IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 22 January 2025, accepted 1 March 2025, date of publication 6 March 2025, date of current version 17 March 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3549279

APPLIED RESEARCH

ADR-SALD: Attention-Based Deep Residual Sign Agnostic Learning With Derivatives for Implicit Surface Reconstruction

ABOL BASHER AND JANI BOUTELLIER, (Senior Member, IEEE)
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 Scientific Advisory Board for Defense under Project VN/17548/2023-SAAP-25.

ABSTRACT Learning 3D shape directly from raw data (i.e., un-oriented meshes, raw point clouds or triangle soups) and reconstructing high fidelity surfaces are still a difficult problem in computer vision and graphics. Several approaches have been proposed to learn from raw data, however, their reconstruction quality is somewhat limited in capturing small detail. Moreover, they introduce surface sheet in case of big gaps and empty spaces, and struggle in reconstructing small openings and thin structure. In this study, we address these problems by proposing a novel attention-based variational autoencoder architecture, *ADR-SALD* where the encoder and decoder are constructed based on the idea of *residual feature learning* and inception-like neural structure. We have adopted two different *self attention* mechanisms for sign agnostic learning in the encoder, which allow the proposed approach to learn the global spatial contextual dependencies and local features simultaneously for the 3D shape. This novel architecture solves the surface sheet problem of previous approaches such as SALD. Moreover, our experimental results show that *ADR-SALD* is more successful in reconstructing thin structure than the state-of-the-art approaches SALD and DC-DFFN, and has significant performance in separating small gaps. The proposed approach outperforms the baseline state-of-the-art approaches by reconstruction quality and quantitative measures.

INDEX TERMS Implicit representation learning, surface reconstruction, ShapeNet, D-Faust, sign agnostic learning, point-wise spatial attention, neighbor to point attention, self attention, convolutional neural networks.

I. INTRODUCTION

Reconstructing continuous and renderable 3D shape / surface from raw point clouds or triangle soups is a challenging task, which has various applications in robotics, computer vision and graphics, and the scientific community has developed various approaches [2], [1], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] to address the problem. However, these approaches have various trade-offs among reconstruction fidelity, expressiveness and processing efficiency. Deep learning based approaches [2], [1], [3], [4], [5], [6], [7] are showing superior performance over traditional approaches [8], [9], [10], [11], [12] in the case of learning 3D geometry and reconstructing high fidelity surfaces. Generative models, such as generative adversarial neural networks (GANs) [13], [14], autoencoder [15], variational autoencoder [2], [17], [1], [16] and autodecoders [2], [1], [5] have been used to learn, reconstruct, and generate surfaces from point clouds, triangle soups and un-oriented meshes. Deep neural networks learn the surface either parametrically or implicitly. In the case of parametric representation, the neural networks are utilized as a parameterization mappings. Implicit representation-based approaches learn the shape as a zero level sets of a neural network. The surface S of an object or scene is expressed in the following manner in implicit neural representation:

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

where $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$ is a neural function and $x \in \mathbb{R}^3$ represents the input data sampled from raw point cloud or triangle soups or un-oriented meshes $\mathcal{X} \in \mathbb{R}^3$.

The associate editor coordinating the review of this manuscript and approving it for publication was Muhammad Sharif.

FIGURE 1. Example from single sample reconstruction that shows the superior performance of the proposed ADR-SALD architecture. ADR-SALD can capture large gaps and empty spaces in the structure.

Most of the recent deep learning-based approaches [3], [4], [18], [19], [20] use implicit neural representation to express surfaces, however, they transform the irregular format of the raw point clouds into a regular grid representation, such as voxels, to define the inside and outside boundaries of surfaces. In contrast, several methods [2], [17], [1], [16], [21] have been proposed to handle raw point clouds directly, however, their reconstruction quality is limited to a certain extent in the case of large structure gaps and capturing a small detail.

In this work, we are concentrating on these shortcomings and propose a novel neural architecture to overcome these challenges. The proposed variational autoencoder architecture, *ADR-SALD*[1], consists of an encoder and a decoder; self-attention and residual learning structures [22] are integrated to the encoder part of the network. Residual learning allows the network to avoid the vanishing gradient problem, capture complex structural semantic features and improves convergence. On the other hand, 1D convolutional layers with 1×1 kernel is used in the decoder that has structural similarity with inception layers [23] and ResNeXt layer [24]. Moreover, we use the Kullback-Leibler divergence (KLD) as the latent regularization instead of the regularization proposed in [2]. Finally, the proposed variational autoencoder, *ADR-SALD* learns unsigned distances with derivatives from raw point clouds in a sign agnostic manner and predicts signed distances in the inference phase, where the predicted signed distances are processed with the marching cubes algorithm [25] to produce high fidelity surface reconstructions. The proposed *ADR-SALD* architecture can capture large gaps in the structure (shape) of an object and capture small detail of the surface. The superior reconstruction quality of the proposed *ADR-SALD* is shown in Fig. 1 compared to the baseline state-of-the-art studies.

The contributions of this work can be summarised as follows:

- The first neural architecture for implicit shape learning to include single-attention and multi-headed attention together
- Leveraging the KLD latent regularization loss in sign agnostic learning
- The first architecture that is capable of simultaneously learning from raw data, and avoiding reconstruction issues related to large structural gaps and/or fine detail,
- State-of-the-art reconstruction quality in both quantitative and qualitative terms.

The rest of the article is organized as follows: in Section II, we illustrate the related studies on surface reconstruction and associated machine learning based approaches; Section III describes the proposed ADR-SALD architecture in detail; dataset, evaluation metrics, qualitative and quantitative performance of the proposed approach and comparative analysis are shown in Section IV. Finally, discussion and summary are provided in Section V.

II. RELATED WORKS

In this section, we review the recent studies on 3D surface reconstruction, as well as major advances in neural architectures and feature learning, which have inspired the work.

Surface reconstruction related studies can be broadly divided into three categories: (a) traditional analytic prior-based surface reconstruction approaches, (b) classical representation learning and (c) implicit neural representation learning. We will briefly discuss these approaches in the following sections.

A. ANALYTIC PRIOR-BASED RECONSTRUCTION METHODS

Various analytic prior-based traditional reconstruction methods [8], [9], [10], [11], [12] have been proposed to approximate 3D surfaces. Poisson surface reconstruction [8] (PSR) considers the surface reconstruction problem from the oriented point clouds as a spatial Poisson problem and works based on a global smoothness priors. PSR considers all the points at once, avoiding any need for heuristic spatial data blending or partitioning. This helps PSR to show strong resilience against noise in the data. However, PSR has a problem of over-smoothing the data. Screened Poisson Reconstruction [9] (SPSR) has been proposed to address the data over-smoothing problem of PSR. SPSR casts 3D surface reconstruction from oriented points as a spatial Poisson problem and works in the frequency domain [30]. However, SPSR does not work without oriented normal information similar to PSR. The radial basis function approach [12]

[1] https://github.com/basher8488881/ADR-SALD

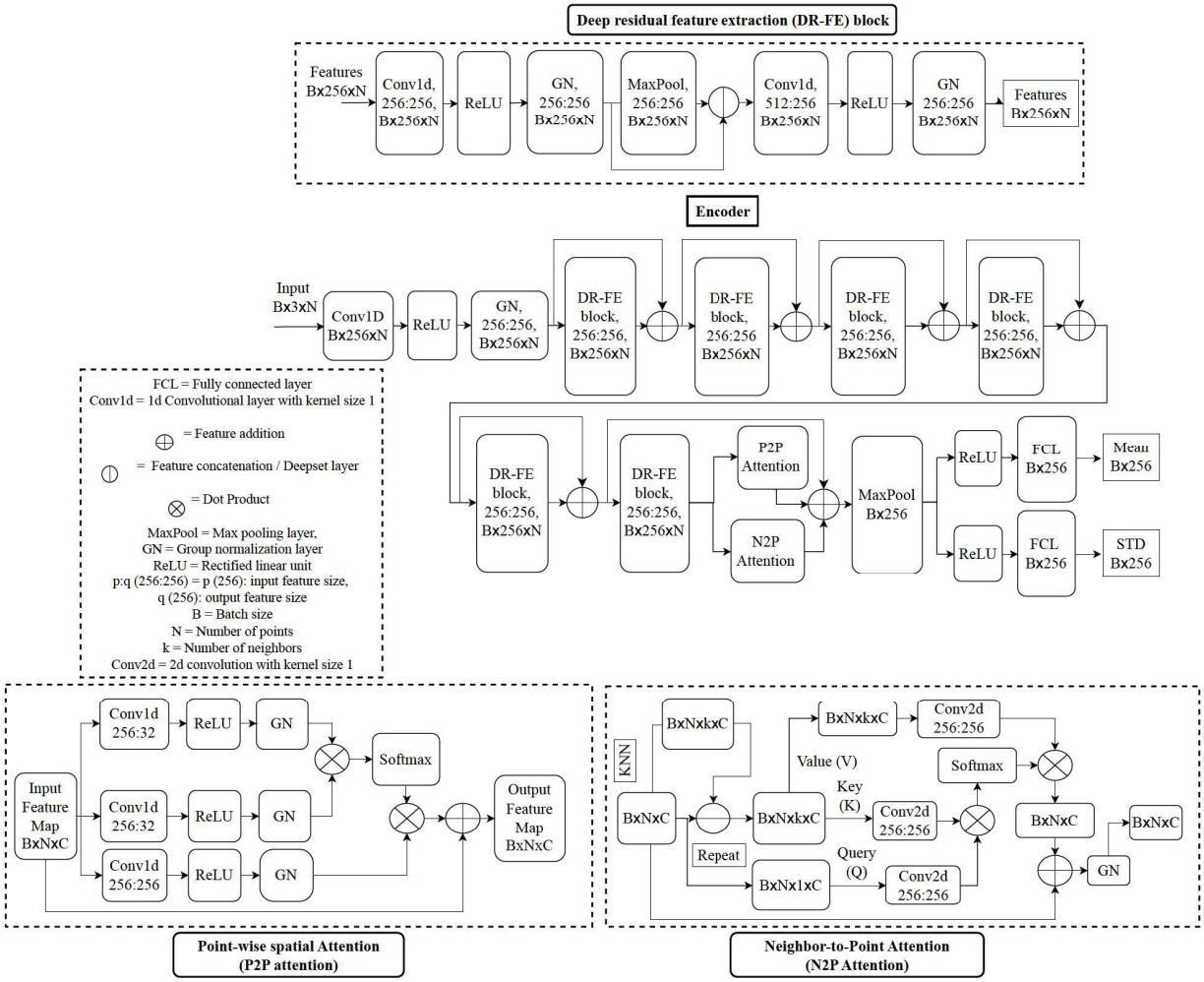

FIGURE 2. **The proposed ADR-SALD encoder consists of six deep residual feature extraction (DR-FE) blocks, two self-attention layers, one max-pooling layer, and two fully connected layers. Each DR-FE block consists of two 1D convolutional layers, two ReLU activation functions, two group normalization layers, one max-pooling layer and one deepset layer. The point-wise spatial attention (P2P attention) and neighbor-to-point attention (N2P attention) layers have been adopted from the studies [26], [27] with minor adjustment: the batch normalization [28] layer was replaced by group normalization [29]. The N2P attention layer was designed similar to the multi-headed attention layer, where the Q, K, and V represent the query, key and value, respectively. The input and output features of each DR-FE block are added together before feeding to the next DR-FE block similar to the feature addition strategy in residual networks [22]. Here, k, and K in the N2P attention layer stand for number of neighbours and Key, respectively.**

also performs the surface reconstruction based on global smoothness priors similar to SPSR and leverages radially symmetric basis functions to approximate the surfaces. Moving least squares [11] (MLS) is a mesh independent method of surface interpolation or approximation based on moving least square approximation. MLS works based on a projection strategy for surface interpolation, which includes a local reference and a moving least square approximation with respect to the reference domain. On the other hand, the ball pivoting [10] algorithm computes the triangle mesh by formulating a triangle using three points from a point cloud if a ball of user defined radius touches them. The ball revolves around the edges to form triangles until it reaches to the point where there is no more point to construct the triangle.

B. CLASSICAL REPRESENTATION LEARNING-BASED METHODS

3D data-driven based representation learning can be categorized into following three types: (a) voxel-based, (b) point-based and (c) mesh-based representation learning.

Voxels are a non-parametric volumetric representation that represents the data in a regular 3D grid. It is one of the most popular choices for data-driven representation learning as it allows a natural extension from learning-based algorithms that have succeeded in the 2D image domain. In 3D representation learning [31], [32], [33], voxels are usually used to learn dense occupancy values where the occupancy grid (occupied or unoccupied) can represent the boundary information of a curve or a surface. Several data-driven 3D representation methods [3], [19] have proposed to

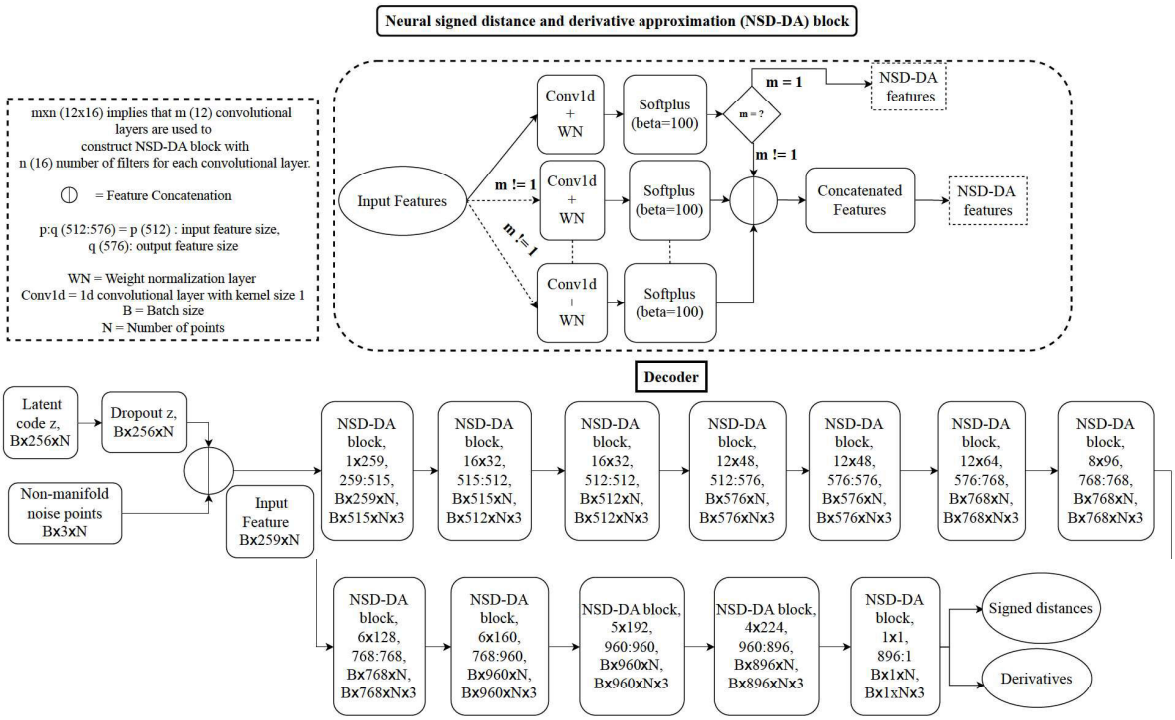

FIGURE 3. The proposed ADR-SALD decoder architecture consists of 12 neural singed distance and derivative approximation (NSD-DA) blocks. NSD-DA block is designed in this study inspiring from inception layer [23] and ResNeXt layer [24], however, it is significantly different from the inception and ResNeXt layers. DSD-DA block is constructed with only 1D convolutional with 1×1 kernel, where the inception layer has various types of kernel size for example 1×1, 3×3, 5×5 with max-pooling layer, and ResNeXt layer also has different types of kernel size with shortcut connection. In the NSD-DA block, m number of parallel convolutional layers followed by a weight normalization, and Softplus activation layers are used to extract the deep features. If m is greater than 1, the extracted features were concatenated in the channel dimension and fed to next NSD-DA block. Moreover, for each m^{th} layer of NSD-DA block has n number of filters, where input features size is p and the concatenated output feature size is q. NSD-DA block learns from the latent code concatenated with a noise points and outputs the signed distance and their derivatives.

learn the occupancy values and render the mesh surface using various rendering algorithms [25]. However, the memory requirements of voxel-based 3D data representation increases cubically, therefore, its usages are limited. Moreover, the voxel based approaches do not preserve the sharp details of the shape, failing to produce high fidelity reconstruction [34], [35], [36].

Point cloud-based 3D representation learning is also a popular choice in computer vision, graphics and robotic communities due to its lightweight nature and availability as a raw data representation format from various sensors, such as LiDARs, and depth cameras. PointNet [37] and PointNet++ [38] are two pioneering works which popularized deep learning on point clouds. These works use max-pooling to extract global shape features that are utilized for classification, segmentation and encoding of 3D surfaces for generative models. However, point clouds as such are not suitable for producing watertight surfaces for rendering.

Mesh-based 3D presentation carries more intuitive information than voxels or point clouds. Meshes provide connectivity between the 3D points and can be used to directly regress the vertices and faces using neural networks [15], [34], [39], [40]. However, those methods lack surface continuity and sometimes result in self-intersecting mesh faces. The shape can be inferred by deforming template using mesh-based methods [41], [42], [43], [44], [45], however, those methods are known to be restricted to a single topological representation. Several other methods [46], [47], [48] have proposed using meshes to perform various tasks, such as classification and segmentation. The output of implicit neural representation-based methods [2], [1], [3], [5], [16], [18] can also be the mesh format, which allows rendering and texture mapping using established graphics hardware and software.

C. IMPLICIT NEURAL REPRESENTATION LEARNING-BASED METHODS

In implicit representation learning-based methods, the 3D surface of an object or scene is modeled as zero level sets of a neural function (see Equation 1). Implicit neural functions can be modeled in the following two approaches: (a) predefined volumetric grid values (occupied or unoccupied) in the form of $f(x, z) : \mathbb{R}^3 \times \mathbb{Z} \longrightarrow [1, 0]$ using neural networks [3], [19], or (b) continuous volumetric function learning using generative models in the form $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$, such as a multi-layer perceptron. The continuous volumetric function can be trained with signed distances $f(x, z) : \mathbb{R}^3 \times \mathbb{Z} \longrightarrow \mathbb{R}$ or unsigned distances $f(x, z) :$

$\mathbb{R}^3 \times \mathbb{Z} \longrightarrow \mathbb{R}_0^+$. Training supervision for implicit function learning can be performed by regressing the known pre-computed occupancy functions [3], [18], signed distance functions [5], particle methods [49], and in a sign agnostic manner [1], [2]. In this study, we use the sign agnostic loss to train our methods.

D. NETWORK DESIGN AND FEATURE LEARNING

Feature extraction and stable training of deep neural networks greatly relies on carefully designed of neural architectures. Deep networks have shown improved performance over their shallow counterparts in computer vision related application. In the recent years, there have been several important advances in the field of neural architectures [22], [23], [50], [51], [52], [53], [54] that improve the performance of computer vision tasks. Out of these, residual networks [22] is a very deep neural architecture designed with shortcut connections that have shown to provide significant improvement in feature extraction, and the proposed *ADR-SALD* encoder architecture has taken inspiration from residual networks [22].The residual neural architecture mitigates the vanishing gradient problem and makes training more stable. On other hand, the decoder of ADR-SALD has similarity with inception layers proposed in [23]. The motivation to adopt inception-like structure is to enable learning simultaneously with special geometric initialization and extracting different types of semantic contextual features associated with input point clouds.

The pioneering attention-based feature learning strategy proposed in [55] has significantly improved the performance of deep learning model in various application. Superior learning capability of the attention layer has inspired to introduce this mechanism to 3D feature learning domain for various application [26], [27], [56]. In the case of self-attention mechanism, the same local feature is transformed into a query (Q), key (K) and value (V) using three multilayer perceptron (MLP) layers, where all these transformed features are from some dimension d_k. Using Q, K and V, the following expression is formulated to construct the weighted attention feature map:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt(d_k)})V \tag{2}$$

Here, the spatial correlation map is computed between Q feature map and inverse of K feature map. Softmax operation is performed to normalize the spatial attention map. The weighted attention feature map is computed by taking a matrix multiplication between the spatial correlation map and the feature map V. Finally, the weighted attention feature map is multiplied with the scaling parameter $\frac{1}{\sqrt{d_k}}$ to counter the small gradients problem [55] introduced by the Softmax function.

On the other hand, applying attention function in parallel the multi-headed attention function [55] helps the model to simultaneously focus on learning information located in different positions of input data. It has shown superior performance over single attention function. More detail information about multi-headed attention function can be found in this study [55].

III. PROPOSED ADR-SALD ARCHITECTURE

In this section, we describe the proposed ADR-SALD architecture for implicit representation learning. The ADR-SALD *encoder* consists of multiple Deep Residual Feature Extraction (DR-FE) blocks with dual self-attention layers, whereas the ADR-SALD *decoder* is constructed from multiple Neural Signed Distance and Derivative Approximation (NSD-DA) blocks. In the ADR-SALD encoder and decoder architecture, 1D convolutional layers with 1×1 kernels are used with zero padding and stride of 1. For simplicity, we refer to these as 1D convolutional layers in the following sections — the kernel shape, padding and stride will remain same. The proposed ADR-SALD encoder and decoder architectures are described in detail in the following sections, and illustrated in Fig. 2 and Fig. 3.

A. ENCODER

In the encoder, the input data $\mathbf{X} \in \mathcal{X}_i$ is fed to a 1D convolutional layer followed by a ReLU [57] activation function and a group normalization [29] layer. Consequently, 6 DR-FE blocks follow along with two self-attention layers: (a) point-wise spatial attention [27] (P2P attention) layer and (b) neighbor to point attention [26] (N2P attention) layer. The attention layers are followed by a MaxPooling layer, ReLU activation function, and finally two fully connected layers are used to construct a probability measure $\mathcal{N}(\mu, \Sigma)$ to create the compact latent representation of the input point clouds. Here, μ and η are two latent vectors (μ, η) extracted from two fully connected layers where μ is used as a mean value of the probability measures and the diagonal covariance matrix, $\Sigma = \text{diag exp } \eta$ is computed using η. Therefore, the ADR-SALD encoder $(\mu, \eta) = g_E(\mathbf{X}, w_1)$ takes $\mathbf{X} \in \mathbb{R}^3$ as input raw point cloud and returns two 256 dimensional latent vectors, $\mu \in \mathbb{R}^{256}$, and $\eta \in \mathbb{R}^{256}$. The complete encoder architecture with its sub-blocks are shown in Fig. 2.

Each DR-FE block is constructed as follows: 1D convolutional layer $\rightarrow$ ReLU activation function $\rightarrow$ Group Normalization layer $\rightarrow$ MaxPooling layer $\rightarrow$ DeepSet Layer $\rightarrow$ 1D convolutional layer $\rightarrow$ ReLU activation function $\rightarrow$ Group Normalization layer. The output from the first group normalization layer and the MaxPooling layer are concatenated in the DeepSet layer [58] and fed to the next 1D convolutional layer. The input features of the DR-FE block are propagated to the next DR-FE block through a shortcut connection similar to a residual network [22]. The shortcut connection is shown in DR-FE subfigure within Fig. 2.

In this work, we have adopted two self-attention layers with some modifications from the original studies [26], [27]. The N2P self-attention layer is a multi-headed attention layer which computes correlation feature maps between

each point and its neighbors. The N2P self-attention layer extracts local features and adjusts the latent vector [26]. P2P attention, on the other hand, is applied to extract point-wise global spatial semantic context and to capture long-range global contextual relations of the points. When two points have similar semantic information, they show strong correlation [27]. In both self-attention layers, we replaced the batch normalization [28] layer with a group normalization [29] layer, which behaves like layer normalization layer. We have removed a batch normalization layer and two convolutional layers from original implementation of the N2P self-attention layer. Shortcut connections have also been applied to both P2P and N2P self-attention layers. To reduce parameter counts, the extra convolutional layers were removed from N2P attention layer. Moreover, empirically it was seen that the batch normalization is not suitable for point cloud-based reconstruction method. However, group normalization, instance normalization, and layer normalization can be used to improve the performance of the network.

B. DECODER

The decoder architecture is similar to the vanilla inception layer [23], except that 1D convolutional layers are used to constructed the architecture followed by a weight normalization layer and SoftPlus activation function. The decoder is constructed with 12 NSD-DA blocks, where each NSD-DA block has m 1D convolutional layers followed by a weight normalization layer and a SoftPlus activation function. Here, the minimum and maximum values of m are between 1 and 16. The geometric initialization proposed in [1] is used to initialize the network weight parameters in each decoder layer. The NSD-DA features extracted from the m^{th} layer are processed by 1D convolution $\rightarrow$ weight normalization $\rightarrow$ SoftMax, then concatenated and fed to the next NSD-DA block. There is no shortcut connection inside the decoder NSD-DA blocks, nor between the encoder and decoder architecture. The decoder architecture with its NSD-DA block are shown in Fig. 3.

C. DATA PREPARATION

Both D-Faust and ShapeNet datasets were pre-processed for training and testing in the same manner illustrated in this section and used for both human shape space learning IV-F and object shape space learning IV-G. We precomputed 500k signed agnostic loss (unsigned distances) $\{h(x)\}_{x\in D}$ and their derivatives $\{\Delta_x h(x')\}_{x'\in D'}$ on points to each shape of the data $\mathcal{X} \in \mathbb{R}^3$ using the CGAL library [59], where the sampled data are from some distribution D and D'. The distribution D is chosen by uniformly sampling 250k points $\{y\}$ from $\mathcal{X}$ by placing two isotropic Gaussians, $\mathcal{N}(y, \sigma_1^2 I)$ and $\mathcal{N}(y, \sigma_2^2 I)$ for each $\{y\}$ similar to [2]. Here, the distribution σ_1 is set to be the 50^{th} closest point to y and σ_2 is set to be a fixed value of 0.3. The gradient values $\{\Delta_x h(x')\}_{x'\in D'}$ are computed using automatic differentiation forward mode illustrated in [60] for each shape following the gradient computation strategy explained in [2], where the distribution D' is chosen to be uniform on $\mathcal{X}$. In the case of single sample reconstruction, we set up three isotropic Gaussian parameters, $\mathcal{N}(y, \sigma_1^2 I)$, $\mathcal{N}(y, \sigma_2^2 I)$ and $\mathcal{N}(y, \sigma_3^2 I)$. The first two distribution parameters σ_1, and σ_2 are kept same as before, however, σ_3 is set to a fixed value of 0.5.

D. TRAINING AND INFERENCE

It is assumed in sign agnostic learning that there is a critical neural network weight $w^* \in \mathbb{R}^m$ which can be found by optimizing $f(x; w^*)$ using gradient descent, where $f : \mathbb{R}^3 \longrightarrow \mathbb{R}$. It is also assumed that this critical weight will produce a favorable signed distance function in such a way that it will approximate the surface S to $\mathcal{X} \in \mathbb{R}^3$ where $\mathcal{X} \in \mathbb{R}^3$ is sampled from [3]. In the *ADR-SALD* encoder, the input raw data, $\mathcal{X} \in \mathbb{R}^3$ is encoded to probability measures (latent code z) using Equation 3.

$$z = \mathcal{N}(\mu, \Sigma), \tag{3}$$

where $\Sigma = \text{diag Exp}(\eta)$.

We use the same sign agnostic loss function with derivative as proposed in [2] to train our *ADR-SALD* architecture shown in Equation 4.

$$\begin{aligned} L_{SALD}(w) = \mathbb{E}_{x\sim D}\tau(f(x; w), h(x)) \\ + \lambda\mathbb{E}_{x\sim D'}\tau(\Delta_x f(x; w), \Delta_x h(x)) \end{aligned} \tag{4}$$

where τ is an unsigned similarity function ensuring the sign agnostic and monotonic nature of learning [1]. Moreover, $h(x)$ and $\Delta_x h(x)$ are the pre-computed sign agnostic loss (unsigned distances) and gradient values (normals).

We replace the latent regularization terms used in [2] with Kullback-Leibler divergence (KLD) loss to optimize the latent code z by adding this loss term in Equation 4. KLD tries to minimize the difference between the approximate posterior data distribution and input prior data distribution.

$$KLD_{latent} = -0.5 \times (\exp(\eta) + \mu - \exp(\exp(\eta))) \tag{5}$$

Moreover, the following square norm of the latent code z is also added with Equation 4.

$$NS_{Latent} = ||z||_2^2 \tag{6}$$

Therefore, we try to optimize the following loss function shown in Equation 7 during training of our proposed *ADR-SALD*.

$$Loss(w) = L_{SALD} + KLD_{Latent} + NS_{latent} \tag{7}$$

All the experimental models of the proposed *ADR-SALD* were trained with the Adam [61] optimizer with an initial learning rate of 5e-4 and 2000 epochs. The learning rate was reduced by 15% after every 500 epochs. Moreover, the latent code was shrinked by 5%, 3%, and 2% in the range of 0 to 500 epochs, 501 to 800 epochs and 801 to 1500 epochs, respectively. Latent dropout reduces

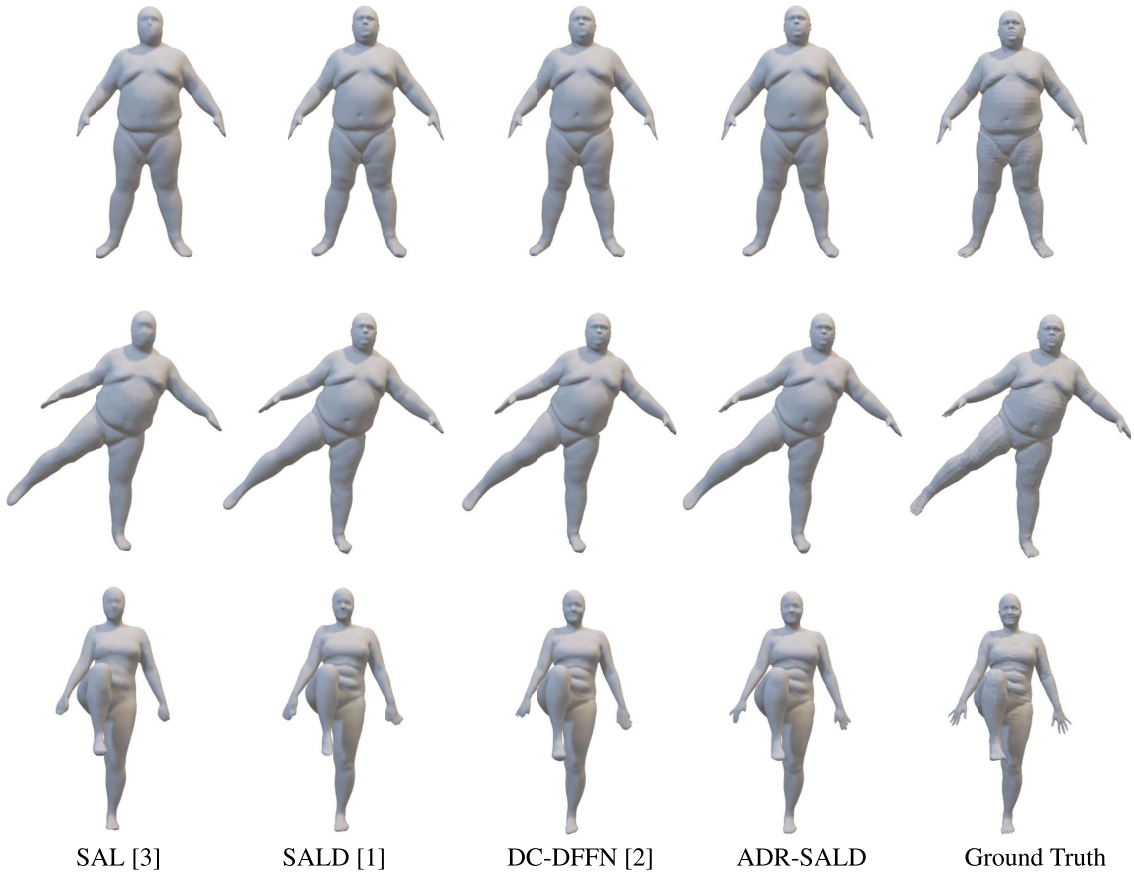

FIGURE 4. The qualitative results for ADR-SALD and the baseline methods for the experiment of human shape space learning on the D-Faust dataset. It can be seen from the reconstructed mesh that ADR-SALD can capture more small detail (overall face especially lip shape, eyes, facial expression, and belly button) than the baseline state-of-the-are approaches.

TABLE 1. D-Faust quantitative results. The chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean+std scores, as well as chamfer distances have been multiplied by 10^3. ↓ means a lower value is better. The result is calculated for 1997 samples out of 2038 test samples. The remaining samples have extra shape which does not match with the ground truth. The example of such shape can be seen in Fig. 7 (row 1).

Dataset: experiment	Method	Direction	Percentile (↓)			Mean ± STD (↓)
			5%	50%	95%	
D-Faust [62]: Human shape space learning	SAL [3]	Rg→Gn	0.035	0.063	0.190	0.093 ± 0.157
		Sc→Gn	0.025	0.037	0.111	0.054 ± 0.131
	SALD [1]	Rg→Gn	0.036	0.074	0.394	0.131 ± 0.209
		Sc→Gn	0.026	0.048	0.287	0.092 ± 0.155
	DC-DFFN [2]	Rg→Gn	0.028	0.052	0.188	0.086 ± 0.136
		Sc→Gn	0.022	0.033	0.118	0.050 ± 0.073
	ADR-SALD (proposed)	Rg→Gn	**0.027**	**0.051**	**0.149**	**0.074 ± 0.097**
		Sc→Gn	**0.021**	**0.031**	**0.084**	**0.043 ± 0.067**

the chances of over-fitting the decoder and introduces additional expressiveness to the decoder to properly learn and approximate surfaces. As usual, dropout is not present during inference.

During evaluation, the trained model approximates the signed distances for the test samples, which are later used to generate the output mesh using the Marching Cubes algorithm [25]. The resolution of the output mesh is set to a value of 100^3 for the test samples in all experiments.

IV. EXPERIMENTAL RESULTS

In this section, we show the experimental results for *ADR-SALD* on two challenging benchmark datasets: (a) Dynamic Faust [62] and (b) ShapeNet [35], and compare the results to state-of-the-art research works [2], [17], [1].

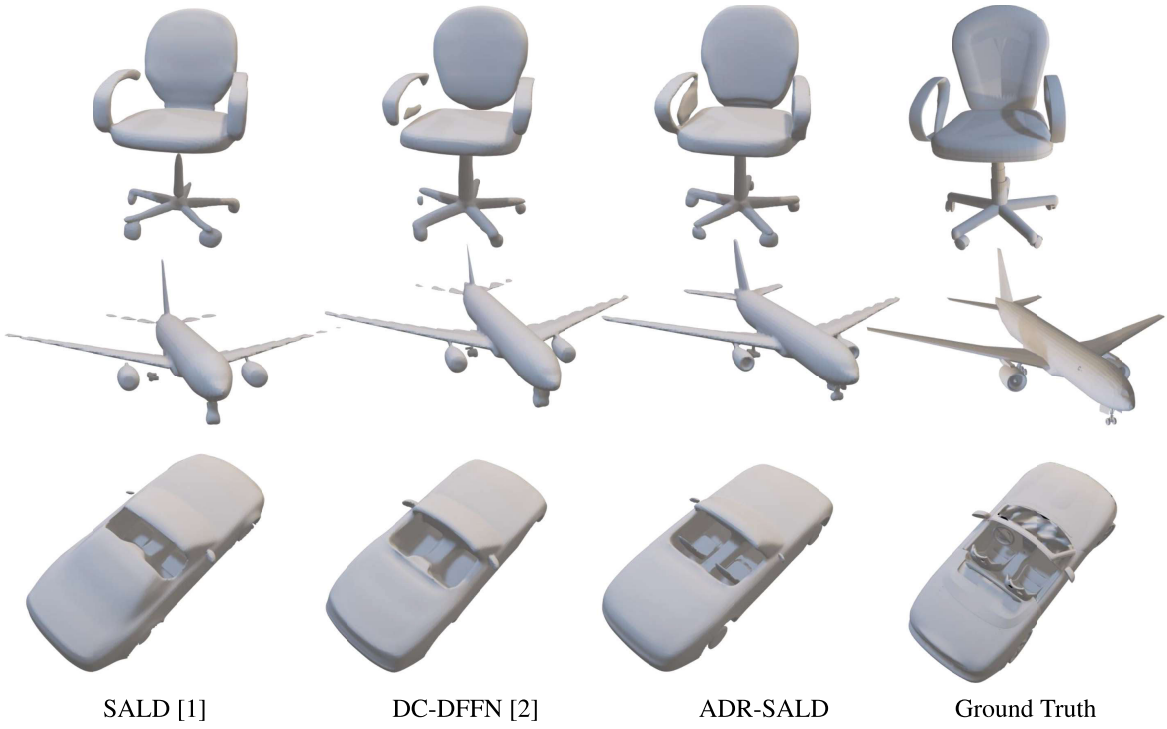

FIGURE 5. Qualitative results for object shape space learning for the ShapeNet dataset. The proposed method successfully reconstructs car seats, both airplane engines and chair handles, whereas the baseline approaches are struggling with reconstructing these details.

TABLE 2. ShapeNet quantitative results. The chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean+std scores as well as chamfer distances have been multiplied by 10^3. ↓ means lower is better.

Dataset: experiment	Class	Method	Percentile (↓)			Mean ± STD (↓)
			5%	50%	95%	
ShapeNet [63]: Object shape space learning	Car	SALD [1]	0.098	0.244	0.734	0.317 ± 0.276
		DC-DFFN [2]	0.140	0.329	0.778	0.383 ± 0.241
		ADR-SALD (proposed)	**0.080**	**0.148**	**0.460**	**0.190 ± 0.147**
	Chair	SALD [1]	0.034	0.215	1.459	0.429 ± 0.649
		DC-DFFN [2]	**0.019**	0.148	1.277	0.360±0.708
		ADR-SALD (proposed)	0.021	**0.120**	**0.864**	**0.260±0.482**
	Airplane	SALD [1]	0.011	0.057	0.518	0.143± 0.323
		DC-DFFN [2]	**0.008**	**0.044**	0.464	0.122± 0.203
		ADR-SALD (proposed)	0.009	0.046	**0.447**	**0.116± 0.215**

A. DYNAMIC FAUST DATASET

The Dynamic Faust [62] (D-Faust) dataset contains approximately 41000 raw scans of human subjects expressed as triangle soups. The subjects consist of 5 males and 5 females in various poses while performing 129 different actions. A 4D scanner was used to capture the subjects at 60 frames per second. Common defects that can be found in the D-Faust scans are noise, missing body parts, holes, and ghost geometry [1]. More details about data capture related information can be found in [62]. Because of dense temporal sampling, 1 out of 5 samples are used in our experiments similar to the experimental setup used in [1], with the same data split files.

B. SHAPENET DATASET

ShapeNet [63] is a large-scale repository of more than 3,000,000 3D CAD models, of which 220,000 models have been classified into 3,135 categories. A subset of the ShapeNet dataset is the ShapeNetCore dataset, which contains manually verified and alignment-annotated 55 common objects as clean 3D models. More detailed information can be found in [63]. Out of 55 classes, we consider 3 objects classes in our shape space learning experiments: car, airplane, and chair classes have 7497, 4045, and 6778 samples respectively. We use 75 percent of samples for training and 25 percent of the samples for evaluating the trained models. The train and test split files were created locally and used to train

and evaluate all the models presented in this study for the ShapeNet experiment.

Additionally, the authors of the work 3D-R2N2 [64] have processed the ShapeNet dataset and prepared watertight point clouds, which we use to train and test single sample reconstruction (Section IV-E) for *bench, gun, lamp and boat* classes. From bench, lamp and boat classes, four point cloud samples were separately trained and mesh files were reconstructed for qualitative and quantitative evaluation, whereas for the gun class three samples were used.

C. METRICS

The main metric for assessing the performance of ADR-SALD against previous state-of-the-art approaches is **chamfer distance**, which is a distance measure between two sets of data points. In this study, we use chamfer distance to measure similarity between the model-generated mesh and the ground truth mesh. The reported measurements are the mean distance in single direction (Rg→Gn), i.e., from the ground truth mesh (Rg) to the generated mesh (Gn) for D-Faust and ShapeNet datasets. Moreover, one directional mean chamfer distance is also computed and reported for raw input scans (Sc) against the generated meshes (Sc→Gn) in the case of D-Faust dataset. In D-Faust dataset, the scan file (triangle soups) and ground truth registration are two separate files. The training and test data are not prepared from the ground truth registrations. The ground truth registrations are only used to evaluate the reconstruction. One the other hand, training and test samples are prepared from the ground truth mesh in ShapeNet dataset and evaluated against the same mesh object files.

D. BASELINES

As baselines, we only consider approaches that can be trained directly on raw point clouds without doing any transformation into regular grid data formats, such as voxels. The works [2], [17], [1] can directly process the raw point clouds and reconstruct mesh files. Moreover, they use neural implicit representations for training, similar to the proposed approach.

1) SAL

SAL [1] proposes a sign agnostic learning-based training approach for implicit surface representation without normals. SAL learns unsigned distances during training and predicts signed distances during inference. SAL has a limitation in reconstructing thin structure. We compare *ADR-SALD* against SAL using D-Faust and ShapeNet datasets.

2) SALD

SALD [2] is a generative model which learns unsigned distances with their derivatives in a sign agnostic manner. However, the derivatives are only used to train the model. During inference, the model predicts the signed distances which later are used to reconstruct the surface using the marching cubes algorithm. The proposed *ADR-SALD* is compared against SALD model for both D-Faust and ShapeNet dataset.

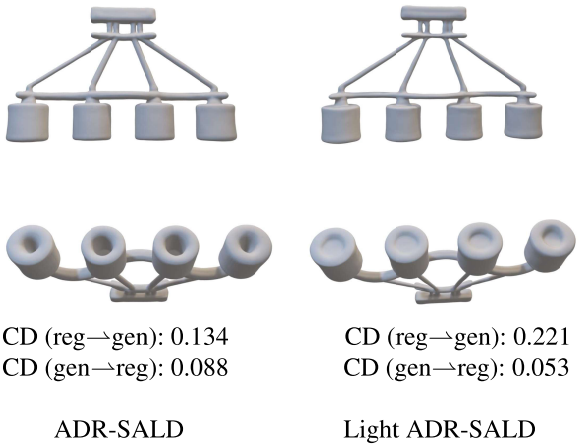

FIGURE 6. The ADR-SALD and Light ADR-SALD qualitative and quantitative comparison. ADR-SALD captures more detail than Light ADR-SALD, however, Light ADR-SALD is computationally more efficient and faster both in training and test time reconstruction. Here, CD stands for chamfer distance.

3) DC-DFFN

In the DC-DFFN [17] work, the authors have proposed using a densely connected feature fusion-based neural architecture for implicit surface reconstruction. In this study, both the encoder and the decoder have dense feature connections which prevents gradient vanishing and promotes learning rich features during training and test phases. Shortcomings of DC-DFFN are related to large structural gaps in the samples and capturing of small details. We compare the proposed *ADR-SALD* method against DC-DFFN for both D-Faust and ShapeNet datasets.

4) IMPLICIT FILTERING

In Implicit Filtering [65], the authors proposed a method to smooth the implicit field based on a non-linear implicit filter. It can preserve the high-level geometric details by filtering the surface, which is achieved by moving the input point clouds along the gradient of the signed distance field. Although this approach can achieve a high degree of geometric detail, it can miss parts of the object or scene and can introduce excess shape.

5) UNSUPERVISED OCCUPANCY

In Unsupervised Occupancy [66], the authors have proposed a method to estimate the occupancy field from a very sparse point cloud instead of using signed distance functions. The authors used margin-based uncertainty measures to sample points differentially from the occupancy function, which are supervised by the input point cloud. The method learns the occupancy field by minimizing the distance between uncertain sample points and their nearest point cloud samples. Although this study is designed to reconstruct from very sparse point clouds, it can handle dense point clouds as well.

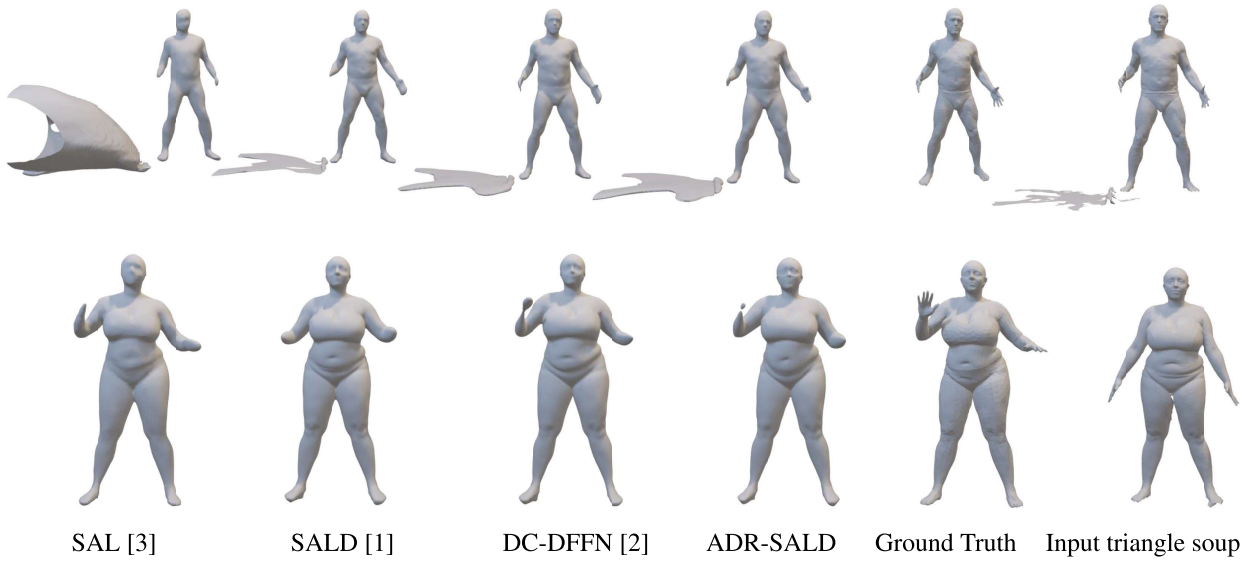

FIGURE 7. Reconstruction defects by the proposed ADR-SALD and baseline methods in human shape space learning on the D-Faust dataset. For a few cases all the experimental methods suffer in reconstructing the human hands during inference.

FIGURE 8. Hallucination in the case of reconstructing thin structure and separating small gaps. Here, all the methods reconstruct the object towards a different shape than the ground truth (incorrect number of spokes in the chair back support). This issue remains open for future study. These defected reconstructions are from object shape space learning.

The usual defects of this approach include missing parts, and producing grained surfaces.

E. SURFACE RECONSTRUCTION

In this experiment, we consider individual complex shapes of objects and reconstruct the implicit neural surfaces for them. Given a single input raw point cloud, triangle soup or un-oriented mesh $\mathcal{X} \in \mathbb{R}^3$, we want to estimate the approximate surface S of $\mathcal{X}$. As explained in Section III-C, after uniformly sampling 250k points, three isotropic Gaussians $\mathcal{N}(y, \sigma_1^2 I)$, $\mathcal{N}(y, \sigma_2^2 I)$ and $\mathcal{N}(y, \sigma_3^2 I)$ are set up to compute 750k unsigned distances and their derivatives using the CGAL library [59]. The distribution parameters σ_1, σ_2, and σ_3 are chosen according to Section III-C for single sample reconstruction.

The processed watertight samples of the ShapeNet dataset in the study [64] have only 100k data points. Four isotropic Gaussians $\mathcal{N}(y, \sigma_1^2 I)$, $\mathcal{N}(y, \sigma_2^2 I)$, $\mathcal{N}(y, \sigma_3^2 I)$, and $\mathcal{N}(y, \sigma_4^2 I)$ are set up to compute 400k unsigned distances and their corresponding derivatives. The 4^{th} distribution parameter σ_4 is set to a fixed value of 0.05. The rest of the distribution parameters are kept the same as mentioned earlier.

For single sample surface reconstruction, all models were trained with 16k epochs on the input training sample data except Implicit filtering [65] and Unsupervised occupancy learning [66]. During inference, only the sample data points (90^2) are used to reconstruct shape. The reconstruction result is shown in Fig. 1. It can be seen from the qualitative result that the proposed method can successfully process large gaps, capture small detail and reconstruct high fidelity surfaces whereas the baseline approaches fail in this case.

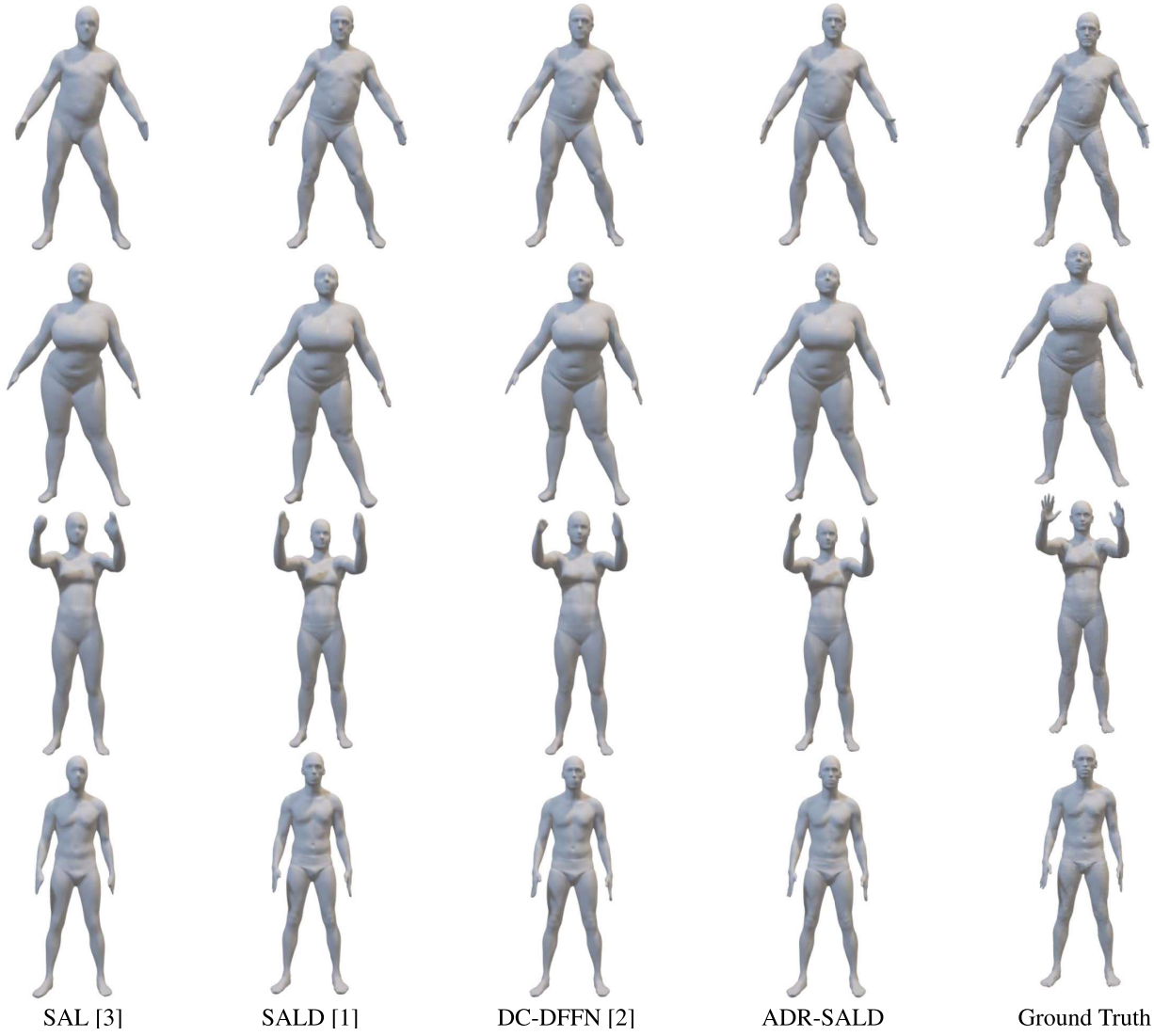

FIGURE 9. Additional results is shown for human shape space learning. The proposed ADR-SALD method can capture better overall small details than the state-of-the-art baseline methods, which can be seen from these qualitative results.

In the case of processed watertight point cloud samples, we compared our approach with two recently proposed methods [65], [66]. These two methods were trained for 40k epochs with the same hyper-parameter settings as mentioned in these studies. The results show that although the proposed approach does not in all cases outperform these recent methods in achieving finest reconstruction detail, the proposed method excels in *consistency*: both the quantitative and qualitative results show that ADR-SALD makes few catastrophic reconstruction mistakes compared to [65] and [66]. The quantitative results are shown in Table 5, and qualitative comparisons are shown in Figs. 12, 13, 14, and 15.

F. HUMAN SHAPE SPACE LEARNING

In this experiment, we selected 1 out of every 5 samples from the 41k D-Faust [62] data samples for training and evaluating the models. The selected data were divided into 75%-25% for training and testing (respectively) the proposed and baseline methods. Experimental data was pre-processed according to the procedure explained in Section III-C. The pre-processed data were used to train the models. Randomly selected 90^2 points with their respective unsigned distances and derivative values were fed to the network to train the models. In the case of SAL, the number of points was 128^2. The output resolution of the reconstructed shape is 100^3 for all architectures.

For this experiment, the quantitative and qualitative results are shown in Table 1 and Fig. 4, respectively. From the quantitative results, it can be seen that the proposed method achieves better or equally good results compared to any of the baseline approaches. Similarly, the qualitative results show that the proposed method can capture small detail, remove empty space and produce high fidelity reconstruction. Additional qualitative results for human shape space learning are shown in Fig. 9.

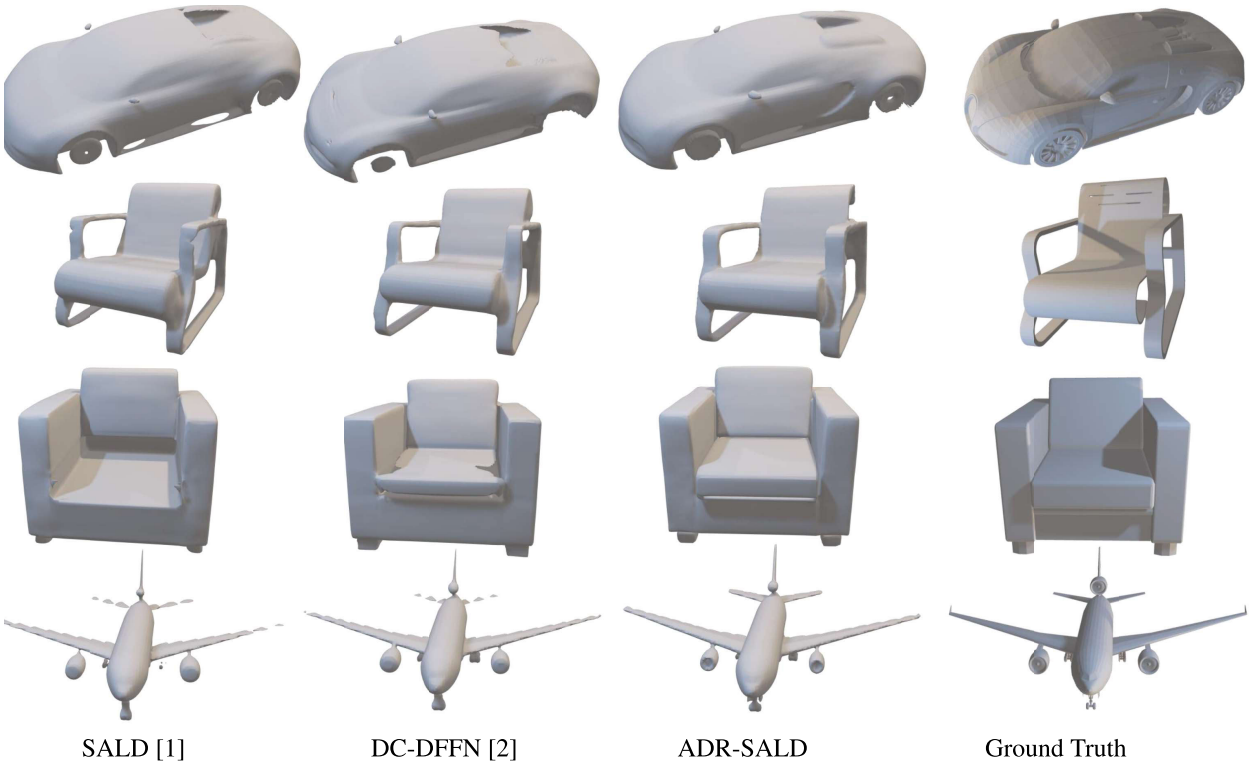

FIGURE 10. Additional qualitative results for object shape space learning – car, chair and airplane classes of the ShapeNet dataset. From the qualitative results, it can be seen that SALD [2] and DC-DFFN [17] struggle to reconstruct the complete shape and sometimes miss parts of the objects. The proposed ADR-SALD method is more resilient and can capture small detail.

G. OBJECT SHAPE SPACE LEARNING

From the ShapeNet [63] dataset, we have selected three classes for this experiment. The object classes are: (a) Airplane (4045 samples), (b) Car (7497 samples) and (c) Chair (6778 samples). Randomly selected 75% of the samples are used to train the models and 25% of the samples were used to evaluate the performance of the proposed and baseline models. Similar to the previous experiment, 90^2 points with their corresponding unsigned distances and derivative values are fed to the network to train the proposed model. The output mesh resolution was 100^3 during inference.

The results shown in Table 2 and Fig. 5 represent the quantitative and qualitative performance of the proposed and baseline approaches. From Table 2, it can be seen that*ADR-SALD* outperforms the baseline architecture in most of the categories. It is also visible from Fig. 5 that the proposed method produces high quality surface reconstruction. Additional qualitative results of test samples of object shape space learning are shown in Fig. 10.

H. ABLATION STUDY

We have shown the impact of the attention layers and KLD latent regularization separately in the following sections.

1) ATTENTION LAYER

We removed both attention layers from the encoder and retrained the remaining network while keeping all the other hyper-parameters same. In this case, we use the ShapeNet car class to train and test the model. The quantitative and qualitative test results are shown in Table 4 and Fig. 11. It can be seen from the qualitative and quantitative result that the attention layers have improved the performance significantly and allows the proposed method to achieve high fidelity reconstruct with small detail.

TABLE 3. Network architecture parameter comparison.

Network	ADR-SALD	Light ADR-SALD	% of param. reduction
Encoder	4'713'216	1'185'152	74.65%
Decoder	5'443'473	2'556'257	53.04%

2) KLD

In this case, we have replaced the KLD latent regularization loss term by the latent regularization loss term used in SALD [2] and trained the network. We also use the same ShapeNet car class train and test samples to conduct this ablation experiment. The qualitative and quantitative test results are shown in Fig. 11 and Table 4. From this experiment, it is clear that KLD has played significant role to improve the reconstruction, consequently the quantitative results have improved and also help to remove the small gaps (see car chair in Fig. 11).

TABLE 4. Ablation study: ShapeNet car class quantitative results. The chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean+std scores as well as chamfer distances have been multiplied by 10^3. ↓ means lower is better. From this experiment, it is clear that both KLD latent regularization loss term and attention layer help to improve the performance of the proposed ADR-SALD architecture.

Dataset: experiment	Class	Method	Percentile (↓)			Mean ± STD (↓)
			5%	50%	95%	
ShapeNet [63]: Object shape space learning	Car	ADR-SALD without Attention layer	0.137	0.327	0.774	0.383 ± 0.269
		ADR-SALD with SALD latent regularization	0.114	0.300	0.809	0.353 ± 0.253
		Proposed ADR-SALD with KLD and attention layer	**0.080**	**0.148**	**0.460**	**0.190 ± 0.147**

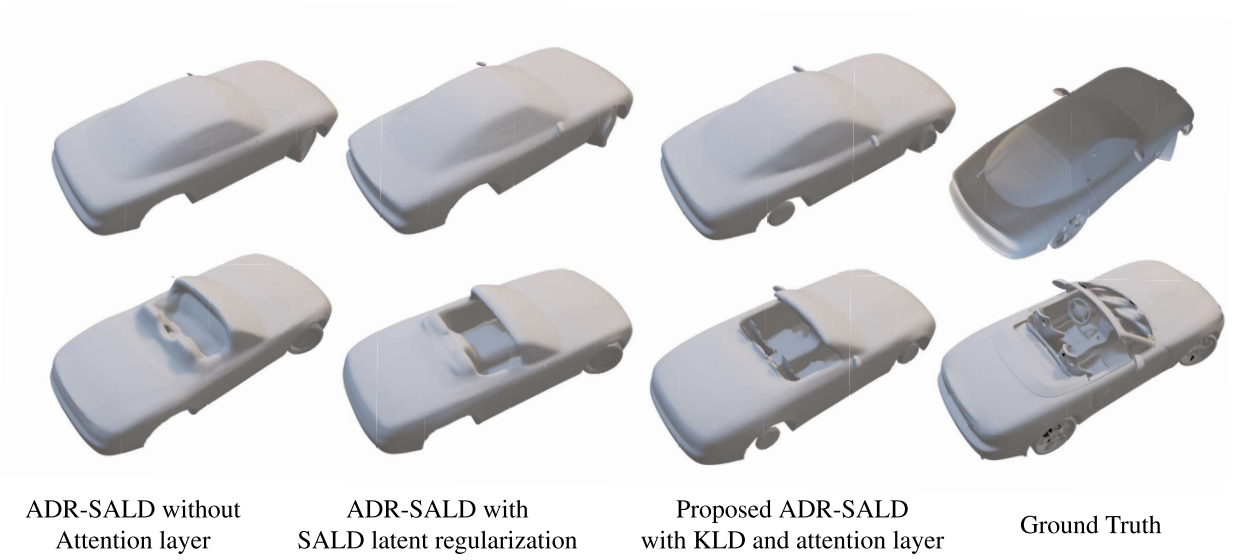

FIGURE 11. Ablation Study: Qualitative results for object shape space learning — car class of the ShapeNet dataset. From the qualitative results, it can be seen that without KLD and attention layers both models struggle to reconstruct the complete shape and sometimes miss parts of the objects. The proposed ADR-SALD method trained with KLD latent regularization and attention layers is more resilient and can capture small detail. Both KLD latent regularization and attention layer are helpful and improve the performance of the model.

TABLE 5. The processed ShapeNet watertight point cloud samples' quantitative results. The chamfer distances are presented in percentiles (5^{th}, 50^{th}, and 95^{th}) and mean+std scores; chamfer distances have been multiplied by 10^3. ↓ means lower is better. For each case, the chamfer distance (first row) is computed from the generated mesh to the input point cloud, whereas in the second row input point cloud to the generated mesh is shown. The noteworthy result is highlighted: the proposed ADR-SALD method excels in robustness: whereas the implicit filter method in the best case produces highest quality reconstructions (5% and 50%), it produces a significant number of catastrophic reconstruction failures (95%).

Dataset: experiment	Class	Method	Percentile (↓)			Mean ± STD (↓)
			5%	50%	95%	
ShapeNet [64]: Single sample surface reconstruction	Car	Implicit Filter [65]	0.003 0.005	0.011 0.020	0.072 5.934	0.022 ± 0.034 1.29±4.75
		Unsupervised Occupancy Learning [66]	0.004 1.266	0.010 0.308	0.019 5.163	0.011 ± 0.005 1.266± 3.629
		Proposed ADR-SALD	0.006 0.013	0.040 0.043	0.332 **0.648**	0.105 ± 0.139 **0.155± 0.318**

I. LIMITATIONS

Although ADR-SALD can capture small detail and reconstruct high fidelity surfaces, it still somewhat lacks in removing small gaps (see Fig. 1) and struggles to reconstruct very thin structure. For example, it can miss thin object structure and parts of the human body (hands or legs) which can be seen in Fig. 8 and Fig. 7, respectively. Moreover, the proposed *ADR-SALD* architecture is computationally costlier than the baseline approaches and requires more time to reconstruct surface during inference.

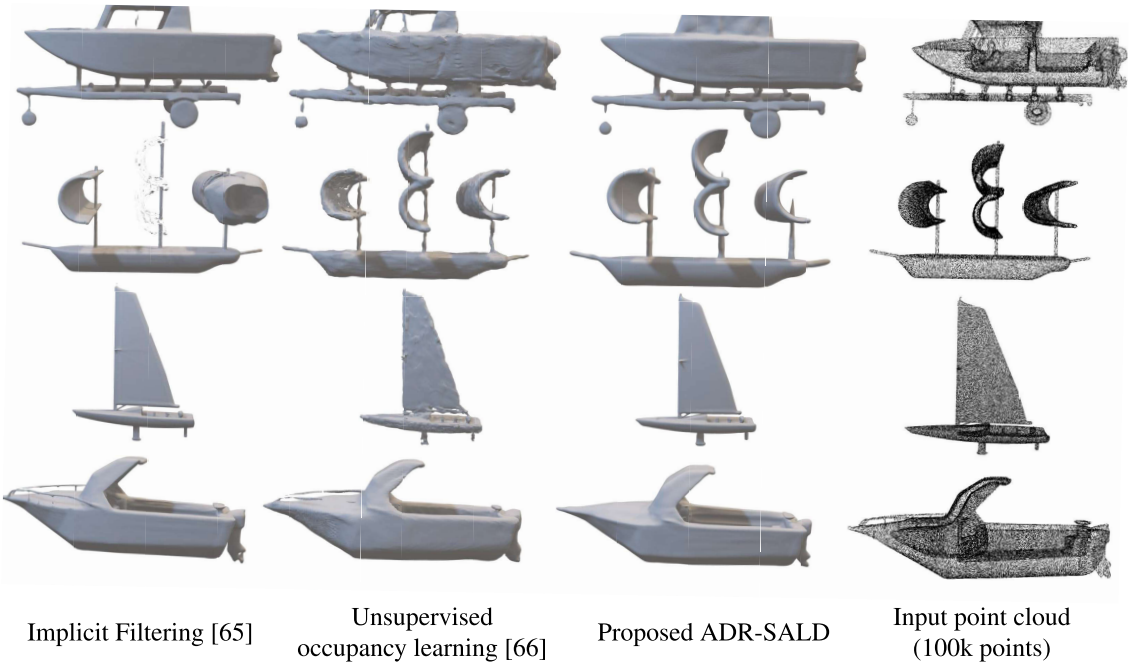

FIGURE 12. Qualitative results for object surface reconstruction – boat class of the processed ShapeNet watertight point cloud samples. From the qualitative results, it can be seen that the proposed ADR-SALD method consistently reconstructs complete and smooth shapes, whereas implicit filtering and unsupervised occupancy learning have significant difficulties with some shapes. ADR-SALD has no significant reconstruction issues, but occasionally misses smaller detail.

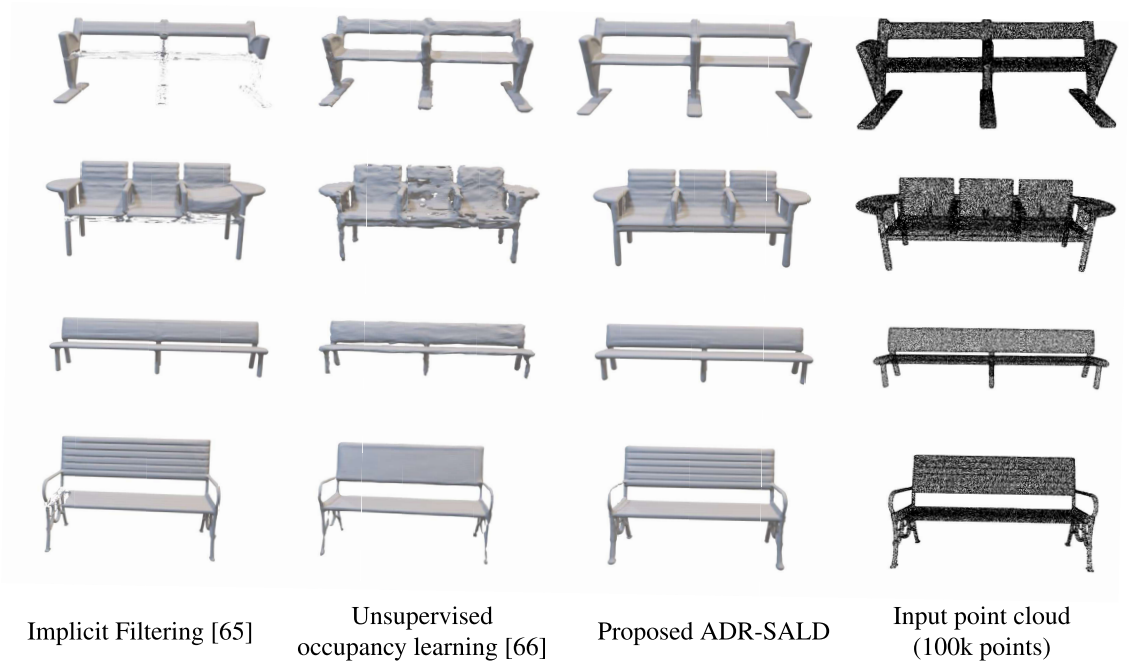

FIGURE 13. Qualitative results for object surface reconstruction – bench class of the processed ShapeNet watertight point cloud samples. It can be seen that the proposed ADR-SALD method consistently reconstructs smooth shape with a significant amount of geometric detail, whereas implicit filtering and unsupervised occupancy learning produce significant artifacts or rough surface, respectively.

J. ARCHITECTURE OPTIMIZATION

The ADR-SALD neural architecture can be optimized to make it computationally more efficient and faster. In the case of ADR-SALD, by reducing the channel count, a 74.65% lighter encoder architecture and 53.04% lighter decoder architecture were developed. The light architecture

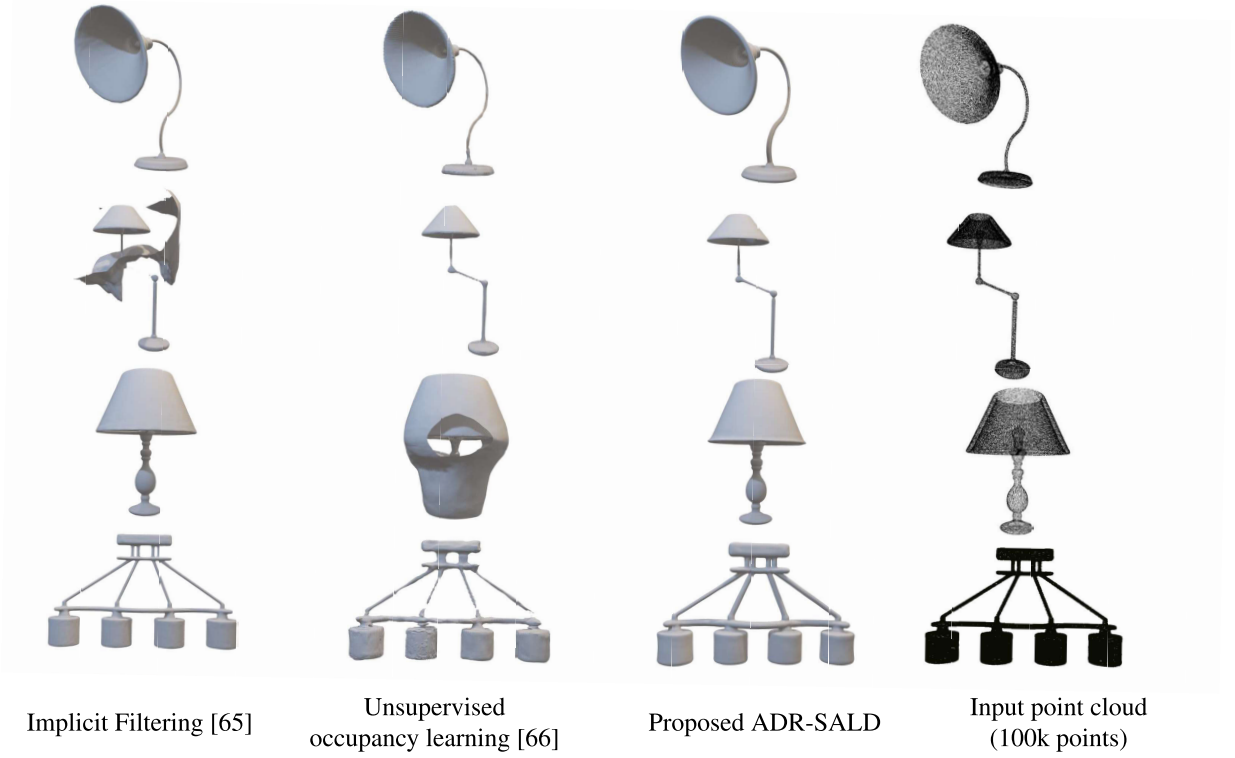

FIGURE 14. Qualitative results for object surface reconstruction — lamp class of the processed ShapeNet watertight point cloud samples. From the qualitative results, it can be seen that the proposed ADR-SALD method consistentlyreconstructs complete and smooth shape, whereas the implicit filtering and unsupervised occupancy learning methods struggle introduce unexpected extra shape with some samples.

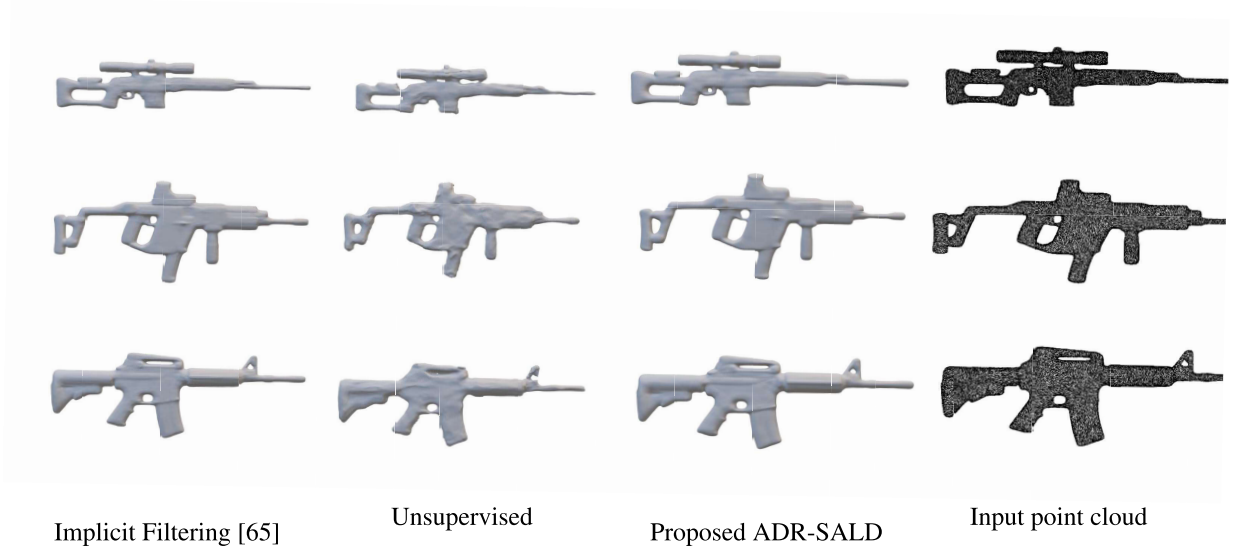

FIGURE 15. Qualitative results for object surface reconstruction — gun class of the processed ShapeNet watertight point cloud samples. It can be seen that the proposed ADR-SALD method and implicit filtering consistently reconstruct good results, whereas unsupervised occupancy learning struggles with surface smoothness.

performs similar to the baseline ADR-SALD architecture in single sample reconstruction. The qualitative and quantitative results are shown in Fig 6, and the parameter counts are shown in Table 3. Beyond channel count reduction, there are some standard approaches such as knowledge distillation, network pruning, and weight quantization that could be applied to further improve the neural architecture's efficiency. However, these approaches are generally only applicable to the trained inference-time network and do not improve training time.

V. DISCUSSION AND CONCLUSION

We have proposed a novel self-attention-based variational autoencoder architecture, *ADR-SALD* for implicit surface representation learning. The objectives of this study are to address the shortcomings mentioned in the baseline methods [2], [17] and generate high fidelity surface reconstruction from raw point clouds. To validate the proposed method and to address those shortcomings, we used two benchmark datasets: (a) D-Faust, and (b) ShapeNet, in three experimental setups: (I) single shape reconstruction, (II) human shape space learning, and (III) object shape space learning. The proposed *ADR-SALD* architecture has shown better or equal quantitative and qualitative performance than the state-of-the-art in all three setups on both datasets.

Importantly, our proposed approach does not introduce surface sheets in the case of large structural gaps (open areas) which is one of the shortcomings mentioned of the baseline methods [2], [17] (see Fig. 1). Moreover, the proposed architecture can better capture sharp detail (see Fig. 4) and is more capable of constructing thin structure (see Fig. 5) than previous works.

From the quantitative results, it can be seen that *ADR-SALD* achieves significantly smaller mean chamfer distances in all categories. The smaller chamfer distance signifies that our method generalizes the test data better than the baseline methods. Moreover, it also highlights that the complex structure reconstruction ability of our method is superior compared to the baseline methods, and can generate high fidelity reconstruction with sharp small detail.

Similar to the baseline methods, our method also has some shortcomings. *ADR-SALD* is computationally more expensive and requires more reconstrution time than the baseline method. Moreover, it struggles to reconstruct extremely thin structure (see Fig. 8), and sometimes misses the parts of the shape (see Fig. 7). These issues can be addressed in the future study by improving the latent representation of the input shape.

Finally, we demonstrated that our method can generalize the test data better than the baseline method, therefore, *ADR-SALD* has more expressive power in terms of network architecture. Furthermore, *ADR-SALD* has outperformed the baseline in almost all categories quantitatively and produces superior reconstruction with sharp small detail.

REFERENCES

[1] M. Atzmon and Y. Lipman, "SALD: Sign agnostic learning with derivatives," in *Proc. Int. Conf. Learn. Represent.*, Jan. 2020, pp. 1–14.
[2] A. Basher and J. Boutellier, "DC-DFFN: Densely connected deep feature fusion network with sign agnostic learning for implicit shape representation," *IEEE Access*, vol. 11, pp. 46399–46412, 2023.
[3] 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. 2562–2571.
[4] 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.
[5] J. Chibane, M. A. Mir, and G. Pons-Moll, "Neural unsigned distance fields for implicit function learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, Jan. 2020, pp. 21638–21652.
[6] 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.
[7] 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*.
[8] 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*.
[9] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc. 4th Eurograph. Symp. Geometry Process.*, Jun. 2006, pp. 61–70.
[10] M. Kazhdan and H. Hoppe, "Screened Poisson surface reconstruction," *ACM Trans. Graph. (ToG)*, vol. 32, no. 3, pp. 1–13, Jun. 2013.
[11] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, and G. Taubin, "The ball-pivoting algorithm for surface reconstruction," *IEEE Trans. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 349–359, Oct. 1999.
[12] D. Levin, "Mesh-independent surface interpolation," in *Geometric Modeling for Scientific Visualization*. Cham, Switzerland: Springer, 2004, pp. 37–49.
[13] 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.
[14] H. Ben-Hamu, H. Maron, I. Kezurer, G. Avineri, and Y. Lipman, "Multi-chart generative surface modeling," *ACM Trans. Graph. (TOG)*, vol. 37, no. 6, pp. 1–15, Nov. 2018.
[15] J. Wu, C. Zhang, T. Xue, W. T. Freeman, and J. B. Tenenbaum, "Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, Jan. 2016, pp. 1–9.
[16] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mache approach to learning 3D surface generation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 216–224.
[17] A. Basher, M. Sarmad, and J. Boutellier, "LightSAL: Lightweight sign agnostic learning for implicit surface representation," 2021, *arXiv:2103.14273*.
[18] S. Peng, M. Niemeyer, L. Mescheder, M. Pollefeys, and A. Geiger, "Convolutional occupancy networks," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2020, pp. 523–540.
[19] J. Chibane and G. Pons-Moll, "Implicit feature networks for texture completion from partial 3D data," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2020, pp. 717–725.
[20] J. Ye, Y. Chen, N. Wang, and X. Wang, "GIFS: Neural implicit function for general shape representation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 12819–12829.
[21] A. Gropp, L. Yariv, N. Haim, M. Atzmon, and Y. Lipman, "Implicit geometric regularization for learning shapes," in *Proc. Mach. Learn. Syst.*, Jan. 2020, pp. 3569–3579.
[22] 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.
[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[24] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.
[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] C. Wu, J. Zheng, J. Pfrommer, and J. Beyerer, "Attention-based point cloud edge sampling," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jan. 2023, pp. 5333–5343.
[27] M. Feng, L. Zhang, X. Lin, S. Z. Gilani, and A. Mian, "Point attention network for semantic segmentation of 3D point clouds," *Pattern Recognit.*, vol. 107, May 2020, Art. no. 107446.
[28] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2015, pp. 448–456.
[29] Y. Wu and K. He, "Group normalization," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Jan. 2018, pp. 3–19.

[30] 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. 10251–10260.

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

[32] G. Varol, D. Ceylan, B. Russell, S. Yan, E. Yumer, I. Laptev, and C. Schmid, "BodyNet: Volumetric inference of 3D human body shapes," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2018, pp. 20–38.

[33] 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. 7738–7748.

[34] 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.

[35] 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.

[36] 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. 209–217.

[37] 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. 77–85.

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

[39] G. Gkioxari, J. Johnson, and J. Malik, "Mesh R-CNN," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9784–9794.

[40] 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*.

[41] C.-H. Lin, O. Wang, B. C. Russell, E. Shechtman, V. G. Kim, M. Fisher, and S. Lucey, "Photometric mesh optimization for video-aligned 3D object reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 969–978.

[42] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y. Jiang, "Pixel2Mesh: Generating 3D mesh models from single RGB images," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2018, pp. 52–67.

[43] A. Kanazawa, S. Tulsiani, A. A. Efros, and J. Malik, "Learning category-specific mesh reconstruction from image collections," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2018, pp. 386–402.

[44] A. Ranjan, T. Bolkart, S. Sanyal, and M. J. Black, "Generating 3D faces using convolutional mesh autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, Jul. 2018, pp. 704–720.

[45] G. Pons-Moll, J. Romero, N. Mahmood, and M. J. Black, "Dyna: A model of dynamic human shape in motion," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 1–14, 2015.

[46] 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.

[47] 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, Jul. 2017.

[48] 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.

[49] M. Atzmon, N. Haim, L. Yariv, O. Israelov, H. Maron, and Y. Lipman, "Controlling neural level sets," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Jan. 2019, pp. 1–10.

[50] 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. 2261–2269.

[51] R. K. Srivastava, K. Greff, and J. Schmidhuber, "Training very deep networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Jul. 2015, pp. 1–9.

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

[53] 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.

[54] 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.

[55] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, Jun. 2017, pp. 5998–6008.

[56] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, and H. Lu, "Dual attention network for scene segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3141–3149.

[57] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2010, pp. 807–814.

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

[59] 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.

[60] A. G. Baydin, B. A. Pearlmutter, A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," *J. Marchine Learn. Res.*, vol. 18, no. 153, pp. 1–43, Jan. 2018.

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

[62] 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.

[63] 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*.

[64] C. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A unified approach for single and multi-view 3D object reconstruction," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Jan. 2016, pp. 628–644.

[65] S. Li, G. Gao, Y. Liu, M. Gu, and Y.-S. Liu, "Implicit filtering for learning neural signed distance functions from 3D point clouds," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2024, pp. 234–251.

[66] A. Ouasfi and A. Boukhayma, "Unsupervised occupancy learning from sparse point cloud," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2024, pp. 21729–21739.

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 also a Project Researcher in the digital economy research platform. His research interests include 3-D data representation, computer vision, machine learning, and medical image processing.

JANI BOUTELLIER (Senior Member, IEEE) received the Ph.D. degree in 2009. He is an Associate Professor with the University of Vaasa, Finland. He is leading projects that concentrate on efficient neural networks and 3-D computer vision. He has co-authored more than 70 peer-reviewed articles. His other research interests include parallel computing and 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 *Journal of Signal Processing Systems*.