2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) | 979-8-3503-0129-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/CVPR52729.2023.01175

Learning to Measure the Point Cloud Reconstruction Loss in a Representation Space

Tianxin Huang[1] Zhonggan Ding[2] Jiangning Zhang[2] Ying Tai[2] Zhenyu Zhang[2]
Mingang Chen[3] Chengjie Wang[2] Yong Liu[1*]
[1]APRIL Lab, Zhejiang University [2]Tencent YouTu Lab
[3]Shanghai Development Center of Computer Software Technology

Abstract

For point cloud reconstruction-related tasks, the reconstruction losses to evaluate the shape differences between reconstructed results and the ground truths are typically used to train the task networks. Most existing works measure the training loss with point-to-point distance, which may introduce extra defects as predefined matching rules may deviate from the real shape differences. Although some learning-based works have been proposed to overcome the weaknesses of manually-defined rules, they still measure the shape differences in 3D Euclidean space, which may limit their ability to capture defects in reconstructed shapes. In this work, we propose a learning-based Contrastive Adversarial Loss (CALoss) to measure the point cloud reconstruction loss dynamically in a non-linear representation space by combining the contrastive constraint with the adversarial strategy. Specifically, we use the contrastive constraint to help CALoss learn a representation space with shape similarity, while we introduce the adversarial strategy to help CALoss mine differences between reconstructed results and ground truths. According to experiments on reconstruction-related tasks, CALoss can help task networks improve reconstruction performances and learn more representative representations.

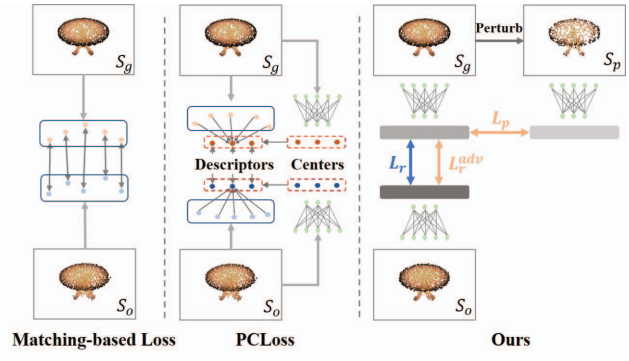

Figure 1. S_g and S_o denote ground truths and point clouds generated by the task network. S_p is a positive sample with similar shapes as S_g acquired by perturbation [2]. Matching-based losses measure distances between points matched by different predefined rules. PCLoss [6] learns to extract descriptors in 3D Euclidean space by linearly weighting coordinates according to their distances to predicted center points, while our method dynamically measures the shape differences with distances between learned global representations in the constructed representation space. L_p and L_r denote representation distances between S_g, S_p and S_g, S_o, respectively. L_r^{adv} is an adversarial loss to maximize representation distances between S_g, S_o. L_r^{adv} and L_p are used to optimize CALoss, while L_r is adopted to train the task network.

1. Introduction

Point clouds, as the common description for 3D shapes, have been broadly used in many areas such as 3D detection [17, 18] and surface reconstruction [9, 13, 19]. For the point cloud reconstruction-related tasks [5, 7, 11, 16], networks need to predict point clouds as similar as possible to the ground truths. Reconstruction losses that can differentiably calculate the shape differences between reconstructed results and ground truths are required to train the task networks. Existing works often use the matching-based reconstruction losses Chamfer Distance (CD) and Earth Mover's Distance (EMD) to constrain shape differences. CD matches points firstly with their nearest neighbors in another point cloud and then calculates the shape difference as average point-to-point distance, while EMD calculates the average point-to-point distance under an optimization-based global matching. We can find that CD and EMD actually measure the distances between *matched points* instead of the distances between *shapes*. As the predefined matching rules are static and unlearnable, training results of CD and EMD may fall into inappropriate local minimums where the reconstruction losses are small but the shapes are obviously different. Learning-based losses in

* means the corresponding author

979-8-3503-0129-8/23/$31.00 ©2023 IEEE
DOI 10.1109/CVPR52729.2023.01175

PFNet [8], PUGAN [10], and CRN [21] introduce extra supervision from discriminators trained with the adversarial strategy to find the detailed differences. Their reconstruction performances are improved by introducing adversarial losses, while they still need CD/EMD to evaluate the basic shape distances and cannot fully get rid of the influence from predefined matching rules. PCLoss [6] presents a reconstruction loss measured by the distances between extracted intermediate descriptors in 3D Euclidean space without any manually-defined matching rule, which is updated together with the task network in a generative-adversarial process to search shape differences during training. However, the descriptor extraction in Euclidean space actually limits the searching of shape differences, while the training efficiency is also restricted as the descriptors are constructed with dense connections between multiple predicted center points and all points. In summary, existing reconstruction losses mainly rely on distances in *3D Euclidean space* to measure the shape differences.

In this work, we propose a novel framework named Contrastive Adversarial Loss (CALoss) learning to measure the point cloud reconstruction loss dynamically in a *high dimensional representation space* constructed by a series of fully differentiable structures. The differences between our work and existing works are presented in Fig. 1. CALoss is composed of L_p, L_r, and L_r^{adv} acquired from distances between global representations.L_r^{adv} and L_p are used to optimize CALoss, where L_r is used to train the task network.

We introduce L_p as the contrastive constraint to help CALoss construct a representation space with the *shape similarity* that similar shapes should have close representations. In this way, by adding adversarial loss on representations, L_r^{adv} can guide CALoss to search for the shape differences between ground truths S_g and reconstructed results S_o. By updating dynamically according to the reconstructed results in each iteration, CALoss can continuously find existing defects in reconstructed shapes and prevent the task network from falling into unexpected local minimums. As the measurement for shape differences is implemented in non-linear representation space, CALoss has more extensive searching space. Besides, the representations adopted to measure shape differences are aggregated with the global pooling operation without any requirement of dense connections as PCLoss [6], which can improve the training efficiency.

Our contribution in this work can be summarized as

- We propose a novel Contrastive Adversarial Loss (CALoss) learning to measure the point cloud reconstruction loss with distances between high-dimensional global representations.
- By combining the contrastive constraint and adversarial training strategy, CALoss can construct a representation space where similar shapes have close representations and learn to search for shape differences in this space dynamically during training.
- Experiments on point cloud reconstruction, unsupervised classification, and point cloud completion confirm that CALoss can help the task network improve reconstruction performances and learn more representative representations with higher training efficiency.

2. Related Work

2.1. Point Cloud Reconstruction-related Tasks

Base on the basic point cloud reconstruction framework to predict similar output point clouds as inputs, e.g. auto-encoder, many related tasks have been developed such as the unsupervised classification and point cloud completion. The unsupervised classification task raised by [1, 25] trains auto-encoders to learn representations of point clouds. The representations are then adopted to train a Support Vector Machine (SVM) with provided labels for further classification. The classification accuracy of SVM can reflect the distinctiveness of learned representations. Many researchers have improved the classification performances by modifying the network structures [12, 22, 28], while some researchers [16] introduce extra supervision to enhance the learning effection. Point cloud completion predicts completed point clouds as identical as possible to the ground truths from partial input point clouds. Early works [11, 27] often use typical auto-encoders to abstract long global features from partial inputs and predict completed results, while recent work [7, 21] add more diverse network structures to improve the completion performances. Reconstruction losses CD or EMD to capture the shape differences are always adopted in these works. In this condition, we adopt three tasks including basic point cloud reconstruction, point cloud unsupervised classification, and completion to evaluate the performances of CALoss.

2.2. Losses to Evaluate Shape Differences

The Chamfer Distance (CD) [27] and Earth Mover's Distance (EMD) [3] are two basic and broadly used reconstruction losses to constrain the shape differences, which calculate the distance between point clouds with different matching strategies. With the matching by nearest neighbors, CD concentrates on differences between contours, while it often constructs non-uniform surfaces as discussed in [3, 23]. EMD aims to find an one-to-one optimal mapping ϕ from one point set to another by optimizing the minimum matching distances between the point sets. Though EMD can create more uniform shapes, it takes large time cost due to the optimized matching and can only be applied to reconstructed output with the same number of input, where the quality of optimization also limits the performances. Since the development of GAN [4], researchers

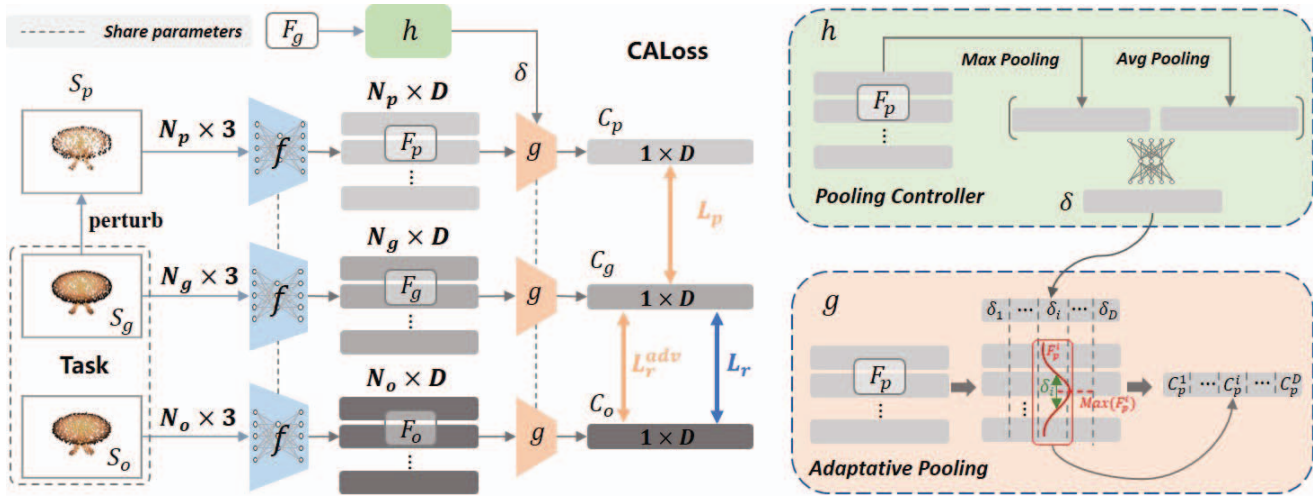

Figure 2. The pipeline of CALoss. S_g and S_o denote ground truths and point clouds generated by the task network. S_p is acquired from S_g with small perturbations. $f(\cdot)$ is *a group of 1D-convolutions* to transform point clouds S_g, S_p, and S_o with N_g, N_p and N_o points into D-dim features F_g, F_p, and F_o. $g(\cdot)$ denotes our proposed Adaptive Pooling to aggregate F_g, F_p, and F_o into global representations C_g, C_p, and C_o with multiple weight distributions, where $h(\cdot)$ is Pooling Controller predicting parameters δ to control the widths of weight distributions in $g(\cdot)$ according to F_g. C_g, C_p, and C_o will be used to calculate the required losses L_p, L_r^{adv}, and L_r to train CALoss and the task network. We introduce adversarial loss to dynamically search for the shape defects in S_o, while maximizing representation distances in a mini-batch like [2] may not work because it lacks dynamic feedback from S_o and cannot capture detailed shape differences.

have introduced different learning-based discriminators in reconstruction losses as extra supervisions to better capture the shape differences and improve reconstruction performances [8, 10, 21]. However, these works still need CD or EMD as basic shape constraints. Some works [14, 23] further modify the matching rules to improve the constraining performances. All these works measure shape differences based on the point-to-point distance calculated with predefined matching rules. DPDist [20] estimates the shape distances with a pre-trained network without matching. But it is mainly designed for registration and has inferior performances on point cloud reconstruction. PCLoss [6] presents a fully learning-based reconstruction loss to overcome the limitation of predefined matching rules, which learns to aggregate descriptors from point clouds and measure reconstruction losses with distances between extracted descriptors. It learns to search shape differences by an adversarial training together with the task network.

However, PCLoss constructs descriptors in 3D Euclidean space with linear weights calculated by distances between multiple predicted center points and all existing points, which may limit both its searching space for shape differences and training efficiency. In this work, we propose CALoss to measure the reconstruction losses in a high dimensional representation space. By the concise designation with only non-linear transformation and the global pooling operation, CALoss can achieve both better performances and higher training efficiency than existing learning-based reconstruction losses.

3. Methodology

The pipeline of CALoss is presented as Fig. 2. The reconstructed result S_o and ground truth S_g from the task is fed into CALoss to evaluate the shape differences. A positive sample S_p is constructed by small perturbations. S_g, S_p, and S_o are transformed into features F_g, F_p and F_o by 1-D convolutions $f(\cdot)$. Features F_g, F_p, and F_o are finally aggregated into global representations C_g, C_p, and C_o with Adaptive Pooling $g(\cdot)$ to calculate losses L_p, L_r^{adv}, and L_r. L_p and L_r can be calculated by:

$$L_p = \|C_g, C_p\|_1, L_r = \|C_g, C_o\|_1, \tag{1}$$

where the adversarial loss L_r^{adv} is defined as:

$$L_r^{adv} = -log(L_r + \sigma_r). \tag{2}$$

σ_r is a tiny value to avoid errors when $L_r \to 0$. These losses are used to optimize CALoss and the task network together.

3.1. Perturbation Operation

In this work, we perturb ground truths S_g with tiny Gaussian noises, which can be defined as

$$S_p = S_g + \mathcal{N}_\sigma, \tag{3}$$

where $\mathcal{N}_\sigma = \text{randn}(\sigma)$. The noise width is controlled by a constant σ. This operation creates perturbed point clouds with similar but different shapes as ground truths.

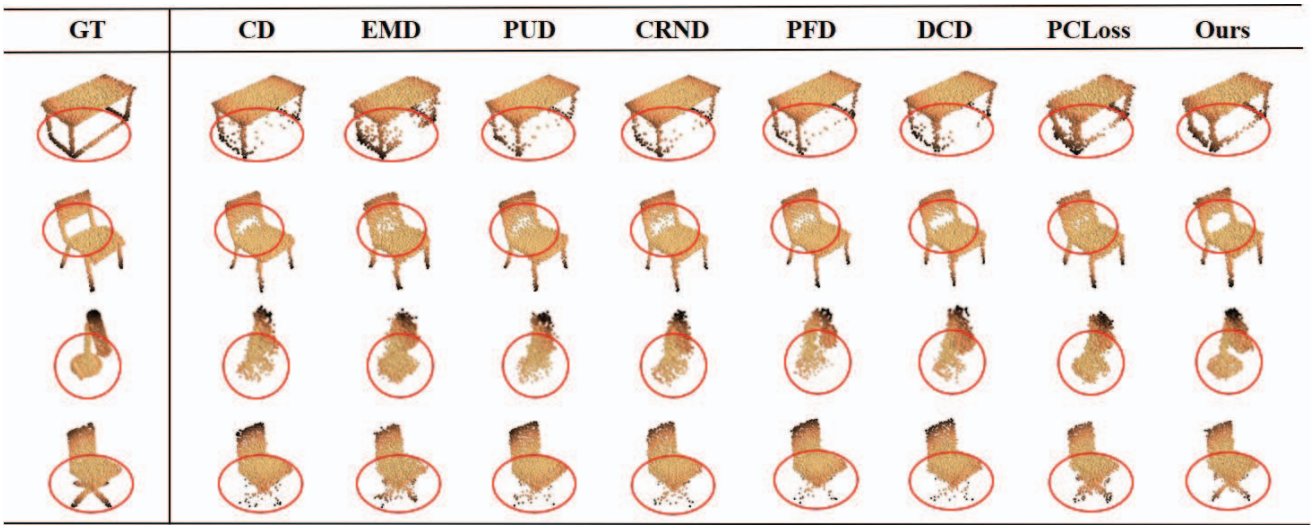

Figure 3. The qualitative comparisons with different losses based on AE [1]. Our method can help the task network create more complete shapes and clearer details.

3.2. Adaptive Pooling and Pooling Controller

Adaptive Pooling is an important operation to aggregate features based on all points into a global representation. Unlike max pooling or average pooling, Adaptive Pooling is dynamically changed and controlled by Pooling Controller during the training process. The structure of Pooling Controller includes a simple network structure to predict parameters δ for Adaptive Pooling as shown in Fig. 2. If we defined $Con(\cdot)$ as concatenation, $maxpool(\cdot)$, $avgpool(\cdot)$ and $MLP(\cdot)$ as max pooling, average pooling and Multi Layer Perceptrons (MLPs), the Pooling Controller can be described as:

$$\delta = h(F_g) = MLP(Con(maxpool(F_g), avgpool(F_g))). \tag{4}$$

It takes both max pooled and average pooled features to acquire more extensive information about F_g. In this condition, let us take F_g as an example, then the global representation C_g can be defined as:

$$C_g = g(F_g, \delta) = \sum_{i=1}^{|F_g|} \frac{e^{-\|F_g^i - maxpool(F_g)\|/\delta}}{\sum_{i=1}^{|F_g|} e^{-\|F_g^i - maxpool(F_g)\|/\delta}} \cdot F_g. \tag{5}$$

C_p and C_o can be acquired by the same equations:

$$C_p = g(F_p, \delta), C_o = g(F_o, \delta). \tag{6}$$

We can see that the features are aggregated into global representations with multiple weight distributions around the max pooled features, where δ actually controls the widths of weight distributions for F_g, F_p, and F_o. So, we share the same δ for F_g, F_p, and F_o to keep that they are aggregated by distributions with same widths. With such an Adaptive Pooling operation in Eq. 5 and Eq. 6, each item in F_g, F_p, and F_o can acquire various gradients during the back propagation process, instead of gradients all the same in average pooling or only constraining max items in max pooling.

Algorithm 1 Training Process

Input: Input S_i, ground truths S_g, the number of iterations $iter$, the task network $TaskNet(\cdot)$
for $n = 1$ **to** $iter$ **do**
  Calculate output of the task network:
  $S_o^n = TaskNet(S_i^n)$.
  Let θ_C and θ_T be the parameters of CALoss and the task network, respectively.
  Fix the parameter of task network and update CALoss by descending gradient:
  $\nabla_{\theta_C} L_C(S_o^n, S_g^n)$.
  Fix CALoss and update the task network by descending gradient:
  $\nabla_{\theta_T} L_T(S_o^n, S_g^n)$.
end for

3.3. Contrastive Adversarial Training

As presented in Fig. 2, losses L_p, L_r^{adv}, and L_r are calculated by ground truths and reconstructed results from the task. The training losses for CALoss and the task network can be defined as:

$$\begin{aligned} L_C =& L_r^{adv} + \frac{\epsilon}{|\mathcal{N}_\sigma|} \cdot L_p + \epsilon_w \cdot |\delta|^2 \\ =& -log(L_r + \sigma_r) + \frac{\epsilon}{|\mathcal{N}_\sigma|} \cdot L_p + \epsilon_w \cdot |\delta|^2, \end{aligned} \tag{7}$$

Dataset	ShapeNet								ModelNet40							
Networks	AE		Folding		LAE		LFolding		AE		Folding		LAE		LFolding	
Metrics	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD	MCD	HD
CD [3]	0.32	1.87	0.40	4.13	0.31	1.02	0.28	1.20	0.75	6.08	0.83	7.35	0.44	1.69	0.39	2.16
EMD [3]	0.25	2.23	-	-	0.23	2.48	0.21	2.49	0.61	6.18	-	-	0.33	3.82	0.32	3.88
PUD [10]	0.32	1.88	0.36	3.83	0.32	1.02	0.27	1.11	0.73	5.85	0.77	7.29	0.45	1.71	0.38	1.97
PFD [8]	0.32	1.87	0.41	4.14	0.31	0.99	0.26	0.97	0.74	6.28	0.88	7.55	0.44	1.69	0.35	1.69
CRND [21]	0.31	1.86	0.34	3.17	0.31	1.00	0.26	0.99	0.71	5.66	0.76	7.24	0.44	1.69	0.35	1.76
DCD [23]	0.28	1.75	0.91	8.41	0.13	0.89	0.18	1.20	0.68	6.02	1.22	11.86	0.17	1.37	0.24	1.81
PCLoss [6]	0.23	1.66	0.33	2.57	0.13	**0.65**	0.14	**0.76**	0.59	5.30	0.75	6.65	0.17	**1.05**	0.19	1.37
Ours	**0.21**	**1.53**	**0.30**	**2.57**	**0.12**	0.76	**0.12**	0.79	**0.58**	**5.23**	**0.72**	**6.32**	**0.16**	1.18	**0.16**	**1.24**

Table 1. Comparison with reconstruction losses on ShapeNet (SP) and ModelNet40 (MN40). **Bold** marks the best results.

and

$$L_T = L_r, \tag{8}$$

where ϵ and ϵ_w are two hyper-parameters to adjust the weights. The whole training process for CALoss and the task network can be described as Alg. 1. Parameters of CALoss and the task network are updated by turns in each iteration. CALoss is updated by L_r^{adv} and L_p. L_p is used to constrain that similar shapes S_p, S_g have close representations, where L_r^{adv} can promote CALoss to find the shape differences between S_g and S_o. We give a dynamic weight for L_p controlled by $1/|\mathcal{N}_\sigma|$, which means more noised S_p are allowed to have relatively further representations. Besides, we add a L2 regularization for δ to prevent the weights for F_g, F_p, and F_o from over-smoothness. According to Eq. 5, too large δ will result in roughly the same weighting for each item in F_g, F_p, or F_o, which is harmful for delivering variable gradients. The task network is optimized by L_r to reduce the differences found by CALoss between reconstructed results S_o and ground truth S_g.

4. Experiments

4.1. Datasets and Implementation Details

In this work, three point cloud datasets: ShapeNet [26], ModelNet10 (MN10), and ModelNet40 (MN40) [24] are adopted. We use the ShapeNet part dataset [1, 25] containing 12288 models in the train split and 2874 models in the test split. ModelNet10 and ModelNet40 are subsets of ModelNet, which contain 10 categories and 40 categories of CAD models, respectively. Each model consists of 2048 points randomly sampled from the surfaces of original mesh models. We conduct comparisons with other losses on three tasks, including point cloud reconstruction, unsupervised classification, and point cloud completion.

For the reconstruction task, we train networks with different reconstruction losses on the train split of ShapeNet part dataset and evaluate performances on both the test split of ShapeNet and MN40 to provide a robust and exhaustive evaluation. For the unsupervised classification task, we compare the performances of different losses on multiple auto-encoders constructed by [1, 15, 22, 25] following PCLoss [6]. As for GLRNet [16], we follow its setting and retrain it with the original adopted CD and CALoss to observe the differences. For the point cloud completion task, we introduce 3 popular works PCN [27], CRN [21], and RFNet [7] to compare the completion performances before and after replacing the adopted reconstruction losses with CALoss. PCN and CRN are trained on the dataset provided by CRN with 2048 points to compare the completion performances on sparse point clouds, while RFNet is trained on the corresponding dataset with 16384 points to see the completion performances on dense point clouds. All data are normalized to $-1 \sim 1$ for the fairness of comparison.

Metrics. To provide a clear and accurate evaluation of the performance, we adopt Multi-scale Chamfer Distance (MCD) and Hausdorff Distance (HD) introduced in [6] as metrics for reconstruction quality assessment in this work.

4.2. Comparisons on Point Cloud Reconstruction

In this section, we conduct comparisons with different reconstruction losses based on a few commonly-used networks AE [1], Folding [25] and local feature-based LAE and LFolding following PCLoss [6]. We retrain the networks with different reconstruction losses and evaluate the reconstruction errors of trained networks on the test split of ShapeNet and ModelNet40. As introduced in Sec. 2.2, CD and EMD are widely-used matching-based reconstruction losses, while PUD [10], PFD [8], CRND [21] are constraints introducing extra discriminators to improve the reconstruction performances. DCD [23] is a recent variant of CD achieved by modifying the matching rule. In this work, we choose the better-performed LNSA proposed in PCLoss [6] as PCLoss for comparison in following sections.

The quantitative results are presented in Table 1. We can see that CALoss can achieve the best performances in

most conditions. To intuitively present the differences in reconstructed results, we also conduct a qualitative comparison in Fig. 3. We can see that CD may create quite non-uniform results with missing local details, while EMD may produce distorted shapes. Though PUD, CRND, and PFD can improve the integrity of shapes, they still produce similar shapes as CD contained within. PCLoss improves both the uniformity and integrity of reconstructed shapes. But it gets relatively rough details as shown in Fig. 3. As shown by circled regions in the last two rows of Fig. 3, the results of PCLoss create rough contours on chair legs and lamp stands, while our method produces clearer details on these regions. It confirms the effectiveness of CALoss.

4.3. Comparisons on Unsupervised Classification

In this section, we evaluate the performances of CALoss on point cloud unsupervised classification based on multiple auto-encoders constructed by [1, 15, 22, 25] with 128-dim bottleneck following PCLoss [6], and GLRNet [16]. The experimental settings are kept the same as [1, 16, 25].

The auto-encoders are trained on ShapeNet and applied on ModelNet10 and ModelNet40 to extract point cloud representations, where the representations extracted from the training splits and corresponding labels are adopted to train Supported Vector Machines (SVMs). Then the distinguishability of extracted representations can be evaluated by the classification accuracy of these SVMs. We conduct comparisons on these networks by replacing the adopted reconstruction losses to train the auto-encoders with CALoss and observe the changing in classification accuracy. From the results in Table 2, we can see that most networks can achieve improvements by replacing the reconstruction losses with CALoss, which confirms that CALoss can help the task networks learn more representative representations.

TaskNet	Dataset	Methods			
		CD	EMD	PCLoss	Ours
AE	MN10	90.60	89.49	91.48	**91.48**
	MN40	85.92	85.47	86.36	**86.81**
Folding	MN10	91.03	-	91.70	**92.26**
	MN40	85.22	-	85.35	**86.24**
AE (PN++)	MN10	90.38	90.15	92.04	**93.47**
	MN40	88.03	88.07	87.54	**88.15**
Folding (PN++)	MN10	91.48	-	91.48	**92.59**
	MN40	87.01	-	86.73	**87.13**
AE (DGCNN)	MN10	91.37	91.26	92.37	**92.81**
	MN40	87.50	87.54	**88.11**	87.46
Folding (DGCNN)	MN10	91.26	-	91.81	**92.70**
	MN40	86.85	-	87.50	**87.74**
GLRNet	MN10	93.58	-	-	**95.24**
	MN40	91.07	-	-	**91.31**

Table 2. Comparison on unsupervised classification.

Network	PCN		CRN		RFNet		RFNet*	
Metri	MCD	HD	MCD	HD	MCD	HD	MCD	HD
w/o CALoss	0.31	2.67	0.29	**2.42**	0.21	2.92	0.29	3.82
w/ CALoss	**0.31**	**2.55**	**0.29**	2.44	**0.20**	**2.63**	**0.27**	**3.28**

Table 3. Comparisons on point cloud completion. RFNet and RFNet* denote results evaluated on known and novel categories on ShapeNet following RFNet [7].

4.4. Comparisons on Point Cloud Completion

Point Cloud Completion predicts completed results as similar as possible to ground truths from partial inputs, which is usually trained with reconstruction losses between completed results and ground truths. To further verify the performances of CALoss, we apply it to a few popular point cloud completion works, including PCN [27], CRN [21], and RFNet [7]. As these works may have multilevel constraints, we conduct comparisons by replacing the reconstruction losses of the last level with CALoss and retraining the networks. The results are presented in Table 3. The completion performances have improvements in most conditions by introducing CALoss, which further confirms that CALoss is effective for different task networks.

4.5. Analysis about the Training Process

Visualization during training. We visualize a model generated by the task network AE [1] during training to observe the convergence of different losses. The results are presented in Fig. 4. We can see that CD and EMD have unchanged results with obvious defects after 200 iterations, which means they actually converge to inappropriate local minimums. The reconstructed results trained with CALoss converge to a simple shape after 100 iterations, which may be the effect of contrastive constraint to help the task network find a shape similar to ground truths. From 200 $\sim$ 600 iterations, the trained results of CALoss gradually remove differences and approach the ground truth, which confirms the adversarial loss can continuously help find the defects and promote the task network to get better performances. Although PCLoss also removes defects during iterations, it produces rougher results, which may come from the limitation of 3D Euclidean space-based descriptors.

Training curves. In this section, we visualize the reconstruction errors based on the AE network [1] and ShapeNet dataset [24] through the training iterations to observe the convergences of difference loss functions. We can see that our method has relatively inferior performances at the beginning of iterations, where it is learning to search shape differences. But it will converge steadily to low errors after

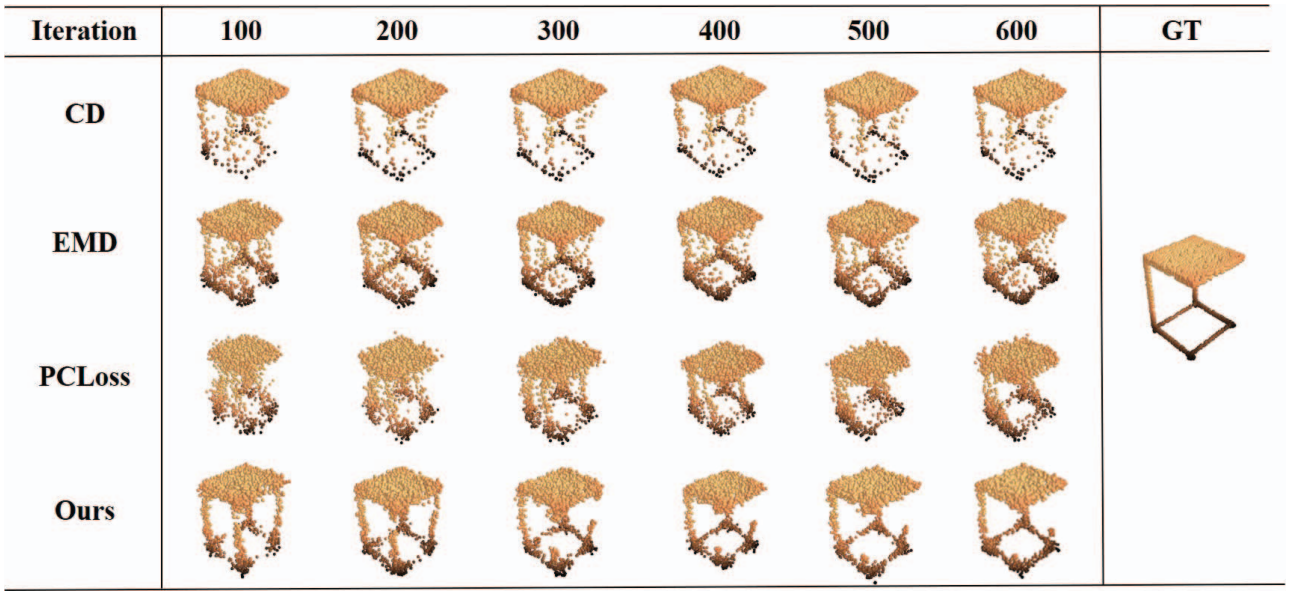

Figure 4. The visualization of training processes with different reconstruction losses. CD/EMD fall into local minimums and acquire unchanged results after 200 iterations, while PCLoss produces relatively rough shapes. Our method can gradually remove the shape defects and create more accurate results.

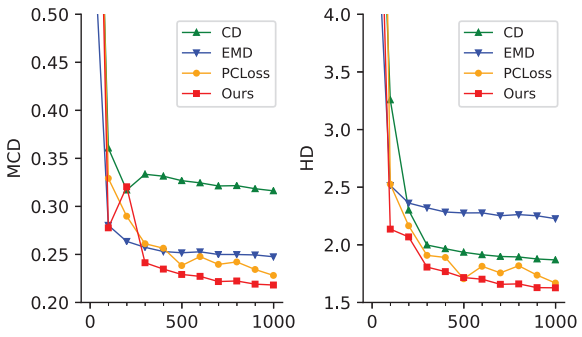

Figure 5. The reconstruction error curves through the iterations.

enough iterations. We can also see that CALoss converge faster than PCLoss, which reveals the priority of our representation space-based measurement.

4.6. Training Efficiency Comparison

In this section, we evaluate the training efficiencies of different reconstruction losses on the AE network [1], which are measured by the time consumed for the training of a single batch. The results are presented in Table 4. Though our method performs slightly slower than CD, it has the highest training efficiency in learning-based losses.

	Non-learning		Learning-based				
Methods	CD	EMD	PUD	PFD	CRND	PCLoss	Ours
Time(ms)	**23**	216	77	45	97	57	**39**

Table 4. Training efficiency comparison conducted on an NVIDIA 2080ti with a 2.9GHz i5-9400 CPU.

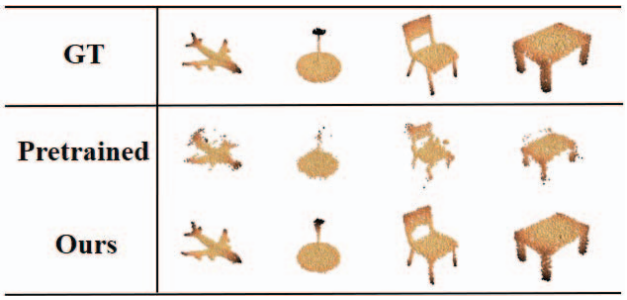

Figure 6. Qualitative comparisons between our method dynamically updated and pre-trained without updating.

4.7. Necessity of the Dynamic Updating

CALoss is dynamically optimized together with the task network as claimed in Sec. 3. To show the necessity of the dynamic updating, we make an attempt to train the task network AE [1] with a pre-trained CALoss directly without further updating. The results are demonstrated in Fig. 6 and Table 5. We can see that the task network trained with pre-trained CALoss can only reconstruct quite rough shapes, which confirms that training with CALoss is actually a continuous procedure to search shape differences between ground truths and the task network outputs.

4.8. Ablation Study

Ablation study for components. We conduct an ablation study for the components adopted in CALoss as mentioned in Eq. 7. The results are presented in Table 6. L_p and

	Pre-trained		Ours	
Dataset	SP	MN40	SP	MN40
MCD	0.75	1.04	**0.21**	**0.58**
HD	14.48	14.15	**1.53**	**5.23**

Table 5. Quantitative comparisons between our method dynamically updated and pre-trained without updating.

L_r^{adv} are the basic contrastive and adversarial constraints, respectively, while $|\delta|^2$ is the regularization constraint for δ. $1/|\mathcal{N}_\sigma|$ is the dynamic coefficient for the weight of L_p.

As finding shape difference is an essential constraint to prevent CALoss from acquiring all-zero output under the supervision of only L_p, we remove the L_r^{adv} by replacing it with the negative implementation of metric-learning method [2, 16] by maximizing the representation distances between models in the same mini-batch.

We can see that L_p and L_r^{adv} both have very significant influences on the final performance, which means they are cores of CALoss. Replacing L_r^{adv} with metric-learning method has weaker results. It confirms that maximizing representations between shapes within a mini-batch is not enough to learn the shape differences because it lacks of dynamic feedback from reconstructed outputs and cannot accurately find the shape differences. The regularization $|\sigma|^2$ also has obvious influence, which means it is important to control the widths of weight distributions to aggregate representations in $g(\cdot)$ as shown in Fig. 2 and Eq. 5.

	L_p	*Perturb*	L_{adv}	$\|\delta\|^2$	$1/\|\mathcal{N}_\sigma\|$	MCD	HD
L_p/L_{adv}			✓			3.55	7.57
	✓	✓				1.83	12.76
Others	✓	✓	✓			0.77	8.13
	✓	✓	✓	✓		0.22	1.63
	✓	✓	✓	✓	✓	**0.21**	**1.53**

Table 6. Ablation for components. Perturb denotes the perturbation, while L_p, L_r^{adv} and $|\delta|^2$ are components included in Eq. 7. $1/|\mathcal{N}_\sigma|$ is the dynamic coefficient for the weight of L_p in Eq. 7.

Ablation study for the Pooling Controller. In Pooling Controller, we introduce both max pooled and average pooled features to acquire more extensive information about the overall distribution in F_g as presented in Fig. 2 and Eq. 4. We conduct a simple ablation study for this operation in Table 7. We can see that the designation of concatenation can indeed reduce the reconstruction errors.

Ablation study the Adaptive Pooling operation. Here, We present a simple discussion about the proposed Adaptive Pooling operation. From Table 8, we can see that both max and average pooling have quite inferior performances, which prove the necessity of Adaptive Pooling operation.

Metrics	Max	Avg	Ours
MCD	0.22	0.23	**0.21**
HD	1.58	1.55	**1.53**

Table 7. Comparisons between different implementations of Pooling Controller on ShapeNet and AE [1]. Max and Avg denote introducing only max and average pooling, respectively.

Metrics	Max	Avg	Ours
MCD	0.34	0.61	**0.21**
HD	1.72	6.79	**1.53**

Table 8. Comparisons between pooling operations on ShapeNet and AE [1]. Max and Avg denote max and average pooling.

In CALoss, the pooling operation is used to aggregate features from all points into a global representation. To train the task network, the global representation needs to provide variable gradients for each point feature to distinguish them. However, average pooling can only propagate same and indistinguishable gradients for each points. Although max pooling can provide different gradients for point features, it provides a hard 0-1 distribution where only max items are constrained. In this condition, we design such an Adaptive Pooling to get a variable weight for each point feature according to the distance to the max pooled feature, which can be regarded as a "soft max pooling". All point features can be constrained with distinguishable gradients, which is controlled by widths of weight distributions predicted with Pooling Controller $h(\cdot)$ as shown in Fig. 2 and Eq. 5.

5. Conclusion

In this work, we propose a novel learning-based framework named CALoss to train the point cloud reconstruction-related task networks by measuring the differences between reconstructed shapes and ground truths with distances in a representation space. With the measurement of reconstruction loss in the learned non-linear representation space, CALoss has more extensive searching space for shape differences and better performances than the Euclidean space-based methods. According to the experiments, CALoss can achieve improvements above existing reconstruction losses based on predefined matching rules on multiple tasks including point cloud reconstruction, unsupervised classification and completion, which confirms it can help the task network achieve better reconstruction performances and extract more representative representations.

Acknowledges

We thank all authors, reviewers and the chair for excellent contributions. This work is supported by the Key Research and Development Project of Zhejiang Province under Grant 2021C01035.

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. 2, 4, 5, 6, 7, 8

[2] 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. 1, 3, 8

[3] Haoqiang Fan, Hao Su, and Leonidas 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*, pages 605–613, 2017. 2, 5

[4] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014. 2

[5] Tianxin Huang and Yong Liu. 3d point cloud geometry compression on deep learning. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 890–898, 2019. 1

[6] Tianxin Huang, Xuemeng Yang, Jiangning Zhang, Jinhao Cui, Hao Zou, Jun Chen, Xiangrui Zhao, and Yong Liu. Learning to train a point cloud reconstruction network without matching. In *European Conference on Computer Vision*, pages 179–194. Springer, 2022. 1, 2, 3, 5, 6

[7] Tianxin Huang, Hao Zou, Jinhao Cui, Xuemeng Yang, Mengmeng Wang, Xiangrui Zhao, Jiangning Zhang, Yi Yuan, Yifan Xu, and Yong Liu. Rfnet: Recurrent forward network for dense point cloud completion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12508–12517, 2021. 1, 2, 5, 6

[8] Zitian Huang, Yikuan Yu, Jiawen Xu, Feng Ni, and Xinyi Le. Pf-net: Point fractal network for 3d point cloud completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7662–7670, 2020. 2, 3, 5

[9] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 1

[10] Ruihui Li, Xianzhi Li, Chi-Wing Fu, Daniel Cohen-Or, and Pheng-Ann Heng. Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7203–7212, 2019. 2, 3, 5

[11] 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. 1, 2

[12] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8895–8904, 2019. 2

[13] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1

[14] Trung Nguyen, Quang-Hieu Pham, Tam Le, Tung Pham, Nhat Ho, and Binh-Son Hua. Point-set distances for learning representations of 3d point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10478–10487, 2021. 3

[15] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 5, 6

[16] Yongming Rao, Jiwen Lu, and Jie 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*, pages 5376–5385, 2020. 1, 2, 5, 6, 8

[17] N Dinesh Reddy, Minh Vo, and Srinivasa G Narasimhan. Carfusion: Combining point tracking and part detection for dynamic 3d reconstruction of vehicles. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1906–1915, 2018. 1

[18] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019. 1

[19] Jia-Heng Tang, Weikai Chen, Jie Yang, Bo Wang, Songrun Liu, Bo Yang, and Lin Gao. Octfield: Hierarchical implicit functions for 3d modeling. *arXiv preprint arXiv:2111.01067*, 2021. 1

[20] Dahlia Urbach, Yizhak Ben-Shabat, and Michael Lindenbaum. Dpdist: Comparing point clouds using deep point cloud distance. In *European Conference on Computer Vision*, pages 545–560. Springer, 2020. 3

[21] 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, 3, 5, 6

[22] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019. 2, 5, 6

[23] 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. *arXiv preprint arXiv:2111.12702*, 2021. 2, 3, 5

[24] 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. 5, 6

[25] 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, 5, 6

[26] 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

[27] Wentao Yuan, Tejas Khot, David Held, Christoph Mertz, and Martial Hebert. Pcn: Point completion network. In *2018 International Conference on 3D Vision (3DV)*, pages 728–737. IEEE, 2018. 2, 5, 6

[28] Yongheng Zhao, Tolga Birdal, Haowen Deng, and Federico Tombari. 3d point capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1009–1018, 2019. 2