

---

# InfoCD: A Contrastive Chamfer Distance Loss for Point Cloud Completion

---

Fangzhou Lin<sup>1,2\*</sup> Yun Yue<sup>1\*</sup> Ziming Zhang<sup>1†</sup> Songlin Hou<sup>1,3</sup> Kazunori D Yamada<sup>2</sup>  
Vijaya B Kolachalama<sup>4</sup> Venkatesh Saligrama<sup>4</sup>

<sup>1</sup>Worcester Polytechnic Institute, USA <sup>2</sup>Tohoku University, Japan

<sup>3</sup>Dell Technologies, USA <sup>4</sup>Boston University, USA

{flin2, yyue, zzhang15, shou}@wpi.edu, yamada@tohoku.ac.jp,  
{vkola, srv}@bu.edu

## Abstract

A point cloud is a discrete set of data points sampled from a 3D geometric surface. Chamfer distance (CD) is a popular metric and training loss to measure the distances between point clouds, but also well known to be sensitive to outliers. We propose *InfoCD*, a novel contrastive Chamfer distance loss, and learn to spread the matched points to better align the distributions of point clouds. As such *InfoCD* leads to an improved surface similarity metric. We show that minimizing InfoCD is equivalent to maximizing a lower bound of the mutual information between the underlying geometric surfaces represented by the point clouds, leading to a *regularized* CD metric which is robust and computationally efficient for deep learning. We conduct comprehensive experiments for point cloud completion using InfoCD and observe significant improvements consistently over all the popular baseline networks trained with CD-based losses, leading to new state-of-the-art results on several benchmark datasets. Demo code is available at <https://github.com/Zhang-VISLab/NeurIPS2023-InfoCD>.

## 1 Introduction

**Point Cloud Completion.** Point clouds, one of the most important data representations that can be easily acquired, play a key role in modern robotics and automation applications [1–3]. However, raw data of point clouds captured by existing 3D sensors are usually incomplete and sparse due to occlusion, limited sensor resolution, and light reflection [4–8], which can negatively impact the performance of downstream tasks that require high-quality representation, such as point cloud segmentation and detection. *Point cloud completion* [9] refers to the task of inferring the complete shape of an object or scene from incomplete raw point clouds. Recently, many (deep) learning based approaches have been introduced to point cloud completion ranging from supervised learning, self-supervised learning to unsupervised learning [10–15]. Among these methods, supervised learning with a general encoder-decoder structure is the prevailing architectural choice for many researchers, consistently achieving state-of-the-art results on mainstream benchmarks [16, 17, 8, 1, 18].

**Learning with Chamfer Distance (CD).** CD is a commonly employed metric in point cloud completion, for example in studies like [19, 20]. It assesses the dissimilarity in shape between two sets of point clouds by calculating the average distance from each point in one set to its nearest neighbor in the other set. Minimizing the Euclidean distances between matched points, as done in CD, is a known method that is sensitive to outliers, resulting in *clumping behavior* that involves

---

\*Co-first authors.

†Corresponding author.

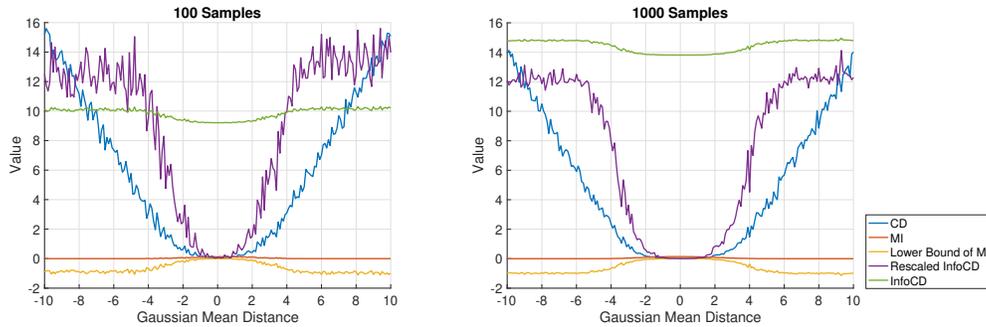


Figure 1: Illustration of comparison among CD, MI, and InfoCD with different numbers of samples.

a substantial number of many-to-one correspondences in point matching, forming small clusters visually. These observations can significantly deviate from the assumption of uniform sampling from the underlying geometric surfaces, which is commonly used to generate point clouds.

**Learning with Mutual Information (MI).** A more fundamental problem in measuring point cloud similarity (or distance alternatively) is: *How can we measure the similarities between the underlying geometric surfaces represented by the point clouds?* To address this issue, one potential method is to compute their MI by taking each point cloud as a discrete set of point samples from a random variable following a certain distribution, and then plugging both point clouds into the MI formula to measure the similarity of the two random variables. However, two limitations prevent us from using it directly as a loss function in learning: (1) It may be nontrivial to define the joint probability distribution between two point clouds since we do not have any prior knowledge, and (2) The requirement for extensive memory, necessary for handling a large volume of data points and deep learning models, could render the practical implementation of the MI (Mutual Information) loss unfeasible.

**Our Approach and Contributions.** Motivated by contrastive learning, we propose a novel contrastive Chamfer distance loss, namely *InfoCD*, to learn to spread the matched points for better distribution alignments between point clouds as well as accounting for surface similarity estimation. Similar to InfoNCE [21], our InfoCD minimization tends to maximize a lower bound of the MI between the underlying geometric surfaces represented by the point clouds. This indeed leads to a regularized CD loss that can be minimized effectively and efficiently with similar computational complexity to CD.

We further illustrate our high level ideas in Fig. 1. To plot such figures, we consider generating point clouds uniformly sampled from two 1D Gaussian distributions both with unit variance but different means, one fixed at 0 and the other varying from -10 to 10 at step-size of 0.1. We then compute CD, InfoCD (see Eq. 5 with  $\tau = 1$ ), the lower bound of MI (see Eq. 8), and MI where the joint probability distribution is computed based on the exponential of negative Euclidean distances. To better view the difference between the curves of CD and InfoCD, we also rescale the InfoCD curve by aligning its minimum and maximum values with those for CD based on linear scaling so that their value ranges are the same. With different numbers of samples, we can clearly see that: (1) InfoCD is more robust to outliers (*i.e.*, the cases far away from 0 mean) than CD by penalizing larger deviations with some similar numbers (though smaller numbers of samples have more fluctuations); (2) The sharper purple curves around 0 indicate that InfoCD may lead to better convergence; (3) The lower bound based on InfoCD can approximate MI well with the maximum at 0, which indicates that InfoCD can be taken as a good MI estimator and lead to better point distributions with fewer visual clusters.

To summarize, we list our main contributions as follows:

- We propose InfoCD by introducing contrastive learning into the CD loss, leading to a regularized CD loss for better point distribution alignments.
- We analyze the connection between InfoCD and MI as well as learning behavior with InfoCD.
- We achieve state-of-the-art results on popular benchmark datasets for point cloud completion.

## 2 Related Work

**Point Cloud Completion.** As the first learning-based point cloud completion network, PCN [10] extracts global features in a similar way PointNet [22] did and generates points through folding

operations as in FoldingNet [23]. In order to obtain local structures among points, Zhang et al. [24] proposed extracting multi-scale features from different layers in the feature extraction part to enhance the performance. CDN [25] uses a cascaded refinement network to bridge the local details of partial input and the global shape information together. Lyu et al. [26] proposed treating point cloud completion as a conditional generation problem in the framework of denoising diffusion probabilistic models (DDPM) [27]. Attention mechanisms such as Transformer [28], demonstrate their superiority in capturing long-range interaction as compared to CNNs' constrained receptive fields. For instance, to preserve more detailed geometry information for point cloud generation in the decoder, SA-Net [29] uses the skip-attention mechanism to merge local region information from the encoder and point features of the decoder. SnowflakeNet [17] and PointTr [16] pay extra attention to the decoder part with Transformer-like designs. PointAttN [1] was proposed solely based on Transformers.

**Distance Metrics for Point Clouds.** Distance in point clouds is a non-negative function that measures the dissimilarity between them. Since point clouds are inherently unordered, the shape-level distance is typically derived from statistics of pair-wise point-level distances based on a particular assignment strategy [20]. With relatively low computational cost fair design, CD and its variants are extensively used in learning-based methods for point cloud completion tasks [30, 26, 31, 32]. Earth Mover's Distance (EMD), which is another widely used metric, relies on finding the optimal mapping function from one set to the other by solving an optimization problem. In some cases, it is considered to be more reliable than CD, but it suffers from high computational overhead and is only suitable for sets with exact numbers of points [33, 34]. Recently, Wu et al. [20] propose a Density-aware Chamfer Distance (DCD) as a new metric for point cloud completion which can balance the behavior of CD and computational cost in EMD to a certain level. Lin et al. [35] proposed a HyperCD that computes CD in a hyperbolic space and achieves significantly better performance than DCD.

**Contrastive Learning.** Recently, learning representations from unlabeled data in contrastive way [36, 37] has been one of the most competitive research fields [21, 38–50]. Popular model structures like SimCLR [42] and Moco [45] apply the commonly used loss function InfoNCE [21] to learn a latent representation that is beneficial to downstream tasks. Several theoretical studies show that contrastive loss optimizes data representations by aligning the same image's two views (positive pairs) while pushing different images (negative pairs) away on the hypersphere [51–54]. A good survey on contrastive learning can be found in [55]. More recently, contrastive learning has found its way into point cloud applications as well. For instance, Tang et al. [56] proposed a contrastive boundary learning framework for point cloud segmentation. Yang et al. [57] proposed the mutual attention module and co-contrastive learning for point cloud object co-segmentation. Jiang et al. [58] proposed a guided point contrastive loss to enhance the feature representation and model generalization ability in semi-supervised settings for point cloud segmentation. Du et al. [59] proposed a self-contrastive learning approach for self-supervised point cloud representation learning. Wang et al. [60] proposed exploring whether maximizing the mutual information across shallow and deep layers is beneficial to improve representation learning on point clouds, leading to a new design of Maximizing Mutual Information (MMI) Module. Afham et al. [61] proposed CrossPoint, a simple cross-modal contrastive learning approach to learn transferable 3D point cloud representations with 2D images. Shao et al. [62] proposed a spatial consistency guided network (SCRnet) using contrastive learning for point cloud registration.

### 3 InfoCD Loss

#### 3.1 Preliminaries

**InfoNCE Loss.** In [21], the InfoNCE loss is defined as follows:

$$\mathcal{L}_{\text{InfoNCE}} = - \sum_x \log \left\{ \frac{\exp \left\{ \frac{1}{\tau} s(x^+, x; \theta) \right\}}{\exp \left\{ \frac{1}{\tau} s(x^+, x; \theta) \right\} + \sum_{x^-} \exp \left\{ \frac{1}{\tau} s(x^-, x; \theta) \right\}} \right\}, \quad (1)$$

where  $x, x^+, x^-$  denote the anchor, its positive and negative samples,  $s$  denotes a similarity function parametrized by  $\theta$ , and  $\tau \geq 0$  is a predefined temperature that controls the sharpness.

**Proposition 1** (InfoNCE vs. MI [21]). *Let  $c_t$  be the context at the  $t$ -th time step, and  $x_{t+k}$  be the future target. Then given a set of  $N$  random samples,  $\{x_1, \dots, x_N\}$ , containing one positive sample from the distribution  $p(x_{t+k}|c_t)$  and  $N - 1$  negative samples from the "proposal" distribution*

$p(x_{t+k})$ , we have

$$I(x_{t+k}, c_t) \geq \log(N) - \mathcal{L}_{\text{InfoNCE}}, \quad (2)$$

where  $I$  denotes the mutual information (MI) operator.

Please refer to the appendix in [21] for the proof. This proposition provides us with an alternative way to measure MI approximately and implicitly. If point clouds can fit into the setting of InfoNCE, we then may better estimate the underlying surface similarities from point clouds.

**Chamfer Distance Loss.** In the sequel, we denote  $(x_i, y_i)$  as the  $i$ -th point cloud pair, with  $x_i = \{x_{ij}\}$  and  $y_i = \{y_{ik}\}$  as two sets of 3D points, and  $d(\cdot, \cdot)$  as a certain distance metric. Then the CD loss for point clouds can be defined as follows:

$$\mathcal{L}_{\text{CD}}(x_i, y_i) = \ell_{\text{CD}}(x_i, y_i) + \ell_{\text{CD}}(y_i, x_i) = \frac{1}{|y_i|} \sum_k \min_j d(x_{ij}, y_{ik}) + \frac{1}{|x_i|} \sum_j \min_k d(x_{ij}, y_{ik}), \quad (3)$$

where  $|\cdot|$  denotes the cardinality of a set. For point cloud completion, function  $d$  usually refers to

$$d(x_{ij}, y_{ik}) = \begin{cases} \|x_{ij} - y_{ik}\| & \text{as L1-distance} \\ \|x_{ij} - y_{ik}\|^2 & \text{as L2-distance} \end{cases} \quad (4)$$

where  $\|\cdot\|$  denotes the Euclidean  $\ell_2$  norm of a vector.

### 3.2 Our Loss Function

Given the considerations above, we propose the following formula as our InfoCD loss:

$$\begin{aligned} \mathcal{L}_{\text{InfoCD}}(x_i, y_i) &= \ell_{\text{InfoCD}}(x_i, y_i) + \ell_{\text{InfoCD}}(y_i, x_i), \text{ where} \\ \ell_{\text{InfoCD}}(x_i, y_i) &= -\frac{1}{|y_i|} \sum_k \log \left\{ \frac{\exp\{-\frac{1}{\tau} \min_j d(x_{ij}, y_{ik})\}}{\sum_k \exp\{-\frac{1}{\tau} \min_j d(x_{ij}, y_{ik})\}} \right\} \\ &\propto \frac{1}{\tau |y_i|} \sum_k \min_j d(x_{ij}, y_{ik}) + \lambda \log \left\{ \sum_k \exp \left\{ -\frac{1}{\tau} \min_j d(x_{ij}, y_{ik}) \right\} \right\} \end{aligned} \quad (5)$$

$$= -\frac{1}{|y_i|} \sum_k \log \left\{ \frac{\exp\{-\frac{1}{\tau} \min_j d(x_{ij}, y_{ik})\}}{[\sum_k \exp\{-\frac{1}{\tau} \min_j d(x_{ij}, y_{ik})\}]^\lambda} \right\} \quad (6)$$

$$\implies \mathcal{L}_{\text{InfoCD}}(x_i, y_i) \propto \frac{1}{\tau} \mathcal{L}_{\text{CD}}(x_i, y_i) + \lambda \mathcal{R}(x_i, y_i) \quad (7)$$

with  $\mathcal{R}(x_i, y_i) = \log \left\{ \sum_{m,n} \exp \left\{ -\frac{1}{\tau} [\min_j d(x_{ij}, y_{in}) + \min_k d(x_{im}, y_{ik})] \right\} \right\}$  as a regularizer and  $\lambda = \frac{\tau'}{\tau} \in (0, 1]$  as a predefined constant controlling the trade-off between the loss and the regularizer. The smaller  $\mathcal{L}_{\text{CD}}(x_i, y_i)$  is, the larger  $\mathcal{R}(x_i, y_i)$  is accordingly. From this perspective, we can easily see that our InfoCD loss is equivalent to a regularized CD loss.

### 3.3 Analysis

**InfoCD vs. MI.** To see the connections between InfoCD and MI, we have the following lemma:

**Lemma 1.** Consider two point clouds  $x_i = \{x_{ij}\}, y_i = \{y_{ik}\}$  representing two underlying geometric surfaces  $\mathcal{X}_i, \mathcal{Y}_i$ . Now we introduce a new random variable  $z_{y_{ik}}$  whose probability distribution  $p(z_{y_{ik}} | \mathcal{X}_i = \mathcal{Y}_i)$  indicates how likely a point  $y_{ik}$  can be sampled from  $\mathcal{Y}_i$  conditional on  $\mathcal{X}_i = \mathcal{Y}_i$ , and ‘‘proposal’’ distribution  $p(z_{y_{ik}})$  indicates the likelihood of generating an arbitrary  $y_{ik}$ . With these notations, we will have

$$I(z_{y_{ik}}; \mathcal{X}_i = \mathcal{Y}_i) \geq \log(|y_i|) - \ell_{\text{InfoCD}}(x_i, y_i). \quad (8)$$

*Proof.* By following Prop. 1 and the proof in [21], we can take  $z_{y_{ik}}, \mathcal{X}_i = \mathcal{Y}_i$  as the replacements for  $x_{t+k}, c_t$ . Then samples from  $p(z_{y_{ik}} | \mathcal{X}_i = \mathcal{Y}_i)$  will be ‘‘positive’’, and ones from  $p(z_{y_{ik}})$  will be

“negative” in the language of contrastive learning. Now we can construct  $|y_i|$  groups, as illustrated in Fig. 2, where each group consists of the entire point cloud  $x_i$  and a point in  $y_i$ . We further can take an arbitrary group as a positive and the rest as negatives. Finally, by parametrizing  $\frac{p(z_{y_{ik}}|\mathcal{X}_i=\mathcal{Y}_i)}{p(z_{y_{ik}})} \propto \exp\{-\frac{1}{\tau'} \min_j d(x_{ij}, y_{ik})\}$  for positives and similarly for negatives, we can complete our proof.  $\square$

Therefore, based on this lemma, minimizing InfoCD tends to better estimate the lower bound of MI that indicates the underlying surface similarities between point clouds.

**Point Spread in Learning & Testing.** To better understand the behavior of different losses, we first introduce another lemma as follows:

**Lemma 2.** Consider an optimization problem  $\min_{\omega \in \Omega} \sum_i g(h(x_i; \omega))$  where  $h : \mathcal{X} \times \Omega \rightarrow \mathbb{R}, g : \mathbb{R} \rightarrow \mathbb{R}$  are both Lipschitz continuous functions,  $h$  is also smooth over  $\omega$ , and  $x_i \in \mathcal{X}$  is a data point. Then based on gradient descent, i.e.,  $\omega_{t+1} = \omega_t - \eta_t \sum_i \nabla g(h(x_i; \omega_t))$  with a learning rate  $\eta_t \geq 0$  at the  $t$ -th iteration and a gradient operator  $\nabla$ , it holds that given a new data point  $\tilde{x}$ ,

$$h(\tilde{x}; \omega_t) - h(\tilde{x}; \omega_{t+1}) \approx \sum_i \eta_t \frac{\partial g}{\partial h} \Big|_{(x_i; \omega_t)} \nabla h(x_i; \omega_t)^T \nabla h(\tilde{x}; \omega_t) = \eta_t \sum_i \frac{\partial g}{\partial h} \Big|_{(x_i; \omega_t)} \kappa(x_i, \tilde{x}; \omega_t), \quad (9)$$

where  $\omega_0$  is the initialization of  $\omega$ ,  $\kappa$  denotes a (neural) tangent kernel function parametrized by  $\omega_t$ ,  $(\cdot)^T$  is the matrix transpose operator, and  $\frac{\partial g}{\partial h} \Big|_{(x_i; \omega_t)}$  is the derivative of  $g$  over  $h$  at point  $(x_i; \omega_t)$ .

*Proof.* Using the linear approximation of  $h$  and the assumptions, we can easily prove this lemma.  $\square$

To connect this lemma with CD and InfoCD, let us first compute the gradients of  $\ell_{\text{CD}}$  and  $\ell_{\text{InfoCD}}$ :

$$\nabla \ell_{\text{CD}}(x_i, y_i) = \sum_k \frac{1}{|y_i|} \nabla d(x_{ik'}, y_{ik}), \quad (10)$$

$$\nabla \ell_{\text{InfoCD}}(x_i, y_i) = \frac{1}{\tau} \sum_k \left[ \frac{1}{|y_i|} - \tau' \frac{\exp\{-\frac{1}{\tau} d(x_{ik'}, y_{ik})\}}{\sum_k \exp\{-\frac{1}{\tau} d(x_{ik'}, y_{ik})\}} \right] \nabla d(x_{ik'}, y_{ik}), \quad (11)$$

where  $k' = \arg \min_j d(x_{ij}, y_{ik}), \forall k$ . By viewing  $d$  as  $h$  and  $\ell_{\text{CD}}$  (or  $\ell_{\text{InfoCD}}$ ) as  $g$ , we can see the weight  $\frac{\partial \ell_{\text{CD}}(x_{ik'}, y_{ik}; \omega_t)}{\partial d} = \frac{1}{|y_i|} > 0$  but  $\frac{\partial \ell_{\text{InfoCD}}(x_{ik'}, y_{ik}; \omega_t)}{\partial d} = \frac{1}{|y_i|} - \tau' \frac{\exp\{-\frac{1}{\tau} d(x_{ik'}, y_{ik})\}}{\sum_k \exp\{-\frac{1}{\tau} d(x_{ik'}, y_{ik})\}} \in \mathbb{R}$ .

To simplify the analysis and explanation, assuming that the normalized gradients,  $\frac{\nabla d(\cdot, \cdot)}{\|\nabla d(\cdot, \cdot)\|}$ , can be viewed as random samples from a high dimensional (i.e., the number of network parameters that is much larger than the number of samples in mini-batches) normal distribution, then for two different random inputs  $(x_{im'}, y_{im}), (x_{in'}, y_{in})$ , it will be expected [63] that the corresponding gradients will be close to being orthogonal to each other, i.e.,  $\nabla d(x_{im'}, y_{ik})^T \nabla d(x_{in'}, y_{ik}) \approx 0$ . Now by substituting this assumption into Eq. 9, we can have

$$\ell_{\text{CD}}(x_i, y_i; \omega_t) - \ell_{\text{CD}}(x_i, y_i; \omega_{t+1}) \approx \eta_t \sum_k \frac{\partial \ell_{\text{CD}}(x_{ik'}, y_{ik}; \omega_t)}{\partial d} \|\nabla d(x_{ik'}, y_{ik})\|^2, \quad (12)$$

$$\ell_{\text{InfoCD}}(x_i, y_i; \omega_t) - \ell_{\text{InfoCD}}(x_i, y_i; \omega_{t+1}) \approx \frac{\eta_t}{\tau} \sum_k \frac{\partial \ell_{\text{InfoCD}}(x_{ik'}, y_{ik}; \omega_t)}{\partial d} \|\nabla d(x_{ik'}, y_{ik})\|^2, \quad (13)$$

with another assumption that the matched point pairs keep unchanged over iterations. From this perspective, we can easily see that the *negative* weights of  $\frac{\partial \ell_{\text{InfoCD}}(x_{ik'}, y_{ik}; \omega_t)}{\partial d}$  with smaller distances will push the predicted points away from the matched ground-truth points, while *positive* weights will tend to reduce the distances in both metrics, as illustrated in Fig. 3. In this contrastive way, the predicted points will be more likely to be spread for better alignment with the ground-truth points.

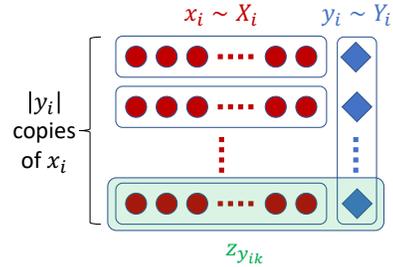
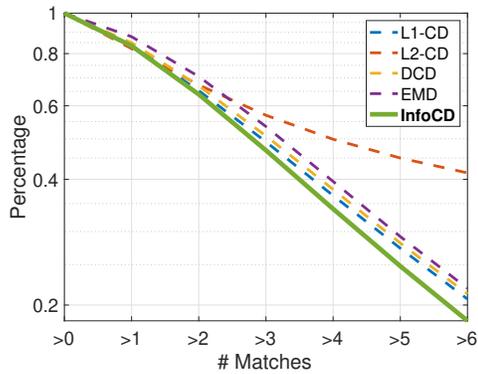
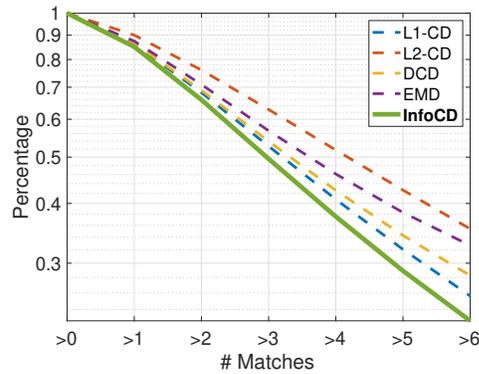


Figure 2: Illustration of group construction process for MI estimation.



(a) Prediction  $\rightarrow$  Ground truth



(b) Ground truth  $\rightarrow$  Prediction

Figure 4: Comparison on point percentage vs. the number of matches per point using CP-Net [64] on ShapeNet-Part [65] dataset, where the CP-Net is trained with five different loss functions. Clearly, InfoCD has better point spread and thus distribution alignments.

To demonstrate the capability of our InfoCD loss to spread the matched points at test time, we illustrate some comparison results in Fig. 4, where we can see clearly that InfoCD can significantly reduce the numbers of many-to-one matched points, leading to better point distribution alignments.

**Convergence.** In general, there is no guarantee that training networks with InfoCD using (stochastic) gradient descent will converge if the matched point pairs between the predictions and ground truth are frequently changed. However, empirically we observe that such training stabilization can be efficiently reached using different networks on different datasets as well. For instance, as illustrated in Fig. 5(a)<sup>3</sup>, InfoCD with L1-distance in Eq. 4 behaves similarly in terms of convergence rate to L1-CD that has convergence guarantee, but reduces the loss more significantly.

As illustrated in Fig. 5(b), ideally CD aims to reach the status where all the predicted points will be aligned perfectly with ground truth with no errors by consistently minimizing the distances. Such a requirement may be so strict that in learning the optimization may be much easier to be stuck at the suboptimal solutions, *e.g.*, forming clusters around some points. In contrast, InfoCD aims to align the point distributions with sufficiently small errors, as illustrated in Fig. 5(c) with the dotted lines indicating a Voronoi diagram for the ground-truth point cloud. This is much less strict than CD, and thus will be more likely to locate better solutions. In fact, the movements of predicted points in Fig. 3(b) provide an effective way towards learning such an alignment in Fig. 5(c) by avoiding bad suboptimal solutions.

**Choice of  $d$  in InfoCD.** In particular, we utilize L1-distance in Eq. 4 as the choice for  $d$  in InfoCD, because it is unbiased to the distance, *i.e.*,  $\nabla d(x_{ij}, y_{ik}) = \nabla \|x_{ij} - y_{ik}\|$ . In practice, we observe that using  $d(x_{ij}, y_{ik}) = \|x_{ij} - y_{ik}\|^p, p > 0, p \neq 1$ , the performance of InfoCD is very unstable for different networks

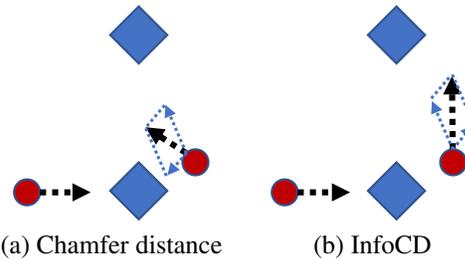


Figure 3: Illustration of the moving directions of matched points using (a) Chamfer distance or (b) InfoCD. Blue: ground truth; Red: predictions.

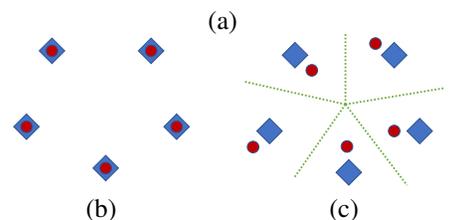
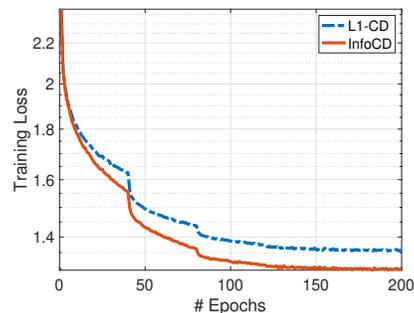


Figure 5: Illustration of (a) training loss using CP-Net on ShapeNet-Part, and ideal point alignment with (b/c) CD/InfoCD.

<sup>3</sup>The two loss curves are aligned for a better view so that the starting values are identical.

across different datasets. This is understandable: in the extreme cases where  $\|x_{ij} - y_{ik}\| = 0$ ,  $p > 1$  would make the predicted point unchanged, while  $0 < p < 1$  would make the move of the predicted point very far away. Such updates will violate the goal of InfoCD as shown in Fig. 5(c). Other distance metrics will be investigated in our future work.

## 4 Experiments

**Datasets.** We conducted experiments for point cloud completion on the following datasets:

- *PCN* [10]: This is a subset of ShapeNet [66] with shapes from 8 categories. The incomplete point clouds are generated by back-projecting 2.5D depth images from 8 viewpoints in order to simulate real-world sensor data. For each shape, 16,384 points are uniformly sampled from the mesh surfaces as complete ground truth, and 2,048 points are sampled as partial input [10, 8].
- *Multi-view partial point cloud (MVP)* [67]: This dataset covers 16 categories with 62,400 and 41,600 pairs for training and testing, respectively. It renders the partial 3D shapes from 26 uniformly distributed camera poses for each 3D CAD model selected from ShapeNet [68], and the ground-truth point cloud is sampled via Poisson Disk Sampling (PDS).
- *ShapeNet-55/34* [16]: ShapeNet-55 contains 55 categories in ShapeNet with 41,952 shapes for training and 10,518 shapes for testing. ShapeNet-34 uses a subset of 34 categories for training and leaves 21 unseen categories for testing where 46,765 object shapes are used for training, 3,400 for testing on seen categories and 2,305 for testing on novel (unseen) categories. In both datasets, 2,048 points are sampled as input and 8,192 points as ground truth. Following the same evaluation strategy with [16], 8 fixed viewpoints are selected and the number of points in the partial point cloud is set to 2,048, 4,096 or 6,144 (25%, 50% or 75% of a complete point cloud) which corresponds to three difficulty levels of *simple*, *moderate* and *hard* in the test stage.
- *ShapeNet-Part* [65]: This is a subset of ShapeNetCore [66] 3D meshes, containing 17,775 different 3D meshes with 16 categories. The ground-truth point clouds were created by sampling 2,048 points uniformly on each mesh. The partial point clouds were generated by randomly selecting a viewpoint as a center among multiple viewpoints and removing points within a certain radius from the complete data. The number of points we remove from each point cloud is 512.

**Implementation.** We considered three state-of-the-art networks, CP-Net [64], PointAttN [1] and SeedFormer [8], as our backbone networks for comparison and analysis. We also applied InfoCD to almost all the popular completion networks, *i.e.*, PCN [10], FoldingNet [23], TopNet [69], MSN [33], Cascaded [25], VRC [67], PMP-Net [73], PoinTr [16], SnowflakeNet [17], to verify its performance by replacing the original CD loss wherever it occurs. We performed the same replacement for all the other comparative losses in our experiments. We trained all these networks from scratch using PyTorch, optimized by either Adam [74] or AdamW [75]. Hyperparameters such as learning rates, batch sizes and balance factors in the original losses for training baseline

Table 1: Comparison on PCN in terms of per-point L1-CD  $\times 1000$ .

Methods	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat	Avg.
TopNet [69]	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12	12.15
AtlasNet [70]	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61	10.85
GRNet [71]	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04	8.83
CRN [25]	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05	8.51
NSFA [24]	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48	8.06
FBNet [72]	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18	6.94
PCN [10]	5.50	22.70	10.63	8.70	11.00	11.34	11.68	8.59	11.27
HyperCD [35] + PCN	5.95	<b>11.62</b>	<b>9.33</b>	12.45	12.58	13.10	<b>9.82</b>	9.85	<b>10.59</b>
<b>InfoCD + PCN</b>	<b>5.07</b>	22.27	10.18	<b>8.26</b>	<b>10.57</b>	<b>10.98</b>	11.23	<b>8.15</b>	10.83
FoldingNet [23]	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99	14.31
HyperCD + FoldingNet	<b>7.89</b>	12.90	<b>10.67</b>	14.55	<b>13.87</b>	<b>14.09</b>	11.86	<b>10.89</b>	<b>12.09</b>
<b>InfoCD+FoldingNet</b>	7.90	<b>12.68</b>	10.83	<b>14.04</b>	14.05	14.56	<b>11.61</b>	11.45	12.14
PMP-Net [73]	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25	8.73
HyperCD + PMP-Net	5.06	10.67	9.30	9.11	6.83	11.01	8.18	7.03	8.40
<b>InfoCD+PMP-Net</b>	<b>4.67</b>	<b>10.09</b>	<b>8.87</b>	<b>8.59</b>	<b>6.38</b>	<b>10.48</b>	<b>7.51</b>	<b>6.75</b>	<b>7.92</b>
PoinTr [16]	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29	8.38
HyperCD + PoinTr	4.42	9.77	8.22	8.22	6.62	9.62	6.97	6.67	7.56
<b>InfoCD + PoinTr</b>	<b>4.06</b>	<b>9.42</b>	<b>8.11</b>	<b>7.81</b>	<b>6.21</b>	<b>9.38</b>	<b>6.57</b>	<b>6.40</b>	<b>7.24</b>
SnowflakeNet [17]	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40	7.21
HyperCD + SnowflakeNet	<b>3.95</b>	9.01	7.88	7.37	<b>5.75</b>	8.94	<b>6.19</b>	6.17	6.91
<b>InfoCD + SnowflakeNet</b>	4.01	<b>8.81</b>	<b>7.62</b>	<b>5.51</b>	5.80	<b>8.91</b>	6.21	<b>5.05</b>	<b>6.86</b>
PointAttN [1]	3.87	9.00	7.63	7.43	5.90	8.68	6.32	6.09	6.86
HyperCD + PointAttN	3.76	8.93	7.49	7.06	5.61	8.48	6.25	<b>5.92</b>	6.68
<b>InfoCD + PointAttN</b>	<b>3.72</b>	<b>8.87</b>	<b>7.46</b>	<b>7.02</b>	<b>5.60</b>	<b>8.45</b>	<b>6.23</b>	<b>5.92</b>	<b>6.65</b>
SeedFormer [8]	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85	6.74
HyperCD + SeedFormer	3.72	<b>8.71</b>	7.79	<b>6.83</b>	5.11	<b>8.61</b>	<b>5.82</b>	5.76	6.54
<b>InfoCD + SeedFormer</b>	<b>3.69</b>	8.72	<b>7.68</b>	6.84	<b>5.08</b>	<b>8.61</b>	5.83	<b>5.75</b>	<b>6.52</b>

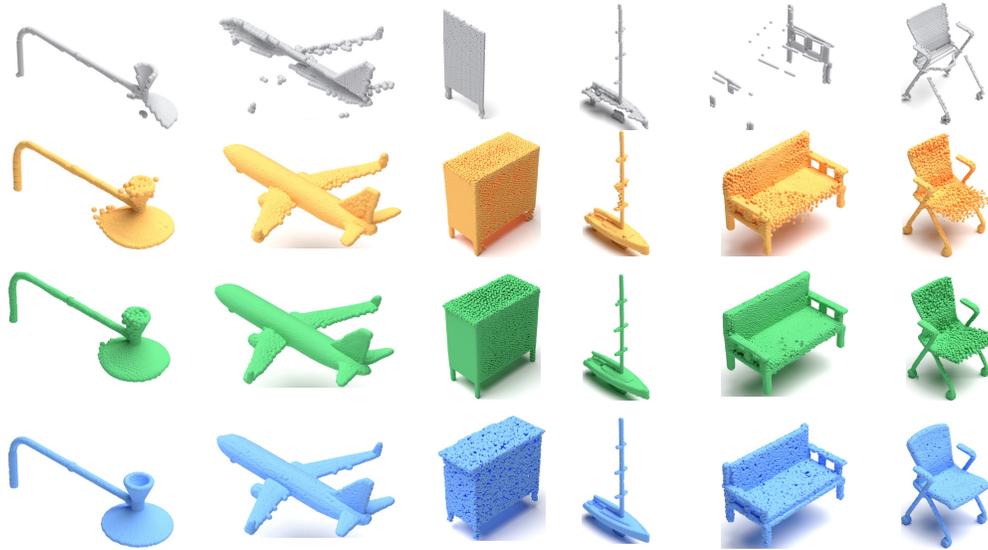


Figure 6: Visual result comparison on PCN. **Row-1:** Inputs of incomplete point clouds. **Row-2:** Outputs of Seedformer with CD. **Row-3:** Outputs of Seedformer with InfoCD. **Row-4:** Ground truth.

Table 2: Completion results on MVP in terms of  $L2-CD \times 10^4$  and  $EMD \times 10^2$ .

Methods		airplane	cabinet	car	chair	lamp	sofa	tabel	watercraft	bed	bench	bookshelf	bus	guitar	motorbike	pistol	skateboard	Avg.
CD	PCN [10]	4.50	8.83	6.41	13.01	21.33	9.90	12.86	9.46	20.00	10.26	14.63	4.94	1.73	6.17	5.84	5.76	9.78
	<b>InfoCD+PCN</b>	<b>3.95</b>	<b>8.82</b>	<b>6.38</b>	<b>12.03</b>	<b>17.43</b>	<b>9.63</b>	<b>12.41</b>	<b>8.69</b>	<b>18.92</b>	<b>8.75</b>	<b>13.40</b>	<b>5.02</b>	<b>1.84</b>	<b>6.06</b>	<b>5.81</b>	<b>4.37</b>	<b>9.41</b>
	TopNet [69]	4.12	9.84	7.44	13.26	18.64	10.77	12.95	8.98	19.99	9.21	16.06	5.47	2.36	7.06	7.04	4.68	10.30
	<b>InfoCD+TopNet</b>	<b>3.98</b>	<b>9.81</b>	<b>7.42</b>	<b>13.24</b>	<b>17.87</b>	<b>10.52</b>	<b>12.45</b>	<b>8.93</b>	<b>19.69</b>	<b>8.52</b>	<b>14.62</b>	<b>5.42</b>	<b>2.35</b>	<b>7.05</b>	<b>6.52</b>	<b>4.21</b>	<b>10.01</b>
	MSN [33]	2.73	8.92	6.50	10.75	13.37	9.26	10.17	7.70	17.27	6.64	12.10	5.21	1.37	4.59	4.62	3.38	7.99
	<b>InfoCD+MSN</b>	<b>2.28</b>	<b>8.51</b>	<b>6.03</b>	<b>10.18</b>	<b>12.91</b>	<b>8.87</b>	<b>9.72</b>	<b>7.24</b>	<b>16.82</b>	<b>6.21</b>	<b>11.67</b>	<b>4.79</b>	<b>0.91</b>	<b>4.15</b>	<b>4.17</b>	<b>2.97</b>	<b>7.56</b>
	Cascaded [25]	2.54	8.62	5.93	8.76	11.22	8.46	9.20	6.61	14.63	6.09	10.17	4.95	1.55	4.34	4.23	3.19	7.25
	<b>InfoCD+Cascaded</b>	<b>2.43</b>	<b>8.05</b>	<b>5.73</b>	<b>8.77</b>	<b>10.47</b>	<b>8.24</b>	<b>9.18</b>	<b>6.41</b>	<b>14.37</b>	<b>6.02</b>	<b>10.45</b>	<b>4.70</b>	<b>1.45</b>	<b>4.23</b>	<b>4.16</b>	<b>2.99</b>	<b>7.12</b>
VRC [67]	2.20	7.92	5.60	7.49	8.15	7.45	7.52	5.20	11.90	4.88	7.39	4.53	1.15	3.90	3.44	3.22	6.09	
<b>InfoCD+VRC</b>	<b>2.03</b>	<b>7.88</b>	<b>5.41</b>	<b>7.31</b>	<b>7.92</b>	<b>7.22</b>	<b>7.30</b>	<b>5.01</b>	<b>11.67</b>	<b>4.65</b>	<b>7.14</b>	<b>4.30</b>	<b>0.97</b>	<b>4.68</b>	<b>3.19</b>	<b>3.04</b>	<b>5.87</b>	
EMD	PCN	4.70	7.99	5.75	6.90	11.99	5.32	6.60	5.40	9.84	4.85	7.87	5.24	10.56	4.93	4.86	5.59	6.80
	<b>InfoCD+PCN</b>	<b>3.75</b>	<b>5.59</b>	<b>3.97</b>	<b>5.23</b>	<b>10.11</b>	<b>4.42</b>	<b>5.45</b>	<b>4.67</b>	<b>7.29</b>	<b>4.21</b>	<b>5.55</b>	<b>3.53</b>	<b>6.12</b>	<b>4.02</b>	<b>4.70</b>	<b>3.84</b>	<b>5.17</b>
	TopNet	4.89	6.30	4.07	7.01	10.75	6.47	7.50	4.68	8.09	6.27	6.80	3.50	4.21	4.26	6.02	3.49	6.18
	<b>InfoCD+TopNet</b>	<b>4.47</b>	<b>6.02</b>	<b>3.81</b>	<b>6.82</b>	<b>10.21</b>	<b>6.05</b>	<b>7.12</b>	<b>4.37</b>	<b>7.87</b>	<b>5.87</b>	<b>6.02</b>	<b>3.31</b>	<b>4.06</b>	<b>4.11</b>	<b>5.82</b>	<b>3.15</b>	<b>5.72</b>
	MSN	2.75	4.02	3.47	4.44	6.28	3.74	4.46	3.82	5.27	3.34	4.28	2.92	2.07	3.30	3.62	2.21	3.94
	<b>InfoCD+MSN</b>	<b>2.18</b>	<b>3.51</b>	<b>2.97</b>	<b>3.96</b>	<b>5.77</b>	<b>3.21</b>	<b>3.92</b>	<b>3.24</b>	<b>4.75</b>	<b>2.86</b>	<b>3.79</b>	<b>2.41</b>	<b>1.50</b>	<b>2.81</b>	<b>3.09</b>	<b>2.64</b>	<b>3.38</b>
	Cascaded	3.03	6.82	5.44	5.16	7.55	5.57	4.73	4.88	6.85	3.51	5.71	5.81	5.30	4.30	4.42	3.44	5.18
	<b>InfoCD+Cascaded</b>	<b>2.87</b>	<b>6.23</b>	<b>5.39</b>	<b>5.06</b>	<b>7.10</b>	<b>5.45</b>	<b>4.57</b>	<b>4.79</b>	<b>6.42</b>	<b>3.49</b>	<b>5.15</b>	<b>5.72</b>	<b>3.58</b>	<b>4.19</b>	<b>4.27</b>	<b>2.91</b>	<b>5.01</b>
VRC	3.03	7.57	6.14	5.49	6.15	5.80	4.65	4.97	6.58	3.45	5.28	6.59	3.08	4.45	4.56	3.20	5.27	
<b>InfoCD+VRC</b>	<b>2.68</b>	<b>7.26</b>	<b>5.83</b>	<b>5.15</b>	<b>5.82</b>	<b>5.49</b>	<b>4.36</b>	<b>4.68</b>	<b>6.22</b>	<b>3.13</b>	<b>4.97</b>	<b>6.26</b>	<b>2.77</b>	<b>4.13</b>	<b>4.15</b>	<b>2.89</b>	<b>4.97</b>	

networks were kept consistent with the baseline settings for fair comparisons. Hyperparameter  $\tau$  in InfoCD was tuned based on grid search, while  $\lambda$  was set to  $10^{-7}$  for all the experiments. We conducted our experiments on a server with 4 NVIDIA A100 80G GPUs and one with 10 NVIDIA Quadro RTX 6000 24G GPUs due to the large model sizes of some baseline networks.

**Evaluation.** We evaluated the best performance of all the methods using CD (lower is better). We also used F1-Score@1% [76] (higher is better) to evaluate the performance. Note that for the PCN method, we report the results based on the PyTorch implementation from PMP-Net. When using the original TensorFlow implementation, it achieves an average result of 10.28 on PCN with the default hyperparameters. However, when trained with our infoCD, it can achieve a lower value of 9.86.

#### 4.1 State-of-the-art Comparison

**PCN.** Following the literature, we report CD with L1-distance in Table 1. As we can see, InfoCD is able to improve the performance of all the baselines consistently and significantly, achieving new

state-of-the-art results. As discussed earlier, numerical metrics such as CD may not faithfully reflect the visual quality, we also provide qualitative evaluation results in Fig. 6, compared with results generated from Seedformer trained with the CD loss. As we can see, both models can reconstruct point clouds in general outlines to some extent, but the completion results with CD are more likely to suffer from distortion in several areas with high noise levels on the surface. In contrast, InfoCD can help the baseline network demonstrably better reconstruct point clouds in general outlines while maintaining the realistic details of the original ground truth with significant noise reduction.

**MVP.** We evaluate the universality of InfoCD on this dataset with several popular networks for tasks with higher diversities and different designs of architectures. Following the literature, we report CD with L2-distance and EMD in Table 2. Similar to Table 1, the introduction of InfoCD can consistently and significantly improve the performance of all different baselines in both two metrics.

**ShapeNet-55/34.** We evaluate the adaptability of InfoCD on both datasets for the tasks with higher diversities. Table 3 lists the L2-CD on three difficulty levels as well as the average. Following the literature, we show the results in 5 categories (Table, Chair, Plane, Car, and Sofa) whose numbers of training samples are more than 2,500. We also provide the results using the F1 metric. Again InfoCD has significantly improved the baseline models, especially when networks are simpler such as FoldingNet. Please refer to our supplementary for more qualitative evaluations.

Table 3: Results on ShapeNet-55 using L2-CD $\times$ 1000 and F1 score.

Methods	Table	Chair	Plane	Car	Sofa	CD-S	CD-M	CD-H	Avg.	F1
PFNet [77]	3.95	4.24	1.81	2.53	3.34	3.83	3.87	7.97	5.22	0.339
TopNet [69]	2.21	2.53	1.14	2.18	2.36	2.26	2.16	4.3	2.91	0.126
PCN [10]	2.13	2.29	1.02	1.85	2.06	1.94	1.96	4.08	2.66	0.133
GRNet [71]	1.63	1.88	1.02	1.64	1.72	1.35	1.71	2.85	1.97	0.238
FoldingNet [23]	2.53	2.81	1.43	1.98	2.48	2.67	2.66	4.05	3.12	0.082
<b>InfoCD + FoldingNet</b>	<b>2.14</b>	<b>2.37</b>	<b>1.03</b>	<b>1.55</b>	<b>2.04</b>	<b>2.17</b>	<b>2.50</b>	<b>3.46</b>	<b>2.71</b>	<b>0.137</b>
PoinTr [16]	0.81	0.95	0.44	0.91	0.79	0.58	0.88	1.79	1.09	0.464
<b>InfoCD + PoinTr</b>	<b>0.69</b>	<b>0.83</b>	<b>0.33</b>	<b>0.80</b>	<b>0.67</b>	<b>0.47</b>	<b>0.73</b>	<b>1.50</b>	<b>0.90</b>	<b>0.524</b>
SeedFormer [8]	0.72	0.81	0.40	0.89	0.71	0.50	0.77	1.49	0.92	0.472
HyperCD + SeedFormer	0.66	0.74	0.35	0.83	0.64	0.47	0.72	1.40	0.86	0.482
<b>InfoCD + SeedFormer</b>	<b>0.65</b>	<b>0.72</b>	<b>0.31</b>	<b>0.81</b>	<b>0.62</b>	<b>0.43</b>	<b>0.71</b>	<b>1.38</b>	<b>0.84</b>	<b>0.490</b>

On ShapeNet-34, we evaluate performances within 34 seen categories (same as training) as well as 21 unseen categories (not used in training) and list our results in Table 4. We can observe that, again, InfoCD can improve the performance of baseline models, indicating that InfoCD is highly generalizable for point cloud completion.

Table 4: Results on ShapeNet-34 using L2-CD $\times$ 1000 and F1 score.

Methods	34 seen categories					21 unseen categories				
	CD-S	CD-M	CD-H	Avg.	F1	CD-S	CD-M	CD-H	Avg.	F1
PFNet [77]	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
TopNet [69]	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PCN [10]	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
GRNet [71]	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
FoldingNet [23]	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
<b>InfoCD + FoldingNet</b>	<b>1.54</b>	<b>1.60</b>	<b>3.10</b>	<b>2.08</b>	<b>0.177</b>	<b>2.42</b>	<b>2.49</b>	<b>5.01</b>	<b>3.31</b>	<b>0.157</b>
PoinTr [16]	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
<b>InfoCD + PoinTr</b>	<b>0.47</b>	<b>0.69</b>	<b>1.35</b>	<b>0.84</b>	<b>0.529</b>	<b>0.61</b>	<b>1.06</b>	<b>2.55</b>	<b>1.41</b>	<b>0.493</b>
SeedFormer [8]	0.48	0.70	1.30	0.83	0.452	0.61	1.08	2.37	1.35	0.402
HyperCD + SeedFormer	0.46	0.67	1.24	0.79	0.459	0.58	1.03	2.24	1.31	0.428
<b>InfoCD + SeedFormer</b>	<b>0.43</b>	<b>0.63</b>	<b>1.21</b>	<b>0.75</b>	<b>0.581</b>	<b>0.54</b>	<b>1.01</b>	<b>2.18</b>	<b>1.24</b>	<b>0.449</b>

**ShapeNet-Part.** Previous results in Figs. 4 and 5(a) show the performance of CP-Net on this dataset. Below we summarize more results: Table 5 lists our comparison results of CP-Net trained with some popular losses in the forms of mean and standard deviation (std). From the perspective of the mean, it is clear that InfoCD outperforms the others. Moreover, from the perspective of the std, training with InfoCD is much more stable than the others. Fig. 7 shows the ablation study for learning rate ( $lr$ ) and  $\tau$  in Eq. 5 in terms of L2-CD. Overall, training with InfoCD is not sensitive to such hyperparameters.

Table 5: CP-Net Avg. results on ShapeNet-Part.

Loss	L2-CD $\times 10^3$
L1-CD	4.16 $\pm$ 0.028
L2-CD	4.82 $\pm$ 0.117
DCD [20]	5.74 $\pm$ 0.049
HyperCD	4.03 $\pm$ 0.007
<b>InfoCD</b>	<b>4.01<math>\pm</math>0.004</b>

## 5 Conclusion

In this paper, we propose a new loss for point cloud completion, namely InfoCD, a contrastive Chamfer distance loss. We show that by regularizing the CD loss with contrastive learning, InfoCD can better align the point distributions between prediction and ground truth, achieving the goal of

measuring the underlying geometric surfaces of the point clouds using mutual information (MI) estimation. In particular, we discuss and analyze the relations between InfoCD and MI, the moves of predicted points in learning, training convergence, loss landscapes, and the choice of distance metric in InfoCD. Comprehensive experiments have been conducted to demonstrate its effectiveness and efficiency using 7 networks on 5 datasets, leading to new state-of-the-art results.

**Limitations.** Due to the introduction of a new hyperparameter  $\tau$  in InfoCD, tuning hyperparameter based on grid search may need more effort, as illustrated in Fig. 7. Also due to the higher nonconvexity of InfoCD than CD, it may take more time (or epochs) to train complicated networks (*e.g.*, with large numbers of parameters, or architectures) with InfoCD.

## Acknowledgement

Fangzhou Lin was supported in part by the Support for Pioneering Research Initiated by the Next Generation (SPRING) from the Japan Science and Technology Agency. Yun Yue and Ziming Zhang were supported partially by NSF CCF-2006738. Dr. Kazunori D Yamada was supported in part by the Top Global University Project from the Ministry of Education, Culture, Sports, Science, and Technology of Japan (MEXT). Vijaya B Kolachalama was supported by the National Institutes of Health (R01-HL159620, R21-CA253498, R43-DK134273, RF1-AG062109, and P30-AG073104), the American Heart Association (20SFRN35460031), and the Karen Toffler Charitable Trust. Venkatesh Saligrama was supported by the Army Research Office Grant W911NF2110246, AFRLGrant FA8650-22-C1039, the National Science Foundation grants CCF2007350 and CCF-1955981. Computations were partially performed on the NIG supercomputer at ROIS National Institute of Genetics.

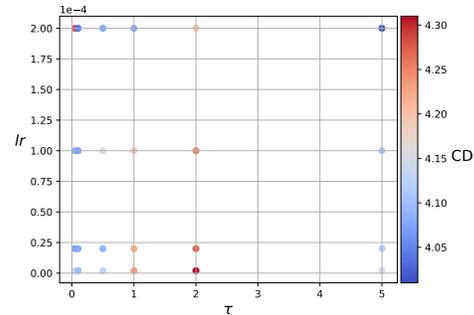


Figure 7: Ablation study on  $lr$  vs.  $\tau$ .

## References

- [1] Jun Wang, Ying Cui, Dongyan Guo, Junxia Li, Qingshan Liu, and Chunhua Shen. Pointattn: You only need attention for point cloud completion. *arXiv preprint arXiv:2203.08485*, 2022.
- [2] Changfeng Ma, Yang Yang, Jie Guo, Chongjun Wang, and Yanwen Guo. Completing partial point clouds with outliers by collaborative completion and segmentation. *arXiv preprint arXiv:2203.09772*, 2022.
- [3] Jieqi Shi, Lingyun Xu, Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Temporal point cloud completion with pose disturbance. *IEEE Robotics and Automation Letters*, 7(2):4165–4172, 2022.
- [4] Lequan Yu, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-net: Point cloud upsampling network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2790–2799, 2018.
- [5] Ruihui Li, Xianzhi Li, Pheng-Ann Heng, and Chi-Wing Fu. Point cloud upsampling via disentangled refinement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 344–353, 2021.
- [6] Shitong Luo and Wei Hu. Score-based point cloud denoising. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4583–4592, 2021.
- [7] Ren-Wu Li, Bo Wang, Chun-Peng Li, Ling-Xiao Zhang, and Lin Gao. High-fidelity point cloud completion with low-resolution recovery and noise-aware upsampling. *arXiv preprint arXiv:2112.11271*, 2021.
- [8] Haoran Zhou, Yun Cao, Wenqing Chu, Junwei Zhu, Tong Lu, Ying Tai, and Chengjie Wang. Seedformer: Patch seeds based point cloud completion with upsample transformer. *arXiv preprint arXiv:2207.10315*, 2022.
- [9] Antonio Alliegro, Diego Valsesia, Giulia Fracastoro, Enrico Magli, and Tatiana Tommasi. Denoise and contrast for category agnostic shape completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4629–4638, 2021.

- [10] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: point completion network. In *3DV*, 2018.
- [11] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *CVPR*, 2020.
- [12] Himangi Mittal, Brian Okorn, Arpit Jangid, and David Held. Self-supervised point cloud completion via inpainting. *arXiv preprint arXiv:2111.10701*, 2021.
- [13] Yingjie Cai, Kwan-Yee Lin, Chao Zhang, Qiang Wang, Xiaogang Wang, and Hongsheng Li. Learning a structured latent space for unsupervised point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5543–5553, 2022.
- [14] Zhaoxin Fan, Yulin He, Zhicheng Wang, Kejian Wu, Hongyan Liu, and Jun He. Reconstruction-aware prior distillation for semi-supervised point cloud completion. *arXiv preprint arXiv:2204.09186*, 2022.
- [15] Yiming Ren, Peishan Cong, Xinge Zhu, and Yuexin Ma. Self-supervised point cloud completion on real traffic scenes via scene-concerned bottom-up mechanism. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022.
- [16] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *ICCV*, 2021.
- [17] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflake-net: Point cloud completion by snowflake point deconvolution with skip-transformer. In *ICCV*, 2021.
- [18] Ben Fei, Weidong Yang, Wen-Ming Chen, Zhijun Li, Yikang Li, Tao Ma, Xing Hu, and Lipeng Ma. Comprehensive review of deep learning-based 3d point cloud completion processing and analysis. *IEEE Transactions on Intelligent Transportation Systems*, 2022.
- [19] Yulan Guo, Hanyun Wang, Qingyong Hu, Hao Liu, Li Liu, and Mohammed Bennamoun. Deep learning for 3d point clouds: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 43(12): 4338–4364, 2020.
- [20] Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Density-aware chamfer distance as a comprehensive metric for point cloud completion. In *Advances in Neural Information Processing Systems*, volume 34, pages 29088–29100, 2021.
- [21] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.
- [22] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017.
- [23] Yaoqing Yang, Chen Feng, Yiru Shen, and Dong Tian. Foldingnet: Point cloud auto-encoder via deep grid deformation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 206–215, 2018.
- [24] Wenxiao Zhang, Qingan Yan, and Chunxia Xiao. Detail preserved point cloud completion via separated feature aggregation. In *European Conference on Computer Vision*, pages 512–528. Springer, 2020.
- [25] Xiaogang Wang, Marcelo H Ang Jr, and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 790–799, 2020.
- [26] Zhaoyang Lyu, Zhifeng Kong, Xudong Xu, Liang Pan, and Dahua Lin. A conditional point diffusion-refinement paradigm for 3d point cloud completion. *arXiv preprint arXiv:2112.03530*, 2021.
- [27] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.
- [28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [29] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *CVPR*, 2020.

- [30] Haowen Deng, Tolga Birdal, and Slobodan Ilic. 3d local features for direct pairwise registration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3244–3253, 2019.
- [31] Kaiyi Zhang, Ximing Yang, Yuan Wu, and Cheng Jin. Attention-based transformation from latent features to point clouds. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 3291–3299, 2022.
- [32] Junshu Tang, Zhijun Gong, Ran Yi, Yuan Xie, and Lizhuang Ma. Lake-net: topology-aware point cloud completion by localizing aligned keypoints. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1726–1735, 2022.
- [33] Minghua Liu, Lu Sheng, Sheng Yang, Jing Shao, and Shi-Min Hu. Morphing and sampling network for dense point cloud completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11596–11603, 2020.
- [34] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pages 40–49. PMLR, 2018.
- [35] Fangzhou Lin, Yun Yue, Songlin Hou, Xuechu Yu, Yajun Xu, Kazunori D Yamada, and Ziming Zhang. Hyperbolic chamfer distance for point cloud completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14595–14606, October 2023.
- [36] Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546. IEEE, 2005.
- [37] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE, 2006.
- [38] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *arXiv preprint arXiv:1808.06670*, 2018.
- [39] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742, 2018.
- [40] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *European conference on computer vision*, pages 776–794. Springer, 2020.
- [41] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. *Advances in neural information processing systems*, 29, 2016.
- [42] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.
- [43] Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.
- [44] Junnan Li, Pan Zhou, Caiming Xiong, and Steven CH Hoi. Prototypical contrastive learning of unsupervised representations. *arXiv preprint arXiv:2005.04966*, 2020.
- [45] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.
- [46] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.
- [47] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey E Hinton. Big self-supervised models are strong semi-supervised learners. *Advances in neural information processing systems*, 33:22243–22255, 2020.
- [48] Philip Bachman, R Devon Hjelm, and William Buchwalter. Learning representations by maximizing mutual information across views. *Advances in neural information processing systems*, 32, 2019.

- [49] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.
- [50] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems*, 33:9912–9924, 2020.
- [51] Tongzhou Wang and Phillip Isola. Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International Conference on Machine Learning*, pages 9929–9939. PMLR, 2020.
- [52] Ting Chen, Calvin Luo, and Lala Li. Intriguing properties of contrastive losses. *Advances in Neural Information Processing Systems*, 34, 2021.
- [53] Feng Wang and Huaping Liu. Understanding the behaviour of contrastive loss. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2495–2504, 2021.
- [54] Sanjeev Arora, Hrishikesh Khandeparkar, Mikhail Khodak, Orestis Plevrakis, and Nikunj Saunshi. A theoretical analysis of contrastive unsupervised representation learning. *arXiv preprint arXiv:1902.09229*, 2019.
- [55] Phuc H Le-Khac, Graham Healy, and Alan F Smeaton. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020.
- [56] Liyao Tang, Yibing Zhan, Zhe Chen, Baosheng Yu, and Dacheng Tao. Contrastive boundary learning for point cloud segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8489–8499, 2022.
- [57] Cheng-Kun Yang, Yung-Yu Chuang, and Yen-Yu Lin. Unsupervised point cloud object co-segmentation by co-contrastive learning and mutual attention sampling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7335–7344, 2021.
- [58] Li Jiang, Shaoshuai Shi, Zhuotao Tian, Xin Lai, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Guided point contrastive learning for semi-supervised point cloud semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6423–6432, 2021.
- [59] Bi'an Du, Xiang Gao, Wei Hu, and Xin Li. Self-contrastive learning with hard negative sampling for self-supervised point cloud learning. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3133–3142, 2021.
- [60] Di Wang, Lulu Tang, Xu Wang, Luqing Luo, and Zhi-Xin Yang. Improving deep learning on point cloud by maximizing mutual information across layers. *Pattern Recognition*, 131:108892, 2022.
- [61] Mohamed Afham, Isuru Dissanayake, Dinithi Dissanayake, Amaya Dharmasiri, Kanchana Thilakarathna, and Ranga Rodrigo. 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*, pages 9902–9912, 2022.
- [62] Huixiang Shao, Zhijiang Zhang, Xiaoyu Feng, and Dan Zeng. Sernet: A spatial consistency guided network using contrastive learning for point cloud registration. *Symmetry*, 14(1):140, 2022.
- [63] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.
- [64] Fangzhou Lin, Yajun Xu, Ziming Zhang, Chenyang Gao, and Kazunori D Yamada. Cosmos propagation network: Deep learning model for point cloud completion. *Neurocomputing*, 507:221–234, 2022.
- [65] Li Yi, Vladimir G Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Transactions on Graphics (ToG)*, 35(6):1–12, 2016.
- [66] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015.
- [67] Liang Pan, Xinyi Chen, Zhongang Cai, Junzhe Zhang, Haiyu Zhao, Shuai Yi, and Ziwei Liu. Variational relational point completion network. In *CVPR*, 2021.

- [68] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015.
- [69] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezafofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR*, 2019.
- [70] Thibault Groueix, Matthew Fisher, Vladimir G Kim, Bryan C Russell, and Mathieu Aubry. A papier-mâché approach to learning 3d surface generation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 216–224, 2018.
- [71] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Jiageng Mao, Shengping Zhang, and Wenxiu Sun. Grnet: Gridding residual network for dense point cloud completion. In *ECCV*, 2020.
- [72] Xuejun Yan, Hongyu Yan, Jingjing Wang, Hang Du, Zhihong Wu, Di Xie, Shiliang Pu, and Li Lu. Fbnet: Feedback network for point cloud completion. In *European Conference on Computer Vision*, pages 676–693. Springer, 2022.
- [73] Xin Wen, Peng Xiang, Zhizhong Han, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Yu-Shen Liu. Pmp-net: Point cloud completion by learning multi-step point moving paths. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7443–7452, 2021.
- [74] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [75] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [76] Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3405–3414, 2019.
- [77] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *CVPR*, 2020.