GCNN-based Loop Closure Detection in Large-Scale SLAM

Srushti Hippargi, Shrey Shah

***Abstract*—Efficient loop closure detection is critical for SLAM in large-scale environments. This work introduces a novel approach leveraging cyclic group convolutional neural networks (GCNNs) for deep loop closure detection. By integrating GCNNs after compressing 3D features into Bird's Eye View (BEV) representations, our method captures symmetry-invariant and rotationally robust features significantly improving detection performance in environments with repetitive or rotationally similar structures. Through this integration, the system achieves lower translation errors and comparable rotational errors, demonstrating robustness and scalability in diverse SLAM scenarios. The code is available in the provided link: https://github.com/Shrey-2303/LCDNet.**

I. Introduction

Simultaneous Localization and Mapping (SLAM) is a fundamental capability for autonomous systems operating in dynamic and complex environments. Loop closure detection, an essential component of SLAM, plays a critical role in reducing accumulated drift and ensuring the consistency of generated maps. However, traditional loop closure detection methods often struggle in challenging scenarios, such as environments with repetitive patterns or rotationally symmetric structures, where distinguishing unique locations becomes difficult. As the demand for robust and scalable SLAM solutions grows, there is a need for advanced methods that can handle these challenges effectively.

The shortcomings of traditional SLAM systems that use feature extractors such as LCDNet's PVRCNN often struggle with reverse loop closures, leading to map correction errors and increased drift due to ambiguities in loop detection. In this work, we integrate GCNNs in BEV feature space taking advantage of symmetry-invariant properties to capture spatial and rotational information. This architecture processes dense BEV feature maps, enabling robust loop detection in environments with severe reverse loop closures. Our approach achieves improved robustness, lower translation errors, and comparable rotational errors.

II. Related Works

A. Simultaneous Localization and Mapping (SLAM)

SLAM [1] is a real-time technique for creating maps of unknown environments while simultaneously determining a robot's position using sensor data like cameras, LiDAR, or range finders. It is essential for autonomous navigation, allowing robots to operate in areas without pre-built or up-to-date maps.

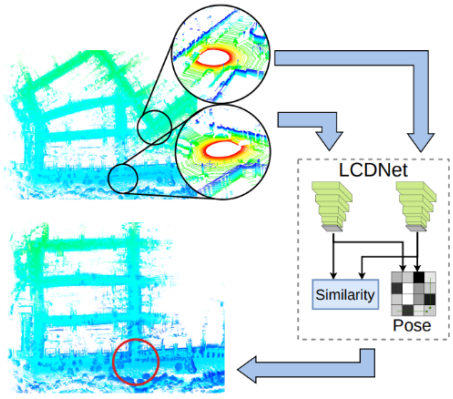

Fig. 1: 2D gloabal map while performing SLAM

B. Hand crafted feature extraction in SLAM

SLAM systems rely on feature extraction to identify and match key points across sensor inputs, enabling tasks like tracking, mapping, and loop closure detection. Traditional handcrafted feature extraction techniques such as SIFT (Scale-Invariant Feature Transform) [2] and ORB (Oriented FAST and Rotated BRIEF) [3] are used in most systems. ORB, widely used in systems like ORB-SLAM [4], is known for its efficiency. These methods rely on predefined algorithms to extract features invariant to scale, rotation, and illumination. However, their sensitivity to environmental variations like lighting changes, occlusions, and repetitive patterns often limits performance in dynamic or complex environments.

C. Deep loop closure detection

Neural network-based approaches extract discriminative global descriptors from sensor data, improving place recognition even in challenging scenarios like reverse loops. PointNetVLAD [5] is a pioneering method that combines PointNet architecture with NetVLAD to encode 3D LiDAR point clouds into global descriptors for

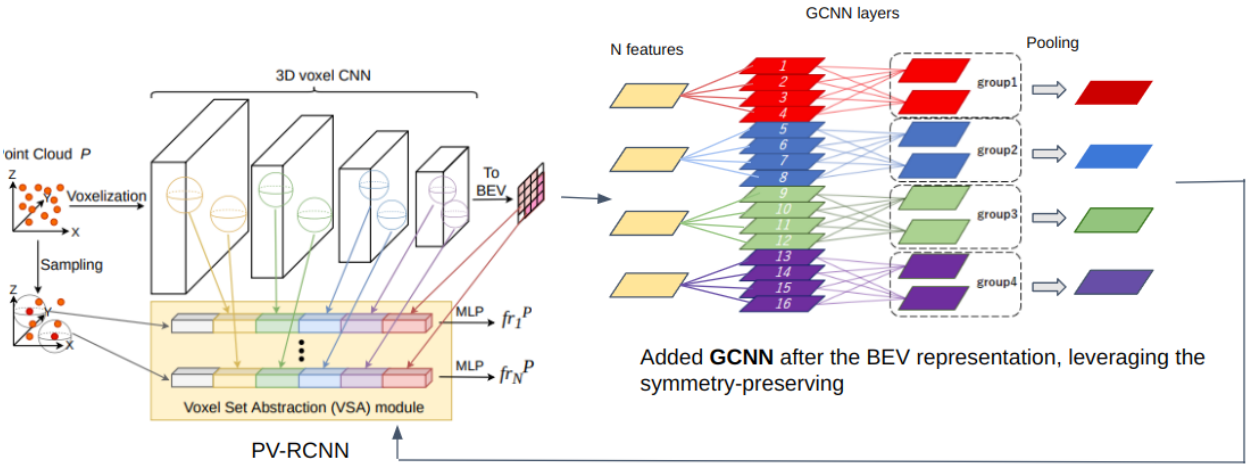

Fig. 2: LCDNet's PV-RCNN and GCNN Architecture

loop closure detection. OverlapNet [6] further enhances performance by directly estimating the overlap between LiDAR scans, ensuring accurate loop closures. LCDNet [6], a more recent approach also enables differentiable point cloud registration. These advancements open up a new path for loop closude detection.

D. Symmetry and GCNNs in SLAM

Group convolutions, using symmetrical groups like C_4 (cyclic group) and P_4 (Dihedral group), embed rotational and reflectional invariance directly into neural networks. These methods, such as Group Equivariant CNNs (G-CNNs) [7], ensure feature consistency under transformations, crucial for tasks like loop closure detection in SLAM. By leveraging symmetry, networks can improve robustness to viewpoint changes and occlusions.

III. Approach

A. LCDNet Architecture

LCDNet is a deep learning architecture for loop closure detection and point cloud registration in LiDAR-based SLAM, consisting of a shared feature extractor, a place recognition head, and a relative pose estimation head. The shared feature extractor uses triplet loss to learn discriminative features from anchor (P_a), positive (P_p), and negative (P_n) point clouds.

The place recognition module leverages NetVLAD to aggregate local features into global descriptors for comparing locations. The relative pose estimation head calculates 6-DoF transformations by matching keypoints via *Unbalanced Optimal Transport (UOT)* by relaxing full correspondence assumptions, enabling pose estimation from soft correspondences.

While effective, this approach does not inherently account for rotational symmetries in the data, motivating the integration of Cyclic GCNN.

B. Overview of PV-RCNN in LCDNet

The PV-RCNN architecture in LCDNet is adapted to extract discriminative keypoint features. The input is a point cloud $P \in \mathbb{R}^{J\times 4}$, and the output is a set of N keypoint features $F_R^P = \{f_1^P, \ldots, f_N^P\}$, where $f_i^P \in \mathbb{R}^D$.

C. Feature Extraction Process

- Voxelization: Converts the raw point cloud into a 3D voxel grid ($L \times W \times H$) by averaging point features.
- 3D Sparse Convolutions: Extracts hierarchical features through four pyramid levels ($1\times$, $2\times$, $4\times$, $8\times$), balancing fine details and high-level context.
- BEV Feature Map: Converts the coarsest feature map to a 2D BEV representation by stacking features along the z-axis for large-scale spatial reasoning.
- Voxel Set Abstraction (VSA) aggregates features from pyramid levels, BEV maps, and raw point clouds by selecting keypoints via farthest point sampling, combining neighbor voxel features, and encoding them with MLPs and max-pooling to generate keypoint features.

Final aggregated features are cmobined and a final MLP produces discriminative keypoint features.

$$f_{3D}^i = [f_{pv}^i, f_{\text{raw}}^i, f_{\text{bev}}^i],$$
$$f_i^r = \text{MLP}(f_{3D}^i).$$

This process efficiently integrates voxel grid, BEV, and raw point cloud features for loop closure detection and pose estimation.

D. Group Convolution Neural Networks GCNN

We incorporate Cyclic Group Convolutional Neural Networks after the BEV representation stage for pre-

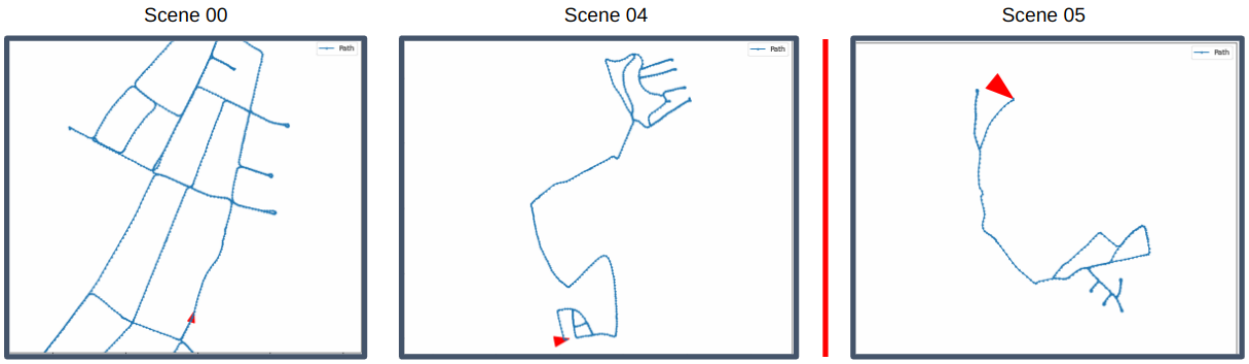

Fig. 3: Scenes trajectories for Kitti360 dataset

serving symmetry in repetitive or symmetric structures.

Group convolution extends standard convolution to include not only translations but also transformations from a broader group G. For example, in the cyclic group $H = C_4$, group convolutions consider 90-degree rotations.

Cyclic-4 Group for GCNN: The C_4 group represents discrete rotations of $0°$, $90°$, $180°$, and $270°$, making networks equivariant to these transformations.

***C_4* in LCDNet:** Applying C_4 GCNNs to BEV maps ensures equivariance to $90°$ rotations, enabling robust loop closure detection despite rotational errors.

Implementation of C_4-Convolutions:

- **Filter Transformation:** A base filter K is rotated to produce $K_0, K_{90}, K_{180}, K_{270}$.
- **Feature Map Update:** Each K_i generates feature maps $F_0, F_{90}, F_{180}, F_{270}$, which are stacked for equivariance or pooled for invariance.
- **Pooling for Invariance:** Rotation-invariant features are obtained via:

$$F_{\text{invariant}} = \max(F_0, F_{90}, F_{180}, F_{270}).$$

Sparse 3D convolutions [8] efficiently process LiDAR voxel grids by skipping empty voxels and capturing details across resolutions ($1\times$ to $8\times$). While GCNNs [7] excel at encoding rotational and reflective symmetries, they are inefficient for sparse data and are better suited for 2D Bird's-Eye-View (BEV) representations, obtained by compressing 3D voxel features along the Z-axis. Applying GCNNs to BEV maps encodes symmetries while reducing computational cost. These GCNN-processed features, integrated into the VSA module [9], produce symmetry-invariant keypoint features for tasks like loop closure detection and pose estimation.

IV. Results

A. Implementation and Training details

To verify the the fidelity of our algorithm we compare results on the KITTI360 dataset designed for autonomous driving for various tasks like SLAM, object detection, Instance Segmentation and much more. The original LCDNet was trained on the entire dataset with 150 epochs using 4 GPUs of 64 GB with a batch size of 4 each (total 24). The original pipeline voxelized the Lidar data into voxels of size [1408 x 1408 x 40]. Due to huge amounts of derived grid size, we had to heavily downsample the data by 4 times in spatial dimension resulting in a size of [351 x 351 x 40].

The C4 required 4 group kernals and an increase in corresponding feature space dimension to maintain the architecture we deviated from the BEV layer to downsample the BEV features to $f/4$ which then increased to f after the group operation before being passed on to the rest of the piepline Due to limited time and resources we only performed experiments with varying GPUs and limit the number of epochs upto 10 for each test. Each training took roughly 10-20 hours to train based on the batch size and GPU.

B. Experiments

For training the model to detect loop closure, we used the scenes 0 and 4 for training and the scenes 5 for validation and testing as showin in []. To remain fair about the comparison, we first created the same baseline results for 10 epochs with all original settings with the same number of scenes in both experiments to compare our results for the same number of epochs. Similar to the original paper we used 4 main metrics to evaluate our GCNN addition for comparison.

- **Max F1**: Measures the harmonic mean of precision and recall at the threshold where their values are balanced (highest F1 score).
- **Validation Recall**: The fraction of correctly identified true positives (loop closures) over the total actual positives during validation.
- **Translational Error**: Euclidean distance between 2 aligned point clouds.
- **Rotational Error**: The angular difference between the ground truth and aligned orientated point clouds.

TABLE I: Performance comparison across different experiments with Baselines and Ours.

Method	Ours				Theirs			
	TE	RE	Max F1	Val. Recall	TE	RE	Max F1	Val. Recall
Downsampled	**4.70**	5.99	**0.754**	**90.66**	4.85	**4.93**	0.744	88.68
Scenes 02 and 04	**3.51**	5.10	**0.754**	**92.06**	4.21	**4.20**	0.728	90.36
Original grid	**3.84**	**4.13**	**0.721**	83.44	3.90	4.80	0.668	**87.42**
Group order, both scenes								
Order 4	**3.51**	5.10	0.754	92.06				
Order 8	4.28	**4.65**	**0.775**	**93.86**				

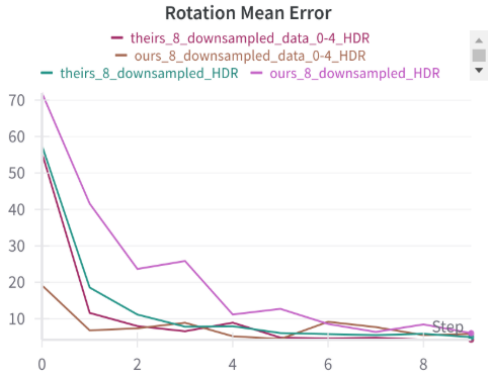

Fig. 4: Rotational mean Error for downsampled and increased data size

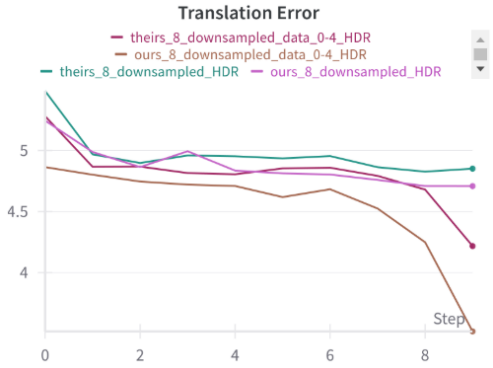

Fig. 5: Translational Error for downsampled and increased data size

The table summarizes our experiments, testing various configurations, including batch size adjustments, removing downsampling, and increasing data by 10%. Adding a GCNN layer in the BEV space after sparse feature extraction significantly preserved rotationally invariant features.

Our method consistently outperformed the baseline in Translational Error (TE) and validation metrics like Max F1 and Recall but slightly underperformed in Rotational Error (RE). We attribute the higher RE to feature loss caused by additional downsampling for group convolution, impacting rotational consistency.

Increasing the group order improved RE by enhancing rotational equivariance but slightly increased TE. This trade-off occurs as the network prioritizes rotational features over translation-sensitive ones, reflecting the inherent limitations of group equivariance for translations.

C. Integration with Lio-SAM

Deep loop detection aligns the point clouds to get the transformation between current and the slow accumulated drift between the entire global map. To test the complete performance similar to the paper, we need to integrate the loop closure detection with any robust lidar based SLAM methods like Lio-SAM for success rates for the entire trajectory. Unfortunately the integrated code to scommunicate with ROS architecture for Lio-SAM needs to be written from scratch which we werent able to complete on time. This is one of the future extentions for the current state of the project.

V. Conclusion

In this project we present Deep loop closure detection using Group convolution Neural Networks specifically to Cyclic groups for rotational invariance. Our extension in the feature extraction module in the PV-RCNN architecture of the original paper We were able to improve the baseline results with equal restrictions on both the methods. The resulting translational error for the corrected loops was lower than original method. For future improvements, We propose using more suited groups and using complete equivalent pipeline for further improvement in the 3D voxel space and finally fusing the detection algorithm with a SOTA complete SLAM system

REFERENCES

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part i," *IEEE Robotics Automation Magazine*, vol. 13, no. 2, pp. 99–110, 2006.

[2] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[3] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 2564–2571, 2011.

[4] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "Orb-slam: A versatile and accurate monocular slam system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[5] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4470–4479, 2018.

[6] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "Overlapnet: Loop closing for lidar-based slam," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1968–1975, 2021.

[7] T. S. Cohen and M. Welling, "Group equivariant convolutional networks," in *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 2990–2999, 2016.

[8] S. Contributors, "Spconv: Spatially sparse convolution library." https://github.com/traveller59/spconv, 2022.

[9] D. Cattaneo, M. Vaghi, and A. Valada, "Lcdnet: Deep loop closure detection and point cloud registration for lidar slam," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 2370–2377, 2022.