Improving VRCNet for point cloud completion

Mayank Bansal,
Department of Robotics Engineering,
Worcester Polytechnic Institute,
Worcester, United States of America
mbansal1@wpi.edu

***Abstract*—In the realm of autonomous systems, a diverse array of sensors, including LiDAR, lasers, and RGB-D scanners, are employed to gather 3D point cloud data, a critical component for environmental perception. Addressing the complex challenge of point cloud completion, which involves restoring missing points in these partial point clouds, is a crucial aspect of 3D computer vision technology. This research focuses on the advanced implementation of VRCNet, a system that utilizes a variational approach for efficient point cloud completion. The study extends beyond the basic implementation of VRCNet, delving into significant enhancements to its architecture. These refinements include the integration of batch normalization, and the innovative use of Swish and Mish activation functions. Additionally, it explores the impact of alternative optimizers, such as AdamW and RAdam, on the network's performance. The research also conducts thorough experiments with various learning rates, providing insights into their influence on the training process and the overall effectiveness of the results. This comprehensive approach not only strengthens the foundational capabilities of VRCNet but also offers valuable contributions to the field of 3D computer vision.**

***Index Terms*—VRCNet, point cloud completion, MVP, Completion3D**

I. Introduction

In the dynamic domain of computer vision, the role of point cloud completion is becoming increasingly crucial, particularly in areas such as autonomous driving, robotic manipulation, and 3D reconstruction. The emergence of 3D scanning devices like LiDAR, laser, and RGB-D scanners has made the acquisition of point clouds more accessible, fueling substantial research in these fields. However, these initial point clouds often exhibit sparsity and partialness owing to occlusions and missing measurements, challenges that traditional methods struggle to overcome, leading to suboptimal performance in practical applications. VRCNet, a cutting-edge point cloud completion network, addresses these challenges, offering a significant advancement in the generation of complete point clouds from partial observations. It introduces two sub-networks: PMNet for embedding global features and generating coarse completions, and RENet for enhancing structural relations using multi-scale local point features. VRCNet demonstrates superior performance on the MVP dataset and benchmarks, effectively reconstructing detailed and plausible shapes from incomplete point clouds.

This research work implements VRCNet as a baseline and then experiments with enhancements to it, examining their impact on the network's efficacy and contributing to the advancement of point cloud completion methodologies in computer vision. The key contributions can be summarized as:

1) **Learning Rate Experiments:** Conducted trials with two distinct learning rates to assess their impact on the network's efficiency and accuracy.
2) **Optimizer Variations:** Evaluated the performance using different optimizers, specifically RAdam, Adam, and AdamW, to determine the most effective optimization strategy.
3) **Batch Normalization:** Implemented batch normalization between layers in the Linear Residual Block to improve the network's generalization and speed up convergence.
4) **Activation Function Investigation:** Explored the use of Swish and Mish activation functions within the Linear Residual Block to observe their effects on the model's learning dynamics.

II. Literature Survey

In my literature survey, I comprehensively explored the field of point clouds to grasp the breadth of research conducted in this area. Initially, my focus was on general studies related to point clouds, aiming to understand the foundational work and developments in this field. Subsequently, my attention shifted towards more specialized research, particularly those papers that employ deep learning techniques for point cloud completion. This in-depth exploration led me to select VRCNet as the baseline network for my investigation into point cloud completion, due to its innovative approach and significant contributions to the field.

"Deep Learning for 3D Point Clouds: A Survey" by Guo et al. [1] offers a detailed overview of deep learning in 3D point cloud processing. It covers three main areas: shape classification, object detection and tracking, and segmentation. The paper compares various methods on public datasets, providing insights and future research directions. Its comprehensive analysis makes it a significant resource in the field of 3D point cloud deep learning.

PointNet, developed by Qi et al. [2], marked a significant advancement in deep learning on point sets, directly processing point clouds while preserving permutation invariance. It catered to various applications like object classification and segmentation. However, PointNet's limitation in capturing local structures led to the development of PointNet++ [3], which enhances the original model by applying PointNet

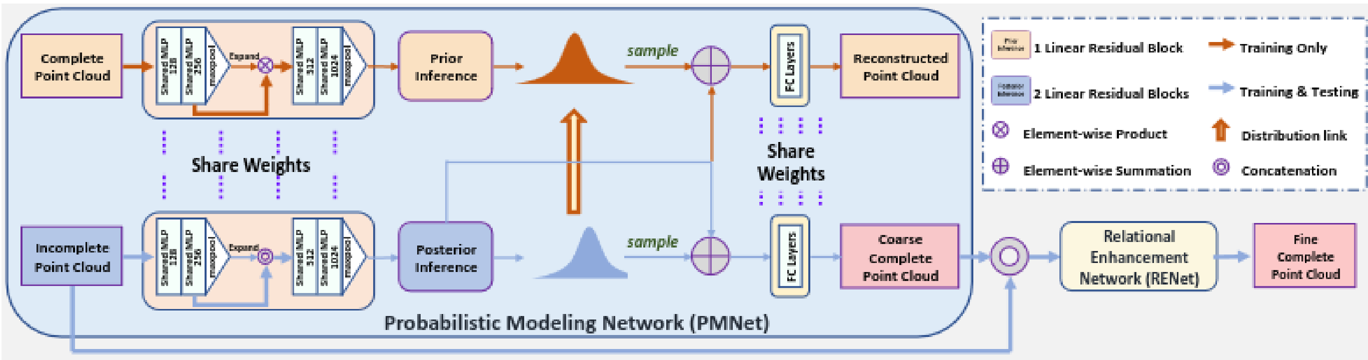


Fig. 1: VRCNet Baseline Network. The left blue part shows the probabilistic modeling sub-network (PMNet) which is responsible for generating a coarse complete point cloud from the input incomplete point cloud.

recursively in a hierarchical structure. PointNet++ effectively learns local features at multiple scales, addressing the issue of varying point densities and improving pattern recognition and generalizability in complex 3D point cloud environments.

Wang et al. present the paper "Dynamic Graph CNN for Learning on Point Clouds" [4] which introduces EdgeConv, an innovative neural network module tailored for point cloud tasks like classification and segmentation. It stands out by incorporating local neighborhood information, enabling the analysis of global shape properties. EdgeConv demonstrates impressive results on key benchmarks such as ModelNet40 and ShapeNetPart, marking a significant advancement in 3D point cloud processing.

Zhou, Yin et al. introduce VoxelNet [5], a comprehensive 3D object detection framework leveraging point cloud data. VoxelNet innovates by segmenting space into voxels and applying CNNs for feature extraction, blending point cloud with image-like data. This method leads to remarkable accuracy in 3D object detection, showcasing its effectiveness in identifying objects with precision.

Zhang et al. have developed an advanced 2D representation for point clouds, using ellipsoid surface projection to highlight local patterns. This is paired with EllipsoidNet [6], a new convolutional neural network for classifying and segmenting point clouds. This approach successfully addresses the accuracy and efficiency issues found in existing methods. The superiority of these techniques is validated through tests on the ModelNet40 and ShapeNet benchmarks, demonstrating marked improvements in point cloud processing.

Fei et al. present a survey [7], which addresses point cloud completion in 3D computer vision, significantly advanced by deep learning. It examines a range of methods, including point-based, view-based, and convolution-based approaches, among others. The survey highlights comparisons between these methods to inspire further research and discusses common datasets and applications, emphasizing the need to enhance point cloud quality for practical use.

In their 2022 Neurocomputing paper, Lin, Fangzhou et al. present the Cosmos Propagation Network [8], a novel deep learning approach for enhancing point cloud completion. This model adeptly fills gaps in point cloud data using propagation techniques, showing promise in delivering more detailed and accurate 3D reconstructions by addressing the challenge of incomplete point cloud information.

In their work, Yuan et al. introduce the Point Completion Network (PCN) [9], a new learning-based approach for shape completion. PCN stands out as it directly processes raw point clouds without relying on structural assumptions or annotations. Its decoder design enables the creation of detailed completions with a minimal number of parameters. PCN has shown effectiveness in generating dense, complete point clouds, even with inputs that have varying levels of incompleteness and noise, such as LiDAR-scanned cars from the KITTI dataset.

In 2021, Pan at el. introduce the Variational Relational point Completion network (VRCNet) [10], a novel approach for point cloud completion. VRCNet addresses the limitations of previous methods in capturing fine local details and structural relations in man-made objects. It features a dual-path architecture for probabilistic modeling and relational enhancement, using point self-attention and selective kernel modules to refine local shape details. Additionally, the paper contributes a comprehensive dataset (MVP dataset) with over 100,000 scans. VRCNet demonstrates superior performance over existing methods in standard benchmarks, showing remarkable generalizability and robustness in real-world scans.

III. Proposed Methodology

The research is centered around the VRCNet architecture, a novel network for point cloud completion. It consists of two consecutive encoder-decoder sub-networks called PMNet (shown in Fig. 1) and RENet (shown in Fig. 2). The initial phase involved setting up and successfully implementing VRCNet. Subsequent efforts focused on an ablation study to examine how various parameters influence the network's performance. In addition to this study, the research explored modifications to VRCNet's architecture, aiming to assess their impact on the network's final results. This comprehensive approach provided a deeper understanding of VRCNet's functionality and potential applications.

The different experimentation are described below:

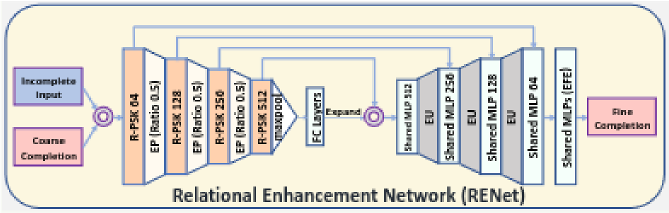


Fig. 2: Relational Enhancement Network (RENet). Responsible for generating fine complete point clouds from the coarse complete point cloud.

A. Batch Normalization in Linear Residual Block

In my work on VRCNet, one of the things I particularly focused on is incorporating Batch Normalization into the network's Linear Residual Block. This step is crucial for processing data effectively. Batch Normalization is known to stabilize and speed up the training of neural networks, and my aim was to use it for normalizing the outputs from each layer. This helps in reducing internal covariate shift, which I hoped would lead to faster convergence of the network and enhance its accuracy in completing point clouds.

The Linear Residual Block is designed to refine point cloud data, and by adding Batch Normalization, I expected a significant improvement in the network's handling of complex 3D structures. This improvement is essential for the network to reconstruct detailed and accurate point clouds, tackling issues like noise and sparsity in 3D data. My hypothesis was that this enhancement would not only make the training process more efficient but also improve the network's performance in real-world scenarios, which require high accuracy and detailed point cloud completion.

B. Activation Function Investigation

In another one of my experimentation, I ventured into refining the activation mechanisms within the Linear Residual Block, specifically replacing the standard ReLU function (Fig. 3). The choice of activation function is pivotal, as it introduces non-linearity into the learning process and significantly influences the network's performance.

My first experiment involved the Mish activation function (Fig. 4), which is acclaimed for its smooth, non-monotonic behavior that effectively maintains negative values, potentially enhancing the network's ability to model complex patterns. The smoothness of Mish is theorized to aid in the mitigation of vanishing gradients, a common pitfall in training deep neural networks. By incorporating Mish, I aimed to enrich VRCNet's competence in capturing and reconstructing the intricate geometrical structures found in point cloud data, expecting to see an uptick in the fidelity of the completed shapes.

Concurrently, I evaluated the Swish activation function (Fig. 5), renowned for its self-gating property that dynamically adjusts the flow of information through the network based on the input. This flexible modulation has the potential to resolve the dying ReLU issue, where neurons become inactive and cease contributing to the model's learning process. Implementing Swish promised to not only circumvent this issue but also to potentially bolster VRCNet's ability to parse and reconstruct detailed 3D point clouds with greater accuracy. The exploration into Swish aimed to propel the network's feature extraction capabilities, ultimately refining the quality of the point cloud completion.

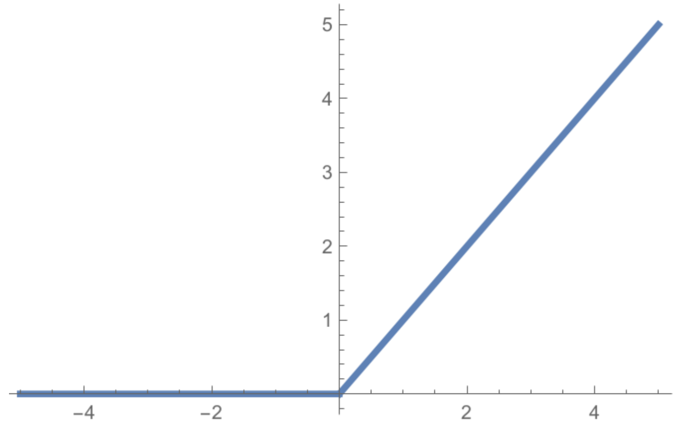


Fig. 3: Graph of ReLU activation function

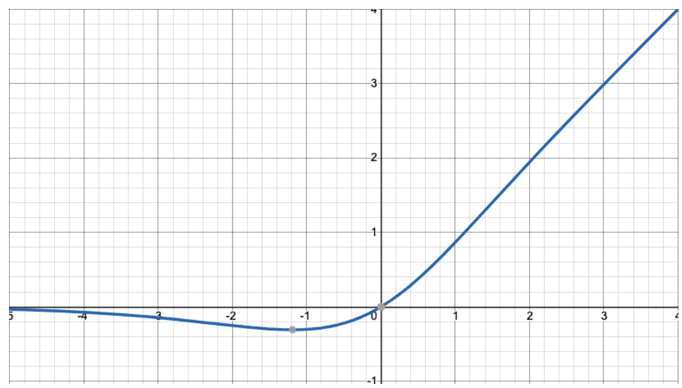


Fig. 4: Graph of Mish activation function

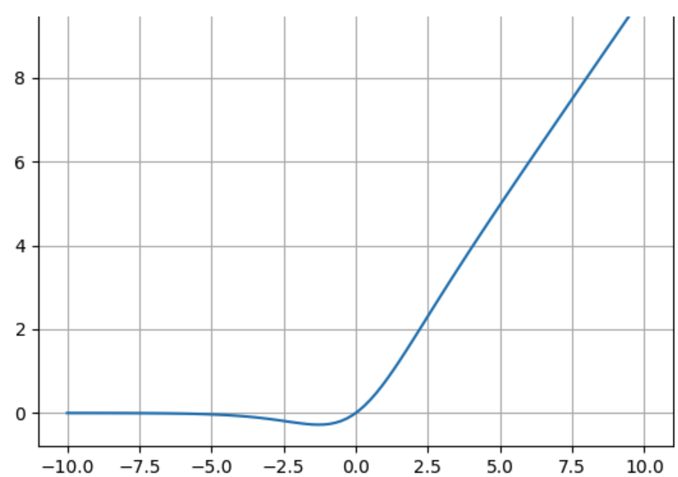


Fig. 5: Graph of Swish activation function

C. Optimizer Variations

I also performed experiments to probe the impact of various optimizers on the performance of the VRCNet architecture. While the baseline network employs the Adam optimizer, renowned for its efficiency in handling sparse gradients and adaptive learning rate capabilities, I was intrigued by the potential benefits of its variants, namely AdamW and RAdam.

AdamW, an adaptation of the original Adam optimizer, introduces a decoupled weight decay regularization, which theoretically can lead to better generalization by separately handling the learning rate and weight decay. The separation of these two parameters is particularly relevant in training deep neural networks on complex data such as point clouds, where the prevention of overfitting is paramount. I hypothesized that AdamW would not only retain the benefits of Adam but also enhance the model's capacity to generalize across diverse datasets, a critical aspect of point cloud completion tasks.

RAdam, or Rectified Adam, is another variant that caught my attention due to its rectification term that dynamically adjusts the adaptive learning rate, promising to stabilize the training process in the early stages when variance is high. Given the sensitivity of VRCNet's performance to the choice of optimizer, RAdam presented an opportunity to refine the convergence behavior of the network, potentially leading to more robust training dynamics and improved outcomes, especially in the nuanced task of point cloud completion.

D. Learning Rate Experimentation

In my final experiment, I explored the effects of altering the learning rate from the baseline setting of 0.0001 to a higher rate of 0.001. The learning rate is a critical hyperparameter that determines the pace at which a neural network updates its weights during training, directly impacting convergence and model accuracy.

Initially, the network was trained using the baseline learning rate of 0.0001. This conservative rate is typically chosen to ensure steady and precise convergence, minimizing the risk of overshooting optimal solutions in the loss landscape. It's particularly beneficial for fine-tuning the model's weights, albeit at a slower training speed.

I then increased the learning rate to 0.001, intending to investigate the trade-offs involved with a faster training process. A higher learning rate can accelerate convergence, potentially making the training process more efficient. However, it also carries the risk of missing the finer details in the loss landscape, which could lead to suboptimal model performance.

This experiment was designed to provide insights into the optimal learning rate for VRCNet, especially considering the balance between training speed and the precision of point cloud completion tasks. The results would inform whether the increased speed of convergence with a higher learning rate justifies any potential decrease in model accuracy.

IV. Experiments

The MVP Dataset: MVP is a comprehensive collection of high-quality, incomplete point clouds. What sets the MVP dataset apart is its diverse and realistic rendering approach, utilizing 26 camera positions uniformly distributed around each CAD model. This method offers a wide array of partial shapes, crucial for robust training and testing in point cloud completion tasks. The dataset, with over 100,000 high-resolution point clouds, surpasses previous datasets in realism and variety. It includes 16 diverse shape categories, adding complexity and breadth to the dataset. I have used the dataset as it is in my research work.

Implementation Details: The network was implemented in PyTorch with a batch size of 32 for the experiments. The initial learning rate and the choice of optimizers varied according to each experiment, though the learning rate was systematically decayed by 0.7 every 40 epochs. For these experiments, I considered 4096 points, as increasing the number of points led to longer training times, while fewer points resulted in outputs too sparse for effective visualization. The models were trained on the NVIDIA A100 GPU, provided by the WPI Turing cluster, with each model undergoing a training period of 24 hours.

Training Process: In my research, I meticulously conducted a series of 9 experiments to evaluate the impact of different optimizers, learning rates, and specific network enhancements like Batch Normalization, and the Swish and Mish activation functions on the VRCNet model.

The first phase of my experimentation involved using a learning rate of 0.0001. Under this condition, I trained three separate models using Adam, RAdam, and AdamW as optimizers to discern the nuances each brought to the network's learning dynamics. This procedure was then replicated with an increased learning rate of 0.001 to observe the effects of a faster convergence rate.

In the second phase, while maintaining the learning rate at 0.0001, I focused on integrating Batch Normalization and experimenting with the Swish and Mish activation functions. Each of these modifications was applied to a separate model to isolate their individual contributions and compare its performance against the baseline Adam optimizer in conjunction with these network enhancements.

Overall, this comprehensive approach provided deep insights into how various parameters influence the network's training process and its overall performance. These experiments were instrumental in determining whether these modifications could enhance the network's efficiency and accuracy in point cloud completion tasks.

V. Results and Discussion

In this study, a series of experiments were conducted to evaluate the impact of various optimizers and architectural modifications on the performance of the VRCNet model for point cloud completion tasks. The baseline for comparison was established using the Adam optimizer with a learning rate of 0.0001. The findings reveal the nuanced effects that different optimizers and learning rates, as well as network modifications, have on model performance.

A. Impact of Optimizers at Learning Rate 0.0001

The data is shown in Table 1. The baseline performance indicated by the Adam optimizer showed consistent `cd_t`

TABLE I: Quantitative Comparison of Point Cloud Completion in term of Chamfer Distance cd_t with learning rate 0.0001

Optimizer	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Bed	Bench	Bookshelf	Bus	Guitar	Motorbike	Pistol	Skateboard	Overview Result
Adam (Baseline)	2.13	5.96	4.09	6.48	10.34	6.40	7.25	4.80	12.86	5.34	8.26	3.29	1.16	2.88	3.61	2.30	0.000589
RAdam	2.09	6.19	4.09	6.85	10.57	6.58	7.63	5.19	13.60	5.45	6.93	3.33	1.32	3.02	3.06	2.05	0.000582
AdamW	2.05	5.92	4.16	6.20	9.71	6.29	6.88	5.17	12.48	5.20	7.17	3.39	1.19	2.90	3.47	2.39	0.000554

TABLE II: Quantitative Comparison of Point Cloud Completion in term of Chamfer Distance cd_t with learning rate 0.001

Optimizer	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Bed	Bench	Bookshelf	Bus	Guitar	Motorbike	Pistol	Skateboard	Overview Result
Adam	2.60	11.51	6.72	10.46	9.97	11.89	12.19	6.24	16.43	6.36	11.28	6.60	3.05	4.18	4.33	3.31	0.000845
RAdam	2.56	7.95	5.20	9.33	13.62	10.31	9.99	6.18	15.30	6.83	8.48	4.48	2.33	3.71	5.74	7.28	0.000780
AdamW	3.79	15.11	7.99	13.31	14.77	20.17	14.80	7.49	23.03	7.92	13.40	8.72	3.47	4.51	7.62	6.22	0.001148

TABLE III: Quantitative Comparison of VRCNet Modifications on Point Cloud Completion Performance Using Chamfer Distance Metric (cd_t) at Learning Rate 0.0001

Modification	Airplane	Cabinet	Car	Chair	Lamp	Sofa	Table	Watercraft	Bed	Bench	Bookshelf	Bus	Guitar	Motorbike	Pistol	Skateboard	Overview Result
Batch Normalization	1.90	6.09	4.13	6.52	7.55	5.99	6.78	4.32	10.80	4.60	6.57	3.33	1.30	3.06	2.92	2.49	0.000515
Swish	2.21	5.96	4.15	6.45	10.60	6.41	7.42	4.94	12.72	4.81	8.05	3.32	1.02	2.97	3.20	2.37	0.000571
Mish	2.06	5.99	4.07	6.42	9.54	6.44	7.25	4.94	12.73	5.19	7.57	3.28	1.11	2.87	3.68	3.01	0.000562

losses across different categories. A slight improvement was observed with the RAdam optimizer, suggesting its effectiveness in stabilizing the training process and enhancing generalization capabilities. AdamW, while incorporating a decoupled weight decay regularization, did not consistently improve performance, as indicated by increased cd_t losses in several categories such as Sofa and Bed. This variability underscores the complexity of optimizing deep learning models for specific tasks.

B. Impact of Optimizers at Learning Rate 0.001

We can see in Table 2 that increasing the learning rate to 0.001 yielded higher cd_t losses for all optimizers, implying a potential trade-off between training speed and convergence precision. Notably, AdamW under this learning rate resulted in the highest losses, indicating that this particular combination may not be conducive to the dataset or the model architecture being used.

C. Modifications with Adam at Learning Rate 0.0001

Table 3 shows that introducing Batch Normalization to the model resulted in lower cd_t losses in most categories, highlighting its benefit in terms of training stability and convergence. In contrast, replacing ReLU with the Swish activation function did not translate to a clear advantage, as evidenced by increased cd_t losses. The Mish activation function displayed mixed results, which suggests context-dependent utility, warranting further investigation.

D. General Observations

The experiments underscore that there is no one-size-fits-all solution when it comes to the choice of optimizer or architectural modification across all categories of point cloud completion. The observed variance in performance across different scenarios underscores the importance of bespoke strategies. This is corroborated by the qualitative results depicted in Table 4, which align with the quantitative findings previously discussed. In particular, lower cd_t loss values—indicative of a model's approximation to the ground truth—were not consistently observed across all modifications, further highlighting the nuanced challenge inherent in the optimization of neural networks.

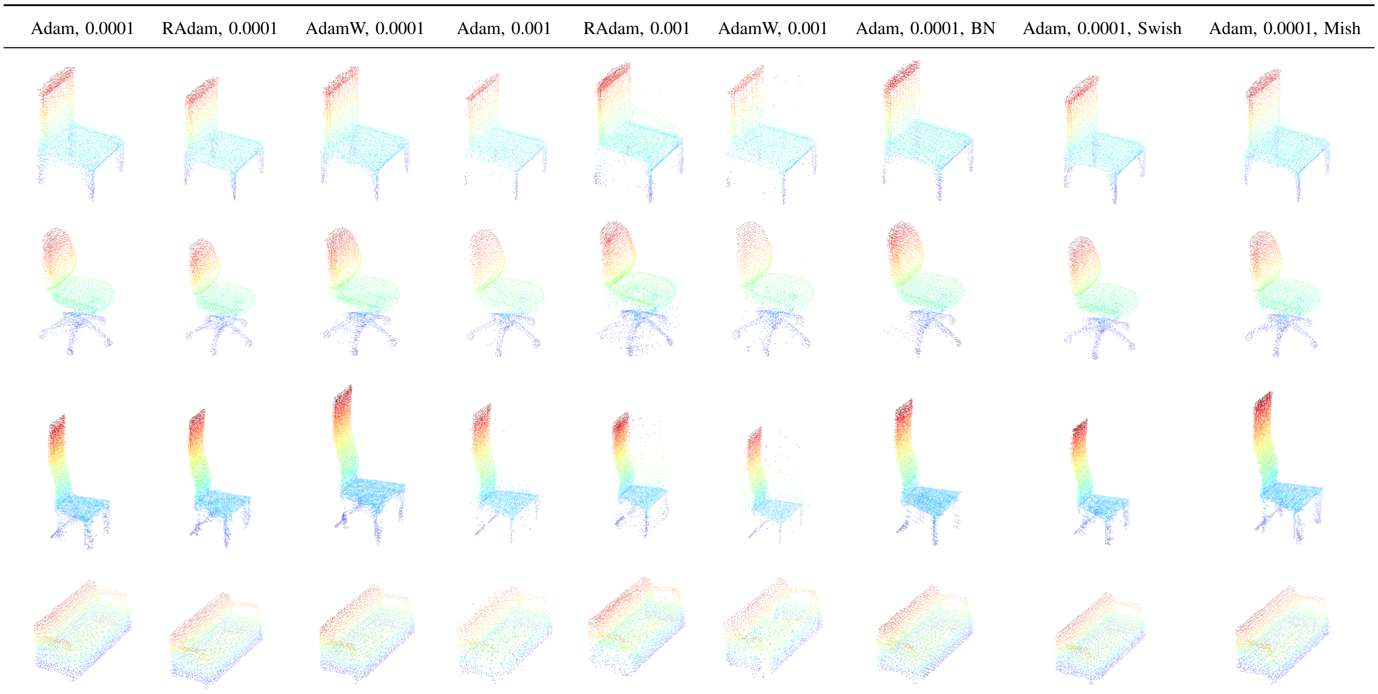
VI. Future Scope

The current research on VRCNet for point cloud completion has established a strong foundation for future investigations aimed at boosting the model's performance. Although the modifications implemented have yielded encouraging results, there is significant scope for further enhancements. An immediate area for future research could involve experimenting with various data augmentation techniques, which might enable the network to better understand and reconstruct complex point cloud structures, leading to more precise completions.

The limited training period in our experiments, due to computational constraints, suggests the possibility of gains from extended training. By increasing the duration of training, the model could potentially develop a deeper understanding of the dataset, which may translate into improved accuracy and finer details in point cloud completion. Further, a detailed analysis of convergence behaviors and an extensive exploration of hyperparameter tuning could reveal more efficient training strategies, thereby improving the network's effectiveness.

Moving forward, incorporating a wider range of datasets and exploring the application of transfer learning methods may greatly enhance the generalization abilities of the VRCNet architecture. Exploring different loss functions, regularization techniques, and minor architectural modifications could also lead to new insights and improvements in point cloud completion. Additionally, a focus on improving the interpretability and visualization of the model will provide clearer insights into the network's decision-making process, helping to identify areas for further refinement. Such a comprehensive

TABLE IV: Qualitative Comparison of Point Cloud Completion on MVP Dataset Across Different Experimental Setups



and exploratory approach is anticipated to make significant contributions to the domain of 3D point cloud processing, paving the way for both practical applications and theoretical progress.

VII. Conclusion

In conclusion, this research has made significant strides in advancing the VRCNet model for point cloud completion, delving deep into the technical aspects and their impacts. The exploration of different optimizers, namely Adam, RAdam, and AdamW, under varying learning rates, provided a nuanced understanding of how these parameters influence model performance. Notably, the introduction of Batch Normalization and the experimentation with Swish and Mish activation functions were pivotal in discerning their effects on the network's accuracy and efficiency in reconstructing point clouds.

The findings from these experiments underscore the complexity of optimizing neural networks for point cloud data. A key insight is that each modification's effectiveness varies based on the specific dataset and task at hand, indicating the need for tailored optimization strategies. The research also highlighted that while certain modifications, like Batch Normalization, consistently improved model performance, others like Swish and Mish activation functions showed mixed results.

This body of work contributes to a deeper understanding of the VRCNet model, offering a solid foundation for future research in 3D computer vision. The progress made in this study not only enhances the VRCNet's capabilities but also opens new doors for exploring different datasets, learning strategies, and further architectural enhancements. The journey through this research has been a step forward in the quest for more accurate and efficient point cloud completion solutions.

References

[1] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep Learning for 3D Point Clouds: A Survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 12, pp. 3061-3079, Dec. 2020.

[2] R. Q. Charles, H. Su, M. Kaichun and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 77-85, doi: 10.1109/CVPR.2017.16.

[3] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in NIPS, 2017.

[4] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic Graph CNN for Learning on Point Clouds.," CoRR, vol. abs/1801.07829, 2018.

[5] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection.," CoRR, vol. abs/1711.06396, 2017.

[6] Y. Lyu, X. Huang and Z. Zhang, "EllipsoidNet: Ellipsoid Representation for Point Cloud Classification and Segmentation," 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 2022, pp. 256-266, doi: 10.1109/WACV51458.2022.00033.

[7] Fei, Ben, Weidong Yang, Wenqing Chen, Zhijun Li, Yikang Li, Tengyu Ma, Xing Hu and Lipeng Ma. "Comprehensive Review of Deep Learning-Based 3D Point Cloud Completion Processing and Analysis." IEEE Transactions on Intelligent Transportation Systems 23 (2022): 22862-22883.

[8] Lin, Fangzhou, et al. "Cosmos Propagation Network: Deep learning model for point cloud completion." Neurocomputing 507 (2022): 221-234.

[9] Yuan, Wentao Khot, Tejas Held, David Mertz, Christoph Hebert, Martial. (2018). PCN: Point Completion Network. 728-737. 10.1109/3DV.2018.00088.

[10] Pan, Liang Chen, Xinyi Cai, Zhongang Zhang, Junzhe Zhao, Haiyu Yi, Shuai Liu, Ziwei. (2023). Variational Relational Point Completion Network for Robust 3D Classification. IEEE transactions on pattern analysis and machine intelligence. PP. 10.1109/TPAMI.2023.3268305.