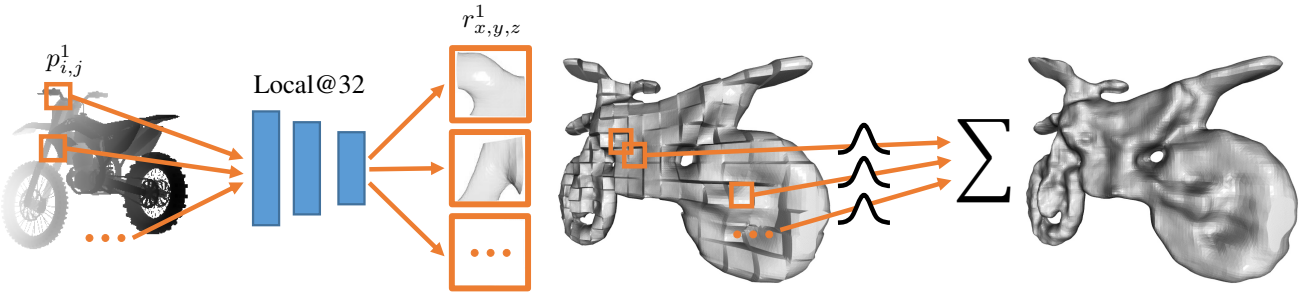

Figure 2. The proposed local reconstruction module independently reconstructs the shapes of individual patches of the input in a sliding window fashion. The resulting overlapping 3D parts are aggregated with Gaussian-weighted averaging into the final shape estimate.

the deep local implicit functions. However, the module that distributes the Gaussian regions requires a global context and can break if major dataset priors, like that of having a single object, are violated.

3. Method

The core idea of our approach is based on two observations: (1) effective generalization to new classes and new configurations requires the recombination of partial shapes seen during training; (2) recombination of such partial shapes requires (local) support regions of different sizes in the input image.

Although the regular encoder-decoder networks consider a hierarchy of multiple receptive field sizes when observing the input, they do not learn local *priors* during training. This is because their loss function only considers the whole object reconstruction, for which all of the input image and all of the ground truth shape is observed. While all the information for recombination is available, there is nothing in the training procedure that requires and fosters recombination.

For this reason, we combine multiple local reconstruction networks that only observe a cut-out part of the image and the corresponding cut-out part of the ground truth shape during training. The different levels of locality yield networks that have learned more specialized (global) or less specialized (local) priors. In their combination, they enable part recombination at all locality levels and consistency of the global shape at the same time.

3.1. Local Reconstruction

Consider a single-channel input depth map $d \in \mathbb{R}^{W \times H}$ of width W and height H pixels, and its corresponding ground truth 3D model D represented as a mesh with vertices V_D and faces F_D. Following the conventional setup in literature, we assume that D is normalized such that it fits into a unit cube.

For the ℓ-th hierarchy level, we denote with $N^\ell \in \mathbb{N}$ the width and height in pixels of a square patch $p^\ell_{i,j} \subset d$ centered at pixel position (i, j). These patches are positioned across the input d using a stride of s^ℓ_{train} pixels. For each $p^\ell_{i,j}$ there is a corresponding 3D volume $r^\ell_{x,y,z}$ centered at position (x, y, z) in the ground-truth 3D model. In the general case, the shape of $r^\ell_{x,y,z}$ is a frustum determined by the internal camera parameters, and the 3D position x, y, z depends on the patch location i, j and the camera model. For simplicity, we assume an orthographic camera model which results in $r^\ell_{x,y,z}$ being a cuboid with $x = y = M \in (0, 1)$ and $z = 1$. However, the whole setup could be extended to support perspective cameras.

Our local reconstruction module is an implicit function f^ℓ, for example an Occupancy Network (ONet), which takes as input a patch $p^\ell_{i,j}$ and some points $\mathbb{SP}^{K\times 3}$ in $r^\ell_{x,y,z}$ and outputs 3D predictions for $r^\ell_{x,y,z}$ in form of an occupancy logit or signed distance value for every input point. ONet could be replaced by any other network that implements an implicit function in 3D.

We extract a mesh from the occupancy logits by using Marching Cubes [18] with an empirically determined threshold τ as described in Occupancy Networks. We use the same procedure if the backbone network predicts SDF values, but determine a new threshold.

At training time, each 3D part is effectively treated as an independent sample, i.e. the only difference to the original ONet is in the training data. Therefore we normalize the training points from the 3D part $r^\ell_{x,y,z}$ to lie within $[-0.5, 0.5]$ in all three dimensions. Similar to Occupancy Networks, during training, we only provide a randomly sampled subset of training points to the network.

During inference, the network is applied in a sliding window fashion with a 2D stride s^ℓ_{infer}, such that each 3D region of the prediction gets updated by multiple parts. This enables smoother transitions between adjacent parts. We fuse predictions from multiple parts together by Gaussian-weighted averaging of the outputs of all contributing parts in the overlapping regions.

Since we assume that the camera model is known, there is a deterministic assignment between the predicted 3D parts and their absolute locations within the unit cube of the full shape. We use it to assemble a full reconstructed shape

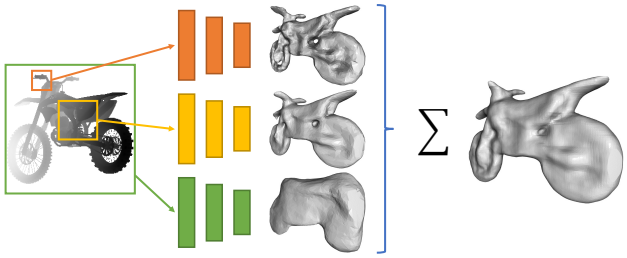

Figure 3. We use a hierarchy of networks operating on input patches of different resolutions (including the global one) to produce multiple shape reconstruction variants. Those are fused by simple averaging to yield the final reconstruction.

from individual predicted parts. An example of such a reconstruction for patches of size $N = 32$ and stride $s = 16$ is shown in Fig. 2.

3.2. Hierarchical fusion

We train multiple local reconstruction networks, each operating on different patch sizes N (including $N = W = H = 256$, i.e. the full image case). In case of non-square input images we suggest to use zero-padding in order to convert them into a square shape. Together the local reconstruction networks form a hierarchy of K independent predictions relying on priors of different locality levels $\ell \in \{1, \ldots, K\}$ which we then fuse into a single final prediction.

Similarly to averaging softmax outputs of overlapping parts in the previous section, we combine predictions from different hierarchy levels by averaging their corresponding softmax outputs. Since individual output values correspond to pseudo-probabilities that a certain 3D region is occupied, averaging them already provides an automatic mechanism to weigh the contribution of each level onto the final fused reconstruction. For example, in areas of the shape which are visible in the input image where the local reconstruction is usually more confident, local occupancy scores dominate those of the global one, and vice versa for invisible shape regions. The fusion of hierarchy levels is illustrated in Fig. 3 where three hierarchy levels of differing local patch sizes are fused to produce a single reconstruction. We call this combination of networks acting at multiple levels of locality Hierarchical Prior Network (HPN).

More sophisticated (learned) averaging schemes are conceivable, but come with the risk of overfitting to the training configurations. As we show in the experiments, already simple averaging leads to consistent shapes and is free from a bias to the training set.

4. Experiments

Existing approaches generalize to a certain degree to novel instances of a category seen during training. We target the more difficult generalization to novel categories and novel object assemblies.

4.1. Datasets

We train our method on two different subsets of the ShapeNet dataset [5]. (1) We report on the train split from Zhang *et al*. [38] referred to as *multi-class*, where networks are trained on planes, cars, and chairs. (2) We train on shapes from only a single category *(single class)*. These training categories are *chair* or *lamp*.

We evaluate our method on individual shape categories as suggested by Zhang *et al*. [38], both on the ones seen during training, corresponding to generalization across instances, and on those not seen during training, corresponding to generalization across classes.

In addition, we propose a new test set referred to as *Composition*, which allows us to explicitly evaluate generalization to novel object arrangements. We create it by placing up to three objects into one image. We exclusively use shape instances from the ShapeNet test set. For each compositional image, we randomly select the shape categories. Then, we pick objects of the selected categories and modify elevation and azimuth of their pose. Before rendering the image with PyTorch3D [26], we shift the objects along the x-axis to reduce their overlap.

4.2. Models

ONet. We train the original occupancy network [19] on the ground truth depth images.

GenRe. GenRe [38] is the pioneer work for generalization to novel categories. The GenRe network architecture consists of two parts. The first one estimates a depth map for a given RGB image. The second one reconstructs the 3D shape, given the depth image. We report the Chamfer distance from their paper for reconstructions from ground truth depth maps. For a comparison of all 13 test classes please see the supplemental.

LDIF. LDIF [11] represents 3D shapes as multiple local implicit functions and improves over ONet and GenRe, thus being the state-of-the-art method. We use their custom data preprocessing pipeline to train LDIF networks on single-view perspective depth images.

ONet-SDF. We use the same training points as for Occupancy Networks, but replace the binary occupancy label with the signed distance (SDF) of each point to the mesh surface. Points within the mesh have a negative distance. This also changes the training of the network from binary classification to regression. Instead of using the binary cross entropy as loss, we now use the L_1 loss. In order to extract a mesh from the SDF-values predicted by the network,

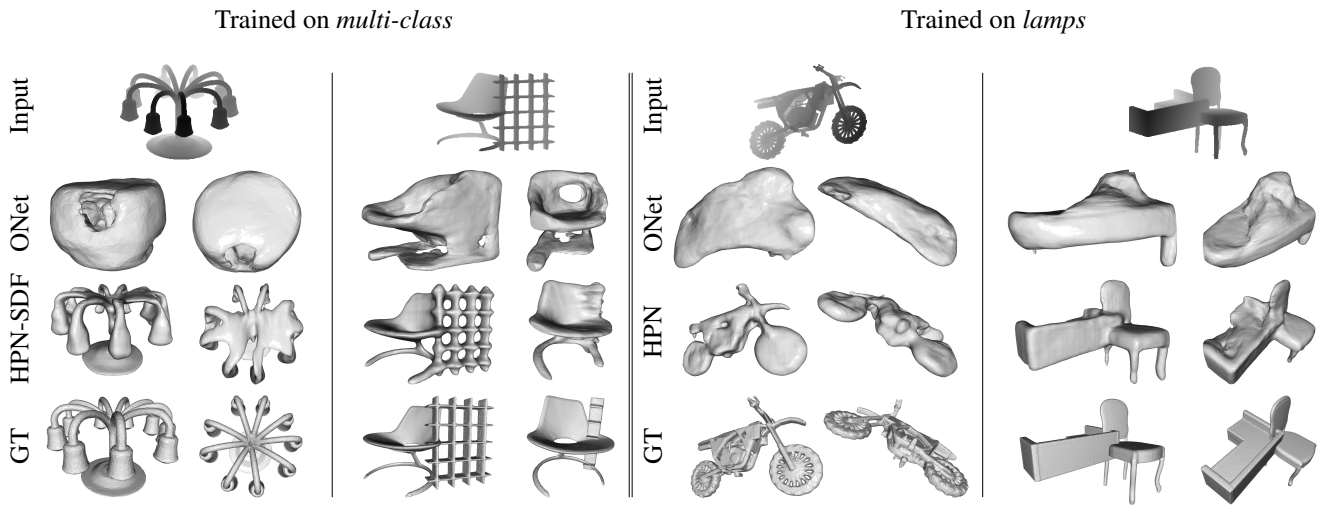

Figure 4. Reconstruction results for unseen classes in the different generalization settings. **Left:** Networks trained in the *multi-class* setting (on planes, cars and chairs). **Right:** Networks trained on lamps. More examples are provided in the Appendix.

		Chair		Lamp		Speaker		Sofa		Table		Mean (unseen)		Composition	
		F↑	CD↓	F↑	CD↓	F↑	CD↓	F↑	CD↓	F↑	CD↓	F↑	CD↓	F↑	CD↓
plane,car,chair	ONet [19]	40.8	4.1	18.8	9.3	38.4	6.0	43.2	4.7	35.2	5.3	29.3	6.8	18.3	8.7
	ONet-SDF [19]	35.9	4.6	19.9	8.5	37.6	5.8	38.3	5.1	33.0	5.6	28.6	6.6	19.3	8.0
	GenRe [38]	-	-	-	6.0*	-	7.7*	-	5.9*	-	5.7*	-	5.7*	-	-
	LDIF$_{svim1d}$ [11]	**62.1**	**0.9**	20.8	9.4	22.9	5.2	52.7	**1.3**	33.0	**3.3**	32.1	**3.5**	16.4	10.9
	HPN (ours)	44.3	3.8	38.4	4.8	**49.7**	**4.8**	46.6	4.5	43.7	4.4	42.9	4.9	30.2	5.7
	HPN-SDF (ours)	53.6	3.3	**56.5**	**3.5**	49.4	5.0	**54.4**	3.9	**53.1**	3.7	**48.2**	4.6	**42.4**	**3.9**
chair	ONet [19]	36.2	4.6	16.6	10.3	34.2	6.5	35.3	5.4	31.6	5.9	24.3	7.9	16.5	9.3
	LDIF$_{svim1d}$ [11]	**59.2**	**1.0**	17.8	10.6	21.6	5.6	**44.4**	**1.4**	31.4	**3.9**	27.7	**4.2**	14.9	13.0
	HPN (ours)	43.0	3.9	40.2	4.6	48.6	**4.8**	**44.4**	4.6	**44.2**	4.3	43.1	4.7	31.2	5.3
	HPN-SDF (ours)	41.2	4.2	**43.6**	**4.3**	**48.8**	5.0	43.8	5.0	**44.2**	4.5	**45.3**	4.7	**31.7**	**5.2**
lamp	ONet [19]	20.4	8.1	42.0	4.7	37.8	5.6	24.2	7.2	29.1	7.1	26.8	6.8	18.1	8.5
	LDIF$_{svim1d}$ [11]	12.4	12.2	48.1	**2.5**	21.6	5.1	11.8	7.4	17.1	10.5	21.1	5.6	12.5	14.0
	HPN (ours)	**42.4**	**4.7**	**50.3**	3.6	**53.2**	**4.6**	**45.2**	**5.0**	**47.1**	**4.7**	**47.1**	**4.7**	**35.8**	**5.0**
	HPN-SDF (ours)	41.1	4.8	48.4	3.6	51.5	**4.6**	44.7	**5.0**	44.8	4.8	46.1	4.8	33.9	5.2

Table 1. Comparison of the hierarchical prior network (HPN) to the state of the art in terms of generalization. The top part of the table shows training in the *multi-class* setting, the lower part shows training on a single class. We report two metrics: F-score (F, shown in %) and Chamfer distance (CD, multiplied by 100 for better readability). * denotes results taken from the original paper. *svim1d* denotes [11]'s data generation - a single perspective ground truth depth map. Results on categories seen during training are marked in blue. *Mean (unseen)* shows the average of per-class scores over all 13 unseen categories. *Composition* shows results on the composition of two objects per image. On compositions, HPN is more than twice as accurate as the state of the art and generally better on unseen classes, while LDIF is better on seen classes. See the supplemental for more results. Best viewed in color.

we empirically determine the new threshold $\tau_{sdf} = -0.02$. Therefore, we pick τ_{sdf} from the interval $[-1, 1]$ with a stepsize of 0.1 and a smaller stepsize of 0.01 in the interval $[-0.1, 0.1]$.

HPN and Local@N. As described in Sec. 3 we design local variants of the ONet and a fused variant for which different hierarchy levels are combined. In general, we refer to the fused variant as hierarchical prior networks (HPN) and to its local variants as Local@N where N is the width and height of a local patch in pixels, *e.g.*, Local@64 for patches of size 64×64 pixels. HPN is the fused version of Global@256, Local@64 and Local@32. HPN-SDF is the fused version of Global@256-SDF, Local@64 and Local@32, i.e. the SDF representation is used for the global but not for the local networks.

4.3. Setup

Training. All networks were trained using the ADAM optimizer [15] with the same optimization settings as used for the Occupancy Network [19]. We trained all networks until convergence. Similar to Occupancy Networks, during training, we only provide a randomly sampled subset of 1500 training points to our local networks.

Evaluation metrics. We report quantitative results for two widely used 3D reconstruction metrics: F-score [16] and Chamfer distance (CD) [2]. The two scores highlight different aspects of the reconstruction, as the F-score is robust to outliers (large deviations) and CD is not. We further discuss this point in Sec. 4.5. For completeness, we list the IoU values in the supplemental.

As part of our analysis, we additionally report F-score and Chamfer distance for the parts of the 3D shape that are visible from the input image and the parts that are invisible (self occluded) from the input image. In order to determine the visibility label, we project a set of points from the ground truth mesh into the depth image and check, whether they coincide with the respective depth value (visible) or are larger than the respective depth value (invisible). We do this for all test shapes, s.t. during evaluation we can look up the visibility label and compute the metrics separately.

Implementation. All the networks are implemented in PyTorch [24]. For visualizing qualitative examples, we used the Open3D [39] framework.

4.4. Results

Fig. 4 shows the drastically improved generalization to new shape classes and shape configurations compared to the state of the art. None of the networks has seen such categories during training, but thanks to the ability to flexibly recombine training parts, the hierarchical prior can also reconstruct completely new shapes in a reasonable quality. This also includes the composition of two objects, which was never observed during training. In contrast, the plain ONet model is bound to the most similar global shapes during training, which is insufficient in all these examples. Remembering the nice-looking reconstructions from literature, one should be aware that these were obtained via largely overlapping training and test sets.

Although the effect of the local recombination principle is already evident and indisputable from just the visual impression, Tbl. 1 also quantifies this effect. In all train-test configurations HPN outperforms the baselines and the previous state of the art in generalization. The performance almost doubles on unseen classes, both in terms of F-Score and Chamfer distance, in comparison to ONet. It also significantly improves over LDIF in terms of F-Score. For Chamfer distance, LDIF is competitive with HPN. We hypothesize that this happens because for some shapes our local networks produce outliers which have a large impact on the mean distance. Interestingly, LDIF represents the training classes better than all other methods but completely fails on compositional shapes. This indicates that LDIF is capable of nicely fitting the training data which is not useful when generalization is required. The use of signed distance functions yields more detailed reconstructions in conjunction with our hierarchical prior network (HPN-SDF), leading to best scores in the *multi-class* setting.

All approaches achieve consistently better scores on the unseen categories than on the new compositional test dataset. We conclude that the compositional setting is more difficult. One reason might be that one shape occludes the other, which requires to reconstruct the front side of the occluding shape (bookshelf), and the backside of the occluded shape (chair); see Fig. 4.

4.5. Analysis

4.5.1 Different hierarchy levels

We investigated the reconstruction by individual local networks and how they contribute to the full hierarchical reconstruction. Fig. 5 shows an example and Tbl. 2 reports test set scores on the full shape, as well as the visible and invisible parts of it. All models are trained on chairs and evaluated on the other categories. In visible areas, the local networks reconstruct details much better than the global network, which highlights the problem that global priors interfere with the measurements in these areas. Local networks with the smallest patch size (16 and 32) are particularly noisy in the invisible areas. Surprisingly, local models with larger patch size also perform a bit better (on average) in the invisible areas. This supports our recombination idea and indicates that explaining even the invisible shape regions with a collection of local priors may have advantages over using a single global one.

		Full		Visible		Invisible	
		F↑	CD↓	F↑	CD↓	F↑	CD↓
chair	Global@256 (ONet)	23.3	7.7	28.6	6.4	22.7	7.7
	Local@128	38.5	4.9	57.8	2.8	27.1	6.0
	Local@64	42.0	4.9	67.3	2.2	27.3	6.4
	Local@32	37.5	5.8	54.7	2.9	28.1	7.4
	Local@16	36.6	6.7	57.8	3.2	24.5	8.7
	HPN@(256+32)	35.7	5.2	48.8	3.3	28.3	6.2
	HPN@(256+64)	38.3	4.9	56.7	2.8	27.4	6.1
	HPN@(256+64+32)	39.7	4.7	57.6	2.6	29.8	5.7
	HPN@(256+128+64+32+16)	42.0	4.4	61.7	2.4	30.4	5.6

Table 2. Mean F-score (F) and Chamfer distance (CD) for multiple hierarchy levels trained on the category chair and evaluated on all unseen categories. We report the F-score and the CD for the full shape (Full), the visible and the invisible parts of it.

The best reconstruction scores are achieved when combining all available sources of information: HPN@(256+128+64+32+16) works better than any

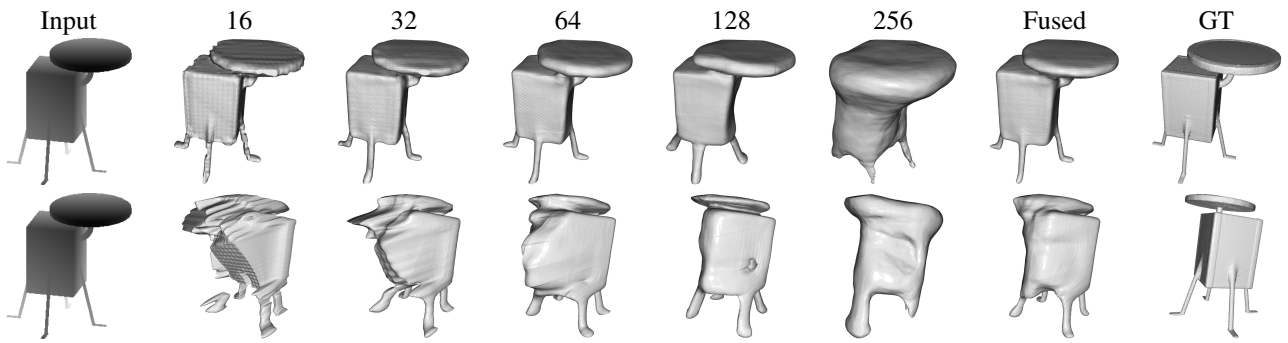

Figure 5. Reconstructions produced by different hierarchy levels. Numbers correspond to different patch sizes N. **Top row:** Same viewpoint. **Bottom row:** Opposite viewpoint. Reconstructions for very small patches are particularly noisy, since they see little context and the smaller overlap area reduces the spatial smoothing effect. However, due to the aggregation with other levels, this has no negative effect on the fused reconstruction.

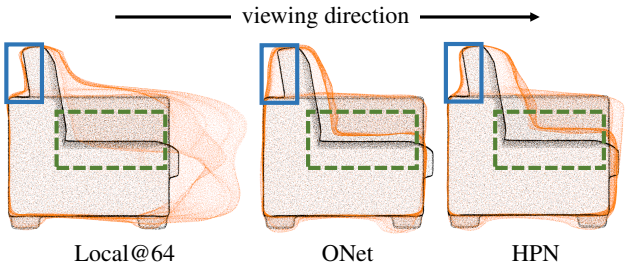

Figure 6. Point cloud of a couch (side-view). **Black:** Ground truth. **Orange:** Predicted shape. Marked boxes indicate regions where the local and global predictions differ significantly. Best viewed in color.

other configuration. This result is slightly unintuitive given that the most local reconstruction levels (32 and 16) on their own do not provide any quantitative improvement over the more global ones (128 and 64). We hypothesize that this can be attributed to the full set of networks acting like an ensemble, which averages out the errors of individual levels thus improving the final score. Note that more hierarchy levels also means higher computational cost. As a trade-off between efficiency and accuracy, in the rest of the paper we only use three levels, i.e. HPN@(256+64+32).

We can better understand the properties captured by individual quantitative metrics by looking at Fig. 6. The local reconstruction is very precise in the visible area (blue box) but completely wrong in the invisible part. The global reconstruction acts the other way around: it is off in the visible area but provides a plausible prior for the invisible part. The fused version gets much better on average - this is captured well by the CD. However, certain parts are still not perfect, e.g. the HPN reconstruction within the dashed green box is a bit off. Because of that, the F-score, being a robust metric, may not react so strongly to such changes. One should be aware of this when interpreting the quantitative results.

4.5.2 Data efficiency

Another expression of improved generalization is the required use of training data. Because the patch-based networks can effectively learn to recombine, hence reuse, parts seen during training to reconstruct novel shapes, we expect to need less training data for the patch-based networks to reach the same performance as the global variant. We evaluated this claim, comparing our Local@64 with the global ONet on the unseen categories of the multi-class setting for different amounts of training examples. Results are summarized in Fig. 7.

As expected, we see a significantly higher mean F-score for the patch-based network (bright orange) compared to the global network (dark blue) for all training dataset sizes. The local network reaches its full performance already with just 1% of the training data. Both networks converge for large amounts of training data. Two effects cause this data efficiency: (1) The local parts are less complex than a global shape, i.e., they require less data to be represented. (2) Each training sample comprises many local parts, which increases the effective training set size.

4.5.3 Failure cases

Fig. 8 shows some failure cases. Since the local reconstruction emphasizes the visible areas more, transfer of the global layout from examples seen during training is less pronounced than with the purely global baseline. Conversely, some details reconstructed correctly with the purely local network can be washed out due to the aggregation with the more global hierarchy levels.

4.6. Real data

In order to verify that our conclusions are not limited to the case of perfect ground truth input depth maps, we run the evaluation on selected depth maps from the ScanNet [8]

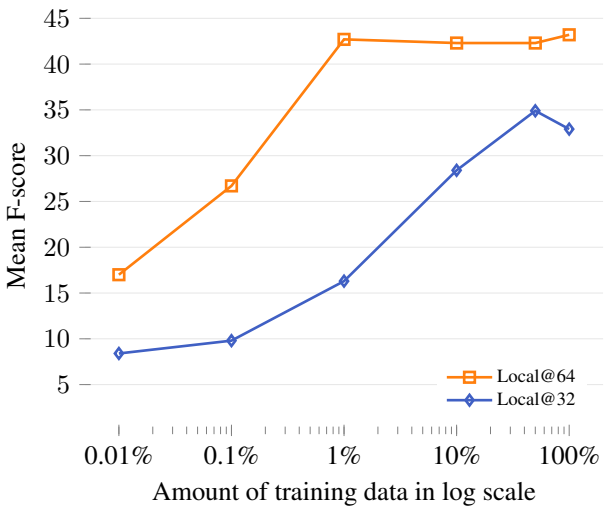

Figure 7. Reconstruction quality in dependence of the number of training samples. Local reconstruction reaches its full performance already with as little as 1% of the training data.

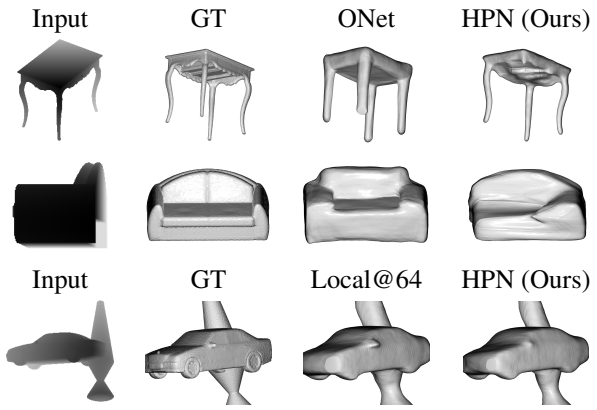

Figure 8. Failure cases of our approach. All models are trained on *multi-class* First row: ONet correctly reconstructs the leg of the table that is invisible in the input. HPN misses this leg. Second row: difficult view of a couch. ONet correctly reconstructs the invisible arm rest, while HPN does not. Third row: detailed reconstructions, like the mirror in the local reconstruction, can disappear due to global aggregation in HPN.

dataset. Both the baseline network and our approach were only trained in the synthetic multi-class setting.

Fig. 9 shows that HPN produces reasonable results both in case of multiple objects per scene and in case of a complex novel object from the statue class, although it has never seen noisy real-world data during training. In contrast, the ONet baseline only captures global blob-like structures.

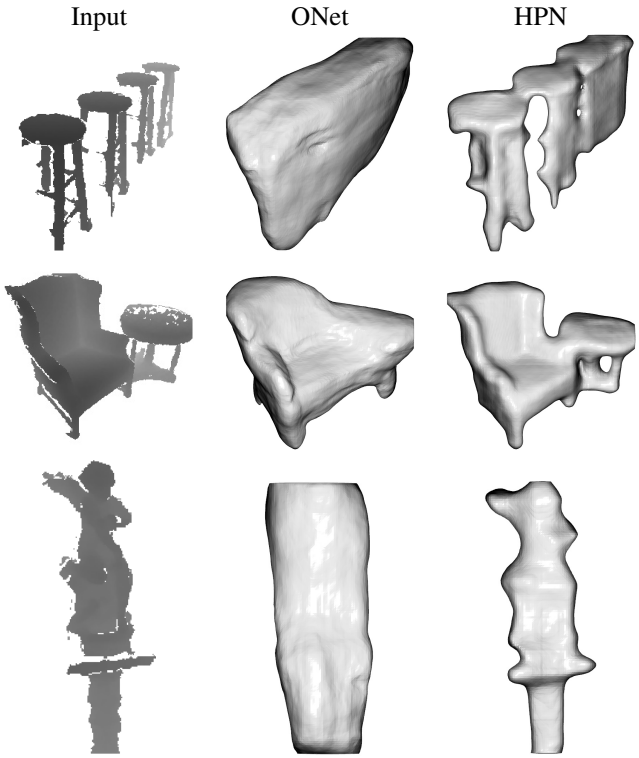

Figure 9. When provided with real-world noisy depth images from ScanNet, our multi-class-trained HPN yields more detailed reconstructions than the ONet baseline.

5. Conclusion

In this paper, we introduced a new paradigm for learning single-view reconstruction priors based on multiple locality levels. The decisive advantage of this paradigm over previous global reconstruction approaches is its ability to recombine local shapes. This recombination not only makes much more efficient use of training data, it also enables the generalization to completely unseen shapes or configurations of objects, which has been the key limitation of single-view object reconstruction to-date. Technically, the presented approach is simple yet flexible. While we used Occupancy Networks, any other network that implements an implicit function (even a retrieval approach) could replace that architecture. In this sense, the proposed hierarchy of local networks should not be regarded as an isolated network architecture, but rather as a working principle.

Acknowledgements: We thank Philipp Schroeppel and Evgeny Levinkov for feedback on the manuscript. We also thank Philipp for his help with the infrastructure.

References

[1] Abhishek Badki, Orazio Gallo, Jan Kautz, and Pradeep Sen. Meshlet priors for 3d mesh reconstruction. In *CVPR*, 2020. 2

[2] Harry G. Barrow, Jay M. Tenenbaum, Robert C. Bolles, and Helen C. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *IJCAI*, 1977. 6

[3] Miguel Ángel Bautista, Walter Talbott, Shuangfei Zhai, Nitish Srivastava, and Joshua M. Susskind. On the generalization of learning-based 3d reconstruction. *arXiv preprint arXiv:2006.15427*, 2020. 1, 2

[4] Rohan Chabra, Jan Eric Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard A. Newcombe. Deep local shapes: Learning local SDF priors for detailed 3d reconstruction. In *ECCV*, 2020. 2

[5] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An information-rich 3D model repository. *arXiv:1512.03012*, 2015. 2, 4

[6] Julian Chibane, Thiemo Alldieck, and Gerard Pons-Moll. Implicit functions in feature space for 3d shape reconstruction and completion. In *CVPR*, 2020. 2

[7] Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *ECCV*, 2016. 1, 2

[8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas A. Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 7

[9] Boyang Deng, Kyle Genova, Soroosh Yazdani, Sofien Bouaziz, Geoffrey E. Hinton, and Andrea Tagliasacchi. Cvxnet: Learnable convex decomposition. In *CVPR*, 2020. 2

[10] Haoqiang Fan, Hao Su, and Leonidas J. Guibas. A point set generation network for 3d object reconstruction from a single image. In *CVPR*, 2017. 1, 2

[11] Kyle Genova, Forrester Cole, Avneesh Sud, Aaron Sarna, and Thomas A. Funkhouser. Local deep implicit functions for 3d shape. In *CVPR*, 2020. 2, 4, 5

[12] Thibault Groueix, Matthew Fisher, Vladimir G. Kim, Bryan C. Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *CVPR*, 2018. 1, 2

[13] Chiyu Max Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, and Thomas A. Funkhouser. Local implicit grid representations for 3d scenes. In *CVPR*, 2020. 2

[14] Li Jun, Niu Chengjie, and Xu Kai. Learning part generation and assembly for structure-aware shape synthesis. In *AAAI*, 2020. 2

[15] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 6

[16] Arno Knapitsch, Jaesik Park, Qian-Yi Zhou, and Vladlen Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *ACM Transactions on Graphics*, 36(4), 2017. 6

[17] Jun Li, Kai Xu, Siddhartha Chaudhuri, Ersin Yumer, Hao (Richard) Zhang, and Leonidas J. Guibas. GRASS: generative recursive autoencoders for shape structures. *ACM Trans. Graph.*, 36(4):52:1–52:14, 2017. 2

[18] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. 3

[19] Lars M. Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *CVPR*, 2019. 1, 2, 4, 5, 6

[20] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *CVPR*, 2019. 2

[21] Chengjie Niu, Jun Li, and Kai Xu. Im2struct: Recovering 3d shape structure from a single rgb image. In *CVPR*, 2018. 2

[22] Despoina Paschalidou, Luc Van Gool, and Andreas Geiger. Learning unsupervised hierarchical part decomposition of 3d objects from a single rgb image. In *CVPR*, 2020. 2

[23] Despoina Paschalidou, Ali Osman Ulusoy, and Andreas Geiger. Superquadrics revisited: Learning 3d shape parsing beyond cuboids. In *CVPR*, 2019. 2

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019. 6

[25] Songyou Peng, Michael Niemeyer, Lars M. Mescheder, Marc Pollefeys, and Andreas Geiger. Convolutional occupancy networks. In *ECCV*, 2020. 2

[26] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 4

[27] Stephan R. Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *CVPR*, 2018. 1, 2

[28] Shunsuke Saito, Zeng Huang, Ryota Natsume, Shigeo Morishima, Hao Li, and Angjoo Kanazawa. Pifu: Pixel-aligned implicit function for high-resolution clothed human digitization. In *CVPR*, 2019. 2

[29] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *CVPR*, 2018. 2

[30] Maxim Tatarchenko, Alexey Dosovitskiy, and Thomas Brox. Octree generating networks: Efficient convolutional architectures for high-resolution 3d outputs. In *ICCV*, 2017. 1, 2

[31] Maxim Tatarchenko, Stephan R. Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *CVPR*, 2019. 2

[32] Anh Thai, Stefan Stojanov, Vijay Upadhya, and James M. Rehg. 3d reconstruction of novel object shapes from single images. *arXiv:2006.07752*, 2020. 1, 2

[33] Shubham Tulsiani, Hao Su, Leonidas J. Guibas, Alexei A. Efros, and Jitendra Malik. Learning shape abstractions by assembling volumetric primitives. In *CVPR*, 2017. 2

[34] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single RGB images. In *ECCV*, 2018. 1, 2

[35] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T. Freeman, and Joshua B. Tenenbaum. Learning 3D Shape Priors for Shape Completion and Reconstruction. In *ECCV*, 2018. 2

[36] Rundi Wu, Yixin Zhuang, Kai Xu, Hao Zhang, and Baoquan Chen. Pq-net: A generative part seq2seq network for 3d shapes. In *CVPR*, 2020. 2

[37] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomír Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. In *NeurIPS*, 2019. 2

[38] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *NeurIPS*, 2018. 1, 2, 4, 5

[39] Qian-Yi Zhou, Jaesik Park, and Vladlen Koltun. Open3D: A modern library for 3D data processing. *arXiv:1801.09847*, 2018. 6

[40] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *ICCV*, 2017. 2