Supplementary for SLACK: Attacking LiDAR-based SLAM with Adversarial Point Injections

Anonymous Authors

I. Video Demo

We provide a video demo of SLACK attacked KITTI sequence and its comparison with the original sequence. The video provided in the Supplementary material - SlackAttackedLiDAR-vs-originalLidarSequence.mp4. It is challenging to distinguish the attacked sequence from the original sequence in the video demo.

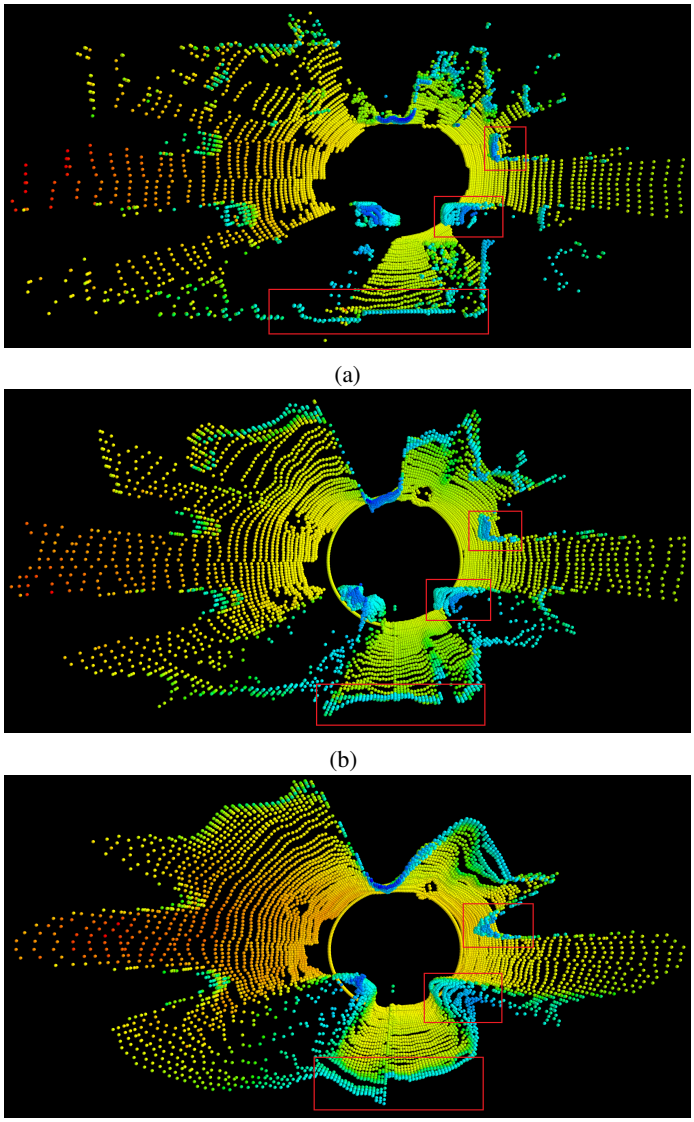
A. Methodology

Details on the segmentation-aware attention-based autoencoder backbone: We employ a binary segmentation mask generated from LiDAR data, distinguishing between stationary and non-stationary points. This mask serves as a prior for reconstructing LiDAR data, particularly focusing on dynamic objects. The binary mask ensures focused attention on dynamic point features, facilitated by the segmentation encoder H_{seg}. Channel-level attention is then applied to the hidden layer features of H_{phi} (the encoder of AE_{mask}) through adaptive average pooling over these features.

Further to ensure that the parameters of H_{seg} are tuned properly to induce proper attention to AE, we model another variant which trains H_{seg} to reconstruct the segmentation mask using Dice coefficient loss. We consider this loss instead of Binary Cross Entropy because dice coefficient loss is shown to work better at class imbalanced segmentation, which is the case here. Dynamic points are quite less in number compared to static points leading to imbalance between the classes. Finally the segmentation encoder (H_{seg}) network receives two signals during backpropagation, one each from the LiDAR autoencoder based MSE loss and the segmentation encoder-decoder network using Dice coefficient loss.

Process of choosing multiple negatives: The LiDAR scans in a sequence are arranged in a contiguous fashion as they would appear while driving. We choose multiple negatives for the anchor as follows - we select the corresponding dynamic scan for the anchor scan as well as a couple of scans ahead and behind it in the driving sequence.

The adversarial module takes heterogeneous pairs (d_j, s_j) as input and produces an adversarial output that assumes that the given input is a homogeneous pair. Among the pair, the segmentation mask (prior) that is fed to the adversarial module can be *manipulated* as follows - The segmentation mask has the same size as the input range image. Portions of the mask are corrupted along the height of segmentation range image. This allows one to control the amount of dynamism and number of the dynamic locations injected into the static LiDAR scan. It discourage the mapping of the static latent



(a)

(b)

(c)

Fig. 1: Comparison of AE_{mask} results with best baseline in high resolution. **Top** - original KITTI scan. **Middle** - reconstruction using AE_{mask} preserves precise static and dynamic structures. **Bottom** - reconstruction using the best baseline.

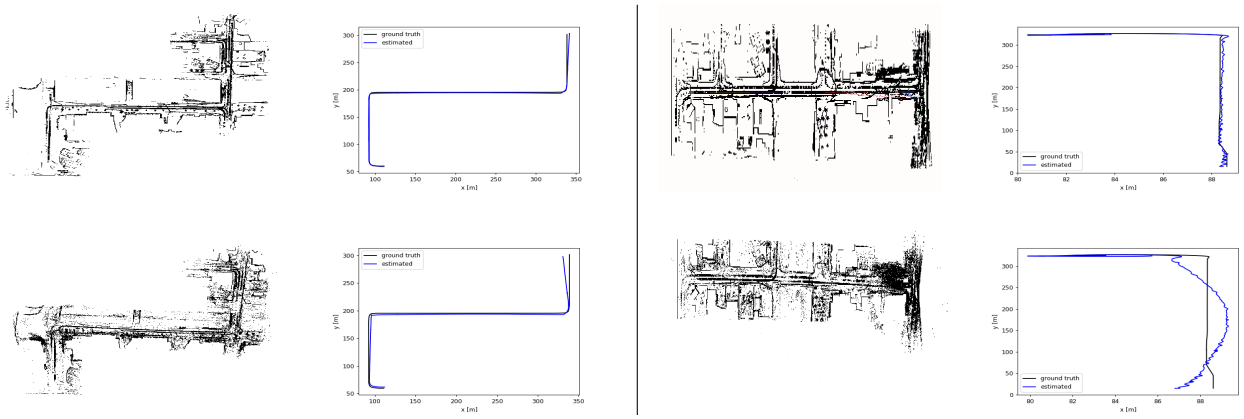


Fig. 2: **(Left)**:SLAM results before and after attack for sequence 1. Top row - SLAM results before attack: the map is clear with distinct structures and details, with minimal to no errors in navigation. Bottom - SLAM results after attack. Map quality and the navigation trajectory are degraded. **(Right)** Similar results for sequence 3 on the right.

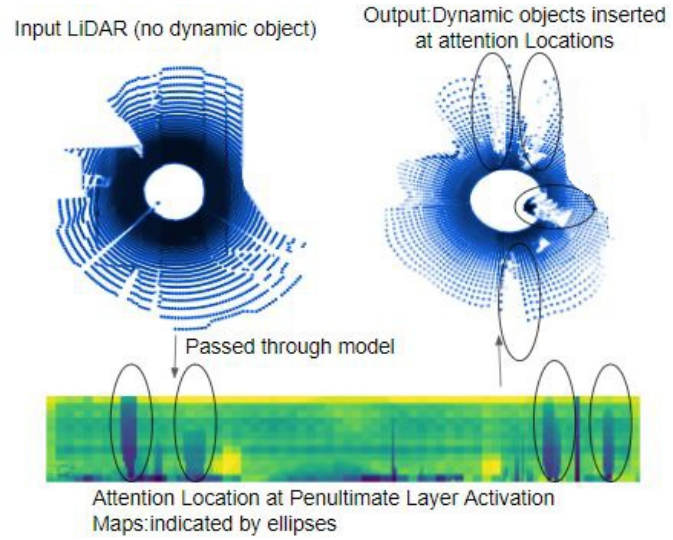


Fig. 3: Effect of segmentation based attention on activation map of SLACK

representation to the same dynamic point on the dynamic manifold and allows for variety in **PiJ**.

II. More SLAM Attack Results

We provide a high-resolution image of Figure 6 in the main paper here (Figure 4). The figure demonstrates the efficacy of AE_{seg} in maintaining the LiDAR quality. This in turn helps SLACK to maintain a desirable LiDAR Quality Index (lower is better) while also injecting LiDAR with *PiJ*.

We demonstrate the effect of SLACK on the two CARLA-64 sequences in Figure 2. We observe that after the *PiJ* attack with less than 1% points, the generated map quality deteriorates significantly. The attacked map for sequence 1 does not accurately map the turns. The attacked map for sequence 2 fails to map some of the regions that are accurately mapped by the un-attacked LiDAR sequence. The trajectory generated after the attack has medium to high deviation from the ground truth trajectory (black color). The attacked trajectory for sequence 2 takes an unwanted turn and deviates significantly from the ground truth trajectory.

It in interesting to note that these degradations in the map and trajectory estimates occur with less than 1% *PiJ*.

We also demonstrate a comparison between the original and attacked LiDAR for the KITTI dataset in Figure 5. It is difficult to distinguish the attacked LiDAR scan from the original scan. For a video demo, please refer to the video in the Supplementary material.

III. Baselines for AE_{mask} and SLACK

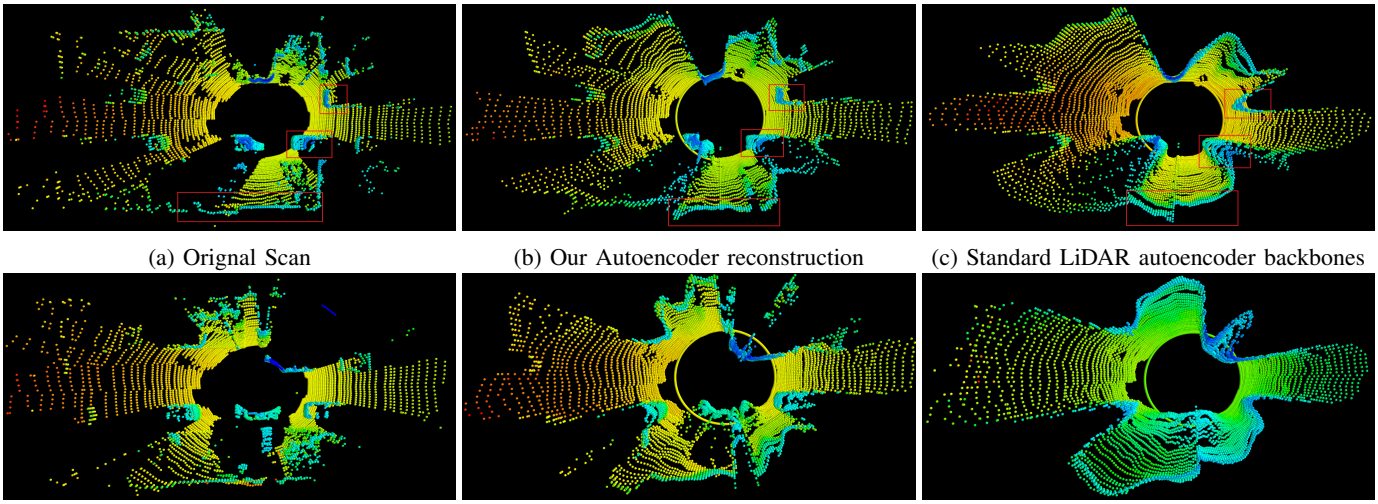
We use LiDAR based Autoencoder backbones that satisfy the following requirements necessary in a SLAM pipeline

- A high generation/sampling rate so that they are integratable in a SLAM pipeline.
- Do not require datasets from modalities other than LiDAR point cloud.
- Do not require manual intervention for *PiJ* attacks.

We use AtlasNet(Groueix et al. [1]), Achlioptas et al., (Achlioptas et. al. [2]), Caccia-AE, Caccia-VAE, Caccia-GAN(Caccia et. al. [3]), DSLR(Kumar et. al. [4]) as baselines based on the above criteria.

Several recent works for generative modelling work very well and generate novel and realistic samples. In order to achieve this, they use data in additional modalities, complex methods (e.g. diffusion models), etc, which generate impressive results but break certain requirements which are necessary for end-to-end SLAM pipelines.

LiDARGen (Zyrianov et. al. [5]) using diffusion models to develop an unconditional generative method for realistic LiDAR generation. Their sampling rate is extremely low to be able to be used in LiDAR SLAM pipelines.



(a) Orignal Scan (b) Our Autoencoder reconstruction (c) Standard LiDAR autoencoder backbones

(d) Original input scan (e) Our autoencoder reconstruction (f) standard LiDAR autoencoder backbones

Fig. 4: Comparison of our Autoencoder module output *w.r.t* a standard LiDAR autoencoder backbones. Column 1 represents original KITTI scan. Column 2 represents the reconstruction using our segmentation-attention assisted autoencoder which is coupled with contrastive learning. Column 3 represents the reconstruction using a standard LiDAR autoencoder backbones. Our autoencoder is able to reconstruct precise static and dynamic structures.

Model	CARLA-64	
	Chamfer	EMD
Caccia-AE	1.52	164
Caccia-AE+TripletLoss	1.30	164
Caccia-AE+N-pairLoss	0.98	139
	1.00	144
AE_{seg}	1.00	153
AE_{seg}+TripletLoss	1.20	157
+TripletLoss	0.94	129
+N-pairLoss	0.93	132
AE_{mask} - (AE_{seg}+N-pair Loss) - **Ours**	**0.93**	**126**

TABLE I: Ablation studies to study the effect of segmentation-aware attention and constrastive losses for AE_{seg}.

Model	LQI		DSR	
	AE_{mask}	Best Baseline Backbone	AE_{mask}	Best w Baseline Backbone
	KITTI-64			
SLACK	**1.97**	3.24	**0.48**	0.45
	CARLA-64			
SLACK	**3.73**	4.48	**0.51**	0.48

TABLE II: Ablation studies to quantify the affect of segmentation aware LiDAR backbone - AE_{mask} on SLACK. For LQI - lower is better. For DSR - higher is better.

UltraLiDAR (Xiong et. al. [6]) learn a compact discrete representation for LiDAR which generates excellent novel samples. However, they require manual intervention for controllable manipulation which is a hindrance for an end-to-end pipeline for *PiJ* attacks on LiDAR.

NeRF-LiDAR (Zhang et. al. [7]) use Neural Radiance Fields for simulation of novel real world LiDAR scans. However, they require multi-view images of a scene apart from LiDAR scans which are not available in our settings.

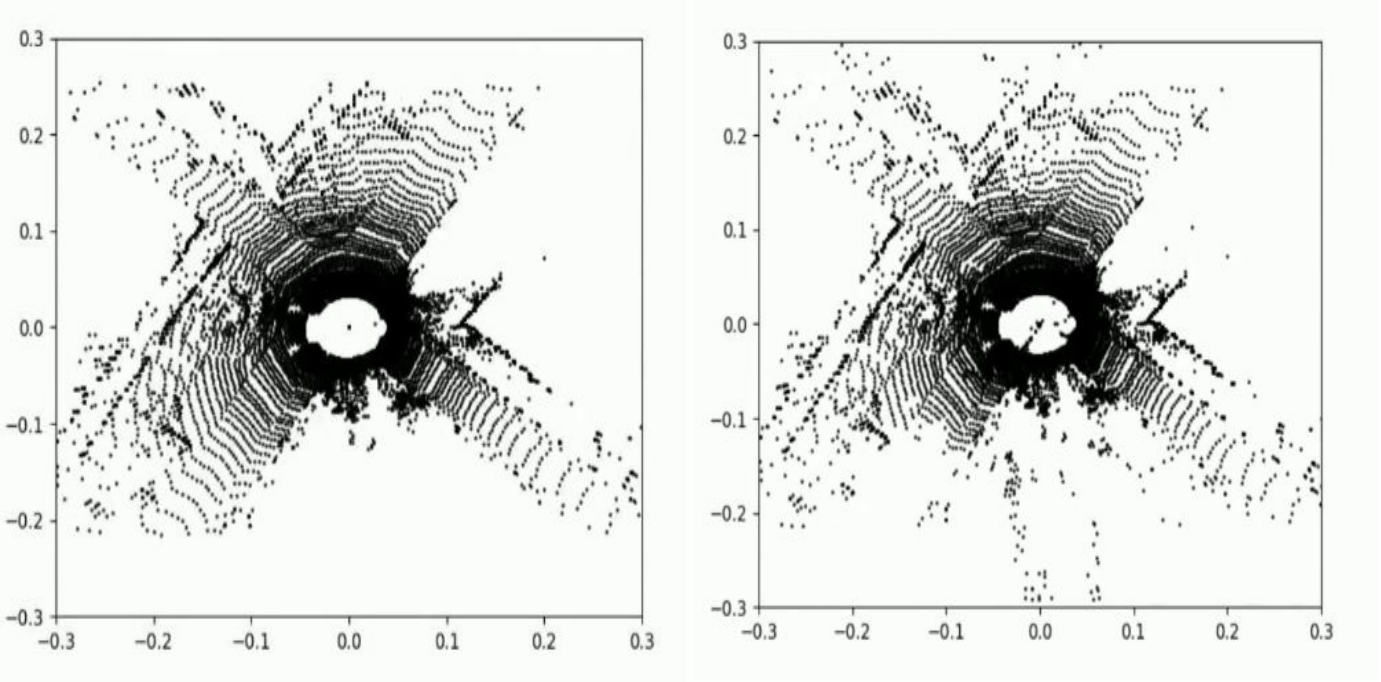
IV. Abation Studies

For AE_{mask}, we use 2 modules - segmentation-aware attention (AE_{seg}) and contrastive learning (Triplet and N-Pair Loss) using hard negatives. We perform Ablation studies to show the effect of these two modules in Table I. In the table, Caccia-AE refers to the best baseline backbone on top of which our segmentation-attention and contrastive module are augmented. We use the Caccia-AE backbone to augment our modifications because this backbone is straight forward to use and integrate our modules and works well with an impressive sampling/inference rate. Our modules can also be easily integrated to other backbones.

We also show that for AE_{mask}, providing segmentation attention using the segmentation encoder to the LiDAR encoder (Figure 5a in main paper) is enough to encode the segmentation prior information into AE_{mask}. The segmentation encoder does not need a decoder to reconstruct the segmentation mask. On the contrary using an encoder-decoder segmentation network to reconstruct the segmentation information () reduces the reconstruction quality. The ablation results in Table I verify this claim.

We further show the benefit of using the AE_{mask} module for SLACK over the best baseline backbone autoencoder in Table II. We observe that segmentation-aware attention and contrastive learning using AE_{mask} helps the network to maintain accurate LiDAR quality while performing *PiJ* (Table I and Figure 5).

We also perform an ablation that demonstrates the importance of using the pretext task based discrimintor as opposed to using a standard discriminator for *DiJ* in Table III.



(a) Original Scan (b) Attacked Scan

Fig. 5: Visual demonstration of an original scan (left) and an attacked scan (right). It is very difficult to distinguish the attacked scan from the original scan. This is the objective of AE_{seg} and SLACK. For a video demo, please refer to the video in the Supplementary.

Dataset	Vanilla Discriminator	SLACK
	LQI/DSR	
KITTI	3.84/0.43	**1.97/0.48**
CARLA-64	5.47/0.50	**3.73/0.51**

TABLE III: Ablation study to show the comparison between attack using a vanilla discriminator and SLACK.

V. Training and Experimental Setup

We provide the details of the training setup for AE_{mask} and SLACK. We train AE_{mask} for 700 epochs. We use an Adam optimizer with an initial leanning rate of 0.001. The pretext task discriminator is trained for 20 epochs using the Adam optimizer with an initial learning rate of 0.0006 and a weight decay of 0.00001. The adversarial module is trained for 200 epochs using the Adam optimizer with an initial learning rate of 0.001 and a decay of 0.00001. For SLACK-MMD we initialize the network with pre-trained weights of AE_{mask} and the pretext task discriminator, PD. The model is trained with KITII data for Domain Adaptation till 200 epochs using Adam optimizer with an initial learning rate of 0.001 and a weight decay of 0.00001.

All models are trained using an NVIDIA RTX-3090Ti GPU with 24 GB of GPU memory. All SLAM experiments are run on an Intel Core i7 CPU with 16 GB RAM. We use the ROS Noetic Distro for setting up Google cartographer for experiments with SLAM.

We will open source the codebase for AE_{mask} and SLACK.

VI. Metrics

A. Evaluation of $AEseg$

We describe the metrics for evaluating AE_{mask}.

Earth Mover's Distance - Given 2 LiDAR point clouds, L_1 and L_2, Earth Mover's Distance (EMD) calculates the least amount of work required to be done in order to transform L_1 to L_2 or vice-versa. Earth Mover's distance requires L_1 and L_2 to have the same number of points. EMD also requires a bijective mapping ψ between the point clouds. EMD is calculated as

$$= \min_{\psi: L_1 \longrightarrow L_2} \sum_{x \in L_1} \|x - \psi(x)\|_2 \tag{1}$$

Chamfer's Distance - Given 2 LiDAR point clouds L_1 and L_2, Chamfer's Distance between them is defined as

$$= \sum_{x \in L_1} \min_{y \in L_2} \|x - y\|_2^2 + \sum_{x \in L_2} \min_{x \in L_1} \|x - y\|_2^2 \tag{2}$$

For every point $x \in L_1$, we find the Euclidean distance to the closest point in L_2 and vice-versa. The sum total of these distances is Chamfer's Distance between L_1 and L_2. Unlike EMD, Chamfer's Distance allows L_1 and L_2 to have different number of points. It does not necessitate the existence of a bijective mapping between L_1 and L_2. This allows a single point in L_1 to map to multiple points in L_2, and vice versa.

B. Evaluating of SLACK

For evaluating SLACK we use 2 metrics - LiDAR Quality Index (LQI) and Dynamic Segmentation Ratio (DSR). DSR has been explained in the main paper. We give more details about LQI here.

LQI - LiDAR Quality Index or LQI, is used to asses the quality of a given LiDAR scan. The prerequisite for using this metric includes training a model with the domain scans against which reconstructed test scans have to be evaluated. To evaluate LQI for a given LiDAR scan, it is passed through the trained model. The model regresses the amount of noise in the input. LQI is based on work done on No-Reference Image Quality Assessment where the visual quality of an image is evaluated without any reference image or knowledge of the distortions present in the image [8].

The validation and usefulness of this metric is assessed by evaluating it against the original and reconstructed LiDAR scan of the domain. The regressed noise (LQI) is less for the original LiDAR as compared to generated/reconstructed ones.

To evaluate LQI for a set of LiDAR scans generated using a model M, we train another model L using the original LiDAR scans - the original LiDAR scans are injected with noise of varying levels. The injected noise is univariate Gaussian with mean 0 and variance σ. The model L is trained to be able to accurately regress the variance of the noise present in the input LiDAR scan [4].

C. Metrics for evaluating SLAM

To evalaute SLAM navigation we use 2 metrics: Absolute Trajectory Error(ATE) and Relative Pose Error(RPE)

Absolute Trajectory Error - ATE is used to measure the overall global consistency between two trajectories. It measures the difference between the translation components of the two trajectories using a 2 step process - aligning the coordinate frames for both trajectories and then comparing the absolute distances between the translational components of the individual poses. For more details on the procedure for calculating ATE, please refer to Sturm et. al. [9].

Relative Pose Error: RPE measures the deviation of the poses between the estimated and the ground truth trajectory locally - it measures the discrepancy between the estimated and ground truth trajectories over a specified time interval. It is robust to the accumulated global error. RPE takes the translational and rotational components of the trajectory into account and calculates the pose error separately for both. For a detailed description of the error and the calculation procedure, please refer to the Sturm et. al. [9].

REFERENCES

[1] T. Groueix, M. Fisher, V. G. Kim, B. C. Russell, and M. Aubry, "A papier-mâché approach to learning 3d surface generation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 216–224, 2018.

[2] P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, "Representation learning and adversarial generation of 3d point clouds," *arXiv preprint arXiv:1707.02392*, 2017.

[3] L. Caccia, H. van Hoof, A. Courville, and J. Pineau, "Deep generative modeling of lidar data," *arXiv preprint arXiv:1812.01180*, 2018.

[4] P. Kumar, S. Sahoo, V. Shah, V. Kondameedi, A. Jain, A. Verma, C. Bhattacharyya, and V. Viswanathan, "Dslr: Dynamic to static lidar scan reconstruction using adversarially trained autoencoder," *arXiv preprint arXiv:2105.12774*, 2021.

[5] V. Zyrianov, X. Zhu, and S. Wang, "Learning to generate realistic lidar point clouds," in *European Conference on Computer Vision*, pp. 17–35, Springer, 2022.

[6] Y. Xiong, W.-C. Ma, J. Wang, and R. Urtasun, "Learning compact representations for lidar completion and generation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1074–1083, 2023.

[7] J. Zhang, F. Zhang, S. Kuang, and L. Zhang, "Nerf-lidar: Generating realistic lidar point clouds with neural radiance fields," *arXiv preprint arXiv:2304.14811*, 2023.

[8] L. Kang, P. Ye, Y. Li, and D. Doermann, "Convolutional neural networks for no-reference image quality assessment," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1733–1740, 2014.

[9] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers, "A benchmark for the evaluation of rgb-d slam systems," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 573–580, IEEE, 2012.