Supplementary material: Multimodal Shape Completion via Conditoinal Generative Adversarial Networks

Rundi Wu[1⋆], Xuelin Chen[2⋆], Yixin Zhuang[1], and Baoquan Chen[1]

[1] Peking University
[2] Shandong University

A Overview

This supplementary material contains:

- implementation details of our network modules (Sec. B);
- details of the competing methods, which consists of the baseline methods and variants of our method (Sec. C).
- the data processing details of the three datasets used in the evaluation (Sec. D).
- details of the quantitative measures for the evaluation (Sec. E).

B Details of network modules

Table 1 shows the detailed architecture of our point set autoencoder(E_{AE}, D_{AE}) and latent conditional GAN(G and F). Note that we fuse the latent code of the input partial shape and a Gaussian-sampled condition $\mathbf{z}$ by direct channel-wise concatenation.

The variational autoencoder(E_{VAE}, D_{VAE}) shares the same architecture as the plain autoencoder (E_{AE}, D_{AE}), while having an extra FC layer at the bottleneck to squeeze the latent code to length $|\mathbf{z}| = 64$ and enabling the re-parameterization trick.

C Details of competing methods

In this section, we describe in detail the design of baseline methods and variants of our method (namely `KNN-latent`, `ours-im-l2z` and `ours-im-pc2z`) in the comparison experimetns.

`KNN-latent`. Given an input partial shape, we encode it into the latent space formed by our point set encoder E_{AE} and find its k-nearest neighbors based on cosine similarity.

`ours-im-l2z`. As a variant of our method, `ours-im-l2z` jointly trains the E_z to implicitly model the multimodality by mapping the complete latent code $\mathbf{x}_c$

⋆ Equal contribution

Pointnet Encoder $E_{\mathbf{AE}}$	
Layer	Output Shape
Input point set	(3, K)
Conv1D+BN+ReLU	(64, K)
Conv1D+BN+ReLU	(128, K)
Conv1D+BN+ReLU	(128, K)
Conv1D+BN+ReLU	(256, K)
Conv1D+BN+ReLU	(128, K)
Global Max Pooling	(128,)
Decoder $D_{\mathbf{AE}}$	
Layer	Output Shape
Latent code	(128,)
FC+ReLU	(256,)
FC+ReLU	(256,)
FC+ReLU	(2048×3,)
Reshape	(3, 2048)

Generator G	
Layer	Output Shape
Concat(Latent code, z)	(128+64,)
FC+lReLU	(256,)
FC+lReLU	(512,)
FC	(128,)
Discriminator F	
Layer	Output Shape
Latent code	(128,)
FC+lReLU	(256,)
FC+lReLU	(512,)
FC	(1,)

Table 1. Left: the architecture of our point set autoencoder. Right: the architecture of our latent conditional GAN. Conv1D: 1D convolution, BN: batch normalization, FC: fully connected layer, Concat: channel-wise concatenation. K is the number of points.

into a low-dimensional space. Hence, the latent space reconstruction loss $\mathcal{L}_{G,E_z}^{\text{latent}}$ becomes:

$$\mathcal{L}_{G,E_z}^{\text{latent}} = \mathbb{E}_{\mathbf{P}\sim p(\mathbf{P}),\mathbf{z}\sim p(\mathbf{z})}[\|\mathbf{z}, E_z(G(E_{\text{AE}}(\mathbf{P}),\mathbf{z}))\|_1].$$

In addition to the loss terms of Eq. 6 in the main paper, to allow stochastic sampling at test time, an additional Kullback-Leibler (KL) loss on the $\mathbf{z}$ space is introduced to force $E_z(\mathbf{x}_c)$ to be close to a Gaussian distribution:

$$\mathcal{L}_{E_z}^{\text{KL}} = \mathbb{E}_{\mathbf{x}_c\sim\mathbb{X}_c}[\mathcal{D}^{KL}(E_z(\mathbf{x}_c)||\mathcal{N}(0,1))]$$

where $\mathcal{D}^{KL}$ stands for Kullback-Leibler divergence. Hence the full training objective function becomes:

$$\underset{(G,E_z)}{\operatorname{argmin}}\ \underset{F}{\operatorname{argmax}}\ \mathcal{L}_F^{\text{GAN}} + \mathcal{L}_G^{\text{GAN}} + \alpha\mathcal{L}_G^{\text{recon}} + \beta\mathcal{L}_{G,E_z}^{\text{latent}} + \gamma\mathcal{L}_{E_z}^{\text{KL}} \tag{1}$$

The weight factors α and β are set to 6.0 and 7.5, same as those of our main model, and γ is set to 1.0.

`ours-im-pc2z`. As stated in the main paper, `ours-im-pc2z` takes complete point clouds as input to implicitly encode the multimodality. The full training objective function is the same as that (Eq. 1) of `ours-im-l2z`, while the KL loss term changes to:

$$\mathcal{L}_{E_z}^{\text{KL}} = \mathbb{E}_{\mathbf{x}_c\sim\mathbb{X}_c}[\mathcal{D}^{KL}(E_z(\mathbf{C})||\mathcal{N}(0,1))],$$

The architectures for `ours-im-l2z` and `ours-im-pc2z` are shown in Fig. 1.

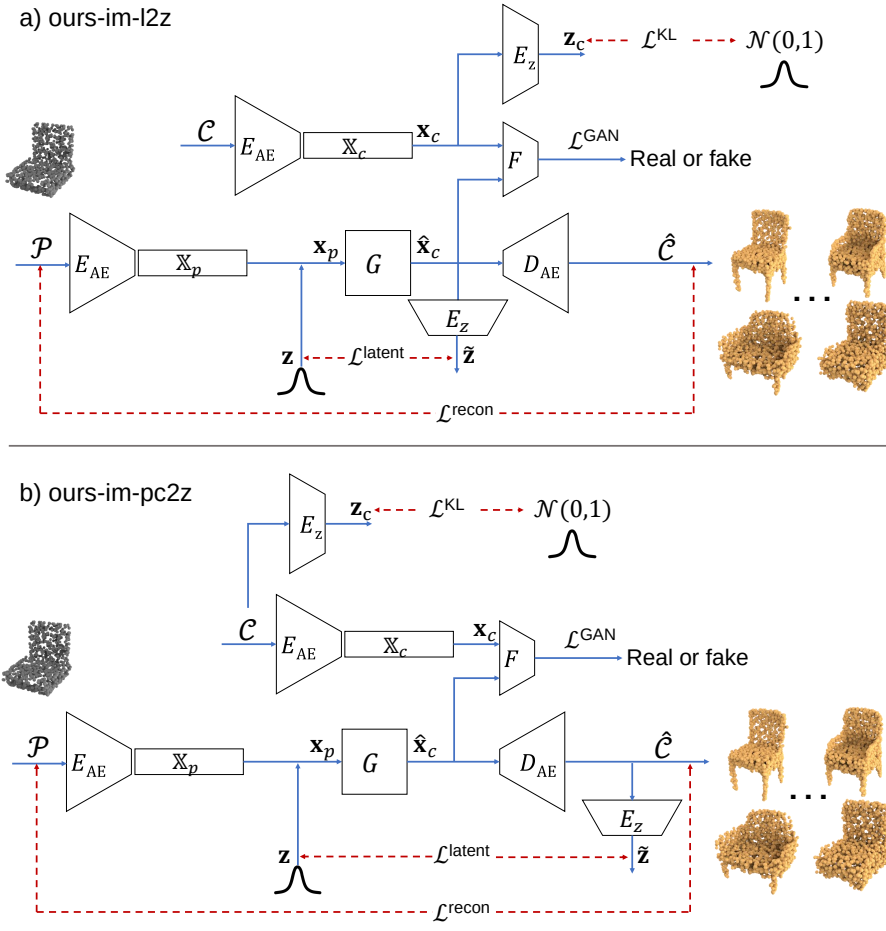

Fig. 1. Illustration for the two variants of our method: `ours-im-l2z` (top) and `ours-im-pc2z` (bottom).

D Details of data processing

In this section, we provide data processing details for our three datasets, especially the acquisition of the complete and partial point clouds in each dataset.

`PartNet`. Original PartNet dataset[3] provides point clouds sampled from shape mesh surface, along with semantic label for each point. The provided point clouds of the complete shape serves directly as our complete shape data. To create the partial shape data, for a shape with k parts, we randomly remove $j(1 \leq j \leq k-1)$ parts by checking the semantic labels for all of its points.

`PartNet-Scan`. To resemble the scenario where the partial scan suffers from part-level incompleteness, both the complete and partial point sets in `PartNet-Scan` are obtained from virtual scan. Complete point sets are acquired by virtually

scanning the complete shape mesh from 27 uniform views on the unit sphere. Partial point sets are acquired by virtually scanning the partial shape mesh from a single view that is randomly sampled. For each complete shape mesh in the original PartNet dataset[3], we create 4 partial shape meshes using the same principle as in `PartNet`. And for each partial shape mesh, we run the single-view scan to get 4 partial point sets from different views.

`3D-EPN`. The provided point cloud representation directly serves as the partial shape data. The complete shape data comes from the virtual scan of ShapeNet[2] objects from 36 uniformly sampled views.

E Evaluation measures

Here we explain in detail the quantitative measures that we adopt for evaluation, *i.e.* , Minimal Matching Distance (`MMD`), Total Mutal Difference (`TMD`), and Unidirectional Haudorff Distance (`UHD`). Given a test set of partial shapes $\mathcal{T}_p$ and a test set of complete shapes $\mathcal{T}_c$. For each shape p_i in $\mathcal{T}_p$, we generate k completed shapes c_{ij}, $j = 1...k$, resulting in a generated set $\mathcal{G}_c = \{c_{ij}\}$. We set $k = 10$ in all our quantitative evaluations.

`MMD` [1]. For each shape s_i in $\mathcal{T}_c$, we find its nearest neighbor $\mathbf{N}(s_i)$ in $\mathcal{G}_c$ by using Chamfer distance as the distance measure. `MMD` is then defined as

$$\texttt{MMD} = \frac{1}{|\mathcal{T}_c|} \sum_{s_i \in \mathcal{T}_c} d^{\mathrm{CD}}(s_i, \mathbf{N}(s_i)),$$

where d^{CD} stands for Chamfer distance.

`TMD`. For each of the k generated shapes $c_{ij} (1 \le j \le k)$ from the same partial shape $p_i \in \mathcal{T}_p$, we calculate its average Chamfer distance to the other $k-1$ shapes and sum up the resulting k distances. `TMD` is then defined as the average value over different input partial shapes in $\mathcal{T}_p$:

$$\begin{aligned}\texttt{TMD} &= \frac{1}{|\mathcal{T}_p|} \sum_{i=1}^{|\mathcal{T}_p|} \Big(\sum_{j=1}^{k} \frac{1}{k-1} \sum_{1 \le l \le k, l \ne j} d^{\mathrm{CD}}(c_{ij}, c_{il}) \Big) \\ &= \frac{1}{|\mathcal{T}_p|} \sum_{i=1}^{|\mathcal{T}_p|} \Big(\frac{2}{k-1} \sum_{j=1}^{k} \sum_{l=j+1}^{k} d^{\mathrm{CD}}(c_{ij}) \Big).\end{aligned}$$

`UMD`. We calculate the average unidirectional Hausdorff distance from the partial shape $p_i \in \mathcal{T}_p$ to each of its k completed shapes $c_{ij} (1 \le j \le k)$:

$$\texttt{UMD} = \frac{1}{|\mathcal{T}_p|} \sum_{i=1}^{|\mathcal{T}_p|} \Big(\frac{1}{k} \sum_{j=1}^{k} d^{\mathrm{HL}}(p_i, c_{ij}) \Big),$$

where d^{HL} stands for unidirectional Hausdorff distance.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3d point clouds. In: International Conference on Machine Learning (ICML). pp. 40–49 (2018)
2. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
3. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: PartNet: A large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)