Supplementary

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 1. The structures of networks are presented in Table 2. 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.

	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 1. Illustrations of hyper-parameters

2. Theoretical Analysis

Analysis. Here we present a simple theoretical argument by considering it as an approach to the Wasserstein distance. The Wasserstein distance between shapes, also known as the minimum transmission distance, may measure the true distance between point clouds. However, the accurate Wasserstein distance is too computational cost to calculate in applications. Existing matching-based reconstruction losses, including CD/EMD, actually work by approaching the Wasserstein distance by different manually-defined rules such as caculating the distances between points to their nearest neighbors in other point clouds. WGAN [1] and WGAN-GP [3] calculate the wasserstein distance between the distributions x and $g_\theta(z)$ by

$$\underset{\omega \in W}{max}\, \mathbb{E}_{x\sim\mathbb{P}_r}[f_\omega(x)] - \mathbb{E}_{z\sim p(z)}[f_\omega(g_\theta(z))], \tag{1}$$

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 [1,3] 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 of the paper.

The adversarial optimization of CALoss can then be described as

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

We can see that our first term can be regarded a symmetric form of Wasserstein distance [1], where the K-Lipschitz [1, 3] can be guaranteed 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 Wasserstein distances between point clouds, which may explain its effectiveness. CALoss does not have to describe the whole shape within a global representation. It works by dynamically searching and constraining the shape differences during the adversarial training.

Note that EMD loss mentioned in point cloud reconstruction is actually *NOT* the same as Wasserstein loss. Wasserstein loss can measure distances between different distributions, which is, however, inaccessible to directly calculate due to its high complexity and continuity. EMD reconstruction loss is an approximation of Wasserstein loss in a discrete way by matching points with the pre-defined optimization algorithm as discussed in [2, 4]. But the predefined algorithm may introduce criticism like limited performances or great time cost. Although our method can be confirmed as a “symmetric form of Wasserstein loss", it is another more accurate and efficient approximation for Wasserstein loss by replacing the pre-defined discrete point-to-point matching operations with dynamic searching process under a more continuous learned representation space.

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)	

Table 2. Illustrations of network structures. All components presented are MLPs.

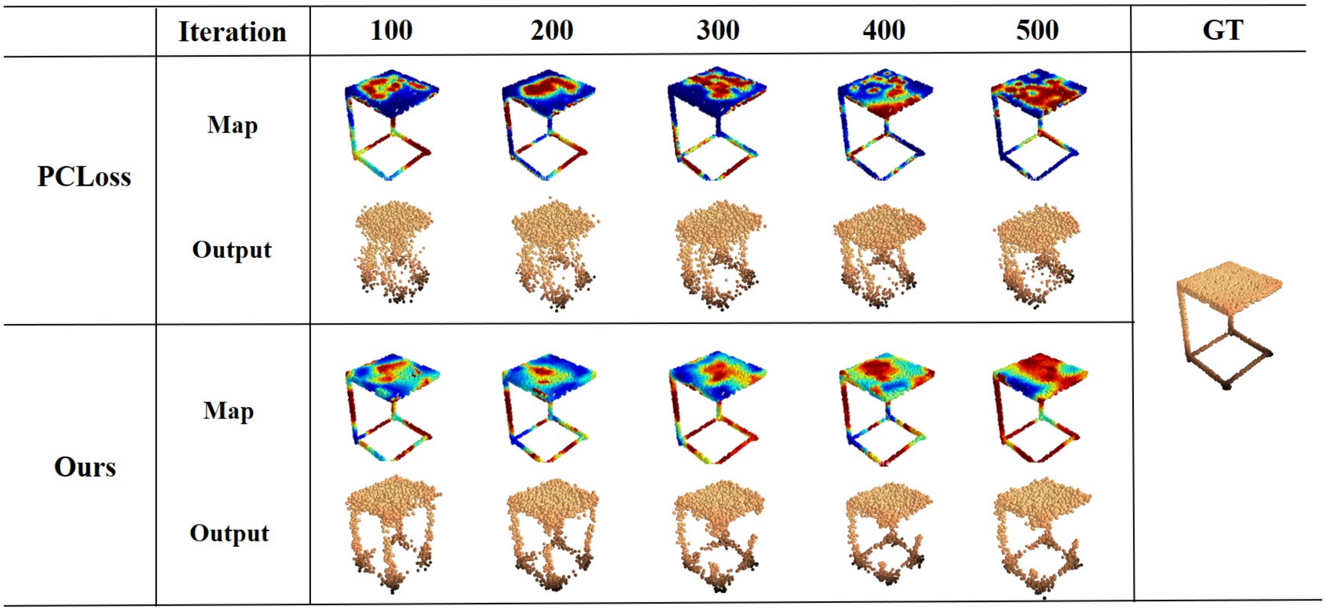

Figure 1. Visualization of the constraint differences during the training of learning-based PCLoss [4] and our method. Map and Output denote the attention map visualized with the weights of constrained points, and the trained task network output. Brighter colors near red means stronger constraints with larger weights.

3. Discussion about the limitation

Although CALoss achieves good performances, there is still limitations during its application. It has relatively weak performances at training beginning as it needs iterations to construct the representation space with shape similarity and learn to search shape differences. This problem may be addressed by constructing an appropriate pre-trained model for initialization. We will focus on it in the future.

4. Visualization about the Constrained Regions

To further explore why CALoss based on the representation space can outperform PCLoss [4] based on the 3D Euclidean space, we conduct a simple visualization here to observe their constrained regions during training iterations in Fig. 1. We can see that PCLoss [4] has discontinuous and relatively small constrained regions because it extracts descriptors around the predicted center points in 3D Euclidean space, where each center point provides a small constrained region. Our method can provide more continuous and wider constrained regions around the whole shapes benefited from the aggregation in representation space, where the leg parts receive more attentions to remove defects during iterations.

5. 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. 2 and Fig. 3. We can see that CALoss still shows good performances to help the task network generate more uniform and complete shapes.

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 1

[2] 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. 1

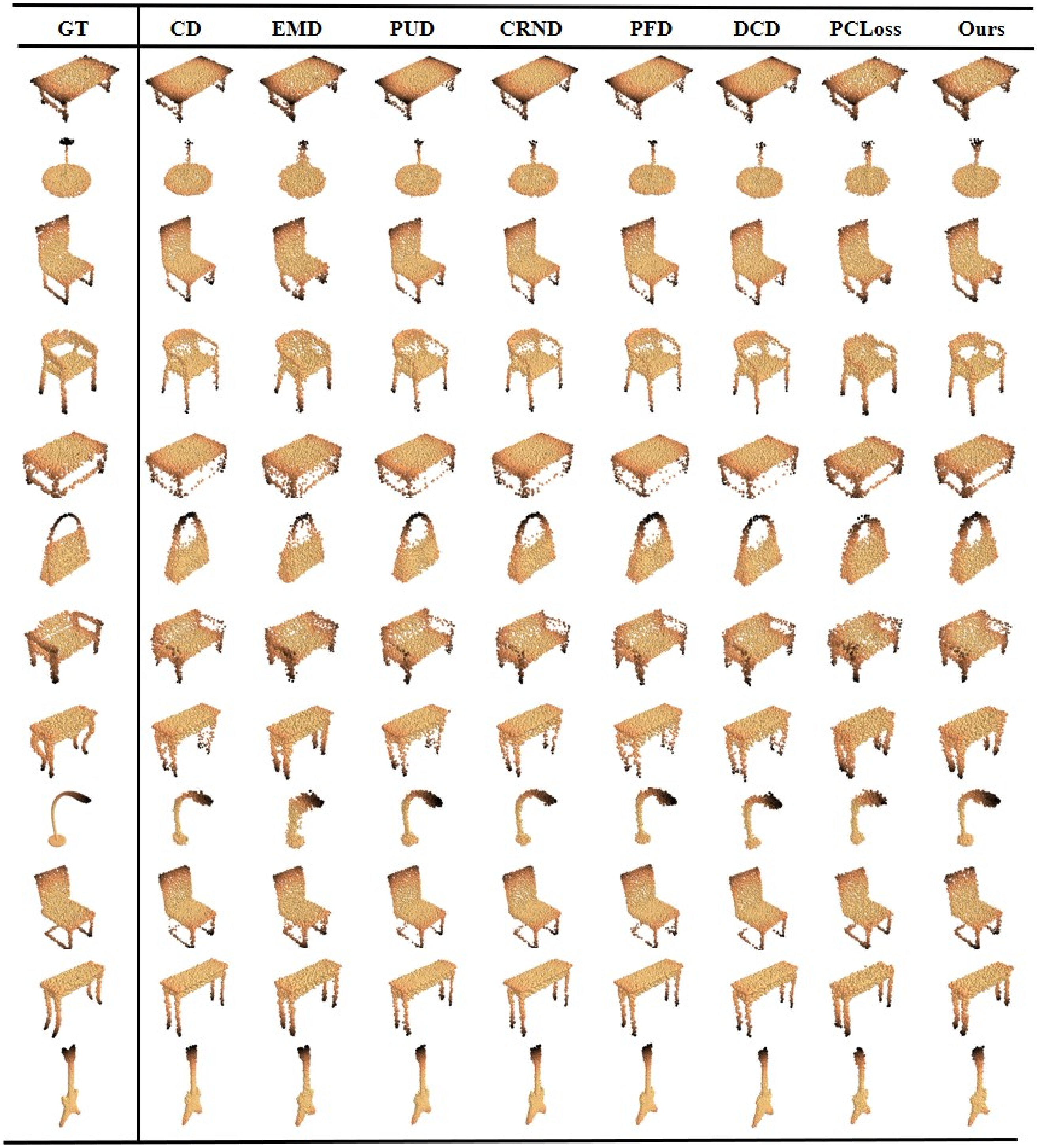

Figure 2. More qualitative comparisons with different reconstruction losses (part a).

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017. 1

[4] 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–

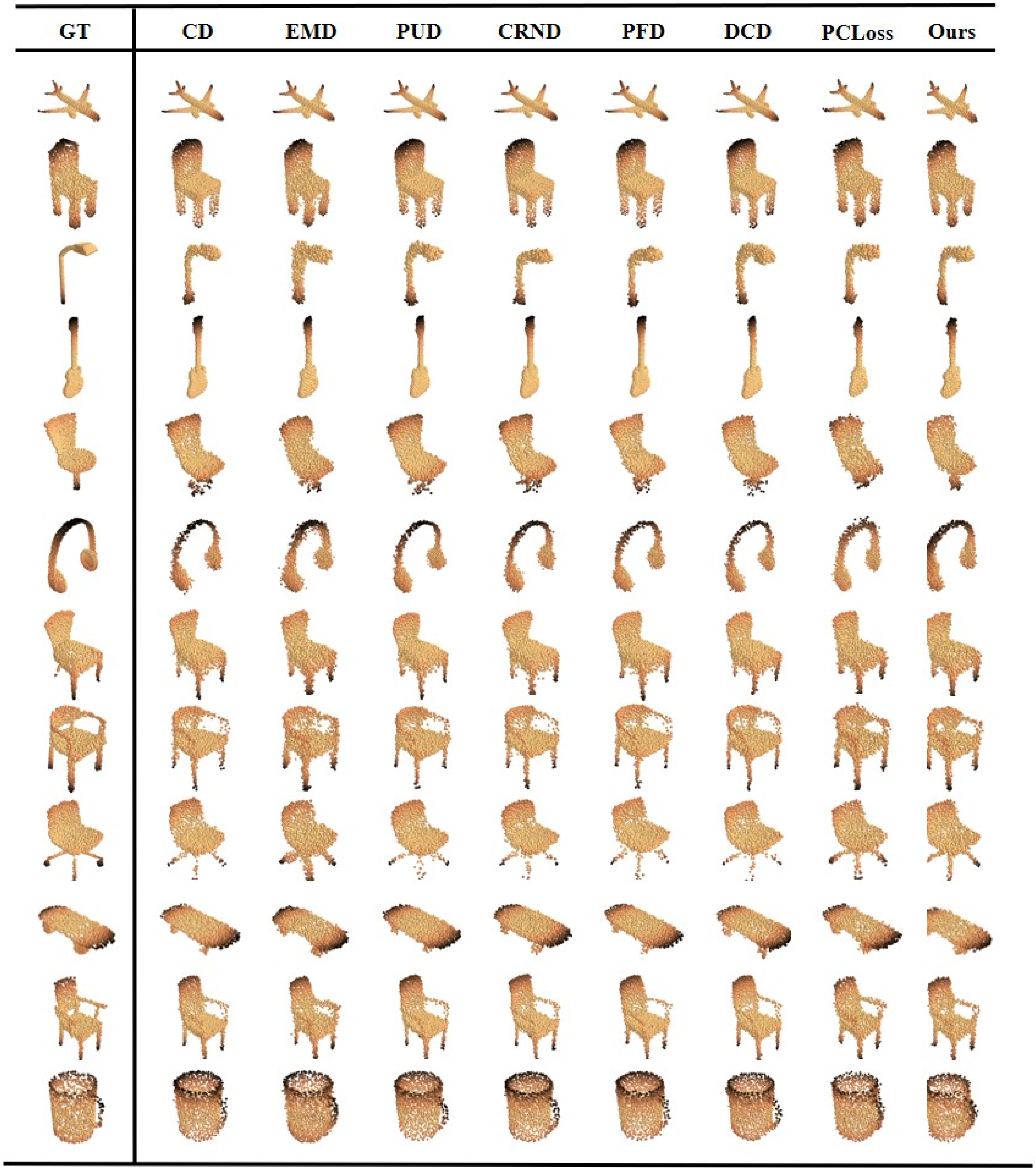

Figure 3. More qualitative comparisons with different reconstruction losses (part b).

194. Springer, 2022. 1, 2