SPU-PMD: Self-Supervised Point Cloud Upsampling via Progressive Mesh Deformation

Supplementary Material

6. Appendix

In this supplementary material, we first present more visual results of different models. Then, we provide more ablation studies to indicate the relationship between the proposed mesh interpolation algorithm and the learning framework. The setting of several essential parameters is also discussed. Finally, we conduct experiments for surface reconstruction, mesh deformation, and robustness evaluation to further exploit the properties of our network.

6.1. Comparison with SoTA methods

More visual comparisons with SoTA methods are shown in Fig. 10. We employ objects with complicated structures to bear out the superiority of SPU-PMD in preserving meticulous details, such as the chair grille and helical structure. From this figure, we can discern that most SoTA approaches result in obvious distortion in the upsampling process.

6.2. Ablation study

Mesh interpolation and SPU-PMD. To demonstrate the relationship between the mesh interpolation algorithm and the complete learning framework, we compare the upsampling results produced by solely adopting mesh interpolation and SPU-PMD. As shown in Table. 5, the SPU-PMD model that combines mesh interpolation with learnable deformation significantly outperforms the mesh interpolation algorithm in all evaluation metrics. From Fig. 8, it can be seen that SPU-PMD provides the upsampled point clouds with more uniform distributions and fewer unexpected holes. These experiments illustrate that mesh interpolation is an auxiliary part rather than the dominant component in SPU-PMD.

Table 5. Ablation study of deep learning structure.

Input	Mesh Interpolation		SPU-PMD	
	CD ↓ 10^{-3}	HD ↓ 10^{-3}	CD ↓ 10^{-3}	HD ↓ 10^{-3}
Sparse (512)	2.515	17.206	**1.690**	**13.568**
Medium (1,024)	1.496	11.174	**0.892**	**8.252**
Dense (2,048)	1.035	7.610	**0.544**	**4.926**

Moving Radius. We regulate the extent of deformation by specifying the moving radius in different deformation steps. In other words, reducing the radius brings about more delicate mesh deformation. To analyze the effect of moving radius, we select four radius settings. At first, we utilize two constant radii R_1 and R_2 that are set to 0.1 and 0.01 respectively. Besides, two varying settings R_3 and R_4 are employed. In R_3, we set $[0.1, 0.08]$ for deformation A and $[0.03, 0.02]$ for deformation B. In comparison, the baseline setting (R_4) is determined as $[0.1, 0.08]$ and $[0.02, 0.01]$. Under the numerical analysis shown in Table. 6, the models with varying radius settings have better performance than the counterparts with constant radii. The corresponding visual results are exhibited in Fig. 9, which demonstrates that a large and constant radius leads to blurred details and more noisy points. This study verifies that varying radii for different deformation steps are more helpful for SPU-PMD to govern the mesh variation and deformation B demands minor discrepancy.

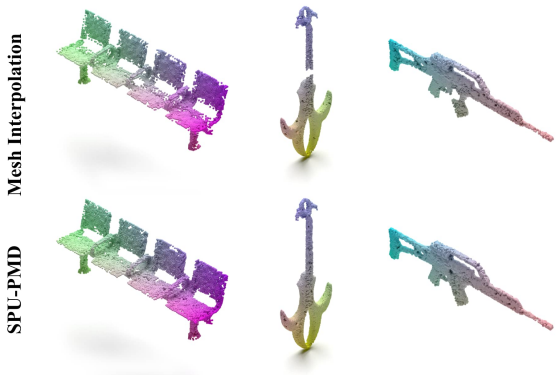

Figure 8. Comparison of mesh interpolation and SPU-PMD. The mesh interpolation method produces the results with fragmented structures. In contrast, SPU-PMD provides complete and uniform results.

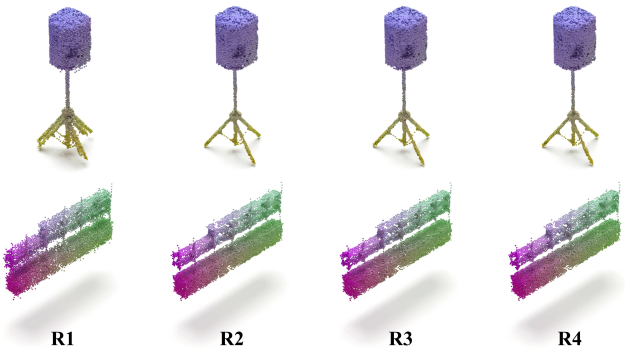

Figure 9. Ablation study of moving radius. Because the moving radius $R1$ is large, the results are more uniform, but there are more noise points and blurred structures.

Figure 10. More visual comparisons with other methods. These results demonstrate the advantages of our method in detailed structure preservation.

Table 6. Ablation Study of moving radius.

Radius	Sparse (512) input			Medium (1,024) input			Dense (2,048) input		
	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}
R_1	3.306	33.570	26.345	2.209	23.412	19.592	1.647	17.076	14.458
R_2	1.705	13.789	4.156	0.887	8.262	2.809	0.553	5.027	1.889
R_3	**1.688**	13.595	4.271	**0.884**	8.352	2.915	0.548	4.954	1.984
R_4	1.690	**13.568**	**4.082**	0.892	**8.252**	**2.765**	**0.544**	**4.926**	**1.861**

Table 7. Ablation study of query radius.

Radius	Sparse (512) input			Medium (1,024) input			Dense (2,048) input		
	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}	CD ↓ 10^{-3}	HD ↓ 10^{-3}	P2F ↓ 10^{-3}
Q_1	2.935	19.098	**2.688**	1.718	12.073	**1.720**	1.281	8.743	**1.065**
Q_2	2.800	18.356	2.739	1.581	11.490	1.760	1.152	8.062	1.091
Q_3	**2.479**	18.161	3.013	**1.465**	11.889	1.914	1.045	8.166	1.168
Q_4	2.515	**17.206**	2.949	1.496	**11.174**	1.876	**1.035**	**7.610**	1.162

Query Radius. Mesh construction is an essential step for our mesh interpolation, which is achieved by using the ball query. As a crucial parameter, the ball radius influences the mesh construction. In this study, we define all ball query radii based on the average distance of the nearest neighbors, denoted as r. Specifically, the first query radius Q_1 is constant, set as r, while the other three are with varying radii (Q_2: $[r, 0.8r]$, Q_3: $[r, 1.8r]$, and Q_4: $[r, 1.3r]$). Notably,

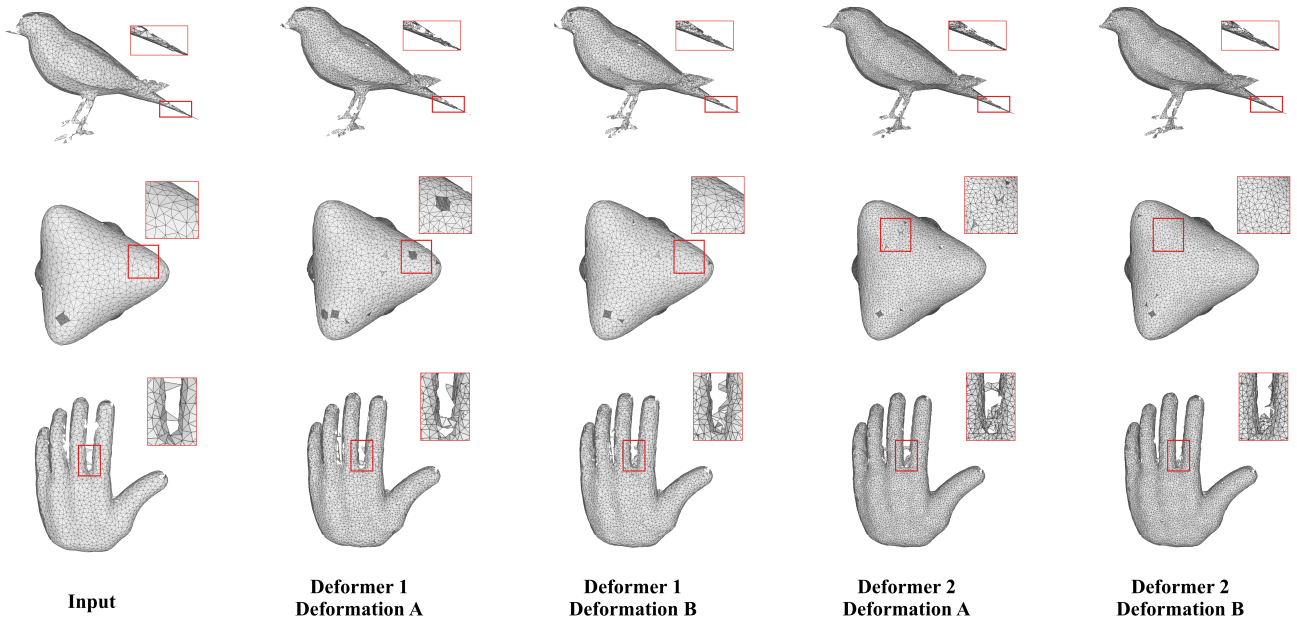

Figure 11. Visualization of mesh deformation.

this study is conducted by only applying mesh interpolation rather than the complete framework for avoiding potential effects from training. From Table. 7, we find out that the query radius within a certain range, like Q_4, results in better performance. Meanwhile, a large query radius leads to obvious degradation (like the results of Q_3). In addition, simply using the average distance as the query radius has great potential to generate points close to the underlying surface. The visual comparisons of the models with different query ball settings are presented in Fig. 12, which indicates that an appropriate query setting enables the mesh interpolation algorithm to generate more compact shapes.

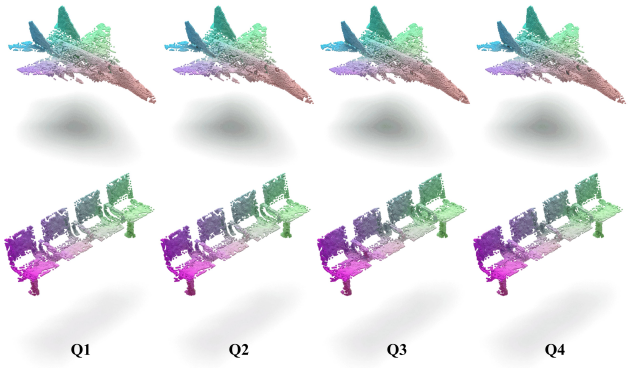

Figure 12. Ablation study of query radius. The smoothness and compactness of the results provided by the models with Q3 and Q4 are better than the others.

6.3. Surface reconstruction

We use the Poisson surface reconstruction algorithm [11] to create watertight surfaces from the results of our model, as shown in Fig. 13. The surfaces reconstructed from the upsampled point clouds are clean and smooth, indicating the smoothness and cleanness of our results. The results also demonstrate that SPU-PMD can simultaneously maintain complex structures and generate high-quality upsampling point clouds.

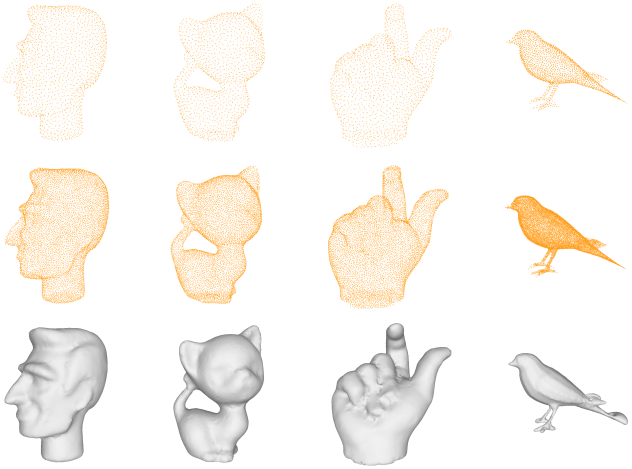

Figure 13. Visualization of surface reconstruction. The results demonstrate that the point clouds upsampled by SPU-PMD are smooth and clean.

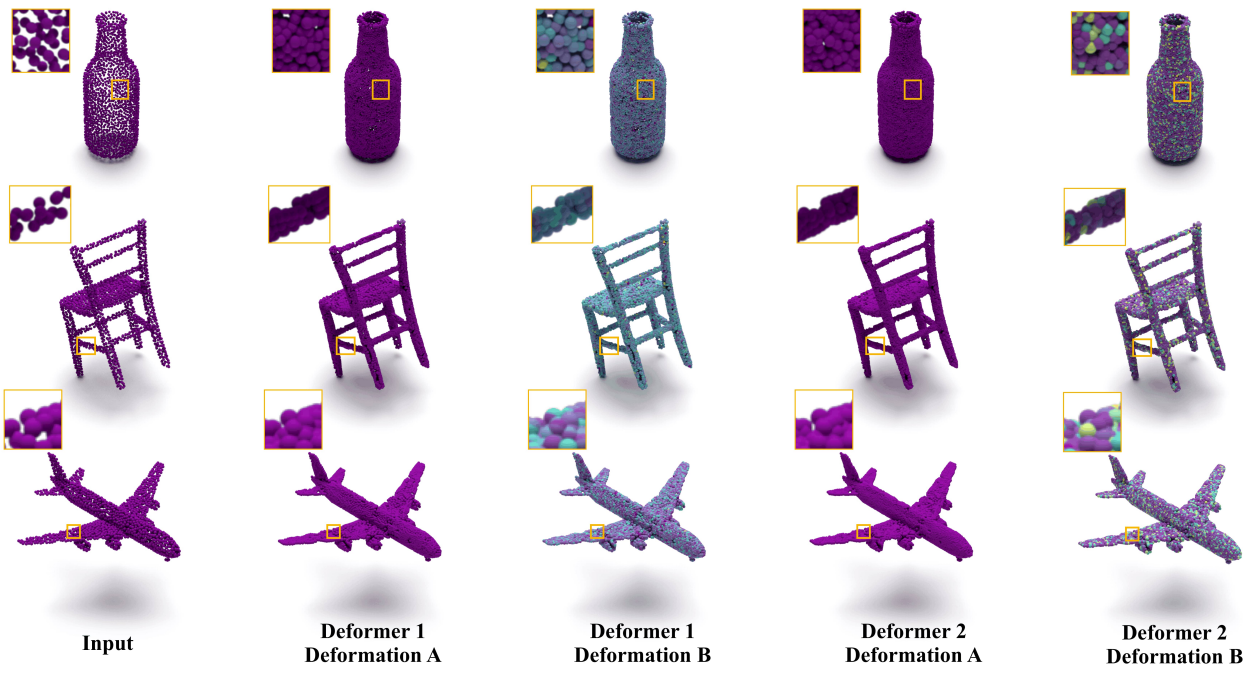

Figure 14. Visualization of mesh node motion.

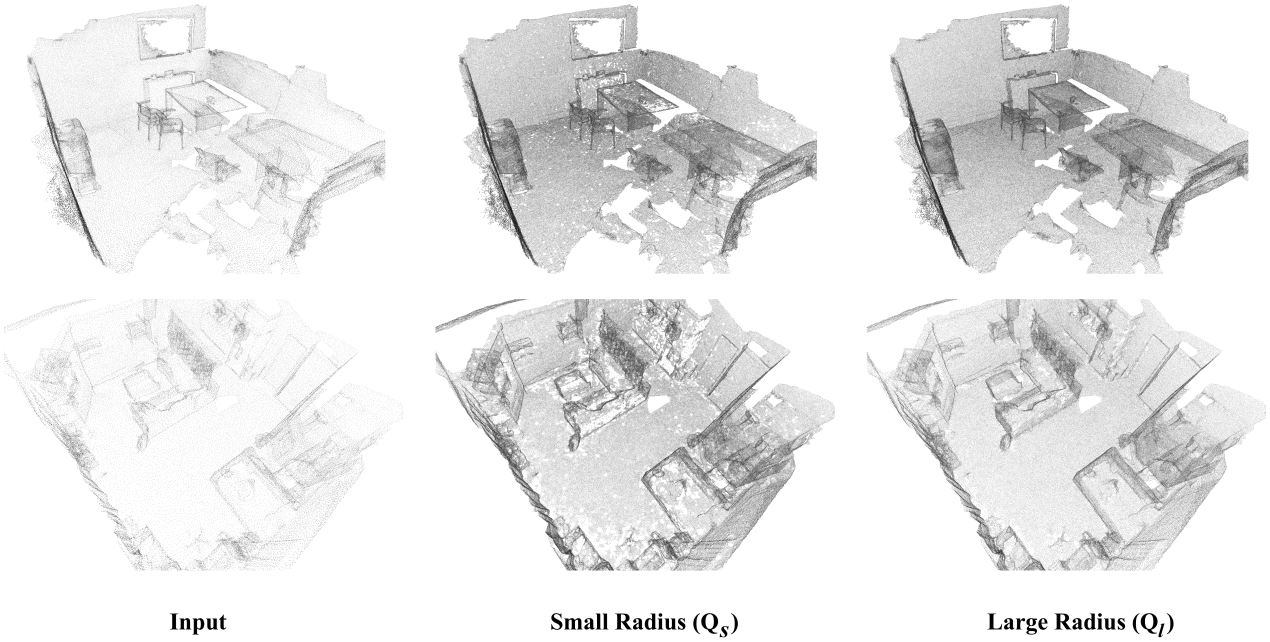

Figure 15. Visualization on ScanNet. Our model can adapt to the point clouds with varying sparsity and uniformity via adjusting the query radius in mesh construction.

6.4. Mesh deformation analysis

In Fig. 11, we show the detailed deformations for diverse objects to reveal the model's working process. Deformation A primarily concentrates on meshing a denser outcome, whereas deformation B tends to compensate for the incomplete structure generated in the previous deformation. Com-

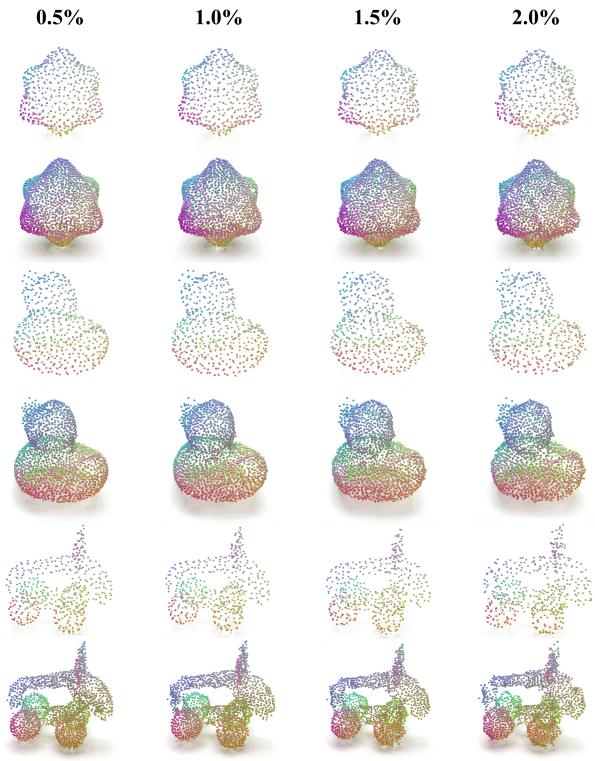

Figure 16. Qualitative evaluation of robustness. The results illustrate that our framework is robust under different noise intensities.

bining these two steps allows the model to recover dense and uniform point clouds. Also, we provide more node motion processes to demonstrate the implementation basis of mesh deformation in Fig. 14.

6.5. Robustness evaluation

In this study, we discuss the robustness of SPU-PMD. First, we randomly add noise to the input point cloud, which conforms to the Gaussian distribution $N(0, \sigma^2)$, where σ represents the standard deviation. By adjusting the value of σ, we control the noise intensity. In practice, we set σ from 0% to 2%. As shown in Fig. 17, only HD grows obviously with the increasing noise intensity in the two selected objects. Other metrics do not increase rapidly, which proves the robustness of SPU-PMD. The corresponding visual results in Fig. 16 also demonstrate that our model is robust to varying noise.

6.6. Evaluation on ScanNet Dataset

Apart from synthetic and KITTI data, we evaluate our model on the real scanning point clouds with non-uniform distributions from ScanNet [3] dataset and present the results produced by using a larger query radius, as shown in Fig. 15. Specifically, we set the small query radius Q_s to $[r, 1.3r]$, and the large query radius Q_l to $[r, 1.3r, 2r, 2.5r, 3r]$, where r represents the average radius of the nearest neighbor. These results demonstrate the flexibility of our model to the point clouds with varying sparsity and uniformity. Adjusting the query radius allows SPU-PMD to handle the point clouds with diverse distributions.

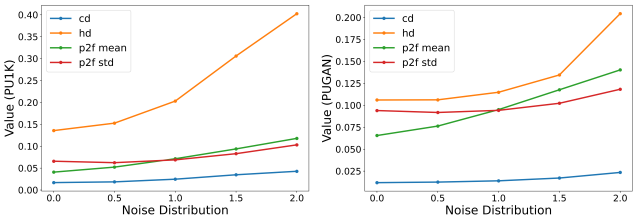

Figure 17. Quantitative evaluation of robustness. We test the robustness of SPU-PMD with the inputs corrupted by random Gaussian noise.