PointRegGPT: Boosting 3D Point Cloud Registration using Generative Point-Cloud Pairs for Training (Supplementary Material)

Suyi Chen[1,4], Hao Xu[2], Haipeng Li[1,4], Kunming Luo[3], Guanghui Liu[1], Chi-Wing Fu[2], Ping Tan[3], and Shuaicheng Liu[1,4⋆]

[1] University of Electronic Science and Technology of China
[2] The Chinese University of Hong Kong
[3] Hong Kong University of Science and Technology
[4] Megvii Technology

1 Overview

To make our PointRegGPT self-contained, we provide additional details in this supplementary document, including:

- Implementation details.
- Evaluation metrics.
- Additional quantitative experiment results.
- More visualization.
- Limitations and future works.

2 Implementation Details

To train our diffusion model and depth correction module, the depth maps are resized to a resolution of 256×256, using downsampling of nearest interpolation and center-cropping. The depth range is clipped to $[0\mathrm{m}, 10\mathrm{m}]$ and then scaled to $[-1, 1]$. We train our diffusion model using 32 GeForce RTX 2080 Tis with a batch size of 128 for $2,000k$ iterations. For the forward process, we set $T = 1000$ and use the sigmoid schedule [12] for β_t. For the reverse process, following [16], we set $T = 250$ to accelerate the generation process. To train our depth correction module, we use 8 GeForce RTX 2080 Tis with a batch size of 32 for 50 epochs. We use the Adam optimizer with an initial learning rate of $1e^{-4}$, which is decayed by 0.95 for each epoch.

For the generation of point cloud registration datasets, the threshold τ_m is set to 0.99. Following [11], the generated point cloud is downsampled to a voxel size of $0.025\mathrm{m}$. To minimize domain gaps, we design a ground-truth-based transformation generation strategy, sampling ground-truth transformations from the 3DMatch training set with turbulence added for randomness and diversity. We also exploit a random-based transformation generation strategy to simulate point clouds from video frames:

⋆ Corresponding author.
[2] Department of Computer Science and Engineering; Institute of Medical Intelligence and XR.
[3] HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute.

rotations are randomly sampled from uniform distributions, $[-\frac{\pi}{24}, \frac{\pi}{24}]$ for pitch and $[-\frac{\pi}{12}, \frac{\pi}{12}]$ for yaw; translations are randomly sampled from a Gaussian distribution with $\sigma = \frac{1}{3}m$. Both of these two strategies can generate reliable transformations. The point-cloud pairs whose bidirectional overlap ratio is under 10% or have a point cloud with a point number under $1,000$ are abandoned. To evaluate the effectiveness of our generative dataset, we choose the previous state-of-the-art methods [11, 15, 20] as baselines, and train them using the generative dataset as an additional dataset, following all the training and testing protocols but increasing the total training epochs, *i.e.*, 60 epochs for PREDATOR [11]/GeoTrans [15] and 30 epochs for CoFiNet [20].

3 Evaluation Metrics

Following the existing works, we give the details on the evaluation metrics we employed for point cloud registration.

(i) **Registration Recall (RR)** is the fraction of successfully registered point-cloud pairs. A point-cloud pair is said to be successfully registered when its transformation error is lower than threshold τ_1. In addition, the transformation error is defined as the root mean square error of the ground-truth correspondences $\mathcal{C}^*$, to which the estimated transformation $\mathbf{T}_{est}(\cdot)$ has applied:

$$\mathrm{RMSE} = \sqrt{\frac{1}{|\mathcal{C}^*|} \sum_{(\mathbf{p}_x^*, \mathbf{q}_y^*) \in \mathcal{C}^*} \|\mathbf{T}_{est}(\mathbf{p}_x^*) - \mathbf{q}_y^*\|_2^2}, \tag{1}$$

$$\mathrm{RR} = \frac{1}{M} \sum_{i=1}^{M} \big[\mathrm{RMSE}_i < \tau_1\big], \tag{2}$$

where $\mathbf{p}_x$ and $\mathbf{q}_y$ denote the x-th point in source $\mathbf{P}$ and y-th point in target $\mathbf{Q}$, respectively; $[\cdot]$ is the Iversion bracket; and M is the number of all point-cloud pairs.

(ii) **Inlier Ratio (IR)** is the fraction of inlier correspondences among all hypothesized correspondences $\mathcal{C}$. A correspondence is regarded as an inlier if the distance between the two points is lower than a certain threshold τ_2 under the ground-truth transformation $\mathbf{T}_{gt}(\cdot)$:

$$\mathrm{IR} = \frac{1}{|\mathcal{C}|} \sum_{(\mathbf{p}_x, \mathbf{q}_y) \in \mathcal{C}} \big[\|\mathbf{T}_{gt}(\mathbf{p}_x) - \mathbf{q}_y\|_2 < \tau_2\big]. \tag{3}$$

(iii) **Feature Matching Recall (FMR)** is the fraction of point-cloud pairs whose IR $>$ threshold τ_3:

$$\mathrm{FMR} = \frac{1}{M} \sum_{i=1}^{M} \big[\mathrm{IR}_i > \tau_3\big]. \tag{4}$$

(iv) **Relative Rotation Error (RRE)** is the geodesic distance in degrees between the estimated and ground-truth rotation matrices $\mathbf{R}_{est}$ and $\mathbf{R}_{gt}$:

$$\mathrm{RRE} = \arccos\left(\frac{\mathrm{trace}(\mathbf{R}_{est}^T \cdot \mathbf{R}_{gt} - 1)}{2}\right). \tag{5}$$

(v) **Relative Translation Error (RTE)** is the Euclidean distance between estimated and ground-truth translation vectors $\mathbf{t}_{est}$ and $\mathbf{t}_{gt}$:

$$\mathrm{RTE} = \|\mathbf{t}_{est} - \mathbf{t}_{gt}\|_2. \tag{6}$$

(vi) **Transformation Recall (TR)** is the fraction of successfully registered point-cloud pairs, similar to RR but using a different definition of successful registration:

$$\mathrm{TR} = \frac{1}{M}\sum_{i=1}^{M}\big[\mathrm{RRE}_i < \tau_4 \text{ and } \mathrm{RTE}_i < \tau_5\big]. \tag{7}$$

TR is also called RR in [2, 15], while we follow [5, 17] to use TR to avoid ambiguity.

Following [3, 5, 11, 15, 19], we set $\tau_1 = 0.2\mathrm{m}$, $\tau_2 = 0.1\mathrm{m}$, $\tau_3 = 0.05$, $\tau_4 = 15^\circ$, and $\tau_5 = 0.3\mathrm{m}$ for evaluation on the 3DMatch and 3DLoMatch benchmarks. Following [5, 17], we set $\tau_1 = 0.5\mathrm{m}$, $\tau_2 = 0.2\mathrm{m}$, and $\tau_3 = 0.05$ for evaluation on the ETH benchmark.

Besides, we report Fréchet Inception Distance (FID) [8] and Minimal Matching Distance (MMD) [1] for realism evaluation. When calculating MMD scores, we exploit Chamfer Distance to measure the distance between point clouds.

4 Additional Quantitative Experiment Results

4.1 Additional Evaluation Results

We present additional experiments that evaluate our method. Note that we present these additional experiments because different existing methods use different evaluation metrics/approaches.

Table 1: Registration results on 3DMatch and 3DLoMatch following the metrics used in [14]. The best and second-best results are marked in **bold** and underlined. Compared with SIRA-PCR [5], our method achieves comparable results on 3DMatch and surpasses it in RR on 3DLoMatch.

Model	3DMatch			3DLoMatch		
	RRE ↓	RTE ↓	RR ↑	RRE ↓	RTE ↓	RR ↑
PREDATOR [11]	2.72	0.078	91.8	4.44	0.116	62,4
Lepard [14]	2.48	0.072	93.5	4.10	0.108	69.0
PREDATOR [11] + ICP	2.06	0.062	92.3	3.46	0.098	65.2
Lepard [14] + ICP	1.96	0.060	93.9	3.17	0.089	71.3
MTR [4]	**1.32**	**0.043**	95.1	**2.49**	**0.072**	75.4
GeoTrans [15]	1.84	0.061	94.1	2.86	0.086	76.1
+ SIRA-PCR [5]	1.80	0.059	**95.4**	2.74	0.084	79.4
+ Ours	1.87	0.061	95.0	2.78	0.086	**80.2**

Following [4, 14], we report the RRE, RTE, and RR which are averaged on all scan pairs, different from the more commonly-used scene-wise averages proposed by [11]. As shown in Tab. 1, boosted by our generative dataset, GeoTrans [15] achieves the RR

Table 2: Registration results on 3DMatch and 3DLoMatch following the metrics used in [2, 13, 15, 17, 21]. —: the results are not released. The best and second-best results are marked in **bold** and underlined. Compared with SIRA-PCR [5], our method achieves comparable results on 3DMatch and surpasses it in RR on 3DLoMatch.

Model	3DMatch			3DLoMatch		
	RRE ↓	RTE ↓	TR ↑	RRE ↓	RTE ↓	TR ↑
FGR [22]	2.82	8.36	78.6	5.28	12.98	20.0
DGR [7]	2.40	7.48	91.3	4.17	10.82	43.8
DHVR [13]	2.25	7.08	91.9	4.14	12.56	54.4
PointDSC [2]	2.06	6.55	93.3	3.87	10.39	56.1
SC^2PCR [6]	2.08	6.55	93.3	3.77	10.46	57.8
RoReg [17]	**1.84**	6.28	95.5	3.09	9.30	72.0
MAC [21]	—	—	95.7	—	—	78.9
GeoTrans [15]	1.98	5.69	95.0	2.98	8.55	77.5
+ SIRA-PCR [5]	1.91	**5.66**	**97.0**	**2.88**	**8.48**	80.8
+ Ours	1.96	5.78	96.5	2.94	8.63	**81.6**

comparable with current state-of-the-art methods MTR [18] and SIRA-PCR [5] on the 3DMatch benchmark, and even outperforms them on the 3DLoMatch benchmark by 4.8 pp and 0.8 pp, respectively. Note that it is meaningless to compare the RRE and RTE when RR or TR is not comparable, as RRE and RTE are computed within successfully registered pairs, so they are not discussed in this section.

To fairly compare with [2, 6, 7, 13, 21, 22], we report the TR, which is defined under the threshold of RRE $< 15^\circ$ and RTE $< 0.3\mathrm{m}$. As shown in Tab. 2, GeoTrans outperforms the current state-of-the-art method RoReg [17] and MAC [21] on both 3DMatch and 3DLoMatch. Compared with SIRA-PCR [5], our method achieves comparable results on 3DMatch and surpasses it in RR on 3DLoMatch by 0.8 pp.

The additional evaluation results in Tab. 1 and Tab. 2 show the robust boost under different metrics. In summary, without additional data sources as used in rendering-based dataset [5], our method achieves comparable results on 3DMatch and better results on 3DLoMatch, boosting GeoTrans [15] to outperform the current state-of-the-art methods.

Table 3: Comparison of transformation generation strategies

Methods	3DMatch			3DLoMatch		
	FMR ↑	IR ↑	RR ↑	FMR ↑	IR ↑	RR ↑
baseline (GeoTrans)	97.8	69.2	91.4	88.0	42.3	73.5
random-based strategy	**98.0**	70.9	91.7	88.9	44.8	75.5
ground-truth-based strategy	**98.0**	**71.6**	**91.9**	**89.4**	**44.9**	**76.5**

For transformation sampling, we design two different strategies, a ground-truth-based strategy and a random-based strategy. As shown in Tab. 3, the ground-truth-based strategy performs better since fewer domain gaps will be introduced in this way.

Table 4: Comparison in data augmentation

Methods	3DMatch			3DLoMatch		
	FMR ↑	IR ↑	RR ↑	FMR ↑	IR ↑	RR ↑
baseline (GeoTrans)	97.8	69.2	91.4	88.0	42.3	73.5
UDGE [9]	97.5	68.2	90.4	84.8	39.4	70.3
Ours	**98.0**	**71.6**	**91.9**	**89.4**	**44.9**	**76.5**

Our PointRegGPT can be deemed as a data augmentation method. From such a perspective, we compare it with another similar approach, UDGE [9]. UDGE [9] is for transfer learning but performs limited when it is used for data augmentation, since no new geometric structure is generated by only augmenting existing data, as is shown in Tab. 4. Our method can generate new geometric structures through diffusion models, thus effectively boosting the performances of baseline methods.

In Tab. 5, we compare our depth correction module with other possible solutions for the point penetration problem. TSDF can fuse tens of depth maps to densify a point cloud to an extremely high density. With such a density of point clouds, the point penetration problem will disappear during re-projection, thus generating datasets without artifacts, boosting the performance consistently as shown in Row (c) of Tab. 5. Nevertheless, the corresponding extrinsic matrix is required for each depth map, and the process of TSDF is really time-consuming. A handcrafted filter is proposed in [10] to tackle the point penetration problem. It works well in many easy cases but fails in challenging cases, which generates unreliable training data and causes a performance drop on 3DLoMatch. On the contrary, our depth correction module can solve challenging cases to stabilize the generative results by avoiding artifact generation.

As the total training epochs are increased when the baselines [11, 15, 20] are trained with the additional generative dataset, we also try increasing the total epochs to the same numbers when the baselines are trained on 3DMatch without the additional data, but no improvement is observed, which means simply increasing training epochs without adding more data is noneffective.

4.2 Time Consumption

Adhering to the official implementations without any modifications, the time consumption for 3D point cloud registration remains consistent across all baselines. For data generation, each pair of point clouds (out of 100 pairs tested) takes an average of $252.75\mathrm{ms}$ for depth correction and $7,889.57\mathrm{ms}$ for the reverse diffusion process. These timing measurements were rigorously validated, using a single GeForce RTX 2080 Ti for testing purposes.

Table 5: Ablation studies of different methods to solve the point penetration problem. The best and second-best results are marked in **bold** and underlined.

Methods	3DMatch			3DLoMatch		
	FMR ↑	IR ↑	RR ↑	FMR ↑	IR ↑	RR ↑
(a) baseline [15]	97.7	70.3	91.5	88.1	43.3	74.0
(b) w/o depth correction	**98.5**	70.6	91.3	87.8	42.6	73.5
(c) w/ TSDF	98.4	**71.6**	**92.8**	88.9	44.8	75.5
(d) w/ handcrafted filter [10]	97.7	71.5	91.6	88.0	43.4	71.9
(e) w/ depth correction	98.0	**71.6**	91.9	**89.4**	**44.9**	**76.5**

4.3 Realism Measurement

To quantitatively measure realism, we calculate FID and MMD scores between synthetic and real data. Tab. 6 shows that our generated data is more realistic than the rendering-based SIRA-PCR [5].

Table 6: Realism comparison with rendering-based data

Datasets	Type	FID ↓	MMD (cm) ↓
SIRA-PCR [5]	Rendering-based	7.21	12.49
Ours	Generative	**3.47**	**8.21**

4.4 Overlap Ratio Distribution

Figure 1 shows the overlap ratio distribution of our data, which indicates the diversity of transformations and overlap ratios in our generative dataset.

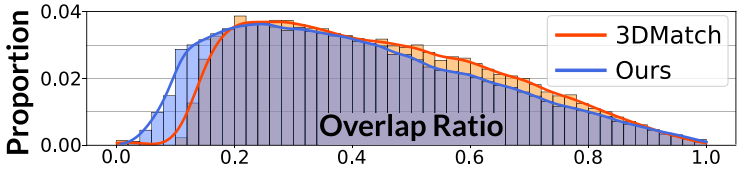

Fig. 1: Visualization of the overlap ratio distribution

5 More Visualization

5.1 Qualitative Comparison with Various Baselines

We provide more qualitative results with various baselines [11, 15, 20] in Fig. 2, Fig. 3, and Fig. 4, respectively. All of these are challenging cases on 3D(Lo)Match and ETH. Specifically, the overlapped structures are highly fragmented, *e.g.*, Case (b) and Case (c)

in Fig. 2, as well as Case (b) and Case (d) in Fig. 3, which are hard to recognize. Besides, repeated structures with ambiguity can cause failure cases, *e.g.*, Case (b), Case (c), and Case (d) in Fig. 4. Plus, extreme noises are made up of the luxuriant leaves of bushes and woods in outdoor scans, *e.g.*, Case (f) in both Fig. 3 and Fig. 4, encompassing and blurring the meaningful structures which are helpful for registration. Boosted by our PointRegGPT, these cases can be properly solved by baseline methods [11, 15, 20].

5.2 Qualitative Comparison of Depth Correction

We provide more qualitative results for depth correction. As shown in Fig. 5, hand-crafted filter [10] can solve part of the point penetration problem but fails in many difficult cases. Our deep correction module can solve the problem better by a learning-based approach. However, the vanilla U-Net cannot solve the problem perfectly, leaving some wrong values when the threshold is low, *i.e.*, $\tau_m = 0.50$, or eliminating too many right values to break the integrity of structures when the threshold is high, *i.e.*, $\tau_m = 0.99$, due to the lack of awareness of 3D structures. Our depth augmentation enhances the ability of the depth correction module, improving the performance. Whether the depth correction module is with or without the depth augmentation module, the wrong values caused by the penetrated points can be better eliminated with a higher threshold τ_m. Thus, we set $\tau_m = 0.99$ for better generated results.

5.3 Visualization of Generated Data

In this section, Fig. 6 provides a visual illustration of the data generated in our study. We successfully produced diverse content in the non-overlapping regions of the point clouds, while ensuring 3D consistency in the overlapping regions. For a comparison, we selected the nearest real-world sample from 3DMatch. The source point cloud in 3DMatch is constructed using a TSDF fusion from multiple depth maps, including the specific depth map that corresponds to the source point cloud in our generated samples.

6 Limitations and Future Works

Our PointRegGPT can rigorously guarantee 3D consistency and generation quality in overlap regions. However, due to indeterminacy during the reverse process of diffusion models, the generated results out of overlap regions are unpredictable, which causes unstable quality of generative data. Some generated cases are textureless or without meaningful structures, which may have a negative effect on models trained on them. A filter rejecting the bad cases in generative datasets could make a difference. Besides, as both the depth generation and the depth correction use a similar U-Net-like backbone, it is possible to reformulate the two parts into a one-step process. We will leave both of them for future work.

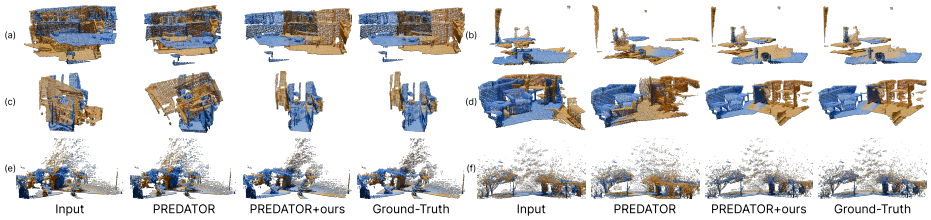

Fig. 2: Qualitative comparison on (a), (b) 3DMatch, (c), (d) 3DLoMatch, and (e), (f) ETH. As can be seen, our PointRegGPT can make PREDATOR [11] perform better in these challenging cases, *e.g.*, highly fragmented structures in (b) and (c).

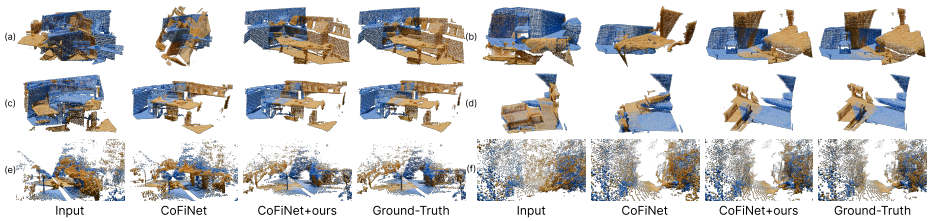

Fig. 3: Qualitative comparison on (a), (b) 3DMatch, (c), (d) 3DLoMatch, and (e), (f) ETH. As can be seen, the performance of CoFiNet [20] in these challenging cases, *e.g.*, highly fragmented structures in (b) and (d), as well as extreme noises in (f), is enhanced by our PointRegGPT.

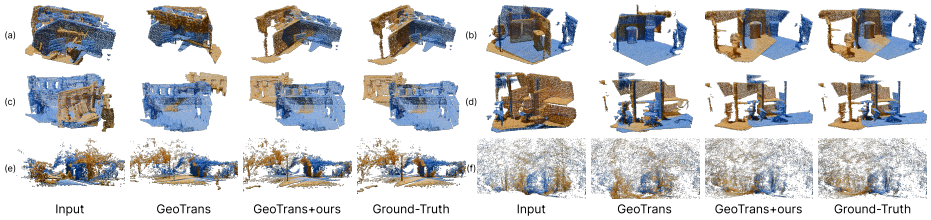

Fig. 4: Qualitative comparison on (a), (b) 3DMatch, (c), (d) 3DLoMatch, and (e), (f) ETH. As can be seen, GeoTrans [15] is boosted by our PointRegGPT to solve many challenging cases, *e.g.*, repeated structures with ambiguity in (b-d), and extreme noises in (f).

Fig. 5: Supplementary comparison for depth correction with different settings. As shown in the figure, the handcrafted filter [10] fails in challenging cases. Besides, under different thresholds τ_m, the depth correction module with the depth augmentation module outperforms the one without the depth augmentation module. Additionally, the depth correction module with a higher threshold, *i.e.*, $\tau_m = 0.99$, can eliminate the wrong values caused by the penetrated points better than a lower threshold, *i.e.*, $\tau_m = 0.5$.

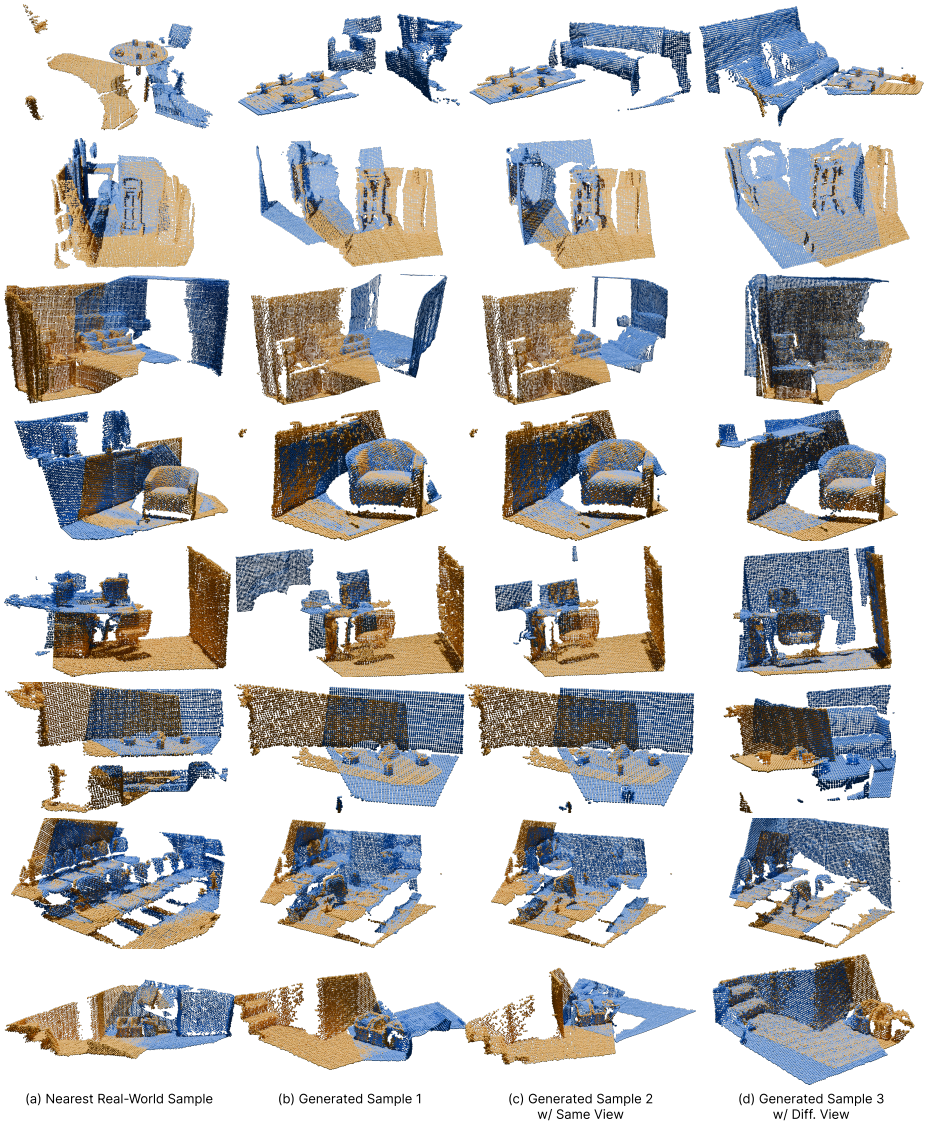

Fig. 6: Visual illustration of the generated data: the source point clouds (yellow) and the target/generated point clouds (blue). (a) Nearest real-world sample from 3DMatch, created by TSDF fusion from multiple depth maps. (b) Generated sample 1, the source point cloud of which is converted from one of the depth maps used in the nearest real-world sample. (c) Generated sample 2, obtained under the same view as the generated sample 1. (d) Generated sample 3, obtained under a view different from the generated sample 1.

References

1. Achlioptas, P., Diamanti, O., Mitliagkas, I., Guibas, L.: Learning representations and generative models for 3D point clouds. In: ICML. pp. 40–49. PMLR (2018)
2. Bai, X., Luo, Z., Zhou, L., Chen, H., Li, L., Hu, Z., Fu, H., Tai, C.L.: PointDSC: Robust point cloud registration using deep spatial consistency. In: CVPR. pp. 15859–15869 (2021)
3. Bai, X., Luo, Z., Zhou, L., Fu, H., Quan, L., Tai, C.L.: D3Feat: Joint learning of dense detection and description of 3D local features. In: CVPR. pp. 6359–6367 (2020)
4. Chen, G., Wang, M., Yuan, L., Yang, Y., Yue, Y.: Rethinking point cloud registration as masking and reconstruction. In: ICCV. pp. 17717–17727 (October 2023)
5. Chen, S., Xu, H., Li, R., Liu, G., Fu, C.W., Liu, S.: SIRA-PCR: Sim-to-real adaptation for 3D point cloud registration. In: ICCV. pp. 14394–14405 (October 2023)
6. Chen, Z., Sun, K., Yang, F., Tao, W.: SC2-PCR: A second order spatial compatibility for efficient and robust point cloud registration. In: CVPR. pp. 13221–13231 (2022)
7. Choy, C., Dong, W., Koltun, V.: Deep global registration. In: CVPR. pp. 2514–2523 (2020)
8. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local nash equilibrium. NeurIPS **30**, 1–12 (2017)
9. Horache, S., Deschaud, J.E., Goulette, F.: 3D point cloud registration with multi-scale architecture and unsupervised transfer learning. In: 3DV. pp. 1351–1361. IEEE (2021)
10. Hsu, C., Sun, C., Chen, H.: Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. CoRR **abs/2106.10859** (2021), `https://arxiv.org/abs/2106.10859`
11. Huang, S., Gojcic, Z., Usvyatsov, M., Wieser, A., Schindler, K.: PREDATOR: Registration of 3D point clouds with low overlap. arXiv:2011.13005 (2020)
12. Jabri, A., Fleet, D., Chen, T.: Scalable adaptive computation for iterative generation. arXiv preprint arXiv:2212.11972 (2022)
13. Lee, J., Kim, S., Cho, M., Park, J.: Deep hough voting for robust global registration. In: CVPR. pp. 15994–16003 (2021)
14. Li, Y., Harada, T.: Lepard: Learning partial point cloud matching in rigid and deformable scenes. CVPR (2022)
15. Qin, Z., Yu, H., Wang, C., Guo, Y., Peng, Y., Xu, K.: Geometric transformer for fast and robust point cloud registration. In: CVPR. pp. 11143–11152 (June 2022)
16. Song, J., Meng, C., Ermon, S.: Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502 (2020)
17. Wang, H., Liu, Y., Hu, Q., Wang, B., Chen, J., Dong, Z., Guo, Y., Wang, W., Yang, B.: RoReg: Pairwise point cloud registration with oriented descriptors and local rotations. IEEE TPAMI (2023)
18. Wang, H., Jiang, Z., Yi, L., Mo, K., Su, H., Guibas, L.J.: Rethinking sampling in 3D point cloud generative adversarial networks. arXiv preprint arXiv:2006.07029 (2020)
19. Yew, Z.J., Lee, G.H.: REGTR: End-to-end point cloud correspondences with transformers. In: CVPR. pp. 6677–6686 (2022)
20. Yu, H., Li, F., Saleh, M., Busam, B., Ilic, S.: CoFiNet: Reliable coarse-to-fine correspondences for robust point cloud registration. NeurIPS **34**, 23872–23884 (2021)
21. Zhang, X., Yang, J., Zhang, S., Zhang, Y.: 3D registration with maximal cliques. In: CVPR. pp. 17745–17754 (June 2023)
22. Zhou, Q.Y., Park, J., Koltun, V.: Fast global registration. In: ECCV. pp. 766–782 (2016)