Mitigating I/O Bottlenecks in LiDAR Pipelines by Directly Merging Neural Decompression and Semantic Segmentation

Ethan Marquez*
Clemson University
Clemson, South Carolina, USA
marque6@clemson.edu

Max Faykus
Clemson University
Clemson, South Carolina, USA
mfaykus@clemson.edu

Oyinlolu Odetoye
Clemson University
Clemson, South Carolina, USA
oodetoy@clemson.edu

Melissa C. Smith
Clemson University
Clemson, South Carolina, USA
smithmc@clemson.edu

Jon C. Calhoun
Clemson University
Clemson, South Carolina, USA
jonccal@clemson.edu

Abstract

The increasing volume of high-resolution LiDAR data poses a significant I/O bottleneck in large-scale analysis and high-performance computing pipelines due to costly intermediary data storage and retrieval. We introduce a novel, end-to-end framework that addresses this issue by proposing the first unified RENO-based neural autoencoder with a Point Transformer v3 (PTV3) segmentation backbone. This integrated architecture directly feeds the high rank feature tensors of the RENO decoder into the segmentation backbone, completely bypassing the need for costly intermediary file storage and I/O operations. Evaluated on the German Outdoor and Offroad (GOOSE) dataset, this approach enables direct semantic analysis on compressed data. Our results demonstrate that this method significantly reduces storage overhead, saving 29.9 GB per 13,076 point clouds and 2.7 GB per minute of LiDAR operation, all while maintaining the accuracy of semantic segmentation. This unified framework represents a major step towards efficient, real-time processing of large-scale point cloud datasets.

Keywords

point cloud segmentation, lidar segmentation, autoencoder, point cloud autoencoder, autoencoder semantic segmentation

ACM Reference Format:
Ethan Marquez*, Max Faykus, Oyinlolu Odetoye, Melissa C. Smith, and Jon C. Calhoun. 2025. Mitigating I/O Bottlenecks in LiDAR Pipelines by Directly Merging Neural Decompression and Semantic Segmentation. In *Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC '25).* ACM, New York, NY, USA, 2 pages. https://doi.org/XXXXXXX.XXXXXXX

SC '25, St. Louis, MO

ACM ISBN 978-x-xxxx-xxxx-x/YYYY/MM
https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The fast growth of high-resolution LiDAR data creates a significant I/O bottleneck in large-scale analysis and high-performance computing (HPC). This is a critical issue for multi-step pipelines, where point clouds are used sequentially and must be rewritten as files, creating a significant I/O bottleneck.

This work introduces a novel, unified framework that eliminates this costly redundant storage by directly integrating a RENO-based neural autoencoder with a Point Transformer v3 (PTV3) segmentation backbone. We designed this end-to-end model to leverage the high rank feature tensors of the RENO decoder as direct input for the segmentation backbone, bypassing the need to write and read point cloud files from the German Outdoor and Offroad dataset (GOOSE).

This approach enables semantic analysis directly on compressed data, significantly reducing I/O overhead and end-to-end processing time while aiming to maintain segmentation accuracy. This is crucial for real-time processing on autonomous vehicles, where space is constrained and a fast response time is required.

2 Background

We used RENO, a neural network autoencoder, as the compressor in this work due to its small size and ability to achieve real-time compression/decompression [4].

Point Transformer V3 (PTV3) is used in this work for LiDAR segmentation as it is fast and scalable [3] built on the modular framework of Pointcept [1].

The German Outdoor and Offroad dataset (GOOSE) [2] was chosen for this study due to being a new dataset with documentation and organized labeled point clouds.

3 Methodology

Our work compared two distinct processing pipelines on the GOOSE dataset. The baseline was a traditional, unmerged pipeline where the RENO autoencoder and PTV3 operated sequentially, requiring costly intermediary file I/O between the compression and segmentation steps. In our novel merged pipeline, we eliminated this I/O bottleneck by directly feeding the high rank feature tensors from the RENO decoder to the PTV3 segmentation backbone.

This approach addresses a key challenge in merging these architectures: RENO's decoder outputs a rank 32 feature tensor to

Figure 1: Comparison of Semantic Segmentation Accuracy: The merged pipeline (Green) achieves comparable semantic segmentation results to the unmerged, file-based pipeline (Red), demonstrating I/O savings without a loss in accuracy.

represent each point, while the GOOSE dataset originally uses a rank 4 tensor (x, y, z, and intensity). Our merged architecture downsamples this high rank feature tensor to one dimension and directly leverages it to bypass the need to save and reload the data in its decompressed format. After decompression, we use a K-Nearest Neighbors (KNN) algorithm within the dataloader to adjust labels to the dimensions of the decompressed point clouds, ensuring proper training of PTV3 without manual data regeneration.

The performance of both pipelines was measured using mean Intersection over Union (mIoU) to confirm that our merged framework maintains segmentation accuracy while providing substantial I/O savings.

4 Results

Merging RENO and PTV3 reduces the required storage to handle compression and segmentation of large point cloud datasets and completely removes the need for intermediary files or processes, which would take up space and processing time, without any cost to the accuracy of semantic segmentation as seen in Fig 1.

Directly using decompressed feature tensors rather than writing to intermediary files saves 29.9 GB for 13,076 point clouds. This means with a 20 HZ LiDAR system, we save 2.7 GB every minute the LiDAR is running.

Lossy compression of point clouds using RENO reduces the size of the data from 29.9 GB to 1.2 GB, while keeping relevant features for downstream tasks such as semantic segmentation.

Acknowledgments

Thank you to the College of Engineering, Computing and Applied Sciences as well as Dr. Smith for her creative inquiry Machine Learning and Big Data. This work was supported in part by the Clemson University Creative Inquiry + Undergraduate Research Program. This research used in part resources on the Palmetto Cluster at Clemson University under National Science Foundation awards MRI 1228312, II NEW 1405767, MRI 1725573, and MRI 2018069. The views expressed in this article do not necessarily represent the views of NSF or the United States government.

References

[1] Pointcept Contributors. 2023. Pointcept: A codebase for point cloud perception research. https://github.com/Pointcept/Pointcept

[2] Peter Mortimer, Raphael Hagmanns, Miguel Granero, Thorsten Luettel, Janko Petereit, and Hans-Joachim Wuensche. 2023. The GOOSE Dataset for Perception in Unstructured Environments. arXiv:2310.16788 [cs.CV] https://arxiv.org/abs/2310.16788

[3] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. 2024. Point Transformer V3: Simpler, Faster, Stronger. arXiv:2312.10035 [cs.CV] https://arxiv.org/abs/2312.10035

[4] Kang You, Tong Chen, Dandan Ding, M. Salman Asif, and Zhan Ma. 2025. RENO: Real-Time Neural Compression for 3D LiDAR Point Clouds. arXiv:2503.12382 [cs.CV] https://arxiv.org/abs/2503.12382