Pattern Recognition Letters 178 (2024) 43–48

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Learnable Chamfer Distance for point cloud reconstruction

Tianxin Huang [a,b], Qingyao Liu [b], Xiangrui Zhao [b], Jun Chen [b], Yong Liu [b,*]

[a] National University of Singapore, Singapore
[b] Zhejiang University, Hangzhou, China

ARTICLE INFO

Editor: Antonio Fernández-Caballero

Keywords:
3D point cloud processing
Reconstruction loss
Adversarial strategy

ABSTRACT

As point clouds are 3D signals with permutation invariance, most existing works train their reconstruction networks by measuring shape differences with the average point-to-point distance between point clouds matched with predefined rules. However, the static matching rules may deviate from actual shape differences. Although some works propose dynamically-updated learnable structures to replace matching rules, they need more iterations to converge well. In this work, we propose a simple but effective reconstruction loss, named Learnable Chamfer Distance (LCD) by dynamically paying attention to matching distances with different weight distributions controlled with a group of learnable networks. By training with adversarial strategy, LCD learns to search defects in reconstructed results and overcomes the weaknesses of static matching rules, while the performances at low iterations can also be guaranteed by the basic matching algorithm. Experiments on multiple reconstruction networks confirm that LCD can help achieve better reconstruction performances and extract more representative representations with faster convergence and comparable training efficiency.

1. Introduction

Point cloud is one signal describing the 3D shape, which is widely-used due to its convenient acquisition from 3D sensors such as RGB-D camera or LiDAR. Different from regular 1-D signals or 2-D images, point clouds are permutation-invariant, which means changing specific permutations of points does not change described shapes. In other words, the permutations of points do not include any useful information. In this condition, commonly-used mean squared errors (MSE) cannot be directly applied to point cloud reconstruction. To train a point cloud reconstruction network, most existing works use the Chamfer Distance (CD) or Earth Mover's Distance (EMD) [1] as training losses. They match points with predefined rules and measure shape differences between input point clouds and reconstructed results by average point-to-point distance. However, the losses based on manually-defined matching rules are *static*, which means the optimization goals are fixed and unchanged for all data during training. They may deviate the actual shape differences and make the reconstruction fall into local minimums with inferior reconstructed results but low reconstruction losses. Although some works [2–4] introduce GAN discriminators [5] to improve the reconstruction performance, they simply add the discriminator constraints to CD or EMD. Their improvements are limited as the discriminators only provide slight corrections to unchanged CD or EMD as shown in [6]. PCLoss [6] replaces the matching-based losses with distances between comparison matrices extracted with *dynamic*-updated learnable structures, which totally avoids the adoption of static matching rules and learns to use changing measurements to measure the shape differences. It learns to search the shape defects by adversarial process, which has better performances due to the removal of predefined rules. But the totally learnable structures perform relatively inferior at the beginning of training process because it needs iterations to learn to find the defects.

Considering the problems mentioned above, we propose a simple but effective learnable point cloud reconstruction loss, named Learnable Chamfer Distance (LCD) by designing a reasonable combination of *dynamic* learning-based strategy and *static* matching-based loss evaluation. The differences between LCD and existing methods are presented in Fig. 1. Unlike the totally learning-based design in PCLoss [6], LCD learns to predict weight distributions for matching distances of different points. During training, LCD is optimized by turns with the reconstruction network through an adversarial strategy to search regions with more shape defects, where the weight distributions are dynamically adjusted to pay more attention to matching distances of different regions. Benefited from the adoption of dynamic learning-based strategy, LCD can achieve outstanding performances for the training of reconstruction networks, while the static matching-based evaluation can provide an initialization prior for the optimization and ensure that LCD has better

* Corresponding author.
E-mail addresses: 21725129@zju.edu.cn (T. Huang), qingyaoliu@zju.edu.cn (Q. Liu), xiangruizhao@zju.edu.cn (X. Zhao), junc@zju.edu.cn (J. Chen), yongliu@iipc.zju.edu.cn (Y. Liu).

https://doi.org/10.1016/j.patrec.2023.12.015
Received 27 February 2023; Received in revised form 7 July 2023; Accepted 22 December 2023
Available online 29 December 2023

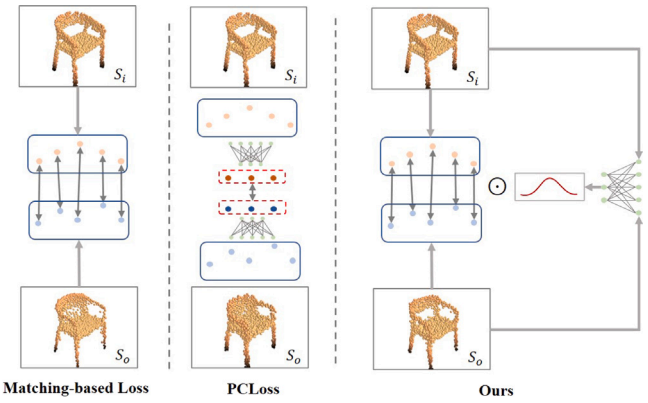

Fig. 1. Differences between our method and existing reconstruction losses. Unlike the totally learning-based loss evaluation by extracting comparison matrices in PCLoss, our method uses networks to dynamically predict weight distributions for matching distances.

performances than totally learning-based PCLoss [6] at the beginning of training process.

Our contributions can be summarized as

- We propose Learnable Chamfer Distance (LCD), which can learn to search shape defects by dynamically predicting weight distributions for matching distances;
- Benefited from the reasonable combination of learning-based strategy and matching-based evaluation, LCD has faster convergence than existing learning-based losses;
- Experiments on multiple point clouds reconstruction networks demonstrate that LCD can help the reconstruction networks achieve better reconstruction performances and extract more representative representations.

2. Related works

2.1. Point cloud reconstruction

Point cloud reconstruction aims to design networks, e.g. auto-encoders, to reconstruct point clouds through the representation extracted from the input point clouds. It can be adopted to related tasks like completing [3,8–11] or sampling [12–14] point clouds, while the extracted intermediate representations can be used for the unsupervised classification [15–17]. Following PCLoss [6], the basic point cloud reconstruction network is often organized with encoders to extract representations from point clouds, and decoders to generate point clouds from the intermediate representations. The commonly-used encoders include PointNet [7], PointNet++ [18], and DGCNN [19], while decoders often come from fully connected networks proposed in AE [16] and FoldingNet proposed in [15]. In this work, we follow PCLoss [6] to construct multiple reconstruction networks to evaluate performances of different losses.

2.2. Reconstruction loss design

Most existing point cloud reconstruction-related tasks rely on the Chamfer Distance (CD) [8] and Earth Mover's Distance (EMD) [1], which evaluate the reconstruction losses based on the average point-to-point distance between matched input and reconstructed point clouds. However, the predefined matching rules are static, which may cause the training processes fall into local minimums due to the deviation of predefined rules. In this condition, many researchers attempt to introduce learning-based strategy to improve the constraining performances. Most researchers [2–4] design GAN discriminators [5] for extra supervisions. However, these works simply add the discriminator constraints to basic CD or EMD losses. Their optimizations still mainly reply on the matching-based CD or EMD, where the discriminators can only provide slight corrections. Therefore, these methods often have limited improvements. Although DCD [20] fixes the matching rules in CD by considering the density distribution of reconstructed points, it is still limited by the static evaluation for reconstruction losses.

PCLoss [6] replaces the usage of static matching rules with a dynamic learning process. It learns to extract comparison matrices from point clouds with differentiable structures and measure shape differences with distances between comparison matrices. But the totally learning-based training process in PCLoss makes it need more iterations to converge well, while the relatively complex structures bring low training efficiency. In this work, we explore to organically combine the learning-based strategy with matching rules by learning to pay attention to different matching connections. Benefited from the learning-based strategy, our method has better performances than matching-based methods, while the adoption of static rules ensures it has faster convergence than totally learning-based methods like PCLoss.

3. Methodology

In this work, we propose a new method named Learnable Chamfer Distance (LCD) to evaluate the reconstruction loss by measuring the average point-to-point distance weighted with dynamically updated distributions. In this work, we use the static matching rules in CD [8] to calculate the matching distances due to its high efficiency. The structure of LCD is presented in Section 3.1. The training process of the reconstruction network with LCD is presented in Section 3.2.

3.1. The structure of Learnable Chamfer Distance

We propose a series of learnable structures to dynamically predict the weight distributions for the matching distances of different points. As shown in Fig. 2, the weight distributions W_i and W_o are predicted with Siamese Concatenation block (**SiaCon**) and Siamese Attention block (**SiaAtt**). **SiaAtt** predicts weight distributions for the matching distances, while **SiaCon** extracts global shape representations from both input and reconstructed point clouds and injects them to **SiaAtt**. Specifically, in **SiaCon** block, two parameter-shared $f_1(\cdot)$ are used to extract global features from input point cloud S_i and reconstructed result S_o. They are concatenated to construct a overall perception for the shapes S_i and S_o. In **SiaAtt** block, two global features extracted by f_2 include independent shape information of S_i and S_o, respectively. This information is fused with overall perception of two models to predict a weight for each coordinates with MLP in $g(\cdot)$.

Let S_i and S_o be the input point clouds and reconstructed results, respectively. CD loss can be defined as

$$L_{CD}(S_i, S_o) = \frac{1}{2}\Big(\frac{1}{|S_i|}\sum_{x\in S_i}\min_{y\in S_o}\|x-y\|_2 + \frac{1}{|S_o|}\sum_{x\in S_o}\min_{y\in S_i}\|x-y\|_2\Big). \tag{1}$$

We can see that CD measures the reconstruction loss through the average distance between points in S_i or S_o and their nearest neighbors in another point set.

Let $f_1(\cdot)$, $f_2(\cdot)$ be the combination of parameter-shared Multi Layer Perceptrons (MLPs) and symmetric pooling operations like PointNet [7], $Con(\cdot)$ be the concatenation of features, $g(\cdot)$ be a group of MLPs, **SiaCon** can be defined as

$$F_{io} = Con(f_1(S_i), f_1(S_o)). \tag{2}$$

In **SiaAtt**, we have

$$\begin{cases} F_i = g(Con(S_i, f_2(S_i), F_{io})), \\ F_o = g(Con(S_o, f_2(S_o), F_{io})). \end{cases} \tag{3}$$

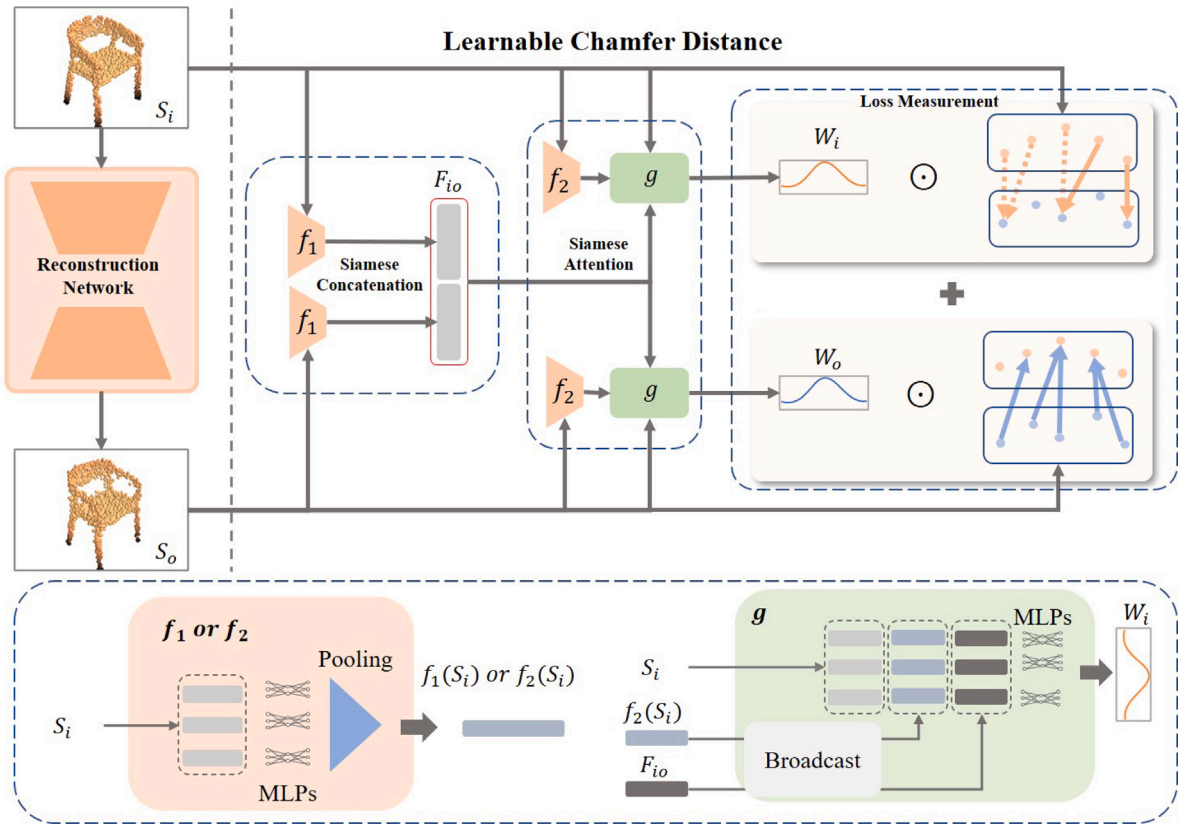

Fig. 2. The pipeline of Learnable Chamfer Distance. f_1 and f_2 are constructed following PointNet [7] framework to extract global features, while g merges multiple features to predict weight distributions W_i and W_o for matching distances. The input point cloud S_i and reconstructed point cloud S_o are introduced to Siamese Concatenation module to extract global shape representation F_{io}, which is injected into Siamese Concatenation module to predict weight distributions W_i and W_o for matching distances between S_i and S_o.

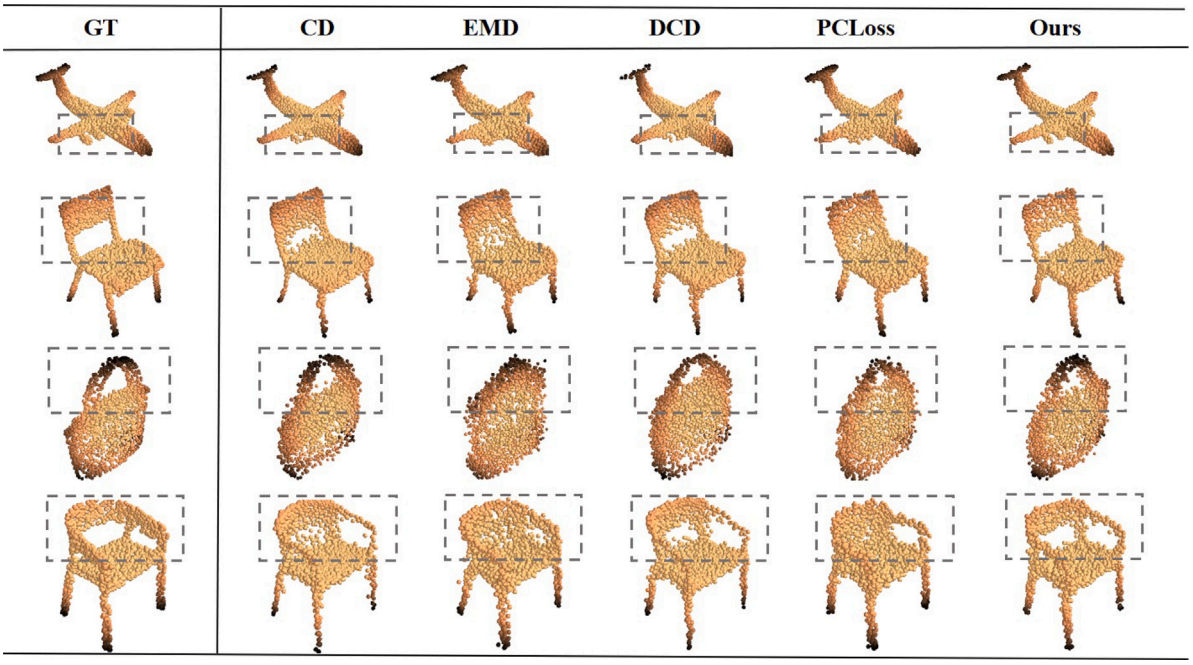

Fig. 3. Qualitative comparison between LCD and other methods based on AE [16] following PCLoss [6]. Our method can help create clearer details as shown in the circled regions.

Algorithm 1 Training Process with LCD.

Input: Input S_i, reconstructed results S_o, the number of iterations $iter$, the reconstruction network $RecNet(\cdot)$
for $n = 1$ **to** $iter$ **do**
Calculate output of the reconstruction network: $S_o^n = RecNet(S_i^n)$.
Let θ_L and θ_R be the parameters of LCD and the reconstruction network, respectively.
Fix θ_R and optimize θ_L by descending gradient: $\nabla_{\theta_L} L_{LCD}(S_o^n, S_i^n)$.
Fix θ_L and optimize θ_R by descending gradient: $\nabla_{\theta_R} L_R(S_o^n, S_i^n)$.
end for

The weight distributions can then be defined as

$$\begin{cases} W_i = \dfrac{\sigma + e^{-F_i^2}}{|F_i| \cdot \sigma + \sum e^{-F_i^2}}, \\ W_o = \dfrac{\sigma + e^{-F_o^2}}{|F_o| \cdot \sigma + \sum e^{-F_o^2}}, \end{cases} \tag{4}$$

where the boundary coefficient σ is a small constant used to adjust the weight distribution. Intuitively speaking, in Eq. (4), F_i is firstly scaled to 0∼1 with $e^{-F_i^2}$, where the scaled results will be normalized into weight distributions satisfying $\sum W_i = 1$ and $\sum W_o = 1$. σ is used to soft the weight distributions and prevent each weight from being too small to optimize. The final loss measurement can be defined as

$$\begin{aligned} L_R(S_i, S_o) = \frac{1}{2}\Big(&\frac{1}{|S_i|}\sum_{x \in S_i} W_i \cdot \min_{y \in S_o} \|x - y\|_2 \\ &+ \frac{1}{|S_o|}\sum_{x \in S_o} W_o \cdot \min_{y \in S_i} \|x - y\|_2\Big). \end{aligned} \tag{5}$$

Note that our method estimates the weight for each point in a same point cloud/sample, which is quite different with the boosting-related re-weighting methods to predict weights for various point clouds/samples in the dataset.

3.2. Training pipeline

LCD is trained with adversarial strategy to consistently search for existing shape differences between reconstructed results and input point clouds.

The whole training process with LCD is a generative-adversarial process similar as GAN [5], which updates the parameters of LCD and the reconstruction network by turns. In each iteration, LCD is optimized by L_{LCD} to explore more shape differences, where the reconstruction network is then optimized with L_R to eliminate the searched differences. Let L_R be the reconstruction loss defined in Section 3.1. We define the adversarial loss to optimize LCD as $L_{LCD} = -log(L_R + \sigma_r)$, where σ_r is a tiny value to avoid errors when $L_R \to 0$.

4. Experiments

4.1. Dataset and implementation dtails

Training details. ShapeNet part dataset [16] is composed of 12288/1870/2874 models in the train/val/test splits. For the reconstruction task, we train the networks on the train split of ShapeNet part dataset, while evaluating on its test split. For the unsupervised classification, we still train networks on the train split of ShapeNet part dataset and use ModelNet10 and ModelNet40 containing 10 and 40 categories of CAD models to evaluate the classification accuracy following FoldingNet [15]. Each model consists of 2048 points randomly sampled from the surfaces of mesh models. In this work, learning rates of reconstruction networks and LCD are set as 0.0001 and 0.002, while σ and σ_r are set as 0.01 and 1e−8. The matching-based evaluation of CD [8] is introduced to calculate the matching distances in LCD.

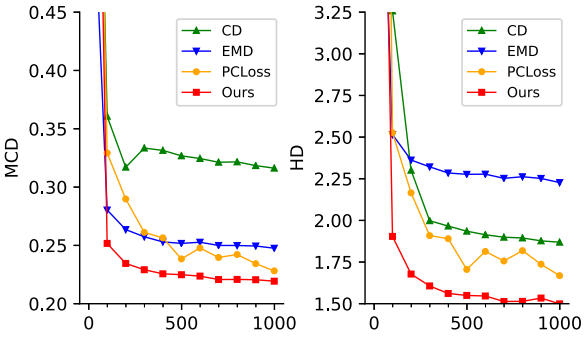

Fig. 4. The error visualization during the whole training process.

Reconstruction Networks. To compare LCD with existing reconstruction losses, we conduct comparisons based on multiple reconstruction networks. AE [16] and FoldingNet [15] are two classic and commonly used point cloud reconstruction networks, which have been used in many works [4,8,10,21]. In this work, we follow PCLoss [6] to construct 6 reconstruction networks with three commonly-used encoders PointNet [7], PointNet++ [18] and DGCNN [19] and 2 basic decoders AE [16] and FoldingNet [15]. The reconstruction performances of whole structures and unsupervised classification accuracy of intermediate representations are adopted to evaluate the performances of different training losses.

4.2. Comparison with existing reconstruction loss

To confirm the performance of our method, we follow PCLoss [6] for the comparison settings. The reconstruction errors and performances of representation learning are adopted for evaluation. The qualitative and quantitative results are presented in Fig. 3 and Table 1, respectively. Multi-scale Chamfer Distance (MCD) proposed by [6,22] and Hausdorff distance (HD) from [6,23] are used as metrics in this work. We can see that our method achieves lowest reconstruction errors on multiple reconstruction networks. As shown in the circled regions of Fig. 3, our method can help the reconstruction network create clearer details such as the wings of airplane and the back of chairs, which confirms its effectiveness.

The reconstruction networks can also be used to extract intermediate representations from point clouds for classification. We also conduct a comparison on unsupervised classification following AE [16], FoldingNet [15], and PCLoss [6]. In details, the reconstruction networks are trained with different losses on ShapeNet [24] and adopted to extract representations from point clouds in ModelNet10 and ModelNet40 [25]. The extracted representations will be used to train Supported Vector Machines (SVMs) with corresponding labels, where the classification accuracy can then reflect the distinguishability of representations.

As shown in Table 2, our method has higher classification accuracy than existing methods in most phenomena, which means LCD can help the reconstruction networks learn more representative representations.

4.3. Training process analysis

To analyze the training process when optimizing the reconstruction networks with LCD. We visualize and compare the reconstruction errors during the iterations between our method and a few representative training losses including CD [8], EMD [1], PCLoss [6] based on AE [16]. The results are presented in Fig. 4. We can see that LCD has

Table 1
Comparison with reconstruction losses. Bold marks the best results.

Networks	AE		Folding		AE (PN++)		Folding (PN++)		AE (DGCNN)		Folding (DGCNN)	
Metrics	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD
CD [1]	0.32	1.87	0.40	4.13	0.37	2.50	0.34	3.37	0.30	1.88	0.52	3.84
EMD [1]	0.25	2.23	–	–	0.26	2.51	–	–	0.21	2.09	–	–
DCD [20]	0.28	1.75	0.91	8.41	0.28	1.84	0.47	5.43	0.26	1.86	–	–
PCLoss [6]	0.23	1.66	0.33	2.57	0.24	1.87	0.31	2.50	0.20	1.51	0.43	3.10
Ours	**0.22**	**1.51**	**0.31**	**2.47**	**0.24**	**1.66**	**0.28**	**2.37**	**0.20**	**1.48**	**0.34**	**2.45**

Table 2
Comparison on unsupervised classification.

RecNet	Dataset	Methods			
		CD	EMD	PCLoss	Ours
AE	MN10	90.60	89.49	91.48	**91.48**
	MN40	85.92	85.47	86.36	**86.60**
Folding	MN10	91.03	–	**91.70**	91.04
	MN40	85.22	–	85.35	**86.81**
AE (PN++)	MN10	90.38	90.15	92.04	**92.70**
	MN40	88.03	88.07	87.54	**88.39**
Folding (PN++)	MN10	91.48	–	91.48	**92.59**
	MN40	87.01	–	86.73	**87.87**
AE (DGCNN)	MN10	91.37	91.26	92.37	**92.81**
	MN40	87.50	87.54	**88.11**	87.54
Folding (DGCNN)	MN10	91.26	–	91.81	**92.37**
	MN40	86.85	–	**87.50**	86.65

Table 3
Training efficiency comparison conducted on an NVIDIA 2080ti with a 2.9 GHz i5-9400 CPU.

Methods	Non-learning			Learning-based	
	CD	EMD	DCD	PCLoss	Ours
Time (ms)	**23**	216	23	57	**43**

Table 4
Ablation for components.

CD	SiaAtt	SiaCon	$log\|\|\cdot\|\|$	MCD	HD
✓				0.32	1.87
✓	✓			0.22	1.98
✓	✓	✓		0.22	1.54
✓	✓	✓	✓	**0.22**	**1.51**

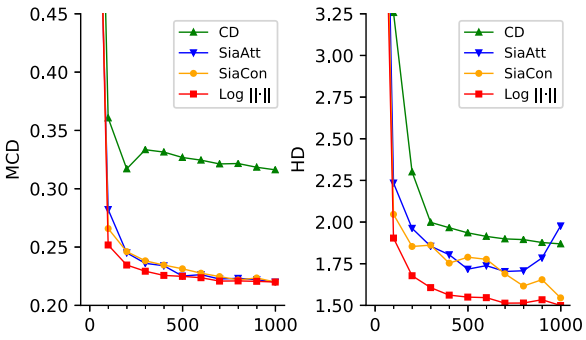

Fig. 5. Ablation study for components over iterations.

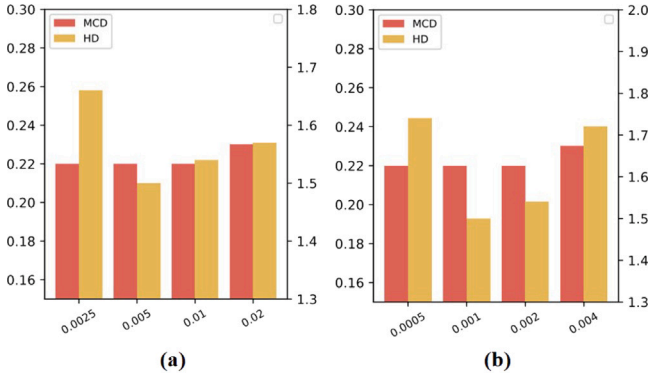

Fig. 6. Influences of σ (a) and LCD learning rate (b).

much faster and steadier convergence than existing methods. Beside, it performs much better than totally learning-based PCLoss at 0~200 iterations, which confirms that the introducing of static matching-based evaluation can ensure the performances at the beginning of training process.

4.4. Comparison on training efficiency

In this section, we compare the time cost consumed by a single iteration between different methods. The results are presented in Table 3. Although LCD is slower than CD and DCD, it has better performances as shown in Table 1. We can see that LCD has higher efficiency than the totally learning-based reconstruction loss PCLoss due to its more concise designation, which can further confirm its effectiveness.

4.5. Ablation study

Ablation study for the components. In this section, we explore the effect of proposed components by removing them and retraining the networks. **SiaAtt** and **SiaCon** denote the Siamese Attention block and Siamese Concatenation block, respectively. $log\|\cdot\|$ means the $log\|\cdot\|$ operation mentioned in Section 3.2 to dynamically adjust the optimization of LCD. The results are presented in Table 4. We can see that removing any component would reduce the final performance. To compare the effect of these components more intuitively, we also visualize the reconstruction errors over all the iterations in Fig. 5. We can see that with adding the **SiaAtt**, **SiaCon**, and $log\|\cdot\|$ gradually makes the errors reduce fast and stably over the whole iterations. An interesting condition is that **SiaAtt** reduces MCD while slightly increasing the HD metric at the end of iterations. It may come from the lack of perception for the overall input and output shapes, making it difficult to find the regions with larger reconstruction errors and mislead the training of reconstruction networks. This condition is then addressed by injecting overall shape features with **SiaCon**.

Influence of the boundary coefficient σ. The boundary coefficient σ defined in Section 3.1 may affect the weight distribution for matching distances. Here, we present experiments to explore its influence as shown in Fig. 6-a. We can see that larger or smaller σ both have negative influences on the results. According to Eq. (4), too small σ makes the distribution steep and hard to train, while too big σ may cause the distribution over-smoothing and limit its performance.

Influence of the LCD learning rate. The LCD learning rate decides the convergence and has influence on final performance. We conduct a group of experiments to observe the influence of the LCD learning rate. The results are presented in Fig. 6-b. We can see that too small or large learning rates both reduce performances. Small learning rates

may limit the ability of LCD to search shape differences, while larger learning rates may lead to its unsteady convergence.

5. Conclusion

In this work, we propose a simple but effective point cloud reconstruction loss, named Learnable Chamfer Distance (LCD), by combining the dynamic learning-based strategy and static matching-based evaluation in a more reasonable way. LCD dynamically predicts weight distributions for matching distances of different points, which is optimized with adversarial strategy to search and pay more attention to regions with larger shape defects. Benefited from the reasonable combination of matching-based evaluation and learning-based strategy, LCD has both faster convergence and higher training efficiency than totally learning-based PCLoss. According to the experiments on multiple reconstruction networks, LCD can help the reconstruction networks achieve better reconstruction performances and extract more representative representations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We thank all reviewers and the editor for excellent contributions. This work is supported by the Key Scientific and Technological Innovation Project of Hangzhou under Grant 2022AIZD0019.

References

[1] H. Fan, H. Su, L.J. Guibas, A point set generation network for 3d object reconstruction from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 605–613.
[2] Z. Huang, Y. Yu, J. Xu, F. Ni, X. Le, PF-net: Point fractal network for 3D point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7662–7670.
[3] X. Wang, M.H. Ang Jr., G.H. Lee, Cascaded refinement network for point cloud completion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 790–799.
[4] R. Li, X. Li, C.-W. Fu, D. Cohen-Or, P.-A. Heng, Pu-gan: a point cloud upsampling adversarial network, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 7203–7212.
[5] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, K. Weinberger (Eds.), Advances in Neural Information Processing Systems, Vol. 27, Curran Associates, Inc., 2014.
[6] T. Huang, X. Yang, J. Zhang, J. Cui, H. Zou, J. Chen, X. Zhao, Y. Liu, Learning to train a point cloud reconstruction network without matching, in: European Conference on Computer Vision, Springer, 2022, pp. 179–194.
[7] C.R. Qi, H. Su, K. Mo, L.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, 2017, pp. 652–660.
[8] W. Yuan, T. Khot, D. Held, C. Mertz, M. Hebert, Pcn: Point completion network, in: 2018 International Conference on 3D Vision (3DV), IEEE, 2018, pp. 728–737.
[9] D. Wang, L. Tang, L. Zhu, Z.-X. Yang, Mutual information maximization based similarity operation for 3D point cloud completion network, IEEE Signal Process. Lett. 29 (2022) 1217–1221.
[10] T. Huang, H. Zou, J. Cui, X. Yang, M. Wang, X. Zhao, J. Zhang, Y. Yuan, Y. Xu, Y. Liu, Rfnet: Recurrent forward network for dense point cloud completion, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 12508–12517.
[11] M. Liu, L. Sheng, S. Yang, J. Shao, S.-M. Hu, Morphing and sampling network for dense point cloud completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 11596–11603.
[12] I. Lang, A. Manor, S. Avidan, Samplenet: Differentiable point cloud sampling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 7578–7588.
[13] O. Dovrat, I. Lang, S. Avidan, Learning to sample, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2760–2769.
[14] L. Zhu, W. Chen, X. Lin, L. He, Y. Guan, Curvature-variation-inspired sampling for point cloud classification and segmentation, IEEE Signal Process. Lett. 29 (2022) 1868–1872.
[15] Y. Yang, C. Feng, Y. Shen, D. Tian, Foldingnet: Point cloud auto-encoder via deep grid deformation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 206–215.
[16] P. Achlioptas, O. Diamanti, I. Mitliagkas, L. Guibas, Learning representations and generative models for 3d point clouds, in: International Conference on Machine Learning, PMLR, 2018, pp. 40–49.
[17] T. Huang, J. Zhang, J. Chen, Y. Liu, Y. Liu, Resolution-free point cloud sampling network with data distillation, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II, Springer, 2022, pp. 54–70.
[18] C.R. Qi, L. Yi, H. Su, L.J. Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in Neural Information Processing Systems, 2017, pp. 5099–5108.
[19] Y. Wang, Y. Sun, Z. Liu, S.E. Sarma, M.M. Bronstein, J.M. Solomon, Dynamic graph cnn for learning on point clouds, ACM Trans. Graph. (TOG) 38 (5) (2019) 1–12.
[20] T. Wu, L. Pan, J. Zhang, T. Wang, Z. Liu, D. Lin, Density-aware chamfer distance as a comprehensive metric for point cloud completion, 2021, arXiv preprint arXiv:2111.12702.
[21] Y. Rao, J. Lu, J. Zhou, Global-local bidirectional reasoning for unsupervised representation learning of 3D point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 5376–5385.
[22] T. Huang, Y. Liu, 3D point cloud geometry compression on deep learning, in: Proceedings of the 27th ACM International Conference on Multimedia, 2019, pp. 890–898.
[23] C.-H. Wu, C.-F. Hsu, T.-C. Kuo, C. Griwodz, M. Riegler, G. Morin, C.-H. Hsu, PCC arena: a benchmark platform for point cloud compression algorithms, in: Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems, 2020, pp. 1–6.
[24] L. Yi, V.G. Kim, D. Ceylan, I.-C. Shen, M. Yan, H. Su, C. Lu, Q. Huang, A. Sheffer, L. Guibas, A scalable active framework for region annotation in 3d shape collections, ACM Trans. Graph. (TOG) 35 (6) (2016) 1–12.
[25] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, J. Xiao, 3D shapenets: A deep representation for volumetric shapes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1912–1920.