Enduring, Efficient and Robust Trajectory Prediction Attack in Autonomous Driving via Optimization-Driven Multi-Frame Perturbation Framework

Yi Yu[1], Weizhen Han[2], Libing Wu[1†], Bingyi Liu[2†], Enshu Wang[1], Zhuangzhuang Zhang[3]
[1]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University
[2]School of Computer Science and Artificial Intelligence, Wuhan University of Technology
[3]Department of Computer Science, City University of Hong Kong
{yui1212, wu, wanges17}@whu.edu.cn
{hanweizhen, byliu}@whut.edu.cn {zhuangzhuang.zhang}@cityu.edu.hk

Abstract

Trajectory prediction plays a crucial role in autonomous driving systems, and exploring its vulnerability has garnered widespread attention. However, existing trajectory prediction attack methods often rely on single-point attacks to make efficient perturbations. This limits their applications in real-world scenarios due to the transient nature of single-point attacks, their susceptibility to filtration, and the uncertainty regarding the deployment environment. To address these challenges, this paper proposes a novel LiDAR-induced attack framework to impose multi-frame attacks by optimization-driven adversarial location search, achieving endurance, efficiency, and robustness. This framework strategically places objects near the adversarial vehicle to implement an attack and introduces three key innovations. First, successive state perturbations are generated using a multi-frame single-point attack strategy, effectively misleading trajectory predictions over extended time horizons. Second, we efficiently optimize adversarial objects' locations through three specialized loss functions to achieve desired perturbations. Lastly, we improve robustness by treating the adversarial object as a point without size constraints during the location search phase and reduce dependence on both the specific attack point and the adversarial object's properties. Extensive experiments confirm the superior performance and robustness of our framework.

1. Introduction

Autonomous Driving (AD) systems [1, 2] typically consist of perception, prediction, and planning modules. Among these, the prediction module plays a critical role, as an autonomous vehicle (AV) relies on the predicted trajectories of nearby vehicles for decision-making. Errors in trajectory prediction can compromise AV safety. Therefore, exploring the vulnerability of the prediction module is vital to improving the safety of the AD system, and research on attacks targeting trajectory prediction [6, 8, 16, 26, 32, 34] also has received increasing attention.

Existing trajectory prediction attacks typically compromise the victim AV's prediction module by manipulating the historical trajectory of an adversarial vehicle [6, 26, 32, 34]. These methods *directly* inject erroneous trajectory states (including coordinates and heading at each frame) into the victim AV's prediction module, which effectively distorts trajectory predictions. However, in real-world scenarios, physical constraints such as coordinate precision and road variability make it difficult to manipulate an adversarial vehicle precisely along a pre-calculated trajectory. Furthermore, the prediction module integrates both historical normal trajectory segments and precomputed adversarial trajectories. As the input window extends backward, the growing dominance of normal trajectories dilutes adversarial impact on earlier time steps, causing direct attacks to decay over time and preventing sustained influence [34].

In contrast, Lou *et al.* [16] introduced an *indirect* trajectory prediction attack method, termed the *SinglePoint Attack* (SP-Attack). It involves strategically placing physical objects near a parked adversarial vehicle to deceive the LiDAR perception module, producing perturbed bounding box attributes for the adversarial vehicle. These perturbed attributes are then passed through the tracking module, which generates incorrect trajectory states that propagate through the prediction module. The resulting prediction errors can lead to unsafe reactions from the victim AV. The SP-Attack consists of two phases: an attack phase to

† Corresponding authors.
https://github.com/y1y5/OMP-ATTACK

calculate the target state perturbations (i.e., perturbations of the trajectory states), and a location search phase to identify adversarial locations capable of producing these perturbations. By carefully placing objects near the adversarial vehicle, SP-Attack can achieve effects comparable to direct attacks. However, the indirect attack faces unique challenges. In the first stage, it uses a single-point attack strategy that generates perturbations only for the latest (current) frame of historical trajectory, which spans multiple frames of states. The single-point attack is easily filtered out by the model and lacks the endurance to affect earlier frames. This underscores the need for a multi-frame attack. Therefore, a key challenge lies in extending and maintaining the impact of a single-point attack across multiple frames to ensure persistent perturbations over time (**Challenge 1**). The second stage involves identifying optimal adversarial locations within a vast 3D space. To achieve the multi-frame attack, state perturbations must be sustained over time, complicating the task of efficiently identifying the best locations. The existing method struggles with the scalability required to find such optimal locations within this large domain (**Challenge 2**). The success of the indirect attack depends on whether the victim AV precisely reaches the target attack point. Any deviation from the attack point can reduce its impact. Additionally, the size and orientation of adversarial objects significantly affect the attack's performance. These properties are predetermined before the attack planning, making the attack sensitive to changes in the object's characteristics. Ensuring robustness against these variations remains critical, but developing a strategy to achieve this robustness introduces further complexity (**Challenge 3**).

To address the aforementioned challenges, we propose a novel *Optimization-driven Multi-frame Perturbation Framework* (OMP-Attack) aimed at *simultaneously* achieving endurance, efficiency, and robustness. OMP-Attack follows a similar two-stage pipeline as SP-Attack. In the attack phase, rather than focusing on the single-point attack at the current frame as in SP-Attack, we extend the attack endurance by selecting a sequence of consecutive historical frames, which can be divided into multiple temporally consecutive history trajectories. Specifically, starting from the current frame, we sequentially apply a single-point attack by stepping back one frame at a time. The perturbations from each frame are then concatenated to form a multi-frame perturbation sequence that effectively disrupts each frame and addresses **Challenge 1**. In the location search phase, we design three specialized loss functions to optimize adversarial locations by aligning detected state perturbations with multi-frame state perturbations. These functions focus on per-frame positional distance, directional alignment, and overall trajectory shape similarity respectively. Guided by these loss functions, we employ swarm intelligence (SI) algorithm [20, 31] to efficiently navigate the vast search space, tackling **Challenge 2**. Furthermore, to enhance robustness, we propose a *Robust Attack Strategy* that ignores the adversarial object's size and orientation during the search phase, treating it as a single central point. By focusing solely on the 3D central coordinates of an adversarial object, we ensure that the real object only needs to overlap with this central point to generate the target perturbations. This strategy effectively mitigates the reliance on attack point and object properties, addressing **Challenge 3**.

Our main contributions are as follows:

- We identify key challenges in existing trajectory prediction attacks and propose solutions that enhance attack endurance, optimize adversarial location search efficiency, and improve attack robustness.
- We introduce a novel *Optimization-driven Multi-frame Perturbation Framework* that can *simultaneously* achieve endurance, efficiency, and robustness in LiDAR-induced trajectory prediction attacks.
- We conduct a comprehensive evaluation of the proposed framework on the nuScenes [3] dataset from multiple perspectives. The results show a strong attack performance, with attack degradation kept within less than 25% under varying conditions.

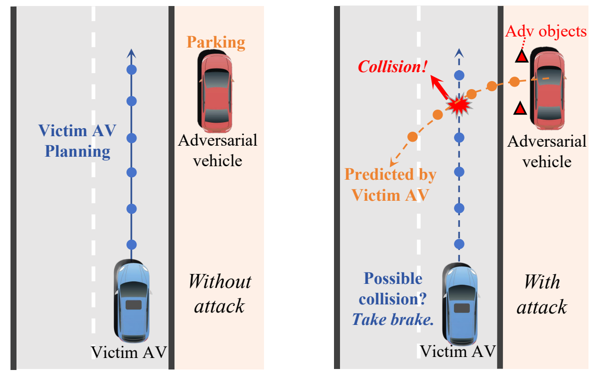
2. Background and Related Works

Autonomous driving systems. Autonomous driving (AD) systems (*e.g.*, Autoware [1], Baidu Apollo [2]) typically comprise three key modules: object perception, trajectory prediction, and motion planning. The perception module utilizes sensors (*e.g.*, LiDAR, camera) to detect and track on-road objects, the prediction module forecasts nearby agents' future trajectories, and the planning module determines the AV's actions based on these predictions.

Adversarial attacks. Deep learning models are highly susceptible to adversarial examples, which are perturbed input samples that mislead the model to generate incorrect outputs [9, 11, 17, 22, 24, 27]. Based on this vulnerability, numerous studies have explored the impact of adversarial examples on various modules within AD systems. Prior research indicates that perturbations in sensor signals or physical patches can disrupt tasks such as object detection [4, 23, 25] and tracking [13, 15, 21]. Similarly, trajectory prediction [6, 16, 26, 32, 34] and motion planning [30] modules are also vulnerable to perturbations.

LiDAR-induced attacks. LiDAR is widely utilized in perception modules due to its high precision. Many LiDAR-induced attacks involve placing physical objects to generate adversarial points that deceive the perception module [5, 28, 36] or indirectly disrupt prediction and planning processes. Another form of LiDAR-induced attack involves spoofing and injecting fake points into LiDAR's data stream by intercepting its emitted laser pulses with a receiver [7, 23, 25], but such an approach incurs high costs.

Trajectory prediction attacks. Trajectory prediction attacks can fall into two categories: direct and indirect. Direct attacks compromise the prediction module by manipulating the historical trajectory of an adversarial agent [6, 26, 32, 34]. Indirect attacks place physical objects near the adversarial vehicle to generate adversarial point clouds [16], which disrupt the perception module's detection results and indirectly attack the prediction module. Compared to direct attacks, indirect methods offer greater reproducibility and lower costs in the real world. As such, this paper focuses on the LiDAR-induced indirect method. Our approach achieves sustained attacks while ensuring strong and robust performance through location optimization.



(a) The normal scenario. (b) The scenario after the attack.

Figure 1. Attack scenario of indirect trajectory prediction attack.

3. Problem Formulation

3.1. Attack Model

In this work, we focus on a road scenario where the victim AV is driving along a road with a parked adversarial vehicle on the roadside, as depicted in Fig. 1a. In the attack setup, the victim AV is targeted by an attacker using a parked adversarial vehicle and adversarial objects, *e.g.*, circular cardboards, to generate adversarial state perturbations (i.e., perturbations of the historical trajectory states). We assume that victim AV runs an AD system for LiDAR-based perception (including detection and tracking), trajectory prediction, and motion planning. The attacker first strategically places adversarial objects at pre-computed locations near the adversarial vehicle. These adversarial objects are intended to create maximum interference at a specific position (i.e., the attack point) identified during the attack planning phase. When the victim AV reaches the attack point, it will input the observed historical trajectory states of the adversarial vehicle and apply trajectory prediction at the current frame. However, the adversarial objects already distort the perception module's output, causing erroneous historical trajectory states for the adversarial vehicle and indirectly leading to inaccurate trajectory predictions. As a result, the victim AV may take sudden actions such as emergent braking to avoid a perceived collision illustrated in Fig. 1b.

In a real-world scenario, the attacker needs to access all parameters or only APIs of the prediction model equipped by the victim AV for a white-box attack. The attacker could acquire information through reverse engineering or analyzing open-source models. Even in a black-box setting, the adversarial locations from the white-box attack remain effective in black-box attack. Our experimental results validate the success of the black-box attack.

3.2. Attack Formulation

We assume the current frame is at time step t when the victim AV reaches the attack point. We denote the state of vehicle i at time t as s_t^i, which includes spatial coordinates and dynamic features. The states of all N (set to 1 in our scenario) on-road vehicles (except the victim AV) at time t are represented by $S_t = \{s_t^i \mid i \in [1, N]\}$. From this, we derive the historical trajectory $H_t = \{S_{t-L_h+1}, ..., S_t\}$ over L_h time steps and the future trajectory $F_t = \{S_{t+1}, ..., S_{t+L_f}\}$ over L_f time steps, which serve as the input and label for the prediction module. The originally planned trajectory of victim AV without attack is denoted as $Y_t^v = \left\{S_{t+1}^{ego}, ..., S_{t+L_f}^{ego}\right\}$, and the predicted trajectory of adversarial vehicle by victim AV is denoted as $Y_t^a = f_{\text{pre}}(H_t + P_t)$, where $f_{\text{pre}}(\cdot)$ represents the prediction model and $P_t = \{p_{t-L_h+1}, ..., p_t\}$ represents possible perturbations over L_h historical time steps where p_t includes the state perturbation of x-coordinate δ_t^x, y-coordinate δ_t^y, and heading δ_t^h. Through multiple iterations of Projected Gradient Descent (PGD) [12, 18, 19], we obtain the target trajectory state perturbations P_t^{tar}. Specifically, the goal is to minimize the average trajectory distance (ATD) D_{avg} between Y_t^v and Y_t^a as follows:

$$\min D_{avg} = \min \ \mathcal{L}\left(Y_t^v, Y_t^a\right), \tag{1}$$

where D_{avg} denotes the ATD and $\mathcal{L}$ represents the loss function, typically the mean squared error (MSE). By this way, the target adversarial trajectory $H_t^{tar} = P_t^{tar} + H_t$ is obtained for direct attacks. However, we focus on indirect attack, aiming to find M adversarial locations (3D coordinates) $L_{adv} = \{l_i \mid i \in [1, M]\}$ that meet the attack goal by minimizing the following objective:

$$\begin{aligned} &\min \ \text{Similarity}\left(P_t^{det}, P_t^{tar}\right), \\ &\text{s.t. } P_t^{det} = H_t^{det} - H_t, \\ &\quad H_t^{det} = f_{\text{track}}\left(f_{\text{det}}\left(g(L_{adv})\right)\right), \end{aligned} \tag{2}$$

where the similarity between P_t^{det} and P_t^{tar} can be measured by custom-designed loss functions, P_t^{det} represents the detected state perturbations, $f_{\text{track}}(\cdot)$, $f_{\text{det}}(\cdot)$ denote the

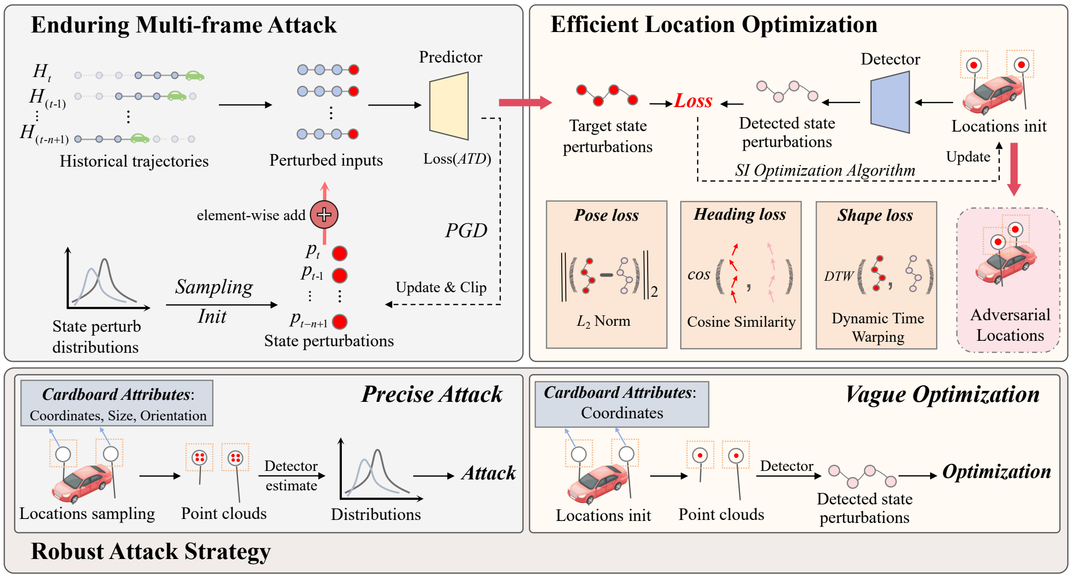


Figure 2. The optimization-driven multi-frame perturbation framework (OMP-Attack) methodology overview. Predictor refers to the trajectory prediction module, and detector refers to the perception module (including object detection and state tracking).

tracking model and detection model respectively, $g(L_{adv})$ represents the perceived point clouds over L_h historical time steps after placing the adversarial objects, denoted as $\{C_{t-L_h+1}, ..., C_t\}$.

4. Attack Method

In this section, we introduce the OMP-Attack framework, and the overall framework is depicted in Fig. 2.

4.1. Enduring Multi-frame Attack

To ensure endurance, we propose the *Enduring Multi-frame Attack* across multiple frames to achieve P_t^{tar}, as illustrated in Fig. 2. Compared to a multi-point attack within a single historical trajectory, the advantage of using the single-point attack over multiple temporally continuous historical trajectories lies in the fact that single-point attack focuses solely on perturbing the current frame of each trajectory. This approach avoids interference with gradient updates from other frames, maximizing the attack's effectiveness.

Perturbations initialization. To facilitate convergence, we utilize a sampling method [16] to initialize perturbations. First, we randomly generate Q (the number of *query*) sets of adversarial locations and use the perception module to detect the state perturbations, represented as $P = \{\delta_i^x, \delta_i^y, \delta_i^h \mid i \in [1, Q]\}$. We then estimate Gaussian distributions for the perturbations using values from the *query* sets $\{\delta_i^j \mid i \in [1, Q], j \in \{x, y, h\}\}$, represented as $\{\mathcal{N}_x, \mathcal{N}_y, \mathcal{N}_h\}$. Finally, we sample these distributions to initialize n state perturbations, denoted by $\{p_{t-n+1}, ..., p_t\}$.

Target state perturbations generation. Starting from time step t, we perform the single-point attack for the preceding n historical trajectories successively. For the historical trajectory H_{t-i} at time step $(t-i)$, where $i \in [0, n-1]$, we first add the initialized state perturbation p_{t-i} element-wise to the current frame, resulting in a perturbed input, $\widetilde{H}_{t-i} = H_{t-i} + p_{t-i}$. We then retrieve the original planned trajectory Y_{t-i}^v of victim AV and obtain the predicted trajectory $Y_{t-i}^a = f_{\text{pre}}(\widetilde{H}_{t-i})$ of the adversarial vehicle. Using PGD, we update the perturbation according to the objective defined in Eq. (1) and apply clipping to p_{t-i} using the 3-σ rule from the perturbation distributions [10]. After the gradient backpropagation for multiple epochs, the final target state perturbation p_{t-i}^{tar} is obtained. By concatenating the target state perturbation across n frames, we derive the final n-frame successive target state perturbations, denoted as $P_t^{tar} = \{p_{t-n+1}^{tar}, ..., p_t^{tar}\}$. The multi-frame target state perturbations introduce adversarial states across multiple time steps, ensuring continuous impact on the victim AV throughout the attack.

4.2. Efficient Location Optimization

In this phase, our goal is to efficiently identify adversarial locations where generate the desired successive state perturbations P_t^{tar} (Sec. 4.1). However, the continuous nature of the location space presents a vast search problem. To address this, we design three loss functions according to the objective defined in Eq. (2) and apply *SI* algorithms for efficient location identification. We select Particle Swarm Optimization (PSO) [14] due to its balance between search performance and efficiency. The evidence is in the appendix.

Location optimization. We begin by initializing a set of adversarial locations, $L = \{l_i \mid i \in [1, M]\}$. At each location, cardboard is placed, and the perception module detects perturbations over n historical time steps, yielding the detected state perturbations P_t^{det} as derived in Eq. (2). To guide the optimization process in locating adversarial locations, we design three loss functions: $\mathcal{L}_{pose}$, $\mathcal{L}_{heading}$ and $\mathcal{L}_{shape}$. The pose loss $\mathcal{L}_{pose}$ enforces precise alignment between P_t^{det} and P_t^{tar} by measuring their spatial distance using the Euclidean norm. The heading loss $\mathcal{L}_{heading}$ preserves directional consistency across frames via cosine similarity. Additionally, to account for trajectory shape in filtering, $\mathcal{L}_{shape}$ employs Dynamic Time Warping distance [29, 35] to evaluate overall shape similarity between P_t^{det} and P_t^{tar}. The total loss $\mathcal{L}_{total}$ is expressed as:

$$\mathcal{L}_{total} = \alpha \cdot \mathcal{L}_{pose} + \beta \cdot \mathcal{L}_{heading} + \gamma \cdot \mathcal{L}_{shape} \ , \qquad (3)$$

where α, β, and γ represent weights. Details of these loss functions are in the appendix. Guided by $\mathcal{L}_{total}$, PSO iteratively updates the adversarial locations L and ultimately converges on the optimal adversarial locations L_{adv}.

4.3. Robust Attack Strategy

To enhance the attack robustness, we focus on two key factors: (1) the distance offset between victim AV and the attack point, and (2) the impact of the adversarial object's size and orientation on the attack. We propose a *"Precise Attack, Vague Optimization"* strategy to address these factors.

Precise attack. In the attack phase, our goal is to generate state perturbations that maximize the attack's effectiveness. To achieve this, we need to accurately model the characteristics of cardboard, including its central coordinates, size, and orientation. Specifically, we simulate the point cloud detection result for the i-th cardboard at time t using 4 points, represented as $C_t^i = \{c_i^1, c_i^2, c_i^3, c_i^4\}$, where each c_i corresponds to a 3D coordinate and reflectance feature. By accurately simulating the cardboard characteristics, we can derive precise perturbation distributions, ensuring that the target perturbations are highly effective and well-targeted.

Vague optimization. In contrast, the optimization phase simplifies the object representation by focusing only on its core feature, namely the center coordinates. We simulate the i-th cardboard's point cloud detection result as a single point $C_t^i = \{c_i^1\}$, corresponding to its center point. This simplification assumes that the center point alone is sufficient to generate the desired state perturbations at the adversarial location. By considering only the center point, we eliminate the need to accurately define the object's size and orientation, enabling more flexible and efficient optimization of adversarial locations. In practice, we position the object to ensure that at least one point on it covers the center point. This alignment generates the primary state perturbation from the center point's point cloud, which sustains attack performance. Meanwhile, additional perceived points on the object contribute auxiliary perturbations, thereby effectively expanding the attack region. This setup reduces the reliance on precise attack points in any given frame.

By combining *Precise Attack* for generating accurate perturbations with *Vague Optimization* for flexible adversarial location search, we enhance both the effectiveness and robustness. This approach ensures that the attack remains effective even when victim AV experiences lateral deviations or when the adversarial object's properties vary.

5. Experiments

In this section, we evaluate the proposed OMP-Attack from multiple perspectives and analyze the results.

5.1. Experimental Setups

Dataset. We use the nuScenes [3] dataset for evaluation, which includes both perception and prediction tasks. We manually select 50 driving scenes from the dataset that meet the aforementioned attack scenario. The specific selection criteria are provided in the appendix.

Baselines. We compare the OMP-Attack with two baselines. The first is *SinglePoint Attack* [16] (SP-Attack), the only state-of-the-art indirect method. The second is *Brute-Force* search for finding adversarial locations. In the *Brute-Force* method, we randomly generate adversarial locations per iteration and select the most effective locations.

Evaluation metrics & AD models. The evaluation metrics include the average trajectory distance (ATD) between the adversarial vehicle's predicted trajectory and victim AV's original planned trajectory, the average planning-response error (PRE) between the victim AV's attacked and non-attacked planned trajectories, and the collision rate (CR) in future frames. CR is computed by assessing whether the victim and adversarial vehicles overlap at any of the L_f future time steps based on their center positions, lengths, and widths. A collision is recorded if an overlap occurs in any frame. A lower ATD, higher PRE, and increased CR indicate an enhanced effect. To ensure a fair comparison with SP-Attack, we employ the identical AD models as theirs. The details of AD models are in the appendix.

Implementation details. For the attack phase, we set the number of $query$ in the perturbation distribution to 1000 and the number of initialized perturbations n to 5, applying a single-point attack over 5 consecutive historical trajectories. For the optimization phase, we use the PSO with 10 particles and set the maximum iteration to 100. The number of adversarial objects is set to 3 and the object's radius is set to 0.1 m. Further details are available in the appendix.

5.2. Experimental Results

We present the experimental results from multiple perspectives, including attack effectiveness, endurance, and robustness. Additional results are provided in the appendix.

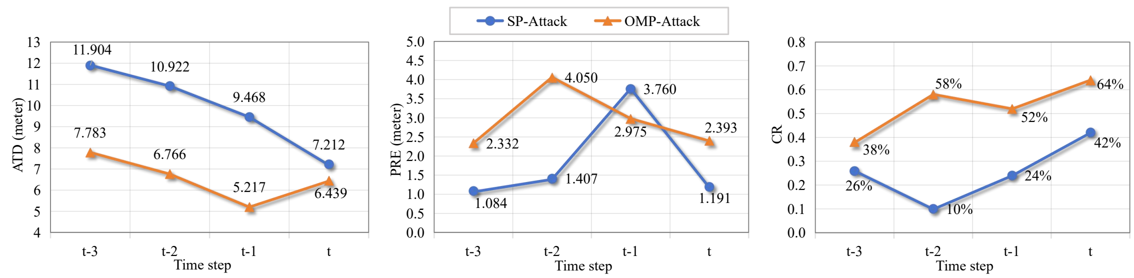


Figure 3. The quantitative results of attack endurance. For the preceding 4 time steps starting from the current frame t, the victim AV predicts the trajectory of the adversarial vehicle and re-planning at each time step.

5.2.1 Attack Effectiveness

Comparison with counterparts. Following the default configuration with L_h=5 and L_f=6, we compare the effectiveness of our proposed attack framework against baseline methods, with results summarized in Tab. 1. OMP-Attack outperforms SP-Attack by reducing ATD by 47%, increasing PRE by 100%, and improving CR by 52%, indicating a significant performance boost. Additionally, we evaluate the impact of using PSO in SP-Attack and replacing PSO in OMP-Attack with the manual matching method from SP-Attack. Both changes lead to reduced attack effectiveness. This is because SP-Attack focuses solely on disrupting the current frame's state and using PSO weakens the attack by searching normal states in other frames. In contrast, OMP-Attack disrupts multi-frame states, where the states are uncorrelated, making PSO effective. Manual matching requires continuous detection across frames and is less suited for uncorrelated multi-frame disruptions, further lowering the performance of enduring attacks. This highlights PSO's suitability for continuous multi-frame attacks, while manual matching works better for single-point attacks.

Varying particle numbers. In addition, we conduct a parameter selection experiment to assess the effect of varying particle numbers $\{5, 10, 15, 20\}$ in PSO. The results in Tab. 1 show that the ATD decreases as the number of particles increases, while the PRE exhibits an overall upward trend. At the optimal outcome, PRE reaches 3.183 meters and CR achieves 68%. This suggests that an increased number of particles enhances the ability to identify adversarial locations that correspond to target state perturbations, further validating the feasibility of our method in Sec. 4.2.

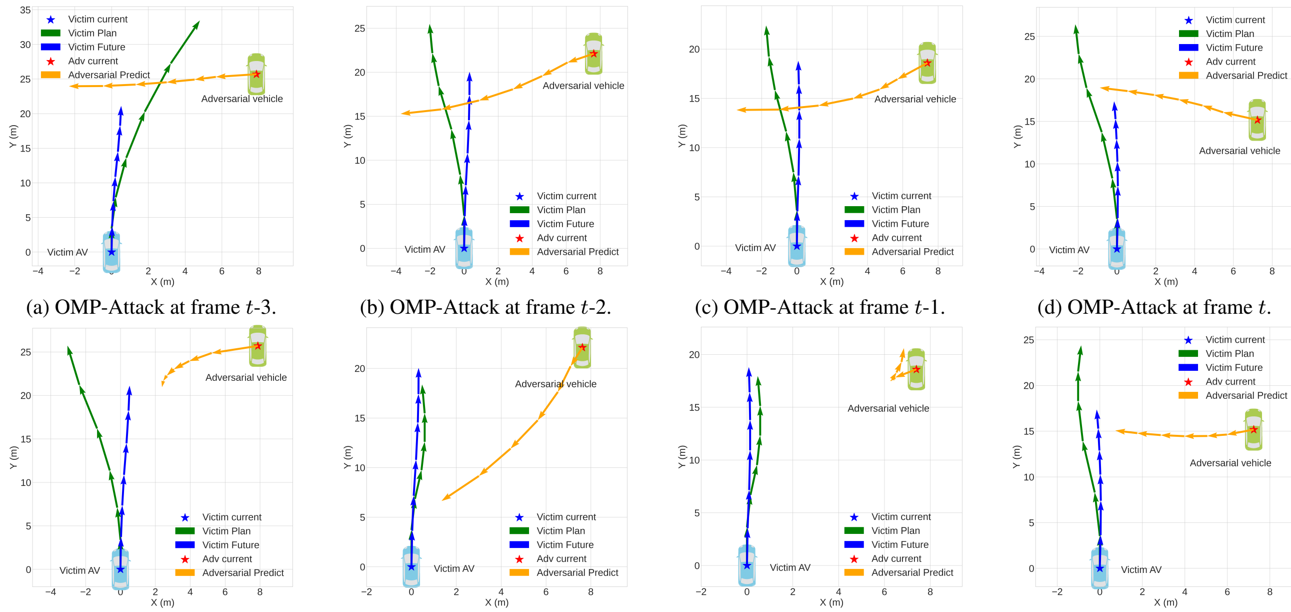
Table 1. Average attack results compared with counterparts and the effect of varying particle numbers on the attack effectiveness within the SI optimization algorithm. w represents 'with' and w/o represents 'without'.

Attack Methods	**ATD**$(m)\downarrow$	**PRE**$(m)\uparrow$	**CR**$\uparrow$
Comparison with Counterparts			
Clean Scene	8.855	0	0
Brute-Force	8.196	0.998	27%
SP-Attack [16]	7.212	1.191	42%
OMP-Attack	6.439	2.393	64%
SP-Attack w PSO	8.099	1.333	24%
OMP-Attack w/o PSO	7.588	0.813	34%
Varying Particle Numbers			
OMP-Attack w 5 particles	6.734	2.075	50%
OMP-Attack w 10 particles	6.439	2.393	64%
OMP-Attack w 15 particles	6.143	2.054	44%
OMP-Attack w 20 particles	5.821	3.183	68%

5.2.2 Attack Endurance

We conduct endurance experiments to evaluate the persistence of our proposed framework. The results are presented both quantitatively and visually. Starting from the manually selected current frame t in each scene, we allow the victim AV to perceive, predict, and plan over the preceding 4 time steps. We then calculate the average ATD, PRE, and CR for each frame. The quantitative results in Fig. 3 reveal two valuable insights. First, OMP-Attack maintains a persistent impact across multiple frames, showing only a minor decrease in effectiveness as we trace back through earlier frames. Notably, the maximum increase in ATD for OMP-Attack is under 1.6 meters, in contrast to SP-Attack's increase of 4 meters. Second, OMP-Attack outperforms SP-Attack across key metrics. Over the 4 time steps, OMP-Attack achieves an average ATD of 6.551 meters, a PRE of 2.938 meters, and a CR of 53%, compared to 9.877 meters, 1.86 meters, and 26% for SP-Attack respectively. In some earlier frames, the OMP-Attack performance even surpasses that of the current frame. For instance, in the frame $(t-1)$, the ATD decreases by 1.222 meters, while PRE increases by 0.582 meters compared to the frame t. This indicates that OMP-Attack not only demonstrate attack endurance but also maintain strong effectiveness.

In Fig. 4, we visually compare the results of our proposed method with SP-Attack in a real dataset scene. The predicted trajectories in OMP-Attack exhibit a much more



(a) OMP-Attack at frame t-3. (b) OMP-Attack at frame t-2. (c) OMP-Attack at frame t-1. (d) OMP-Attack at frame t.

(e) SP-Attack at frame t-3. (f) SP-Attack at frame t-2. (g) SP-Attack at frame t-1. (h) SP-Attack at frame t.

Figure 4. Visualization of attack endurance results for OMP-Attack and SP-Attack [16]. The attack results of OMP-Attack at frame t-3, t-2, t-1, and t are presented in (a)-(d); those of SP-Attack at the same frames are shown in (e)-(h). The yellow line represents the predicted trajectory of the adversarial vehicle by victim AV, the blue line represents the original planning trajectory of victim AV, and the green line represents the actual planning trajectory after the attack.

aggressive behavior in each time step. Moreover, the predicted trajectories are remarkably smooth, resembling normal driving patterns. This implies that filtering algorithms may struggle to detect the OMP-Attack based on frame-to-frame trajectory changes.

5.2.3 Attack Robustness

We focus on two aspects of robustness: the changed adversarial object's properties (size and orientation) on attack effectiveness, and the victim AV's deviation from the attack point. First, we evaluate the impact of varying object placement orientations on OMP-Attack. By rotating the object center along the z-axis, we set its diameter at angles of 0°, 45°, 90°, and 135° relative to the x-axis. As presented in Tab. 2, the ATD consistently remains around 6.5 meters, PRE around 2 meters, and the CR approximately 60%, regardless of orientation. This insensitivity to object orientation is attributed to our vague optimization, which treats the object as a point, diminishing the significance of orientation on attack efficacy. Second, in Fig. 5, we change the cardboard diameter from 0.1 to 0.5 meters. For the cardboards of different diameters, we place them at pre-calculated adversarial locations, and the metric results are recorded. The results clearly show that the performance of OMP-Attack is minimally affected by variations in object diameter, with ATD fluctuating within a narrow range of 0.456 meters and

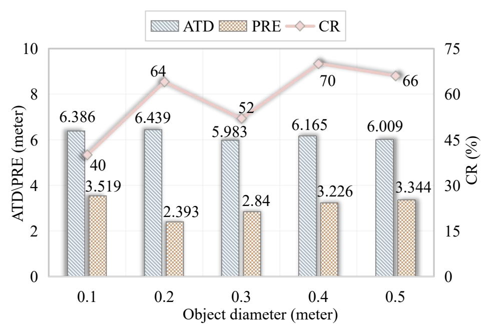


Figure 5. The impact of varying object diameters on OMP-Attack.

Table 2. The impact of different object orientations at adversarial locations on OMP-Attack. The orientation represents the angle between the object and the positive x-axis, observed from an overhead perspective along the z-axis.

Angle	ATD$(m)\downarrow$	PRE$(m)\uparrow$	CR$\uparrow$
0°	6.439	2.393	64%
45°	6.821	1.854	52%
90°	5.591	2.150	72%
135°	7.252	1.966	46%

PRE within 1.126 meters. Attack performance remains stable with fluctuations under 10%. Moreover, two key findings arise: (1) When the object is small enough to approxi-

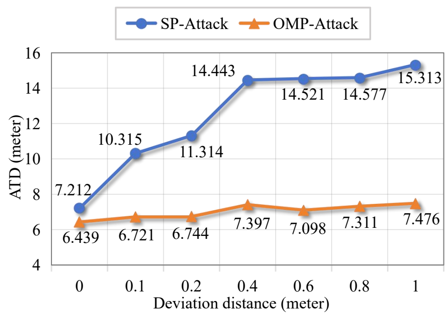


Figure 6. The impact of the deviation distance between the victim AV and attack point on attack performance.

mate a point, its attack efficiency equals that of the assumed target point in our *Vague Optimization* strategy. The smaller the object, the closer the clustered points resemble the assumed target point. For example, at a 0.1-meter diameter, the PRE is 0.1 meters higher than the 0.5-meter case, validating the assumption behind our optimization strategy. (2) As the diameter grows, the overall attack performance improves since more perturbed points are generated from the object's surface, strengthening the disruption to the prediction module. These findings verify the effectiveness of the robust attack strategy described in Sec. 4.3. The results for varying object properties on SP-Attack are in the appendix.

In Fig. 6, we explore the victim AV's deviation from the attack point in both longitudinal and lateral directions with offsets from 0 to 1 meter. The average results for each deviation distance across four directions are recorded, with the ATD results presented in Fig. 6, and appendix covering the PRE and CR results. The results show that the SP-Attack significantly weakens even with a minimal offset. For example, at a 0.1-meter offset, SP-Attack's ATD increases by 3 meters, rendering it ineffective in posing a threat to the victim AV. By contrast, our framework maintains an average ATD of 7.027 meters, compared to SP-Attack's 12.528 meters. Across a deviation range of 0 to 1 meter, OMP-Attack's maximum ATD error is only 1.037 meters, with less than 25% decay in attack strength. These results underscore our framework's robust and adaptable attack range, demonstrating its effectiveness without dependence on precise alignment with the attack point.

5.3. Attack Transferability Evaluation

To validate the generalizability of our framework, we assess its attack transferability. We first identify adversarial locations through a white-box attack on the original AD system. We then apply these adversarial locations directly to perform the black-box attack on the *Agentformer* model [33], a representative trajectory prediction model on nuScenes dataset that integrates map information and is trained under default configurations. The results in Tab. 3 illustrate that

Table 3. The average attack results of white-box attack (w-box) and black-box attack (b-box).

Attack Methods	**ATD**$(m)\downarrow$	**PRE**$(m)\uparrow$	**CR**$\uparrow$
Clean Scene (w-box)	8.855	0	0
Clean Scene (b-box)	11.083	0	0
Brute-Force (w-box)	8.196	0.998	27%
Brute-Force (b-box)	10.956	0.610	14%
SP-Attack [16] (w-box)	7.212	1.191	42%
SP-Attack [16] (b-box)	8.398	1.443	28%
OMP-Attack (w-box)	6.439	2.393	64%
OMP-Attack (b-box)	7.697	1.705	26%

the Brute-Force search fails to pose any threat in black-box scenarios. In contrast, adversarial locations identified by both OMP-Attack and SP-Attack exhibit attack transferability, and OMP-Attack achieves superior outcomes. Specifically, OMP-Attack reduces ATD by 3.386 meters and increases PRE by 1.705 meters, outperforming SP-Attack's 2.685 meters ATD reduction and 1.443 meters PRE increase. These results show that our framework induces higher prediction errors in the target model, highlighting its effectiveness in both white-box and black-box settings.

6. Conclusion

In this paper, we propose a novel *Optimization-driven Multi-frame Perturbation Framework* (OMP-Attack) to address three key challenges in indirect attacks. Our approach employs a multi-frame attack, supported by three specialized loss functions that guide the optimization of adversarial locations. Additionally, we implement a *"Precise Attack, Vague Optimization"* strategy which ignores the adversarial object's properties and treats it as a central point. Our method *simultaneously* enhances attack endurance, efficiency, and robustness. Comprehensive experiments show that our framework effectively maintains attack performance across multiple frames and minimizes reliance on the attack point and object properties. Moreover, our method demonstrates superior transferability and performs robustly even in black-box settings. We believe our work will provide valuable research insights for the community.

Acknowledgement. This work was supported by the National Key Research and Development Program of China (No.2022YFB3104502), National Natural Science Foundation of China (62441237, U24A20336, 62302326, 62272357), Wuhan Science and Technology Project for Key Research and Development (No.2024050702030090), and Wuhan Science and Technology Joint Project for Building a Strong Transportation Country (No.2023-2-7, No.2024-2-7).

References

[1] Autoware.ai. https://www.autoware.ai/, 2020.

[2] Baidu apollo. http://apollo.auto/, 2021.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020.

[4] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 2267–2281, 2019.

[5] Yulong Cao, Ningfei Wang, Chaowei Xiao, Dawei Yang, Jin Fang, Ruigang Yang, Qi Alfred Chen, Mingyan Liu, and Bo 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*, pages 176–194. IEEE, 2021.

[6] Yulong Cao, Chaowei Xiao, Anima Anandkumar, Danfei Xu, and Marco Pavone. Advdo: Realistic adversarial attacks for trajectory prediction. In *European Conference on Computer Vision*, pages 36–52. Springer, 2022.

[7] Yulong Cao, S Hrushikesh Bhupathiraju, Pirouz Naghavi, Takeshi Sugawara, Z Morley Mao, and Sara Rampazzi. You can't see me: Physical removal attacks on {lidar-based} autonomous vehicles driving frameworks. In *32nd USENIX Security Symposium*, pages 2993–3010, 2023.

[8] Yulong Cao, Danfei Xu, Xinshuo Weng, Zhuoqing Mao, Anima Anandkumar, Chaowei Xiao, and Marco Pavone. Robust trajectory prediction against adversarial attacks. In *Conference on Robot Learning*, pages 128–137. PMLR, 2023.

[9] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *ArXiv Preprint ArXiv:1902.06705*, 2019.

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9185–9193, 2018.

[11] Yinpeng Dong, Qi-An Fu, Xiao Yang, Tianyu Pang, Hang Su, Zihao Xiao, and Jun Zhu. Benchmarking adversarial robustness on image classification. In *proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2020.

[12] Simon Geisler, Tom Wollschläger, MHI Abdalla, Johannes Gasteiger, and Stephan Günnemann. Attacking large language models with projected gradient descent. *ArXiv Preprint ArXiv:2402.09154*, 2024.

[13] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred, Hao Chen, Zhenyu Zhong, and Tao Wei. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations*, 2020.

[14] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks*, pages 1942–1948. ieee, 1995.

[15] Yiming Li, Congcong Wen, Felix Juefei-Xu, and Chen Feng. Fooling lidar perception via adversarial trajectory perturbation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7898–7907, 2021.

[16] Yang Lou, Yi Zhu, Qun Song, Rui Tan, Chunming Qiao, Wei-Bin Lee, and Jianping Wang. A first Physical-World trajectory prediction attack via LiDAR-induced deceptions in autonomous driving. In *33rd USENIX Security Symposium*, pages 6291–6308, 2024.

[17] Chen Ma, Ningfei Wang, Qi Alfred Chen, and Chao Shen. Slowtrack: Increasing the latency of camera-based perception in autonomous driving using adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 4062–4070, 2024.

[18] Yanxiang Ma, Minjing Dong, and Chang Xu. Adversarial robustness through random weight sampling. *Advances in Neural Information Processing Systems*, 36, 2024.

[19] Aleksander Madry. Towards deep learning models resistant to adversarial attacks. *ArXiv Preprint ArXiv:1706.06083*, 2017.

[20] Michalis Mavrovouniotis, Changhe Li, and Shengxiang Yang. A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33:1–17, 2017.

[21] Raymond Muller, Yanmao Man, Z Berkay Celik, Ming Li, and Ryan Gerdes. Physical hijacking attacks against object trackers. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pages 2309–2322, 2022.

[22] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.

[23] Jonathan Petit, Bas Stottelaar, Michael Feiri, and Frank Kargl. Remote attacks on automated vehicles sensors: Experiments on camera and lidar. *Black Hat Europe*, 11(2015): 995, 2015.

[24] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, 2016.

[25] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust {LiDAR-based} perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th USENIX Security Symposium*, pages 877–894, 2020.

[26] Kaiyuan Tan, Jun Wang, and Yiannis Kantaros. Targeted adversarial attacks against neural network trajectory predictors. In *Learning for Dynamics and Control Conference*, pages 431–444. PMLR, 2023.

[27] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 954–962, 2020.

[28] James Tu, Mengye Ren, Sivabalan Manivasagam, Ming Liang, Bin Yang, Richard Du, Frank Cheng, and Raquel Urtasun. Physically realizable adversarial examples for lidar object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13716–13725, 2020.

[29] Aleksandar Vakanski, Iraj Mantegh, Andrew Irish, and Farrokh Janabi-Sharifi. Trajectory learning for robot programming by demonstration using hidden markov model and dynamic time warping. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 42(4):1039–1052, 2012.

[30] Ziwen Wan, Junjie Shen, Jalen Chuang, Xin Xia, Joshua Garcia, Jiaqi Ma, and Qi Alfred Chen. Too afraid to drive: systematic discovery of semantic dos vulnerability in autonomous driving planning under physical-world attacks. *In Network and Distributed System Security Symposium*, 2022.

[31] Feng Yang, Pengxiang Wang, Yizhai Zhang, Litao Zheng, and Jianchun Lu. Survey of swarm intelligence optimization algorithms. In *2017 IEEE International Conference on Unmanned Systems*, pages 544–549. IEEE, 2017.

[32] Huilin Yin, Jiaxiang Li, Pengju Zhen, and Jun Yan. Sa-attack: Speed-adaptive stealthy adversarial attack on trajectory prediction. *ArXiv Preprint ArXiv:2404.12612*, 2024.

[33] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[34] Qingzhao Zhang, Shengtuo Hu, Jiachen Sun, Qi Alfred Chen, and Z Morley Mao. On adversarial robustness of trajectory prediction for autonomous vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15159–15168, 2022.

[35] Jiaping Zhao and Laurent Itti. shapedtw: Shape dynamic time warping. *Pattern Recognition*, 74:171–184, 2018.

[36] Yi Zhu, Chenglin Miao, Tianhang Zheng, Foad Hajiaghajani, Lu Su, and Chunming Qiao. Can we use arbitrary objects to attack lidar perception in autonomous driving? In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 1945–1960, 2021.