Grad-Lidar-SLAM: Fully Differentiable Global SLAM for Lidar with Pose-Graph Optimization

Aryan[*1], Dheeraj Vattikonda[*2], Erqun Dong[3,4], Sabyasachi Sahoo[3,5]

***Abstract*— While Lidar-based SLAM systems are critical for autonomous vehicles, most existing methods are non-differentiable. This limits their use with deep neural networks to learn valuable representations. In this work, we propose Grad-Lidar-SLAM, a novel, fully differentiable SLAM framework for Lidar. We show its effectiveness by using it to solve point cloud completion task. We also show that when adapted for Lidar-based SLAM, existing differentiable baselines fail to perform well on real-world datasets. We address this problem by proposing a novel differentiable pose graph optimization framework. Our experiments on real-world datasets show that our proposed approaches outperform the baselines in static and dynamic environments.**

I. Introduction

Gradient descent and back-propagation have transformed the outlook of several domains with the successful application in deep learning (e.g. computer vision, natural language processing). Such success can also benefit other areas, including simultaneous localization and mapping (SLAM). If the SLAM modules are fully differentiable, then it has the advantage of being integrated into deep learning frameworks so that the deep learning tasks can benefit from some self-supervision from SLAM. However, the pipelines of classical SLAM systems [1], [2] are non-differentiable, especially the non-linear optimization. This disadvantage is because we cannot it stack with many state-of-the-art neural nets.

Making fully differentiable SLAM has thus become of interest in the research field. The pioneering work of GradSLAM (∇SLAM) [3] introduces the gradient-based approach to SLAM problems, which solves the dense visual SLAM problem using a fully differentiable formulation that iteratively optimizes the states with gradients from back-propagation. While GradSLAM tackles the non-differentiable functions in SLAM with softening strategies to make them differentiable, it is a local SLAM which drifts over time, and there is a need to use global SLAM and adopt differentiable global SLAM.

In this work, we tackle the differentiability issue of Lidar SLAM. Lidar has the advantages of high accuracy, low calculation volume, and easy-to-realize real-time SLAM. Lidar actively collects point clouds of the environment, less affected by environmental conditions like light and rain. Most differentiable SLAM method mentioned above works only for cameras, and lidar-based SLAM systems are ignored. Furthermore, simply adapting them for lidar systems is not good enough.

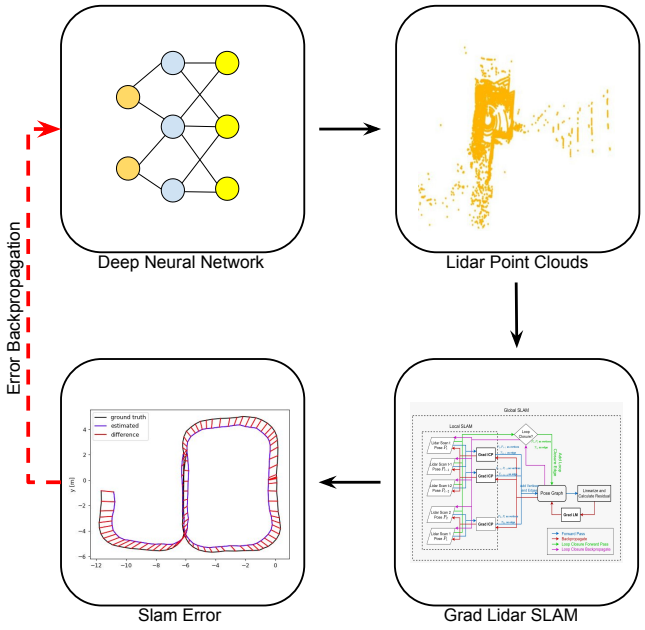

Fig. 1. Grad-LiDAR SLAM can be used end-to-end with upstream deep learning tasks that wish to minimize SLAM error. **Top**: Output pointclouds from deep neural networks are passed to the proposed Grad-Lidar SLAM. **Bottom**: SLAM error is propagated back to neural networks and used for its training.

To address these problems, we build a lidar-based differentiable pose-graph SLAM, named Grad-Lidar-SLAM (∇L-SLAM), a global SLAM that maintains and optimizes the global pose-graph. We show its effectiveness by solving the point cloud completion task using the pipeline described in Fig. 1. Different from the strategy of GradSLAM, this method can achieve less drift. We summarize the main contributions of this paper as follows:

- We propose a novel fully-differentiable lidar based SLAM framework called Grad-Lidar SLAM. We demonstrate its effectiveness by using it to solve the problem of Point Cloud Completion using the gradients backpropagated via our differentiable SLAM framework.
- We also propose a differentiable pose-graph optimization framework to build a fully differentiable global SLAM. We show that our differentiable global SLAM significantly outperforms existing differentiable SLAM

[1] Delhi Technological University, India
[2] National Institute of Technology, Hamirpur, India
[3] Mila - Quebec AI Institute, Canada
[4] McGill University, Montreal, Canada
[5] IID, Université Laval, Quebec City, Canada
[*] equal contribution

systems on real-world datasets.

II. Background

We define $x = (x_1, ..., x_T)^T$ to be a vector of parameters, where x_i is pose of the node i, z_{i_j} and Ω_{i_j} are the mean and the information matrix of the measurement between node i and node j. This measurement acts as transformation which makes the observation obtained from i maximally overlap with the observation from j. Let $z_{i_j}(x_i, x_j)$ be the prediction of the measurement given the nodes x_i and x_j. The log-likelihood l_{i_j} of a measurement z_{i_j} is

$$l_{i_j} \propto [z_{i_j} - z_{i_j}(x_i, x_j)]^T \Omega [z_{i_j} - z_{i_j}(x_i, x_j)] \tag{1}$$

Let $e(x_i, x_j, z_{i_j})$ be the error function which calculates the difference between the expected observation and the real observation.

$$e_{i_j}(x_i, x_j) = z_{i_j} - z_{i_j}(x_i.x_j) \tag{2}$$

The goal of a maximum likelihood approach is to find the configuration of the nodes x^* that minimizes the negative log-likelihood F(x) of all observations

$$F(x) = \sum e_{i_j}^T \Omega_{i_j} e_{i_j}. \tag{3}$$

Thus, it seeks to solve the following equation

$$x^* = \arg\min_x F(x) \tag{4}$$

The proposed approach utilizes a differential Levenberg-Marquardt algorithm, and with a good initial guess x of the robot poses, we can obtain a good numerical solution. The idea is to approximate the error function using first-order Taylor expansion around the current initial guess x

$$e_{i_j}(x_i + \Delta x_i, x_j + \Delta x_j) \approx e_{i_j} + J_{i_j} \Delta x \tag{5}$$

$$\begin{aligned} F_{i_j}(x + \Delta x) &= e_{i_j}(x + \Delta x)^T \Omega_{i_j} e_{i_j}(x + \Delta x) \\ &\approx c_{i_j} + 2 b_{i_j} \Delta x + \Delta x^T H_{i_j} \Delta x \end{aligned} \tag{6}$$

With the local approximation, we can rewrite the function F(x) as

$$\begin{aligned} F(x + \Delta x) &= \sum F_{i_j}(x + \Delta x) \\ &= c + 2 b_{i_j} \Delta x + \Delta x^T H \Delta x \end{aligned} \tag{7}$$

We can solve the above quadratic form and find the minimum Δx by solving the linear system $H \Delta x^* = -b$

LM optimizer uses μ as damping parameter $(J^T J + \mu I) x^* = -g$ Where damping parameter has several effects :

1) For all $\mu > 0$, the coefficient matrix is positive definite, ensuring that x^* is a descent direction.
2) For large values of μ, we get $x^* \approx -\frac{1}{\mu} g$ is a short step in the steepest descent direction. This is good if the current iterate is far away from the solution.
3) If μ is very small, then $x^* \approx x^*_{g_n}$, which is a good step at the final stages of the iteration, when x is almost close to x^*.

III. Methodology

A. Differentiable LiDAR-based SLAM

The main objective of ∇L-SLAM is to make every computation in pose-graph SLAM a composition of differentiable functions so that we can solve the whole global SLAM problem via backpropagation. The sequence of operations in our system can be termed odometry estimation, pose-graph building and global optimization.

We build our system with the GradSLAM framework [3]. Fig. (2) illustrates the pipeline of our proposed approach. Since Lidar scan range is limited and cannot grow indefinitely with the map, it is evident that the constraints between consecutive nodes exist. Therefore, we first realize a functioning front-end odometry that takes every pair of consecutive nodes and outputs a cumulative trajectory. To achieve this, we adopt the differentiable iterative closest point module, ∇ICP in GradSLAM, utilizing the point cloud matching between the consecutive poses to estimate the current pose.

B. Differentiable Pose Graph Optimization

One major challenge is that as time accumulates, the length and scale of the trajectory will become longer, leading to the drift in the trajectory, and the global map will become inconsistent.

To solve this problem, we introduce a differentiable pose-graph SLAM approach which uses a global optimizer to resolve the inconsistency in the trajectory. We save the trajectory and construct a back-end global pose-graph optimization [4] to reduce the errors. The vertices to be optimized in the pose graph are the poses of the agent at each Lidar scan. Most state-of-the-art SLAM systems optimize the likelihood of all the constraints in the graph to obtain local/global consistent estimates of the robot state, which is equivalent to optimizing the nonlinear least squares objectives in Eq. (3). Such objectives are of form $\frac{1}{2} \sum r(x)^2$, where $r(x)$ is a nonlinear function of residuals. The ordinary edge comes from point clouds matching between consecutive nodes and loop edges come from the rematch of the point clouds during loop closure detection.

The objective functions are first linearized and then solved using the Levenberg-Marquardt (LM) solver. We first implemented the Gauss-Newton (GN) solver in our code. However, while the GN solver is differentiable, due to the data problem's potential degeneration, which makes the linear system ill-conditioned [5], it does not provide convergence guarantees. So we use the LM solver instead to ensure numerical stability. However, the trust region is not differentiable as it involves calibration of the parameters based on a lookahead operation over the next iterate and discretely switching between damping or undamping the linear system. This discrete switching does not allow the gradient to flow backwards. So we used soft reparametrization of the damping mechanism as in ∇LM [3] with

$$\lambda = \lambda_{min} + \frac{\lambda_{max} - \lambda_{min}}{1 + De^{-\alpha(r_1 - r_0)}}. \tag{8}$$

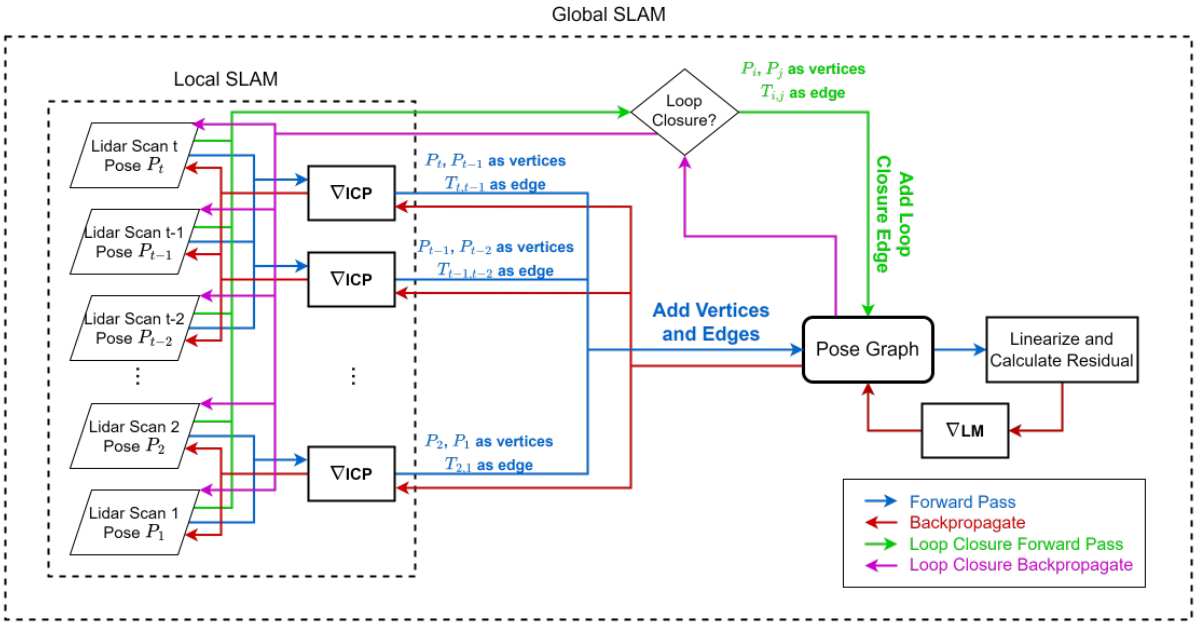

Fig. 2. The pipeline of our ∇L-SLAM. It is a globally differentiable SLAM system compared to local SLAM like GradSLAM. The pose of the Lidar scans are the nodes to optimize in the global pose graph. They first go through ∇ICP [3] to provide pose-pose constraints in the pose graph. Then the global pose graph optimization is formulated considering all pose-pose constraints and the loop closure constraints. The update is calculated using ∇LM [3], a differentiable version of the Levenberg-Marquardt solver, and the update is backpropagated to the poses both through ordinary pose-pose constraints and loop closure constraints.

Fig. 3. We use our proposed Grad-Lidar SLAM for the Point Cloud Completion task. **Left**: randomly initialized point cloud, **Middle**: final reconstructed point cloud, learnt using gradients on SLAM pose error, **Right**: ground truth point cloud. We observe that the reconstructed point clouds are quite close to the ground truth point clouds.

Timestep	Grad value of GN	Grad value of diff-LM
0	117.81	31.32
1	62.39	7.66
2	624.96	2.89
3	1980.36	1.94
4	23729.50	1.58

TABLE I

COMPARISON OF THE GRADIENTS INDICATE EXPLODING GRADIENT PHENOMENON IN A PARTICULAR ITERATION.

This smooth parameterization of the trust region makes LM solver differentiable and allows the gradient to pass backwards.

To demonstrate our choice of the ∇LM solver over the ∇GN solver, we show the L-2 norm of the gradient in some iterations of the pose graph in Table I. GN is based on a linear approximation of objective function $f(\boldsymbol{x})$ in the neighbourhood of $\hat{\boldsymbol{x}}$. GN solver converges quadratically when the first-order Taylor series gives a sufficiently good approximation of the objective function of f. If not, then the algorithm diverges from the optimal value. GN experienced exploding gradient problem where the solver started to diverge, which led to the collapse of the global SLAM, whereas LM stays good numerical stability.

IV. EXPERIMENTS

In this section, we present the quantitative and qualitative results for comparing lidar-based SLAM adopted from Grad-SLAM and the proposed Grad-Lidar-SLAM. Furthermore, we also present the results showing the application of the framework in the point cloud completion task.

A. Results

To show the potential of the proposed framework, we compare it with GradSLAM [3]. GradSLAM was originally proposed for images. We adapt its ICP module and use it for lidar SLAM. We present results on two matrics: ATE (Absolute Trajectory Error) and RPE (Relative Pose Error). We ideally want the method to work seamlessly in real-world conditions. For this purpose, we use the ARD dataset [6] for our evaluation purposes. ARD is a real-world dataset having both static and dynamic runs. There is no moving object in the environment for static runs, and for dynamic runs, there are moving objects in the same environment. The dataset also contains ground truth poses for every lidar scan.

Table II and Table III present the quantitative results for comparison of GradSLAM and our approach for static and

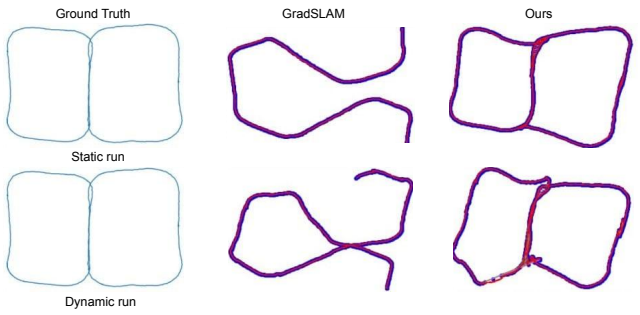

Fig. 4. SLAM Trajectory comparison between our proposed approach with the baseline. **Top**: ARD static run, **Bottom**: ARD dynamic run. While baseline approaches significantly deteriorate over time in the real-world dataset, our proposed approaches can recover the ground truth trajectory.

No. of Scans	Ours		GradSLAM [3]	
	ATE	RPE	ATE	RPE
150	**0.30**	4.89	0.75	**4.30**
300	**0.68**	10.54	1.05	**9.66**
450	**0.65**	**10.23**	2.73	10.71
600	**0.61**	**9.55**	4.55	10.01
700	**0.56**	**9.08**	5.15	10.54

TABLE II
COMPARISON ON ARD STATIC RUNS

dynamic runs, respectively. The results are presented for the different number of scans taken as input. It can be seen from both the tables that ATE, and RPE score for our approach surpasses GradSLAM in most of the cases. Furthermore, it can also be observed that with the increasing number of lidar scans taken, the difference between ATE of GradSLAM and our approach increases exponentially, showing the effectiveness of the proposed pose graph optimizer.

Figure 4 shows the qualitative results for comparison of GradSLAM and our approach for static and dynamic runs. It is visible that GradSLAM cannot identify the loop in both cases, whereas our approach can identify loop closure in both cases. Furthermore, the obtained map from GradSLAM is far from the ground truth as opposed to our approach, which gives a reliable map.

B. Learning through SLAM error

To show the effectiveness of the proposed differentiable Lidar SLAM and pose graph optimization, we conduct experiments involving the backward flow of gradients through the differentiable lidar SLAM back to the point clouds, which are further used for learning. Specifically, we present results on the point cloud completion task, shown in Fig. 3. The task involves multiple ground truth point clouds from the ARD dataset with one point cloud replaced with uniform noise, which is passed through the SLAM pipeline to get the map and relative poses. We supervise it using a ground truth map with an original point cloud in place of the uniform noise discussed above. Using mean square error as a loss function, we successfully retrieved the original point cloud, which shows the potential of the proposed approach. Similar to the above-described experiment we can use use the SLAM-based map loss and pose loss to train deep neural networks in both supervised and unsupervised manner.

No. of Scans	Ours		GradSLAM [3]	
	ATE	RPE	ATE	RPE
150	**0.28**	4.75	0.67	**4.29**
300	**0.51**	9.53	1.22	**9.02**
450	**0.62**	10.99	1.51	**9.87**
600	**0.69**	10.02	2.18	**9.71**
700	**0.75**	**9.08**	2.82	9.15

TABLE III
COMPARISON ON ARD DYNAMIC RUNS

V. RELATED WORKS

Several works have been proposed in literature where SLAM is used in a differentiable manner for deep learning purposes, either directly or indirectly. DROID-SLAM [7] proposes a Dense Bundle Adjustment layer having recurrent iterative updates of camera pose and pixel-wise depth. LEO [8] proposes a method to directly optimize end-to-end tracking performance by learning observation models with the graph optimizer in the loop.

Both CodeSLAM [9] and SceneCode [10] try to express scenes with compact codes representing a 2.5D depth map. DeepTAM trains a tracking network and a mapping network to learn how to rebuild a voxel representation from a pair of photos. Expanding on the well-known monocular SLAM system LSD-SLAM [11], CNN-SLAM uses single-image depth predictions from a convnet. Another recent approach has been to attempt to formulate the SLAM issue over higher-level features such as objects, which may be detected with learnt detectors such as those mentioned in [12], [13], and [14].

Real-time dense visual odometry is carried out using the Lucas-Kanade approach by Kerl et al. [15]. Their system is differentiable and has been utilized widely for the self-supervised depth and motion estimation, as noted in [16]–[18].

VI. CONCLUSION

This work presents a differentiable lidar-based SLAM pipeline along with a differentiable pose-graph optimizer. This framework can be used in many robotics perception applications, such as moving object segmentation and Lidar point cloud completion, that require deep learning tasks to minimize SLAM error. To perform SLAM and pose-graph optimization in an end-to-end differentiable manner, we use differentiable least square optimizers such as Levenberg Marquardt. Our experiments show that the proposed method can retrieve a complete point cloud from a uniform noise which shows the effectiveness and usability of the method.

REFERENCES

[1] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *IEEE transactions on robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[2] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "Dtam: Dense tracking and mapping in real-time," in *2011 international conference on computer vision*. IEEE, 2011, pp. 2320–2327.

[3] J. Krishna Murthy, S. Saryazdi, G. Iyer, and L. Paull, "gradslam: Dense slam meets automatic differentiation," *arXiv*, 2020.

[4] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard, "A tutorial on graph-based slam," *IEEE Intelligent Transportation Systems Magazine*, vol. 2, no. 4, pp. 31–43, 2010.

[5] J. Zhang, M. Kaess, and S. Singh, "On degeneracy of optimization-based state estimation problems," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 809–816.

[6] P. Kumar, S. Sahoo, V. Shah, V. Kondameedi, A. Jain, A. Verma, C. Bhattacharyya, and V. Vishwanath, "Dynamic to static lidar scan reconstruction using adversarially trained auto encoder," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 3, 2021, pp. 1836–1844.

[7] Z. Teed and J. Deng, "Droid-slam: Deep visual slam for monocular, stereo, and rgb-d cameras," *Advances in Neural Information Processing Systems*, vol. 34, pp. 16 558–16 569, 2021.

[8] P. Sodhi, E. Dexheimer, M. Mukadam, S. Anderson, and M. Kaess, "Leo: Learning energy-based models in factor graph optimization," in *Conference on Robot Learning*. PMLR, 2022, pp. 234–244.

[9] M. Bloesch, J. Czarnowski, R. Clark, S. Leutenegger, and A. J. Davison, "Codeslam—learning a compact, optimisable representation for dense visual slam," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2560–2568.

[10] S. Zhi, M. Bloesch, S. Leutenegger, and A. J. Davison, "Scenecode: Monocular dense semantic reconstruction using learned encoded scene representations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 776–11 785.

[11] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *European conference on computer vision*. Springer, 2014, pp. 834–849.

[12] S. Yang and S. Scherer, "Cubeslam: Monocular 3-d object slam," *IEEE Transactions on Robotics*, vol. 35, no. 4, pp. 925–938, 2019.

[13] B. Mu, S.-Y. Liu, L. Paull, J. Leonard, and J. P. How, "Slam with objects using a nonparametric pose graph," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4602–4609.

[14] P. Parkhiya, R. Khawad, J. K. Murthy, B. Bhowmick, and K. M. Krishna, "Constructing category-specific models for monocular object-slam," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4517–4524.

[15] C. Kerl, J. Sturm, and D. Cremers, "Robust odometry estimation for rgb-d cameras," in *2013 IEEE international conference on robotics and automation*. IEEE, 2013, pp. 3748–3754.

[16] R. Garg, V. K. Bg, G. Carneiro, and I. Reid, "Unsupervised cnn for single view depth estimation: Geometry to the rescue," in *European conference on computer vision*. Springer, 2016, pp. 740–756.

[17] R. Li, S. Wang, Z. Long, and D. Gu, "Undeepvo: Monocular visual odometry through unsupervised deep learning," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 7286–7291.

[18] T. Zhou, M. Brown, N. Snavely, and D. G. Lowe, "Unsupervised learning of depth and ego-motion from video," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1851–1858.