applied sciences

MDPI

Article

CMD-Net: Self-Supervised Category-Level 3D Shape Denoising through Canonicalization

Caner Sahin [1,2]

[1] Department of Informatics, Istanbul Medeniyet University, Istanbul 34720, Turkey; caner.sahin@medeniyet.edu.tr
[2] Gemsas, Istanbul 34906, Turkey

Abstract: Point clouds provide a compact representation of 3D shapes however, the imperfections in acquisition processes corrupt point clouds by noise and give rise to a decrease in their power for representing 3D shapes. Learning-based denoising methods operate displacement prediction and suffer from shrinkage and outliers. In addition, they require pre-aligned datasets. In this paper, we present a self-supervised learning-based method, Canonical Mapping and Denoising Network (CMD-Net), and address category-level 3D shape denoising through canonicalization. We formulate denoising as a 3D semantic shape correspondence estimation task where we explore ordered 3D intrinsic structure points. Utilizing the convex hull of the explored structure points, the corruption on objects' surfaces is eliminated. Our method is capable of canonicalizing noise-corrupted clouds under arbitrary rotations, therefore circumventing the requirement on pre-aligned data. The complete model learns to canonicalize the input through a novel transformer that serves as a proxy in the downstream denoising task. The analyses on the experiments validate the promising performance of the presented method on both synthetic and real data. We show that our method can not only eliminate corruption, but also remove clutter from the test data. We additionally create a novel dataset for the problem in hand and will make it publicly available in our project web-page.

Keywords: point cloud denoising; canonical mapping; point cloud structure learning

Citation: Sahin, C. CMD-Net: Self-Supervised Category-Level 3D Shape Denoising through Canonicalization. *Appl. Sci.* **2022**, *12*, 10474. https://doi.org/10.3390/app122010474

Academic Editors: Wonjoon Kim, Sekyoung Youm and Sungbum Jun

Received: 12 August 2022
Accepted: 11 October 2022
Published: 17 October 2022

1. Introduction

Point clouds provide a compact representation of 3D shape. However, the imperfections on acquisition processes corrupt clouds by noise, thus diminishing point clouds' 3D shape representation power. Deep architectures directly consuming raw point clouds [1,2], removing the requirement of 3D data preprocessing, pave the way for the learning-based point cloud denoising methods [3]. Forefront studies [4–7] estimate the displacement of noisy points from the noise-free cloud and restore the underlying surface through reconstruction. However, incorrect displacement estimations cause performance degradation, therefore suffering from shrinkage and outliers. Luo et al. [8] present a differentiable manifold reconstruction approach to learn the underlying manifold of a noisy point cloud. They aim to capture *intrinsic* structures in point clouds, and they test their method on a ShapeNet dataset. However, they do not leverage the semantic consistency of the *categories* to learn intrinsic structures consistently.

Learning structures of 3D shapes are important for the tasks where different shape instances in a given category are co-analyzed, e.g., co-segmentation [9], shape abstraction [10], and shape correspondence estimation [11]. Such tasks present a structural understanding of the entire set. To this end, they necessarily leverage semantic consistency among shape instances. Recent deep learning methods learn to generate point clouds with structure information in an unsupervised way [12,13]. Intrinsic structural representation points [11] are yet other unsupervised technique satisfying the consistency requirement through ordered sets of structure points. In this study, we leverage semantic consistency among the

shape instances of a given category and formulate the denoising problem as a 3D semantic shape correspondence estimation task [11].

The aforementioned learning-based denoising methods generally require pre-aligned datasets. Contemporary studies for 3D scene understanding examine categorical objects in different poses. One distinct class of such studies is the category-level 6D object pose estimators [14–17]. This class of pose estimators, employing category-level reasoning, align different instances of a given category in a shared pose-normalized canonical shape space and prompt interest in canonicalization learning, particularly in designing self-supervised approaches [18–20]. Condor [20] uses rotation information as a supervisory signal and is able to canonicalize point clouds. Our work circumvents the requirement on pre-aligned data, canonicalizing arbitrarily rotated clouds through a novel transformer.

In this paper, we present a self-supervised method, Canonical Mapping and Denoising Network (CMD-Net), and address category-level point cloud denoising through canonicalization. Inspired by [11], we formulate denoising as a 3D semantic shape correspondence estimation task where we find out semantically consistent and ordered 3D intrinsic structure points. To this end, we leverage semantic consistency among the shape structures co-analyzing the variety of shape instances of a given category in the pose-normalized canonical space. Utilizing the convex hull of the explored structure points, the corruption on objects' surfaces is eliminated. Our method is also capable of canonicalizing noise-corrupted clouds under arbitrary rotations, therefore circumventing the requirement on pre-aligned data. The learning of the complete model is employed in a self-supervised fashion: the network automatically creates rotation matrices as supervisory signals through axis-aligned representation. By multiplying those with the unlabelled foreground clouds of the canonical 3D CAD models, rotated data are populated, and they are corrupted by noise. Using the data, the complete model learns to canonicalize the noise-corrupted input cloud through a novel transformer that serves as a proxy in the downstream denoising task.

The analyses on the experiments validate the promising performance of the presented method on both synthetic and real data. We show that our method can not only eliminate corruption, but also remove the clutter from the test data. Figure 1 depicts sample results of CMD-Net. To summarize, our main contributions are as follows:

- To the best of our knowledge, this is the first time we adopt representation learning through intrinsic structure points for 3D shape denoising.
- We alleviate the requirement of pre-aligned data by canonicalizing input clouds through a transformer and present a self-supervised method for the denoising problem.
- Our method does not only deal with 3D synthetic shapes, but also handles cluttered real objects.

Organization of the Paper. Section 2 presents the relevant studies available in the literature. In Section 3, we detail CMD-Net along with our training strategy. Experimental results are provided in Section 4. We conclude the paper in Section 5 with several remarks and future works.

Denotation. Throughout the paper, we denote scalar variables by lowercase letters (e.g., x) and vector variables by bold lowercase letters (e.g., $\mathbf{x}$). Capital letters (e.g., P) are adopted for specific functions or variables. We use bold capital letters (e.g., $\mathbf{I}$) to denote a matrix or a set of vectors and use calligraphic letters (e.g., $\mathcal{B}$) to express batches or datasets. The symbols not commonly employed will be defined in the context. The objects in all figures in this paper are color-coded: blueish gray is for noise-free ground truth 3D shapes in the canonical space; gray depicts noise-corrupted (noisy) input; orange represents canonicalized instances; magenta shows denoised objects.

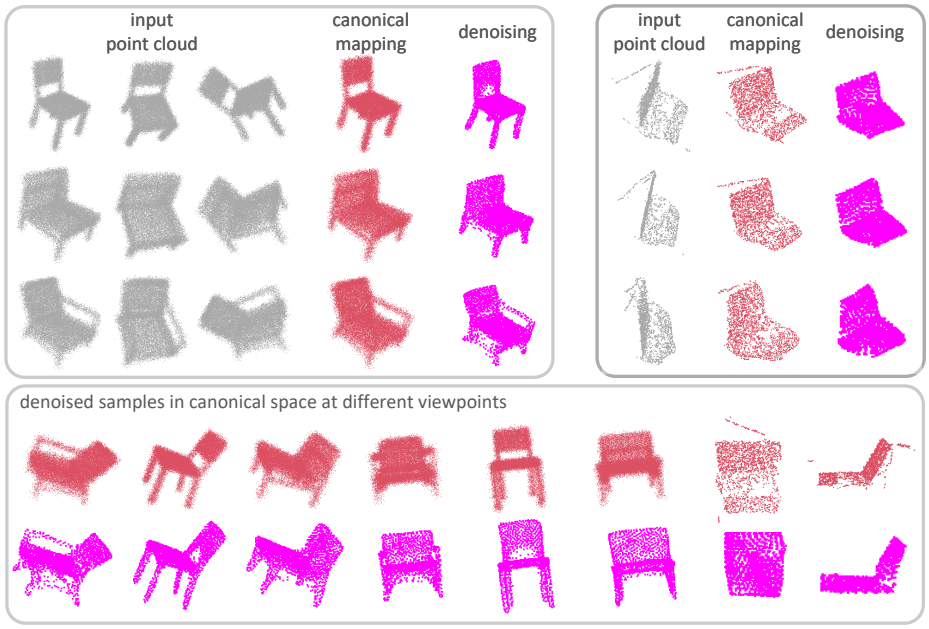

Figure 1. CMD-Net is a self-supervised method that addresses category-level 3D shape denoising through canonicalization: (**top left**) synthetic objects; (**top right**) cluttered real data; and (**bottom**) denoised samples in the canonical space shown at different viewpoints.

2. Related Work

In this section, we form three subsections, each of which involves the studies most relevant to CMD-Net: (i) learning-based point cloud denoising, (ii) 3D structure learning, and (iii) canonicalization.

2.1. Learning-Based Point Cloud Denoising

Deep architectures directly consuming raw point clouds [1,2] and removing the requirement on 3D data preprocessing pave the way for the learning-based point cloud denoising methods [3–8]. Neural projection [4] estimates a reference plane for each point in the noisy cloud and projects the noisy point onto the corresponding plane to eliminate corruption on object surfaces. Total denoising [5] assumes that points with denser surroundings are closer to the underlying surface. The introduced unsupervised loss function converges to this underlying surface. PointCleanNet [6] has a two-stage denoising operation. It first removes the outliers and then refines the resultant cloud calculating a correction vector for each point. The study in [7] improves the robustness of point cloud denoising, proposing graph-convolutional layers for the network. As these methods are based on noise distance prediction, incorrect estimations give rise to performance degradation; therefore, they suffer from shrinkage and outliers. Luo et al. [8] presented a differentiable manifold reconstruction approach to learn the underlying manifold of a noisy point cloud. They aimed to capture *intrinsic* structures in point clouds and test their method on the ShapeNet dataset; however, they did not leverage the semantic consistency of the *categories* to learn intrinsic structures consistently. The study in [3] presents a score-based point cloud denoising approach where the distribution of noise-corrupted point clouds are exploited.

2.2. 3D Structure Learning

There are important studies in the literature addressing 3D structure learning in an unsupervised fashion. FoldingNet [21] presents a folding decoder that constructs the 3D surface of a point cloud by deforming a 2D canonical manifold. The sequential folding operations preserve the point correspondence. PPF-FoldNet [22] leverages the well-known point pair features (PPFs) to generate discriminatively encoded code words, which are then fed into the decoder to reconstruct PPFs by folding. AtlasNet [12] deforms

primitive structures to generate structured point clouds; however, the primitives depend on hand-chosen parametric shapes. TopNet [13] circumvents this dependency designing a tree-structured decoder network. Being similar to FoldingNet, the study in [23] deforms primitive shapes, but it also has a supervised setting in which point correspondences across training samples are utilized. 3D-PointCapsNet [24] encodes shape information in latent capsules. The Tree-GAN method [25] generates 3D point clouds introducing tree-structured graph convolution network (TreeGCN). TreeGCN increases the representational power of the features through ancestor information, which is also utilized for generating objects' parts. The study in [26] explores geometrically and semantically consistent category-specific 3D keypoints exploiting the symmetric linear basis representation. It exploits the symmetric linear basis representation for modeling the 3D shapes of objects of a given category. KeypointDeformer [27] represents different instances of a given category with semantically consistent 3D keypoints. When the instance changes, it controls the deviation in shape through its deformation model that propagates the keypoint displacement to the rest of the shape. The autoencoder architecture of Skeleton Merger [28] leverages the skeletal representation; the encoder generates keypoint proposals, and the decoder reconstructs point clouds from their skeletons. The generator of SP-GAN [29] makes use of a uniform sphere as a global prior and a random latent code as a local prior for each point involved in the sphere to finally generate part-aware shapes. LAKe-Net [30] localizes aligned keypoints for the topology-aware point cloud completion task.

There recently has been a surge of interest in self-supervised point cloud understanding. Some important works learn 3D structures of the point clouds via 3D keypoints [31–35]. The framework in KeypointNet [31] is applicable for any downstream task as long as the loss function is differentiable with respect to the location of the keypoints. KeypointNet formulates 3D pose estimation as the downstream task for which 3D keypoints are jointly optimized as a set of latent variables. USIP [32] detects stable (repeatable and accurate) keypoints from 3D point clouds using the feature proposal network (FPN) whose training is employed through transformation matrices automatically created for each training pair. As employed in [32,33], the authors use rotation information in an auxiliary task and produce semantic features 3D keypoint prediction. D3Feat [34] proposes a joint keypoint and descriptor learning framework for point cloud alignment. It designs a density-invariant keypoint selection scheme to increase repeatability and uses a keypoint-guided loss to increase consistency. The studies in [36,37] present a self-supervised upsampling autoencoder. Zhang et al. [37] use a pretrained upsampling autoencoder network for initializing any downstream task (e.g., classification, segmentation) that will benefit from the network's capabilities on learning high-level semantics and low-level geometry. Zhao et al. [36] leverage implicit neural representations and learn arbitrary-scale point cloud upsampling. Thabet et al. [38] exploit Morton sequences as pseudo-labels to represent each point in a cloud and use their task for semantic segmentation. Cross-modal contrastive learning framework of CrossPoint [39] minimizes the distance between a point cloud and its 2D image captured from a random viewpoint and preserves invariance to transformations.

2.3. Canonicalization

The methods capable of registering point clouds into a canonical frame lessen the dependencies on pre-aligned datasets. This registration meanwhile reduces intra-class variations for the scene understanding techniques that reason about categorical objects in different poses. Therefore, canonicalization has emerged as a useful tool. Gu et al. [40] estimate 3D canonical shape and 6D pose of an instance using its partial observations. Draco [41] canonicalizes instances of a given category to make the method robust across unseen target objects. These studies [40,41] are trained in a weakly-supervised fashion.

The power of self-supervision has recently been exploited [18–20,42–44]. C3DPO [42] learns a canonical 3D frame based on 2D input keypoints. Canonical Point Cloud Autoencoder (CPAE) [18] maps semantically and topologically corresponding points of the instances in a given category to the same location of the canonical 3D sphere primitive

placed in the bottleneck. It then recovers original shapes by inverse mapping. The correspondence learning strategy in [18] is leveraged by an autoregressive model [19] which generates point clouds from sequentialized shape compositions. Condor [20] uses rotation information as a supervisory signal and is able to canonicalize full or partial clouds.

3. Proposed Method

In this section, we introduce our method, Canonical Mapping and Denoising Network (CMD-Net), a self-supervised deep neural network for the *category-level 3D shape denoising through canonicalization* problem. We first present an overview of the method and then provide the details on denoising and canonical mapping. We lastly elaborate our training strategy.

3.1. Method Overview

Given a category of interest G (e.g., *chair*), U is the 3D shape of an **unseen** instance. $\mathbf{C}^{can}$ is the clean (noise-free) canonical point cloud of U, and $\mathbf{N}^{can}$ is the noisy (noise-corrupted) canonical point cloud of U. The denoiser (downstream task) of CMD-Net targets eliminating corruption from $\mathbf{N}^{can}$ so that $\mathbf{H}_u^{can}$, the noise-free canonical point cloud that lies within the convex hull of the unseen instance U, is recovered (see Figure 2). We define a distance function d between two point clouds $\mathbf{a}$ and $\mathbf{b}$ as $d(\mathbf{a}, \mathbf{b}) : \mathbb{R}^{M\times 3} \times \mathbb{R}^{M\times 3} \rightarrow \mathbb{R}^{+}$, and the following objective is optimized in the downstream task:

$$\mathbf{H}_u^{can^*} = \arg\min_{\mathbf{H}_u^{can}} d(\mathbf{N}^{can}, \mathbf{H}_u^{can}). \tag{1}$$

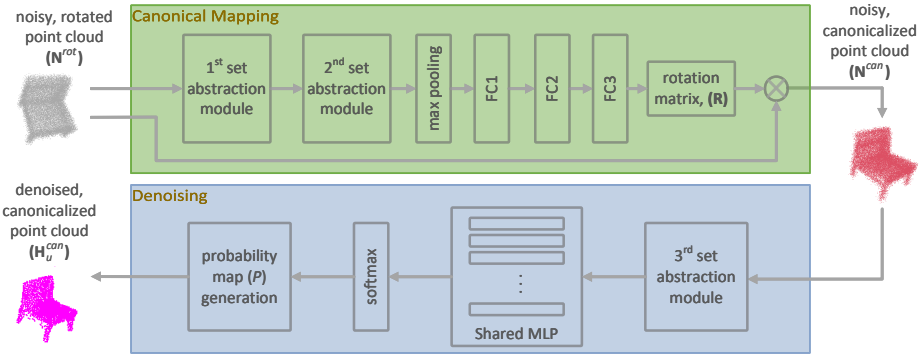

Figure 2. CMD-Net: Canonical Mapping (**light green**) and denoising (**dark blue**) parts are encoder–decoder architectures based on PointNet++. Canonical Mapping (auxiliary task) serves as a proxy for denoising (downstream task). See Section 3 for details.

Our method is capable of canonicalizing noise-corrupted clouds under arbitrary rotations, therefore circumventing the requirement of pre-aligned data. The canonical mapping part (auxiliary task) of CMD-Net takes the noise-corrupted rotated point cloud $\mathbf{N}^{rot}$ of U as input and aims to predict the best network parameters that minimize the following objective:

$$\mathbf{N}^{can^*}, \mathbf{R}^* = \arg\min_{\mathbf{N}^{can}, \mathbf{R}} d(\mathbf{R}(\mathbf{N}^{rot}), \mathbf{N}^{can}), \tag{2}$$

where $\mathbf{R}$ is the rotation matrix that aligns $\mathbf{N}^{rot}$ to $\mathbf{N}^{can}$. Based on Equations (1) and (2), the complete network intakes an arbitrarily rotated noisy point cloud $\mathbf{N}^{rot}$, first canonicalizes it to obtain $\mathbf{N}^{can}$ (auxiliary task), and then employs denoising (downstream task) to obtain $\mathbf{H}_u^{can}$, the noise-free canonical point cloud that lies within the convex hull of the unseen instance U (see complete network in Figure 2).

3.2. Denoising

The denoising part of CMD-Net is an encoder–decoder architecture. The encoder of the denoiser represents points involved in a cloud through a PointNet++-based 3rd set

abstraction (SA) module (see Figure 2). Given the noise-corrupted canonical input cloud $\mathbf{N}^{can} = \{\mathbf{n}_1, \mathbf{n}_2, ..., \mathbf{n}_M\}$, the module samples a new set of points $\mathbf{W}^{can} = \{\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_q\}$ ($\mathbf{n}_i, \mathbf{w}_i \in \mathbb{R}^3$). Each point $\mathbf{w}_i$ is then represented with a z dimensional feature vector $\mathbf{f}_i \in \mathbb{R}^z$, and the feature set $\mathbf{F}^{can}$ of the sampled points $\mathbf{W}^{can}$ is as follows: $\mathbf{F}^{can} = \{\mathbf{f}_1, \mathbf{f}_2, ..., \mathbf{f}_q\}$. Hence, the input of the 3rd SA module is $M \times 3$ dimensional point cloud, and its output is $q \times c$ dimensional tensor. Note that q is less than or equal to M ($q \leq M$).

The decoder of the denoiser processes $\mathbf{W}^{can}$ and its feature representation $\mathbf{F}^{can}$ with a shared multi-layer perceptron (MLP) block (shown in Figure 3) to obtain the noise-free canonical point cloud that lies within the shape's convex hull $\mathbf{H}_u^{can} = \{\mathbf{u}_1, \mathbf{u}_2, .., \mathbf{u}_M\}$ where $\mathbf{u}_i \in \mathbb{R}^3$. Inspired by [11], we apply softmax activation at the shared MLP exit to generate probability map $P_i = \{p_i^1, p_i^2, .., p_i^q\}$ for each point $\mathbf{u}_i$ in $\mathbf{H}_u^{can}$. Each element p_i^j in the probability map depicts the probability of the point $\mathbf{w}_j$ being the denoised point $\mathbf{u}_i$ [11]. The set $P = \{P_1, P_2, .., ..P_M\}$ is the probability map for the complete cloud. The denoised cloud $\mathbf{H}_u^{can}$ is then calculated as in the following:

$$\mathbf{u}_i = \sum_{j=1}^{q} \mathbf{w}_j p_i^j \quad \text{and} \quad \sum_{j=1}^{q} p_i^j \quad \text{and} \quad i = 1, 2, ..., M. \tag{3}$$

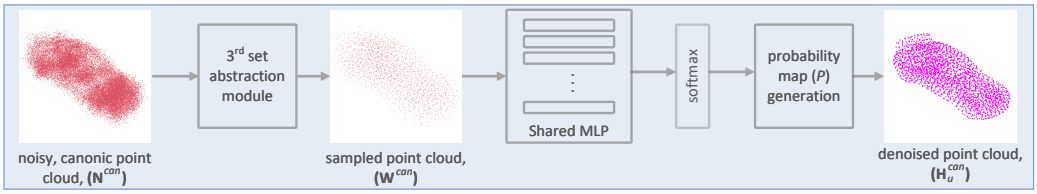

Figure 3. Denoising process of a noise-corrupted instance.

3.3. Canonical Mapping

The canonical mapping part of CMD-Net is a PointNet++-based encoder–decoder architecture. Given a noise-corrupted rotated point cloud $\mathbf{N}^{rot}$ as input, the encoder represents the input with b dimensional feature vector through two SA modules (see Figure 2, the 1st and the 2nd SA modules) plus one max-pooling layer. Feeding this representation into fully-connected (FC) layers (FC1, FC2, FC3 in Figure 2), it estimates a rotation matrix $\mathbf{R}$ that when multiplied with $\mathbf{N}^{rot}$ produces $\mathbf{N}^{can}$ to be fed into the denoiser.

3.4. Training Strategy and Loss Functions

We provide several definitions for describing the network training: S_i denotes the 3D shape of the i^{th} **seen** instance of the category G, and the set of 3D shapes of t seen instances forms $\mathcal{G}$:

$$\mathcal{G} = \{S_i | i = 1, 2, .., t\}. \tag{4}$$

$\mathbf{X}_i^c$ depicts the noise-free canonical point cloud of the instance S_i, and the set of noise-free canonical point clouds forms the batch $\mathcal{B}^{can}$:

$$\mathcal{B}^{can} = \{\mathbf{X}_i^c | i = 1, 2, .., t\}. \tag{5}$$

Arbitrary rotation application on $\mathbf{X}_i^c$ generates noise-free rotated point cloud $\mathbf{X}_i^r$, and the set of noise-free rotated point clouds forms $\mathcal{B}^{rot}$:

$$\mathcal{B}^{rot} = \{\mathbf{X}_i^r | i = 1, 2, .., t\}. \tag{6}$$

Corrupting $\mathbf{X}_i^c$ by noise results $\mathbf{X}_{n_i}^c$. We represent the set of noise-corrupted canonical point clouds with $\mathcal{B}_{noisy}^{can}$:

$$\mathcal{B}_{noisy}^{can} = \{\mathbf{X}_{n_i}^c | i = 1, 2, .., t\}. \tag{7}$$

Corrupting $\mathbf{X}_i^r$ by noise results $\mathbf{X}_{n_i}^r$. We depict the set of noise-corrupted rotated point clouds with $\mathcal{B}_{noisy}^{rot}$:

$$\mathcal{B}_{noisy}^{rot} = \{\mathbf{X}_{n_i}^r | i = 1, 2, .., t\}. \tag{8}$$

Note that all point clouds are composed of M points in 3D space: $\mathbf{X}_i^c, \mathbf{X}_i^r, \mathbf{X}_{n_i}^c, \mathbf{X}_{n_i}^r \in \mathbb{R}^{M\times 3}$.

Auxiliary task training starts with feeding instances in the batch $\mathcal{B}^{can}$ to the network. Each instance involved in $\mathcal{B}^{can}$ is automatically corrupted by noise, and the network transforms $\mathcal{B}^{can}$ to $\mathcal{B}_{noisy}^{can}$. Then, instances in $\mathcal{B}_{noisy}^{can}$ are rotated by multiplying those with the rotation matrices randomly produced through axis-angle representation, and $\mathcal{B}_{noisy}^{rot}$ is formed. In order to allow the canonical mapping part of the network to learn the identity matrix as well, $\mathcal{B}_{noisy}^{can}$ and $\mathcal{B}_{noisy}^{rot}$ are merged and are used together to train the auxiliary task. Canonical Mapping Loss (L_{CM}) is the Mean-Squared Error (MSE) loss that is used to align the rotated instances on to the canonical space:

$$L_{CM}(\mathbf{R}^{tar}, \mathbf{R}) = \frac{1}{2t}\sum_{j=1}^{2t}(\mathbf{R}_j^{tar} - \mathbf{R}_j)^2, \tag{9}$$

where $\mathbf{R}^{tar}$ is the target rotation matrix.

Once the auxiliary task training is completed, we train our downstream task. as employed in auxiliary task training, $\mathcal{B}_{noisy}^{can}$ and $\mathcal{B}_{noisy}^{rot}$ are automatically generated and are fed into the canonical mapping network to obtain canonicalized clouds. The denoiser then learns to hypothesise canonical noise-free points lying within the convex hull of the shapes $\mathbf{H}^{can}$. Denoising Loss (L_D) is the Chamfer Distance (also known as cloud-to-cloud distance) loss:

$$L_D(\mathbf{H}^{can}, \mathbf{X}^c) = \sum_{\mathbf{h}_i \in \mathbf{H}^{can}} \min_{\mathbf{x}_j \in \mathbf{X}^c} || \mathbf{h}_i - \mathbf{x}_j ||_2^2 + \sum_{\mathbf{x}_j \in \mathbf{X}^c} \min_{\mathbf{h}_i \in \mathbf{H}^{can}} || \mathbf{h}_i - \mathbf{x}_j ||_2^2. \tag{10}$$

Discussion on Denoising. According to Equations (3) and (10), the network learns to produce the probability map of the input cloud, each element of which depicts the probability of an input point being the denoised point. Since the denoising part of CMD-Net is trained by using a set of instances of a given category, the learnt probability map employs semantic shape correspondence estimation and produces consistent denoised points spatially lying on the semantically corresponding parts of the objects, and thus leveraging the semantic consistency within the training instances to generate intrinsic structure points. As Equation (3) suggests, any point estimated as an intrinsic structure point equals the linear combination of the points involved in the input cloud, thus enabling the network to estimate an intrinsic structure point lying within the convex hull of the object. These characteristics improve the robustness of CMD-Net across shrinkage and outliers.

Discussion on Canonical Mapping. In the progress of our research, we produced different network designs, one of which was based on PointNet. However, in this research, our aim is to address the problem on both synthetic and real data. As the instances of real data quite likely contain missing pixels (points) plus background clutter, we decided to use PointNet++-based SA modules in our design to improve the robustness of CMD-Net. More discussion on this issue is presented in Section 4.

4. Experiments

In this section, we perform a series of experiments to measure the performance of CMD-Net comparing it with the state-of-the-art approaches. We also analyse our method and present important discussions and findings.

4.1. Dataset and Metric

Dataset. We first evaluated the performance of CMD-Net on the GPDNet dataset [7]. The test set of this dataset was composed of 10 categories selected from ShapeNet [45] repository: *airplane*, *bench*, *car*, *chair*, *lamp*, *pillow*, *rifle*, *sofa*, *speaker*, and *table*. Each category of the test set involved 10 instances, each of which was subjected to three levels of Gaussian noise: low-level ($\sigma = 0.01$), medium-level ($\sigma = 0.015$), and high-level ($\sigma = 0.02$). Each instance consisted of 30,720 points. Note that, every instance in this dataset was pre-aligned to the canonical space, and there was no rotated instance. Since our method CMD-Net is capable of canonicalizing arbitrarily rotated clouds, we were unable to measure the performance of our method on the GPDNet dataset regarding canonicalization. To this end, we created a new dataset committed to the problem of *category-level 3D shape denoising through canonicalization*: We first took 10 instances of the *airplane* category from the GPDNet dataset. Then, we rotated the low-level, mid-level, and high-level noise-corrupted samples (30 in total) of each instance around the x axis (θ_x), the y axis (θ_y), and the z axis (θ_z), respectively. We applied the set Θ that involved five set of rotations $\Theta = \{(\theta_{x_i}, \theta_{y_i}, \theta_{z_i})\}_{i=1}^{5}$ where all numbers were in degree:

$$\Theta = \{(10, 10, 10), (15, -20, -30), (30, 20, 40), (50, 45, -20), (-20, -20, -20)\} \quad (11)$$

After performing the same procedure for 10 categories of the GPDNet dataset, we finally obtained a test set that involved 1500 samples of 100 instances of 10 categories. We call this dataset the "CMD-Net dataset", and will make it publicly available. Sample instances are shown in Figure 4.

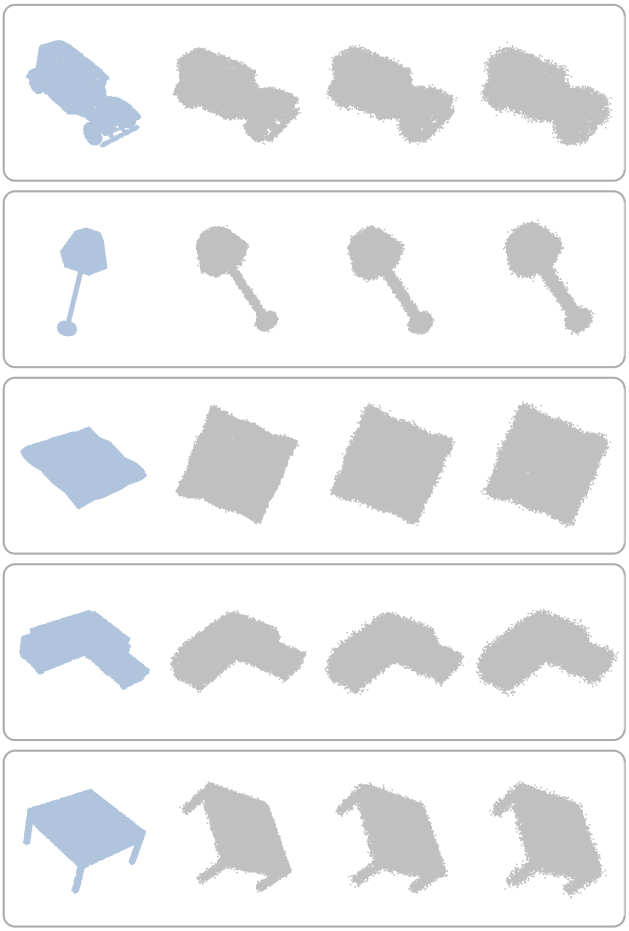

Figure 4. Samples from CMD-Net Dataset: (**left-most column**) noise-free 3D point clouds in the canonical space; rotated instances (**2nd column**) with $\sigma = 0.01$; (**3rd column**) with $\sigma = 0.015$; (**right-most column**) with $\sigma = 0.02$.

Metric. We evaluated the denoising performance of the proposed method using the Chamfer Measure (cloud-to-cloud distance) [7] that produces the score C2C for the average distance of the denoised points $\mathbf{H}_u^{can}$ from the ground truth cloud of the underlying surface $\mathbf{C}^{can}$:

$$C2C = \frac{1}{2M}\left[\sum_{i=1}^{M}\min_{j}|| \mathbf{u}_i - \mathbf{c}_j ||_2^2 + \sum_{j=1}^{M}\min_{i}|| \mathbf{u}_i - \mathbf{c}_j ||_2^2\right], \mathbf{u}_i \in \mathbf{H}_u^{can}, \mathbf{c}_j \in \mathbf{C}^{can}. \quad (12)$$

The lower the score of C2C, the better the denoising performance of the method is.

4.2. Network Structure and Implementation Details

Network Structure. The Canonical Mapping part of CMD-Net is composed of two SA modules plus three FC layers. The first and the second SA modules have 512 and 128 grouping centers, respectively. Both employ Multi-Scale Grouping (MSG) to capture multi-scale patterns and concatenate features at different scales to acquire multi-scale features: The scales of the first module were 0.1, 0.2, and 0.4, and the scales of the second module were 0.2, 0.4, and 0.8. The output of the second module is a 640-dimensional vector representing the input cloud. This feature was fed into the set of three FC layers, FC1, FC2, and FC3, each of which had 512, 256, and 9 neurons, respectively. The FC3 layer generated the rotation matrix that when multiplied with the input cloud transformed it to the canonical space.

The denoising part of the network was composed of one SA layer (the third one) plus a shared MLP block which was followed by a softmax activation. The design of the third SA module was the same as the second SA module of the Canonical Mapping part of the complete network. The shared MLP block comprised three layers with the neuron numbers 640, 1024, and 2048. We prevented over-fitting of the MLP block by setting the dropout ratio to 0.2.

Implementation Details. We trained a separate CMD-Net for each category of interest and our training was performed on an 8 GB NVIDIA GeForce RTX 2080/PCIe/SSE2 GPU. The training had two stages: the first stage was the auxiliary task training where we trained "Canonical Mapping" part of the network for 1000 epochs. The batch size was 16. In the second stage, we trained "Denoising" part (downstream task) for 400 epochs freezing the rest of the network. The batch size was nine. We used an initial learning rate of 0.01 and the ADAM optimizer ($\beta_1 = 0.9$ and $\beta_2 = 0.9999$) with a 10^{-5} weight decay.

4.3. Experiments on GPDNet Dataset

In this experiment, we measured the performance of CMD-Net on the seven categories of the GPDNet dataset, *airplane*, *bench*, *car*, *chair*, *pillow*, *rifle*, and *sofa*, comparing our method with the state of the art. The instances in this dataset were canonicalized, and CMD-Net took in those from the canonical mapping part of the network though, since it had already learned the identity matrix. For a fair comparison, we used the evaluation codes provided in [7].

When the instances were corrupted by the low-level noise ($\sigma = 0.01$), our method presented relatively competitive results (see Table 1a). GPDNet outperformed on four categories, *airplane*, *bench*, *car*, and *chair*. In the *chair* category, CMD-Net reported slightly less than GPDNet: 29.50 was of GPDNet and our result was 30.05. APSS showed the best performance in the *pillow* and *sofa* categories. Our method performed best on *rifle*, the category that involved instances with high curvature.

As the noise level increased, the merits of our method became distinctive. When the instances were corrupted by the mid-level noise ($\sigma = 0.015$), CMD-Net performed on a par with GPDNet, reporting best results (see Table 1b). GPDNet reached the lowest C2C scores on *airplane*, *car*, *pillow*, and *sofa* categories. For *airplane* and *pillow*, our method closely kept track of GPDNet: 28.47-by-30.30 and 23.32-by-26.56. The success of CMD-Net was particularly demonstrated in the categories that involved high-curvature instances,

bench, *chair*, and *rifle*. Moving Least Squares (MLS)-based surface fitting [46,47], point set resampling [48], sparsity-based [49], and graph-based [50,51] methods showed degraded performance as the level of noise increased since they employ surface reconstruction or normal estimation. Learning-based PCN [6] deteriorated as it suffered from shrinkage and outliers due to direct displacement prediction. Our method estimated the points lying within the convex hull of the objects' shape through intrinsic structure point representations instead of estimating the displacement of the noisy points. This shows the resilience of our method across shrinkage and outliers.

Table 1. Denoising performance of our method (CMD-Net) on the GPDNet dataset is compared using the Chamfer Measure evaluation metric. The numbers in this table are C2C scores (see the metric in Section 4.1).

(a) Chamfer Measure ($\times 10^{-6}$), ($\sigma = 0.01$)

Method	Airplane	Bench	Car	Chair	Pillow	Rifle	Sofa
Noisy	50.320	48.71	64.340	60.78	69.79	38.97	69.630
DGCNN [50]	44.82	38.700	60.47	59.69	64.28	26.99	65.05
APSS [47]	28.22	26.97	47.73	37.31	**15.64**	36.01	**22.27**
RIMLS [46]	39.73	32.76	55.56	45.650	21.23	49.37	28.04
AWLOP [48]	31.27	34.08	54.21	47.91	46.36	27.79	53.08
MRPCA [49]	28.19	32.93	44.33	38.41	23.95	23.49	32.14
GLR [51]	19.56	20.43	42.22	34.98	17.59	15.84	30.88
PCN [6]	26.36	27.64	75.34	55.10	21.07	15.09	43.36
GPDNet [7]	**17.22**	**19.33**	**38.09**	**29.50**	17.110	14.450	25.870
Ours	24.08	23.16	54.63	30.05	18.12	**13.12**	41.17

(b) Chamfer Measure ($\times 10^{-6}$), ($\sigma = 0.015$)

Method	Airplane	Bench	Car	Chair	Pillow	Rifle	Sofa
Noisy	97.78	94.82	102.23	105.16	132.57	80.40	121.02
DGCNN [50]	84.40	64.760	93.43	94.45	113.32	61.04	99.63
APSS [47]	86.42	75.51	72.56	81.47	22.74	92.14	42.80
RIMLS [46]	106.33	91.93	103.52	104.38	42.54	110.51	69.92
AWLOP [48]	73.32	82.04	93.38	92.47	112.54	69.35	107.58
MRPCA [49]	67.39	70.05	69.88	73.45	73.67	55.65	72.62
GLR [51]	36.76	32.19	55.92	48.62	31.38	31.81	51.12
PCN [6]	35.27	30.10	92.23	69.18	29.02	21.45	61.15
GPDNet [7]	**28.47**	28.72	**52.92**	46.28	**23.32**	28.43	**40.10**
Ours	30.30	**26.50**	66.89	**42.94**	26.56	**17.42**	52.41

(c) Chamfer Measure ($\times 10^{-6}$), ($\sigma = 0.02$)

Method	Airplane	Bench	Car	Chair	Pillow	Rifle	Sofa
Noisy	161.79	161.52	148.74	163.75	215.58	144.18	184.11
DGCNN [50]	127.44	99.36	113.94	132.91	190.32	131.91	155.51
APSS [47]	175.68	166.85	141.69	160.01	164.83	195.68	166.34
RIMLS [46]	186.24	182.42	167.78	155.38	196.53	176.07	190.91
AWLOP [48]	145.94	157.29	145.51	158.12	206.14	144.22	178.93
MRPCA [49]	123.71	127.51	109.49	122.70	150.65	105.87	133.98
GLR [51]	90.55	83.99	77.56	79.85	85.86	89.19	89.31
PCN [6]	74.17	90.34	160.08	145.56	92.84	71.57	144.72
GPDNet [7]	45.96	**41.24**	**72.06**	67.91	**34.47**	43.07	**62.58**
Ours	**43.24**	45.53	84.59	**61.25**	41.15	**36.94**	84.58

In case target instances are subjected to high-level noise ($\sigma = 0.02$), CMDNet performs on a par with GPDNet showing the best values (see Table 1c). In general, CMD-Net had a similar pattern as the other methods, when the noise level increased from $\sigma = 0.01$ to $\sigma = 0.02$, the generated C2C scores also increased (see Table 1). For instance, for the *airplane* category, the scores our method produced were 24.08 for $\sigma = 0.01$, 30.30 for $\sigma = 0.015$, and 43.24 for $\sigma = 0.02$. However, the ratio of the raise of our method was relatively lower.

Figure 5 presents several qualitative results. In the figure, the first row demonstrates ground truth noise-free canonical point clouds, the second row presents unseen noise-corrupted test clouds from *airplane* and *car* categories for $\sigma = 0.02$, and the third row shows the denoising results of our method. Since the canonical mapping part of the network has already learned the identity matrix, it does not change the rotation of the test samples, and the denoising part eliminates the corruption from the objects' underlying surfaces and estimates noise-free point clouds (compare the second and the third rows).

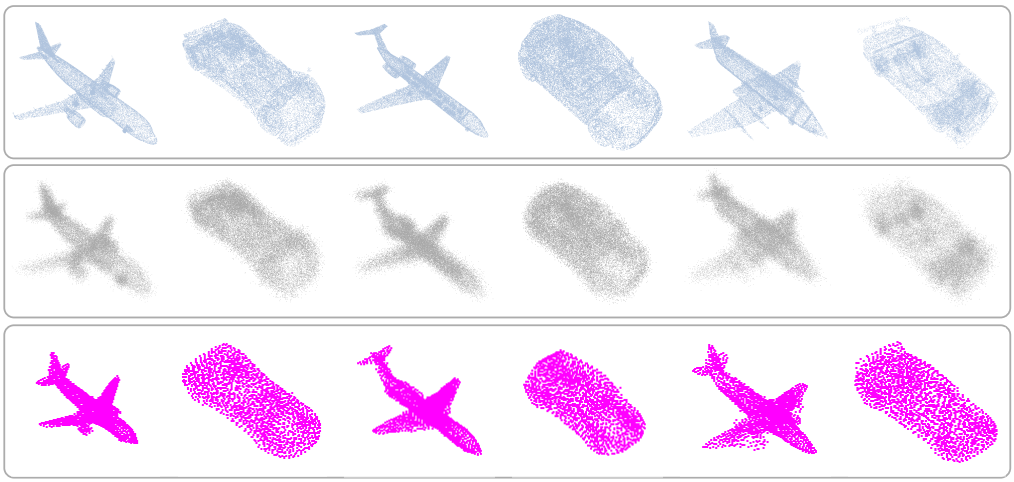

Figure 5. Sample results from experiments on GPD-Net dataset: instances in the first row are of ground truth canonical point clouds, the second row shows inputs to be denoised in the **testing stage** ($\sigma = 0.02$), the third row shows the outputs of our method.

4.4. Experiments on CMD-Net Dataset

We performed this class of experiments on the CMD-Net dataset to demonstrate the performance of our method on rotated unseen instances. Unlike the experiments in Section 4.3, we measured the performance of our method using our evaluation codes.

Since this is the first time we introduce the CMD-Net dataset, we present intra-class analyses. Table 2 depicts the denoising performance of our method for $\sigma = 0.01$, $\sigma = 0.015$, and $\sigma = 0.02$, and *noisy* rows show the C2C scores of the input noise-corrupted test instances. According to the table, our method reached the lowest score on *airplane*, and the score of the *car* category was the highest for the low-level noise ($\sigma = 0.01$). In case the test instances wase corrupted by mid-level noise ($\sigma = 0.015$), the value of the *chair* category was the minimum, and *pillow* was the maximum. At the high-level noise ($\sigma = 0.02$), *rifle-car* pair had 0.64-by-0.96 as the minimum–maximum pair of the scores. Despite giving an idea about the performance of CMD-Net, we find this analysis insufficient, since the C2C scores of the test instances involved in each category were different. Therefore, our next error analysis presents the scores of CMD-Net normalized by *noisy* rows of Table 2.

The left-most image of Figure 6 is of the normalized C2C scores for the low-level noise. As depicted in the image, our method performed best on the *airplane* category. Although *chair* was the third successful category for the low-level noise, CMD-Net performed best on this category when the objects were corrupted by the mid-level noise (see the middle image in Figure 6). The right-most image expresses the normalized score for $\sigma = 0.02$. As apparent, the scores of the *pillow* category were the minimum, and the method showed degraded performance on the *car* category.

Table 2. Denoising performance of our method (CMD-Net) on the CMD-Net dataset is evaluated using the Chamfer Measure evaluation metric. The numbers in this table are C2C scores (see the metric in Section 4.1).

(a) Chamfer Measure, ($\sigma = 0.01$)							
Method	**Airplane**	**Bench**	**Car**	**Chair**	**Pillow**	**Rifle**	**Sofa**
Noisy	0.56	0.59	0.81	0.73	0.79	0.44	0.83
Ours	0.31	0.54	0.77	0.46	0.57	0.42	0.51
(b) Chamfer Measure, ($\sigma = 0.015$)							
Method	**Airplane**	**Bench**	**Car**	**Chair**	**Pillow**	**Rifle**	**Sofa**
Noisy	1.03	1.06	1.18	1.18	1.34	0.90	1.26
Ours	0.57	0.58	0.83	0.48	0.69	0.52	0.65
(c) Chamfer Measure, ($\sigma = 0.02$)							
Method	**Airplane**	**Bench**	**Car**	**Chair**	**Pillow**	**Rifle**	**Sofa**
Noisy	1.67	1.71	1.64	1.78	2.05	1.60	1.80
Ours	0.68	0.80	0.96	0.74	0.80	0.64	0.84

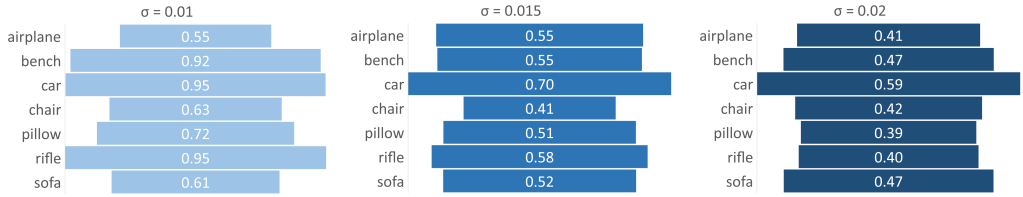

Figure 6. C2C scores generated by CMD-Net are normalized by their *noisy* counterparts. Left-most, middle, and right-most images are of $\sigma = 0.01$, $\sigma = 0.015$, and $\sigma = 0.02$, respectively.

The normalized errors are presented in Figure 7, the left image of which presents the cumulative errors, while the right one indicates the weights of noise levels on the cumulative errors. On the left, we see that when the normalized errors of all noise-levels are summed up, our method performed best on the *chair* category, and *airplane*, *sofa*, and *pillow* subsequently follow. The categories of *bench* and *rifle* were relatively degraded, and the CMD-Net lowest record was on the *car* category. Given the right image, it is apparent that the error scores of the low-level noise had the highest weight, demonstrating that our method showed better performance as the noise-level increased.

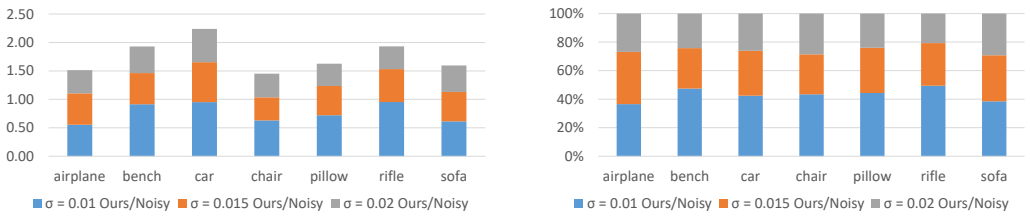

Figure 7. Error analysis for the normalized C2C scores. Left image explains the cumulative error, while the right one shows the weights of the errors for low-, mid-, and high-level noise.

We lastly observe the influence of a different registration approach altering the self-supervised canonical mapping part of CMD-Net. We changed the Canonical Mapping part of our network with PointNetLK (PNLK) [52] and named it PNLKD-Net. We show the results in Table 3 in the categories in which our method most underperformed: *car*, *bench*, *rifle*. When we alter the canonical mapping part of our network with PNLK, our method outputs increased performance ensuring more accurate canonicalization results. CMD-Net,

in turn, showed overperformance for the *car* and *rifle* categories. This clearly depicts the merit of our complete **self-supervised** methodology circumventing the dependency on pre-aligned data.

Table 3. Influence of PointNetLK on the denoising performance.

	$\sigma = 0.01$			$\sigma = 0.015$			$\sigma = 0.02$		
Method	**Bench**	**Car**	**Rifle**	**Bench**	**Car**	**Rifle**	**Bench**	**Car**	**Rifle**
Noisy	0.59	0.81	0.44	1.06	1.18	0.90	1.71	1.64	1.60
CMD-Net	0.54	**0.77**	**0.77**	0.58	**0.83**	**0.52**	0.80	**0.96**	**0.64**
PNLK-D Net	**0.52**	0.78	0.90	**0.50**	0.90	0.63	**0.71**	1.02	0.81

Figure 8 presents several qualitative results for $\sigma = 0.02$. In the figure, the first row depicts rotated and noise-corrupted test input instances from the *airplane* and *car* categories. These instances were fed into CMD-Net, the canonical mapping part of which first canonicalized them. In the second row of the figure, the test inputs and canonicalization results are overlaid, and the third row depicts canonicalization only. The canonicalized inputs were then fed into the denoising part of the network, and the final results are given in the fourth row of the figure. This characteristic of CMD-Net alleviates the dependency on pre-aligned data for the denoising problem.

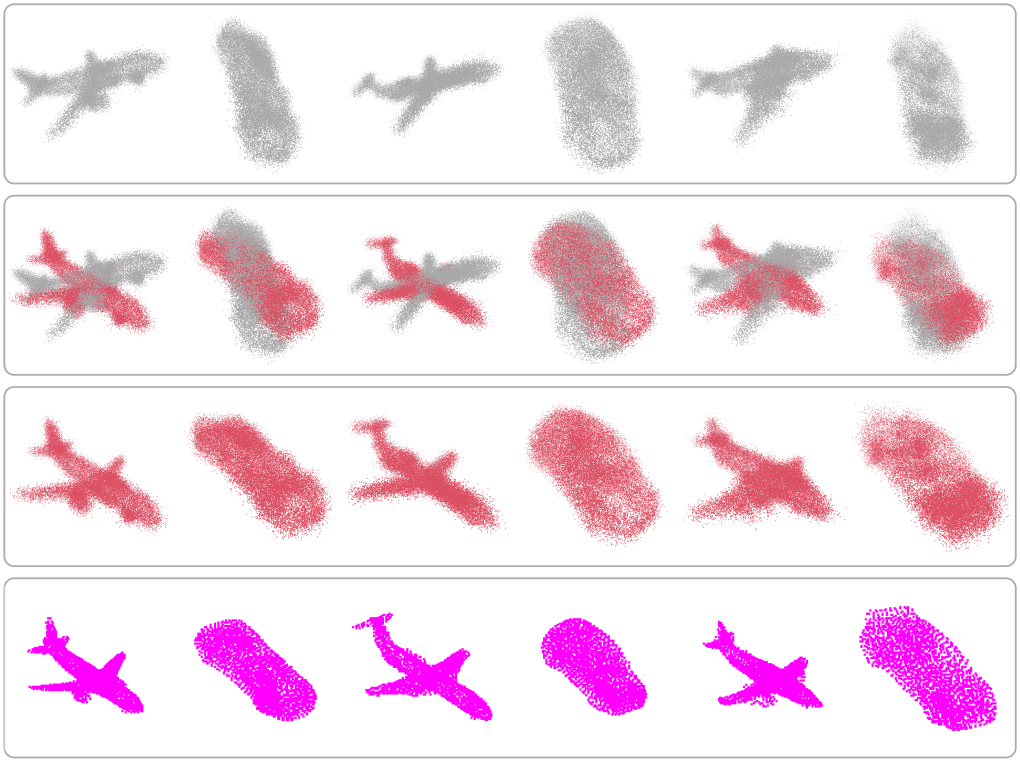

Figure 8. Sample results from experiments on CMD-Net dataset. Instances in the 1first row shows rotated noise-corrupted point clouds in the **testing stage** ($\sigma = 0.02$). In the second and the third rows, CMD-Net canonicalizes test inputs, and the last row depicts denoised instances in the canonical space. Please compare this figure with Figure 5.

4.5. Experiments on Real Data

After measuring the performance of our method on GPD-Net and CMD-Net datasets, we perforedm experiments on real data. The NOCS dataset [14] was presented for the category-level 6D object pose and size estimation problem, and the Mask RCNN [53] was used in [14] to hypothesise initial 2D bounding boxes of the interested objects. Regarding

the object viewpoints, true positives of Mask-RCNN involve data from background clutter. In addition, the camera viewpoint and the imperfections on the acquisition sensor give rise to missing points on the object surface in the depth channel. We tested our method on the *laptop* category of this dataset and demonstrated the superiority of CMD-Net across background clutter and missing pixels (points) through several qualitative examples. In Figure 9, each triplet contains arbitrarily rotated test instances. As depicted, they are cluttered, and the objects' surfaces are corrupted by missing points. Despite this fact, our method can successfully canonicalizes and denoises test images. The resilience of CMD-Net emanates from: (i) the training strategy we employ. Since our complete algorithm is capable of learning the convex hull of the shapes, it learns the deteriorating effects and eliminates those in the test process. We achieve this elimination as follows: During the self-supervised training phase, we simulated the clutter coming from both background and foreground. Our method extends the convex hull of the shapes injecting the shapes 3D outlier points. Optimizing the loss between the outlier-corrupted and noise-free shapes, the method learns to remove the cluttered data. (ii) the design of CMD-Net. Throughout our research, we initially started engineering our method with PointNet; however, it was not effective across missing points, thus being relatively unable to correctly canonicalize and denoise the input. Therefore, we re-engineered our model using PointNet++-based modules, hierarchically learning given clouds.

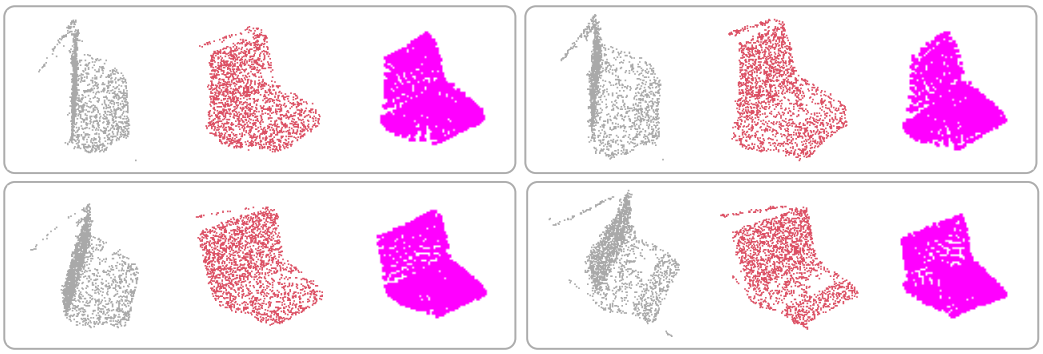

Figure 9. Sample results from experiments on real data. Our method can deal with clutter and missing pixels (points) challenges. In each triplet, left is the rotated input cloud, middle is the canonicalization, and the right-most is the denoised and the canonicalized.

5. Conclusions

In this paper, we have introduced the Canonical Mapping and Denoising Network (CMD-Net) for *category-level 3D shape denoising through canonicalization*. Unlike the standard learning-based denoising methods that predict displacement of noise-corrupted points from the underlying surface, CMD-Net formulates denoising as a 3D semantic shape correspondence estimation task. Exploring ordered intrinsic structure points is particularly important to eliminate corruption from unseen instances of given categories, thus bringing increased resilience across shrinkage and outliers.

From the data point of view, the strengths of our method are four-fold: (i) our self-supervised design. Pseudo labels are automatically populated, and thus with no resort to data labels, our training is employed. (ii) Canonical Mapping. Being capable of canonicalizing noise-corrupted clouds under arbitrary rotations, we circumvent the dependency on pre-aligned data. (iii) CMD-Net can operate on both synthetic and real data. (iv) The CMD-Net dataset is another plus of the presented research. It comprises rotated and noise-corrupted instances of given categories. These strengths enable us to increase the generalization capability of our method.

Regarding the challenges, CMD-Net has multiple merits: (i) clutter. The depth image of an hypothesis generated by any 2D (or 3D) object detector potentially contains background clutter that affects the representation of the object of interest. Our self-supervised method, learning the convex hull of the shapes, copes with this challenge and suppresses the points

existing beyond the object's hull. (ii) Missing pixels (points). The designs of both the canonical mapping part and the denoising part of CMD-Net are based on PointNet++. Learning the representations of objects in a hierarchical fashion, CMD-Net handles missing points. (iii) Intra-class variation. CMD-Net leverages intrinsic structure points to learn the structures of the instances given in a category of interest. This correspondingly enables our method to handle intra-class variations.

Despite the advantages of CMD-Net, there are limitations: (i) Number of structure points and GPU memory. CMD-Net denoises the interested objects exploring their intrinsic structures. In accordance with our network design and interested problem, the generation of a high number of structure points gives rise to an increase in the memory usage exceeding the limits of resources. In such cases, we generate a limited number of structure points and then employ upsampling. (ii) Canonicalization. Currently, CMD-Net operates in a relatively narrower search space for canonicalizing any arbitrarily rotated objects.

In the future, we are planning to improve the performance of CMD-Net, particularly on arbitrarily rotated objects. Investigating the effect of our method on the instances that have severe topological discrepancies will be a potential research direction. One of our aims will be to extend the search space of CMD-Net for canonical mapping. We are planning to define novel loss functions to improve the performance of our method for canonicalization purposes.

Funding: This research received no external funding.

Informed Consent Statement: Not applicable.

Data Availability Statement: One project page for this research is under construction. The project web-page will be announced in the following web-page: https://avesis.medeniyet.edu.tr/caner.sahin/en.

Acknowledgments: The author would like to thank the anonymous reviewers and journal editors for their constructive comments and suggestions, which improve the quality of the paper.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep learning on point sets for 3D classification and segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017.
2. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Adv. Neural Inf. Process. Syst.* **2018**, *30*, 5099–5108.
3. Luo, S.; Hu, W. Score-based point cloud denoising. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 4583–4592.
4. Duan, C.; Chen, S.; Kovacevic, J. 3D point cloud denoising via deep neural network based local surface estimation. In Proceedings of the ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Brighton, UK, 12–17 May 2019; pp. 8553–8557.
5. Hermosilla, P.; Ritschel, T.; Ropinski, T. Total Denoising: Unsupervised learning of 3d point cloud cleaning. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 52–60.
6. Rakotosaona, M.J.; La Barbera, V.; Guerrero, P.; Mitra, N.J.; Ovsjanikov, M. Pointcleannet: Learning to denoise and remove outliers from dense point clouds. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2020; Volume 39, pp. 185–203.
7. Pistilli, F.; Fracastoro, G.; Valsesia, D.; Magli, E. Learning graph-convolutional representations for point cloud denoising. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 103–118.
8. Luo, S.; Hu, W. Differentiable manifold reconstruction for point cloud denoising. In Proceedings of the 28th ACM International Conference on Multimedia, Seattle, WA, USA, 12–16 October 2020; pp. 1330–1338.
9. Chen, Z.; Yin, K.; Fisher, M.; Chaudhuri, S.; Zhang, H. BAE-NET: Branched autoencoder for shape co-segmentation. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 8490–8499.
10. Sun, C.Y.; Zou, Q.F.; Tong, X.; Liu, Y. Learning adaptive hierarchical cuboid abstractions of 3D shape collections. *ACM Trans. Graph.* **2019**, *38*, 241. [CrossRef]
11. Chen, N.; Liu, L.; Cui, Z.; Chen, R.; Ceylan, D.; Tu, C.; Wang, W. Unsupervised Learning of Intrinsic Structural Representation Points. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 9121–9130.

12. Groueix, T.; Fisher, M.; Kim, V.G.; Russell, B.C.; Aubry, M. A papier-mâché approach to learning 3D surface generation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 216–224.
13. Tchapmi, L.P.; Kosaraju, V.; Rezatofighi, H.; Reid, I.; Savarese, S. TopNet: Structural point cloud decoder. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 383–392.
14. Wang, H.; Sridhar, S.; Huang, J.; Valentin, J.; Song, S.; Guibas, L.J. Normalized object coordinate space for category-level 6D object pose and size estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 2642–2651.
15. Tian, M.; Ang, M.H.; Lee, G.H. Shape Prior Deformation for Categorical 6D Object Pose and Size Estimation. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 530–546.
16. Chen, D.; Li, J.; Wang, Z.; Xu, K. Learning canonical shape space for category-level 6D object pose and size estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11973–11982.
17. Sahin, C.; Kim, T.K. Category-level 6D object pose recovery in depth images. In Proceedings of the European Conference on Computer Vision (ECCV) Workshops, Munich, Germany, 8–14 September 2018.
18. Cheng, A.C.; Li, X.; Sun, M.; Yang, M.H.; Liu, S. Learning 3D Dense Correspondence via Canonical Point Autoencoder. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 6608–6620.
19. Cheng, A.C.; Li, X.; Liu, S.; Sun, M.; Yang, M.H. Autoregressive 3D Shape Generation via Canonical Mapping. *arXiv* **2022**, arXiv:2204.01955.
20. Sajnani, R.; Poulenard, A.; Jain, J.; Dua, R.; Guibas, L.J.; Sridhar, S. ConDor: Self-Supervised Canonicalization of 3D Pose for Partial Shapes. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.
21. Yang, Y.; Feng, C.; Shen, Y.; Tian, D. FoldingNet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 206–215.
22. Deng, H.; Birdal, T.; Ilic, S. PPF-Foldnet: Unsupervised learning of rotation invariant 3D local descriptors. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 602–618.
23. Deprelle, T.; Groueix, T.; Fisher, M.; Kim, V.; Russell, B.; Aubry, M. Learning elementary structures for 3D shape generation and matching. In Proceedings of the Advances in Neural Information Processing Systems 32 (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
24. Zhao, Y.; Birdal, T.; Deng, H.; Tombari, F. 3D point capsule networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 1009–1018.
25. Shu, D.W.; Park, S.W.; Kwon, J. 3D point cloud generative adversarial network based on tree structured graph convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 3859–3868.
26. Fernandez-Labrador, C.; Chhatkuli, A.; Paudel, D.P.; Guerrero, J.J.; Demonceaux, C.; Gool, L.V. Unsupervised learning of category-specific symmetric 3D keypoints from point sets. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 546–563.
27. Jakab, T.; Tucker, R.; Makadia, A.; Wu, J.; Snavely, N.; Kanazawa, A. KeypointDeformer: Unsupervised 3D Keypoint Discovery for Shape Control. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 12783–12792.
28. Shi, R.; Xue, Z.; You, Y.; Lu, C. Skeleton Merger: An unsupervised aligned keypoint detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 19–25 June 2021; pp. 43–52.
29. Li, R.; Li, X.; Hui, K.H.; Fu, C.W. SP-GAN: Sphere-guided 3D shape generation and manipulation. *ACM Trans. Graph.* **2021**, *40*, 151. [CrossRef]
30. Tang, J.; Gong, Z.; Yi, R.; Xie, Y.; Ma, L. LAKe-Net: Topology-Aware Point Cloud Completion by Localizing Aligned Keypoints. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.
31. Suwajanakorn, S.; Snavely, N.; Tompson, J.J.; Norouzi, M. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In Proceedings of the Advances in Neural Information Processing Systems 31 (NeurIPS 2018), Montréal, QC, Canada, 3–8 December 2018.
32. Li, J.; Lee, G.H. USIP: Unsupervised stable interest point detection from 3D point clouds. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 361–370.
33. Poursaeed, O.; Jiang, T.; Qiao, H.; Xu, N.; Kim, V.G. Self-supervised learning of point clouds via orientation estimation. In Proceedings of the 2020 International Conference on 3D Vision (3DV), Fukuoka, Japan, 25–28 November 2020; pp. 1018–1028.
34. Bai, X.; Luo, Z.; Zhou, L.; Fu, H.; Quan, L.; Tai, C.L. D3Feat: Joint learning of dense detection and description of 3D local features. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 6359–6367.
35. You, Y.; Liu, W.; Ze, Y.; Li, Y.L.; Wang, W.; Lu, C. UKPGAN: A General Self-Supervised Keypoint Detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.

36. Zhao, W.; Liu, X.; Zhong, Z.; Jiang, J.; Gao, W.; Li, G.; Ji, X. Self-Supervised Arbitrary-Scale Point Clouds Upsampling via Implicit Neural Representation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.
37. Zhang, C.; Shi, J.; Deng, X.; Wu, Z. Upsampling Autoencoder for Self-Supervised Point Cloud Learning. *arXiv* **2022**, arXiv:2203.10768.
38. Thabet, A.; Alwassel, H.; Ghanem, B. Self-supervised learning of local features in 3D point clouds. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 938–939.
39. Afham, M.; Dissanayake, I.; Dissanayake, D.; Dharmasiri, A.; Thilakarathna, K.; Rodrigo, R. CrossPoint: Self-supervised cross-modal contrastive learning for 3D point cloud understanding. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–20 June 2022.
40. Gu, J.; Ma, W.-C.; Manivasagam, S.; Zeng, W.; Wang, Z.; Xiong, Y.; Su, H.; Urtasun, R. Weakly-supervised 3D shape completion in the wild. In *European Conference on Computer Vision*; Springer: Cham, Switzerland, 2020; pp. 283–299.
41. Sajnani, R.; Sanchawala, A.; Jatavallabhula, K.M.; Sridhar, S.; Krishna, K.M. DRACO: Weakly supervised dense reconstruction and canonicalization of objects. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, 30 May–5 June 2021; pp. 10302–10309.
42. Novotny, D.; Ravi, N.; Graham, B.; Neverova, N.; Vedaldi, A. C3DPO: Canonical 3D pose networks for non-rigid structure from motion. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27 October–2 November 2019; pp. 7688–7697.
43. Spezialetti, R.; Stella, F.; Marcon, M.; Silva, L.; Salti, S.; Stefano, L.D. Learning to orient surfaces by self-supervised spherical cnns. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5381–5392.
44. Sun, W.; Tagliasacchi, A.; Deng, B.; Sabour, S.; Yazdani, S.; Hinton, G.E.; Yi, K.M. Canonical capsules: Self-supervised capsules in canonical pose. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 24993–25005.
45. Chang, A.X.; Funkhouser, T.; Guibas, L.; Hanrahan, P.; Huang, Q.; Li, Z.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. ShapeNet: An Information-Rich 3D Model Repository. *arXiv* **2015**, arXiv:1512.03012.
46. Oztireli, A.C.; Guennebaud, G.; Gross, M. Feature preserving point set surfaces based on non-linear kernel regression. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2009; Volume 28, pp. 493–501.
47. Guennebaud, G.; Gross, M. Algebraic point set surfaces. In Proceedings of the SIGGRAPH07: Special Interest Group on Computer Graphics and Interactive Techniques Conference, San Diego, CA, USA, 5–9 August 2007.
48. Huang, H.; Wu, S.; Gong, M.; Cohen-Or, D.; Ascher, U.; Zhang, H.R. Edge-aware point set resampling. *ACM Trans. Graph.* **2013**, *32*, 9. [CrossRef]
49. Mattei, E.; Castrodad, A. Point cloud denoising via moving RPCA. In *Computer Graphics Forum*; Wiley Online Library: Hoboken, NJ, USA, 2017; Volume 36, pp. 123–137.
50. Wang, Y.; Sun, Y.; Liu, Z.; Sarma, S.E.; Bronstein, M.M.; Solomon, J.M. Dynamic graph CNN for learning on point clouds. *ACM Trans. Graph.* **2019**, *38*, 146. [CrossRef]
51. Zeng, J.; Cheung, G.; Ng, M.; Pang, J.; Yang, C. 3D point cloud denoising using graph Laplacian regularization of a low dimensional manifold model. *IEEE Trans. Image Process.* **2019**, *29*, 3474–3489. [CrossRef] [PubMed]
52. Aoki, Y.; Goforth, H.; Srivatsan, R.A.; Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7163–7172.
53. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.