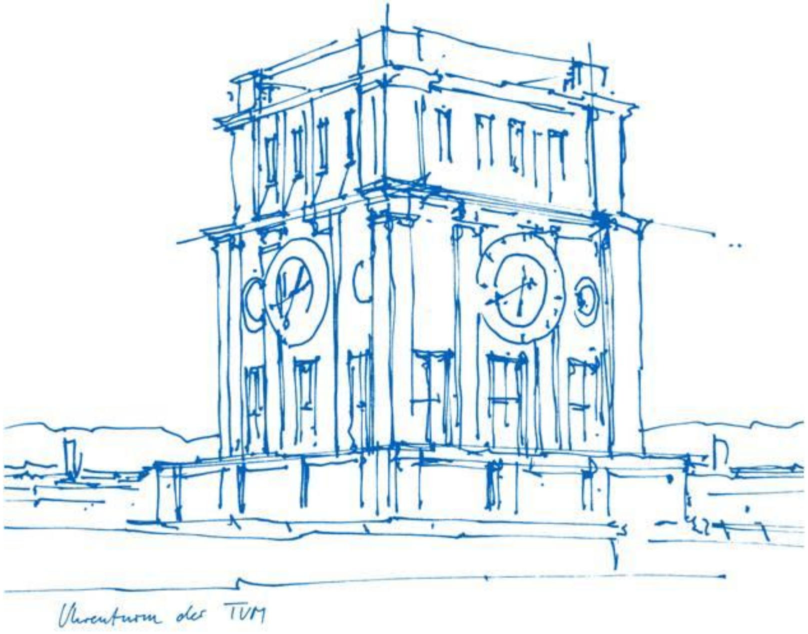
3D Point Cloud Completion Using Improved MSN with Novel SoftPool++ Architectures

Supervisor: Prof. Dr. Angela Dai

Tutor: M.Sc. Yujin Chen, M.Sc. Can Guemeli

Students: Jiongyan Zhang, Zichen Zhang, Chengzhi Shen



Outline

- Introduction
- Related Work
- Method - Baseline
- Method - Our Approach
- Experiment & Evaluation
- Conclusion

Introduction & Related Work

- Point cloud completion takes the incomplete point cloud as input and outputs the complete point cloud. The completion process includes both global structure reconstruction and local fine detail preservation.
- There have been many outstanding architectures proposed in recent years, such as FoldingNet [1], PCN[2], and GRNet [3].

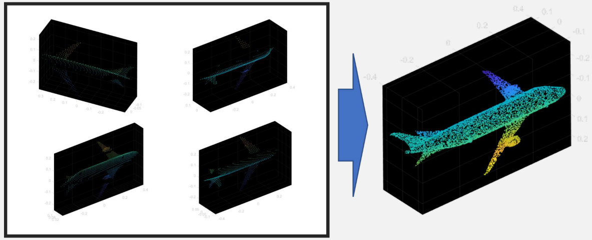


Figure 1. Input and output

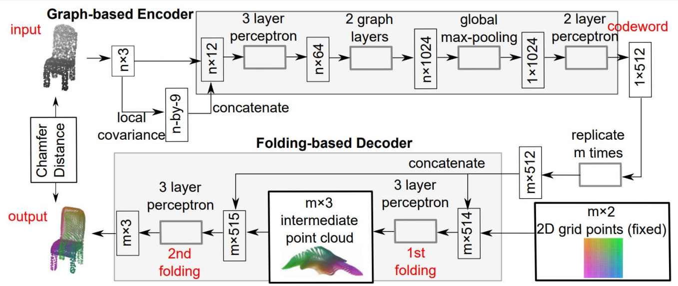


Figure 2. FoldingNet [1]

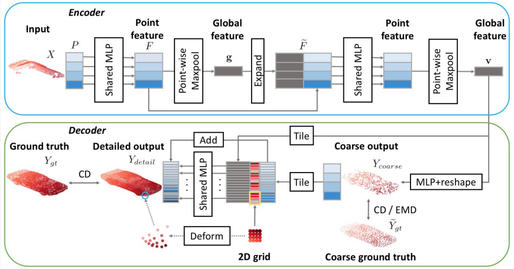


Figure 3. PCN [2]

[1] Yang, Y., Feng, C., Shen, Y., & Tian, D. (2018b). Foldingnet: Point cloud auto-encoder via deep grid deformation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 206–215).
[2] Yuan, W., Khot, T., Held, D., Mertz, C., & Hebert, M. (2018). PCN: Point completion network. In 2018 international conference on 3D vision (3DV) (pp. 728–737). IEEE.
[3] Xie, H., Yao, H., Zhou, S., Mao, J., Zhang, S., & Sun, W. (2020b). GRNET: Gridding residual network for dense point cloud completion. In A. Vedaldi, H. Bischof, T. Brox, & J. M. Frahm (Eds.), Computer vision—ECCV 2020 (pp. 365–381). Springer.

Method - Baseline

Morphing and Sampling Network [4]

- Encoder-Decoder
- Decoder has a morphing operation
- Coarse completion merged with input
- Minimum density sampling

$$p_i = \underset{x \notin \mathrm{P}_{i-1}}{\operatorname{argmin}} \sum_{p_j \in \mathrm{P}_{i-1}} \exp(-\|x - p_j\|^2/(2\sigma^2))$$

- Residual network
- Fine completion
- Loss functions

$$\mathcal{L}_{\text{expansion}} = \frac{1}{KN} \sum_{1 \le i \le K} \sum_{(u,v) \in \mathcal{T}_i} \mathbb{1}\{\text{dis}(u,v) \ge \lambda l_i\} \text{dis}(u,v)$$

$$\mathcal{L}_{\text{EMD}}(S_1, S_2) = \min_{\phi: S_1 \to S_2} \frac{1}{|S_1|} \sum_{x \in S_1} \|x - \phi(x)\|_2$$

$$\mathcal{L} = \mathcal{L}_{\text{EMD}}(\mathrm{S}_{\text{coarse}}, \mathrm{S}_{\text{gt}}) + \alpha \mathcal{L}_{\text{expansion}} + \beta \mathcal{L}_{\text{EMD}}(\mathrm{S}_{\text{final}}, \mathrm{S}_{\text{gt}})$$

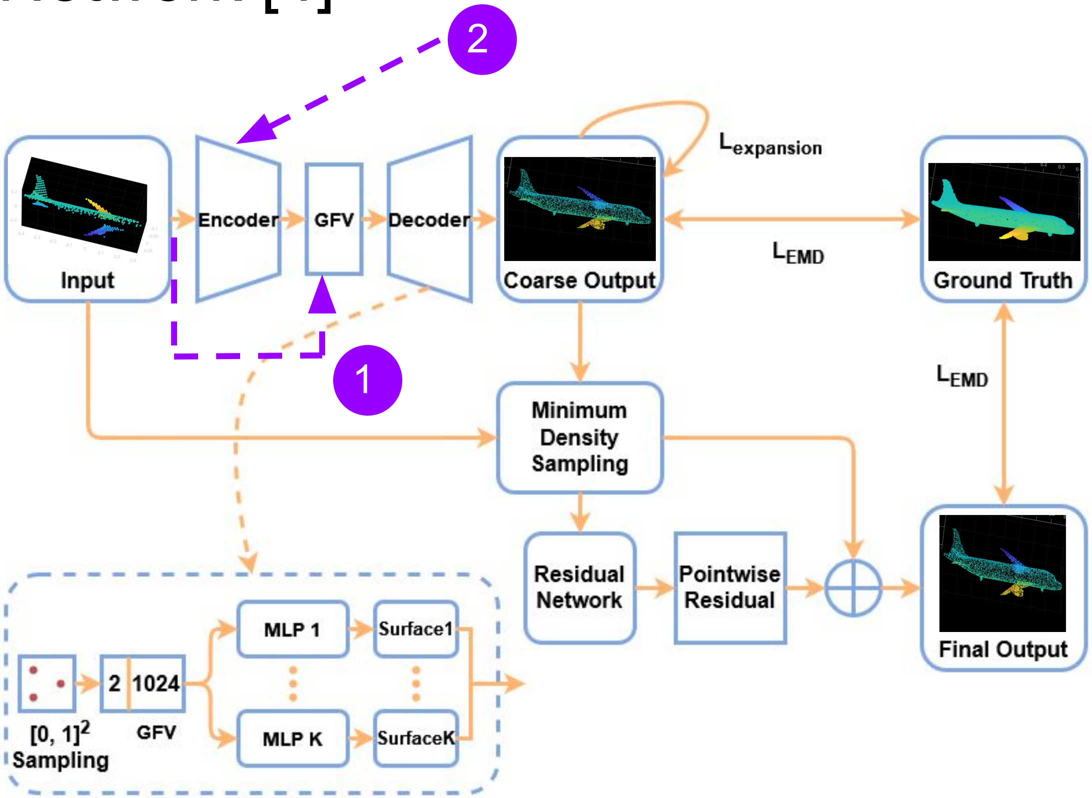


Figure 4. MSN

[4] Liu, M., Sheng, L., Yang, S., Shao, J., & Hu, S. M. (2020). Morphing and sampling network for dense point cloud completion. In Proceedings of the AAAI conference on artificial intelligence (vol. 34, pp. 11596–11603).

Method - Our Approach

Global Feature Transform 1

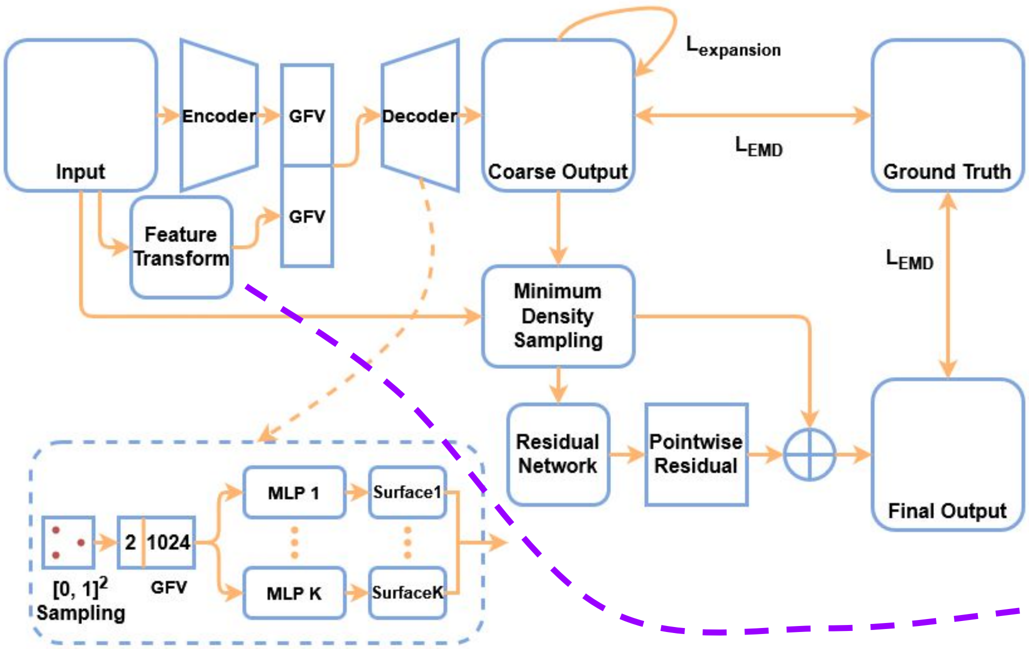


Figure 5. Our complete architecture

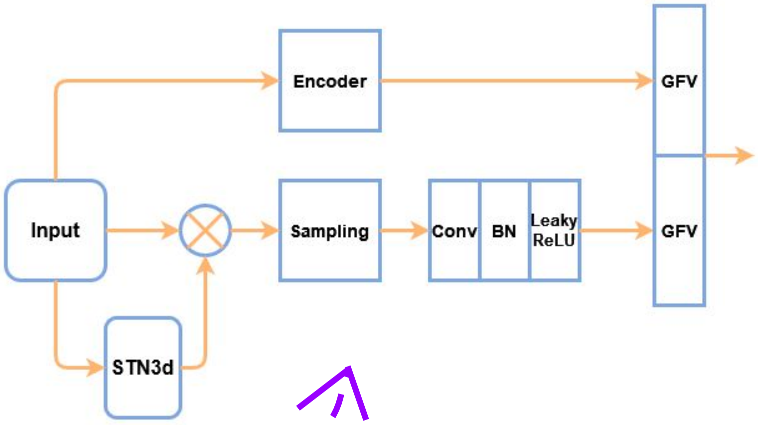


Figure 6. Global feature transform

Encoder - PointNet [5] -> SoftPoolPointNet 2

- STN3d
- SoftPool operator
- Truncation strategy
- Local convolution
- ReLU -> LeakyReLU(0.2)
- Dropout(0.1)

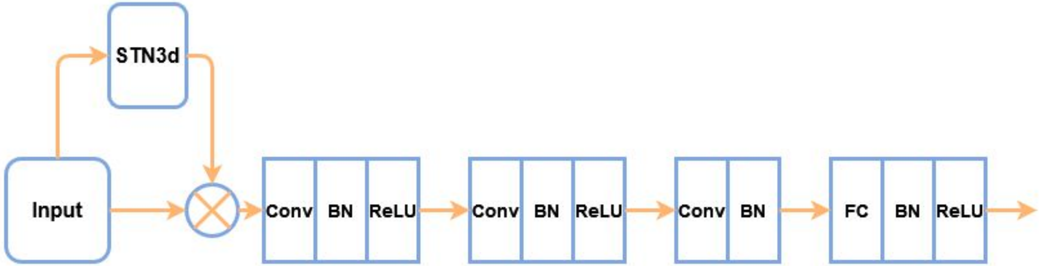


Figure 7. PointNet in MSN

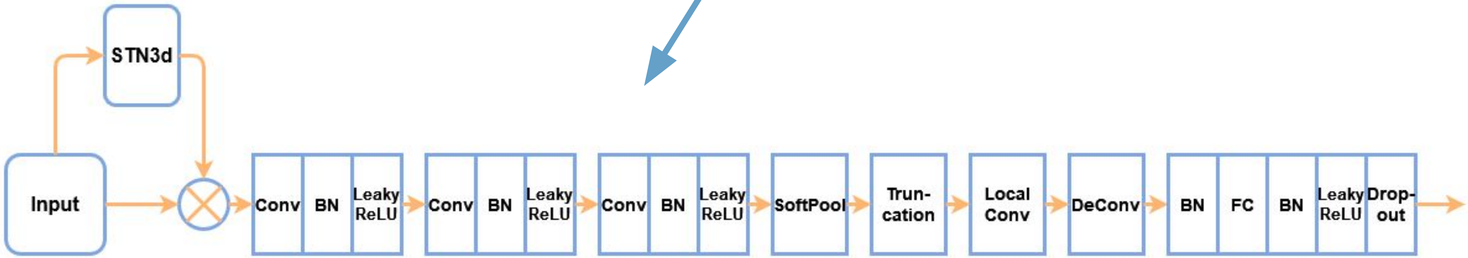


Figure 8. Our SoftPoolPointNet

[5] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017a). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652–660).

SoftPool-Truncated Features [6] 2

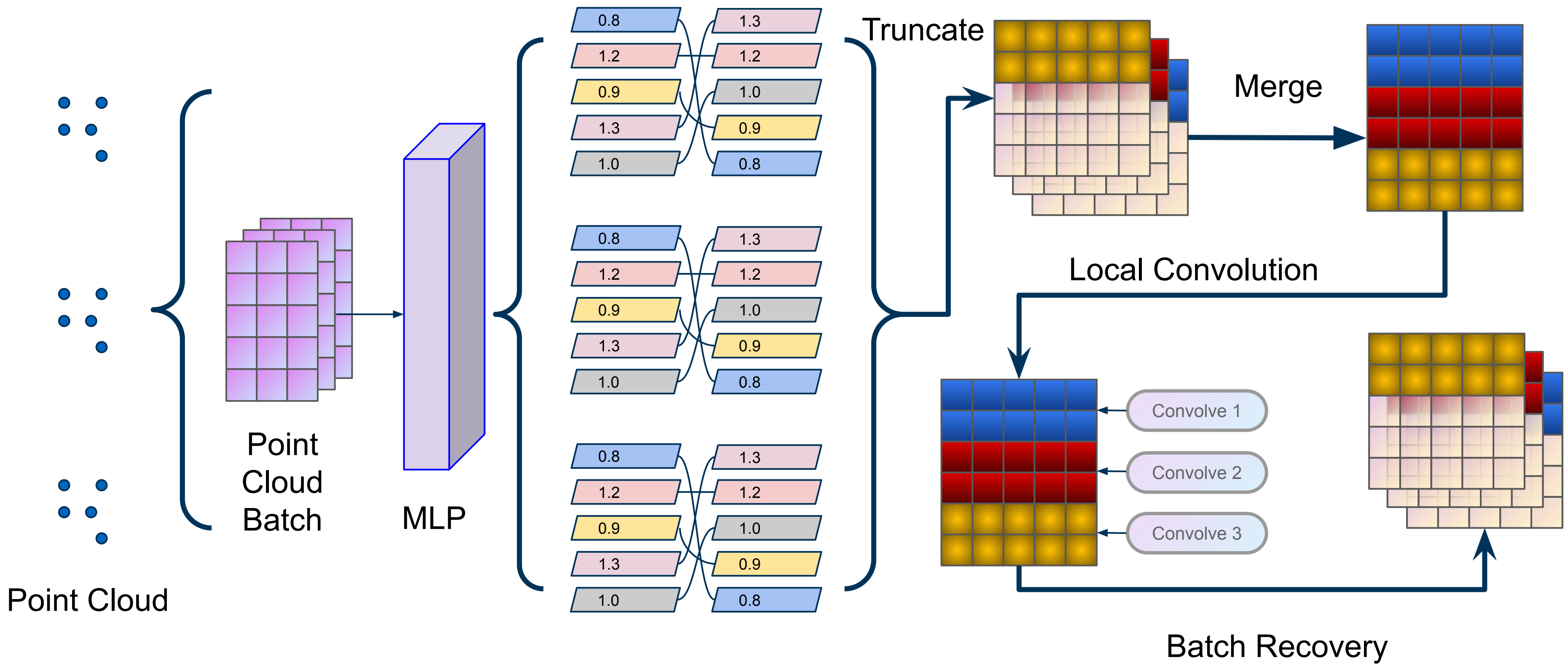


Figure 9. SoftPool with truncation strategy, local convolution

[6] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Softpool++: Anencoder–decoder network for point cloud completion. International Journal of ComputerVision, 130(5):1145–1164, 2022.

Experiment & Evaluation

Data Preparation

It costs a lot of time to generate random scans and render the corresponding models.

Data Source:
We use the ShapeNet [7] dataset (Core.v1) as our original 3D model source.

Pose Scan:
Use blender to fetch the random 50 poses of the 3D model and record its depth images.

Render:
Render the fetched poses back to partial 3D models (the Point Cloud format).

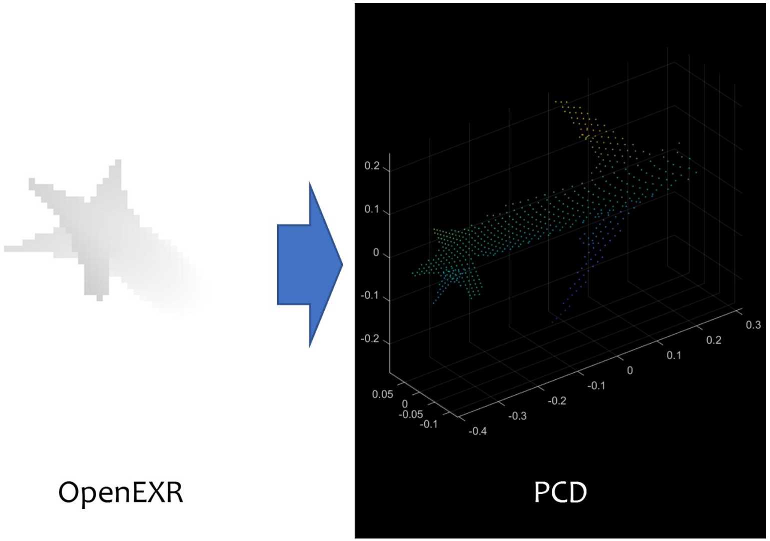


Figure 10. Data generation

[7] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An informationrich 3d model repository. arXiv preprint arXiv:1512.03012, 2015.

Training Data vs. Validation Data

- The training set is produced by our team - 7000 3D models with 50 scans each (meaning 7000x50 scans).
- The validation set comes from the MSN authors - 700 models (50 scans each) with denoise are randomly selected.
- The test set also comes from the validation set provided by the MSN authors - 50 models with 50 scans each are randomly selected.

The ground truth is provided by MSN authors, originally from the ShapeNet dataset.

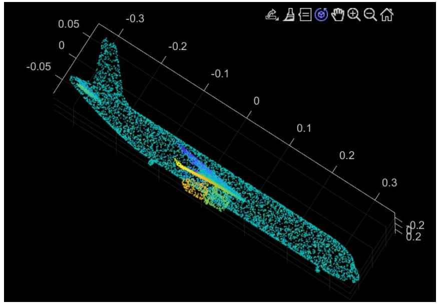


Figure 11. The point cloud

Hardware & Hyper-parameters Configuration

Hardware:

Nvidia RTX 3090 (24 GB) [AutoDL]

Hyper-Parameters:

- Batch size: 64
- Points number: 4096
- Primitive number: 16
- Epoch: 10
- Learning rate: 0.001
- Optimizer: Adam optimizer

Evaluation - validation loss

- Number of parameters: baseline(MSN) - 30.2M, our approach - 33.7M
- Our approach has a faster convergence, and better validation loss within the first 10 epochs
- Due to the limitation of hardware resources, the limit performance of the model is not tested.
- Test set: baseline(MSN) - 0.083, our approach - 0.074

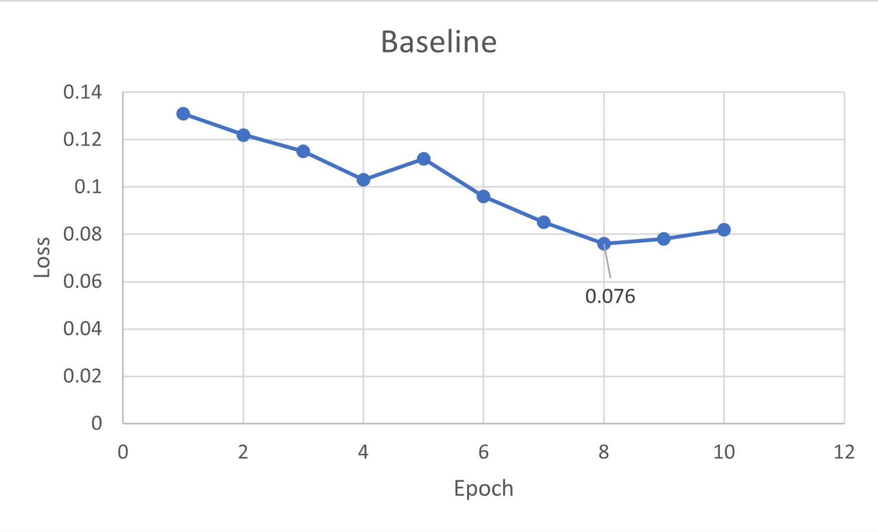


Figure 12. Performance of baseline(MSN)

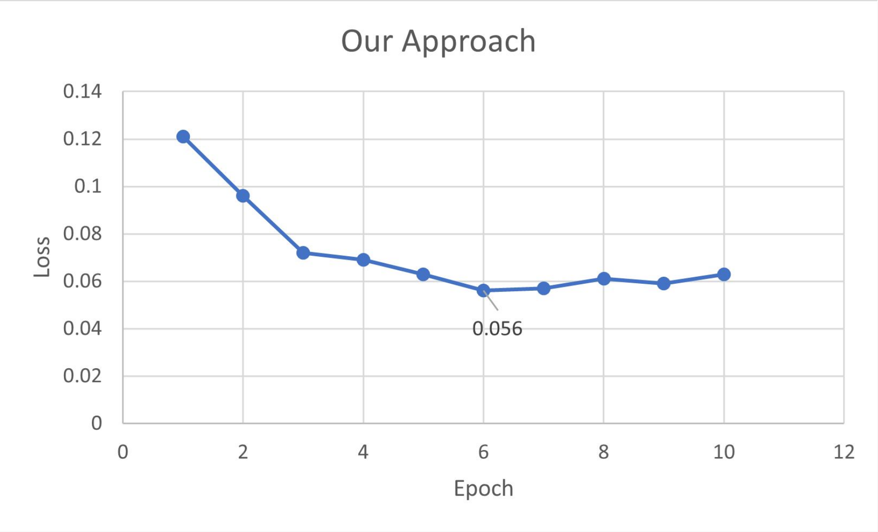


Figure 13. Performance of our approach

Baseline(MSN) Result Visualization

Left is the ground truth and the right is our output.

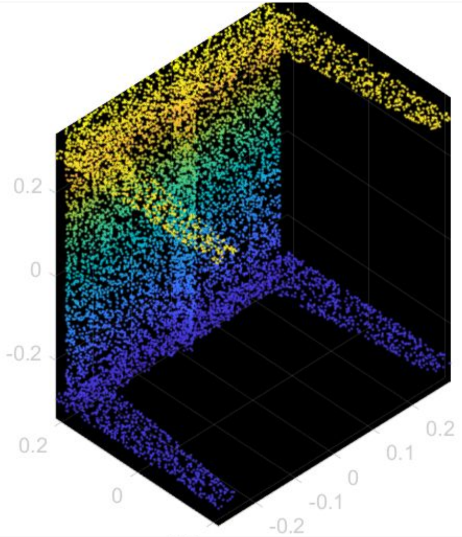


Figure 14. The ground truth

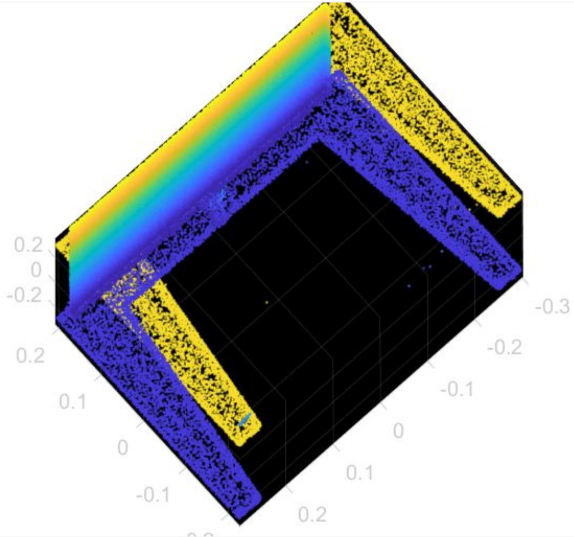


Figure 15. The result of baseline(MSN)

Our new approach effect

Left is the ground truth and the right is our output.

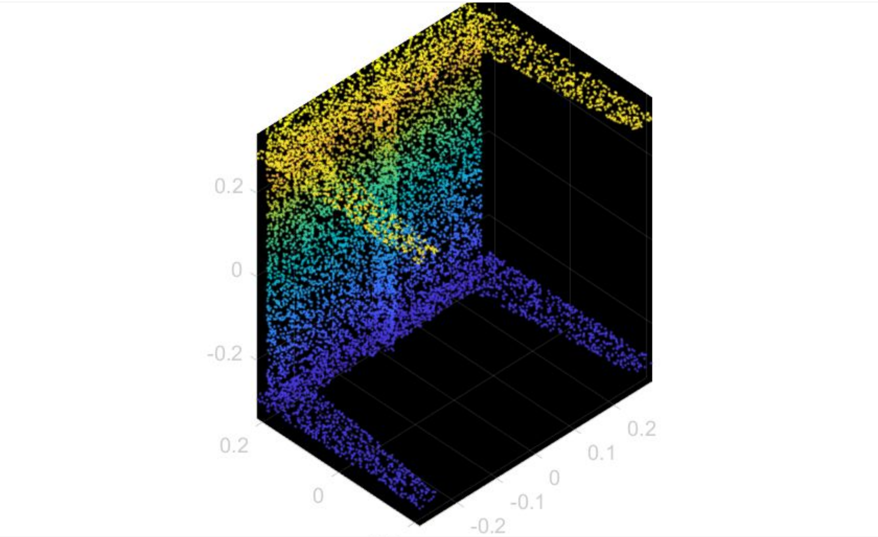


Figure 16. The ground truth

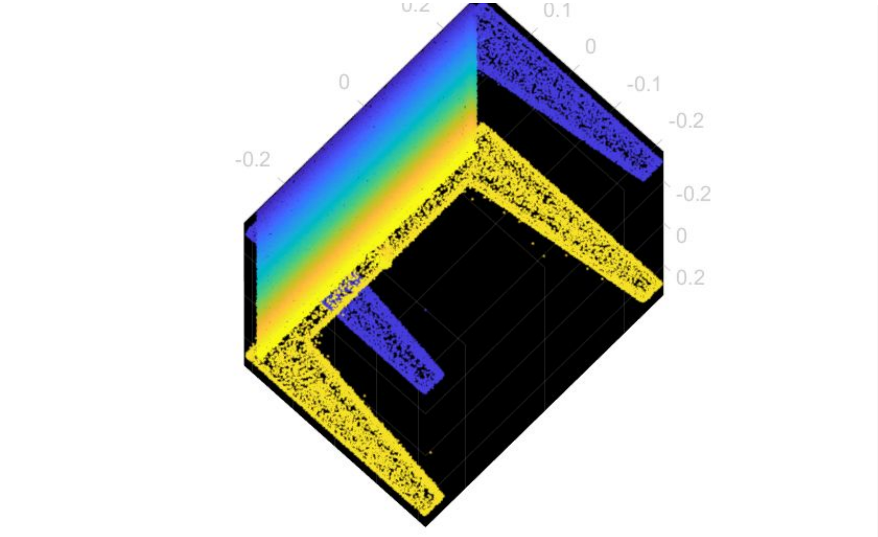


Figure 17. The result of our approach

Conclusion

- The original MSN is indeed a sufficiently powerful model for point cloud completion.
- Our approach achieves a faster convergence and better validation loss within the first 10 epochs with a very close number of parameters to the MSN.
- This proves the effectiveness of SoftPool-truncated features, local convolution, and global feature transform.
- The training process shows overfitting in later epochs, this might be due to insufficient training data. The original paper uses 30000 models with 50 scans, ours is 7000 models with 50 scans.

Thank you for your attention!