

Received 2 December 2023, accepted 18 December 2023, date of publication 25 December 2023,
date of current version 18 January 2024.

Digital Object Identifier 10.1109/ACCESS.2023.3347033

RESEARCH ARTICLE

Enhancing LiDAR-Based Object Recognition Through a Novel Denoising and Modified GDANet Framework

ODDY VIRGANTARA PUTRA^{1,2}, (Student Member, IEEE),
MOCH. ISKANDAR RIANSYAH³, (Graduate Student Member, IEEE),
FARAH ZAKIYAH RAHMANTI³, (Student Member, IEEE),
ARDYONO PRIYADI¹, (Member, IEEE), DIAH PUSPITO WULANDARI¹, (Member, IEEE),
KOHICHI OGATA⁴, (Member, IEEE), EKO MULYANTO YUNIARNO⁵, (Member, IEEE),
AND MAURIDHI HERY PURNOMO⁵, (Senior Member, IEEE)

¹Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

²Department of Informatics, Universitas Darussalam Gontor, Ponorogo 63472, Indonesia

³Department of Electrical Engineering, Institut Teknologi Telkom Surabaya, Surabaya 60231, Indonesia

⁴Department of Computer Science and Electrical Engineering, Kumamoto University, Kumamoto 860-8555, Japan

⁵Department of Computer Engineering, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Corresponding author: Mauridhi Hery Purnomo (hery@ee.its.ac.id)

This work was supported in part by the Indonesian Government through the 2022–2025 Lembaga Pengelola Dana Pendidikan (LPDP) Scholarship, and in part by the Penelitian Fundamental–Riset Dasar Research Grant.

ABSTRACT Object recognition in Point Cloud data from LiDAR sensors often faces challenges like noise, clutter, and ground interference, significantly affecting tasks such as segmentation, classification, and detection. To address these issues, we introduced a framework comprising a denoiser and a classifier, enhancing the robustness of LiDAR-based object recognition. The denoiser plays a crucial role in noise mitigation and operates as a two-part system, utilizing ScoreNet and the Guided Filter. ScoreNet employs advanced scoring techniques to separate valuable information from noise, while the Guided Filter further refines the data, preserving crucial details. The output from the denoiser seamlessly feeds into the classifier, leveraging a modified GDANet architecture with depthwise overparameterized convolution (DOConv) to capture intricate features. We evaluated our approach using Point-to-Point, Hausdorff distance, and Accuracy metrics, comparing it with other denoising methods and point cloud classifiers. Our models demonstrated significant improvements in denoising and classification tasks, with the denoiser achieving outstanding results in the Hausdorff Distance metric, reaching a score of 0.177. Simultaneously, the classifier outperformed other point cloud classifiers, achieving accuracy scores of 90.7% and 96.7% for ModelNet40-C and Human Pose Dataset, respectively. These achievements underscore the importance of our framework in addressing the challenges of noise and clutter in Point Cloud data, ultimately advancing LiDAR-based object recognition.

INDEX TERMS Depthwise convolution, LiDAR, human pose classification, object recognition, point cloud denoising.

I. INTRODUCTION

LiDAR recent advancements in LiDAR technology have marked a significant and transformative shift across multiple

The associate editor coordinating the review of this manuscript and approving it for publication was Prakasam Periasamy¹.

domains, yielding substantial progress, contributing to an array of applications such as remote sensing, human activity analysis, and the burgeoning field of autonomous vehicles, as evidenced by several notable studies [1], [2], [3], [4], [5], [6]. In the automotive domain, LiDAR has become a crucial component in autonomous vehicles by providing

high-resolution 3D information of the surrounding environment, thus improving many object detection tasks [7], [8]. LiDAR also plays a vital role in the field of urban planning. LiDAR can provide a replicable and scalable high-resolution forest map, enabling the capability to identify, map, and capture individual trees [9], [10]. In the realm of remote sensing, LiDAR has revolutionized the way we capture and interpret geographical data. It provides an unparalleled ability to map terrain, forests, and urban landscapes with exceptional precision, aiding in environmental monitoring, land management, and disaster response. These advancements have allowed us to gather vast quantities of detailed 3D data, thus enhancing our capacity to study and understand the Earth's surface and its changes over time, as suggested by [11]. Notably, the cost-effectiveness of this state-of-the-art technology has democratized access to high-quality LiDAR data, making it more accessible to researchers, environmentalists, and various industries.

In 3D data representation, a diverse array of formats exist at our disposal, including point clouds (PC), 2.5-D images, and volumetric structures. Regarding 3D data representation, point clouds have gained prominence due to their unique attribute of preserving the fundamental geometric information [12], [13] within the Euclidean domain without undergoing quantization. However, this characteristic comes with challenges, especially when rendering and comprehending such data. This situation becomes even more intricate in the context of autonomous vehicles and robots designed with human-like features. The challenges associated with working on 3D point clouds are substantial, primarily owing to their unstructured nature and the high dimensionality of the data. Researchers, particularly those immersed in deep learning, have grappled with these formidable obstacles as they strive to harness the full potential of 3D point clouds. Despite these challenges, the field has seen remarkable progress, largely thanks to the availability of pivotal public datasets such as KITTI [14], ModelNet10, ModelNet40 [15], [16], and ShapeNet [17]. These datasets have served as catalysts for advancing point cloud research, fostering the development of a broad spectrum of innovative methods to address the multifaceted issues associated with point cloud processing. These issues encompass many tasks, from 3D point cloud categorization, detection, tracking, and segmentation to the intricacies of registration and the art of reconstruction.

Discrete 3D points obtained from the object's surface generate a point cloud. Unfortunately, the desired outcome is frequently the underlying surface rather than an established foundation. Capturing 3D point clouds has become simple and effective because of developments in scanning technology and image-based reconstruction approaches. Furthermore, LiDAR-captured items are only partially covered in the scanned region, making them difficult to classify.

Since the noisy PC disrupts the performance of the classification task, we proposed a framework for 3D point cloud object classification with a denoising module. Our denoising model combines two submodules, ScoreNet and

Guided Filter (GF). The classification model is composed of a modified geometry disentanglement module with the addition of depthwise overparameterized convolution to its residual layer. The contribution of this study are highlighted as follows:

- 1) The development of a novel framework aimed at addressing challenges in noisy 3D point cloud object recognition.
- 2) The introduction of an effective method for denoising 3D Point Cloud data, enhancing data quality and reliability.
- 3) The modification of the classifier model to improve its efficiency in object recognition tasks within the 3D Point Cloud.
- 4) Significant enhancements in both denoising and classification performance within the 3D Point Cloud domain, ultimately contributing to the accuracy and precision of object recognition processes.

The subsequent sections are organized as follows: Section II provides an in-depth exploration of prior research in denoising and classification tasks, offering valuable context and insights. Section III is dedicated to presenting our novel framework, offering a comprehensive understanding of its components and the innovative approaches we have employed. In Section IV, we present the results of our experiments and provide a detailed analysis of the performance of our models, shedding light on their effectiveness and applicability. Section V offers a concise and insightful conclusion summarizing our key findings and their implications in the broader context. Finally, in Section VI, we acknowledge and discuss the limitations of our research, providing a critical perspective on the scope and areas for potential future improvements.

II. RELATED WORKS

This section provides an in-depth examination of the related work that informs and supports our denoising and classification task research. By delving into prior studies and methodologies, we have positioned our research within the broader context of these critical areas, highlighting the existing knowledge and techniques that have paved the way for our innovative approaches. This discussion elucidates the evolution of strategies and solutions in the domains of denoising and classification, setting the stage for introducing and exploring our own novel framework in subsequent sections.

A. POINT CLOUD DENOISING

Ambiguity and distortions often affect point cloud data (PC) due to the presence of outliers. To address this challenge, Rakotosaona and colleagues introduced a denoising network in their study [18]. This network is specifically tailored to handle multi-surface-based two-level point clouds known for their varying noise intensities.

The denoising process includes two crucial operations in its initial phase: noise filtering and vector correction. The noise filtering technique in this context draws inspiration from PCPNet and PointNet. It employs advanced methods to eliminate unwanted noise elements that might obscure the point cloud. Concurrently, vector correction adjusts vectors by analyzing nearby local surroundings and patches. The vector disparity estimate plays a vital role in the network's loss function, driving the refinement of the denoised point cloud in the second stage.

In a different approach to denoising point clouds, the FWD technique by Zou for downsampling, as outlined in [19], focuses on identifying and preserving critical feature points by averaging multiple points. Further enhancements are introduced by applying wavelet and Gaussian smoothing, allowing for the retention of important Eigenvalues.

Moreover, the recommendation to use an optimal Principal Component Analysis with Bilateral Filtering (PWB) for denoising is worth noting. This recommendation offers an alternative approach to refining point cloud data. Collectively, these methods represent a dynamic landscape of approaches for addressing the challenges of denoising point clouds, providing valuable solutions for enhancing the clarity and reliability of 3D data in various applications and domains.

Noise frequently affects dynamic PCs. Hu introduced spatio-temporal graph-based PC denoising [20]. While the temporal distance is based on surface patches, the spatial information is gathered from immediately nearby patches. The most recent research on PC denoising, which incorporates filtering and deep learning, was performed in [21]. The system is known as PointFilterNet (PFN). PFN builds denoised PCs by filtering them using three learned coefficient vectors rather than directly producing them using networks.

PC clutter is the presence of unnecessary points in data collection collected using sensors like depth cameras or LiDAR scanners. Extracting pertinent data from point clouds is essential in several applications, including 3D scanning, object recognition, and autonomous driving. In contrast, PC datasets may be noisy and include erroneous or irrelevant points that do not accurately represent the objects of interest. The ground points (GP) are one of these obtrusive additions [22]. A series of coplanar points are produced by a LiDAR scan on a flat surface, which causes GP.

B. POINT CLOUD CLASSIFICATION

Directly processing raw point clouds prevalently into 2D deep learning is impossible due to its fickle nature. Charles et al. [23] proposed a deep learning network to extract 3D geometry from point clouds called PointNet. This model directly feeds the raw point cloud to the network. Instead of putting all points, this method samples data with only a 2048 sample set. To be precise, PointNet utilizes several layers of MLP to classify objects based on pointwise features.

To achieve permutation invariance, Deepsets [24] aggregate all nonlinear transform points. Wang et al. [25] put

out a point cloud graph-based categorization approach. The feature model is incrementally modified on each layer as it is learned from space. EdgeConv and MLP make up the model's inner core. The points are combined in EdgeConv using a channel-wise operator. Another study for point cloud recognition suggested a network based on adaptive feature modification. Within areas, it used ultimately linked point pairs. Momenet et al. [26] used the geometric moments of the point cloud to categorize the forms. Huang et al. [27] has proposed a multiple-level contextual encoding technique for point categorization. By structurally considering a point and its surrounds, Liu et al. [28] came up with a context-aware network to address a common and generic feature learning challenge in 3-D PC classification: expressing geometric characteristics more effectively and discriminatively.

III. PROPOSED WORK

This section introduces a comprehensive framework tailored for human pose classification using raw 3D PC data. This framework encompasses crucial steps, including GP removal to eliminate unwanted clutter, noise removal to enhance data quality, and the final classification stage. A visual representation of this innovative framework can be observed in Fig. 1, offering a succinct and insightful overview of the proposed approach's key components and their interplay in achieving accurate human pose classification. Our proposed work is structured into three key modules: ground point removal, the denoiser, and the classifier. We eliminate ground points from the raw point cloud (PC) in the initial stage. Subsequently, the processed point cloud undergoes denoising to eliminate unwanted noise and clutter. The PC is refined within the denoiser using both ScoreNet and Guided Filter methods, resulting in a polished and noise-free PC. The refined PC is extracted through the Geometry-Disentangle module to emphasize its sharp and gentle features. Before the final step, both components are concatenated using the Sharp-Gentle Complementary Attention module. Ultimately, the processed results undergo classification using a Multi-Layered Perceptron to identify the objects within the point cloud. This systematic approach ensures the effective removal of noise, extracting relevant features, and accurately classifying objects in the point cloud data.

A. GROUND POINT REMOVAL

Ground point removal (GPR) is a common preprocessing step in various geospatial applications, including analyzing LiDAR data and point clouds. The points in a point cloud or LiDAR dataset, referred to as "GP," represent the ground surface, such as topography or terrain. It is often necessary to remove these points as they can hinder the study of aboveground characteristics. GPR is frequently employed in the processing of point clouds produced by LiDAR sensors or other remote sensing technologies. During this process, LiDAR sensors emit laser pulses and measure the time it takes for the pulses to return. A dense point cloud is generated

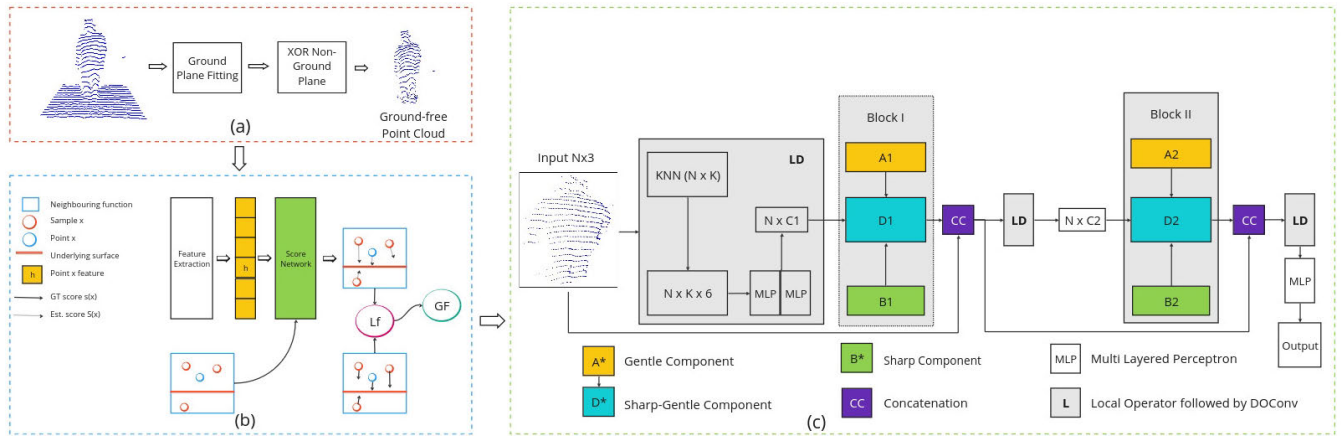


FIGURE 1. The proposed framework, (a) is the ground point removal module, (b) is the denoising module, and (c) is the classifier module.

TABLE 1. Table of notation.

Notation	Description
a, b, c	Coefficient of 2D plane in Section III, part B
x, y, z	3D point cloud coordinate in Section III, part B
d	distance between 2D plane and a point in Section III, part B
\mathbf{n}	vectors of coefficient Section III, part B
\mathbf{C}	Covariance matrix in Equation 3
\mathbf{U}	left matrix
$\mathbf{\Sigma}$	diagonal matrix
\mathbf{V}	right matrix
S_i	score value from ScoreNet
N	the number of point clouds
h	the output of feature extraction in Section III, part D
K	the number of point in ScoreNet, Section III, part D
kNN	k-Nearest Neighbor
α	the step size
E	the gradient ascent in ScoreNet
C	the number of feature dimension in Section III, part E
\mathbf{P}	a matrix of point similarity
ϑ	the graph function of GDANet
Ψ	the vertex of graph
\tilde{P}	normalized of neighboring points
$h(\tilde{P})$	high-pass filter in the sharp and gentle features
W_s, W_g	Learnable adjacency matrices for sharp and gentle, respectively
$\Theta_o, \Theta_s, \Phi_o, \Phi_s$	1D Convolution
X_o, X_s, X_g	Point cloud features, sharp components, and gentle features, respectively
Y_s, Y_g	Sharp-Gentle Complementary Attention Module from Geometry Disentangle Module from sharp and gentle, respectively
Ψ_s, Ψ_g	Non-linear functions to refine X_s and X_g , respectively
Z	Concatenated output from Y_s and Y_g
μ, σ	Mean and standard deviation for Gaussian function
λ	The decay rate for Laplacian

by utilizing the data obtained from these pulses, which accurately represents the surfaces of the objects within the scanned area. Categorizing each point in the point cloud is the first step in eliminating ground points. Points are classified

based on their height, intensity, and reflectance. Elevation is vital in this categorization, as ground points often have lower elevations than non-ground points.

GP can be categorized using a variety of algorithms and techniques. Points can be eliminated using straightforward filters based on elevation criteria. Points are classified as GP if they are below a particular height threshold and are kept; non-GP is deleted. Machine learning algorithms with more sophisticated techniques may categorize points based on elevation, intensity, and point density characteristics. Here, inspired by [29], we employ Ground Plane Fitting (GPF) to remove GP. The first crucial stage in the situational assessment pipeline is segmenting the 3D point cloud produced by current LiDAR sensors. This stage must accurately segment the terrain and any impediments in the vehicle’s route and process each point cloud in real time.

The number of points utilized in future computations is drastically reduced when cloud points that are a component of the ground surface are removed from the point cloud, making up most of the point cloud. These GPs are easy to recognize because they belong to flat surfaces, which have an easy mathematical model. Additionally, it is reasonable to assume that points in the point cloud with the lowest height values are most likely a part of the ground surface. These two factors make it well-suited for this application to identify and extract these GPs. A specific collection of points is chosen using this prior information to start the process. This process eliminates the random selection frequently used in plane-fitting approaches like RANDOM-Sample Consensus (RANSAC), resulting in considerably quicker convergence.

B. GROUND POINT DETECTION

In most cases, using a single-plane model cannot accurately represent the ground surface because GP does not align perfectly into a flat plane. A notable amount of noise is introduced when considering LiDAR measurements over long distances. To determine the original plane model of the ground surface, the ground plane fitting procedure for each

point cloud segment begins by systematically extracting a collection of point seeds with low height values. The distance from every point in the PC segment P to the orthogonal projection on the potential plane is computed by comparing each point to the estimated plane model. Then, the P is compared with the threshold T_{thres} . This comparative analysis serves to identify the specific points associated with the ground. The points identified as part of the ground truth are carried over to the subsequent iteration until a threshold is reached. Eventually, the results from each iteration are cumulatively combined to present the final ground truth.

The lowest point representative (LPR) is defined as an average point obtained from the lowest N_{LPR} . The points from LPR and P lower than T_{thres} are coined for plane model estimation. The model can be mathematically formulated as:

$$ax + by + cz = -d \quad (1)$$

$$\mathbf{n}^T \mathbf{x} = -d \quad (2)$$

where $\mathbf{n} = [a \ b \ c]^T$ and $\mathbf{x} = [x \ y \ z]^T$. To solve n , it can be calculated from a covariance matrix C from the seed points. Thus, we have:

$$C = \sum_{i=1}^s (s_i - \hat{s})(s_i - \hat{s})^T \quad (3)$$

$$U \Sigma V^T = SVD(C) \quad (4)$$

Singular value decomposition (SVD) in (4) is a matrix factorization function for a rectangular matrix. For example, given a matrix C with a size of $m \times n$, we can decompose C into $U \Sigma V^T$ where C is the original matrix, U is the left matrix containing eigenvector of CC^T , Σ is a diagonal matrix which has eigenvalues, while V is the right matrix consists of $C^T C$. The dispersion of C from seed points can be calculated from its SVD from (4) by taking the left singular vector U to get the normal vector of n . Then, the d is obtained from (2) by replacing it with \hat{s} . Once the GP is obtained, finding the non-GP is simple. We can finally obtain the non-GP points using the XOR operation on P by the GP points as seen in Algorithm 1.

C. NOISE REMOVAL

The distribution of a group of clean samples, indicated as $p(x)$, has been convolved with a noise model and considered a noisy point cloud representation. This convolution produces a new distribution with a peak corresponding to the original clean surface, denoted as $(p * n)(x)$.

ScoreNet [30] includes employing gradient ascent to increase each point's log-likelihood, generated from the convolution of p and n , to reduce noise in a noisy point cloud. We incorporated ScoreNet with the Guided Filter (GF) [31], [32] to improve the performance denoising. GF effectively removes artifact-like patches in 2D images, for instance, in CT images [33], sandy images [34], and hazy images [35]. Following the success of 2D images, we implement GF for 3D point clouds. The difference between 2D and 3D for GF is in the search radius vector from neighboring points.

Algorithm 1 Ground Point Removal

Input: $P, N_{iter}, N_{LPR}, T_{thres}, T_{dist}$

Output: P_{ng}

```

1:  $P_{sorted} = SortOrderByHeight(P)$ ;
2:  $LPR = mean(P_{sorted}(1 : N_{LPR}))$ ;
3: for  $k = 1$  to  $|P|$  do
4:    $pk = P(k)$ ;
5:   if  $pk.height < LPR.height + T_{thres}$  then
6:      $P_g = pk$ ;
7:   end if
8: end for
9: for  $i = 1$  to  $N_{iter}$  do
10:   $C = CalcCovMatrix(P_g)$ ;
11:   $P_{mean} = mean(P_g)$ ;
12:   $U \Sigma V^T = SVD(C)$ ;
13:   $U_{normal} = U[:, 2]$ 
14:   $d = -1 * CalcInnerProduct(U_{normal}, P_{mean})$ 
15:   $T_{dist}d = T_{dist}d$ ;
16:   $model = MultiplyMatrix(P, U_{normal})$ ;
17:  for  $k = 1$  to  $|P|$  do
18:     $pk = model(k)$ 
19:    if  $pk \geq T_{dist}d$  then
20:       $P_{ng} = pk$ 
21:    end if
22:  end for
23: end for
24:
25: return  $P_{ng}$ 

```

D. SCORE-BASED DENOISER MODEL

Assuming a point cloud X with noises t with the number of points N , we can distribute the noise-free PC as the convolution operation $p * n$. The network of ScoreNet's objective is estimating neighboring x_i to find the score between them. The score $S_i(r)$ is composed of two main processes: feature extraction (FE) and score estimation (SE). FE calculates noisy input from the PC by using the point-wise operator. FE also benefits from k-Nearest Neighbour (KNN) to build graph-related features. The output of FE is denoted as h . SE accepts input of h_i and then forward to the score function $S_i(x)$ where:

$$S_i(x) = Score(x - x_i, h) \quad (5)$$

where $Score(\cdot)$ is a simple neural network (multi-layered perceptron (MLP)). Since the score function is based on a local operator, $x - x_i$ is related to each other. An ensemble score function is proposed to improve the score function and is denoted as:

$$E_i(x) = \frac{1}{K} \sum_{x_j \in kNN(x_i)}^{max} S_j(x), x \in \mathbb{R}^3 \quad (6)$$

where $kNN(x_i)$ is KNN. Lastly, we can calculate and update the denoised PC with the gradient ascent algorithm

as follows:

$$x_i^t = x_i^{t-1} + \alpha_t E_t(x_i^{t-1}), t = 1, \dots, T \quad (7)$$

where α_t is the incremental step a t -th. The value of α must be $0 < \alpha < 1$.

E. GUIDED FILTER

Guided filter for 3D point cloud (GF3DPC) takes input x_i from ScoreNet. For initial processing, GF3DPC computes the eigenvalue and eigenvector from Principal Component Analysis (PCA). However, the covariance matrix from PCA is substituted with the Gaussian function. The cluttered point cloud x_i is filtered using a local linear transformation, which yields a set of neighboring points p_i and n_{ij} are the normal vectors. Since the normal and the guidance is a linear model, a set of coefficients a_i and b_i can be estimated using a gradient descent algorithm.

The conventional approach to estimating normals, which relies on PCA, often grapples with significant difficulties when retaining essential features within the data. This limitation ultimately affects the method's ability to maintain robustness and accuracy. Given these challenges, PCA's conventional uniform weight function is replaced with a Gaussian function. This transformation extends across the entire procedure of estimating point normals, aiming to bolster the method's performance and capacity to capture intricate details and features within the data, thereby advancing its robustness and precision. The whole GF3DPC is presented in Algorithm 2.

IV. GDANET FOR POINT CLOUD

In this part, we proposed an improved learning geometry representation of the 3D point cloud [36] with the depthwise overparameterized convolution (DOConv) [37], [38]. To consider the characteristics of these sharp and mild variation elements as two holistic representations, GDANet employs the Sharp-mild Complementary Attention Module. Both variation elements are fused with the initial cloud features using various attention techniques. This method successfully divides point clouds into two portions, representing the object's contour and flat areas, and then combines them to provide distinct and complimentary geometric insights.

A. GEOMETRY-DISTANGLE MODULE

The Geometry-Disentangle Module (GDM) is a graph-based processing that decomposes point clouds into two subcomponents, called gentle and sharp. The gentle and sharp terminologies resemble flat and contour features of point clouds, respectively. Let's say a point cloud \mathbf{X} with N points and C feature dimensions, we have $\mathbf{X} = [x_1, x_2, \dots, x_n]^T = [s_1, s_2, \dots, s_n] \in \mathbb{R}^{N \times C}$, where $x_i \in \mathbb{R}^C$ is the i -th points and $s_c \in \mathbb{R}^N$ is the c -th channel feature. The channel features are the 3D coordinates, semantic, and normal features. We can build a graph from those features via an adjacent encoded feature space with a point similarity matrix \mathbf{P} . Thus, we have $\vartheta = (\Psi, \mathbf{P})$. Every point in $x_i \in \mathbb{R}^C$ is mapped with the vertex

Algorithm 2 Guided Filter of 3D Point Cloud

Input: x_i, ϵ, n_K, n_M

Output: $\hat{\mathbf{P}}$

```

1:  $N = \text{len}(x)$ 
2: for  $i = 1$  to  $N$  do
3:   Find the neighborhood of  $p_i$ ;
4:    $\bar{p}_i = \frac{1}{K} \sum_{j=1}^K p_{ij}$ 
5:    $C_i = \frac{1}{K} \sum (\mathbf{P}_{ij} - \bar{p}_i)(\mathbf{P}_{ij} - \bar{p}_i)^T$ ;
6:   Find eigenvalue and eigenvector
7:   Initiate normal  $\mathbf{n}_i$ ;
8:    $\bar{n}_i = \frac{1}{K} \sum_{j=1}^K n_{ij}$ 
9:    $v_i = \frac{1}{K} \sum_{j=1}^K (\mathbf{n}_{ij} \cdot \mathbf{n}_{ij} - \bar{n}_i \cdot \bar{n}_i)$ ;
10:   $w_i = \frac{1}{K} \sum_{j=1}^K (\mathbf{n}_{ij} \cdot \mathbf{n}_{ij} - \bar{n}_i \cdot \bar{n}_i) + \epsilon$ ;
11:   $a_i = \frac{v_i}{w_i}$ ;
12:   $b_i = \mathbf{n}_i - a_i \cdot \mathbf{n}_i$ ;
13:   $\mathbf{n}'_i = a_i \cdot \mathbf{n}_i + b_i$ ;
14: end for
15: for  $m = 1$  to  $M$  do
16:   for  $k = 1$  to  $K$  do
17:      $\mathbf{p}_{sum} = \mathbf{p}_{sum} + \mathbf{n}'_i \cdot [\mathbf{n}'_i \cdot (\mathbf{p}_{ij}^m - \mathbf{p}_i^m)]$ 
18:   end for
19:    $\mathbf{p}_i^{m+1} = \mathbf{p}_i^m + \frac{\mathbf{p}_{sum}}{K}$  {Updating the weight}
20: end for
21:  $\hat{\mathbf{P}} = \mathbf{p}_i^M$ 
22: return  $\hat{\mathbf{P}}$ 

```

$i \in \Psi$. From the previous graph model, we can find the edge weights between point x_i and x_j with:

$$P_{i,j} = \begin{cases} f(\|x_i - x_j\|_2) & \|x_i - x_j\|_2 \leq \tau \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $f(x)$ is Gaussian function and limited with threshold τ . Due to the variety of neighboring points, the weights can be normalized with:

$$\tilde{P}_{i,j} = \frac{P_{i,j}}{\sum_j P_{i,j}} \quad (9)$$

The sharp feature is similar to the edge in a 2D image. Thus, it works as an intensity of nearby pixel variation in a spatial domain, resulting in high and low frequency. These low and high frequencies are the smooth and sharp features, respectively. The Laplacian operator is used to extract the point cloud into its sharp and gentle features, where $L = 2$, $h_0 = 1$, $h_1 = 1$, resulting in a high-pass filter $h(\tilde{\mathbf{P}}) = I - \tilde{\mathbf{P}}$. Thus, we have:

$$y_c = h(\tilde{\mathbf{P}})s_c \in \mathbb{R}^N \quad (10)$$

$s_c \in \mathbb{R}^N$ is processed in (10) and multiplied by filter $h(\tilde{\mathbf{P}})$ resulting a signal y_c . According to [21], since the eigenvalue of $h(\tilde{\mathbf{P}})$ represents a graph frequency in descending order, the filter is considered a high-pass filter. Then, we apply $h(\tilde{\mathbf{P}})$ to

filter point cloud with:

$$(h(\tilde{\mathbf{P}})\mathbf{X})_i = x_i - \sum_j^N \tilde{\mathbf{P}}_{i,j} x_j \quad (11)$$

By using (11), we calculate the vector norm (VN) from each point. The VN holds essential information about point clouds. For instance, a large VN at any point resembles sharp features. We can select M points from descending ordered points clouds. From the selected points, the first and the last points are sharp and gentle components, respectively.

B. SHARP-GENTLE COMPLEMENTARY ATTENTION MODULE

The two output components from GDM are inputs for the Sharp-Gentle Complementary Attention Module (SGCAM). SGCAM has a foundation attention module corresponding to both components by feature weights. Assume we have an input of point cloud, sharp, and gentle components as X_o , X_s , and X_g , respectively. The encoded feature from Gaussian function is calculated with:

$$W_s = \Theta_o(X_o) \cdot \Theta_s(X_s)^T \quad (12)$$

$$W_g = \Phi_o(X_o) \cdot \Phi_g(X_g)^T \quad (13)$$

where Θ_o , Θ_s , Φ_o , and Φ_g are non-linear functions. Each of them has different functions. Then, using element-wise calculation, we can fuse W_s and W_g with:

$$(Y_s)_i = (X_o)_i + \sum_{j=1}^M (W_s)_{ij} \cdot \Psi_s([X_s]_j) \quad (14)$$

$$(Y_g)_i = (X_o)_i + \sum_{j=1}^M (W_g)_{ij} \cdot \Psi_g([X_g]_j) \quad (15)$$

Both (14) and (15) produce their feature sharp and gentle from the attention module, respectively. Finally, the output features are assembled with:

$$Z = Y_s \oplus Y_g \quad (16)$$

This module preserves information from both key and complementary geometric features.

C. THE MODIFIED GDA NETWORK

Here, we proposed a modified version of the Geometry-Disentangle Module Attention Network (GDANet) consisting of two blocks. The network receives input from noise-free 3D point clouds with the length of N and the dimension length of 3, which come from the 3D world coordinates x , y , and z . The points are processed using KNN in the local operator to extract and concatenate point features. Then, a matrix is composed via the GDM module to produce sharp and gentle components. Subsequently, the fused component is formed. By the end of the individual block, a residual block is infused. A residual block contains some 2D convolution (Conv2D) layers. Instead of traditional convolution, we employ depthwise overparameterized convolution (DOConv). The DOConv is described in Section IV-D.

D. DEPTHWISE OVERPARAMETERIZED CONVOLUTION

A combination of DC and Conv2D is the Depthwise Over-parameterization Convolution (DOConv). Studies, such as those conducted by [37], [38], and [39], provide empirical evidence supporting the notion that utilizing empirical data can significantly accelerate the training process of deep non-linear networks. Despite intensive research into innovative network topologies, these findings show unrealized potential in over-parameterization to improve current structures. Therefore, we choose the overparameterization Conv2D layer rather than the standard Conv2D.

A traditional Conv2D is the multiplication between a patch and a matrix pair. For illustration, the Conv2D is denoted as:

$$O = P * W \quad (17)$$

$$O_{out} = \sum_i^{M \times N \times C} W_{cout} P_i \quad (18)$$

where O is the output Conv2D, M is the width of sliding window, N is the height of the sliding window, P and W are patch and matrix, respectively. While the depthwise convolution includes an operation on separated dot-product with the depth as D_{mul} (depth multiplier) and denoted as:

$$O_{d_{mul}c_i} = \sum_i^{M \times N} W_{id_{mul}c} P_{ic} \quad (19)$$

Conversely, DOConv is powerful in its depthwise calculation by taking benefit of the depth multiplier. DOConv has two multiplication compositions, feature-wise and kernel-wise. The DOConv is denoted as:

$$O = (D^T \cdot W) * P \quad (20)$$

where $D^T \in R^{D_{mul} \times M \times N \times C}$ is a transpose from D , W is the kernel operator, and P is the channel. The kernelwise composition in (20) is preferred in the original paper. Overparameterization in the context of depthwise convolutions can be applied to conventional convolutions to create DO-Conv and depthwise convolutions, resulting in DO-DConv. We employ a similar principle to what was used in establishing DO-Conv for DO-Dconv. In simple terms, over-parameterization adds some learnable parameters as a layer. Since DOConv is a linear equivalent transformation, the number of parameters is increased by $(M \times N) \times D_{mul} \times C_{in}$ where the length of $D_{mul} = M \times N$.

Instead of using regular convolution in the residual layer, we replace them with the DOConv on each block in the GDANet. The first residual block receives input x from the local neighboring operator. Then, x is fed to DOConv with ReLU as an activation function.

V. EXPERIMENTS AND RESULTS

In our research, we sought to assess the performance of our proposed framework architecture in denoising and classification tasks. The objective was to compare the Hausdorff and Point-to-Point Distance for denoising and

accuracy of the proposed framework model against existing state-of-the-art models on a well-established dataset, namely the ModelNet40-C [15], [16], which consists of 12,308 PCs in 40 different classes and our human pose dataset [38] which contains 2,247 PCs in six distinguished classes.

We preprocessed the Human Pose Dataset dataset by normalizing PC values to the range [0, 1] and dividing it into a training set (1,797 PCs) and a validation set (450 PCs). We adopted a stochastic gradient descent (SGD) optimizer with momentum. The learning rate was initially set at 0.001, and we applied a learning rate scheduler that reduced it by a factor of 0.1 after every 50 epochs. The batch size was set to 8. We trained the model for 50 epochs with early stopping criteria to prevent overfitting.

Acquiring noisy point cloud data presents a substantial challenge, primarily due to its limited availability within the public domain. Consequently, to create a realistic noise simulation, we employ a noise generator function known as Gaussian noise. This noise generator is systematically applied to all the point cloud objects within the primary dataset to replicate the inherent challenges of working with real-world data.

After training, the model was evaluated on the validation set, and the performance was measured using metrics such as accuracy, P2P, and HD. We also compared our model's performance with existing state-of-the-art CNN architectures on the same dataset. All experiments with the operating system Ubuntu 22.04, Python version 3.9, a Tesla P100 GPU, and 16 GB of RAM.

In this evaluation, we prefer the classification task's accuracy, recall, and precision performance. All parameters in this comparison are set according to their original paper. All models here were trained using a Stochastic Gradient Descent (SDG) optimizer. The learning rate was set to 0.001, momentum was set to 0.9, weight decay was set to 0.0001, and batch size was set to 16.

Two key parameters are essential to configure the Gaussian noise generation: the standard deviation (std) and the mean. In our simulation, these parameters were set to 0.015 and 0.01, respectively. These values were meticulously chosen to emulate noise characteristics that closely resemble those encountered in practical scenarios, allowing for a comprehensive and accurate assessment of the system's robustness and performance in the presence of noise. By introducing Gaussian noise with these specific settings, we create a valuable testbed for evaluating the resilience and effectiveness of algorithms and methods designed to handle noisy point cloud data, addressing a critical aspect of data processing and analysis in various applications and domains.

We compare our modified version of ScoreNet with other point cloud denoising methods such as DBSCAN, Bilateral Filtering (BF), Guided Filter (GF), ScoreNet, Extreme Learning Machine (ELM), and a basic neural network (NN).

A. DATASET GATHERING

In this experiment, we utilized the human pose dataset (HPD) and a public dataset called ModelNet40-C. ModelNet40-C is a simulated 3D object in a point cloud form. This dataset contains 40 different objects. On the other hand, HPD is a 3D point cloud human dataset. HPD is collected using a 32-channel LiDAR Ouster type OS1. The objects in HPD are humans doing six poses: hands-up, lying down, crouching, squatting, and standing. As seen in Fig. 3, the dataset is collected in a small room-sized 6×4 meters. The LiDAR is mounted on a tripod facing forward to the human. Even though LiDAR can do 360-degree scans, in our case, we only limit the scanned area within the human pose range, which is around 45 degrees left and right.

B. DENOISING RESULTS

Obtaining a noisy point cloud is quite challenging. There is hardly available publicly. Thus, we apply a noise generator function called Gaussian noise to simulate the noise. The noise generator is infused to all point cloud objects in the primary dataset. The parameters for generating Gaussian noise were the standard deviation (std) and mean. These two parameters were set to 0.015 and 0.01, respectively. We put the radius and epsilon values for the denoiser setup at 0.25 and 0.1, respectively.

We compared our modified version of ScoreNet with other point cloud denoising methods such as DBSCAN, Bilateral Filtering (BF), Guided Filter (GF), ScoreNet, Extreme Learning Machine (ELM), and a basic neural network (NN). Table 3 exhibits the results of point cloud denoising from different methods measured with Average Point-to-Point Distance (APD) and Average Hausdorff Distance (AHD). At baseline, most methods worked effectively to denoise the point cloud. First, our proposed framework gained the best score with the lowest AHD score in the AHD metrics, defeating ScoreNet. Despite being positioned behind NN and ELM in APD, the disparity scores were insignificant. Fig. 2 presents a visual comparison between noisy and denoised point clouds. In contrast to the Bilateral Filter (Fig. 2c) and Guided Filter (Fig. 2d), which exhibited a noticeable presence of residual noise, our denoiser (Fig. 2e) showcases a remarkable reduction in noise levels. The denoised point cloud generated by our method appears notably more straightforward and less affected by unwanted artifacts than the Bilateral and Guided Filter approaches. This outcome underscores the efficacy of our denoising technique in achieving a more refined and noise-reduced representation of the point cloud data.

C. NOISE EFFECT ON THE DENOISING PERFORMANCE

Here, we evaluate the effect of different noises on the denoising performance measured by Hausdorff (HD) and Point-to-Point Distance (P2P). The raw point clouds are instilled with Laplacian and Gaussian noises. To generate

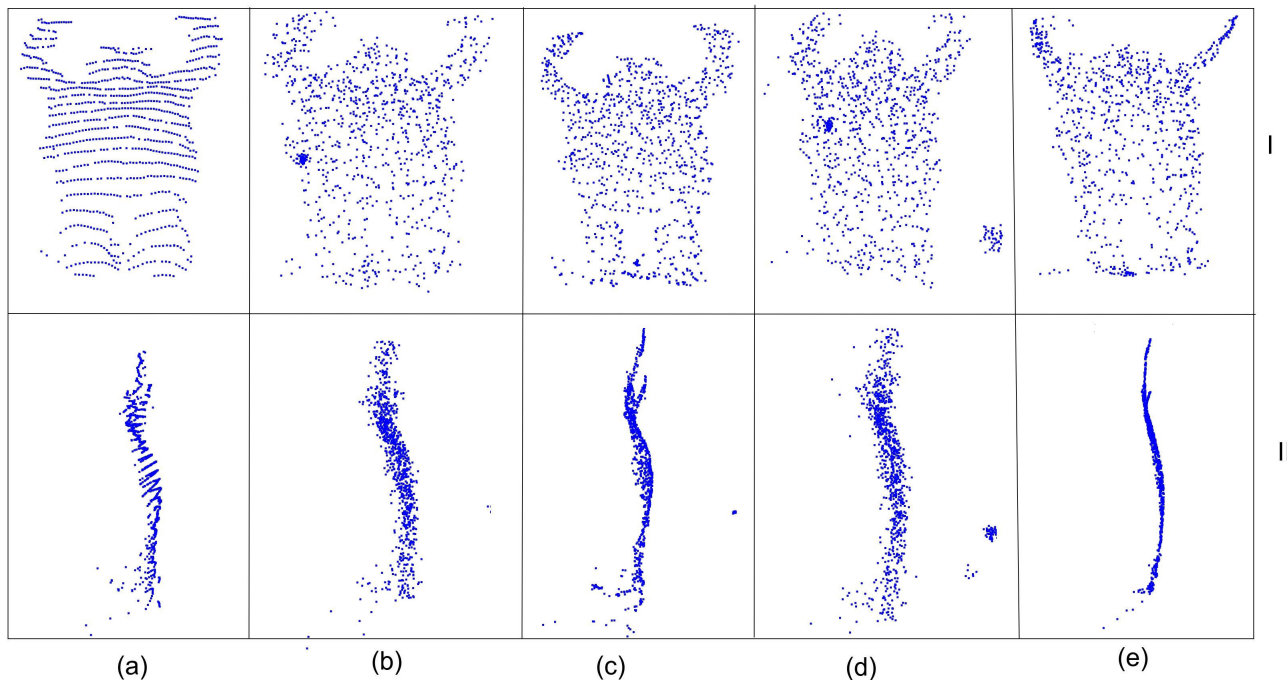


FIGURE 2. Denoising comparison of Human Pose. I and II are viewpoint, back, and side view, respectively. I(a) and II(a) are ground truth, I(b) and II(b) are noisy point clouds, I(c) and II(c) are the product of Bilateral Filtering, I(d) and II(d) results from Guided Filter, and I(e) and II(e) are our denoising model results.

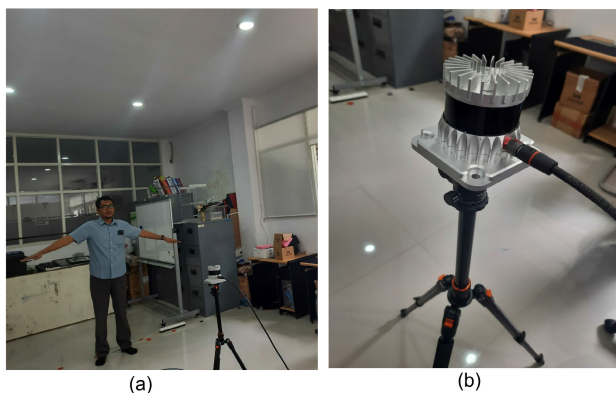


FIGURE 3. Sample of data collection. (a) a man doing a pose, (b) our LiDAR device Ouster 32-channel OS1-type.

Gaussian noise, we utilized the following function:

$$f_{gauss}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (21)$$

where μ is the mean and σ is the variance. Here, we set some different variance (σ) values. From 21, we can generate noises with:

$$p_{noisy} = p_{raw} + f_{gauss} \quad (22)$$

From 22, we can calculate the probability of any given point p being displaced from its original position, resulting in a synthetic Gaussian noise. On the other hand, Laplacian distribution is sharper at the peak and smoother on the

TABLE 2. Different effect of noise type on the denoising performance.

No	Noise type	Params	P2P	HD
1	Gaussian	std=0.03	0.149	0.443
2	Gaussian	std=0.06	0.139	0.433
3	Gaussian	std=0.09	0.119	0.442
4	Laplacian	scale=0.1	0.087	0.760
5	Laplacian	scale=0.3	0.100	2.301
6	Laplacian	scale=0.6	0.153	4.857

tails. A point on a sharp edge or corner most likely has a high Laplacian value. Laplacian noise raises the Laplacian values of points on the point cloud’s surface, making it more challenging to determine the data’s underlying structure.

$$f_{laplace}(x; \mu, \lambda) = \frac{1}{2\lambda} e^{-\frac{|x-\mu|}{\lambda}} \quad (23)$$

Given some 3D point clouds p where p represents the position of a point cloud (x, y, z) within 3D coordinate, μ , and decay rate λ , we can generate Laplacian noise using 24:

$$p_{noisy} = p_{raw} + f_{laplace} \quad (24)$$

where λ is the scale of exponential decay. As seen in Table 2, we notice that manipulating the σ parameter influences the P2P value. A decrease in the σ value corresponds to a reduction in the P2P value. The trend underscores the sensitivity of P2P distance to changes in the σ parameter, indicating that smaller σ values contribute to a more precise alignment of points in the denoising process.

Interestingly, the HD dynamics diverge from those of P2P. In the case of HD, changing the σ parameter produces

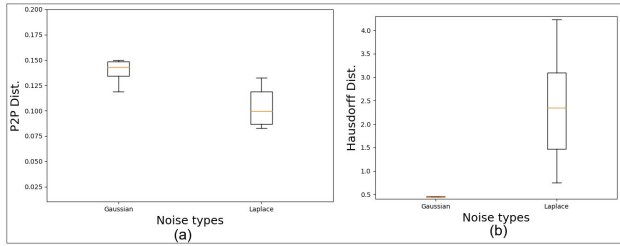


FIGURE 4. Different effect on small step noises tuning for Gaussian and Laplace noises. (a) is the effect on P2P Distance, (b) is the effect on HD.

contrasting effects on the distance score. Larger or smaller σ values increase the HD score, signifying a nuanced relationship between the σ parameter and the HD metric. This divergence highlights the intricate interplay between σ adjustments and their impact on the HD metric, suggesting a more complex sensitivity to parameter variations.

To further enlighten the distribution patterns of P2D and HD, we employ a box plot, as depicted in Fig. 4. This visualization method allows for a comprehensive exploration of the datasets’ spread, central tendency, and potential outliers. The box plot reveals valuable insights into the variability and distribution characteristics of P2D and HD under different σ parameter settings. This nuanced analysis contributes to a holistic understanding of the parameter-dependent behaviors of these distance metrics, offering a more in-depth perspective on their response to variations in the denoising process.

D. CLASSIFICATION RESULTS ON MODELNET40-C

Here, we tested our classifier model, called DOGDANet. The name DOGDANet represents the modified version of GDANet in the convolution layer replaced with the DOConv. Our model is evaluated with other state-of-the-art (SOTA) point cloud classifiers such as PointNet [23], DGCNN [25], GDANet [36], and our previous work [38].

Table 4 compares the classification results for ModelNet-C point clouds, measured primarily in terms of accuracy (Acc), Precision (Prec), Recall (Rec), and F1-Score (FS). This table serves as an essential reference point for evaluating the efficacy of different models in correctly categorizing objects within the point cloud data. Notably, the results indicate that the model with the lowest loss value is DGCNN, followed closely by GDANet. Both of these models demonstrate strong performance in terms of minimizing classification errors.

However, our model, DOGDANet, secures the third position regarding loss value, which might initially seem like a disadvantage. Nevertheless, a closer examination of the accuracy metric reveals a remarkable achievement. Despite its relative position in loss value, DOGDANet emerges as the top-performing model regarding accuracy, boasting an impressive score of 90.7%. This outcome highlights the robustness and effectiveness of our DOGDANet model in object point cloud classification, mainly when applied to the public data ModelNet40-C.

TABLE 3. Results from denoising measured with distance metrics.

No	Method	Parameters	APD	AHD
1	GF	rad=0.25, eps=0.1	0.042	0.517
2	GF + GR	rad=0.25, eps=0.1	0.053	0.650
3	GF + SVD + DBSCAN	rad=0.25, eps=0.1	2.090	3.019
4	ScoreNet		0.012	0.180
5	ScoreNet + GF	rad=0.05, eps=0.1	0.015	0.177
6	Bilateral Filtering	subsample=1, it=2	0.026	0.626
7	NN		0.003	0.257
8	ELM	H. Nodes = 100	0.004	0.287

TABLE 4. The classification results performance from dataset ModelNet40-C.

No	Model	Loss	Acc %	Prec	Rec	FS
1	PointNet [23]	1.565	82.36	64.9	60.8	59.6
2	DGCNN [25]	1.459	88.86	86.7	85.5	85.9
3	GDANet [36]	1.52	86.05	84.5	83.6	83.7
4	VoxelBase [38]	6.5	74	60	69	63
5	Our (DOGDANet)	1.533	90.7	87	87	87

In summary, while DGCNN and GDANet may excel in loss minimization, our DOGDANet truly shines when the focus shifts to accuracy. This indicates that our model is adept at ensuring precise and reliable object classification within point cloud data. It is a valuable and resilient tool in object recognition, especially when dealing with publicly available datasets like ModelNet40-C.

E. CLASSIFICATION RESULTS ON HUMAN POSE DATASET

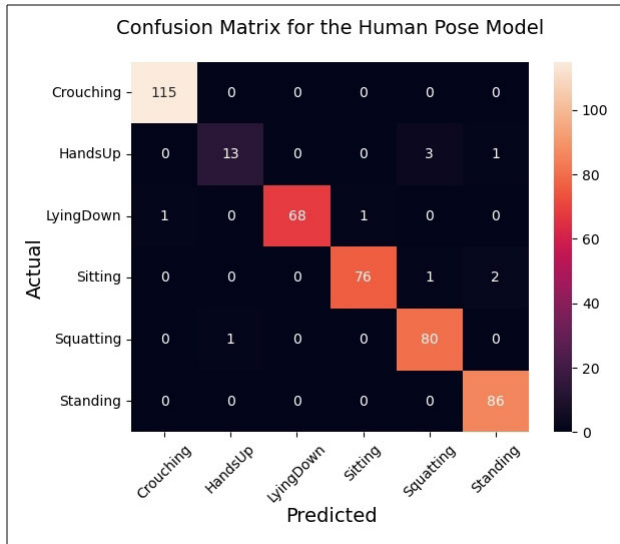
We also compare our method with other SOTA using the human pose dataset. Table 5 presents information about the classification of the human pose dataset obtained using LiDAR from five different methods. The table shows that the models with the lowest accuracy are PointNet and DGCNN, with an accuracy score below 80 %. On the other hand, the top position was achieved by our model with 96.65 %, followed by VoxelBased and GDANet. However, our model is a runner-up position in terms of loss value. The result exhibits the excellent work of our model, improving the accuracy of our previous work [38] significantly by nearly 10 %.

In provide a comprehensive assessment of the classification performance of individual classes, we conducted an extensive evaluation of our classifier model. The results are visually represented through the confusion matrix, as illustrated in Fig. 5. This visualization is a powerful tool for understanding how well the model distinguishes and correctly assigns instances to various classes.

We can make several noteworthy observations upon a thorough analysis of the confusion matrix. First and foremost, the classifier model demonstrates commendable performance across all classes. This is particularly evident when examining the elements’ intensity along the confusion matrix’s main diagonal. The high intensity of these diagonal

TABLE 5. The Classification results from the human pose point cloud dataset.

No	Model	Loss	Acc %	Prec	Rec	FS
1	PointNet [23]	0.953	79.99	73	76	67
2	DGCNN [25]	1.003	78.4	80.5	82.7	81
3	GDANet [36]	0.886	85.7	73.5	71.3	60.1
4	VoxelBased [38]	0.993	87.06	87	87	87
5	Our (DOG DANet)	0.905	96.7	98	96	97

**FIGURE 5.** Confusion matrix of classification results using Human Pose Dataset.

elements indicates that the model has correctly classified most instances for each class.

It is essential to acknowledge that, in practice, no classifier is entirely error-free, and our model is no exception. There are, indeed, instances where the classifier has made incorrect assignments. However, it is worth emphasizing that such occurrences are relatively infrequent and limited in scope. In other words, the model exhibits occasional misinterpretations, where a few classes are erroneously classified.

This analysis underscores the overall robust and reliable performance of our classifier model. It excels in correctly categorizing instances for a wide range of classes, and even when errors do occur, they are isolated incidents rather than widespread issues. Consequently, our classifier model's effectiveness in object recognition remains impressive, and it can be deemed a robust tool for various classification tasks.

Here, we also evaluated the performance of our classifier model in detecting of each classes using the human pose dataset. The model was measured using True Positive Rate (TPR), False Positive Rate(FPR), True Negative Rate (TNR), and False Negative Rate (FNR). It can be seen from Table 6 that our model achieves the best TPR score in Lying Down, Squat, and Standing classes and TNR in Crouching, Lying Down, and Standing.

TABLE 6. The classification results performance of human pose dataset calculated with True Positive Rate (TPR), False Positive Rate (FPR).

Metric	Crouching	Hands Up	Lying Down	Sitting	Squat	Standing
TPR	0.991	0.933	1	0.988	1	1
FPR	0	0.002	0	0.003	0.003	0
TNR	1	0.997	1	0.997	0.997	1
FNR	0.008	0.067	0	0.011	0	0

VI. LIMITATIONS

Despite the notable achievements of our approach, it is imperative to acknowledge several limitations that warrant consideration. While our denoiser model has demonstrated substantial enhancements in the Hausdorff Distance metric, there remains the possibility that specific challenging scenarios or unique noise patterns may still present difficulties for the system. Moreover, the robustness of the classification performance, although impressive, can be subject to the influence of hardware variations and calibration differences among LiDAR sensors. Further research is indispensable to ascertain the framework's generalizability across a spectrum of LiDAR sensor models and diverse environmental conditions, ensuring its adaptability in various real-world settings.

Furthermore, as we explore the practical application of our framework, it is essential to address the considerations of real-time processing and scalability, particularly in resource-constrained environments. Overcoming these challenges is crucial for the seamless deployment of our framework in a wide array of LiDAR-based applications. Consequently, future endeavors should devise solutions that maintain the high-performance standards achieved in controlled settings and ensure the reliability and versatility required for real-world implementation.

VII. CONCLUSION

In conclusion, this paper has successfully addressed a pervasive and critical challenge in the field of point cloud-based classification: the issue of noise and clutter that often hinders the accurate recognition of objects and humans. Directly processing raw point clouds, replete with these unwanted artifacts, can significantly impede the effectiveness of recognition tasks. In response to this challenge, we have introduced an innovative two-pronged approach consisting of a 3D point cloud denoiser and classifier designed to elevate the precision and robustness of object and human recognition in noisy environments.

Our denoiser model, combining ScoreNet with the Guided Filter, represents a novel modification to differentiate essential information from the disruptive noise and clutter in point cloud data. This dynamic duo of techniques works harmoniously to enhance data quality by reducing noise while retaining vital details. Simultaneously, our classifier is an evolved iteration of GDANet, empowered by integrating a depthwise overparameterized convolution (DOConv) layer known as DOGDANet. This enhancement significantly augments the network's ability to capture intricate features

within the point cloud data, further boosting the accuracy and reliability of classification. Through rigorous evaluation, we have demonstrated the effectiveness of our denoiser model, leveraging metrics such as the Hausdorff Distance and Point-to-Point distance with a score of 0.177 and the classifier with an accuracy score of 90.7 % and 96.7% for ModelNet40-C and Human Pose dataset, respectively.

In summary, our models have not only met but surpassed the performance of previous state-of-the-art (SOTA) methods in denoising and classification tasks. This groundbreaking research sets a new standard for the field, offering a solution that significantly advances the accuracy and efficiency of object and human recognition in the challenging domain of noisy point cloud data.

The accomplishments presented in this paper pave the way for several avenues of future work and exploration in the realm of point cloud-based classification. We plan to improve our model by fine-tuning and optimizing the proposed denoiser and classifier models. Continuous refinement might uncover additional enhancements to boost performance metrics and ensure adaptability across a broader range of scenarios. Furthermore, investigating the integration of the proposed models with various sensor technologies, such as LiDAR and millimeter wave radar, could offer valuable insights into the models' adaptability to different sensing modalities. This multimodal approach can lead to advancements in sensor fusion for improved object and human recognition.

REFERENCES

- X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3D LiDAR and camera data for object detection in autonomous vehicle applications," *IEEE Sensors J.*, vol. 20, no. 9, pp. 4901–4913, May 2020.
- I.-S. Weon, S.-G. Lee, and J.-K. Ryu, "Object recognition based interpolation with 3D LiDAR and vision for autonomous driving of an intelligent vehicle," *IEEE Access*, vol. 8, pp. 65599–65608, 2020.
- R. Soitinaho, M. Moll, and T. Oksanen, "2D LiDAR based object detection and tracking on a moving vehicle," *IFAC-PapersOnLine*, vol. 55, no. 32, pp. 66–71, 2022.
- K. Cao and P. Cosman, "Denoising and inpainting for point clouds compressed by V-PCC," *IEEE Access*, vol. 9, pp. 107688–107700, 2021.
- M. A. Irfan and E. Magli, "Joint geometry and color point cloud denoising based on graph wavelets," *IEEE Access*, vol. 9, pp. 21149–21166, 2021.
- A. Liang, H. Zhang, H. Hua, and W. Chen, "To drop or to select: Reduce the negative effects of disturbance features for point cloud classification from an interpretable perspective," *IEEE Access*, vol. 11, pp. 36184–36202, 2023.
- G. Zamanakos, L. Tsochatzidis, A. Amanatiadis, and I. Pratikakis, "A comprehensive survey of LiDAR-based 3D object detection methods with deep learning for autonomous driving," *Comput. Graph.*, vol. 99, pp. 153–181, Oct. 2021.
- L. Fan, X. Xiong, F. Wang, N. Wang, and Z. Zhang, "RangeDet: In defense of range view for LiDAR-based 3D object detection," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 2898–2907.
- R. Pu and S. Landry, "Mapping urban tree species by integrating multi-seasonal high resolution Pleiades satellite imagery with airborne LiDAR data," *Urban Forestry Urban Greening*, vol. 53, Aug. 2020, Art. no. 126675.
- M. Münzinger, N. Pechtel, and M. Behnisch, "Mapping the urban forest in detail: From LiDAR point clouds to 3D tree models," *Urban Forestry Urban Greening*, vol. 74, Aug. 2022, Art. no. 127637.
- U. Stilla and Y. Xu, "Change detection of urban objects using 3D point clouds: A review," *ISPRS J. Photogramm. Remote Sens.*, vol. 197, pp. 228–255, Mar. 2023.
- J.-I. Park, J. Park, and K.-S. Kim, "Fast and accurate desnowing algorithm for LiDAR point clouds," *IEEE Access*, vol. 8, pp. 160202–160212, 2020.
- Y. Fan, Q. Zhang, S. Liu, Y. Tang, X. Jing, J. Yao, and H. Han, "Semantic SLAM with more accurate point cloud map in dynamic environments," *IEEE Access*, vol. 8, pp. 112237–112252, 2020.
- A. Geiger, P. Lenz, C. Stillner, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, Sep. 2013.
- A. Goyal, H. Law, B. Liu, A. Newell, and J. Deng, "Revisiting point cloud shape classification with a simple and effective baseline," 2021, *arXiv:2106.05304*.
- J. Sun, Q. Zhang, B. Kailkhura, Z. Yu, C. Xiao, and Z. M. Mao, "Benchmarking robustness of 3D point cloud recognition against common corruptions," 2022, *arXiv:2201.12296*.
- A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "ScanNet: Richly-annotated 3D reconstructions of indoor scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5828–5839.
- M. Rakotosaona, V. La Barbera, P. Guerrero, N. J. Mitra, and M. Ovsjanikov, "PointCleanNet: Learning to denoise and remove outliers from dense point clouds," *Comput. Graph. Forum*, vol. 39, no. 1, pp. 185–203, Feb. 2020.
- B. Zou, H. Qiu, and Y. Lu, "Point cloud reduction and denoising based on optimized downsampling and bilateral filtering," *IEEE Access*, vol. 8, pp. 136316–136326, 2020.
- W. Hu, Q. Hu, Z. Wang, and X. Gao, "Dynamic point cloud denoising via manifold-to-manifold distance," *IEEE Trans. Image Process.*, vol. 30, pp. 6168–6183, 2021.
- X. Wang, X. Fan, and D. Zhao, "PointFilterNet: A filtering network for point cloud denoising," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 3, pp. 1276–1290, Mar. 2023.
- M. Arora, L. Wiesmann, X. Chen, and C. Stachniss, "Static map generation from 3D LiDAR point clouds exploiting ground segmentation," *Robot. Auto. Syst.*, vol. 159, Jan. 2023, Art. no. 104287.
- 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.
- M. Zaheer, S. Kottur, S. Ravanbakhsh, B. Póczos, R. Salakhutdinov, and A. J. Smola, "Deep sets," in *Proc. 31st Conf. Neural Inf. Process. Syst. Red Hook, NY, USA: Curran Associates*, 2017, pp. 3394–3404.
- Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, Oct. 2019.
- M. Joseph-Rivlin, A. Zvirin, and R. Kimmel, "Momen⁴t: Flavor the moments in learning to classify shapes," in *Proc. Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 4085–4094.
- R. Huang, D. Hong, Y. Xu, W. Yao, and U. Stilla, "Multi-scale local context embedding for LiDAR point cloud classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 4, pp. 721–725, Apr. 2020.
- S. Liu, T. Wang, Y. Zhang, R. Zhou, C. Dai, Y. Zhang, H. Lei, and H. Wang, "Rethinking of learning-based 3D keypoints detection for large-scale point clouds registration," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 112, Aug. 2022, Art. no. 102944.
- D. Zermas, I. Izzat, and N. Papanikolopoulos, "Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications," in *Proc. Int. Conf. Robot. Automat.*, May 2017, pp. 5067–5073.
- S. Luo and W. Hu, "Score-based point cloud denoising," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 4563–4572.
- K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
- Z.-A. Feng and X.-F. Han, "Guided normal filter for 3D point clouds," *Multimedia Tools Appl.*, vol. 82, no. 9, pp. 13797–13810, Apr. 2023.
- Z. Shen, C. Gong, W. Yu, and L. Zeng, "Guided image filtering reconstruction based on total variation and prior image for limited-angle CT," *IEEE Access*, vol. 8, pp. 151878–151887, 2020.
- Y. Cheng, Z. Jia, H. Lai, J. Yang, and N. K. Kasabov, "A fast sand-dust image enhancement algorithm by blue channel compensation and guided image filtering," *IEEE Access*, vol. 8, pp. 196690–196699, 2020.
- O. V. Putra, B. Prianto, E. M. Yuniarno, and M. H. Purnomo, "Visibility restoration of lake crater hazy image based on dark channel prior," in *Proc. Int. Comput. Sci. Eng. Conf. (ICSEC)*, Dec. 2016, pp. 1–6.

- [36] M. Xu, J. Zhang, Z. Zhou, M. Xu, X. Qi, and Y. Qiao, "Learning geometry-disentangled representation for complementary understanding of 3D object point cloud," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, no. 4, pp. 3056–3064, May 2021.
- [37] J. Cao, Y. Li, M. Sun, Y. Chen, D. Lischinski, D. Cohen-Or, B. Chen, and C. Tu, "DO-Conv: Depthwise over-parameterized convolutional layer," *IEEE Trans. Image Process.*, vol. 31, pp. 3726–3736, 2022.
- [38] O. V. Putra, Riandini, E. M. Yuniarno, and M. H. Purnomo, "Depthwise over-parameterized CNN for voxel human pose classification," in *Proc. Int. Seminar Intell. Technol. Appl. (ISITIA)*, Jul. 2023, pp. 54–59.
- [39] S. Arora, N. Cohen, and E. Hazan, "On the optimization of deep networks: Implicit acceleration by overparameterization," in *Proc. 35th Int. Conf. Mach. Learn.*, vol. 80, 2018, pp. 244–253.



vision, artificial intelligence, and 3-D image processing.

ODDY VIRGANTARA PUTRA (Student Member, IEEE) received the B.S. degree in information system and the M.S. degree in electrical engineering from Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2012 and 2017, respectively, where he is currently pursuing the Ph.D. degree with the Department of Electrical Engineering. Since 2015, he has been a Lecturer with Universitas Darussalam Gontor, Ponorogo, Indonesia. His research interests include computer



ests include robotics, intelligent systems, computer vision, and control systems.

MOCH. ISKANDAR RIANSYAH (Graduate Student Member, IEEE) received the Diploma degree from Politeknik Elektronika Negeri Surabaya (PENS), and the master's degree in control and intelligent systems from Institut Teknologi Bandung (ITB). He is currently pursuing the Ph.D. degree with Institut Teknologi Sepuluh Nopember (ITS). He is also a Lecturer of electrical engineering with Institut Teknologi Telkom Surabaya (ITTS), Surabaya, Indonesia. His research inter-



Institut Teknologi Telkom Surabaya. She is interested in machine learning, deep learning, image processing, and computer vision.

FARAH ZAKIYAH RAHMANTI (Student Member, IEEE) was born in Surabaya, Indonesia, in December 1987. She received the Bachelor of Applied Science degree from Politeknik Elektronika Negeri Surabaya, in 2010, and the master's degree in electrical engineering from Institut Teknologi Sepuluh Nopember, in 2013, where she is currently pursuing the Ph.D. degree in electrical engineering. She is also a Lecturer with the Information Technology Study Program,



renewable energy, and artificial intelligence application in power systems.

ARDYONO PRIYADI (Member, IEEE) received the Ph.D. degree in electrical engineering from Hiroshima University, Hiroshima, Japan, in 2011. He is currently an Associate Professor with the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia. He is registered as a professional assessing the transient stability of the industrial power systems. His research interests include power system dynamics and control, transient stability,



Engineering, Institut Teknologi Sepuluh Nopember. Her research interests include signal processing and artificial intelligence.

DIAH PUSPITO WULANDARI (Member, IEEE) received the bachelor's degree from the Department of Electrical Engineering, Institut Teknologi Bandung, Indonesia, in 2004, the master's degree from the School of Informatics, The University of Edinburgh, U.K., in 2006, and the Ph.D. degree from the Department of Electrical Engineering, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, in 2016. Since 2005, she has been a Lecturer with the Department of Computer



Communication Engineers (IEICE); the Institute of Electrical Engineers of Japan (IEEJ); and the Acoustical Society of Japan (ASJ).

KOHICHI OGATA (Member, IEEE) was born in Kumamoto, Japan, in 1967. He received the B.Eng., M.Eng., and Ph.D. degrees in engineering from Kumamoto University, Kumamoto, in 1989, 1991, and 1994, respectively. He is currently an Associate Professor with the Department of Computer Science and Electrical Engineering, Graduate School of Science and Technology, Kumamoto University. He is a member of the



Currently, he is a Senior Lecturer of computer engineering with the Department of Computer Engineering, ITS. His research interests include computer vision, image processing, machine learning, and deep learning.

EKO MULYANTO YUNIARNO (Member, IEEE) received the bachelor's, master's, and Ph.D. degrees in electrical engineering from Institut Teknologi Sepuluh Nopember (ITS), Surabaya, Indonesia, in 1995, 2005, and 2013, respectively.



Chair of the IEEE Industrial Electronics Society Indonesia Section.

MAURIDHI HERY PURNOMO (Senior Member, IEEE) received the Ph.D. degree from Osaka City University, Osaka, Japan, in 1995. He is currently a Professor with Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia, where he has involved in teaching research philosophy, artificial intelligence, neural networks, and image processing. His research interests include smart grids, renewable energy, artificial intelligence application on healthcare, and power systems. He is the