A Simple and Scalable Shape Representation for 3D Reconstruction

Mateusz Michalkiewicz[1]
m.michalkiewicz@uq.net.au
Eugene Belilovsky[2]
eugene.belilovsky@umontreal.ca
Mahsa Baktashmotlagh[1]
m.baktashmotlagh@uq.edu.au
Anders Eriksson[1]
a.eriksson@uq.edu.au

[1] University of Queensland
Brisbane, Australia
[2] Mila, University of Montreal
Montreal, Canada

Abstract

Deep learning applied to the reconstruction of 3D shapes has seen growing interest. A popular approach to 3D reconstruction and generation in recent years has been the CNN encoder-decoder model usually applied in voxel space. However, this often scales very poorly with the resolution limiting the effectiveness of these models. Several sophisticated alternatives for decoding to 3D shapes have been proposed typically relying on complex deep learning architectures for the decoder model. In this work, we show that this additional complexity is not necessary, and that we can actually obtain high quality 3D reconstruction using a linear decoder, obtained from principal component analysis on the signed distance function (SDF) of the surface. This approach allows easily scaling to larger resolutions. We show in multiple experiments that our approach is competitive with state-of-the-art methods. It also allows the decoder to be fine-tuned on the target task using a loss designed specifically for SDF transforms, obtaining further gains.

1 Introduction

In recent years, we have witnessed an increased interest in extending the successes of deep learning to the analysis and representation of 3D shapes. This includes long standing problems, such as 3D shape reconstruction from single or multiple views [8, 32], shape from silhouettes [7], shape from contours [2], and shape completion [21]. Solutions to these problems can have a significant impact to applications in robotics [3], surgery [20], and augmented reality [14].

One of the preferred categories of models for tackling these problems is the CNN encoder-decoder architecture [8], popularized originally in the context of segmentation [5, 19]. For example, in the single view reconstruction task a 2D CNN will encode the 2-D image and a 3D CNN decoder model will produce the final representation in voxels. Standard decoders, however, are ineffective in larger resolutions and do not make full use of the structure of the object. Similar problems arise in more general attempts to learn latent variable models of 3D

shapes [11, 33]. Here, one may be interested in tasks such as unconditional generation and reconstruction.

More recently authors have considered alternative representations of shapes to a standard 3D discretized set of voxels [8, 12, 32, 34, 35], one that can permit more efficient learning and generation. These include point clouds [10], meshes [13, 31], and signed distance transform based representations [22, 23]. To date there is not an agreed upon canonical 3-D shape representation for use with deep learning models nor a canonical decoder architecture for use with any of the described shape representations. Indeed, many complex alternative decoder architectures have been used [25, 29]. In this work, we ask whether a very simple decoder architecture matched with the right shape representation can yield strong results. Building on the recent use of the Signed Distance Function (SDF) in shape representation we demonstrate a simple latent shape representation that can be used in downstream tasks and easily decoded. More specifically, in this work, we consider a latent shape representation obtained by applying PCA on the SDF transformed shape. We show this leads to a latent shape representation that can be used directly in downstream tasks like 3D shape reconstruction from a single view and 3D shape completion from a point cloud.

Our work a) reinforces the relevance of SDF as a representation for 3D deep learning; and b) demonstrates that a simple representation obtained by applying PCA on the SDF transform can lead to an effective latent shape representation. This representation allows for results competitive to state of the art in standard benchmarks. Our work also suggests more complex benchmarks than the current ones may be needed to push forward the study of learned 3D shape reconstruction.

The paper is structured as follows. In Sec. 2 we discuss the related work. We outline the basic methods used in the experiments in Sec. 3. We show extensive quantitative and experimental results comparing our approach to existing methods in Sec. 4.

2 Related Work

Several shape representations have been studied in the literature. Point cloud based representation requires a tedious step of sampling points from the surface and to generate shape subsequently inferring the continuous shape from a sample of points. Meshes present a challenge in that no clear way to generate valid meshes is available. Proposals have consisted of starting with template shapes and progressively deforming them Wang et al. [31]. This, however, can be problematic as it never explicitly represents the shape and may suffer issues with local coherence.

Deep Level Sets [22] and DeepSDF [23] also use the SDF representation as in our work. Unlike our method, Deep Level Sets still relies on an encoder-decoder CNN architecture thus not removing the desired computational constraints associated with the 3D shape modeling. DeepSDF attempts to directly fit a continuous function to each shape which gives the SDF representations. Despite avoiding discretization, this function can lead to a complex decoder model, e.g. an 8 layer network is used to fit the SDF. Another recent work [21], similar in spirit to Park et al. [23], learns a classifier to predict whether a point is inside or outside of the boundary, using this classifier as the shape representation. Different from our proposal, these methods cannot easily learn a latent shape representation to be applied on downstream task, since the shape is represented by the weights of the classifier or regression model. On the other hand, our latent representation can easily encode an unseen shape and be conveniently used as a prediction target for deep learning models.

Our work can also be seen as complementary to the very recent observations in Tatarchenko et al. [30] which highlights that good 3D single view reconstruction performance can be achieved by using retrieval or clustering methods. We note, however, that the descriptors used in that work are more complex.

PCA has been classically used to represent shapes in a variety of contexts. For example, classical methods in computer vision such as the active appearance model Edwards et al. [9] and the 3D morphable model used in face analysis Blanz et al. [1] are based on PCA shape representations. However, these typically are applied in a different context requiring transforming the shape to a reference set of points and applying PCA on the coordinates. Leventon et al. [17] used signed distance functions to embed 2D curves applying PCA to obtain statistical models. To the best of our knowledge it has not been combined with the SDF representing a surface in 3D. We note that level set methods and the SDF have only recently been revisited as an effective representation that can be combined with 3D deep learning [6, 22, 23]. Moreover, it is enlightening that this classic approach to shape representation can be competitive with deep learning methods on standard benchmarks.

3 Methods

In this section, we start with reviewing the SDF transform and then describe our simple yet effective approach to shape representation.

3.1 Signed Distance Functions

Consider a 3D shape and its closed surface $\Gamma \subset \mathbb{R}^3$. The *Signed Distance Function* (SDF) of Γ is a mapping $\phi : \mathbb{R}^3 \mapsto \mathbb{R}$ from any point $x \in \mathbb{R}^3$ to the surface:

$$\phi(x) = \pm \inf_{y \in \Gamma} \|x - y\|, \tag{1}$$

with the convention that $\phi(x)$ is positive on the interior and negative on the exterior of Γ.

In Michalkiewicz et al. [22] a CNN decoder model is used to predict the SDF representation from a latent space as well as to learn autoencoders. We note, however, that this representation is well structured and objects are often grouped by category, we thus ask if a much simpler linear and non-convolutional decoder model can be effective at capturing its variability, leading to the *eigenSDF* representation described in the next section.

The above paper [22] further considers a loss function for the SDF representation that approximately minimizes the point-wise distance:

$$L_\varepsilon(\theta) = \left(\sum_{x \in \Omega} \delta_\varepsilon(\tilde{\phi}^j(x)) d^j(x)^p \right)^{1/p} + \alpha \sum_{x \in \Omega} (\|\nabla \tilde{\phi}^j(x)\| - 1)^2 \tag{2}$$

with θ being parameters of the network, α a weighting factor, Ω an equidistant grid on which ϕ is evaluated, δ_ε approximated Dirac delta, $\tilde{\phi}$ inferred Signed Distance Function, and $d(x)$ the closest distance between grid point x and the ground truth shape. We will use this loss function to fine-tune our decoder model in the sequel.

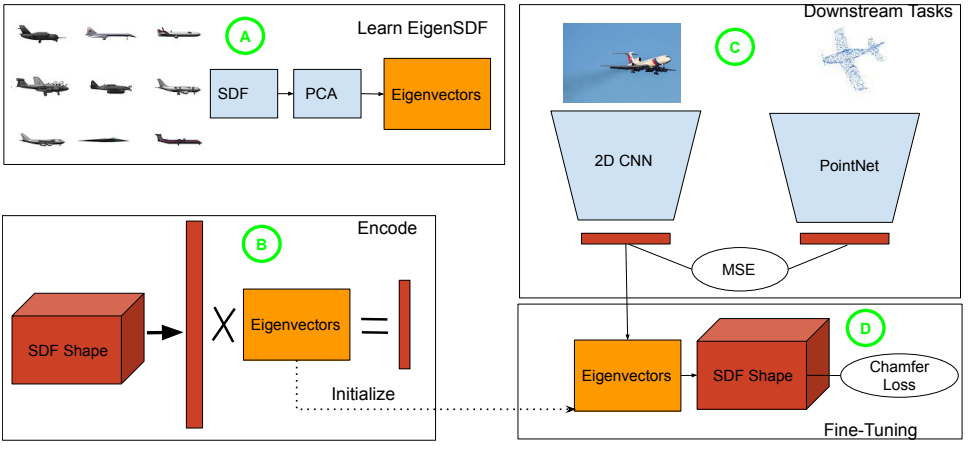

Figure 1: Overview of our experiments. We first apply PCA to all ShapeNet categories in order to retrieve eigenvectors (A). We encode every shape by applying eigenvectors to the signed distance function that is representing it (B). Our network for various experiments, *eigenSDF*, consists of an input encoder (2D CNN or PointNet) and a linear layer (C). It uses an ℓ_2 loss between its output and shape encodings from (B). We can directly decode predictions from the 2D CNN or PointNet using the eigenvectors. We can also further enhance performance, *eigenSDF (finetuned)*, by finetuning the eigenvectors with the loss function in Eq. 2 that is designed for SDF representation (D). Here, the weights of the decoder are initialized from eigenvectors obtained in (B).

3.2 EigenSDF

We apply the PCA transform to $\phi_{\text{all}} = \{\phi_i\}_{i=1..N}$, with N being the number of training examples. The eigenvectors E have the shape of (k, M^3) with M being the grid resolution and k being the number of used eigenvectors. We project each SDF ϕ to the latent representation ϕ_c using the eigenvectors E: $\phi_c = \phi E^T$. Here, ϕ_c has a shape of $(1,k)$. In the sequel, we will denote this representation as the *eigenSDF*. Note that applying PCA to the naive voxel representation would be inappropriate as the data is binary and therefore ill-suited for linear subspace methods such as PCA. For downstream tasks we predict directly the latent representation ϕ_c. We will also consider using E as an initialization which is finetuned by training on the SDF shape representation directly using Eq. 2. A high level overview of our framework is given in Figure 1.

4 Experiments

We evaluate the proposed representations on 3 tasks: i) 3D reconstruction; ii) 3D reconstruction from point cloud; and iii) 3D reconstruction with autoencoders. These applications are evaluated on 13 categories from the ShapeNet repository [4].

Preprocessing. In order to work on SDFs, we need to have a well defined interior and exterior of an object. We first preprocess the meshes to make them watertight using the method proposed in [27]. Following common practice, we render each ground truth mesh

into 24 2D views using equally spaced azimuth angles. For each ground truth mesh, we compute a corresponding SDF in a $128 \times 128 \times 128$ discretized voxel grid.

Metrics. Following the [21] experimental setup, we report 3 metrics. The first one is Intersection over Union (IoU), also known as Jaccard Index, between ground truth shape S and prediction $\tilde{S}$:

$$\text{IoU} = \frac{|S \cap \tilde{S}|}{|S \cup \tilde{S}|}.$$

The second metric measures point-wise distance between ground truth point set S_P and prediction $\tilde{S}_Q$ using the symmetric Chamfer distance:

$$\text{chamfer}(S_P, \tilde{S}_Q) = \frac{1}{2|P|} \sum_{p \in P} \min_{q \in Q} |p - q| + \frac{1}{2|Q|} \sum_{q \in Q} \min_{p \in P} |p - q|.$$

Finally, we measure the angular distance using normal consistency (nc) metric:

$$\text{nc}(S_P, \tilde{S}_Q) = \frac{1}{2|P|} \sum_{p \in P} |N_{S_P}(p) \cdot N_{\tilde{S}_Q}(n_{\tilde{S}_Q}(p))| + \frac{1}{2|Q|} \sum_{q \in Q} |N_{\tilde{S}_Q}(q) \cdot N_{S_P}(n_{S_P}(q))|,$$

where $N_S(p)$ denotes normal of point p lying on surface S and $n_S(q)$ denotes nearest neighbour of point q lying on surface S.

4.1 3D Reconstruction from Single 2D View

In this set of experiments, we evaluate the *eigenSDF* approach described in Sec 3. We perform PCA jointly on all categories using a starting resolution of $128 \times 128 \times 128$. For memory efficiency, we use incremental PCA [26]. k eigenvectors were chosen to capture at least 99.5% of the variance within the dataset. The image encoder is a 2D CNN whose architecture is taken from [25]. We minimize the ℓ_2 loss between the SDF projected into the latent space ϕ_c, and the prediction of the 2D CNN. This network is trained for 100 epochs using an ADAM [15] optimizer. Initial learning rate was set to 10^{-3} and dropped at epoch 30 to 10^{-4}. Furthermore, we consider finetuning the representation starting with the eigenvectors from PCA and using Eq 2. This baseline is referred to as *eigenSDF (finetuned)*.

In order to demonstrate that a gain is made by PCA versus just architecture, we also train a linear autoencoder of the same size ($M \times k$) and finetune it with Eq 2. This baseline is referred to as *linearSDF* and *linearSDF(finetuned)*.

Finally, we compare to a set of standard benchmarks from the recent literature including voxel based CNN encoder-decoder [8], point cloud based methods [10], a mesh based method [31], and the recently introduced ONet [21].

Complete results are given in Table 1. First, we observe that simply using a same sized linear model *linearSDF (finetuned)* is outperformed by using the *eigenSDF*. Compared to alternatives, our method gives more significant gains in Chamfer metric than all competitors and can be further improved with the finetuning. We also observe improvements in the normal consistency metric. For the IoU metric, we observe that *eigenSDF* outperforms all methods except Mescheder et al. [21]. Note that according to Sun et al. [28] Chamfer distance is a far better metric for shape comparison than IoU.

Chamfer↓									
Cat name	3D R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	linearSDF	linearSDF (ft)	eigenSDF	eigenSDF (ft)
airplane	0.227	0.137	0.187	0.104	0.147	0.262	0.253	0.093	**0.078**
bench	0.194	0.181	0.201	0.138	0.155	0.255	0.243	0.091	**0.076**
cabinet	0.217	0.215	0.196	0.175	0.167	0.229	0.222	0.077	**0.062**
car	0.213	0.169	0.180	0.141	0.159	0.233	0.230	0.068	**0.055**
chair	0.270	0.247	0.265	0.209	0.228	0.269	0.262	0.113	**0.095**
display	0.314	0.284	0.239	0.198	0.278	0.285	0.278	0.112	**0.110**
lamp	0.778	0.314	0.308	0.305	0.479	0.642	0.627	0.469	**0.388**
loudspeaker	0.318	0.316	0.285	0.245	0.300	0.261	0.255	0.101	**0.095**
rifle	0.183	0.134	0.164	0.115	0.141	0.291	0.271	0.141	**0.139**
sofa	0.229	0.224	0.212	0.177	0.194	0.275	0.268	0.192	**0.137**
table	0.239	0.222	0.218	0.190	0.189	0.207	0.196	0.124	**0.111**
telephone	0.195	0.161	0.149	0.128	0.140	0.175	0.170	0.051	**0.047**
vessel	0.238	0.188	0.212	0.151	0.218	0.425	0.414	0.351	**0.339**
mean	0.278	0.215	0.216	0.175	0.215	0.292	0.283	0.152	**0.133**
IoU↑									
Cat name	3D R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	linearSDF	linearSDF (ft)	eigenSDF	eigenSDF (ft)
airplane	0.426	-	0.420	-	**0.571**	0.421	0.432	0.524	0.541
bench	0.373	-	0.323	-	**0.485**	0.368	0.378	0.372	0.393
cabinet	0.667	-	0.664	-	**0.733**	0.655	0.667	0.688	0.703
car	0.661	-	0.552	-	**0.737**	0.666	0.679	0.716	0.732
chair	0.439	-	0.396	-	**0.501**	0.382	0.401	0.401	0.412
display	0.440	-	**0.490**	-	0.471	0.385	0.397	0.431	0.439
lamp	0.281	-	0.323	-	**0.371**	0.208	0.215	0.234	0.275
loudspeaker	0.611	-	0.599	-	**0.647**	0.558	0.566	0.596	0.606
rifle	0.375	-	0.402	-	**0.474**	0.259	0.265	0.392	0.395
sofa	0.626	-	0.613	-	**0.680**	0.606	0.621	0.624	0.639
table	0.420	-	0.395	-	**0.506**	0.393	0.399	0.419	0.430
telephone	0.611	-	0.661	-	**0.720**	0.588	0.611	0.680	0.714
vessel	0.482	-	0.397	-	**0.530**	0.447	0.451	0.476	0.501
mean	0.493	-	0.480	-	0.571	0.456	0.467	0.504	0.521
Normal Consistency↑									
Cat name	3D R2N2	PSGN	Pix2Mesh	AtlasNet	ONet	linearSDF	linearSDF (ft)	eigenSDF	eigenSDF (ft)
airplane	0.629	-	0.759	0.836	**0.840**	0.707	0.715	0.819	0.822
bench	0.678	-	0.732	0.779	0.813	0.748	0.763	0.817	**0.828**
cabinet	0.782	-	0.834	0.850	0.879	0.773	0.777	0.885	**0.889**
car	0.714	-	0.756	0.836	0.852	0.781	0.799	0.874	**0.878**
chair	0.663	-	0.746	0.791	0.823	0.751	0.772	0.815	**0.827**
display	0.720	-	0.830	0.858	0.854	0.750	0.781	0.870	**0.877**
lamp	0.560	-	0.666	0.694	0.731	0.579	0.585	0.783	**0.792**
loudspeaker	0.711	-	0.782	0.825	0.832	0.733	0.749	0.855	**0.862**
rifle	0.670	-	0.718	0.725	0.766	0.661	0.669	0.816	**0.819**
sofa	0.731	-	0.820	0.840	**0.863**	0.738	0.740	0.855	0.861
table	0.732	-	0.784	0.832	**0.858**	0.722	0.729	0.804	0.811
telephone	0.817	-	0.907	0.923	0.935	0.830	0.852	0.921	**0.936**
vessel	0.629	-	0.699	0.756	0.794	0.700	0.733	0.815	**0.817**
mean	0.695	-	0.772	0.811	0.834	0.728	0.743	0.840	**0.847**

Table 1: Single View 3D Reconstruction Results on ShapeNet. We observe that our *eigenSDF* approach outperforms other state-of-the-art learning based methods in normal consistency and Chamfer distance. Finetuning can further improve this result. Compared to training a linear autoencoder or just finetuning the performance is substantially better, showing that eigen decomposition obtains the best results.

4.2 3D Shape Completion from Point Clouds

We next consider shape completion from a point cloud. This task has been studied in [18, 21]. Similar to experimental setup of [21], we use 13 categories from ShapeNet repository and we pre-process the meshes to make them watertight. We randomly sample 300 points from ground truth meshes and add a Gaussian noise with 0 mean and 0.05 standard deviation. The same metrics have been used as described in section 4.1.

We have encoded the input point cloud with PointNet encoder with a bottleneck dimension of 512 [24] and decoded it with linear decoder from section 4.1. A similar set of baselines has been used as in the previous section and compared to *eigenSDF*. We observe similar large gains in the Chamfer metric and competitive performance in other metrics. Results in Table 2 show that, similar to 3D reconstruction task, our performance is much better in Chamfer distance, similar in normal consistency, and the second best in IoU.

method	IoU↑	Chamfer ↓	nc ↑
eigenSDF (ours)	0.568	**0.077**	0.852
3D-R2N2 ([8])	0.565	0.169	0.719
PSGN ([10])	-	0.144	-
DMC ([18])	0.674	0.117	0.848
ONet ([21])	**0.778**	0.079	**0.895**

Table 2: Results on 3D shape completion

4.3 3D Reconstruction from Latents

Finally, we consider a simple 3D reconstruction task [12]. This can also be viewed as measuring the representational power of the model [21]. We evaluate reconstruction quality of *eigenSDF* versus other methods, particularly CNN-based autoencoders. The goal is to reconstruct test set shapes. An initial resolution of $128 \times 128 \times 128$ was used and reduced to $k = 512$ as done in other works [21]. We use *cars* category from ShapeNet repository and evaluate reconstruction on unseen data. For the evaluations, in addition to the metrics used in section 4.1, we further analyse the decoders using F-score [16]. Results are shown in Table 3.

method	IoU↑	Chamfer↓	NC↑	F-score↑
eigenSDF	0.746	0.0425	0.869	0.484
eigenSDF (ft)	**0.758**	**0.0325**	**0.896**	**0.529**
Linear (ϕ) (chamfer)	0.582	0.050	0.773	0.315
Linear(voxels)	0.637	0.067	0.737	0.384
DLS ([22])	0.681	0.047	0.858	0.103
TL ([12])	0.656	0.082	0.847	0.081

Table 3: We compare *eigenSDF* to the state-of-the-art methods in terms of reconstruction. We find that *eigenSDF* performs better than linear autoencoders trained on voxels or SDFs.

4.4 Comparison with Deep Level Sets

In this section, we compare our method to the other recent approach relying on the signed distance transform [22] and learning with the chamfer loss L_ε. This one, however, uses the CNN decoder model and does not learn a latent shape representation. We have chosen a similar experimental setup of 3 subsets each having 2 000 examples from ShapeNet repository: *cars*, *sofas*, *chairs*. We observed that remaining 2 categories, *bottles* and *phones*, are too simple to allow for a difference in higher resolution.

We compare the training time for both methods for various resolutions. The results shown in Figure 2 demonstrate that, as opposed to *eigenSDF*, it is not feasible to use the CNN-based decoder in higher resolutions. It is consistent with the findings of [25]. Note

that *eigenSDF (ft)* is first trained in latent space and then shortly finetuned which also significantly shortens convergence time. Quantitative comparison between *eigenSDF (ft)* and *DLS* is shown in Table 4.

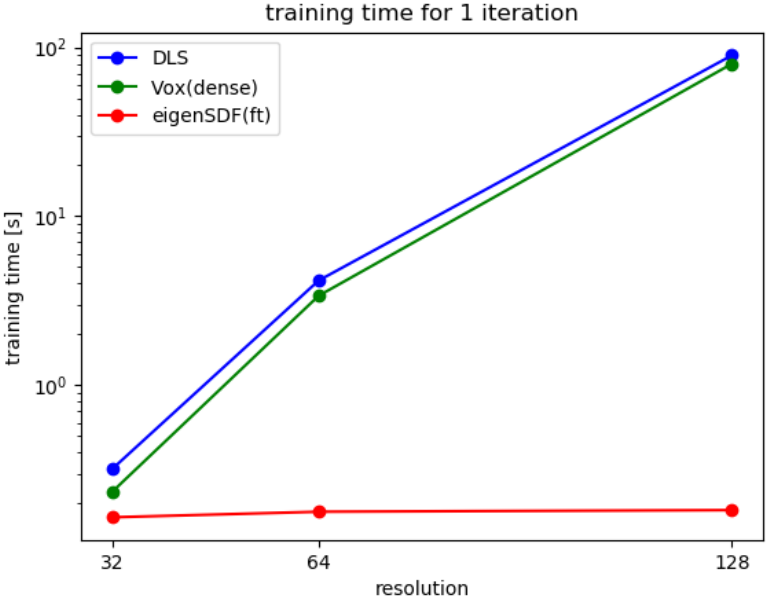

Figure 2: Training time of *eigenSDF (finetuned)* and dense convnets (TL [12], DLS [22]). Figure shows on a logscale amount of seconds a network needs for forward and backward pass of 1 iteration using a batchsize of 32. A single linear fully-connected layer scales much better in the output size compared to a 3D CNN decoder. Note that on single GPU standard CNN-based decoders can take weeks or even months to train using a higher resolution if they can fit into memory. Resolution 256 not shown due to clarity.

4.5 Reconstruction and Generation

Finally, we evaluate the performance on the single view reconstruction qualitatively. In Figure 5, we can see that reconstructions (on unseen data) can be effective capturing more complex structures ignored by [22]. We further compare reconstructions when limiting the output resolution of proposed method to the one used in DLS (see Figure 4).

Multiple authors have also consider generating unconditionally shapes, typically using sophisticated non-linear deep learning models like GANs and VAEs. We compare some of these to sampling a gaussian in the latent space of the *eigenSDF*. Qualitative results are shown in Figure 3. As it can be seen, our simple approach yields comparable shape representation to the complex non-linear models.

5 Conclusion

We have shown that using a simple linear decoder coupled with the SDF representation yields competitive results. The SDF lends itself effectively to the application of PCA yielding a

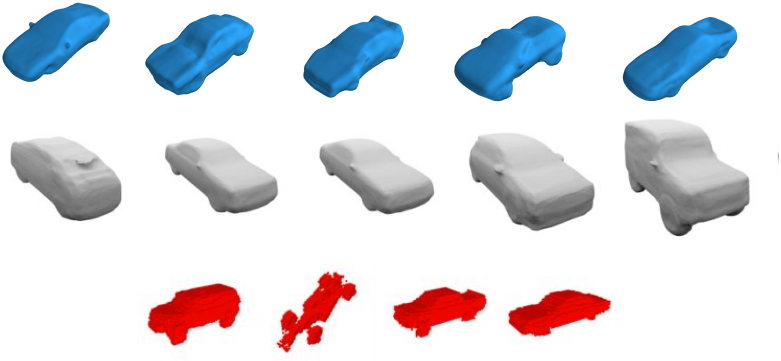

Figure 3: We compare unconditional generations of *cars* category. Generations from a gaussian fit to *eigenSDF* is shown in the top row (blue). Second row are generations from [21] and the third row is from a 3D GAN [33].

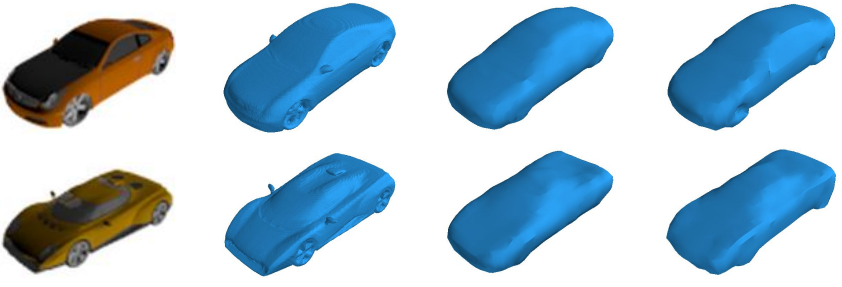

Figure 4: We compare reconstructions of *eigenSDF* and the CNN decoder based Deep Level Sets [22], which also uses SDF representation, at low resolution (32^3).

strong but simple baseline for future work in learned 3D shape analysis. Moreover, our work suggests that more complex baseline datasets may be needed to further evaluate deep learning methods on 3D shape inference.

Acknowledgements. This work has been funded by the Australian Research Council through grant FT170100072. EB is funded by IVADO. Authors would like to thank Stavros Tsogkas and Ming Xu for constructive comments.

Figure 5: We compare reconstructions of *eigenSDF* and the CNN decoder based Deep Level Sets [22] without limiting *eigenSDF* to low resolution as it allows us to operate at a higher resolution and generally produces more locally coherent results.

category	DLS				eigenSDF(ft)			
	IoU↑	Chamf↓	NC↑	F-score↑	IoU↑	Chamf↓	NC↑	F-score↑
cars	0.784	0.055	0.804	0.148	**0.821**	**0.040**	**0.909**	**0.432**
chairs	0.434	0.360	0.743	0.066	**0.553**	**0.125**	**0.820**	**0.168**
sofas	0.581	0.132	0.779	0.089	**0.647**	**0.082**	**0.867**	**0.248**

Table 4: Comparison to the SDF based method [22] in single view reconstruction. There is marked improvement due to the ability to model higher resolution.

References

[1] Volker Blanz, Thomas Vetter, et al. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, volume 99, pages 187–194, 1999.

[2] Michael Brady and Alan Yuille. An extremum principle for shape from contour. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 288–301, 1984.

[3] Erik Bylow, Jürgen Sturm, Christian Kerl, Fredrik Kahl, and Daniel Cremers. Real-time camera tracking and 3d reconstruction using signed distance functions. In *Robotics: Science and Systems*, 2013.

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

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017.

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

[7] KMG Cheung, Simon Baker, and Takeo Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.* IEEE, 2003.

[8] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[9] Gareth J Edwards, Christopher J Taylor, and Timothy F Cootes. Interpreting face images using active appearance models. In *Proceedings Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 300–305. IEEE, 1998.

[10] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, page 6, 2017.

[11] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3d shape induction from 2d views of multiple objects. In *2017 International Conference on 3D Vision (3DV)*, pages 402–411. IEEE, 2017.

[12] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *European Conference on Computer Vision*, pages 484–499. Springer, 2016.

[13] Georgia Gkioxari, Jitendra Malik, and Justin Johnson. Mesh r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9785–9795, 2019.

[14] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, pages 559–568, 2011.

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[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.

[17] Michael E Leventon, W Eric L Grimson, and Olivier Faugeras. Statistical shape influence in geodesic active contours. In *5th IEEE EMBS International Summer School on Biomedical Imaging, 2002.*, pages 8–pp. IEEE, 2002.

[18] Yiyi Liao, Simon Donné, and Andreas Geiger. Deep marching cubes: Learning explicit surface representations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2916–2925, 2018.

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[20] Andres Marmol, Artur Banach, and Thierry Peynot. Dense-arthroslam: Dense intra-articular 3-d reconstruction with robust localization prior for arthroscopy. *IEEE Robotics and Automation Letters*, 4(2):918–925, 2019.

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

[22] Mateusz Michalkiewicz, Jhony K. Pontes, Dominic Jack, Mahsa Baktashmotlagh, and Anders P. Eriksson. Deep level sets: Implicit surface representations for 3d shape inference. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2019.

[23] 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 Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019.

[24] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[25] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1936–1944, 2018.

[26] David A Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *International Journal of Computer Vision*, 77(1-3): 125–141, 2008.

[27] David Stutz and Andreas Geiger. Learning 3d shape completion from laser scan data with weak supervision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE Computer Society, 2018.

[28] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[29] 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.

[30] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019.

[31] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[32] Jiajun Wu, Chengkai Zhang, Tianfan Xue, Bill Freeman, and Josh Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[33] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T Freeman, and Joshua B Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *Advances in Neural Information Processing Systems*, pages 82–90, 2016.

[34] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems*, pages 1696–1704, 2016.

[35] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.