This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Hyperbolic Chamfer Distance for Point Cloud Completion

Fangzhou Lin[1,2] Yun Yue[1] Songlin Hou[1,3] Xuechu Yu[1] Yajun Xu[4]
Kazunori D Yamada[2] Ziming Zhang[1*]
[1]Worcester Polytechnic Institute, USA [2]Tohoku University, Japan
[3]Dell Technologies, USA [4]Hokkaido University, Japan
{flin2, yyue, xyu4, shou, zzhang15}@wpi.edu, yamada@tohoku.ac.jp,

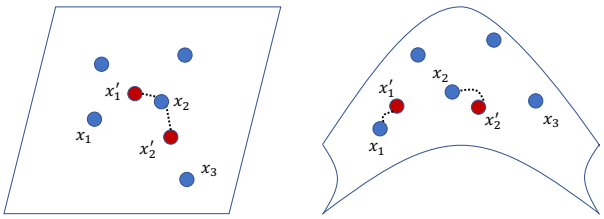
Abstract

Chamfer distance (CD) is a standard metric to measure the shape dissimilarity between point clouds in point cloud completion, as well as a loss function for (deep) learning. However, it is well known that CD is vulnerable to outliers, leading to the drift towards suboptimal models. In contrast to the literature where most works address such issues in Euclidean space, we propose an extremely simple yet powerful metric for point cloud completion, namely Hyperbolic Chamfer Distance (HyperCD)*, that computes CD in hyperbolic space. In backpropagation, HyperCD consistently assigns higher weights to the matched point pairs with smaller Euclidean distances. In this way, good point matches are likely to be preserved while bad matches can be updated gradually, leading to better completion results. We demonstrate state-of-the-art performance on the benchmark datasets,* i.e. *PCN, ShapeNet-55, and ShapeNet-34, and show from visualization that HyperCD can significantly improve the surface smoothness. Code is available at:* https://github.com/Zhang-VISLab.

1. Introduction

Point clouds, one of the most important data representations that can be easily acquired, play a key role in modern robotics and automation applications [54, 33, 44]. However, raw data of point clouds captured by existing 3D sensors is usually incomplete and sparse due to occlusion, limited sensor resolution and light reflection [68, 23, 31, 24, 75], which can negatively impact the performance of downstream tasks that require high-quality representation, such as point cloud segmentation and detection. In this paper, we address this issue by inferring the complete shape of an object or scene from incomplete raw point clouds. This task is referred to as *point cloud completion* [2].

*corresponding author



(a) Euclidean (b) Hyperbolic

Figure 1. Illustration of point matching in the **(a)** Euclidean space and **(b)** hyperbolic space. With the *position-aware* embeddings in hyperbolic space, the mismatched point pairs in Euclidean space may be corrected, leading to better completion performance.

Point cloud completion is usually non-trivial due to the unordered and unstructured characteristics of point clouds (especially obtained from real-world environments). Recently, many (deep) learning-based approaches have been introduced to point cloud completion ranging from supervised learning, self-supervised learning to unsupervised learning [70, 55, 35, 6, 11, 43]. Amongst them, supervised learning with a general encoder-decoder structure serves as the dominant paradigm architectural choice for many researchers and achieves state-of-the-art on nearly all mainstream benchmarks. Their works largely focus on the design of different structures in the encoder and decoder for more informative feature extractions and better point cloud generation [69, 62, 75, 54, 12], in Euclidean space.

Unequal Point Importance in Point Clouds. Humans often perceive the visual quality of point clouds in a non-homogeneous way by putting a higher emphasis on the points with certain geometric structures such as planes, edges, corners, *etc*. For example, point clouds with smooth surfaces and sharp edges tend to be more visually appealing than their counterparts [64, 26]. Surprisingly, this simple yet nontrivial fact in point clouds, however, is hardly explored in the literature of point cloud completion. For in-

stance, Chamfer distance (CD) is a widely used metric in point cloud completion, *e.g*. [15, 61], to measure the shape dissimilarity between any pair of point clouds by calculating the average distance between each point in one set to its nearest neighbor found in another. While CD can faithfully reflect the global dissimilarity between the prediction and ground truth, the distances of all nearest-neighbor pairs between both sets are treated with equal importance (even higher weights to outliers). Thus, CD is sensitive to outliers.

Density-aware Chamfer Distance (DCD) in Euclidean Space. To address such an equal weighting problem in CD, recently Wu *et al*. [61] proposed a DCD metric by exploring the disparity of density distributions in point clouds. As illustrated in Fig. 1 (a), due to the different point density in point clouds, denser points may easily have multiple matches, while sparser points may not. This phenomenon is considered in DCD as a weighting mechanism (inverse to the number of matches for balance) so that sparser points have higher weights. Meanwhile, DCD also proposed using an exponential approximation (the first order approximation of Taylor expansion) of CD to overcome the sensitivity to outliers, as illustrated in Fig. 2.

Though empirically DCD seems to work better than CD for point cloud completion, it may have some serious issues:

- *Density-aware mechanism in DCD may assign higher weights to sparser points.* This not only tends to obtain good matches at the edges and corners, but also favors the matches with outliers that leads to inferior completion.
- *CD approximate functions hardly preserve good matches.* From Fig. 2, CD is sensitive to outliers because its gradient assigns higher (or equal) weights to the points with larger distances. DCD can mitigate this problem, but the weights either decrease too fast (exponentially) with the ℓ_1 distance or are small for good matches (even zeros for perfect matches) with the ℓ_2 distance.

Hyperbolic Chamfer Distance (HyperCD). To mitigate these aforementioned problems in CD for point cloud completion, in contrast to the literature, we believe that preserving good matches while improving bad matches gradually during training is the key to the success in point cloud completion. We call this matching property *position-aware*, as it only depends on the point positions. Recently hyperbolic space has been demonstrated as a means to represent the inherent compositional nature of point clouds using position-aware embeddings within tree-like geometric structures, *e.g*. [36]. Such works highly motivate us to explore CD in hyperbolic space.

As illustrated in Fig. 1 (b), hyperbolic space provides more flexibility than Euclidean distance as a measure between points, and thus it may be possible to correct matching errors in CD. Besides, as illustrated in Fig. 2 (left), with different power functions, the hyperbolic spaces (defined by arcosh) can better approximate CD than DCD. Meanwhile, as we see, the curves of gradients of $y = \operatorname{arcosh}(1 + x)$ and $y = \operatorname{arcosh}(1 + x^3/3)$ are quite similar to those for DCD, while the gradient of $y = \operatorname{arcosh}(1 + x^2/2)$ produces a nice curve that exactly follows what we expect for a good weighting mechanism in point cloud completion. This observation provides us new insights on defining HyperCD.

Specifically, by matching the points with nearest neighbors in Euclidean space (represented by x-axis in Fig. 2), we first obtain the matches between the prediction and ground truth, and vice versa. We then plug these Euclidean distances into arcosh to represent them in hyperbolic space. Empirically we demonstrate that models trained with such a simple metric can significantly outperform the counterparts trained with CD as well as DCD.

Contributions. We list our main contributions as follows:

- We propose an extremely simple yet powerful distance metric, HyperCD, for point cloud completion. To the best of our knowledge, we are the *first* to explore hyperbolic space for point cloud completion.
- We demonstrate state-of-the-art performance on several benchmark datasets based on popular networks that are trained with HyperCD.

2. Related Work

3D Shape Completion. Elder methods in 3D shape completion generally focus on voxel grid, which have network architecture similar to 2D image networks [34, 8, 22]. However, information loss will inevitably happen when the intermediate representations have been involved, and voxelization will cause high computational cost regard to voxel resolution [56]. Therefore, recent state-of-the-art models are designed to consume raw point cloud data directly. As the pioneering work PointNet [42], it independently applies MLPs on each point and subsequently aggregates features through max-pooling operation to achieve permutation invariance. Following this design, a clear-cut way is employing permutation invariance neural networks as a tool to design an encoder for input partial feature extraction and a decoder to complete point clouds. As the first learning-based point cloud completion network, PCN [70] extract global feature in a similar way PointNet did and generate points through folding operations [66]. In order to obtain local structures among points, Zhang *et al*. [74] extract multi-scale features from different layers in the feature extraction part to enhance the performance. CDN [57] uses a cascaded refinement network to bridge the local details of partial input and the global shape information together.

Lyu *et al*. [32] treat point cloud completion as a conditional generation problem in the framework of denoising diffusion probabilistic models (DDPM) [46, 16, 76, 30]. They also mentioned the problem where CD loss is not sensitive to overall density distribution. Their solution is us-

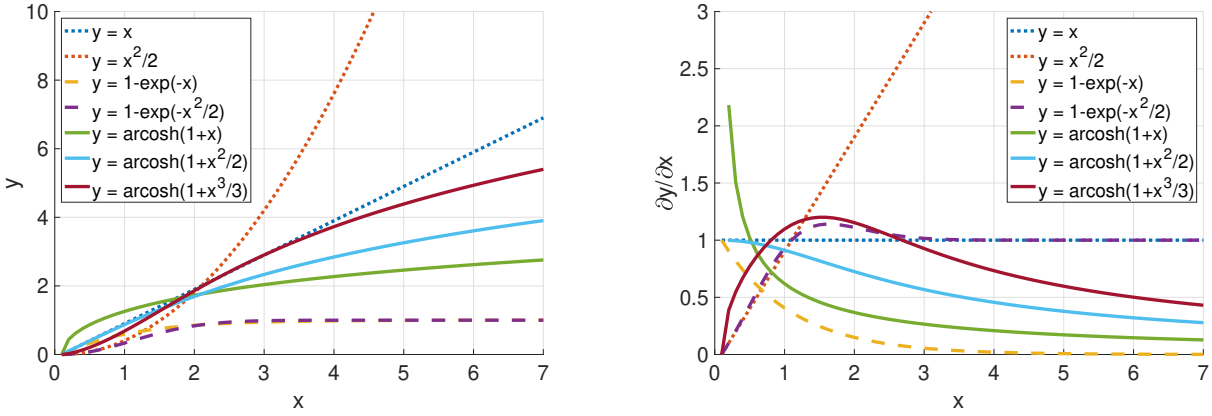


Figure 2. Illustration of **(left)** some distance metrics and **(right)** their corresponding gradients, where the dotted curves are used in ℓ_1 and ℓ_2-CD, the dash ones are used in density-aware Chamfer distance (DCD) [61], and the solid curves are special cases of our HyperCD.

ing DDPM to define a one-to-one point-wise mapping between two consecutive point clouds in the diffusion process, which uses a simple mean squared error loss function [4] for training. However, it is computationally intensive and only works at the coarse point cloud generation stage.

Attention mechanisms like Transformer [53], demonstrate their superiority in long-range interaction capturing as compared to CNNs' constrained receptive fields. For instance, to preserve more detailed geometry information for point cloud generation in the decoder, SA-Net [58] uses the skip-attention mechanism to merge local region information from the encoder and point features of the decoder. SnowflakeNet [62] and PointTr [69] pay extra attention to the decoder part with Transformer-like designs. PointAttN [54] further proposes an architecture design solely based on Transformers. These works have demonstrated the ability of Transformers in point cloud completion tasks.

Point Cloud Distance. 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 [61]. With relatively low computational cost fair design, CD and its variants are extensively used in learning-based methods for point cloud completion tasks [9, 32, 73, 47]. 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 [27, 1]. Recently, Wu *et al*. [61] 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.

Hyperbolic Learning. Euclidean space has been widely used in machine learning because it is a natural generalization of human intuition-friendly, visual three-dimensional space, and easy for measuring distances and inner-products [13, 28, 20, 40]. However, the Euclidean embedding is not the suitable choice for some complex tree-like data fields such as Biology, Network Science, Computer Graphics, or Computer Vision that exhibit highly non-Euclidean latent anatomy [13, 5]. This encourages the research community to develop deep neural networks in non-Euclidean space, such as hyperbolic space, which is a Riemannian manifold of constant negative curvature. Recently, the gap between the hyperbolic embeddings and the Euclidean embeddings has been narrowed by deriving the essential components of deep neural networks in hyperbolic geometry [13, 45] (*e.g*. multinomial logistic regression, fully-connected layers, recurrent neural networks *etc*.).

Unlike Euclidean space with polynomial volume growth *w.r.t.* the radius, hyperbolic space $\mathbb{H}^n$ has exponential growth that is suitable for tree-like structure data. The representation power of hyperbolic space has been demonstrated in NLP [37, 38], image segmentation [60, 3], few-shot [18] and zero-shot learning [29] as well as metric learning equipped with vision transformers [10]. For point clouds of 3D objects, the data naturally exhibit a hierarchy property, where simple parts can be assembled into progressively more complex shapes to form whole objects. Recently, the work of [36] has shown that the features from a point cloud classifier could be embedded into hyperbolic space that leads to the state-of-art supervised models for point cloud classification. Intuitively, points near the boundary of hyperbolic space are sparser compared with points at the center. While the hierarchy property between part and whole could provide useful clues in classifying ob-

jects, it cannot be directly applied in generation tasks like point cloud completion. However, the property of hyperbolic embedding with exponential offers a clue in designing a new loss for point cloud completion tasks that focus more on the surface. To the best of our knowledge, we are the first to introduce hyperbolic space in point cloud completion.

3. Method

In this section, we will introduce HyperCD, its fast calculation and weighting mechanism in backpropagation.

3.1. Chamfer Distance

Notations. We denote (x_i, y_i) as the i-th point cloud pair in the training data, $x_i = \{x_{ij}\}$ as the incomplete input point cloud with 3D points $x_{ij}, \forall j$, and $y_i = \{y_{ik}\}$ as the ground-truth point cloud with points $y_{ik}, \forall k$. We denote $d(\cdot, \cdot)$ as a certain distance metric, f as a neural network for generating a new point cloud from an incomplete input point cloud that is parametrized by ω.

Definition. Based on such notations above, a Chamfer distance for point clouds can be defined as follows, in general:

$$D(x_i, y_i) = \frac{1}{|x_i|}\sum_j \min_k d(x_{ij}, y_{ik}) + \frac{1}{|y_i|}\sum_k \min_j d(x_{ij}, y_{ik}), \quad (1)$$

where $|\cdot|$ denotes the cardinality of a set. Based on this definition, we can instantiate the distance metric with different geometric spaces, such as:

- *Euclidean distance:* For point cloud completion, function d is usually defined in Euclidean space, referring to

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

  where $\|\cdot\|$ denotes the Euclidean ℓ_2 norm of a vector. As we show, such distances make CD sensitive to outliers.
- *Hyperbolic distance:* Hyperbolic space is a homogeneous space with a constant negative curvature whose distance between two points is determined by the curvature that goes through both points. In general, a hyperbolic space can be constructed using five different isometric models, among which Poincaré model is popular in deep learning [41]. Specifically, supposing that two points x_{ij}, y_{ik} lie in the Poincaré unit ball, *i.e.* $\|x_{ij}\| < 1, \|y_{ik}\| < 1$, their hyperbolic distance is defined as

$$d(x_{ij}, y_{ik}) = \operatorname{arcosh}\left(1 + 2\frac{\|x_{ij} - y_{ik}\|^2}{(1 - \|x_{ij}\|^2)(1 - \|y_{ik}\|^2)}\right). \quad (3)$$

  Note that the hyperbolic distance can be always defined based on arcosh, no matter what model is used to represent the hyperbolic space.

Learning Objective for Point Cloud Completion. Based on the definition of CD in Eq. 1, a simple learning objective can be written as follows:

$$\min_{\omega\in\Omega}\sum_i F_i(\omega) \stackrel{def}{=} \min_{\omega\in\Omega}\sum_i D(f(x_i;\omega), y_i), \quad (4)$$

where Ω denotes the feasible solution space for ω defined by some constraints such as regularization.

3.2. Hyperbolic Chamfer Distance

Challenges. There are several challenges that prevent us from directly substituting Eq. 3 into Eq. 1 as listed below:

- *Domain constraint:* The norm of each 3D point should be strictly smaller than 1. Some implementation tricks such as clipping [72] can be applied here to mitigate the issue.
- *Computational burden:* The calculation in Eq. 3 is much more complex than the Euclidean distance, leading to significantly higher computational burden especially in large scale settings as the matching complexity per point cloud pair is $O(|x_i||y_i|)$, *e.g.* the numbers of points in point clouds bigger than 10K. To mitigate this issue, often the hyperbolic distance is computed in Gyrovector space instead [50, 51, 52, 71], a generalization of Euclidean vector spaces, based on the Möbius transformations [21]. However, such operations still require too much computation to be efficient in large scale settings.

So far, we have discovered that (1) computing Euclidean distances is much faster than computing hyperbolic distances, and (2) hyperbolic distances are defined based on arcosh. So, how shall we define hyperbolic Chamfer distance and accelerate its computation?

3.2.1 Definition

Motivated by hyperbolic distance in Eq. 3, we propose a novel distance measure based on Eq. 1, namely *Hyperbolic Chamfer Distance (**HyperCD**)*, where

$$d(x_{ij}, y_{ik}) = \operatorname{arcosh}\left(1 + \alpha\|x_{ij} - y_{ik}\|^2\right), \alpha > 0. \quad (5)$$

Note that the hyperbolic distance in Eq. 3 can be taken as a special case of Eq. 5, *i.e.*, $\alpha = \frac{2}{(1-\|x_{ij}\|^2)(1-\|y_{ik}\|^2)}$ as a function of x_{ij}, y_{ik}. We provide an efficient way to compute HyperCD between point clouds in Alg. 1, whose complexity is similar to Euclidean CD.

3.2.2 Learning with HyperCD as Loss Function

Now let us discuss the learning process in backpropagation for updating network weights. Considering the learning ob-

Algorithm 1 HyperCD

Input : a point cloud pair (x_i, y_i), hyperparameter $\alpha > 0$
Output: HyperCD $D(x_i, y_i)$

Initialize a matrix M, $D_1 \leftarrow 0, D_2 \leftarrow 0$;
foreach j, k **do**
  $M_{jk} \leftarrow \|x_{ij} - y_{ik}\|^2$;
end
foreach j **do**
  $D_1 \leftarrow D_1 + \operatorname{arcosh}(1 + \alpha \min_k M_{jk})$;
end
foreach k **do**
  $D_2 \leftarrow D_2 + \operatorname{arcosh}(1 + \alpha \min_j M_{jk})$;
end
return $D(x_i, y_i) \leftarrow \frac{D_1}{|x_i|} + \frac{D_2}{|y_i|}$;

jective in Eq. 4, then we have

$$\frac{\partial F_i}{\omega} = \frac{1}{|\tilde{x}_i|} \sum_j \frac{\partial d(\tilde{x}_{ij}, y_{im(j)})}{\partial \omega} + \frac{1}{|y_i|} \sum_k \frac{\partial d(\tilde{x}_{in(k)}, y_{ik})}{\partial \omega} \tag{6}$$

where $\tilde{x}_i$ denotes the output point cloud from the network f by taking x_i as input, and $m(j), n(k)$ denote the corresponding indexes for the nearest neighbor matches. Further,

$$\frac{\partial d(\tilde{x}_{ij}, y_{im(j)})}{\partial \omega} = z_{ij} \cdot \frac{\partial \|\tilde{x}_{ij} - y_{im(j)}\|}{\partial \omega} \tag{7}$$

where $z_{ij} = \frac{2\alpha \|\tilde{x}_{ij} - y_{im(j)}\|}{\sqrt{\left(1+\alpha \|\tilde{x}_{ij} - y_{im(j)}\|^2\right)^2 - 1}} \in \mathbb{R}$ denotes the weight for the gradient feature in backpropagation. This *implicit* weighting mechanism only depends on the Euclidean distances, and thus it is position-aware. The same gradient feature, $\frac{\partial \|\tilde{x}_{ij} - y_{im(j)}\|}{\partial \omega}$, is also used in CD and DCD. The only difference among such distance metrics in learning is the weighting mechanism.

3.3. Analysis on HyperCD

Proposition 1. *Consider* $d(x_{ij}, y_{ik}) = g(\|x_{ij} - y_{ik}\|)$ *in Eq. 1 where function* g *is* strictly *increasing. It holds that*

$$\min_k d(x_{ij}, y_{ik}) = g\left(\min_k \|x_{ij} - y_{ik}\|\right). \tag{8}$$

This proposition states that g and $\min$ are switchable for a strictly increasing function g. In this way, the complexity of $g\left(\min_k \|x_{ij} - y_{ik}\|\right)$ is just slightly higher than computing Euclidean distances (with extra calculation for g).

Proposition 2. *Consider a function* $h(x) = \operatorname{arcosh}(1 + \alpha x^\beta), \forall x \geq 0$. h *is strictly increasing iff* $\alpha > 0, \beta > 0$.

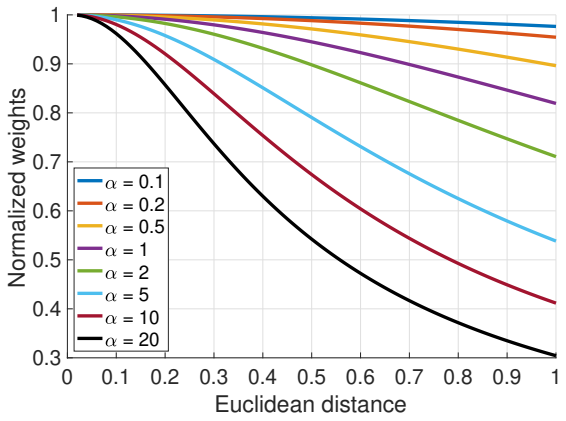


Figure 3. Illustration of the gradient weights using our HyperCD. All the numbers are normalized by $\frac{1}{\sqrt{2\alpha}}$.

This proposition can be easily proven by the fact that the derivative of h is always positive iff $\alpha > 0, \beta > 0$.

Proposition 3. *Considering function* h *in Prop. 2, then its derivative,* $\frac{\partial h}{\partial x} = \frac{\alpha\beta x^{\beta-1}}{\sqrt{(1+\alpha x^\beta)^2 - 1}}$, *satisfies that*

$$\lim_{x \to 0^+} \frac{\partial h(x)}{\partial x} = \lim_{x \to 0^+} \frac{\alpha\beta}{\sqrt{2\alpha}} x^{\frac{\beta}{2}-1} = \begin{cases} +\infty, & 0 < \beta < 2 \\ \frac{\alpha\beta}{\sqrt{2\alpha}}, & \beta = 2 \\ 0, & \beta > 2 \end{cases} \tag{9}$$

$$\lim_{x \to +\infty} \frac{\partial h(x)}{\partial x} = \frac{\beta}{x}. \tag{10}$$

When x is approaching 0, function h will perform as a power function, with a single special case where $\beta = 2$ returns a constant. This analysis coincides with Fig. 2 (right), and it seems that only $\beta = 2$ can provide a reasonable weighting mechanism to preserve good point matches. When x is approaching infinity, all the curves with the same β will converge to a power function as well.

Proposition 4. *The weight* z_{ij} *in Eq. 7 is strictly decreasing* w.r.t. $\|\tilde{x}_{ij} - y_{im(j)}\|$ *for an arbitrary* $\alpha > 0$.

Fig. 3 illustrates the change of weights *w.r.t.* the distances using different α's. When α is small, *e.g.* $\alpha \leq 2$, the curves decrease gradually, which can potentially better preserve good matches through backpropagation.

4. Experiments

Datasets. We verify and analyze our HyperCD for point cloud completion on the following benchmark datasets.

- *ShapeNet-Part:* The benchmark ShapeNet-Part [67] is a part segmentation subset of ShapeNetCore [7] 3D meshes. It contains 17,775 different 3D mesh which belong to 16 categories. The ground truth point cloud data

Table 1. Completion results on PCN in terms of per-point L1 Chamfer distance $\times 1000$ (lower is better).

Methods	Average	Plane	Cabinet	Car	Chair	Lamp	Couch	Table	Boat
FoldingNet [66]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
TopNet [49]	12.15	7.61	13.31	10.90	13.82	14.44	14.78	11.22	11.12
AtlasNet [14]	10.85	6.37	11.94	10.10	12.06	12.37	12.99	10.33	10.61
GRNet [63]	8.83	6.45	10.37	9.45	9.41	7.96	10.51	8.44	8.04
CRN [57]	8.51	4.79	9.97	8.31	9.49	8.94	10.69	7.81	8.05
NSFA [74]	8.06	4.76	10.18	8.63	8.53	7.03	10.53	7.35	7.48
FBNet [65]	6.94	3.99	9.05	7.90	7.38	5.82	8.85	6.35	6.18
PCN [70]	11.27	**5.50**	22.70	10.63	**8.70**	**11.00**	**11.34**	11.68	**8.59**
HyperCD + PCN	**10.59**	5.95	**11.62**	**9.33**	12.45	12.58	13.10	**9.82**	9.85
FoldingNet [66]	14.31	9.49	15.80	12.61	15.55	16.41	15.97	13.65	14.99
HyperCD + FoldingNet	**12.09**	**7.89**	**12.90**	**10.67**	**14.55**	**13.87**	**14.09**	**11.86**	**10.89**
PMP-Net [59]	8.73	5.65	11.24	9.64	9.51	6.95	10.83	8.72	7.25
HyperCD + PMP-Net	**8.40**	**5.06**	**10.67**	**9.30**	**9.11**	**6.83**	**11.01**	**8.18**	**7.03**
PoinTr [69]	8.38	4.75	10.47	8.68	9.39	7.75	10.93	7.78	7.29
HyperCD + PoinTr	**7.56**	**4.42**	**9.77**	**8.22**	**8.22**	**6.62**	**9.62**	**6.97**	**6.67**
SnowflakeNet [62]	7.21	4.29	9.16	8.08	7.89	6.07	9.23	6.55	6.40
HyperCD + SnowflakeNet	**6.91**	**3.95**	**9.01**	**7.88**	**7.37**	**5.75**	**8.94**	**6.19**	**6.17**
PointAttN [54]	6.86	3.87	9.00	7.63	7.43	5.90	8.68	6.32	6.09
DCD + PointAttN	7.54	4.47	9.65	8.14	8.12	6.75	9.60	6.92	6.67
HyperCD + PointAttN	**6.68**	**3.76**	**8.93**	**7.49**	**7.06**	**5.61**	**8.48**	**6.25**	**5.92**
SeedFormer [75]	6.74	3.85	9.05	8.06	7.06	5.21	8.85	6.05	5.85
DCD + SeedFormer	24.52	16.42	26.23	21.08	20.06	18.30	26.51	18.23	18.22
HyperCD + SeedFormer	**6.54**	**3.72**	**8.71**	**7.79**	**6.83**	**5.11**	**8.61**	**5.82**	**5.76**

Table 2. Completion results on ShapeNet-55 based on L2 Chamfer distance $\times 1000$ (lower is better) and F-Score@1% (higher is better).

Methods	Table	Chair	Plane	Car	Sofa	CD-S	CD-M	CD-H	CD-Avg	F1
PFNet [17]	3.95	4.24	1.81	2.53	3.34	3.83	3.87	7.97	5.22	0.339
FoldingNet [66]	2.53	2.81	1.43	1.98	2.48	2.67	2.66	4.05	3.12	0.082
TopNet [49]	2.21	2.53	1.14	2.18	2.36	2.26	2.16	4.3	2.91	0.126
PCN [70]	2.13	2.29	1.02	1.85	2.06	1.94	1.96	4.08	2.66	0.133
GRNet [63]	1.63	1.88	1.02	1.64	1.72	1.35	1.71	2.85	1.97	0.238
PoinTr [69]	0.81	0.95	0.44	0.91	0.79	0.58	0.88	1.79	1.09	0.464
SeedFormer [75]	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**

was created by sampling 2,048 points uniformly on each mesh. The partial point cloud data were generated by randomly selecting a viewpoint as a center among multiple viewpoints and removing points within a certain radius from the complete data, which is similar to the generation of PoinTr's ShapeNet-55/34 [69] benchmark. The number of points we remove from each point cloud is 512.

- *PCN:* One of the most popular benchmark datasets for point cloud completion is the PCN dataset [70]. It is a subset of ShapeNet [7] 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 [70, 75].
- *ShapeNet-55/34:* ShapeNet-55 and ShapeNet-34 datasets datasets are also generated from the synthetic ShapeNet [7] dataset while they contain more object categories and incomplete patterns. All 55 categories in ShapeNet are included in ShapeNet-55 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,

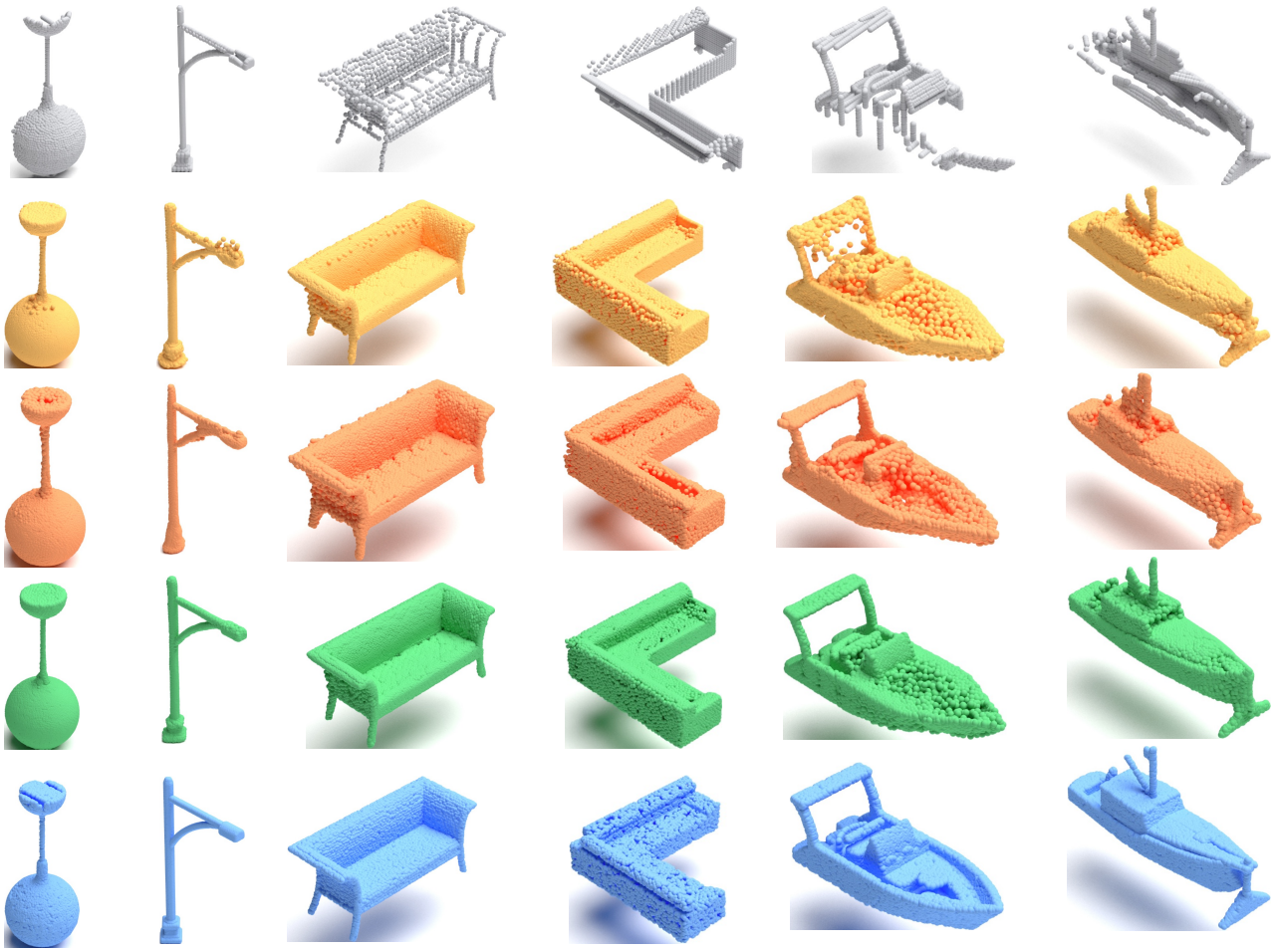


Figure 4. Visual comparison of point cloud completion results on PCN. **Row-1:** Inputs of incomplete point clouds. **Row-2:** Outputs of PointAttN with CD. **Row-3:** Outputs of PointAttN with DCD. **Row-4:** Outputs of PointAttN with HyperCD. **Row-5:** Ground truth.

Table 3. Completion results on ShapeNet-34 based on L2 Chamfer distance $\times 1000$ (lower is better) and F-Score@1% (higher is better).

Methods	34 seen categories					21 unseen categories				
	CD-S	CD-M	CD-H	CD-Avg	F1	CD-S	CD-M	CD-H	CD-Avg	F1
PFNet [17]	3.16	3.19	7.71	4.68	0.347	5.29	5.87	13.33	8.16	0.322
FoldingNet [66]	1.86	1.81	3.38	2.35	0.139	2.76	2.74	5.36	3.62	0.095
TopNet [49]	1.77	1.61	3.54	2.31	0.171	2.62	2.43	5.44	3.50	0.121
PCN [70]	1.87	1.81	2.97	2.22	0.154	3.17	3.08	5.29	3.85	0.101
GRNet [63]	1.26	1.39	2.57	1.74	0.251	1.85	2.25	4.87	2.99	0.216
PoinTr [69]	0.76	1.05	1.88	1.23	0.421	1.04	1.67	3.44	2.05	0.384
SeedFormer [75]	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**

2,048 points are sampled as input and 8,192 points as ground truth. Following the same evaluation strategy with [69], 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 the complete point cloud) which corresponds to three difficulty levels of *simple*, *moderate* and *hard* in the test stage.

Implementation. We take three state-of-the-art networks, *i.e*. CP-Net [25], PointAttN [54] and SeedFormer [75], as our backbone networks for comparison and analysis. We train all these networks with or without HyperCD from scratch using PyTorch [39] with the Adam optimizer [19].

All three networks are designed solely using the dissimilarity between generated and ground truth point clouds as supervision signal. For the lightweight network CP-Net, its original dissimilarity metric between prediction and ground truth is only calculated at the first stage. We replace its dissimilarity metric at this stage using HyperCD in our experiment. For multi-stage networks, *i.e*. PointAttN and SeedFormer, the loss functions are introduced at every stage from coarse to fine point clouds. We replace the loss functions from all the stages with HyperCD so it can participate in the whole training process. To ensure fairness in comparison, all the other loss functions compared with HyperCD in this paper are processed in the same way as HyperCD.

All the hyperparameters such as learning rate, batch size and training epochs are kept consistently with the setting of baselines for a fair comparison. Hyperparameter α in HyperCD is tuned with grid search, by default. We conduct our experiments on a server with 10 NVIDIA RTX 2080Ti 11G GPUs for CP-Net with ShapeNet-Part, on a server with 4 NVIDIA A100 80G GPUs for PointAttN with PCN, on a server with 4 NVIDIA V100 16G GPUs for SeedFormer with PCN and ShapeNet-55/34.

Evaluation Metrics. To make a fair comparison, we evaluate the performance of all the methods using CD. F1-Score@1% [48] is also used to evaluate ShapeNet-55/34 with the same experiment setting in the literature. For better comparison we also list the original results of some other methods on PCN and ShapeNet-55/34.

4.1. State-of-the-art Comparison

PCN. Following the literature, we report CD with L1-distance in Table 1 with numbers per category. We also include the results trained with DCD. As we can see, the replacement of HyperCD loss enables two baselines to outperform their previous state-of-the-art results by a certain amount, while the performance gets slightly worse when DCD is used. As we discussed earlier, numerical metric (*i.e*. CD) may not faithfully reflect the visual quality, so we also provide qualitative evaluation results shown in Fig. 4, compared with results generated from the baseline model with CD and DCD loss functions. As we can see, both models can reconstruct point clouds in general outline to some extent, but the reconstructed results with CD are more likely to suffer from distortion on several areas with high noise level on the surface. With the introduction of HyperCD loss during training, the baseline network can further demonstrably well-reconstructed point cloud in general outline while maintaining the realistic details of the original ground truth with significantly reduced noise level. Although DCD exhibits better capacity in controlling noise level in generated point clouds, it fails to preserve fine details and also suffers from distortion.

Table 4. Completion results of CP-Net with different losses on ShapeNet-Part in terms of per-point L2 Chamfer distance $\times 1000$.

Loss function	CD-Avg
L1-CD	4.16
L2-CD	4.82
DCD	5.74
$y = \text{arcosh}(1 + x)$	4.43
$y = \text{arcosh}(1 + x^3)$	4.22
Hyperbolic Distance	4.09
HyperCD	**4.03**

ShapeNet-55/34. We also test on the ShapeNet-55 dataset to evaluate the adaptability of HyperCD on tasks with higher diversities. Table 2 summarizes the average L2 Chamfer distances on three difficulty levels and the overall CDs. Following the convention, we show results in 5 categories (Table, Chair, Plane, Car and Sofa) with more than 2,500 samples in the training set. Complete results for all 55 categories are available in the supplemental material. We also provide results under the F-Score@1% metric. As we can see from Table 2, the introduction of HyperCD improves the baseline performance to some extent. To have an intuitive evaluation of reconstructed results, we also provide qualitative evaluations in supplemental materials compared with results generated from the baselines. We can clearly see while reconstructed results have better numerical performance, the model trained with HyperCD works better in reconstructing the surface areas and preserving the details with less noise. The improvement in both numerical and qualitative evaluations indicates that HyperCD is capable to adapt to point completing tasks with high diversities.

On ShapeNet-34, we evaluate performances within 34 seen categories (same as training) as well as 21 unseen categories (not used in training). In Table 3, we can observe that HyperCD is capable of improving baseline model performance in terms of achieving higher scores. The improvement of performance indicates our loss function is highly-generalizable for point clouds completion tasks with both seen and unseen categories.

4.2. Analysis

We choose ShapeNet-Part as the dataset to analyze and compare with different loss functions. As introduced previously, ShapeNet-Part is a relatively small dataset comprising 16 categories objects, which is sufficient for analysis in our case. For the model part, we choose a light-weighted network called CP-Net [25].

Hyperparameters. We provide the relationship between α with learning rate (lr) on the effect of training performance in Fig. 5. As a complimentary to our aforementioned discuss on arcosh, we train and test the performance of differ-

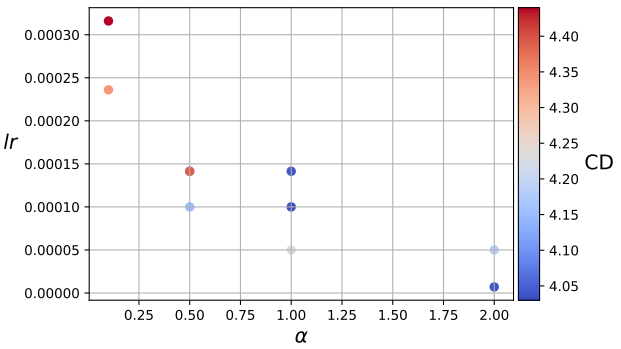


Figure 5. The L1-CD with different α and lr.

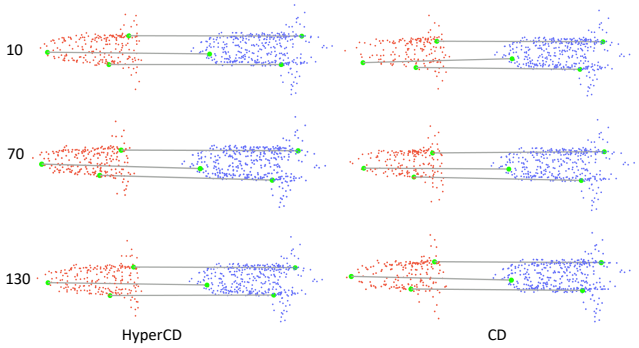


Figure 6. Illustration of point correspondence change over epochs.

ent arcosh-family metrics (as shown in Fig. 2) on CP-Net, and we reported their results in Table 4. We also provide the performance of several popular loss functions in Table 4, in which HyperCD outperforms both CD and DCD by a large margin.

Computation. Note that our HyperCD has only one more operation, arcosh, than CD and thus in theory both computational efficiency should be very similar, but both are much simpler than hyperbolic distance. Numerically, it takes 0.4239±0.0019, 0.4298±0.0014 and 0.5335±0.0368 second per iteration for training CP-Net with CD, HyperCD and hyperbolic distance, respectively.

Point Correspondences. We also provide an visualization on how the point correspondences change during training. We plot some point correspondences over epochs (10,70,130), as shown in Fig. 6 where the blue points are ground truth and the red ones are predictions. HyperCD is able to help stabilize the (correct) correspondences much faster in training, leading to better convergence than CD. Also, as visualized in Fig. 6 at the 10-th epoch, the predicted airplane head (red) is much smoother and geometrically closer to the ground truth (blue) based on HyperCD than CD, demonstrating that HyperCD can preserve geometric features much better.

5. Conclusion

In this work, we explore the possibility of introducing the hyperbolic space into point cloud completion tasks. Inspired by the exponential growth nature of hyperbolic space, we extend the CD loss using the hyperbolic space. Motivated by hyperbolic distance and the recent literature on CD, we propose a novel distance measure, namely Hyperbolic Chamfer Distance (HyperCD), to mitigate the well-known outlier issues in CD. To demonstrate the performance of HyperCD, we evaluate it on several benchmark datasets with popular networks and achieve new state-of-the-art performance for point cloud completion.

Acknowledgements

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. Dr. Ziming Zhang, Yun Yue and Xuechu Yu 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). Computations were partially performed on the NIG supercomputer at ROIS National Institute of Genetics.

References

[1] 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. 3

[2] 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. 1

[3] Mina Ghadimi Atigh, Julian Schoep, Erman Acar, Nanne van Noord, and Pascal Mettes. Hyperbolic image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4453–4462, 2022. 3

[4] Peter J Bickel and Kjell A Doksum. *Mathematical statistics: basic ideas and selected topics, volumes I-II package*. Chapman and Hall/CRC, 2015. 3

[5] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017. 3

[6] 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. 1

[7] 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. 5, 6

[8] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5868–5877, 2017. 2

[9] 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. 3

[10] Aleksandr Ermolov, Leyla Mirvakhabova, Valentin Khrulkov, Nicu Sebe, and Ivan Oseledets. Hyperbolic vision transformers: Combining improvements in metric learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7409–7419, 2022. 3

[11] 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. 1

[12] 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. 1

[13] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic neural networks. *Advances in neural information processing systems*, 31, 2018. 3

[14] 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. 6

[15] 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. 2

[16] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[17] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *CVPR*, 2020. 6, 7

[18] Valentin Khrulkov, Leyla Mirvakhabova, Evgeniya Ustinova, Ivan Oseledets, and Victor Lempitsky. Hyperbolic image embeddings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6418–6428, 2020. 3

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[20] Anna Klimovskaia, David Lopez-Paz, Léon Bottou, and Maximilian Nickel. Poincaré maps for analyzing complex hierarchies in single-cell data. *Nature communications*, 11(1):2966, 2020. 3

[21] Max Kochurov, Rasul Karimov, and Serge Kozlukov. Geoopt: Riemannian optimization in pytorch, 2020. 4

[22] Truc Le and Ye Duan. Pointgrid: A deep network for 3d shape understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9204–9214, 2018. 2

[23] 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. 1

[24] 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. 1

[25] 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. 7, 8

[26] Kangcheng Liu. An integrated lidar-slam system for complex environment with noisy point clouds. *arXiv preprint arXiv:2212.05705*, 2022. 1

[27] 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. 3

[28] Qi Liu, Maximilian Nickel, and Douwe Kiela. Hyperbolic graph neural networks. *Advances in neural information processing systems*, 32, 2019. 3

[29] Shaoteng Liu, Jingjing Chen, Liangming Pan, Chong-Wah Ngo, Tat-Seng Chua, and Yu-Gang Jiang. Hyperbolic visual embedding learning for zero-shot recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9273–9281, 2020. 3

[30] Shitong Luo and Wei Hu. Diffusion probabilistic models for 3d point cloud generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2837–2845, 2021. 2

[31] 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. 1

[32] 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. 2, 3

[33] 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. 1

[34] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 922–928. IEEE, 2015. 2

[35] Himangi Mittal, Brian Okorn, Arpit Jangid, and David Held. Self-supervised point cloud completion via inpainting. *arXiv preprint arXiv:2111.10701*, 2021. 1

[36] Antonio Montanaro, Diego Valsesia, and Enrico Magli. Rethinking the compositionality of point clouds through regularization in the hyperbolic space. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. 2, 3

[37] Maximillian Nickel and Douwe Kiela. Poincaré embeddings for learning hierarchical representations. *Advances in neural information processing systems*, 30, 2017. 3

[38] Maximillian Nickel and Douwe Kiela. Learning continuous hierarchies in the lorentz model of hyperbolic geometry. In *International Conference on Machine Learning*, pages 3779–3788. PMLR, 2018. 3

[39] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7

[40] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *arXiv preprint arXiv:2101.04562*, 2021. 3

[41] Wei Peng, Tuomas Varanka, Abdelrahman Mostafa, Henglin Shi, and Guoying Zhao. Hyperbolic deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(12):10023–10044, 2022. 4

[42] 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. 2

[43] 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. 1

[44] 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. 1

[45] Ryohei Shimizu, Yusuke Mukuta, and Tatsuya Harada. Hyperbolic neural networks++. *arXiv preprint arXiv:2006.08210*, 2020. 3

[46] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[47] 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. 3

[48] 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. 8

[49] Lyne P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, Ian Reid, and Silvio Savarese. Topnet: Structural point cloud decoder. In *CVPR*, 2019. 6, 7

[50] Abraham A Ungar. Hyperbolic trigonometry and its application in the poincaré ball model of hyperbolic geometry. *Computers & Mathematics with Applications*, 41(1-2):135–147, 2001. 4

[51] Abraham Albert Ungar. *Analytic hyperbolic geometry and Albert Einstein's special theory of relativity*. World Scientific, 2008. 4

[52] Abraham Albert Ungar. A gyrovector space approach to hyperbolic geometry. *Synthesis Lectures on Mathematics and Statistics*, 1(1):1–194, 2008. 4

[53] 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. 3

[54] 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. 1, 3, 6, 7

[55] Xiaogang Wang, Marcelo H. Ang Jr. , and Gim Hee Lee. Cascaded refinement network for point cloud completion. In *CVPR*, 2020. 1

[56] Xiaogang Wang, Marcelo H Ang, and Gim Hee Lee. Voxel-based network for shape completion by leveraging edge generation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 13189–13198, 2021. 2

[57] 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. 2, 6

[58] Xin Wen, Tianyang Li, Zhizhong Han, and Yu-Shen Liu. Point cloud completion by skip-attention network with hierarchical folding. In *CVPR*, 2020. 3

[59] 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. 6

[60] Zhenzhen Weng, Mehmet Giray Ogut, Shai Limonchik, and Serena Yeung. Unsupervised discovery of the long-tail in instance segmentation using hierarchical self-supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2603–2612, 2021. 3

[61] 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. 2, 3

[62] Peng Xiang, Xin Wen, Yu-Shen Liu, Yan-Pei Cao, Pengfei Wan, Wen Zheng, and Zhizhong Han. Snowflakenet: Point cloud completion by snowflake point deconvolution with skip-transformer. In *ICCV*, 2021. 1, 3, 6

[63] 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. 6, 7

[64] Yajun Xu, Shogo Arai, Diyi Liu, Fangzhou Lin, and Kazuhiro Kosuge. Fpcc: Fast point cloud clustering-based instance segmentation for industrial bin-picking. *Neurocomputing*, 494:255–268, 2022. 1

[65] 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. 6

[66] 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. 2, 6, 7

[67] 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. 5

[68] 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. 1

[69] 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. 1, 3, 6, 7

[70] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: point completion network. In *3DV*, 2018. 1, 2, 6, 7

[71] Yun Yue, Fangzhou Lin, Kazunori D Yamada, and Ziming Zhang. Hyperbolic contrastive learning. *arXiv preprint arXiv:2302.01409*, 2023. 4

[72] Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. *arXiv preprint arXiv:1905.11881*, 2019. 4

[73] 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. 3

[74] 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. 2, 6

[75] 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. 1, 6, 7

[76] Linqi Zhou, Yilun Du, and Jiajun Wu. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5826–5835, 2021. 2