IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 13 March 2023, accepted 28 April 2023, date of publication 11 May 2023, date of current version 17 May 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3275442

APPLIED RESEARCH

DC-DFFN: Densely Connected Deep Feature Fusion Network With Sign Agnostic Learning for Implicit Shape Representation

ABOL BASHER AND JANI BOUTELLIER
School of Technology and Innovation, University of Vaasa, 65200 Vaasa, Finland

Corresponding authors: Abol Basher (abol.basher@uwasa.fi) and Jani Boutellier (jani.boutellier@uwasa.fi)

This work was supported in part by the Academy of Finland Projects Compact and Efficient Deep Neural Networks for Ubiquitous Computer Vision (CoEfNet) and Robust and Efficient Perception for Autonomous Things (REPEAT).

ABSTRACT Reconstructing 3D surfaces from raw point cloud data is still a challenging and complex problem in computer vision and graphics. Recently emerged neural implicit representations model 3D surfaces implicitly in arbitrary resolution and diverse topologies. In this domain, most of the studies have so far used a single latent code-based variational auto-encoder (VAE) or auto-decoder (AD) architectures, or architectures similar to UNets. Due to the deep architectures of the existing approaches, gradients and/or input information can vanish while passing through the layers, which can cause suboptimal learning at training time and consequently low performance at test time. As a countermeasure, skip connections and feature fusion have been used in related application fields of convolutional neural networks. In this study, we embrace this idea and propose a novel densely connected deep feature fusion network, DC-DFFN, architecture for implicit shape representation. In the experimental results we show that DC-DFFN outperforms baseline approaches in terms visual reconstruction quality and quantitatively based on several measures. In addition, the proposed approach provides faster convergence during training compared to the baseline approaches. The DC-DFFN architecture has been implemented in PyTorch and is available as open source.

INDEX TERMS Convolutional neural network, implicit representation, dense feature fusion, zero-label set, surface reconstruction, ShapeNet, D-Faust.

I. INTRODUCTION

Recent advances in learning-based data driven approaches [1], [2], [3], [4], [5], [6], [7], [8] for reconstructing surfaces from raw un-oriented point clouds, and triangle soups are showing huge potential for several practical application fields, for example AR/VR technology, 3D printing, computer-aided design, and robotics. Recently emerged neural function-based implicit representations [1], [2], [3], [5], [8], [9], [10] can reconstruct a surface with infinite resolution and arbitrary topology compared to classical 3D presentations such as voxels, octrees, point clouds, and meshes, which have various ingrained issues. For example, voxel-based representations have problems related to resolution (memory requirement increases cubically with resolution), whereas point cloud-based representations do not have connectivity among the points, and meshes can have self-intersection issue [9], [11] and are restricted to a fixed topology.

The recently emerged implicit representations of 3D visual data to a great extent solve the problems related to classical representations, but pose new challenges related to complexity and computation time of the involved neural networks. Moreover, most of the implicit representation-based surface reconstruction works [3], [4], [5], [6], [7], [8], [9] focus only on reconstruction quality, use of novel activation functions, and optimization methods, hardly paying attention on network size [12] or training and inference time. In this study, we propose a densely connected feature fusion-based encoder-decoder neural architecture to ensure maximum input information flow through the network for better

The associate editor coordinating the review of this manuscript and approving it for publication was K. C. Santosh.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

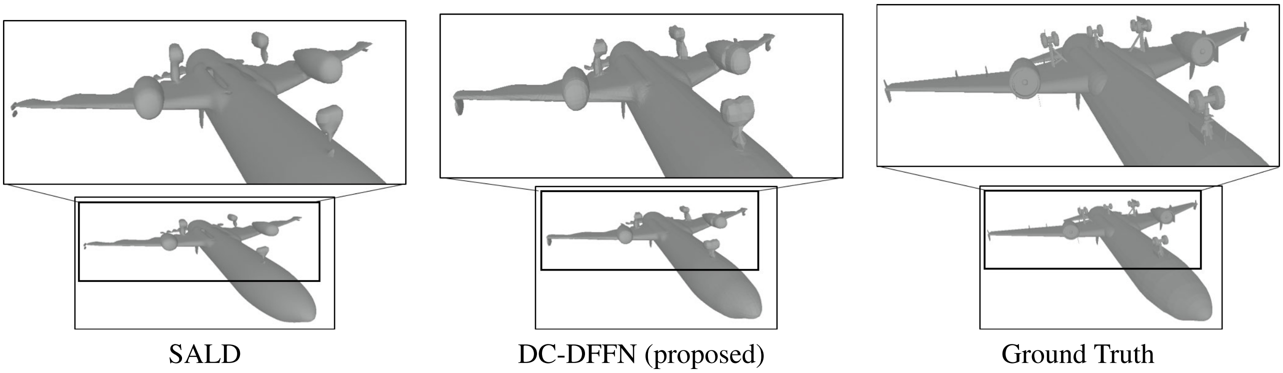


FIGURE 1. Illustration of the proposed architecture's improved capability of capturing small detail, compared to the state-of-of-the art work SALD [1]. Zooming in reveals that SALD causes wing and wheel parts to become disconnected, and renders extra holes to the airplane's hull.

learning and consequently, more expressive reconstruction quality.

In deep neural architectures, the gradients or input information can vanish or wash out while passing through the layers, which has already been addressed in several works of 2D image domain [13], [14], [15] applications, such as object recognition, object detection, and localization. So far, according to the best knowledge of the authors, none of the previous studies have considered dense features simultaneously to learn the surface and address vanishing gradient problem in the network architectures used for neural implicit representations. Besides mitigating the vanishing gradient problem, fusing features densely will allow the model to learn low (blob, edge) and high (object) level features simultaneously, when being extracted from different layers of the network. To this end, we propose a dense feature fusion-based encoder-decoder network architecture to achieve high fidelity surface reconstruction.

The main contributions of this study are:

- A novel deep feature fusion-based variational autoencoder architecture[1] for implicit surface reconstruction, which
- Ensures better learning and shorter convergence time due to improved information flow through the network, and
- Better robustness and generalization to unseen shapes, and
- Reconstruction of high fidelity continuous surfaces and obtaining state-of-the-art quantitative results.

The proposed study is a natural extension of previous approaches [1], [8], however transformed into a densely connected feature fusion-based network architecture. The rest of this paper is organized in the following manner: recently proposed related studies on explicit and implicit image representations are discussed in Section II; the proposed densely connected deep feature fusion-based encoder and decoder architectures are illustrated in Section III; qualitative and quantitative comparisons with the baseline approaches are shown in Section IV, and Section VI concludes the paper.

II. RELATED WORK

3D surface reconstruction approaches can be categorised based on their inherent ways of representing visual data: (a) explicit representations or classical representations, such as voxels, point clouds, and meshes, (b) implicit representations. In this section, we review traditional analytic priors-based reconstruction methods, classical and implicit representation-based approaches. Additionally, we review a few feature fusion-based studies from the 2D image domain to illustrate commonly used strategies for constructing efficient network architectures, which are the backbone of this study.

A. TRADITIONAL RECONSTRUCTION APPROACHES

There are a number of existing methods that are based on analytic priors for surface reconstruction, for example: Screened Poisson Surface Reconstruction (SPSR) [16], Moving Least Squares (MLS) [17], Ball Pivoting Algorithm (BPA) [18] and Radial Basis Functions (RBF) [19]. SPSR was developed on top of the previously proposed Poisson Surface Reconstruction (PSR) algorithm [20], which works based on global surface smoothness priors, and addresses the limitations associated with PSR, for example tendency of over-smoothing the data. This approach casts the surface reconstruction task as a spatial Poisson problem and performs reconstruction in the frequency domain [21]. However, SPSR requires oriented normals of the input points. Similar to SPSR, RBF also works based on global surface smoothness priors and performs the reconstruction using radially symmetric basis functions. MLS is a mesh-independent approach of surface interpolation. It works considering surfaces in differential geometry, which includes a local mapping function and a local reference system for each points of the surface, and uses the moving least squares concept. BPA, on the other hand, reconstructs the surface through computing triangles by interpolating a given point cloud. In BPA, considering a triangle formation of three points from a point cloud, a sphere

[1] source code available: https://github.com/basher8488881/DC-DFFN

with a predefined radius is rotated around the edges until it touches another point.

B. CLASSICAL REPRESENTATIONS OF 3D VISUAL DATA

Voxel-based representations are one of the most popular and earliest representations for learning-based 3D reconstruction of shapes and scenes [22], [23], [24]. In this representation, the 3D space is discretized into a regular grid, making it an intuitive extension for learning-based algorithms that have been developed for the 2D image domain, such as deep (convolutional) neural networks. In its simplest form, voxels can be used to learn the dense occupancy grid (where each voxel is occupied or unoccupied), and utilize this information to render a mesh surface [2], [5], [25]. However, due to cubically increasing memory requirements and lacking fidelity of rendered shapes, the usage of occupancy grids is limited to specific use cases [26], [27], [28].

Point clouds are another classical 3D data representation that expresses the 3D visual information by sparse data points and provide several advantages over voxels, for example their capability of better representing large spaces with fine details. Point clouds also serve as one possible output representation of implicit surface modeling [3]. Drawbacks of point clouds include lack of connectivity information, and high memory footprint of large surfaces that need to be represented densely.

In contrast, the mesh representation bears more information than point cloud-based representations by expressing connectivity among 3D points. The vertices and faces of a mesh can be directly regressed using a neural network [29], [30]. Meshes have a wide range of applications, for example in classification and segmentation [31], [32], [33]. More recently, mesh-based representations have also been used as the output representation for implicit 3D surface reconstruction [1], [5], [8], [9], [10], [12].

C. IMPLICIT REPRESENTATIONS

Recently emerged implicit representations express the 3D surface S implicitly using (zero) label sets (Equation 1),

$$S = \{x \in \mathbb{R}^3 | f(x; w) = 0\} \tag{1}$$

of a neural function $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$, where $x \in \mathbb{R}^3$ is the input data (sampled from a point cloud or triangle soup, $\mathcal{X} \in \mathbb{R}^3$), and w are the neural network weights that approximate the surface to $\mathcal{X}$. There are mainly two types of supervised approaches [1] commonly used to train the neural network to become an implicit function representation: (I) regression of known or pre-computed occupancy values $f(\text{p}, \text{z}) : \mathbb{R}^3 \times \mathbb{Z} \longrightarrow [1, 0]$ using an occupancy function [5], [9] or signed distances $f(\text{p}, \text{z}) : \mathbb{R}^3 \times \mathbb{Z} \longrightarrow \mathbb{R}$ using a signed distance function [1], [8], [10] or unsigned distances $f(\text{p}, \text{z}) : \mathbb{R}^3 \times \mathbb{Z} \longrightarrow \mathbb{R}_0^+$ using an unsigned distance function [3], [34], and (II) regression of raw 3D data using sign agnostic losses [1], [8] by relating points on the level sets to the neural network model parameters [35] or supervision with partial differential equations approximating the signed distance functions [36]. In this study, we adopted the second the approach of training the proposed network using a sign agnostic loss function [1], [8]. Our proposed network outperforms the recently proposed state-of-the-art method SALD [1] and shows that dense connections in the network provide improved information flow through the layers and faster convergence, consequently generating high fidelity shapes that preserve small details. In addition to proposing dense connectivity and feature fusion, our architecture employs 1D convolutional layers with 1 × 1 kernels that generalize better on complex shapes than the dense layers used by previous works [1], [8], [36].

D. FEATURE FUSION NETWORK ARCHITECTURES

Feature fusion and skip connections are used to enhance the performance of (convolutional) neural network models by mitigating the vanishing gradient problem in deep networks [13], [14], [15], [37], [38], [39]. In this concept, features from previous layer(s) are fused in the next layer(s) either by performing summation or concatenation. For example in the ResNet architecture [14] the previous layer features were simply added to the next layer's output. In contrast, in the DenseNet architecture [13], all features of the preceding layers are concatenated in the next layer's output. Finally, attention-based feature fusion [40] fuses point-wise features and local features to compensate the loss caused by order-invariant max-pooling on point clouds, and to improve the 3D semantic segmentation accuracy of point clouds.

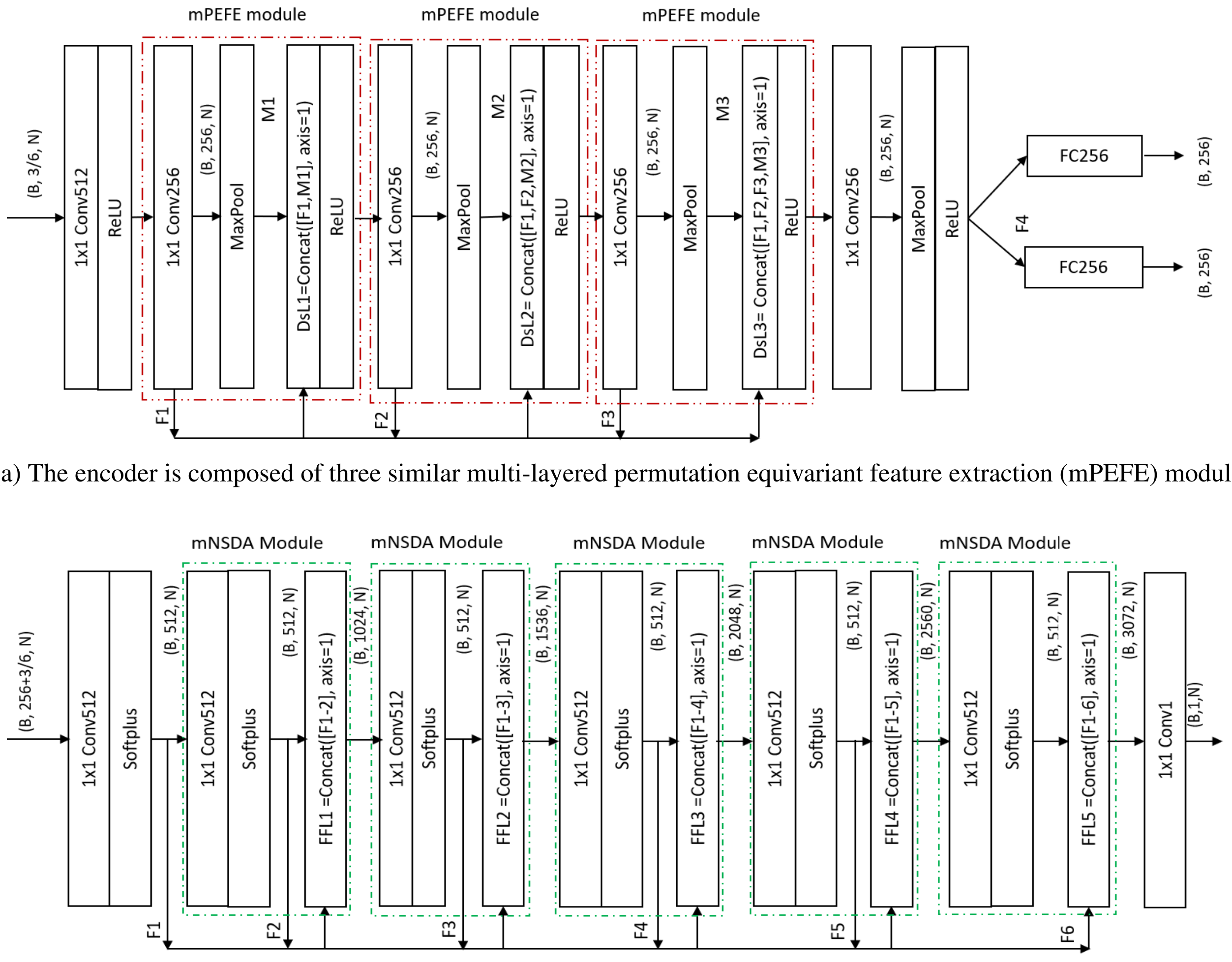
In the context of learning from point clouds, feature fusion and skip connections have been used in a few studies [41], [42], [43] to improve learning of shapes/scenes. However, the way features are fused in previous studies is significantly different from the proposed architecture; previous studies perform feature fusion similar to UNets [44], whereas in the proposed architecture the features are fused in a fashion similar to DenseNet [13] to fuse the features of our proposed variational auto-encoder architecture.

III. PROPOSED ARCHITECTURE

In the following, we present our densely connected deep feature fusion network architecture, DC-DFFN, for implicit 3D representation. DC-DFFN is directly trainable on raw input data, for example raw (un-oriented) point clouds or triangle soups. Our proposed variational auto-encoder consist of an encoder and a decoder, which are constructed from novel mPEFE (encoder) and mNSDA (decoder) convolutional modules. The proposed feature fusion concept is applied within the decoder, and within the encoder, but not between them. We describe the proposed network in the following sections.

A. ENCODER

The DC-DFFN encoder essentially consists of three multi-layered permutation equivariant feature extraction (mPEFE) modules. The mPEFE module includes of two layer types: (a) a modified PointNet [45] layer (Conv1D-MaxPool)



(a) The encoder is composed of three similar multi-layered permutation equivariant feature extraction (mPEFE) module.

(b) Decoder is composed of five symmetric multi-layered neural signed distance approximation (mNSDA) module.

FIGURE 2. The proposed DC-DFFN encoder and decoder architectures consist of 3 mPEFE modules, and 5 mNSDA modules, respectively. In the network figures, DsL{1-3}, B, F{1-3}, M{1-3}, FFL{1-5} (decoder) and N stand for DeepSet layer constructed with densely connected features, batch size, the features after 1D convolutional operation, extracted features after max pooling operation, densely connected deep feature fusion layer (decoder), and the number of input points to the network, respectively. In addition, Concat([F1-5], axis=1) states that the features (F1, F2, F3, F4, F5) are concatenated along channel dimension. The encoder receives raw input data points (N, 3)/(N, 6) (the latter, if surface normals are available) and unsigned distances as ground truth and outputs a latent vector, μ, and diagonal covariance matrix, Σ = diag exp η, which are later used to form a probability measure $\mathcal{N}(\mu, \Sigma)$ to construct the latent code that represents the input object shape. The decoder uses the encoded latent vector as the input to the network and predicts the signed distances, which are later used to mesh the shape using, e.g., the Marching Cubes algorithm.

and (b) a DeepSet [46] layer. The PointNet layer extracts permutation invariant global features using the max-pooling layer as a symmetric function for high dimensional feature embedding learned from unstructured raw point cloud data. Our PointNet layer implementation uses 1D convolutional layers, where the original PointNet [45] layers rely on 2D convolution. The DeepSet layer performs the amalgamation of global features with high dimensional embedding of local features extracted by the convolutional layers. A similar implementation pattern can also be achieved by fully connected layers (used by SAL [8] and SALD [1]) instead of the 1D convolutional layers (see Appendix A for a more information). Through the dense interconnectivity between mPEFE modules the proposed architecture aggregates *multi-layered* local features with order invariant global features within the in-built DeepSet layer. In contrast, the architectures [1] and [8] concatenate only *single-layer* local features with an order invariant global features.

Within the mPEFE layers, the 1D convolutional layers are followed by a pooling layer, a feature fusion layer (modified DeepSet layer) and a ReLU activation function [47]. The last convolutional layer, outside mPEFE modules, is followed by a pooling layer and a ReLU [47] activation function. Two fully connected layers are used at the end of the network to formulate the probability measures $\mathcal{N}(\mu, \eta)$, where μ is the latent vector, and η is used to compute the diagonal covariance matrix, Σ = diag exp η. Therefore, the encoder $(\mu, \eta) = g(X, w_1)$ takes $X \in \mathbb{R}^3$ as input data and outputs the two 256 dimensional vectors, $\mu \in \mathbb{R}^{256}$, and $\eta \in \mathbb{R}^{256}$.

B. DECODER

Our decoder consists of five symmetric multi-layered neural signed distance approximation (mNSDA) modules. The mNSDA module has two main components: (a) signed distance extraction layers (Conv1D-Softplus with $\beta = 100$), and (b) signed distance blending, i.e. feature blending layer. The l^{th} module of the decoder receives input data from all the preceding $(l-1)^{th}$ modules, fused by a concatenation operation in the channel dimension. The mNSDA module has some resemblance to the DeepSDF [10] decoder, however DeepSDF uses a fully connected layer instead of our convolutional layer, and a ReLU activation function instead of our SoftPlus. The mNSDA module promotes maximal information flow through the network layers and also prevents the vanishing gradient problem, mentioned in the DeepSDF [10] work, from reducing performance. The high level architectural structure of our decoder is also similar to that of DeepSDF, except for the dense connectivity introduced in our architecture.

Withing the decoder there are a total of seven 1D convolutional layers (five of which are inside the mNSDA modules). The input size of the first layer is (256 + 3/6, 512, N), the following $([d_{out_{(0)}}, \ldots, d_{out_{(l-2)}}], 512, \mathrm{N})$, and in the last layer $([d_{out_{(0)}}, d_{out_{(1)}}, \ldots, d_{out_{(l-1)}}], 1, \mathrm{N})$. Here, d_{out_l} is the l^{th} layer output of the decoder, and [., .] stands for the concatenation operation, which concatenates the previous $(l-1)^{th}$ layer features of the decoder for the next layer input. N stands for the number of input points, which is in this case 128^2. The decoder's input is $[x, z]$ where $x \in \mathbb{R}^3$, and $z \in \mathbb{R}^3$ is the latent vector.

C. DATA PREPARATION

The unsigned distances of given raw input data $\mathcal{X}$ are pre-computed for 500k sample points using the CGAL library [48] to speed up the training. Moreover, the SALD loss is computed over the data points and their corresponding unsigned distance derivatives sampled from distributions $\mathcal{D}$ and $\mathcal{D}'$. Following [1], we set $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2$, where $\mathcal{D}_1$ is set to be uniformly distributed sampling points $\{\mathbf{y}\}$ from $\mathcal{X}$, putting two isotropic Gaussians, $\mathcal{N}(y, \sigma_1^2 I)$ and $\mathcal{N}(y, \sigma_2^2 I)$ for each $\mathbf{y}$. Here, σ_1 depends on each sampled point $\mathbf{y}$ and is the distance of the 50th closest point to $\mathbf{y}$, however, σ_2 is set to be a fixed value of 0.3. On the other hand, $\mathcal{D}_2$ is estimated by projecting $\mathcal{D}_1$ to surface $\mathcal{S}$.

D. TRAINING AND INFERENCE

We used the SALD loss proposed in [1] with the Adam optimizer [49] to train our proposed DC-DFFN architecture. The SALD loss requires gradient incorporation in a differential manner, which is done based on automatic differentiation [50] forward mode by constructing similar network layers as in [36]. For the D-Faust dataset, a fixed learning rate of 0.0005 and 500 training epochs were used for all models. On the other hand, for the ShapeNet dataset, the initial learning rate was set to 0.0005 and all models were trained for 1500 epochs. Moreover, a scheduler was set to decrease the learning rate by a factor of 0.5 after every 1000 epochs for the Shapenet dataset. For both datasets, the training was performed on a dual 24GB GeForce RTX 3090 GPU in the Ubuntu (20.04) Linux environment.

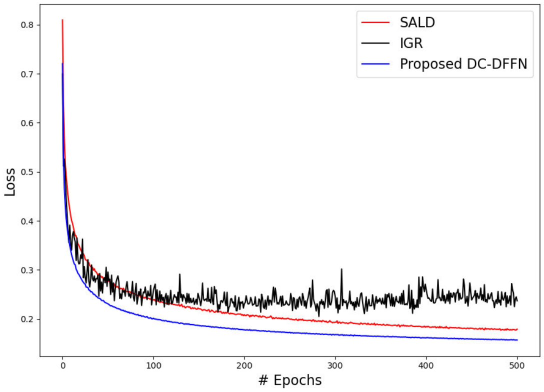


FIGURE 3. Training loss curves of IGR, SALD, and the proposed DC-DFFN model. It can be clearly seen that the proposed architecture learns faster and reaches lower loss than the baseline architectures. The SAL loss curve is not provided because the numerical scale of SAL loss significantly differs from the other three methods.

In the inference phase, the implicit representation of test samples was meshed using the Marching Cubes algorithm [25]. For quantitative comparisons Chamfer distances and intersection-over-union (IOU) between the reconstructed surface against the ground truth (for both datasets) and input raw scans (for D-Faust dataset) were computed.

IV. EXPERIMENTAL RESULT

The proposed DC-DFFN architecture is evaluated on two challenging benchmark datasets, and compared to recently proposed three state-of-the-art approaches [1], [8], [36].

A. DATASETS

1) D-FAUST [51]

The D-Faust dataset contains 41k raw 3D scans (triangle soups) of 10 Human subjects including 5 female and 5 male subjects, in multiple poses. The scans have various defects, such as noise, holes, missing body parts, and occasional artifacts caused by reflections. In training and testing, we include only 1 out of 5 samples from the total 41k scans due to the dense temporal sampling of the dataset. We establish three types of experiments on the D-Faust dataset [51] following the experimental setup used in [8]: (a) shape space learning where 10 human subjects in various poses (129 different actions) are used for training and testing, (b) generalization on unseen human shapes where 8 human (4 females and 4 males) subjects are used to train the model, and 2 human (1 male and 1 female) subjects are used to test the performance of the trained model, (c) generalization to unseen poses, where randomly selected two human poses (from 10 human subjects) were used to test the model, and the rest of the data (10 humans) were used to train the model.

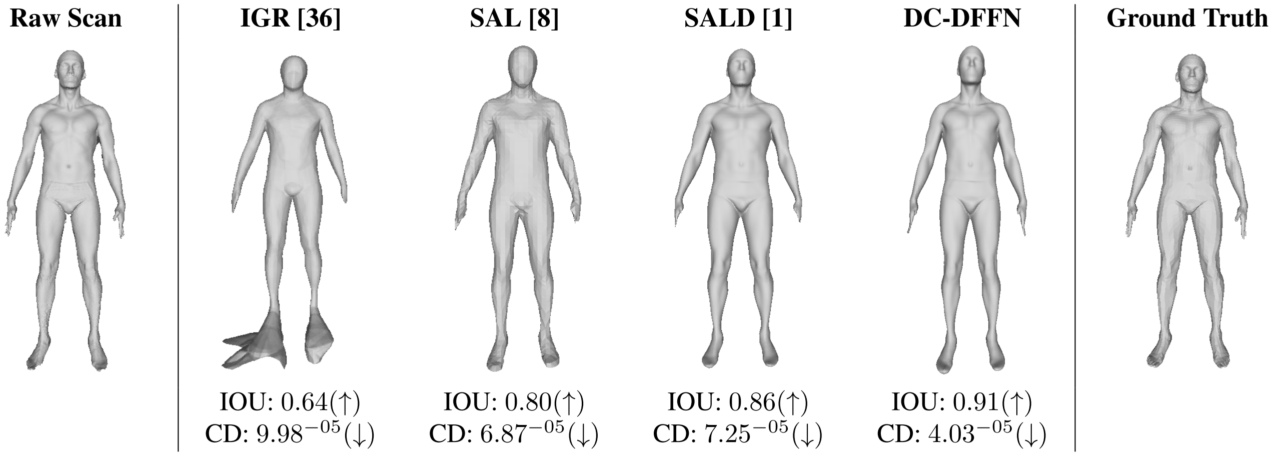


FIGURE 4. **Visual and quantitative results of a single test sample for *shape space learning*. From the quantitative and qualitative results, it is clearly visible that the proposed DC-DFFN generates high fidelity reconstruction and achieves better IOU and CD scores than the baseline architectures.**

TABLE 1. **Quantitative results on shape space learning for the D-Faust dataset. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. ↓: lower value is better; ↑ higher value is better.**

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test data (Shape Space Learning) : Section IV-D	SAL [8]	Rg→Gn	0.052	0.087	0.219	0.122 ± 0.181	0.747 ± 0.069
		Gn→Rg	0.034	0.057	0.124	0.066 ± 0.042	
		Sc→Gn	0.038	0.055	**0.139**	0.073 ± 0.152	0.750 ± 0.070
		Gn→Sc	0.06	0.095	0.164	0.101 ± 0.045	
	IGR [36]	Rg→Gn	0.063	0.163	3.896	5.295 ± 79.044	0.627 ± 0.077
		Gn→Rg	0.576	2.209	14.599	12.530 ± 95.377	
		Sc→Gn	0.047	0.103	3.577	4.491 ± 63.153	0.631 ± 0.077
		Gn→Sc	0.637	2.261	14.996	5.018 ± 20.778	
	SALD [1]	Rg→Gn	0.046	0.098	0.514	0.161 ± 0.245	0.805 ± 0.066
		Gn→Rg	0.041	0.063	0.198	0.085 ± 0.091	
		Sc→Gn	0.038	0.067	0.365	0.118 ± 0.169	0.807 ± 0.067
		Gn→Sc	0.052	0.0.83	0.229	0.105 ± 0.098	
	Proposed DC-DFFN	Rg→Gn	**0.034**	**0.067**	**0.244**	**0.107** ± **0.162**	**0.858** ± **0.053**
		Gn→Rg	**0.031**	**0.049**	**0.108**	**0.057** ± **0.043**	
		Sc→Gn	**0.025**	**0.042**	0.165	**0.064** ± **0.098**	**0.862** ± **0.066**
		Gn→Sc	**0.042**	**0.067**	**0.137**	**0.075** ± **0.039**	

We consider the same train and test split as [8]. However, we have removed those poses from the test shapes that contain scanning artifacts such as floor or side walls. The cleaned test split shapes have been used during the inference for all methods. Therefore, for each experimental setting the number of test shapes before and after the removal are: (a) (2044 → 2003), (b) (1920 → 1869), and (c) (652 → 651). However, the results with the original test splits (with artifacts) can be found in Appendix B.

2) ShapeNet [52]

The ShapeNet dataset contains non-manifold meshes with inconsistent orientation. We consider four different object classes in our experiments: (1) Car (3533), (2) Sofa (3173), (3) Guitar (797), and (4) Airplane (4045). The performance of the proposed DC-DFFN architecture. For ShapeNet, train and test split files (75/25) were created locally.

B. METRICS

For performance evaluation, we consider Chamfer distance (CD) and volumetric intersection over union (IOU).

Volumetric IOU is the quotient of the volume of the generated and the ground truth meshes' union and their intersection. As the baseline implicit reconstruction methods and our proposed DC-DFFN architecture produce only the mesh file, we create voxelized volumes of the test-time ground truth meshes and of the generated meshes. In order to obtain unbiased estimates of the union and intersection volumes, we randomly sample 100k data points from the ground truth and generated meshes, and determine whether the points are occupied or not occupied.

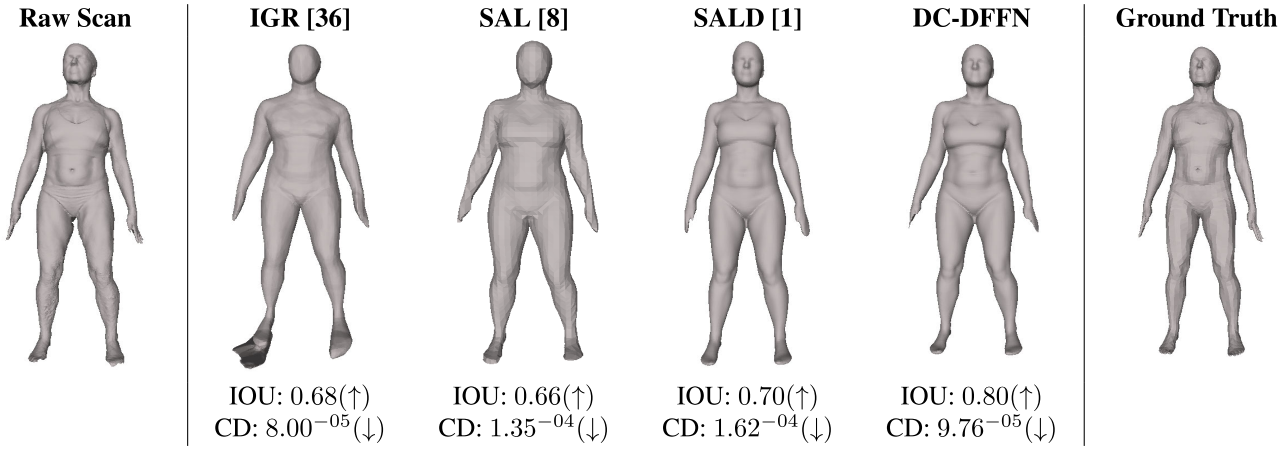


FIGURE 5. Visual comparison of a single test sample for *unseen human shape learning*. Quantitative scores are for reconstructed mesh against the ground truth. ↓: lower value is better; ↑: higher value is better.

TABLE 2. Generalization performance on *unseen human shape reconstruction* for the D-Faust dataset. The chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. ↓: lower value is better; ↑ higher value is better.

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test Data (Unseen Human) : Section IV-E	SAL [8]	Rg→Gn	0.129	0.385	1.646	0.756 ± 1.992	0.450 ± 0.118
		Gn→Rg	0.090	0.286	0.967	0.630 ± 3.302	
		Sc→Gn	0.117	0.349	1.404	**1.287** ± **14.128**	0.447 ± 0.119
		Gn→Sc	0.117	0.327	1.029	0.437 ± 0.631	
	IGR [36]	Rg→Gn	**0.081**	**0.260**	9.415	4.803 ± 58.316	0.582 ± 0.086
		Gn→Rg	0.276	1.335	18.855	17.249 ± 95.730	
		Sc→Gn	**0.063**	**0.202**	7.268	3.661 ± 48.195	0.585 ± 0.085
		Gn→Sc	0.335	1.331	19.512	4.331 ± 17.014	
	SALD [1]	Rg→Gn	0.243	0.779	6.544	1.607 ± 2.202	0.490 ± 0.117
		Gn→Rg	0.231	0.558	2.481	0.913 ± 3.054	
		Sc→Gn	0.233	0.726	4.807	2.705 ± 17.230	0.489 ± 0.118
		Gn→Sc	0.245	0.580	2.477	0.867 ± 0.816	
	Proposed DC-DFFN	Rg→Gn	0.108	0.423	**5.442**	**1.321** ± **2.834**	**0.637** ± **0.098**
		Gn→Rg	**0.098**	**0.216**	**0.858**	**0.466** ± **5.249**	
		Sc→Gn	0.101	0.381	**3.850**	1.772 ± 9.916	**0.636** ± **0.099**
		Gn→Sc	**0.106**	**0.239**	**0.824**	**0.329** ± **0.304**	

Chamfer distance is computed as the mean distance of points from the generated mesh to the ground truth mesh and in the opposite direction as well. Additionally, we compute the Chamfer distances between the generated mesh and the input scan. Similar to the evaluation approach taken in [5], we define *completeness* as the computed mean Chamfer distance from the direction of registration, Rg, (ground truth) / raw input scan (Sc) to the generated mesh (Gn) (Rg→Gn, and Sc→Gn), whereas the opposite direction (Gn→Rg, and Gn→Sc) is defined as *accuracy*.

C. BASELINES

We compare the proposed DC-DFFN architecture to several related generative approaches that are capable of learning the shape space directly from raw 3D data.

SAL [8]: SAL is a generative implicit 3D reconstruction approach that learns the shape space from raw unsigned geometric data in a sign agnostic manner. We compare the proposed work against SAL using the D-Faust dataset, as SAL has inherent difficulties [8] in reconstructing thin shapes that are common in the ShapeNet dataset.

SALD [1]: SALD is a state-of-the-art approach for reconstructing 3D surfaces, which uses a sign agnostic regression loss function with derivatives, and learns the shape space directly from raw unsigned geometric data. The proposed approach is compared against SALD in all of our experiments.

IGR [36]: As shown in [36], a simple loss function can possess the implicit geometric regularization (IGR) property, which allows to generate smooth and high fidelity

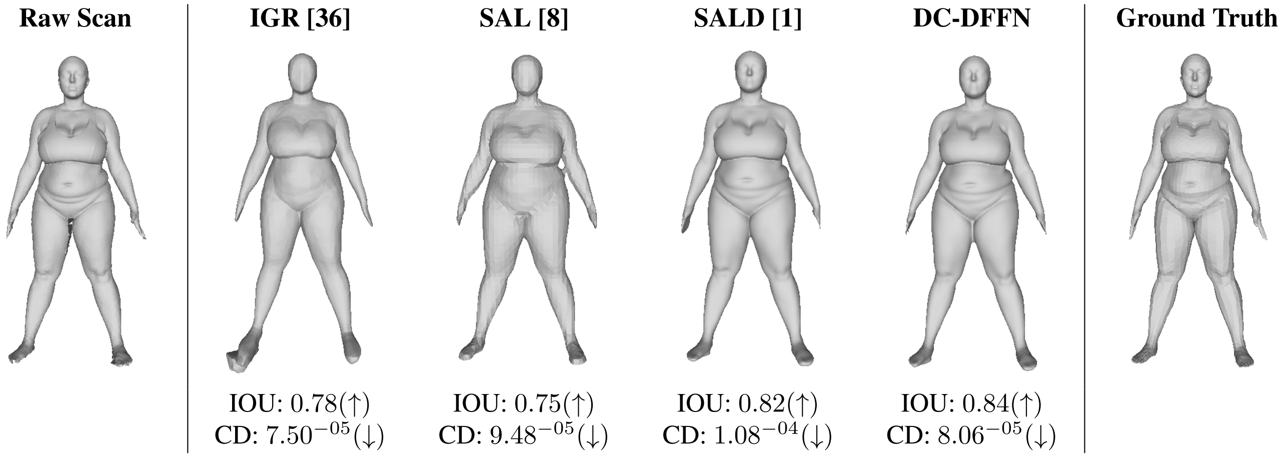


FIGURE 6. Visual comparison of baseline methods and the proposed DC-DFFN on *unseen pose learning* with IOU and Chamfer distance scores. ↓: lower value is better; ↑: higher value is better.

TABLE 3. Comparison on *unseen pose learning* for D-Faust test samples. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. ↓: lower value is better; ↑ higher value is better.

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test Data (Unseen Pose) : Section IV-F	SAL [8]	Rg→Gn	0.077	0.256	**1.334**	**0.418** ± **0.448**	0.556 ± 0.143
		Gn→Rg	0.042	0.145	0.695	0.216 ± 0.218	
		Sc→Gn	0.063	0.207	**1.180**	**0.356** ± **0.393**	0.554 ± 0.144
		Gn→Sc	0.067	0.182	0.749	0.256 ± 0.230	
	IGR [36]	Rg→Gn	0.056	0.190	4.029	0.965 ± 2.414	0.621 ± 0.072
		Gn→Rg	0.484	2.615	8.090	4.428 ± 34.848	
		Sc→Gn	0.042	0.109	3.318	0.668 ± 1.422	0.624 ± 0.072
		Gn→Sc	0.518	2.740	8.290	3.352 ± 4.781	
	SALD [1]	Rg.→Gn	0.065	0.531	3.065	1.053 ± 1.628	0.607 ± 0.147
		Gn→Rg	0.059	0.322	1.221	0.434 ± 0.394	
		Sc→Gn	0.060	0.496	2.553	0.864 ± 1.027	0.606 ± 0.148
		Gn→Sc	0.068	0.341	1.277	0.462 ± 0.411	
	Proposed DC-DFFN	Rg→Gn	**0.042**	**0.169**	2.062	0.493 ± 1.060	**0.742** ± **0.099**
		Gn→Rg	**0.037**	**0.118**	**0.424**	**0.161** ± **0.142**	
		Sc→Gn	**0.033**	**0.138**	1.623	0.361 ± 0.657	**0.743** ± **0.099**
		Gn→Sc	**0.049**	**0.142**	**0.471**	**0.184** ± **0.149**	

surfaces directly from raw (un-oriented point cloud or triangle soup) input data, by directing the neural network to vanish on the input data, and ensuring unit norm gradient. Previously, IGR has achieved state-of-the-art quantitative results and high fidelity reconstruction [36]. The proposed approach is compared against IGR using the D-Faust dataset.

D. SHAPE SPACE LEARNING

In this experiment, randomly selected 70% of 10 D-Faust human subjects are used to train the proposed and the baseline architectures with 500 epochs. The remaining data, 30% of the samples, are used to test the trained models. Additionally, randomly drawn 128^2 points from pre-computed 500k sample points are used to train and test the proposed architecture. For baseline architectures, the number of randomly drawn points is as follows: SAL — 128^2, SALD — 92^2, and IGR — 128^2 as given in the respective original implementations. The final shape reconstruction has been generated with a resolution of 100^3 for all architectures.

The quantitative and qualitative results of this experiment are shown in Table 1, and Fig. 4, respectively. From the quantitative results, it can be clearly seen that the proposed architecture achieves superior results compared to the baseline methods, with one exception: SAL outperforms the proposed approach in 95% percentile using the completeness (Sc→Gn) measure. Moreover, the proposed architecture generates surface reconstruction with superior accuracy in small details compared to the baselines, which can be seen in the IOU results and in Fig. 1.

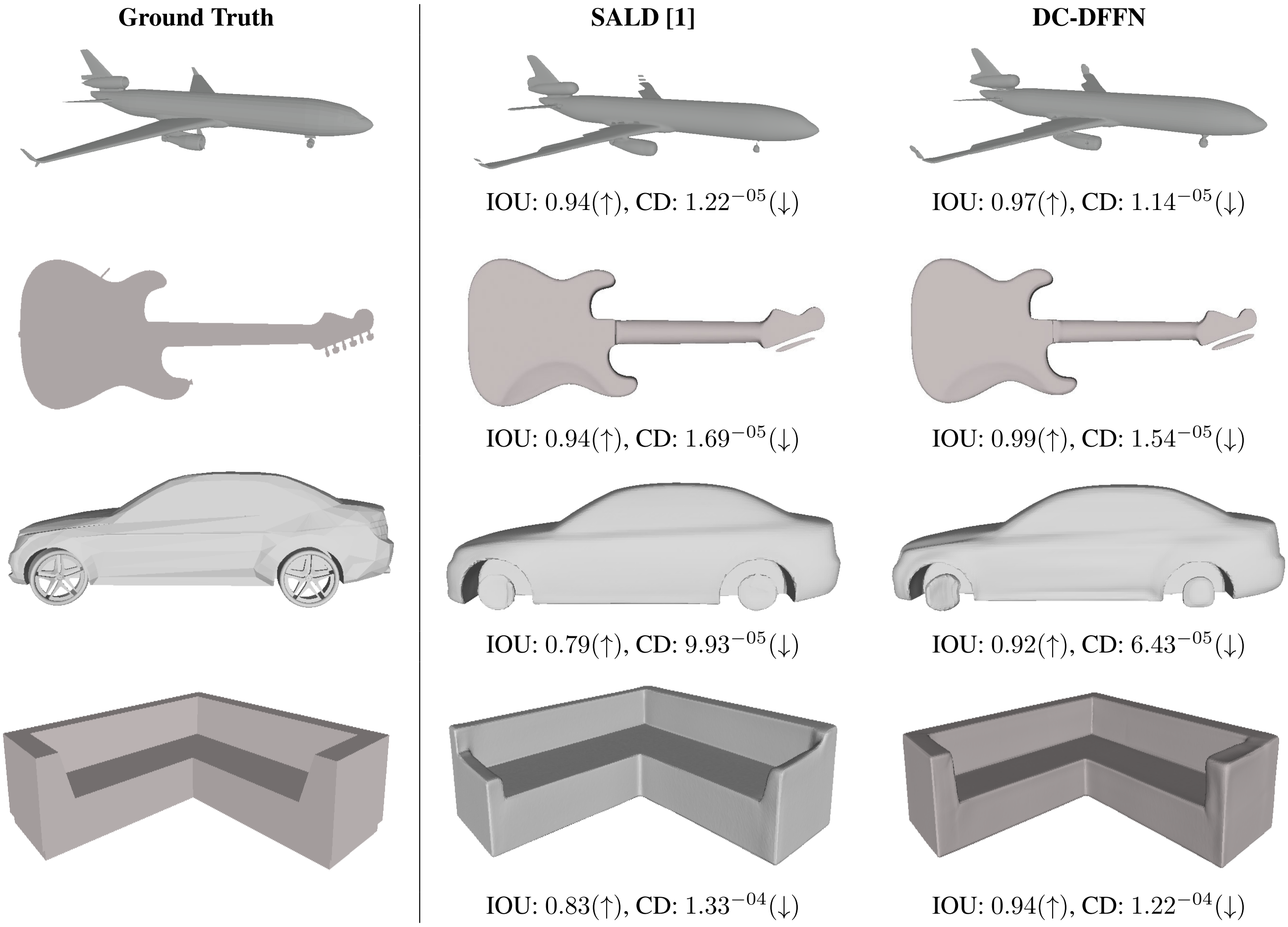

Ground Truth
SALD [1]
DC-DFFN
IOU: 0.94(↑), CD: 1.22^{-05}(↓)
IOU: 0.97(↑), CD: 1.14^{-05}(↓)
IOU: 0.94(↑), CD: 1.69^{-05}(↓)
IOU: 0.99(↑), CD: 1.54^{-05}(↓)
IOU: 0.79(↑), CD: 9.93^{-05}(↓)
IOU: 0.92(↑), CD: 6.43^{-05}(↓)
IOU: 0.83(↑), CD: 1.33^{-04}(↓)
IOU: 0.94(↑), CD: 1.22^{-04}(↓)

FIGURE 7. Visual quality comparison of a single test sample from each ShapeNet [52] test class. ↓: lower value is better, ↑: higher value is better.

E. GENERALIZATION TO UNSEEN HUMANS

In this experiment on the D-Faust dataset, we test our the proposed architecture's generalization capability on previously unknown subjects. The training samples have been drawn from randomly selected 8 human subjects (4 females and 4 males subjects), whereas the remaining 2 human subjects (1 male and 1 female subject) were used to test the architectures. The same number of randomly drawn data points (SAL: 128^2, SALD: 92^2, IGR:128^2, and DC-DFFN:128^2) were used to train and test the models.

The evaluation results in Table 2 show that the proposed architecture has outperformed the baseline architectures in most cases in each metric. As exceptions, Table 2 shows that IGR outperforms the proposed architecture in two Chamfer distance cases (5%, and 50% percentiles, Rg→Gn and Sc→Gn, respectively). However, in the case of mean Chamfer distance and IOU, DC-DFFN outperforms IGR by a large margin. Additionally, the proposed architecture can preserves structural detail better than the baseline architectures. Compared to the shape space learning experiment (Section IV-D), all architectures however provide overall worse results, as expected.

F. GENERALIZATION TO UNSEEN POSES

In this experiment on the D-Faust dataset, two poses have randomly been selected from 10 humans subjects for testing, and the rest of the data is used to train the models. Similar to previous settings, the number of data points used are: SAL — 128^2, SALD — 92^2, IGR — 128^2, and DC-DFFN — 128^2, drawn from the pre-computed sample data points during the training and testing the models.

We results are shown in Table 3, and Fig. 6: on average, DC-DFFN achieves better or comparable results in all metrics. As an exception, SAL outperforms the proposed architecture in terms of Chamfer distance (95% percentile) and completeness (Rg→Gn and Sc→Gn) scores. However, in terms of volumetric IOU DC-DFFN outperforms all other approaches without exceptions.

G. GENERALIZATION TO OBJECT SHAPES

Beyond learning human shapes, also experiments on learning object shapes were performed to evaluate the proposed architecture using the ShapeNet dataset comprises of non-manifold/non-oriented meshes that depict various objects.

TABLE 4. ShapeNet quantitative results. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. Moreover, ↓ means lower is better, whereas ↑ means higher is better.

Class	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Car	SALD [1]	Rg.→Gn	0.125	0.319	0.881	0.383 ± 0.265	0.671 ± 0.091
		Gn→Rg	0.069	0.209	0.787	0.316 ± 0.328	
	Proposed DC-DFFN	Rg→Gn	**0.095**	**0.266**	**0.822**	**0.336 ± 0.279**	**0.721 ± 0.097**
		Gn→Rg	**0.033**	**0.164**	**0.527**	**0.207 ± 0.207**	
Sofa	SALD [1]	Rg.→Gn	0.064	0.315	1.080	0.422 ± 0.374	0.636 ± 0.159
		Gn→Rg	0.043	0.181	1.412	0.383 ± 0.604	
	Proposed DC-DFFN	Rg→Gn	**0.040**	**0.214**	**0.874**	**0.300 ± 0.284**	**0.716 ± 0.143**
		Gn→Rg	**0.026**	**0.102**	**0.712**	**0.220 ± 0.471**	
Guitar	SALD [1]	Rg.→Gn	**0.012**	0.039	**0.192**	**0.069** ± 0.140	0.778 ± 0.131
		Gn→Rg	0.010	0.052	0.403	0.109 ± 0.194	
	Proposed DC-DFFN	Rg→Gn	0.013	**0.028**	0.215	0.402 ± 3.252	**0.808 ± 0.145**
		Gn→Rg	**0.007**	**0.019**	**0.202**	**0.056 ± 0.120**	
Airplane	SALD [1]	Rg.→Gn	0.011	0.057	0.518	0.143 ± 0.323	0.761 ± 0.145
		Gn→Rg	0.008	0.044	0.481	0.124 ± 0.287	
	Proposed DC-DFFN	Rg→Gn	**0.009**	**0.050**	**0.515**	**0.133 ± 0.266**	**0.784 ± 0.137**
		Gn→Rg	**0.006**	**0.029**	**0.399**	**0.096 ± 0.250**	

TABLE 5. Comparison against inferior DC-DFFN variants on *unseen human* reconstruction. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, where the Chamfer distance numbers are multiplied by 10^3. All variants were with 500 epochs. ↓: lower value is better; ↑: higher value is better.

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test Data (Unseen Human) : Section IV-E	DC-DFFN-MLLCs	Rg→Gn	0.153	0.509	**4.1776**	**1.078 ± 1.588**	0.583 ± 0.101
		Gn→Rg	0.147	0.317	1.188	3.399 ± 20.871	
	DC-DFFN-Lin	Rg→Gn	0.170	0.711	7.247	1.745 ± 3.137	0.559 ± 0.113
		Gn→Rg	0.151	0.372	1.556	3.861 ± 24.500	
	Proposed DC-DFFN	Rg→Gn	**0.108**	**0.423**	5.442	1.321 ± 2.834	**0.637 ± 0.098**
		Gn→Rg	**0.098**	**0.216**	**0.858**	**0.466 ± 5.249**	

Compared to human shapes, objects exhibit sharp corners, holes and thin structure. For baseline architectures, we have resorted to the train and test settings as described in the original works, except for data split files where we used 75% of the samples for training and 25% samples for testing. The network architectures were trained with 1500 epochs on each class (Car, Guitar, Airplanes, and Sofa). The IGR method was not included in this evaluation, as IGR expects consistently oriented normals, which were unavailable for the ShapeNet dataset.

The results are shown in Table 4 and Fig. 7, respectively. DC-DFFN significantly outperforms the state-of-the-art SALD architecture, except for the Guitar class, where measured Chamfer distances favor SALD in 5% and 95% percentile cases. The qualitative results of Fig. 7 show that DC-DFFN can capture thin structure (airplane wings) and sharp corners (sofa armrests) much better than SALD.

V. DISCUSSION

In this study, we proposed the feature fusion-based variational auto-encoder network DC-DFFN. The novel characteristics in the architecture design improve training speed (see Fig. 3), improve performance at inference time (see Tables 1, 2, 3, and 4), and provide better generalisation in the 3D shape space compared to reviewed baseline approaches.

In the test data split of D-Faust, we decided to remove the samples (See Section IV-A) that include scanning artifacts (see also Appendix B and Fig. 9) for improving the interpretability of the results. However, this decision can also be questioned, as keeping the scanning artifacts could on the other hand evaluate the architectures' capability of rejecting outliers.

However, Table 6, and Table 7 in Appendix B show the D-Faust results with artifacts included, and confirms that the proposed DC-DFFN architecture outperforms in general the baseline architectures in the presence of artifacts as

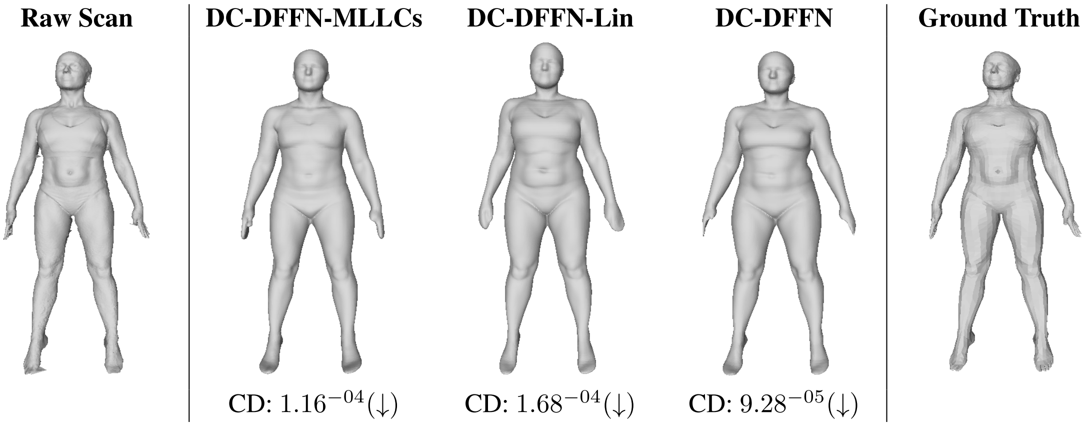

Raw Scan
DC-DFFN-MLLCs
DC-DFFN-Lin
DC-DFFN
Ground Truth
CD: $1.16^{-04}(\downarrow)$
CD: $1.68^{-04}(\downarrow)$
CD: $9.28^{-05}(\downarrow)$

FIGURE 8. The qualitative results are shown for other alternative variant of the feature fusion networks and the proposed DC-DFFN on the experimental setup illustrated in section IV-E of main paper for D-Faust dataset. Additionally, we reported the computed Chamfer distance for each model, where ↓ means lower is better.

TABLE 6. Results on D-Faust *shape space learning* including samples with artifacts. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. ↓: lower value is better; ↑ higher value is better.

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test data (Shape Space Learning) : Section IV-D	SAL [8]	Rg→Gn	0.052	0.088	0.277	0.161 ± 0.502	0.737 ± 0.100
		Gn→Rg	0.035	0.058	0.146	6.698 ± 56.754	
		Sc→Gn	0.037	0.056	**0.168**	0.192 ± 2.489	0.740 ± 0.099
		Gn→Sc	0.060	0.095	0.183	1.362 ± 11.562	
	SALD [1]	Rg→Gn	0.047	0.100	0.523	0.165 ± 0.249	0.800 ± 0.071
		Gn→Rg	0.041	0.064	0.225	1.909 ± 19.725	
		Sc→Gn	0.039	0.069	0.462	2.615 ± 33.149	0.802 ± 0.073
		Gn→Sc	0.052	0.0.84	0.257	0.125 ± 0.325	
	Proposed DC-DFFN	Rg→Gn	**0.034**	**0.067**	**0.248**	**0.110 ± 0.171**	**0.853 ± 0.067**
		Gn→Rg	**0.031**	**0.049**	**0.123**	**3.632 ± 33.574**	
		Sc→Gn	**0.025**	**0.042**	0.203	**1.878 ± 27.973**	**0.857 ± 0.066**
		Gn→Sc	**0.042**	**0.067**	**0.152**	**0.269 ± 3.568**	

TABLE 7. Generalization to D-Faust *unseen humans* including samples with artifacts. The Chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean scores, Chamfer distances multiplied by 10^3. ↓: lower value is better; ↑ higher value is better.

Experimental Data Setup	Method	Direction	Percentile (↓)			Mean ± STD (↓)	IOU ± STD (↑)
			5%	50%	95%		
Test Data (Unseen Human) : Section IV-E	SAL [8]	Rg→Gn	0.130	**0.398**	**3.1186**	**1.026 ± 3.740**	0.443 ± 0.125
		Gn→Rg	0.091	0.295	1.356	2.672 ± 18.480	
		Sc→Gn	0.118	**0.358**	**3.041**	**2.450 ± 20.675**	0.447 ± 0.119
		Gn→Sc	0.117	0.332	1.347	0.496 ± 0.786	
	SALD [1]	Rg→Gn	0.242	0.801	6.647	1.619 ± 2.215	0.490 ± 0.117
		Gn→Rg	0.233	0.565	2.469	1.050 ± 5.194	
		Sc→Gn	0.233	0.757	6.074	6.404 ± 32.635	0.489 ± 0.118
		Gn→Sc	0.245	0.589	2.461	0.871 ± 0.810	
	Proposed DC-DFFN	Rg→Gn	**0.108**	0.429	5.500	1.328 ± 2.817	**0.635 ± 0.099**
		Gn→Rg	0.098	**0.221**	**1.159**	**0.825 ± 10.200**	
		Sc→Gn	**0.102**	0.408	7.585	4.106 ± 23.229	**0.631 ± 0.103**
		Gn→Sc	**0.108**	**0.239**	**0.845**	**0.369 ± 1.586**	

well. In individual cases (See Fig. 9) the baseline SALD approach is sometimes better in removing outliers.

In all experimental setups for both D-Faust and ShapeNet datasets, the proposed DC-DFFN significantly outperformed the baseline approaches in almost all the cases. In a few cases the baseline approaches provided better Chamfer distance results with a small margin. Considering volumetric IOU, in contrast, DC-DFFN outperformed the

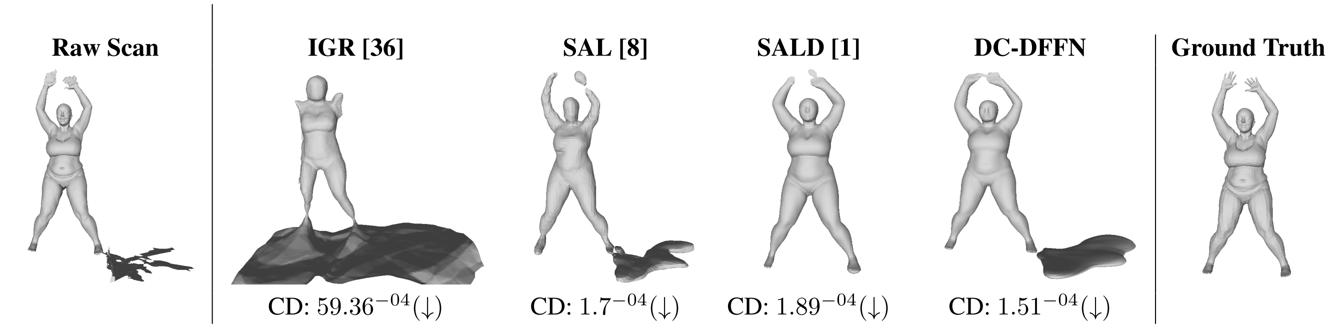

Raw Scan
IGR [36]
SAL [8]
SALD [1]
DC-DFFN
Ground Truth
CD: $59.36^{-04}(\downarrow)$
CD: $1.7^{-04}(\downarrow)$
CD: $1.89^{-04}(\downarrow)$
CD: $1.51^{-04}(\downarrow)$

FIGURE 9. Visual reconstruction result comparison for a sample with scanning artifacts. ↓: lower value is better.

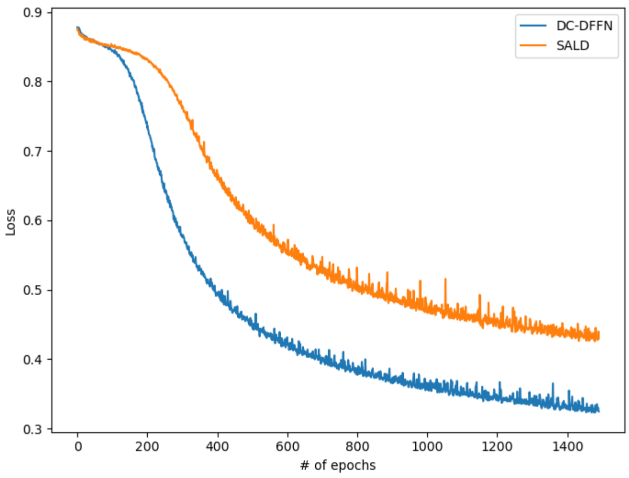


FIGURE 10. Training Loss curve of SALD and DC-DFFN models on ShapeNet dataset (class: lamp). In this dataset, we compared our approach only with SALD. From the learning curve, it can be seen that DC-DFFN is learning faster than the baseline SALD model, however, both architectures are not yet reached to the saddle point at 1500 epochs. Based on the complexities of the data, the required number of training epochs will vary for all models. However, DC-DFFN converges faster than all other baseline approaches presented in this study.

baseline approaches in all experimental setups in both datasets.

Although DC-DFFN performs better than the baseline approaches, however, the proposed architecture still suffers in reconstructing the thin structures to some extent.

VI. CONCLUSION

In this paper we have proposed the densely connected deep feature fusion network architecture DC-DFFN for neural implicit shape learning and reconstruction from raw input data. In the experimental section the proposed work is shown to learn faster, generalize better and outperform the baseline works quantitatively by a clear margin in all experiments. Additionally, the visual results show that the proposed architecture can especially capture small detail better than the previous works. As the broader impact of our work we see that in the future DC-DFFN has potential to serve as the prevailing neural architecture for upcoming studies on shape learning from raw 3D data.

APPENDIX A
ARCHITECTURE ALTERNATIVES

In the process of designing DC-DFFN, two alternative implementation variants of the proposed architecture were developed, out of which the proposed architecture was identified as the best one. The other alternative variants were: (I) Densely connected feature fusion network with multilayered latent codes, DC-DFFN-MLLCs, and (II) Dense layer feature fusion with dense neural network, DFF-DFFN-Lin. Results of these architecture variants are shown in one experimental setting in Table 5.

A. DC-DFFN-MLLCs

In the DC-DFFN-MLLCs architecture variant, one latent code is extracted after every Conv1D-MaxPool-DeepSet-Relu block, and finally, concatenated in the channel dimension. In this case, the final latent code shape is (B, 1024, N), where N=128^2 and B is the batch size. The assumption was that multiple multi-layered latent codes would contain more information than a single latent code and provide better reconstruction quality. However, in practice this architecture variant was performing worse than DC-DFFN.

1) DC-DFFN-LIN

In this variant, the 1D convolutional layers (Kernel: 1×1) of the encoder and decoder were replaced by fully connected layers, keeping the rest of the network settings similar to DC-DFFN-MLLCs. Eventually, DC-DFFN-Lin performed comparatively worse than the multiple latent codes-based architecture, DC-DFFN-MLLCs, constructed with of 1D convolutional layers. Even clearer, DC-DFFN-MLLCs performed significantly worse than the proposed DC-DFFN.

APPENDIX B
RECONSTRUCTION OF SAMPLES WITH ARTIFACTS

In the main paper experiments, the D-Faust samples that contain scanning artifacts in the test samples were removed from the experiments shown in Section IV-D and Section IV-E. Here, in Table 6, Table 6 and Fig. 9 we present as additional quantitative results the D-Faust results without any removed

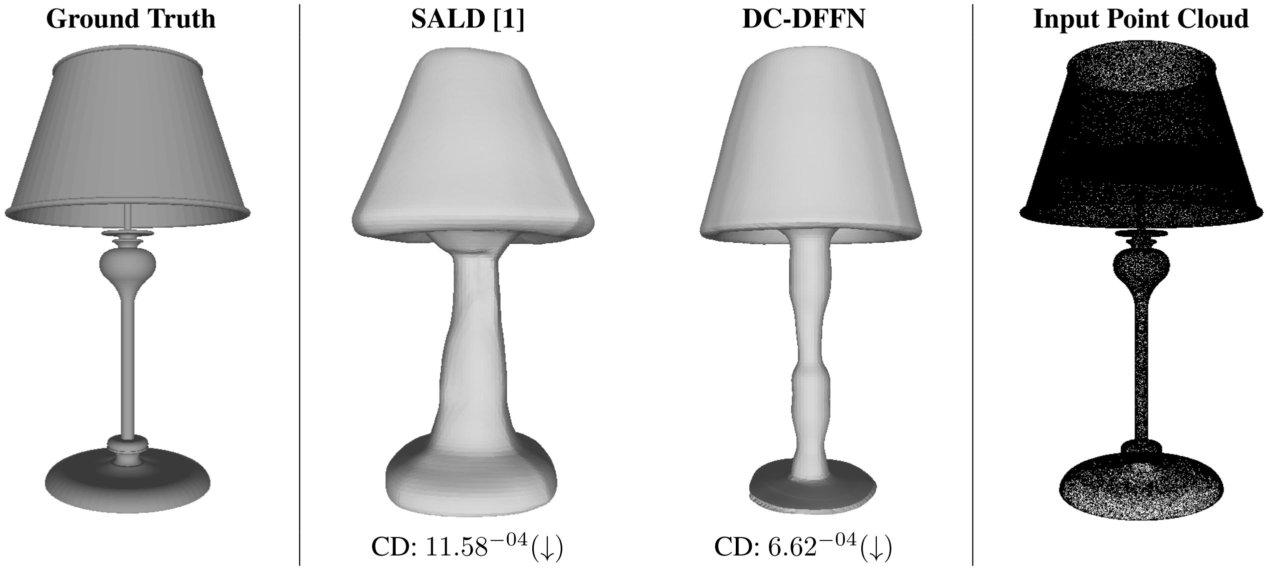


FIGURE 11. Additional qualitative results are shown for ShapeNet dataste (Class: lamp). We reported the computed Chamfer distance for both models, where ↓ means lower is better.

samples, making the test sets identical to the ones used in [1] and [8].

APPENDIX C ADDITIONAL RESULTS

Additional qualitative results are shown in Fig. 11 from ShapeNet dataset (class: Lamp). Moreover, the training loss curve on lamp class is shown in Fig. 10 for SALD, and the proposed DC-DFFN models. It can be seen from the learning curve that the proposed architecture learns faster than the baseline SALD architecture.

REFERENCES

[1] M. Atzmon and Y. Lipman, "SALD: Sign agnostic learning with derivatives," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*. Austria: OpenReview.net, May 2021. [Online]. Available: https://openreview.net/forum?id=7EDgLu9reQD

[2] J. Chibane and G. Pons-Moll, "Implicit feature networks for texture completion from partial 3D data," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 717–725.

[3] J. Chibane, M. A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 33, Jan. 2020, pp. 21638–21652. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/f69e505b08403ad2298b9f262659929a-Paper.pdf

[4] M. Niemeyer, L. Mescheder, M. Oechsle, and A. Geiger, "Occupancy flow: 4D reconstruction by learning particle dynamics," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5379–5389.

[5] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3D reconstruction in function space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4455–4465.

[6] V. Sitzmann, J. N. P. Martel, A. W. Bergman, D. B. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," 2020, *arXiv:2006.09661*.

[7] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "NeRF: Representing scenes as neural radiance fields for view synthesis," 2020, *arXiv:2003.08934*.

[8] M. Atzmon and Y. Lipman, "SAL: Sign agnostic learning of shapes from raw data," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2565–2574.

[9] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. 16th Eur. Conf. Comput. Vis.—ECCV*. Glasgow, U.K.: Springer, Aug. 2020, pp. 523–540.

[10] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning continuous signed distance functions for shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 165–174.

[11] Y. Li and J. Barbič, "Immersion of self-intersecting solids and surfaces," *ACM Trans. Graph.*, vol. 37, no. 4, pp. 1–14, Aug. 2018.

[12] A. Basher, M. Sarmad, and J. Boutellier, "LightSAL: Lightweight sign agnostic learning for implicit surface representation," 2021, *arXiv:2103.14273*.

[13] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[15] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Montreal, QC, Canada, vol. 2. Cambridge, MA, USA: MIT Press, 2015, pp. 2377–2385.

[16] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, Jun. 2013.

[17] D. Levin, "Mesh-independent surface interpolation," in *Geometric Modeling for Scientific Visualization*. Berlin, Germany: Springer, 2004, pp. 37–49.

[18] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. Comput. Graphics*, vol. 5, no. 4, pp. 349–359, Oct. 1999.

[19] J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans, "Reconstruction and representation of 3D objects with radial basis functions," in *Proc. 28th Annu. Conf. Comput. Graph. Interact. Techn.*, Aug. 2001, pp. 67–76.

[20] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," Jun. 2006, pp. 61–70, doi: 10.2312/SGP/SGP06/061-070.

[21] W. Zhao, J. Lei, Y. Wen, J. Zhang, and K. Jia, "Sign-agnostic implicit learning of surface self-similarities for shape modeling and reconstruction from raw point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10256–10265.

[22] A. Gilbert, M. Volino, J. Collomosse, and A. Hilton, "Volumetric performance capture from minimal camera viewpoints," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 566–581.

[23] G. Varol, D. Ceylan, B. Russell, J. Yang, E. Yumer, I. Laptev, and C. Schmid, "BodyNet: Volumetric inference of 3D human body shapes," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–36.

[24] Z. Zheng, T. Yu, Y. Wei, Q. Dai, and Y. Liu, "DeepHuman: 3D human reconstruction from a single image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7739–7749.

[25] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3D surface construction algorithm," *ACM SIGGRAPH Comput. Graph.*, vol. 21, no. 4, pp. 163–169, Aug. 1987.

[26] Y. Liao, S. Donné, and A. Geiger, "Deep marching cubes: Learning explicit surface representations," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2916–2925.

[27] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. Xiao, "3D ShapeNets: A deep representation for volumetric shapes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1912–1920.

[28] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik, "Multi-view supervision for single-view reconstruction via differentiable ray consistency," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2626–2634.

[29] G. Gkioxari, J. Johnson, and J. Malik, "Mesh R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9785–9795.

[30] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "AtlasNet: A papier-Mâché approach to learning 3D surface generation," 2018, *arXiv:1802.05384*.

[31] M. M. Bronstein, J. Bruna, Y. LeCun, A. Szlam, and P. Vandergheynst, "Geometric deep learning: Going beyond Euclidean data," *IEEE Signal Process. Mag.*, vol. 34, no. 4, pp. 18–42, Jul. 2017.

[32] P. Wang, Y. Gan, P. Shui, F. Yu, Y. Zhang, S. Chen, and Z. Sun, "3D shape segmentation via shape fully convolutional networks," *Comput. Graph.*, vol. 70, pp. 128–139, Feb. 2018.

[33] K. Guo, D. Zou, and X. Chen, "3D mesh labeling via deep convolutional neural networks," *ACM Trans. Graph.*, vol. 35, no. 1, pp. 1–12, Dec. 2015.

[34] R. Venkatesh, S. Sharma, A. Ghosh, L. Jeni, and M. Singh, "DUDE: Deep unsigned distance embeddings for hi-fidelity representation of complex 3D surfaces," 2020, *arXiv:2011.02570*.

[35] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman, "Controlling neural level sets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 2032–2041.

[36] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *Proc. Mach. Learn. Syst.*, 2020, pp. 3569–3579.

[37] A. Pandey and D. Wang, "Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 6629–6633.

[38] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3194–3203.

[39] Y. Li, G. Han, and X. Liu, "DCNet: Densely connected deep convolutional encoder–decoder network for nasopharyngeal carcinoma segmentation," *Sensors*, vol. 21, no. 23, p. 7877, Nov. 2021.

[40] H. Zhou, Z. Fang, Y. Gao, B. Huang, C. Zhong, and R. Shang, "Feature fusion network based on attention mechanism for 3D semantic segmentation of point clouds," *Pattern Recognit. Lett.*, vol. 133, pp. 327–333, May 2020.

[41] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[42] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun, "Point transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16259–16268.

[43] X. Wu, Y. Lao, L. Jiang, X. Liu, and H. Zhao, "Point transformer v2: Grouped vector attention and partition-based pooling," 2022, *arXiv:2210.05666*.

[44] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*. Munich, Germany: Springer, Oct. 2015, pp. 234–241.

[45] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 652–660.

[46] M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Poczos, R. Salakhutdinov, and A. Smola, "Deep sets," 2017, *arXiv:1703.06114*.

[47] V. Nair, and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," Jun. 2010, pp. 807–814. [Online]. Available: https://icml.cc/Conferences/2010/papers/432.pdf

[48] A. Fabri and S. Pion, "CGAL: The computational geometry algorithms library," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geograph. Inf. Syst.*, 2009, pp. 538–539.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[50] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Mach. Learn. Res.*, vol. 18, pp. 1–43, Apr. 2018.

[51] F. Bogo, J. Romero, G. Pons-Moll, and M. J. Black, "Dynamic FAUST: Registering human bodies in motion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5573–5582.

[52] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An information-rich 3D model repository," 2015, *arXiv:1512.03012*.

ABOL BASHER received the B.Sc. degree in electrical and electronics engineering from the Mymensingh Engineering College, University of Dhaka, Bangladesh, in 2015, and the M.E. degree in computer engineering from Chosun University, Gwangju, South Korea, in 2020. He is currently pursuing the Ph.D. degree in computer science with the University of Vaasa, Finland. He is a Project Researcher with the Digital Economy Research Platform, University of Vaasa. His research interests include 3D data representation, computer vision, machine learning, and medical image processing.

JANI BOUTELLIER received the Ph.D. degree, in 2009. He is currently an Associate Professor with the University of Vaasa, Finland. He is leading projects that concentrate on efficient neural networks and 3D computer vision. He has coauthored more than 70 peer-reviewed articles. His research interests include parallel computing, model-based design, and signal processing. He is a member of the IEEE Signal Processing Society ASPS Technical Committee. He is an Associate Editor of the *Journal of Signal Processing Systems* (Springer).