CONTRASTIVE ADVERSARIAL POINT CLOUD RECONSTRUCTION LOSS

Anonymous authors
Paper under double-blind review

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. The Chamfer Distance (CD) and Earth Mover's Distance (EMD) are two widely-used reconstruction losses, which firstly use predefined strategies to match points in two point clouds and then apply the average distances from points to their matched neighbors as differentiable measurements of shape differences. However, the predefined matching rules may deviate from the real shape differences and cause defective reconstructed results. To solve the above problem, we propose a learning-based Contrastive adversarial Loss (CALoss) to train a reconstruction-related task network without the predefined matching rules. CALoss learns to evaluate shape differences by combining the contrastive constraint with the adversarial strategy. Specifically, we use the contrastive constraint to help CALoss learn 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.

1 INTRODUCTION

Point clouds, as the common description for 3D shapes, have been broadly used in many areas such as 3D detection Reddy et al. (2018); Shi et al. (2019) and surface reconstruction Mescheder et al. (2019); Jiang et al. (2020); Tang et al. (2021). For the point cloud reconstruction-related tasks Huang & Liu (2019); Huang et al. (2021); Liu et al. (2020); Rao et al. (2020), 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 Chamfer Distance (CD) and Earth Mover's Distance (EMD) as reconstruction losses 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 shapes are obviously different but with small point-to-point CD/EMD distances. PFNet Huang et al. (2020), PUGAN Li et al. (2019), and CRN Wang et al. (2020) 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.

In this work, we propose a novel framework named Contrastive Adversarial Loss (CALoss) to get over the limitation of predefined matching rules in existing losses and learn to evaluate the shape differences during training. The differences between our work and existing commonly-used CD and EMD are presented in Fig. 1. CALoss is composed of L_p, L_r, and L_r^{adv} as shown in Fig. 1. L_p is calculated by the representation distances between ground truths S_g and positive samples S_p constructed by perturbation Chen et al. (2020), while L_r and L_r^{adv} are acquired from the

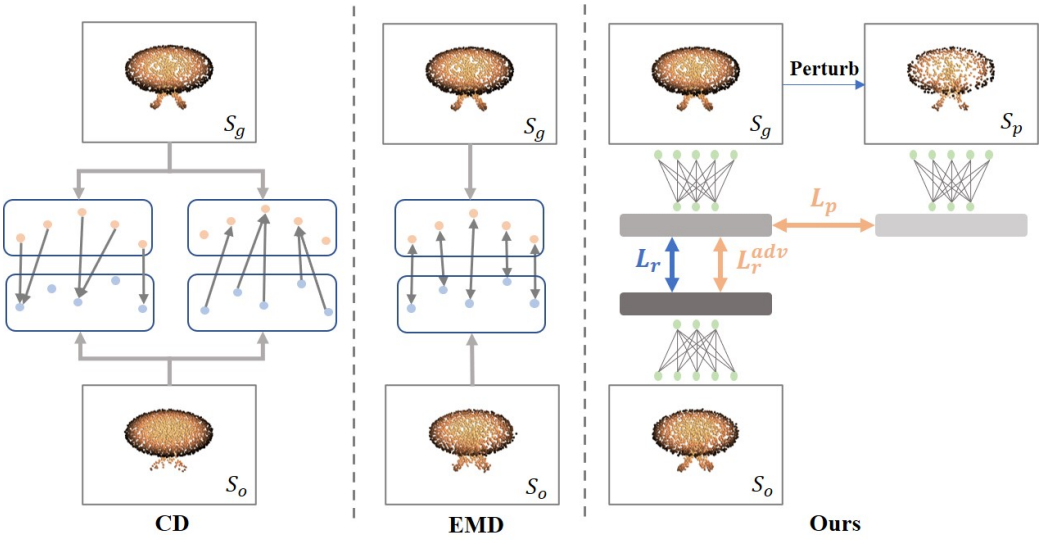

Figure 1: The comparison of reconstruction losses. S_g and S_o denote ground truths and point clouds generated by the task network. S_p is a positive sample with similar shapes as S_g acquired by perturbation. CD and EMD constrain shape differences by reducing distances between points matched by different *static* predefined rules, while our CALoss uses *dynamically* learned representation distances as shape differences. 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.

representation distances between ground truths S_g and reconstructed results S_o. L_r^{adv} and L_p are used to optimize CALoss, where L_r is adopted to train the task network. L_p is the contrastive constraint used to help CALoss learn 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.

Our contribution in this work can be summarized as

- We propose a novel framework named Contrastive adversarial loss (CALoss) to replace matched point-to-point distances with global representation distances in point cloud reconstruction-related tasks.
- By combining the contrastive constraint and adversarial training strategy, CALoss can learn to evaluate point cloud differences dynamically and avoid unexpected local minimums.
- Experiments on point cloud reconstruction, unsupervised classification, and point cloud completion demonstrate that CALoss can help the task network improve reconstruction performances and learn more representative representations.

2 Related Work

2.1 Point Cloud Reconstruction-related Tasks

Base on the basic point cloud reconstruction framework, 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 Yang et al. (2018); Achlioptas et al. (2018) 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 Wang et al. (2019); Zhao et al. (2019); Liu et al. (2019), while some researchers Rao et al. (2020) introduce extra supervision to

enhance the learning effection. Point Cloud Completion predicts completed point clouds as identical as possible to the ground truth point clouds from partial input point clouds. Early works Liu et al. (2020); Yuan et al. (2018) often use typical auto-encoders to abstract long global features from partial inputs and predict completed results, while recent work Wang et al. (2020); Huang et al. (2021) 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) Yuan et al. (2018) and Earth Mover's Distance (EMD) Fan et al. (2017) are two basic and broadly used reconstruction losses to constrain the shape differences, which calculate the distance between point clouds with different matching strategies. The Chamfer Distance (CD) is defined as:

$$\mathcal{L}_{CD}(S_g, S_o) = \frac{1}{2}(\frac{1}{|S_g|}\sum_{x\in S_g}\min_{y\in S_o}\|x-y\|_2 + \frac{1}{|S_o|}\sum_{x\in S_o}\min_{y\in S_g}\|x-y\|_2), \tag{1}$$

where S_g and S_o are two point sets. CD is the average distance from points in one set to their nearest neighbors in another set. A same nearest neighbor is allowed for multiple points for the calculation of CD. With the matching by nearest neighbors, CD concentrates on differences between contours, while it often constructs non-uniform surfaces as discussed in Fan et al. (2017); Wu et al. (2021).

The Earth Mover's Distance (EMD) can be presented as:

$$\mathcal{L}_{EMD}(S_g, S_o) = \min_{\phi:S_g\to S_o}\frac{1}{|S_g|}\sum_{x\in S_g}\|x-\phi(x)\|_2, \tag{2}$$

where S_g and S_o are two point sets. 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. An optimization process is needed to construct the bijection in each iteration. In practice, exact computation of EMD is too expensive for deep learning, even on graphics hardware, where we follow Fan et al. (2017) to conduct a $(1+\epsilon)$ approximation scheme for EMD. The algorithm is easily parallelizable on GPU. EMD can create more uniform shapes by constructing bijection, while the optimized matching may cause distortions. Besides, EMD can only be applied to reconstructed output with the same number of input due to the one-to-one matching, which limits its application. Since the development of GAN Goodfellow et al. (2014), researchers have introduced different discriminators as extra supervisions to better capture the shape differences and improve reconstruction performances Huang et al. (2020); Wang et al. (2020); Li et al. (2019). However, these works still need CD or EMD as basic shape constraints. Some works Nguyen et al. (2021); Wu et al. (2021) further modify the matching rules to improve the constraining performances. All these works by *point-to-point distance* calculated with *predefined* matching rules. Though DPDist Urbach et al. (2020) estimates the shape distances with a pre-trained network without any matching, it is mainly designed for registration instead of reconstruction, which is also inflexible due to the requirements of appropriate pre-training process. In this work, we propose CALoss to evaluate shape differences and train reconstruction-related networks. The similarities and dissimilarities of shapes are both *learned dynamically* during training, which may overcome the defects from predefined rules.

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:

$$L_p = \|C_g, C_p\|_1, L_r = \|C_g, C_o\|_1, \tag{3}$$

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

$$L_r^{adv} = -log(L_r + \sigma_r). \tag{4}$$

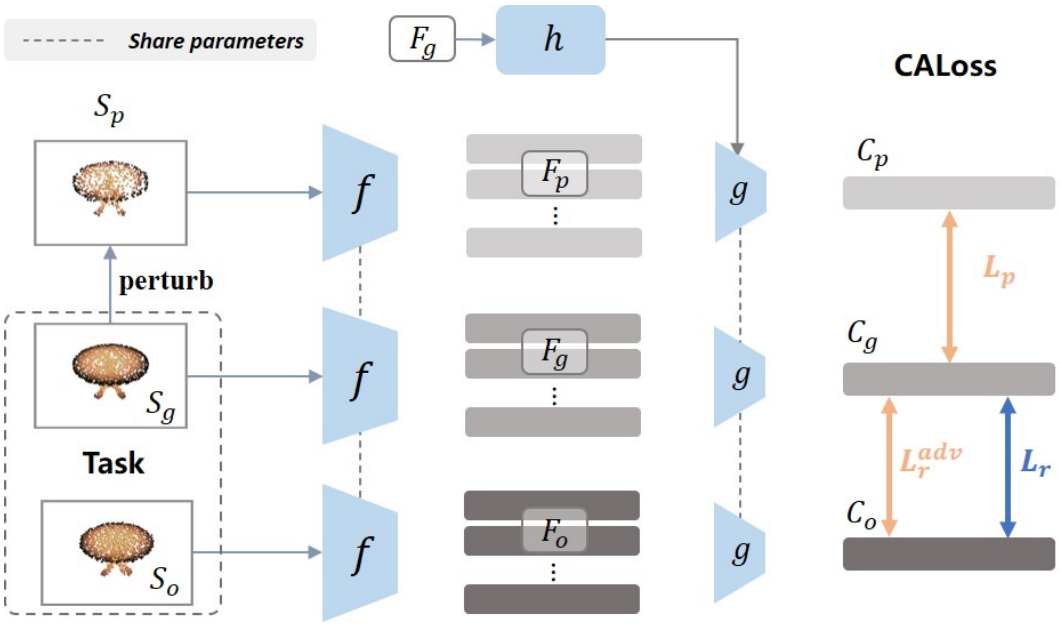

Figure 2: The illustration of CALoss. S_p is acquired from S_g with small perturbations. $f(\cdot)$ is a group of 1D-convolutions to transform point cloud S_g, S_p, and S_o into F_g, F_p, and F_o in the feature space. $g(\cdot)$ denotes our proposed adaptive pooling operation to aggregate the features F_g, F_p, and F_o based on all points into global representations C_g, C_p, and C_o, where $h(\cdot)$ is Pooling Controller predicting parameters to control the adaptive pooling $g(\cdot)$ 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 Chen et al. (2020) may not work because it lacks of dynamic feedback from S_o and cannot capture detailed shape differences.

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

$$S_p = S_g + N_\sigma, \tag{5}$$

where $N_\sigma = \text{randn}(\sigma)$. In other words, the noise width is controlled by σ. This operation creates perturbed point clouds with similar but different shapes as ground truths.

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. 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:

$$\delta = h(F_g) = MLP(Con(maxpool(F_g), avgpool(F_g))). \tag{6}$$

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

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 representation C_g can be defined as:

$$C_g = g(F_g, \delta) = \sum_{i=1}^{|F_g|} \frac{e^{-\|F_g^i - maxpool(F_g)\|/\delta}}{\sum_{i=1}^{|F_g|} e^{-\|F_g^i - maxpool(F_g)\|/\delta}} \cdot F_g. \tag{7}$$

C_p and C_o can be acquired by the same equations:

$$C_p = g(F_p, \delta), C_o = g(F_o, \delta). \tag{8}$$

We can see that 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. 7 and Eq. 8, 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.

3.3 Contrastive Adversarial Training

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

$$L_C = L_r^{adv} + \frac{\epsilon}{|N_\sigma|} \cdot L_p + \epsilon_w \cdot |\delta|^2 = -log(L_r + \sigma_r) + \frac{\epsilon}{|N_\sigma|} \cdot L_p + \epsilon_w \cdot |\delta|^2, \tag{9}$$

$$L_T = L_r, \tag{10}$$

where σ_r is a tiny value to avoid errors when $L_r \to 0$. 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 turn 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 $1/|N_\sigma|$, which means more noised S_p are allowed to have relatively further representations.

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. Besides, we add a L2 regularization for δ to prevent the weights for F_g from over-smoothness. According to Eq. 7, too large δ will result in roughly the same weighting for each item in F_g, which is harmful for delivering variable gradients.

4 Experiments

4.1 Datasets and Implementation Details

In this work, three point cloud datasets: ShapeNet Yi et al. (2016), ModelNet10 (MN10), and ModelNet40 (MN40) Wu et al. (2015) are adopted. We use the ShapeNet part dataset Achlioptas et al. (2018); Yang et al. (2018) 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

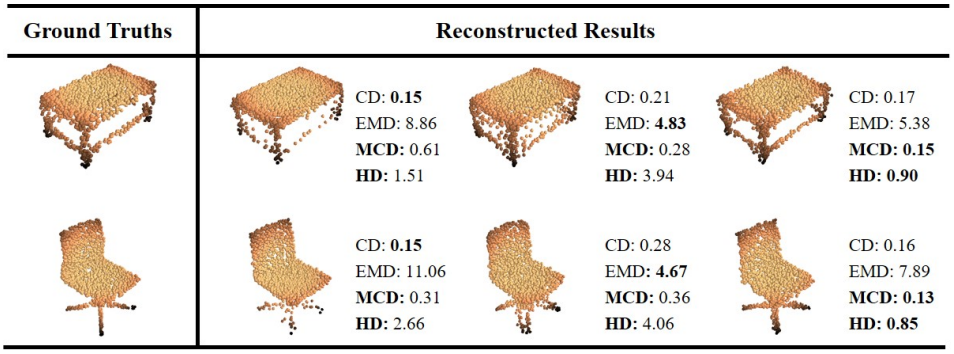

Figure 3: The comparisons between metrics. Lower CD and EMD may not mean more similar shapes, while MCD and HD metrics can better evaluate the shape differences in these conditions.

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 Qi et al. (2017); Wang et al. (2019); Yang et al. (2018); Achlioptas et al. (2018). As for GLRNet Rao et al. (2020), 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 Yuan et al. (2018), CRN Wang et al. (2020), and RFNet Huang et al. (2021) 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. As we have claimed in Sec. 1, CD and EMD may be limited by the predefined matching rules. An example is presented in Fig. 3. We can see that some reconstructed results may still deviate from the ground truths even with small CD or EMD metrics. To provide a clear and accurate evaluation of the reconstruction performance, we adopt multi-scale Chamfer Distance (MCD) proposed by Huang & Liu (2019) and Hausdorff Distance (HD) from Wu et al. (2020) as metrics in this work. Let the ground truth be S_g, reconstructed point cloud be S_o, MCD can be defined as:

$$MCD = \xi \cdot CD(S_g, S_o) + \frac{1}{|K|} \sum_{\forall k \in K} \frac{1}{|C|} \sum_{\forall c \in C} CD(S_g^{c,k}, S_o^{c,k}), \tag{11}$$

where C denotes centers of evaluated local regions, which is acquired with farthest point sampling (FPS) Qi et al. (2017) from S_g, S_o. K is a list including multiple k values to control the local region scales. $S_g^{c,k}$ means the local region on S_g with k points around center c. We can see that MCD evaluates both local and global reconstruction errors with Chamfer Distance, while ξ is a parameter to control their importance. HD can be defined as:

$$HD = \frac{1}{2}(\max_{x \in S_g} \min_{y \in S_o} \|x - y\|_2^2 + \max_{x \in S_o} \min_{y \in S_g} \|x - y\|_2^2). \tag{12}$$

We can see that HD measures the global worst reconstruction distortions, which is less influenced by the average matched results. MCD also reduces the sensitivity to the predefined matching rule by making an overall evaluation of both multiple local regions and the whole models. As shown in Fig. 3, MCD and HD can measure the shape differences well when CD and EMD meet failures. They are used as the metrics for reconstruction performances in later comparisons.

4.2 Comparisons with Matching-based Reconstruction Losses

In this section, we conduct comparisons with different reconstruction losses based on a few commonly-used networks. AE Achlioptas et al. (2018) and Folding Yang et al. (2018) are popular point

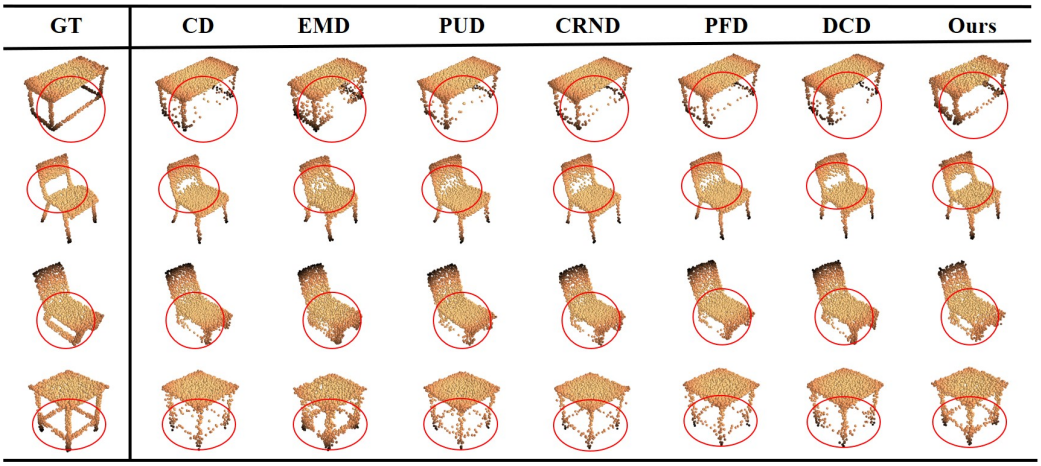

Figure 4: The qualitative comparisons with different losses based on AE Achlioptas et al. (2018).

cloud reconstruction networks based on global features, where we also apply AE and Folding in multiple local regions acquired following PointNet++ Qi et al. (2017) to construct local feature-based reconstruction networks LAE and LFolding. We retrain the networks with different reconstruction losses and evaluate the reconstruction errors of trained networks on the test split of ShapeNet and ModelNet40. CD and EMD are widely-used matching-based reconstruction losses, while reconstruction losses PUD Li et al. (2019), PFD Huang et al. (2020), CRND Wang et al. (2020) are constraints introducing extra discriminators to improve the reconstruction performances. DCD Wu et al. (2021) is a recent variant of CD by modifying the matching rule.

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. 4. 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 are still limited and produce similar shapes as CD contained within. The reconstructed results constrained with CALoss have uniform and complete shapes, which confirms its effectiveness.

Table 1: Comparison with reconstruction losses on ShapeNet (SP) and ModelNet40 (MN40).

RecNet	Metrics	SP							MN40						
		CD	EMD	PUD	PFD	CRND	DCD	CALoss	CD	EMD	PUD	PFD	CRND	DCD	CALoss
AE	MCD	0.32	0.25	0.32	0.32	0.31	0.28	**0.21**	0.75	0.61	0.73	0.74	0.71	0.68	**0.58**
	HD	1.87	2.23	1.88	1.87	1.86	1.75	**1.53**	6.08	6.18	5.85	6.28	5.66	6.02	**5.23**
Folding	MCD	0.40	-	0.36	0.41	0.34	0.91	**0.30**	0.83	-	0.77	0.88	0.76	1.22	**0.72**
	HD	4.13	-	3.83	4.14	3.17	8.41	**2.57**	7.35	-	7.29	7.55	7.24	11.86	**6.32**
LAE	MCD	0.31	0.23	0.32	0.31	0.31	0.13	**0.12**	0.44	0.33	0.45	0.44	0.44	0.17	**0.16**
	HD	1.02	2.48	1.02	0.99	1.00	0.89	**0.76**	1.69	3.82	1.71	1.69	1.69	1.37	**1.18**
LFolding	MCD	0.28	0.21	0.27	0.26	0.26	0.18	**0.12**	0.39	0.32	0.38	0.35	0.35	0.24	**0.16**
	HD	1.20	2.49	1.11	0.97	0.99	1.20	**0.79**	2.16	3.88	1.97	1.69	1.76	1.81	**1.24**

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 Achlioptas et al. (2018); Yang et al. (2018); Qi et al. (2017); Wang et al. (2019) with 128-dim bottleneck and GLRNet Rao et al. (2020). The experimental settings are kept the same as Achlioptas et al. (2018); Yang et al. (2018); Rao et al. (2020). We conduct comparisons on these networks by replacing the adopted CD or EMD reconstruction losses 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 loss with CALoss, which confirms that CALoss can help the task networks learn more representative representations.

Table 2: Comparison on unsupervised classification.

TaskNet		AE		Folding		AE(PN++)		Folding(PN++)		AE(DGCNN)		Folding(DGCNN)		GLRNet	
Dataset		MN10	MN40	MN10	MN40	MN10	MN40	MN10	MN40	MN10	MN40	MN10	MN40	MN10	MN40
Methods	CD	90.60	85.92	**91.03**	85.22	90.38	88.03	91.48	87.01	91.37	87.50	91.26	86.85	93.58	91.07
	EMD	89.49	85.47	-	-	90.15	88.07	-	-	91.26	87.54	-	-	-	-
	CALoss	**91.15**	**86.93**	90.92	**85.71**	**93.47**	**88.15**	**92.59**	**87.13**	**91.92**	**87.62**	**92.26**	**87.13**	**95.24**	**91.31**

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 Yuan et al. (2018), CRN Wang et al. (2020), and RFNet Huang et al. (2021). 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.

Table 3: Comparisons on point cloud completion. RFNet and RFNet* denote results evaluated on known and novel categories on ShapeNet following RFNet Huang et al. (2021).

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**	**2.56**	**0.29**	2.44	**0.20**	**2.63**	**0.27**	**3.28**

4.5 Analysis about the Training Process

We visualize a model generated by the task network AE Achlioptas et al. (2018) during training to observe the convergence of different losses. The results are presented in Fig. 5. 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 similar simple shape after 50 iterations, which may be the effect of contrastive constraint to help the task network find a shape similar to ground truths. From 100 $\sim$ 400 iterations, the trained results will 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.

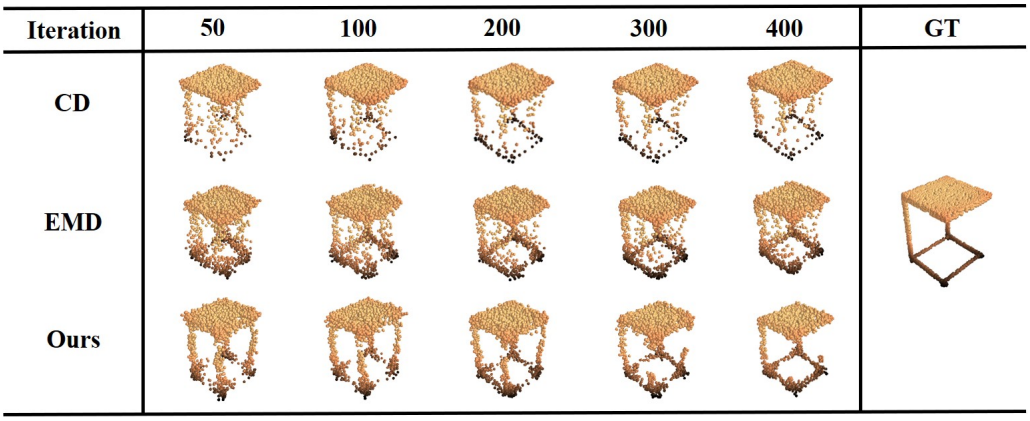

Figure 5: The visualization of training processes with different reconstruction losses.

4.6 TRAINING EFFICIENCY COMPARISON

In this section, we evaluate the training efficiencies of different reconstruction losses on AE networks Achlioptas et al. (2018), which are measured by the time consumed for the training of a single batch. The results are presented in Table 4. Though our method is a little slower than CD, it has much better performances according to discussions in former sections. CALoss has higher training efficiency than most reconstruction losses, which confirms its potential for efficiency.

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

Methods	CD	EMD	PUD	PFD	CRND	CALoss
Train Time(ms)	**23**	216	77	45	97	39

4.7 ABLATION STUDY

In this section, we conduct an ablation study for the components adopted in CALoss as mentioned in Eq. 9. The results are presented in Table 5. L_p and L_r^{adv} are the basic contrastive and adversarial constraints, respectively, while $|\delta|^2$ is the regularization constraint for δ. $1/|N_\sigma|$ 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 Rao et al. (2020); Chen et al. (2020) by maximizing the representation distances between models in the same mini-batch. We can see that L_p and L_r^{adv} 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. The regularization $|\sigma|^2$ also has obvious influence, which means it is important to control the aggregation of representations in Eq. 7.

Table 5: Ablation for components. Perturb denotes the perturbation, while L_p, L_r^{adv} and $|\delta|^2$ are components included in Eq. 9. $1/|N_\sigma|$ is the dynamic coefficient for the weight of L_p in Eq. 9.

	L_p	perturb	L_{adv}	$\|\delta\|^2$	$1/\|N_\sigma\|$	MCD	HD
L_p/L_{adv}			✓			3.55	7.57
	✓	✓				1.83	12.76
others	✓	✓	✓		✓	0.65	7.06
	✓	✓	✓	✓		0.22	1.63
	✓	✓	✓	✓	✓	**0.21**	**1.53**

5 CONCLUSION

In this work, we propose a novel learning-based framework named CALoss to train the reconstruction-related task networks in order to get over the limitation of predefined matching rules in CD or EMD. CALoss uses the representation distances between point clouds to evaluate shape differences. With contrastive constraints, CALoss learns the shape similarity that similar shapes have close representation distances, while the adversarial loss can guide CALoss to search for the differences between reconstructed results and ground truths. By updating dynamically with the task network, CALoss can help the task network avoid incorrect local minimums and promote the final reconstruction performances. According to the experiments, CALoss can achieve improvements above commonly-used 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 acquire better reconstruction performances and extract more representative representations.

REFERENCES

Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3d point clouds. In *International conference on machine learning*, pp. 40–49. PMLR, 2018.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pp. 1597–1607. PMLR, 2020.

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*, pp. 605–613, 2017.

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.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

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

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*, pp. 12508–12517, 2021.

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*, pp. 7662–7670, 2020.

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*, pp. 6001–6010, 2020.

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*, pp. 7203–7212, 2019.

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, pp. 11596–11603, 2020.

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*, pp. 8895–8904, 2019.

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*, pp. 4460–4470, 2019.

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*, pp. 10478–10487, 2021.

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*, pp. 5099–5108, 2017.

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*, pp. 5376–5385, 2020.

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*, pp. 1906–1915, 2018.

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*, pp. 770–779, 2019.

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.

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

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*, pp. 790–799, 2020.

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.

Cheng-Hao Wu, Chih-Fan Hsu, Ting-Chun Kuo, Carsten Griwodz, Michael Riegler, Géraldine Morin, and Cheng-Hsin 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*, pp. 1–6, 2020.

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.

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*, pp. 1912–1920, 2015.

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*, pp. 206–215, 2018.

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.

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

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*, pp. 1009–1018, 2019.

A Appendix

A.1 Implementation Details

Our work is implemented in **Tensorflow**. The batch size is set as 16. The learning rates of the target network and CALoss are set as 0.0001 and 0.003, respectively. They are both optimized with Adam optimizer. The specific settings of all hyper-parameters are illustrated in Table 6. The structures of

networks are presented in Table 7. For LAE and Lfolding, we adopt AE and Folding in 32 local regions, where each local network generates 64 points to acquire a 2048 points final output. All experiments are conducted on a NVIDIA 2080ti GPU with a 2.9GHZ i5-9400 CPU.

Table 6: Illustrations of hyper-parameters

	CALoss				MCD		
Name	σ	ϵ	ϵ_w	σ_r	K	C	ξ
Constants	0.01	0.003	2.0	10^{-8}	[4,8,16,32,64]	256	0.1

Table 7: Illustrations of network structures. All components presented are MLPs.

TaskNet	Encoder	Decoder
FC	MLPs(64,128,128,256,128)+Max-Pooling	FCs(256,256,2048*3)
Folding	MLPs(64,128,128,256,128)+Max-Pooling	MLPs(128,128,3) + MLPs(128,128,3)
CALoss	Layers	
Pooling Controller $h(\cdot)$	MLPs(256,128,128)	
1-D Convs $f(\cdot)$	MLPs(3,64,128)	

A.2 Further Theoretical Analysis

Analysis. Here we present a simple theoretical argument by considering it as an approach to the Wassertein distance. The Wassertein distance between shapes, also known as the minimum transmission distance, may measure the true distance between point clouds. However, the accurate Wassertein distance is too computational cost to calculate in applications. Existing matching-based reconstruction losses, including CD/EMD, actually work by approaching the Wassertein distance by different manually-defined rules such as caculating the distances between points to their nearest neighbors in other point clouds. WGAN Arjovsky et al. (2017) and WGAN-GP Gulrajani et al. (2017) calculate the wasserstein distance between the distributions x and $g_\theta(z)$ by

$$\underset{\omega \in W}{max}\, \mathbb{E}_{x\sim\mathbb{P}_r}[f_\omega(x)] - \mathbb{E}_{z\sim p(z)}[f_\omega(g_\theta(z))], \tag{13}$$

where $f_\omega(\cdot)$ and $g_\theta(\cdot)$ mean discriminator and generation network with parameters ω and θ, respectively. $\mathbb{P}_r$ is the domain of real data, while the $p(z)$ denotes the distribution of latent variables. To ensure the convergence of $f_\omega(\cdot)$, K-Lipschitz Arjovsky et al. (2017); Gulrajani et al. (2017) should be satisfied that $|f_\omega(x_1) - f_\omega(x_2)| < K \cdot |x_1 - x_2|$, where x_1 and x_2 are two values in the domain.

If we define the whole transformation from shape S to global representation C as $f_c(\cdot)$, then we have $C = f_c(S)$. Let us define the task network transform input S_i to reconstructed S_o as f_T, that is $S_o = f_T(S_i)$. S_g and S_p are the ground truths and perturbed ground truths as described in Sec. 3.

The adversarial optimization of CALoss can then be described as

$$\begin{aligned}
\underset{\omega_c\in\theta_C}{min}\, L_C &= -\underset{\omega_c\in\theta_C}{min}\, log(|f_c(S_g) - f_c(S_o)|) + \frac{\epsilon}{|N_\sigma|}\cdot|f_c(S_g) - f_c(S_p)| + \epsilon_w \cdot |\delta|^2 \\
&= -\underset{\omega_c\in\theta_C}{min}\, log(|f_c(S_g) - f_c(S_o)|) + \frac{\epsilon}{|S_g - S_p|}\cdot|f_c(S_g) - f_c(S_p)| + \epsilon_w \cdot |\delta|^2 \\
&\propto \underset{\omega_c\in\theta_C}{max}\, |f_c(S_g) - f_c(f_T(S_i))| + \underset{\omega_c\in\theta_C}{min}\, \frac{|f_c(S_g) - f_c(S_p)|}{|S_g - S_p|} + \underset{\omega_c\in\theta_C}{min}\, |\delta|^2
\end{aligned} \tag{14}$$

We can see that our first term can be regarded a symmetric form of WGAN Arjovsky et al. (2017) distance, where the K-Lipschitz Arjovsky et al. (2017); Gulrajani et al. (2017) can be garanteed by the second term of adversarial loss that $\frac{|f_c(S_g)-f_c(S_p)|}{|S_g-S_p|} < \eta < K$ can be satisfied after enough iterations. η is a tiny value related to the convergence. In this condition, the optimization of CALoss can be approximately regarded as dynamically learning the Wassertein distances between point clouds, which may explain its effectiveness. CALoss does not have to describe the whole shape within a

Table 8: Comparison with reconstruction losses on ShapeNet(SP) and ModelNet40(MN40). **Bold** and underline mark the best and second best items, respectively.

RecNet	Metrics	SP							MN40						
		CD	EMD	PUD	PFD	CRND	DCD	CALoss	CD	EMD	PUD	PFD	CRND	DCD	CALoss
AE	CD	**0.23**	0.28	0.23	0.23	0.23	0.23	0.26	0.87	**0.79**	0.85	0.86	0.82	0.84	0.80
	EMD	13.01	**5.33**	12.95	13.20	13.09	12.30	8.47	19.28	**8.80**	19.70	19.30	18.99	17.39	12.83
Folding	CD	**0.31**	-	0.31	0.32	0.31	1.08	0.39	1.00	-	0.96	1.11	**0.95**	1.71	1.08
	EMD	13.74	-	12.69	14.38	11.89	16.43	**10.36**	20.62	-	16.96	20.11	15.86	23.09	**13.98**
LAE	CD	0.17	0.25	0.17	0.17	0.17	**0.11**	0.14	0.28	0.39	0.28	0.27	0.28	0.22	**0.20**
	EMD	14.53	**5.86**	14.57	14.61	14.51	10.46	7.54	14.55	**7.23**	14.62	14.65	14.55	7.25	8.24
LFolding	CD	0.16	0.23	0.16	0.14	0.14	0.14	**0.14**	0.28	0.37	0.26	0.23	0.24	0.21	**0.20**
	EMD	13.59	**5.55**	13.42	13.54	13.56	10.61	7.53	13.76	**7.00**	13.62	13.62	13.67	10.78	12.40

global representation. It works by dynamically searching and constraining the shape differences during the adversarial training. As CALoss can learn to approach the Wassertein distance more accurately without any predefined rules, it may achieve better performances.

A.3 Comparisons on CD/EMD metrics

To present a more comprehensive evaluation for the reconstruction performances, we also compare the task networks trained with different reconstruction losses based on CD/EMD metrics. As EMD cannot work when the reconstructed results and ground truths have different point numbers, the EMD metric under this condition is estimated by randomly re-sampling the reconstructed output from 2025 points to 2048 points same as ground truths.

From Table 8, losses performs the best in different cases, where CALoss can achieve the best or second best performances in most conditions. Note that it is normal for task networks to have better CD/EMD metrics when the they also use CD/EMD to train. Our method performs good on CD/EMD metrics without introducing any matching operations during training, where it always performs the best on MCD/HD metrics as shown in Table 1. It confirms that CALoss is a meaningful method.

A.4 Necessity of the Adversarial Strategy

To show the necessity of the adversarial strategy, we make an attempt to train the task network AE Achlioptas et al. (2018) with a pre-trained CALoss directly without further adversarial training. The results are demonstrated in Fig. 6 and Table 9. We can see that the task network trained with pre-trained CALoss can only reconstruct quite rough shapes, which confirms that the training with CALoss is actually a continuous procedure by searching shape differences with adversarial strategy.

Table 9: Quantitative comparisons between CALoss and pre-trained CALoss.

Metrics	MCD	HD	CD	EMD
Pre-trained	1.12	68.59	1.94	21.83
Ours	**0.21**	**1.53**	**0.26**	**8.47**

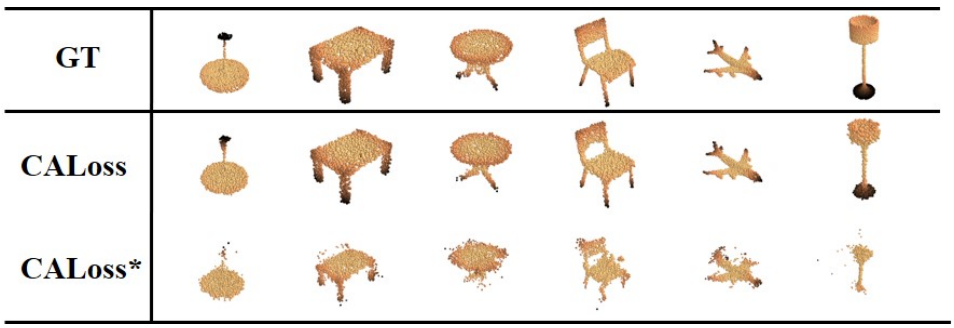

Figure 6: Comparisons with pre-trained CALoss. CALoss* denotes pre-trained CALoss.

A.5 TRAINING CURVES

In this section, we visualize the reconstruction errors measured with MCD/HD/CD/EMD metrics based on the AE network Achlioptas et al. (2018) and ShapeNet dataset Wu et al. (2015) through all the training iterations to observe the convergences of difference loss functions. We can see that our CALoss has relatively inferior performances at the beginning of iterations, where CALoss is learning to search shape differences. But it will converge steadily to low errors after enough iterations.

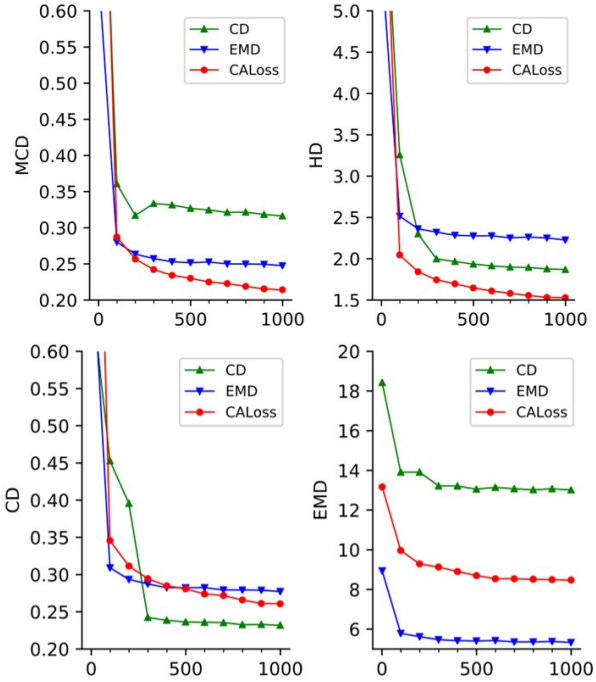

Figure 7: The reconstruction error curves through the iterations.

A.6 DISCUSSION ABOUT THE LIMITATION

The limitation of CALoss may lie in its relative behind performance at the beginning of training like shown in Fig. 7. As a network optimized together with the task network, CALoss needs iterations to learn to search shape differences, which may take more iterations for the task network to converge. This problem may be addressed by introducing an appropriate initial pre-trained CALoss and fine-tuning it later. We will focus on it in the future.

A.7 DISCUSSION ABOUT THE ADAPTIVE POOLING OPERATION

In this section, We present an ablation study for the proposed adaptive pooling operation. From

Metrics	MCD	HD	CD	EMD
Max pooling	0.34	1.72	0.64	23.10
Avg pooling	0.61	6.79	0.63	**7.28**
Ours	**0.21**	**1.53**	**0.26**	8.47

Table 10: Quantitative comparisons between max/average pooling and adaptive pooling.

Table 10, we can see that both max pooling and average pooling have quite inferior performances on most metrics, which prove the necessity of our adaptive pooling operation. In CALoss, the pooling operation is introduced 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 features are constrained. In this condition, we design such an

adaptive pooling operation to get a variable weight for each point according to their distance to the max pooled representation, which can be regarded as a "soft max pooling". All point features can be constrained with distinguishable gradients, which is controlled by the width of weights predicted with pooling controller $h(\cdot)$ as shown in Fig. 2 and Eq. 7.

A.8 More Qualitative Results

In this section, we present more qualitative results based on AE trained with different reconstruction losses. The results are presented in Fig. 8 and Fig. 9. We can see that CALoss still shows good performances to help the task network generate more uniform and complete shapes.

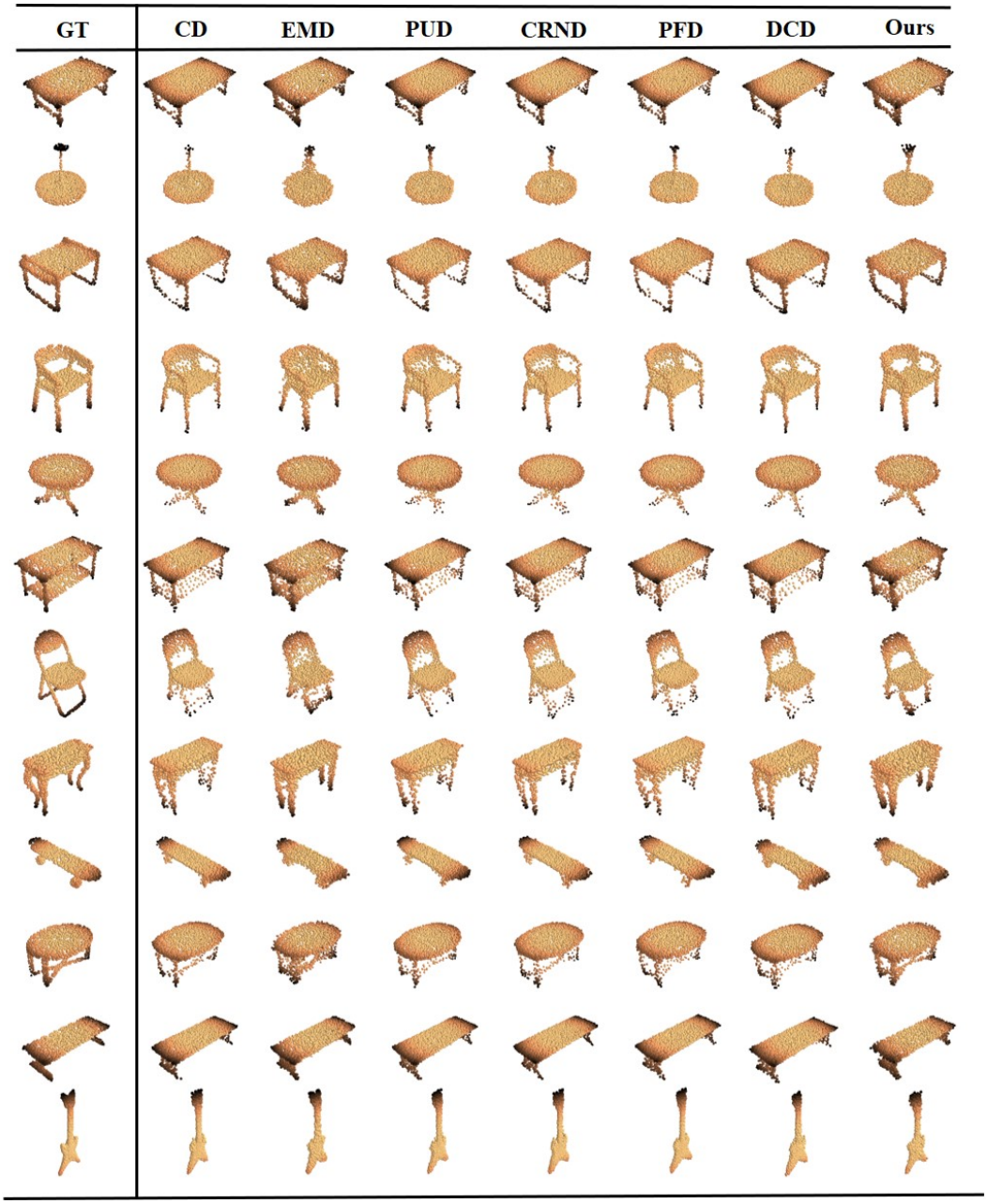

Figure 8: More qualitative comparisons with different reconstruction losses (part a).

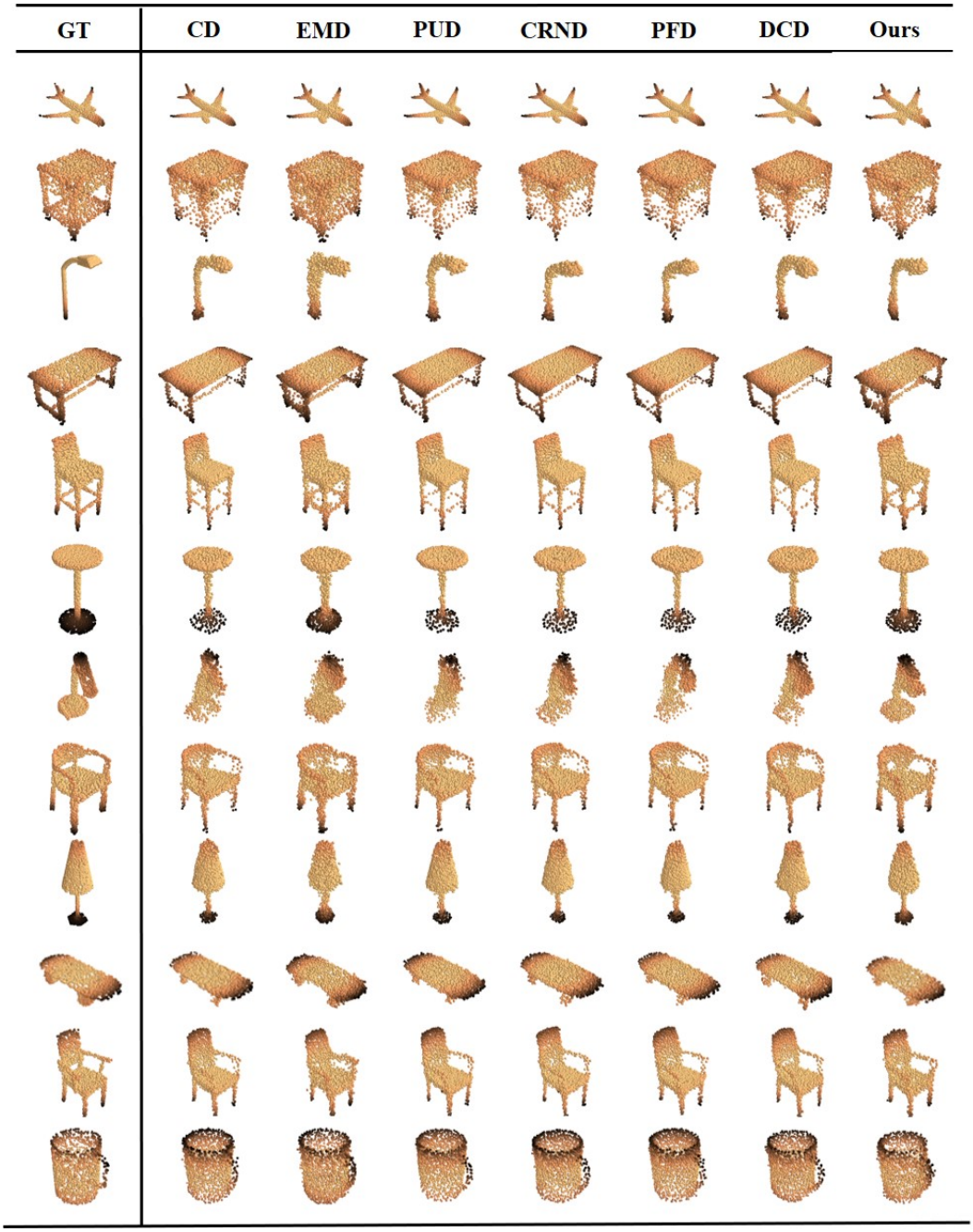

Figure 9: More qualitative comparisons with different reconstruction losses (part b).