USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

WIP: Understanding the Mechanisms Behind NDT-Based Localization Vulnerabilities in Autonomous Driving

Yuna Tanaka and Kazuki Nomoto, *Waseda University, Deloitte Tohmatsu Cyber LLC;*
Ryunosuke Kobayashi and Go Tsuruoka, *Waseda University;*
Tatsuya Mori, *WasedaUniversity, NICT, RIKEN AIP*

https://www.usenix.org/conference/vehiclesec25/presentation/tanaka

This paper is included in the Proceedings of the
3rd USENIX Symposium on Vehicle Security and Privacy.

August 11–12, 2025 • Seattle, WA, USA

978-1-939133-49-6

Open access to the Proceedings of the 3rd USENIX Symposium
on Vehicle Security and Privacy is sponsored by USENIX.

WIP: Understanding the Mechanisms Behind NDT-Based Localization Vulnerabilities in Autonomous Driving

Yuna Tanaka
Waseda University,
Deloitte Tohmatsu Cyber LLC

Kazuki Nomoto
Waseda University,
Deloitte Tohmatsu Cyber LLC

Ryunosuke Kobayashi
Waseda University

Go Tsuruoka
Waseda University

Tatsuya Mori
Waseda University, NICT, RIKEN AIP

Abstract

Accurate localization is critical for autonomous driving (AD), yet its security risks remain insufficiently explored, particularly in driving scenarios involving sensor fusion. This study investigates the vulnerabilities of Normal Distributions Transform (NDT) scan matching, a widely used localization method, and analyzes the conditions under which localization errors occur. We reveal that NDT relies primarily on nearby LiDAR point cloud structures from the pre-built map, making it susceptible to gradual manipulations that accumulate over time. To evaluate the impact of such manipulations, we conduct experiments simulating real-world scenarios, incorporating sensor fusion with an Extended Kalman Filter (EKF). Our findings identify key factors influencing localization errors, including target object selection and movement patterns, and confirm that these manipulations can induce errors of up to 23 m. End-to-end evaluation demonstrates that these errors can lead to hazardous driving behaviors, such as lane departures, missed traffic signals, and unintended sidewalk encroachments. By systematically analyzing the vulnerability of NDT-based localization, this study highlights the need for more robust localization mechanisms in AD.

1 Introduction

Accurate localization is essential for autonomous driving (AD). Many systems rely on High Definition maps (HD maps) containing lane structures, traffic light locations, and road signs [1, 2]. Localization errors can lead to lane departures, missed signals, and potential collisions. To achieve high accuracy, AD systems rely on multiple sensors, including Light Detection And Ranging (LiDAR), Global Navigation Satellite System (GNSS), and Inertial Measurement Unit (IMU), as well as vehicle speed and steering information. Among these, LiDAR enables highly accurate localization by aligning real-time 3D scans with pre-built point cloud maps [3]. Autoware [4, 5] and Apollo [6], two leading

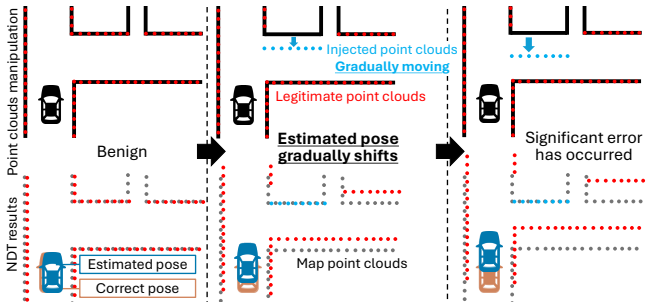


Figure 1: Overview of the *AND* attack.

open-source AD platforms widely used in commercial services, employ Normal Distributions Transform (NDT) scan matching [7] for this purpose [8, 9]. GNSS provides an initial position estimate that is refined by NDT scan matching, while IMU and vehicle dynamics data further stabilize the pose estimate through Extended Kalman Filter (EKF) [10].

Despite its importance, localization security has not been fully evaluated. To the best of our knowledge, only a limited number of studies have investigated localization attacks, including those targeting NDT algorithms. Yoshida et al. [11] demonstrated that LiDAR-based mapping is vulnerable to spoofing and conducted their evaluation in a controlled indoor environment using 2D LiDAR, enabling precise validation of attacks under specific conditions. Fukunaga et al. [12] investigated the impact of random spoofing on Simultaneous Localization And Mapping (SLAM)-based mapping, successfully inducing height errors while primarily focusing on vertical localization effects. Nagata et al. [13] identified effective attack locations and demonstrated their feasibility.

While these work empirically studied the vulnerabilities in LiDAR-based localization, they relied on heuristic point cloud manipulation, lacked a systematic analysis of NDT vulnerabilities, and did not evaluate attacks in realistic driving scenarios involving IMU fusion and vehicle control. To address these gaps, we analyze the fundamental vulnerability of NDT scan matching and introduce *Adversarial NDT Drift*

(*AND*), an attack that strategically manipulates LiDAR measurements to alter pose estimation. As shown in Fig. 1, the attack gradually shifts point cloud features such as walls and poles, leading to significant localization errors (up to 23 m). Our experiments identify key factors that influence the success of the attack, including point cloud selection and gradual shifts. We also analyze how EKF sensor fusion affects attack effectiveness and show that certain occlusion scenarios amplify the effect. End-to-end evaluation confirms that these localization errors lead to dangerous driving behaviors such as lane departures and signal misses.

This paper makes the following contributions:

- **Systematic Analysis of NDT Vulnerabilities**: We provide a systematic analysis of how the NDT scan matching algorithm processes point clouds and identify conditions under which adversarial point cloud manipulation effectively degrades localization accuracy.
- **Optimized Attack Strategy**: We demonstrate that injecting structured point clouds (e.g., vertical features such as walls and poles) significantly affects localization, and we quantify the relationship between displacement patterns and localization errors.
- **Realistic Sensor Fusion Evaluation**: Unlike previous work, we evaluate attacks under realistic simulation conditions using IMU fusion and EKF. Our results show that while EKF mitigates some errors, attacks still induce errors of more than 7 m, primarily in the forward-backward direction.
- **End-to-End Autonomous Driving Impact**: Using Autoware and AWSIM [14], we validate the attack in four environments and show that the resulting localization errors cause critical failures such as lane departures.

2 Background and Related Work

2.1 Localization system in AD

Localization of autonomous vehicle (AV) relies on multiple sensors, including LiDAR, GNSS, and IMU, combined with vehicle speed and steering data. These measurements are fused to achieve accurate pose estimation. Fig. 2 illustrates an example localization system architecture. GNSS provides an initial position estimate, which is used to identify the corresponding region in the pre-built map. LiDAR scan matching then refines this estimate by aligning real-time LiDAR scans with the pre-built point cloud map. IMU and vehicle speed measurements provide additional motion information, and EKF fuses these inputs to ensure stable localization, even in the presence of sensor noise or temporary occlusions.

Several scan matching algorithms exist, including NDT, Iterative Closest Point (ICP) [15, 16], and Lidar Odometry and Mapping (LOAM) [17]. Among these, NDT is widely adopted in commercial-grade open-source AD software such as Autoware [4,5] and Apollo [6] [8,9]. Due to its widespread use, this study focuses on the security of NDT.

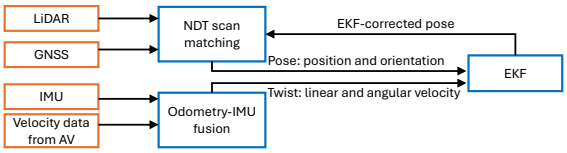


Figure 2: An illustration of the localization architecture.

NDT Scan Matching. NDT scan matching [7] is a method used to align two point clouds by estimating an optimal transformation. While it is also used in SLAM, this study focuses on its role in localization using pre-built maps. In this approach, the pre-built map is first divided into voxels, and each voxel is modeled as a normal distribution. Given an initial pose estimate, the real-time LiDAR scan is transformed into the map coordinate frame using a transformation parameter $\vec{p}$. A score function $score(\vec{p})$ is then defined to evaluate the alignment between the transformed scan and the map. The transformation is iteratively refined using Newton's method, where $\vec{p}$ is updated based on the gradient and Hessian of $score(\vec{p})$ until convergence is reached. Owing to its efficiency and robustness, NDT is widely employed in AD systems, such as Autoware and Apollo. However, the algorithm's reliance on an accurate initial pose estimate can introduce vulnerabilities, which are analyzed in detail in §3.1.

Fundamental Differences between NDT and Other Scan Matching Algorithms. NDT [7] and ICP [15] are categorized as direct matching methods, whereas LOAM [17] is classified as a feature-based matching method [18]. As described above, NDT performs alignment by leveraging normal distributions to represent the pre-build map point cloud structures. In contrast, ICP establishes correspondences between nearest neighbor points in two scans and computes the transformation that minimizes the point-to-point distance. LOAM extracts planar and edge feature points and aligns scans based on those features. Therefore, when designing attack strategies, it is effective to account for these fundamental differences in algorithmic behavior.

EKF-Based Sensor Fusion. EKF is commonly used in AD to fuse multiple sensor inputs for localization. By incorporating IMU and vehicle motion constraints, EKF mitigates the impact of transient sensor errors. For example, it can enforce physical constraints such as preventing lateral drift, improving localization stability.

2.2 Related Work

Physical Object Placement Attacks. Several studies have demonstrated that physical object placement can mislead LiDAR-based perception [19–25]. For instance, adversarial objects mounted on vehicle rooftops [19, 20] or hovering drones [21] have been used to evade detection systems.

Additionally, roadside objects have been shown to introduce false vehicle detections [22]. These attacks exploit LiDAR's reliance on geometric structures, manipulating perceived obstacles without directly interfering with sensor hardware.

LiDAR Spoofing Attacks. LiDAR spoofing manipulates sensor measurements by emitting malicious laser, enabling point cloud injection or removal [26–31]. The state-of-the-art spoofing attack [29] demonstrated the ability to inject 99% of point cloud within an 83° range and remove nearly all returns within an 80° range. Such attacks can disrupt object detection, and mislead localization systems.

Attacks on LiDAR-Based Localization. Several studies have explored attacks targeting LiDAR-based localization. Yoshida et al. [11] demonstrated that LiDAR-based SLAM is vulnerable to spoofing, causing pose estimation errors. However, their evaluation was confined to a simplified, two-dimensional corridor without intersections, relying on a 2D LiDAR that captures only planar data—making it an artificial setting far removed from real-world AD. Moreover, they did not assess how localization errors would affect vehicle control. Their attack methodology was designed based on the ICP algorithm and introduced a loss function that minimizes the loss on the adversarial target pose. Fukunaga et al. [12] introduced a random spoofing attack that disrupted SLAM-based mapping, inducing errors in the vertical axis. However, they found that horizontal localization remained largely unaffected. The spoofing pattern they employed is random. Nagata et al. [13] analyzed effective attack locations and demonstrated that LiDAR spoofing can induce localization errors in open environments. Their findings showed that both point cloud removal and injection could lead to pose estimation drift. However, their study did not investigate effective spoofing patterns, nor did it evaluate the impact of sensor fusion or vehicle control.

Unlike previous work, this study systematically analyzes the fundamental vulnerability of NDT scan matching. By leveraging gradual shifts in point cloud structures, such as building facades and poles, the proposed attack induces significant localization errors. Furthermore, this work evaluates, in a realistic simulation environment using a 3D LiDAR, how these errors propagate through sensor fusion, particularly through IMU and EKF-based integration, and ultimately affect AV control.

3 Analysis of Vulnerability Mechanisms

3.1 Fundamental Vulnerabilities of NDT

We analyze the inherent vulnerabilities of the NDT algorithm. Our analysis is based on the implementation provided by the Point Cloud Library (PCL) [32], which is widely adopted in AD systems, including Autoware, Apollo, and several other localization packages [33–36]. Because of its broad use, the PCL implementation serves as a representative example of NDT in real-world applications. The general procedure of the NDT algorithm is summarized in Algorithm 1.

Algorithm 1 NDT Scan Matching

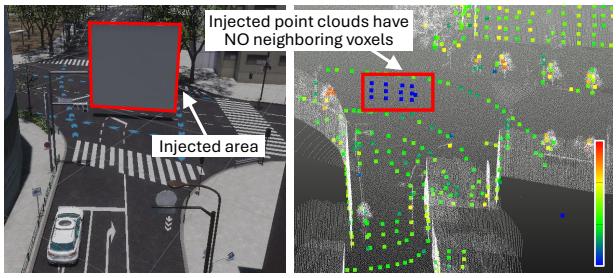
Input: Current scan point cloud $\mathcal{X} = \{\vec{x}_1, \ldots, \vec{x}_n\}$, map point cloud $\mathcal{Y} = \{\vec{y}_1, \ldots, \vec{y}_n\}$, and initial transformation parameter $\vec{p}_{\text{initial}}$
Output: Estimated pose

Divide $\mathcal{Y}$ into voxels and compute the mean and covariance of each voxel.
$\mathcal{X}' \leftarrow T(\vec{p}_{\text{initial}}, \mathcal{X})$
while not converged **do**
  $score \leftarrow 0$, $\vec{g} \leftarrow 0$, $\mathbf{H} \leftarrow 0$
  for each point $\vec{x}'_k \in \mathcal{X}'$ **do**
    Identify the set of neighboring voxels $\mathcal{B}_k$ in map.
    for each neighboring voxel $b_i \in \mathcal{B}_k$ **do**
      $score \leftarrow score + \text{CalcScore}(\vec{x}'_k, b_i)$
      $\vec{g} \leftarrow \vec{g} + \text{CalcGradient}(\vec{x}'_k, b_i)$
      $\mathbf{H} \leftarrow \mathbf{H} + \text{CalcHessian}(\vec{x}'_k, b_i)$
    end for
  end for
  Solve $\mathbf{H}\Delta\vec{p} = -\vec{g}$ for $\Delta\vec{p}$.
  $\vec{p} \leftarrow \vec{p} + \Delta\vec{p}$.
  Update $\mathcal{X}' \leftarrow T(\vec{p}, \mathcal{X}')$.
end while

In the algorithm above, the function $T(\vec{p}, \mathcal{X})$ transforms $\mathcal{X}$ using the transformation parameter $\vec{p}$. The functions `CalcScore`, `CalcGradient`, and `CalcHessian` compute the score, gradient $\vec{g}$, and Hessian $\mathbf{H}$ for each point $\vec{x}'_k$ relative to its corresponding neighborhood voxel b_i. Newton's method is then used to iteratively update $\vec{p}$ to maximize the alignment score between the transformed scan and pre-built map.

Vulnerabilities Induced by Adversarial Point Cloud Injection. One key vulnerability of NDT arises when adversarial point clouds are injected near the regions corresponding to the pre-built map. For each real-time LiDAR scan point $\vec{x}'_k$, the algorithm searches for neighboring voxels in the map. If no neighborhood voxel is found for a given point, that point is excluded from the computation of the score, gradient, and Hessian, and thus does not influence the update $\Delta\vec{p}$. As illustrated in Fig. 3, point cloud injections placed in areas far from the map points (e.g., the middle of a road) have minimal impact on the alignment process. Even when a large number of points are injected over a wide area, as shown in (a), if these points do not correspond to pre-built map points, they remain unutilized by NDT, rendering the injection entirely ineffective.

By gradually shifting the injected point clouds away from map structures, the alignment result can be subtly manipulated. Since the initial transformation parameter $\vec{p}_{\text{initial}}$ is



(a) Camera view (b) LiDAR point cloud (neighborhood voxel)

Figure 3: Injected point clouds in the red-framed area (a) appear as blue dots in the LiDAR point cloud visualization (b). These injected scan points lack corresponding pre-built map points, resulting in zero neighboring voxels. In (b), the injected blue dots are ignored by NDT due to the absence of neighboring voxels. The densely packed white dots in (b) represent high-resolution pre-built map points.

derived from the previous frame's NDT estimate or its EKF-corrected pose, even a small error introduced in one frame can accumulate over successive iterations. Under an attack, the transformed scan $\mathcal{X}'$—computed using a manipulated $\vec{p}_{\text{initial}}$—becomes misaligned with the map and shifts toward the injected points. This misalignment further biases the neighborhood search, allowing the injected points to continue influencing the pose update in a cascading fashion.

Persistence of Localization Errors. Another critical issue is the persistence of the induced errors. Once a significant localization error has been introduced, the iterative nature of the NDT algorithm prevents recovery, even if the adversarial injection ceases. This is due to the feedback loop in which the erroneous pose estimate is used as the initial estimate $\vec{p}_{\text{initial}}$ in subsequent frames.

In addition, physical obstructions such as parked trucks or roadside objects can occlude legitimate map features, thereby reducing the influence of correct measurements and making the attack more effective in cluttered environments.

In summary, the vulnerabilities of NDT stem from its reliance on local neighborhood matching and iterative pose refinement based on previous estimates. These features can be exploited by targeted point cloud injections, which induce persistent and accumulating errors, ultimately compromising the accuracy and reliability of the localization system.

3.2 Attack Based on NDT Vulnerabilities

Based on the vulnerability presented in §3.1 that NDT ignores injected point clouds that do not correspond to pre-built map points, we propose the *Adversarial NDT Drift (AND)* attack. By carefully manipulating point clouds to appear consistent with the map structure, the attacker can induce controlled localization drift, causing the AV to misperceive its position. The *AND* attack achieves this by gradually shifting point clouds representing parts of a building wall or poles (e.g., traffic lights, street lamps, utility poles), leading the AV to perceive itself as an arbitrary distance ahead or behind its actual location (Fig. 1). This displacement can be performed through LiDAR spoofing, as demonstrated in previous work [29], or by physically placing and moving objects in the real world. However, the latter approach may present practical deployment challenges.

The *AND* attack is executed in the following steps:

Step 1: Selection of Attack Locations and Creation of Map Point Clouds. To induce localization errors in the longitudinal direction, the attack is more effective in areas with fewer vertically oriented structures relative to the AV. Moreover, in areas with many tall buildings, a larger manipulation of point clouds is required. Considering these factors, the attacker selects a suitable attack location, collects LiDAR data, and generates a point cloud map.

Step 2: Specification of the Point Clouds for Manipulation. The attacker determines point cloud regions to manipulate based on the measured data and the map from Step 1. The manipulated point cloud is input into NDT to identify areas where manipulations most impact localization errors. To induce errors in the longitudinal direction, shifting point clouds that are vertical to the AV, horizontally relative to their original structures is effective (Fig. 1). Free point cloud processing software simplifies manipulation, and libraries such as PCL [32] provide NDT scan matching implementations. Leveraging these tools, the attacker efficiently analyzes the impact of point cloud manipulations on localization errors.

Step 3: Real World Manipulation of LiDAR Measurements. The attacker manipulates LiDAR measurements as determined in Step 2, using either spoofing or physical object placement. Spoofing can be executed using the method in prior work [29]. In contrast, physical object placement requires continuously moving physical objects, such as boards or poles, in the real world, which may be impractical. Potential approaches to achieving this include creating boards or poles from lightweight, portable materials such as plastic and transporting them via delivery drones, mounting them on carts or truck beds, or having them carried by humans disguised as construction workers or delivery personnel.

4 Evaluation

4.1 Experimental Setup

We evaluate the impact of the *AND* attack on the localization and control of AV utilizing NDT scan matching. Specifically, our evaluation examines how adversarial point cloud injections influence the NDT-estimated pose, the EKF-fused pose (integrating IMU and vehicle velocity), and the AV's planning and control. To conduct this evaluation, we use Autoware.Universe and AWSIM, where a Velodyne VLP-16 LiDAR [37], accurately modeled in AWSIM, is mounted

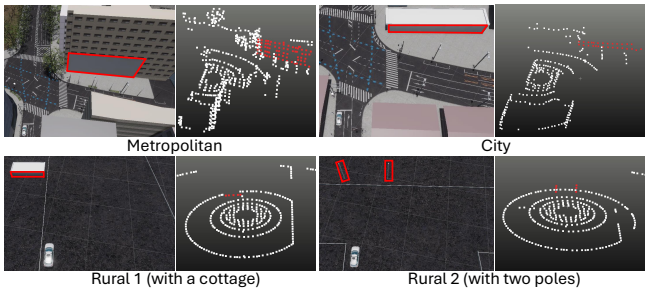


Figure 4: Four evaluation scenes. The red-highlighted areas and red points indicating manipulated point clouds.

on the AV's rooftop to capture point cloud data. The captured point clouds undergo preprocessing, including ego vehicle removal, outlier removal, range filtering, and voxel grid downsampling. The voxel grid downsampling replaces points within each 3 m $\times$ 3 m $\times$ 3 m voxel with their centroid, improving processing efficiency and NDT robustness.

To assess the impact of adversarial point cloud injections, we evaluate in four distinct driving environments. As illustrated in Fig. 4, these environments are recreated in AWSIM with different structural characteristics:

- **Metropolitan:** Tall buildings with dense urban features.
- **Urban:** A mix of single-story and two-story buildings.
- **Rural 1:** A road with a cottage and curbs.
- **Rural 2:** A road with two vertical poles (e.g., streetlights or traffic signals) and curbs.

Autoware includes a "threshold safety mechanism" that rejects the estimated pose if the alignment score between the pre-built map and the real-time scan falls below a predefined threshold; we use the default value for this threshold. This mechanism is effective in detecting erroneous localization. It operates to mitigate the impact of the attack, and we evaluate under more challenging conditions. In our evaluation, the IMU and velocity data are accurate and derived from the simulated motion of the vehicle in AWSIM. We evaluate static AV scenarios in §4.2 and dynamic AV scenarios in §4.3.

4.2 Attack Evaluation on NDT and EKF

Experimental Method. In this experiment, we evaluate the impact of the *AND* attack on NDT-based localization in four scenarios (Fig. 4) where the AV is stopped at a traffic light. First, we identify the effective point cloud manipulation areas that induce localization error in each scenario. To gradually shift the identified target point cloud, we move the Unity object horizontally along the original structure at a speed of 0.1 m/s. Here, the Unity object replicates point clouds generated by spoofing or physical object placement.

We quantify localization error by measuring the Euclidean distance between the estimated poses in the benign state and under the *AND* attack. We conduct five trials for each case: without and with twist (IMU and velocity) fusion. We then compute the average of the maximum error distances.

Results. Fig. 4 highlights the effective point cloud manipulation areas that induce localization errors, marked in red in the camera view and as red points in the point cloud view. In the metropolitan scene, the manipulated point cloud consists of a 15 m high, 40 m wide planar structure, while in the city scene, it is a 4.5 m high, 40 m wide planar structure. In the rural 1 (with a cottage) scene, a 2 m high, 10 m wide planar structure is used, whereas in the rural 2 (with two poles) scene, two 6 m high, 0.4 m wide pillars are manipulated.

Table 1 presents the total number of preprocessed measurement points, the number of manipulated points, and the horizontal manipulation angle for the four scenarios. The required number and size of manipulated points are larger in areas surrounded by buildings with widely distributed point clouds and smaller in open areas. The horizontal manipulation angles in all scenarios fall within the 83° range feasible for spoofing-based injection [29], demonstrating the practical feasibility of this attack. In contrast, implementing the attack through physical object movement may be impractical in environments requiring large-scale manipulation, though the approach discussed in §3.2 could be considered.

Table 1 shows the average and standard deviation of the maximum localization error distances over five trials for the four scenarios, both without and with twist fusion. The shifting distance of the manipulated point clouds closely matches the error distance, with larger displacements resulting in greater errors. Therefore, within the maximum error range shown in Table 1, stopping the shift of the manipulated point clouds at any position can induce an arbitrary error distance.

Without twist fusion, the maximum error distance corresponds to the distance where the score falls below the predefined threshold. As the legitimate point cloud shifts away from the map point cloud, the score decreases, causing the "threshold safety mechanism" to reject the estimated pose. While this mechanism is effective in limiting error distance and detecting localization failures, errors exceeding 8 m occurred in all scenarios, with the city scene reaching 23.91 m.

When twist fusion is applied, the maximum error distance is reduced compared to the case without fusion. While twist fusion helps mitigate localization error caused by the *AND* attack, errors of over 7 m occur in all scenarios. As discussed in §4.3, this error distance poses a significant threat.

A direct comparison is not feasible due to differences in experimental conditions; however, we confirmed that our attack induces comparable or greater errors than those in prior work. A direct comparison is left for future work.

Effect of Occluding Legitimate Point Clouds. In this experiment, we demonstrate that occluding legitimate point clouds enables more effective attacks in complex scenes. As shown in Fig. 5, we add a building to the rural-2 scene from Fig. 4 and place a truck to the right of the AV. When the truck is absent, the effect of the building's point cloud align-

Table 1: Point cloud count, manipulated angles, and maximum localization error (averaged over five trials) in four scenes.

Scene	Total points	Manipulated points	Manipulated angle [deg]	Localization error [m]	
				without twist fusion	with twist fusion
Metropolitan	592	72	30.47	8.965 ± 0.062	7.012 ± 0.007
City	306	28	31.05	23.91 ± 0.017	7.083 ± 0.251
Rural 1 (with a cottage)	305	5	20.00	13.43 ± 0.029	12.25 ± 1.347
Rural 2 (with two poles)	297	6	29.20	8.704 ± 0.053	8.661 ± 0.065

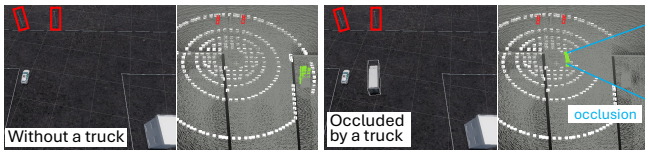


Figure 5: Occluding map-consistent legitimate point clouds (e.g., a building) can effectively enable a successful attack.

ing with the map is significant, making the attack difficult. When the truck is present, it occludes the point clouds of the building. The five-trial average of the maximum localization error with twist fusion is 8.758 m, which closely matches the rural-2 value in Table 1, showing that occlusion enables effective attacks even in complicated scenes.

4.3 End-to-End Evaluation

Experimental Method. We evaluate the impact of the *AND* attack on the AV's planning and control during right and left turns. In the metropolitan scene shown in Fig. 4, as described in §4.2, point clouds within a range of 15 m in height and 40 m in width is shifted to induce pose errors. Considering the position of the stop line and the lane width, with the aim of inducing critical failures such as lane departures, the final displacement of the manipulated point cloud is set to approximately 5 m for the right turn and 6 m for the left turn. The target point cloud is gradually shifted and moved to these distances. We use a HD map for left-hand traffic. The designated goal locations for the right and left turns are indicated by red stars in Fig. 6. The AV's maximum speed is set to 36 km/h. For acceleration and deceleration during departure, turning, and arrival at the goal, we use the default implementation provided by Autoware.Universe.

Results. Fig. 6 shows the results of the end-to-end evaluation. The green line indicates the AV trajectory without the attack, while the white line shows the trajectory under the *AND* attack. During the right turn, a localization error of 4.929 m caused the AV to misinterpret its position as being ahead of its actual location. This shortened the perceived straight-line distance to the turn, leading to unintended entry into the opposing lane. Similarly, during the left turn, a 6.355 m error led to sidewalk encroachment. The AV then detected guardrails and vegetation as obstacles and stopped

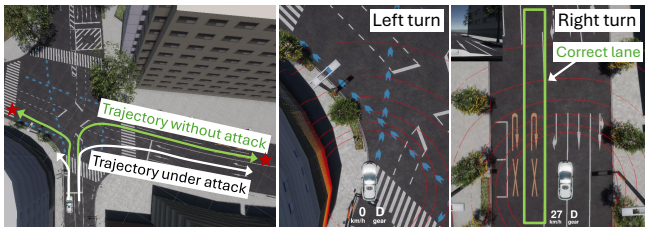


Figure 6: Results of end-to-end evaluation. During right turn, AV crossed into the opposite lane (correct lane is highlighted in green). During left turn, AV drove onto sidewalk. The red stars indicate the designated goal locations.

at the position shown in Fig. 6. During the right turn, the AV made the turn at 8–10 km/h, accelerated up to 27 km/h on the subsequent straight segment, and decelerated to stop at the goal location. During the left turn, the AV accelerated up to 7 km/h but began decelerating upon detecting obstacles and came to a stop at the position shown in the middle of Fig. 6. In both cases, the misinterpreted position placed the AV beyond the stop line of a traffic light, erroneously clearing the stop condition and causing a failure to stop.

5 Discussion and Future Directions

Defense. A potential defense against the *AND* attack is to partition the input scan point cloud and perform NDT scan matching on each subset, then compare the estimated poses. Since the *AND* attack modifies only a portion of the point cloud, excluding the affected points and relying on the remaining points may improve accuracy. This idea is similar to ObjectSeeker [38], which applies multiple masks to image inputs to improve robustness against adversarial patches. Implementing and evaluating a more robust NDT method under computational constraints is an important future task.

If IMU and velocity data are reliable while the NDT-estimated pose is potentially compromised, reducing the weight of NDT in sensor fusion could mitigate the impact of the attack. Assessing the reliability of NDT using metrics such as score values or iteration counts may improve robustness. Integrating camera-based localization could further help mitigate errors. Future work includes implementing and evaluating these strategies.

Attacks on Multi-Modal Sensor Data Fusion. If the AV

is equipped with camera-based localization, it may still estimate its position accurately even under the *AND* attack. For instance, Autoware implements a method that aligns lane structures from an HD map with road surface markings extracted from camera images [39]. Because road surface markings are less affected by the *AND* attack, this approach is considered more robust. Furthermore, visual odometry [40] and visual SLAM [41] may also provide accurate localization. However, further research is needed to evaluate AV behavior when localization results from LiDAR- and camera-based methods diverge.

Furthermore, obstacle detection using cameras or radars may prevent collisions between the AV and surrounding objects. In the left-turn scenario of the end-to-end evaluation in §4.3, we demonstrated that LiDAR-based obstacle detection enabled the AV to stop before colliding with obstacles. Even if localization errors are induced by the *AND* attack, camera- or radar-based obstacle detection may still prevent the AV from mounting sidewalks with guardrails or trees, or from colliding with buildings and other obstacles.

Future Directions. Generalizing attack success conditions across different scenarios is an open challenge. This WIP paper demonstrated the effectiveness of the *AND* attack in four different scenarios (metropolitan, urban, rural, and mixed) by manipulating point clouds of different shapes and sizes to induce localization errors. However, further research is needed to analyze optimal attack parameters and identify conditions that make environments more vulnerable.

Evaluating the feasibility of the *AND* attack in real-world conditions is an important next step. While this study validated the attack in simulation, future work should explore its practicality through LiDAR spoofing or physical object placement. Prior work on point cloud injection suggests feasibility under certain conditions, and physical attacks may also be possible. In addition, real-world factors such as changes in vegetation, construction, and traffic can alter point cloud and affect the attack effectiveness. Future studies should also evaluate the accuracy of IMU, GNSS, and velocity data in real-world and their impact on EKF fusion.

Evaluating the *AND* attack across different AD systems, preprocessing pipelines, and LiDAR models remains an open challenge. This study focused on Autoware, but future work should test its effectiveness on other systems such as Apollo. Differences in point cloud preprocessing could affect the feasibility of attacks. Additionally, while this study used a VLP-16 LiDAR, further evaluation with different LiDAR models and multi-LiDAR setups is needed. Extending the analysis to other scan matching techniques such as ICP and LOAM is another future direction.

Ethical Considerations. The purpose of this study is to understand fundamental vulnerabilities in NDT-based localization, not to target specific commercial products. Responsible disclosure will be followed if vulnerabilities in actual products are identified.

6 Conclusion

This paper systematically analyzes the vulnerabilities of NDT scan matching and demonstrates the *AND* attack, which strategically manipulates point clouds to induce localization errors. Our experiments reveal key factors influencing the success of the attack, including point cloud placement and gradual shifts, and show that while EKF sensor fusion mitigates some errors, it does not fully prevent adversarial drift. End-to-end evaluations confirm that these errors cause critical driving failures. We also discuss potential defenses, including robust NDT and adaptive sensor fusion, and outline future research directions, such as real-world validation and evaluation across different systems. This study highlights fundamental weaknesses in LiDAR-based localization and underscores the need for more robust AD systems.

Acknowledgments

This work was partially supported by JSPS KAKENHI 22H00519 and JST CREST JPMJCR23M4.

References

[1] R. Liu, J. Wang, and B. Zhang. High Definition Map for Automated Driving: Overview and Analysis. *Journal of Navigation*, Vol. 73, No. 2, p. 324–341, 2020.

[2] P. Bonetti. HERE introduces HD Live Map to show the path to highly automated driving. https://www.here.com/learn/blog/here-introduces-hd-live-map-to-show-the-path-to-highly-automated-driving.

[3] M. Choi, J. Ryu, Y. Son, S. Cho, and J. Paek. LiDAR-based Localization for Autonomous Vehicles - Survey and Recent Trends. In *15th International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 456–460, 2024.

[4] S. Kato, S. Tokunaga, Y. Maruyama, S. Maeda, M. Hirabayashi, Y. Kitsukawa, A. Monrroy, T. Ando, Y. Fujii, and T. Azumi. Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems. In *ACM/IEEE 9th ICCPS*, 2018.

[5] Autoware Foundation. Autoware.Universe. https://github.com/autowarefoundation/autoware.universe.

[6] Baidu Apollo. Apollo: Open Source Autonomous Driving. https://github.com/ApolloAuto/apollo.

[7] M. Magnusson. The three-dimensional normal-distributions transform : an efficient representation for

registration, surface analysis, and loop detection. In *Ph.D. dissertation, Orebro universitet*, 2009.

[8] Autoware Foundation. autoware_ndt_scan_matcher. https://autowarefoundation.github.io/autoware.universe/main/localization/autoware_ndt_scan_matcher/.

[9] ApolloAuto. NDT-based Lidar Localization. https://github.com/ApolloAuto/apollo/blob/master/modules/localization/ndt/README.md.

[10] G. Im. Notes on Kalman Filter (KF, EKF, ESKF, IEKF, IESKF). arXiv preprint arXiv:2406.06427, 2024.

[11] K. Yoshida, M. Hojo, and T. Fujino. Adversarial Scan Attack against Scan Matching Algorithm for Pose Estimation in LiDAR-Based SLAM. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, Vol. E105.A, No. 3, pp. 326–335, 2022.

[12] M. Fukunaga and T. Sugawara. Random Spoofing Attack against Scan Matching Algorithm SLAM. In *VehicleSec*, 2024.

[13] R.Nagata, K. Koide, Y. Hayakawa, R. Suzuki, K. Ikeda, O. Sako, Q. A. Chen, T. Sato, and K. Yoshioka. SLAM-Spoof: Practical LiDAR Spoofing Attacks on Localization Systems Guided by Scan Matching Vulnerability Analysis. arXiv preprint arXiv:2502.13641, 2025.

[14] TIER IV. AWSIM: Open source simulator for self-driving vehicles. https://github.com/tier4/AWSIM.

[15] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239–256, 1992.

[16] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152, 2001.

[17] J. Zhang and S. Singh. LOAM: Lidar odometry and mapping in real-time. In *Robotics: Science and systems*, Vol. 2, pp. 1–9, 2014.

[18] D. Lee, M. Jung, W. Yang, and A. Kim. LiDAR odometry survey: recent advancements and remaining challenges. *Intelligent Service Robotics*, Vol. 17, pp. 1–24, 02 2024.

[19] J. Tu, M. Ren, S. Manivasagam, M. Liang, B. Yang, R. Du, F. Cheng, and R. Urtasun. Physically Realizable Adversarial Examples for Lidar Object Detection. In *IEEE/CVF conference on CVPR*, pp. 13716–13725, 2020.

[20] S. Zhu, Y. Zhao, K. Chen, B. Wang, H. Ma, and C. Wei. AE-Morpher: Improve Physical Robustness of Adversarial Objects against LiDAR-based Detectors via Object Reconstruction. In *33rd USENIX Security Symposium*, pp. 7339–7356, 2024.

[21] Y. Zhu, C. Miao, T. Zheng, F. Hajiaghajani, L. Su, and C. Qiao. Can We Use Arbitrary Objects to Attack LiDAR Perception in Autonomous Driving? In *ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, p. 1945–1960. Association for Computing Machinery, 2021.

[22] K. Yang, T. Tsai, H. Yu, M. Panoff, T. Ho, and Y. Jin. Robust Roadside Physical Adversarial Attack Against Deep Learning in Lidar Perception Modules. In *ACM Asia Conference on Computer and Communications Security*, p. 349–362, 2021.

[23] Y. Cao, C. Xiao, D. Yang, J. Fang, R. Yang, M. Liu, and B. Li. Adversarial Objects Against LiDAR-Based Autonomous Driving Systems. arXiv preprint arXiv:1907.05418, 2019.

[24] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li. Invisible for both Camera and LiDAR: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks. In *2021 IEEE Symposium on Security and Privacy (SP)*, pp. 176–194, 2021.

[25] M. Abdelfattah, K. Yuan, Z. J. Wang, and R. Ward. Towards Universal Physical Attacks On Cascaded Camera-Lidar 3d Object Detection Models. In *IEEE International Conference on Image Processing*, pp. 3592–3596, 2021.

[26] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR. In *Black Hat Europe*, 2015.

[27] H. Shin, D. Kim, Y. Kwon, and Y. Kim. Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications. In *International Conference on Cryptographic Hardware and Embedded Systems*, pp. 445–467, 2017.

[28] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi. You can't see me: Physical removal attacks on lidar-based autonomous vehicles driving frameworks. In *32nd USENIX Security Symposium*, pp. 2993–3010, 2023.

[29] T. Sato, Y. Hayakawa, R. Suzuki, Y. Shiiki, K. Yoshioka, and Q. A. Chen. LiDAR Spoofing Meets the New-Gen: Capability Improvements, Broken Assumptions, and New Attack Strategies. In *Network and Distributed System Security Symposium*, 2024.

[30] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao. Adversarial Sensor Attack on Lidar-based Perception in Autonomous Driving. In *ACM SIGSAC conference on computer and communications security*, pp. 2267–2281, 2019.

[31] J. Sun, Y. Cao, Q. A. Chen, and Z. Mao. Towards robust LiDAR-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium*, pp. 877–894, 2020.

[32] Point Cloud Library. https://pointclouds.org/.

[33] K. Koide, J. Miura, and E. Menegatti. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, Vol. 16, No. 2, 2019.

[34] koide3. hdl_localization. https://github.com/koide3/hdl_localization/.

[35] rsasaki0109. lidar_localization_ros2. https://github.com/rsasaki0109/lidar_localization_ros2.

[36] rsasaki0109. lidarslam_ros2. https://github.com/rsasaki0109/lidarslam_ros2.

[37] Ouster. VLP-16. https://ouster.com/products/hardware/vlp-16.

[38] C. Xiang, A. Valtchanov, S. Mahloujifar, and P. Mittal. ObjectSeeker: Certifiably Robust Object Detection against Patch Hiding Attacks via Patch-agnostic Masking . In *IEEE Symposium on Security and Privacy (SP)*, pp. 1329–1347. IEEE Computer Society, May 2023.

[39] Autoware Foundation. YabLoc. https://autowarefoundation.github.io/autoware_universe/main/localization/yabloc/index.html.

[40] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle, and A. M. Pinto. A Practical Survey on Visual Odometry for Autonomous Driving in Challenging Scenarios and Conditions. *IEEE Access*, Vol. 10, pp. 72182–72205, 2022.

[41] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai. A review of visual SLAM methods for autonomous driving vehicles. *Engineering Applications of Artificial Intelligence*, Vol. 114, p. 104992, 2022.