Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.077768

ARTICLE

# A 3D Object Recovery Framework for Enhancing In-Vehicle Network Resilience to Data Tampering Attack

**Gangtao Han[1], Yurui Chen[1], Song Wang[1,*], Enqing Chen[1], Lingling Li[2] and Gaofeng Pan[3]**

[1]School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou, China
[2]School of Computer Science, Zhengzhou University of Aeronautics, Zhengzhou, China
[3]Zhengzhou Research Institute, Beijing Institute of Technology, Zhengzhou, China

*Corresponding Author: Song Wang. Email: ieswang@zzu.edu.cn

Received: 16 December 2025; Accepted: 11 February 2026; Published: 09 April 2026

**ABSTRACT:** The integrity of perception data transmitted over in-vehicle networks is important for the safety of autonomous driving. However, legacy protocols like the Controller Area Network (CAN) bus which lacks essential security features make In-Vehicle Networks (IVNs) vulnerable to data tampering attacks. Current research typically focuses on detecting the attack itself but ignores the information recovery from the missing data, leading to an unsafe autonomous driving system. To address the issue, we propose a 3D object recovery framework to recover the missing data caused by the tampering attack that occurred in in-vehicle networks. The proposed framework exploits both temporal and spatial context for the 3D object recovery, where a temporal branch is designed to learn the coordinate offsets of 3D objects based on historical data from previous frames, while a spatial branch employs information from the adjacent views of the attacked objects to locate the recovered objects from the overlapped regions in the current frame. By integrating the temporal and spatial clues, the framework effectively recovers the missing objects from the resting ones, thereby enhancing the immunity of in-vehicle networks for the tampering attack. Extensive experiments on the nuScenes dataset demonstrate that the proposed framework significantly improves 3D object detection performance under the attack when compared to the method without recovery. Additionally, the recovery performance becomes better as the attack intensity increases, highlighting the framework's robustness in high-risk scenarios. The source will be available upon publication.

**KEYWORDS:** 3D object recovery; data tampering attack; in-vehicle networks; autonomous driving security; temporal-spatial context

## 1 Introduction

The rapid development of the Internet of Things (IoT) is driving emerging trends in vehicle connectivity and digitalization for smart transportation, thereby continuously elevating the security demands in in-vehicle networks [1,2]. Autonomous driving is leading a major transformation in the automotive industry, driven by the demand for enhanced safety and efficiency [3]. Current autonomous driving systems operate on a perpetual cycle of sensing, perception, decision, planning and action [4], as shown in Fig. 1. To complete this cycle, the In-Vehicle Networks (IVNs) are typically utilized to ensure the effective transmission of information. In a vehicle, IVNs are specifically designed to connect various electronic components and the exchange of data to support different functionalities [5]. Reliable IVNs, such as the Controller Area Network (CAN) bus, are crucial for transmitting environment data from sensors to the Electronic Control Unit

(ECU) [6]. In perception tasks, real-time transmission of 3D object detection maps is extremely important for accurate awareness [7,8].

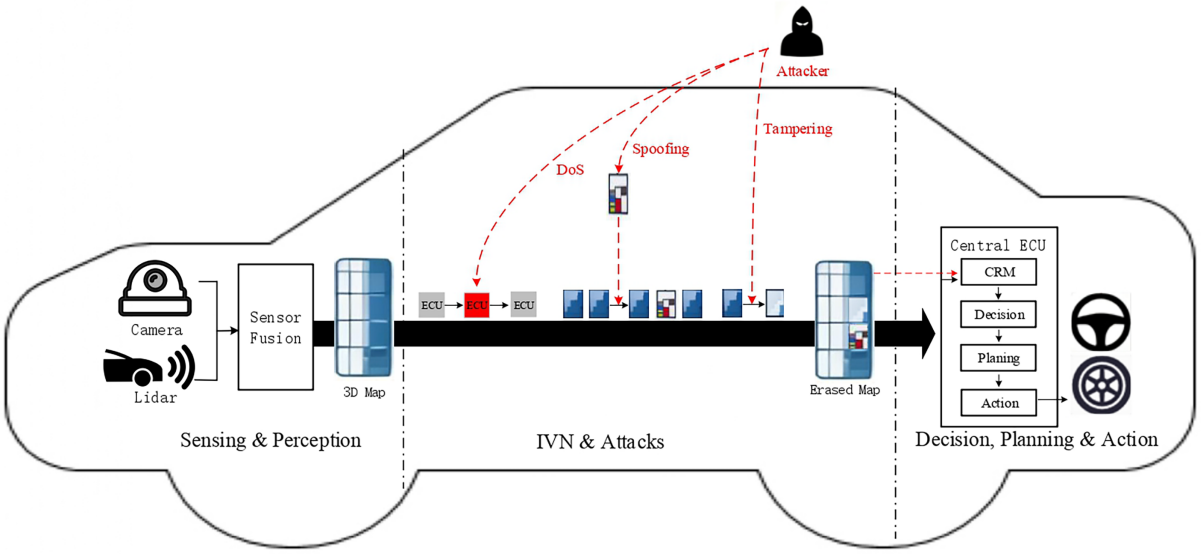



**Figure 1:** Illustration of the perpetual cycle in current autonomous driving systems and different types of attacks act on the IVN during data transmission.

Although there are a variety of protocols in modern IVNs, the CAN bus remains the most widespread protocol for critical functions because of its reliability and low cost [9,10]. However, the CAN bus lacks essential security features like message authentication and data encryption, making it an open broadcast medium and thus an easy target for cyber-attacks [11,12]. This flaw may cause data tampered with during the transmission process in the CAN bus leading to severe traffic accidents [13].

Many protocols in IVNs share similar security flaws, which create a broad attack surface. As illustrated in Fig. 1, we show several types of attacks that act on IVNs. For example, the Denial-of-Service (DoS) attack disrupts critical functions by attacking ECUs, while the message spoofing attack injects forged data into ECUs [14,15]. In this work, we focus on a specific type of data integrity attack: the data tampering attack, in which an adversary alters transmitted data. Such an attack mainly presents two challenges. First, it simulates non-malicious sensor faults, allowing it to bypass conventional detection systems when the data tampering attack is imposed on the data uploaded by the sensors [16]. Second, by creating a persistent perceptual void, it damages the integrity of the input data provided for the decision-making module, which may lead to serious accidents such as incorrect path planning or collisions [17]. Fig. 1 illustrates different types of attacks in the perpetual cycle of current autonomous driving systems.

Currently, research on how to defend against these attacks mainly focus on designing Intrusion Detection Systems (IDS) to monitor malicious activity in IVNs [18,19]. Gu et al. [20] exploit the physical characteristics of transmitted signals as unique device fingerprints for identity verification. Lalouani et al. [21] further utilize combined transmissions to obfuscate attackers, enhancing the robustness of intrusion detection systems. There are also several research works that attempt to disrupt illegal messages in IVNs. For instance, Dong et al. [22] propose to use the arbitration mechanism of the CAN bus to interrupt the unauthorized message transmission, mitigating the adverse impact of the illegal messages. Although these methods are effective at detecting and disrupting attacks, they fail to recover useful information from the manipulated data. An autonomous vehicle is insufficient because the critical information has already

been lost or compromised. Although these existing methods are able to detect attacks or disrupt illegal messages, they fail to recover the missing information from the manipulated data. The information missing tends to harm the performance of environment perception for the self-driving vehicles, resulting in invalid autonomous driving systems [23].

To address the issue, this paper proposes a 3D object recovery framework to predict the missing objects caused by the tampering attack occurred in IVNs, thereby enhancing the robustness of the autonomous driving system against attacks. Specifically, a Collaborative Recovery Module (CRM) is designed to predict the categories and positions of the missing objects by exploiting both temporal and spatial context. In temporal dimension, CRM utilizes historical detection maps to model the coordinate offsets of the objects and predict their moving trajectory. In addition, we also introduce the spatial information into CRM to fully exploit the adjacent views of the missing objects in the current frame to guide the recovery of the missing objects. With the integration of the temporal and spatial information, the proposed framework accurately recovers the missing objects, thereby improving the robustness of perception systems under the tampering attack. It is worth noting that CRM sits on the Central ECU. It ensures that perception results tampered with during transmission over the IVNs are recovered before decision-making, as shown in Fig. 1.

The main contributions of our work are summarized as follows:

- We propose a 3D object recovery framework to predict the categories and positions of the missing objects caused by the tampering attack that occurred in IVNs.
- To recover the missing objects, CRM is designed to utilize both temporal and spatial detection maps. The temporal information is utilized to learn coordinate offsets from historical data, while the spatial branch employs the adjacent views in the current frame to predict the position of missing objects based on overlapping regions.
- Extensive experiments on the nuScenes benchmark [24] demonstrate that the proposed framework effectively recovers the missing objects, thereby enhancing the robustness of perception systems under tampering attacks in IVNs. Furthermore, the recovery performance tends to be better as the attack intensity increases indicating the framework's effectiveness in high-risk scenarios.

## 2 Related Work

In this section, we first review various attack methods targeting data integrity. Then, we introduce IDS methods which are designed to tackle the security threats in IVNs of autonomous driving systems. At last, the recovery strategies for perception results are reviewed.

### *2.1 Data Integrity Attack*

The attacks in IVNs are diverse and increasingly complex, ranging from network disruptions to targeted data tampering attacks. For example, the DoS attacks aim to disrupt critical functions by flooding the bus with messages. Message spoofing attackers inject false data to mislead the system. Dasgupta et al. [15] simulate scenarios such as creating false turns or stops by falsifying vehicle position and velocity information. Periodic replay attacks attempt to replay previously recorded sensor data cyclically rather than once [25]. Fuzzing attacks use malformed data to identify system weaknesses [26]. Different from the above attacks, this paper focuses on the data tampering attack, where the attack alters the payload of legitimate messages, such as modifying critical control data on the CAN bus [17]. Our work specifically addresses the tampering of high-level perception data by invalidating the coordinates of detected objects. The stealth of this approach is a significant concern, as it can simulate non-malicious sensor faults and bypass conventional detection systems [16].

### 2.2 Intrusion Detection System

Recently, most research efforts addressing security threats in autonomous driving have focused on developing IDS [18,19]. These systems monitor the network traffic in real time, such as the data transmitted on the CAN bus, to identify malicious activities [27,28]. The research works on IDS are typically categorized into physical-layer methods and data-layer methods. The physical-layer methods verify message authenticity by leveraging unique hardware features, such as ECU voltage fingerprints [21,29], effectively countering spoofing attacks. However, the requirement for hardware modifications imposes significant limitations on the deployment of these physical-layer methods. In contrast, the data-layer methods are more readily deployable as they detect anomalies by analyzing statistical patterns in CAN traffic, like message ID frequency [22,30]. Nonetheless, their robustness to advanced attacks, such as the mimicing normal behavior, is poor. More recently, the learning-based IDS have demonstrated superior performance by employing deep neural networks to capture complex temporal dependencies in message sequences [31,32], though they often face challenges with data dependency and generalization. Despite of the accurate and real-time attack detection achieved by the above methods, most of them fail to recover important perception data [12,23]. For the decision-making unit of autonomous vehicles, it is insufficient to simply detect the attack and issue a warning [33]. The unit requires sufficient information to make the right decision and keep the smooth and safe operations of autonomous vehicles. Therefore, such a "detect-but-not-recover" paradigm falls short of the requirements of reliable autonomous driving systems.

### 2.3 Perception Result Recovery

In the field of environmental perception, various recovery methods have been explored to handle missing or corrupted data. At the object level, traditional recovery methods often rely on kinematic assumptions. For instance, the Kalman Filter is utilized to estimate object states during short-term occlusions. At the pixel level, image inpainting methods aim to fill in incomplete regions of an image [34]. There are some methods which formulate image restoration as a constrained optimization problem, where Generative Adversarial Networks (GANs) are employed to estimate the missing regions of the original [35]. More recently, diffusion models have shown their potential in image inpainting. Pang et al. [36] propose diffusion-based face image restoration method, combining progressive sampling with sample scheduling to improve the quality of the restored images. Wang et al. [37] design a diffusion self-attention and a diffusion feed-forward network to tackle the severe degradation of images. These methods mentioned above mainly focus on visual data recovery, which cannot be applied in the object recovery task in autonomous driving systems.

The most related work, M-BEV [38], is a feature-level approach which attempts to recover the missing feature maps caused by the camera failure. By randomly masking camera views and reconstructing, M-BEV learns to reconstruct the bird's-eye view (BEV) features of the masked views to boost the robustness of 3D perception. M-BEV improves the tolerance of the perception system for the data missing but is unable to achieve the 3D object recovery after the detected object information is tampered with in IVNs. This paper focuses on recovering the missing objects when IVN has already been attacked. Based on the history detection results and current unmanipulated ones from adjacent camera views, this paper infers the categories and positions of missing objects, thereby enhancing the immunity of IVNs.

While the existing methods mainly focus on image-level or feature-level reconstruction, this paper proposes to directly recover the objects when a tampering attack occurs in IVNs. Based on the history detection results and current unmanipulated ones from adjacent camera views, this paper aims to infer the categories and positions of missing objects. The main benefit of the proposed method is its rapid recovery of missing objects, since it avoids the need to re-execute the perception model, which is very time-consuming. The traditional methods, e.g., Kalman Filter, also intend to predict the possible trajectory of the objects.

However, they rely on the linear motion assumption, which fail in complex urban driving behaviors, such as sudden braking and sharp turns.

There are also tracking [39], motion prediction [40] methods which usually utilize the current state of the objects to predict their action in the next timestep. However, these methods rely on reliable sensor input, which is hard to satisfy in the data tampering scenario. While BEV completion methods [41] focus on feature-level imputation for sensor failures, this paper intends to recover the missing targets in object-level. The proposed object-level recovery mechanism is able to efficiently recover missing targets without re-executing the perception model, supporting real-time and robust operation of the autonomous driving system against attacks.

## 3 Method

In this section, we first present the overall 3D object recovery framework. Then, a detailed presentation of CRM is provided. Finally, we illustrate the training strategy and the inference pipeline.

### *3.1 Overall Framework*

To tackle the potential tampering attack in IVNs, we propose a 3D object recovery framework to predict the categories and positions of the missing objects. The overall framework is shown in Fig. 2. Since the multi-camera perception technique becomes the most popular 3D object detection solution in the autonomous driving system [42,43], we follow the same technique roadmap to generate the detection map for each view's image. Although current multi-camera perception models are likely to generate a unified detection map by integrating all the cameras [44], this paper chooses the scenario where detection maps of different cameras are transmitted separately in IVNs. That is because the distributed transmission reduces the information loss when IVNs get attacked [9]. Given the detection maps, the autonomous driving system transmits them to ECU via IVN. When a view is compromised (e.g., front), CRM is triggered to recover the missing objects. To focus on the recovery, we assume that the tampering attack has already been detected and the proposed framework is designed to deal with the missing objects. At first, the previous frame before the attack provides the initial categories and positions of the missing objects. Then, history frames are inputted into the Temporal Recovery Branch (TRB) to predict motion offsets of the missing objects. The Spatial Recovery Branch (SRB) is designed to exploit useful information from the unattacked detection maps in the current timestamp $t$. With the integration of the temporal and spatial information, the proposed framework enables accurate recovery of missing objects.

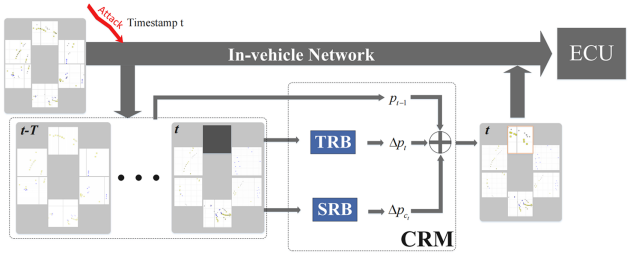



**Figure 2:** Overview of the 3D object recovery framework. Multi-camera detection results are transmitted over IVN. A simulated attack corrupts the output from the "Front" view on the timestamp $t$. CRM explores both the temporal and spatial context to predict the coordinate offsets of the missing objects. After the recovery of missing objects by CRM, the detection maps are fed back to IVN and transmitted to ECU.

In the real world, attackers tend to partially corrupt data to lower the probability of being detected. This paper considers the whole view in which the manipulated data are located to be unreliable. There are two main reasons for the consideration. (1) Locating the attacked view is much easier than locating every single manipulated object. (2) The "zeroing an entire camera view out" strategy in this paper provides a more efficient solution than the one which attempts to recover the manipulated objects one-by-one. Therefore, CRM is designed to recover all the objects in the whole zeroing-out view.

### *3.2 Collaborative Recovery Module*

The Collaborative Recovery Module (CRM) serves as the central component of framework, aiming to recover the attacked objects by integrating spatial and temporal information. Compared with traditional methods, such as Kalman filtering, CRM is able to model complex and nonlinear object motion. All results processed by CRM are represented in a global BEV coordinate system, ensuring that object positions are comparable across different camera views.

Given the current timestamp $t$ when the IVN get attacks, the category and position of a missing object is defined as $o_i = \left[c_t^i,\ p_t^i\right],\ i = 1, \ldots,\ N_t$, where $N_t$ stands for the number of missing objects in the current frame. To recover the missing object $o_i$, we leverage temporal consistency to directly inherit the category $c_{t-1}^i$ from the historical instance at timestamp $t-1$. And for each missing object, CRM receives its last historical position $p_{t-1}^i$ as the benchmark. These inputs enable CRM to perform recovery without relying on raw camera images. The position of the missing object $o_i$ is defined as $p_t^i = p_{t-1}^i + \Delta p^i$. The position $p_t^i$ is a 3-dimension vector $\left[x_t^i,\ y_t^i,\ z_t^i\right]$ to represent the coordinate in the world space. This paper designs a dual branch network, i.e., TRB and SRB, to predict $\Delta p^i$.

#### *3.2.1 Temporal Recovery Branch*

To model the complex and non-linear motion of the 3D objects in roads, we design the temporal recovery branch with the Transformer architecture to perceive the global dependency. It operates on historical detection maps from timestamps $t-m$ to $t-1$. Each detection map contains the 3D coordinates and categories of all objects in the scene. For every missing object in timestamp $t$, the framework retrieves its instance across these historical frames from $t-m$ to $t-1$. The input of TRB is a carefully designed feature vector $x_t^i$ that combines this spatiotemporal context. For the $i$-th missing object, its corresponding feature vector is constructed as follows.

$$x_t^i = \left[p_{(t-1)}^i, \Delta p_{(t-1)}, s_t^i, t, r_t^i\right], \tag{1}$$

where $p_{t-1}^i$ stands for the position of the $i$-th missing object in timestamp $t-1$. The second component $\Delta p_{t-1}$ represents the object's instantaneous velocity by calculating its motion from $t-2$ to $t-1$. To capture the global traffic flow of the $c_t^i$-class objects along with the temporal dimension, an average position $s_t$ is constructed by computing the mean coordinate of all the $c_t^i$-class objects in the bilateral symmetric view from $t-m$ to $t$, where $m$ stands for the number of history frames. In cases where no symmetric objects exist (e.g., asymmetric roads or low traffic density), $s_t$ is set to a zero vector. The objects from the bilateral symmetric view typically have similar moving trajectories, providing a great reference to predict the position offset. The component $t$ is the current timestamp which provides the time information for the network. We also encode the position of the missing object. For the $i$-th missing object, we divide the corresponding view into three area, i.e., the overlap area with the left view, the exclusive area, and the the overlap area with the right view. If its initial position $p_{t-1}^i$ is located in the overlap area with the left view, its encoding vector is $r_t^i = [1,\ 0,\ 0]$. The components in $r_t^i$ stand for the left overlap area, the exclusive area and right overlap area separately.

The architecture of TRB is shown in Fig. 3. For the input vector $x_t^i$, it is firstly to be mapped into a 256-dimensional latent space via a linear embedding layer, forming a unified feature token. Subsequently, a multi-head cross-attention module is employed to learn information from history frames. Within the module, features related to the object's own historical motion serve as the Query, while features representing the external spatial environment act as the Key and Value. This mechanism allows the model to adaptively attend to the most relevant spatial cues based on the object's own movement history. The resulting context-enriched token is then processed by a 4-layer Transformer encoder stack to capture more complex dependencies. Finally, a lightweight decoder head (MLP) regresses the high-dimensional features to the final prediction. Thus, the entire process is represented by:

$$\Delta p_t^i = f_T\left(x_t^i\right), \tag{2}$$

where $f_T$ is the designed network. The output vector $\Delta p_t^i$ serves as a context-aware motion prior for the final coordinate recovery.

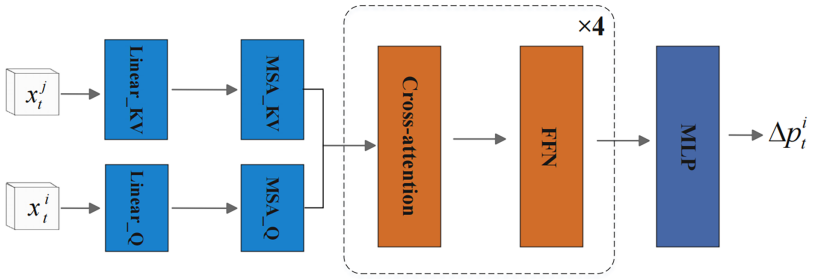



**Figure 3:** The architecture of TRB. The branch constructs the feature vectors $x_t^i$ of missing objects $o_i$, $(i = 1, \ldots, N_t)$ and leverages a cross-attention mechanism to learn the motion offset $\Delta p_t^i$.

### *3.2.2 Spatial Recovery Branch*

While the temporal module predicts object motion in an open-loop manner, SRB provides a closed-loop correction. It extracts objects from adjacent camera views whose 3D coordinates fall within the overlap regions of the attacked view. These regions are determined using camera calibration parameters under the unified global BEV coordinate representation to ensure the view-consistency. Instead of tracking each missing object, SRB relies on a simple idea that objects with the same class usually move in a similar way within a short time. This pattern is used to estimate a reliable correction for the missing objects.

For the object $o_i$ with the class $c_t^i$, we first extract all the $c_t^i$-class objects from the overlap areas between the attacked view and the adjacent views. Then, as shown in Fig. 4, their average moving distance between the previous frame $t-1$ and the current one $t$ are represented as follows.

$$\overline{\Delta p_{c_t^i}} = \frac{\sum_{j=1}^{M}\left(p_{t,j} - p_{t-1,j}\right)}{M}, j = 1, \cdots, M, \tag{3}$$

where $M$ stands for the number of the $c_t^i$-class objects from the overlap areas.

This vector $\overline{\Delta p_{c_t^i}}$ represents an average motion prior for the $c_t^i$-class objects. If no $c_t^i$-class objects are found in the overlap areas, $\overline{\Delta p_{c_t^i}}$ becomes a zero vector. In the final integration stage, it serves as a scene-flow-based observational correction to the motion prediction from the temporal branch.

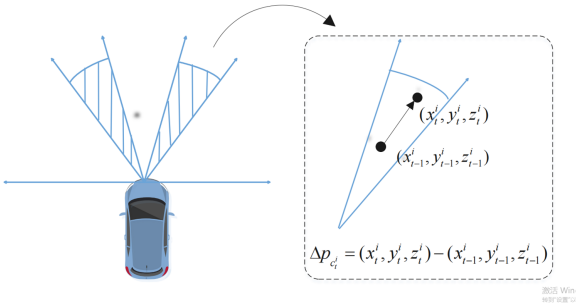



**Figure 4:** An illustration of how the class-specific motion vector is computed in SRB. The average motion between the timestamp $t-1$ and $t$ is computed by averaging the motion vectors of same-class objects found in the adjacent views.

To obtain predict $\hat{\Delta} p^i$, we integrate the position offsets from TRB and SRB as Eq. (4).

$$\hat{\Delta} p^i = \Delta p_t^i + \overline{\Delta p_{c_t^i}}. \tag{4}$$

Then, the ultimate prediction result for the object $o_i$ is formulated as Eq. (5).

$$\hat{p}_t^i = p_{t-1}^i + \hat{\Delta} p^i. \tag{5}$$

It should be noted that the recovery process relies on the correct association between missing objects and their historical tracks. While association errors occur in dense traffic, the dual-branch design helps mitigate this risk, as the SRB provides spatial corrections that are independent of historical data.

### *3.3 Training and Inference*

This subsection introduces the training strategy of the proposed recovery framework and the inference pipeline.

#### *3.3.1 Training*

To learn the recovery model, we use the L1 loss to constrain the predicted position of the $i$-th object $o_i$ as Eq. (6).

$$\mathcal{L}^i = ||p_t^i - \hat{p}_t^i|| \tag{6}$$

By minimizing the loss $\mathcal{L}^i$, the model is trained to accurately predict the position of the missing object $o_i$.

For the model training, we set the same training strategy with the baseline method BEVFormer [45]. The network is optimized using the AdamW optimizer [46], with an initial learning rate of $2 \times 10^{-4}$ scheduled via cosine annealing over 24 epochs.

#### *3.3.2 Inference*

During inference, the proposed recovery framework is activated whenever a data tampering attack is detected in IVNs. Once the tampered view has been identified, the detection outputs with invalid values are marked as missing objects. CRM performs recovery on each missing object individually by (1) retrieving its last valid position at timestamp $t-1$, (2) collecting historical motion cues for TRB, and (3) extracting

same-class objects in adjacent views at timestamp $t$ for SRB. To provide a clear view of the recovery process, the overall workflow of the framework is summarized in Algorithm 1.

**Algorithm 1:** The overall workflow of CRM

**Input:** History detection results from $t - m$ to $t - 1$; Current attacked detection map at $t$.
**Output:** Recovered object categories $c_t^i$ and 3D positions $\hat{p}_t^i$
1. Identify missing object placeholders in the tampered view;
2. **For** each missing object $o_i$ **do**:
3. Inherit category $c_t^i$ from history $c_{t-1}^i$;
4. Predict motion offset $\Delta p_t^i$ via TRB using historical motion cues (Eq. (2));
5. Calculate class-specific correction $\overline{\Delta p_{c_t^i}}$ via SRB from adjacent views (Eq. (3));
6. Fuse offsets (Eq. (4));
7. Update final position (Eq. (5));
8. End **For**
9. Return recovered 3D perception map.

The pipeline operates on each compromised object placeholder by first linking it with its historical instance. If the correspondence is found, the module proceeds with a two-pronged recovery. The object's category is restored via direct temporal inheritance. Meantime, its coordinate is recovered by integrating the temporal and spatial offsets as detailed in Eq. (5).

## 4 Experiments

In this section, we present extensive experiments to evaluate the proposed 3D object recovery framework. Specifically, we assess the performance of the framework on the large-scale nuScenes benchmark [24], and compare it with the reference method [45] to validate its effectiveness and superiority. Additionally, we perform comprehensive experiments to analyze the impact of the number of missing views and the attack pattern on the recovery performance.

### *4.1 Datasets and Metrics*

We conduct our experiments on the nuScenes benchmark [24], a large-scale dataset for autonomous driving that captures diverse urban scenes using a 360-degree multi-camera system. Following the official data splits and evaluation protocol, we assess performance using the nuScenes Detection Score (NDS), mean Average Precision (mAP), and five True Positive (TP) error metrics: mATE, mASE, mAOE, mAVE, and mAAE [47–49]. For NDS and mAP, higher values are better, while lower values mean better performance in TP error metrics. Notably, we adopt NDS and mAP as they provide an evaluation of both category recovery and position accuracy, where an object is considered a True Positive only if its category is correctly identified.

### *4.2 Experimental Settings*

We adopt BEVFormer [45] as the 3D object detection reference model due to its promising performance. Specifically, we use the BEV Former-Small network which replaces the temporal self-attention module with standard self-attention, reducing the requirements for the computation resource. Input images are resized to 900 × 1600, with a BEV perception range of [−51.2, 51.2 m] and a BEV grid resolution of 200 × 200. The performance of this trained model on the nuScenes validation set (see Table 1) serves as the baseline for all subsequent attack and recovery evaluations.

**Table 1:** NDS and mAP results under 25%, 50%, 75%, and 99% attack intensities. "All" represents the standard reference (BEVFormer: NDS 0.4504, mAP 0.3518). "Attack" denotes results under attack, and "Recovery" displays the results of the proposed recovery method.

| View | Method | Metric | Baseline | |
|---|---|---|---|---|
| **All** | **BEVFormer** | **NDS (↑)**<br>**mAP (↑)** | **0.4504**<br>**0.3518** | |
| **View** | **Attack Intensity** | **Metric** | **Attack** | **Recovery** |
| Front_Left | 25% | NDS (↑) | 0.3823 | **0.3891** |
| | | mAP (↑) | 0.2326 | **0.2466** |
| | 50% | NDS (↑) | 0.3700 | **0.3831** |
| | | mAP (↑) | 0.2084 | **0.2351** |
| | 75% | NDS (↑) | 0.3571 | **0.3757** |
| | | mAP (↑) | 0.1848 | **0.2226** |
| | 99% | NDS (↑) | 0.3391 | **0.3601** |
| | | mAP (↑) | 0.1572 | **0.1973** |
| Front | 25% | NDS (↑) | 0.3808 | **0.3876** |
| | | mAP (↑) | 0.2290 | **0.2436** |
| | 50% | NDS (↑) | 0.3659 | **0.3783** |
| | | mAP (↑) | 0.1988 | **0.2261** |
| | 75% | NDS (↑) | 0.3524 | **0.3706** |
| | | mAP (↑) | 0.1732 | **0.2106** |
| | 99% | NDS (↑) | 0.3375 | **0.3555** |
| | | mAP (↑) | 0.1483 | **0.1842** |
| Front_right | 25% | NDS (↑) | 0.3809 | **0.3893** |
| | | mAP (↑) | 0.2293 | **0.2469** |
| | 50% | NDS (↑) | 0.3661 | **0.3827** |
| | | mAP (↑) | 0.2005 | **0.2328** |
| | 75% | NDS (↑) | 0.3506 | **0.3747** |
| | | mAP (↑) | 0.1744 | **0.2194** |
| | 99% | NDS (↑) | 0.3359 | **0.3604** |
| | | mAP (↑) | 0.1507 | **0.1947** |
| Back_Left | 25% | NDS (↑) | 0.3808 | **0.3890** |
| | | mAP (↑) | 0.2298 | **0.2466** |
| | 50% | NDS (↑) | 0.3647 | **0.3810** |
| | | mAP (↑) | 0.1966 | **0.2312** |
| | 75% | NDS (↑) | 0.3501 | **0.3733** |
| | | mAP (↑) | 0.1685 | **0.2172** |
| | 99% | NDS (↑) | 0.3326 | **0.3563** |
| | | mAP (↑) | 0.1389 | **0.1871** |

(Continued)

**Table 1 (continued)**

| View | Method | Metric | Baseline | |
|---|---|---|---|---|
| **All** | **BEVFormer** | **NDS (↑)**<br>**mAP (↑)** | **0.4504**<br>**0.3518** | |
| **View** | **Attack Intensity** | **Metric** | **Attack** | **Recovery** |
| Back | 25% | NDS (↑) | 0.3747 | **0.3827** |
| | | mAP (↑) | 0.2166 | **0.2334** |
| | 50% | NDS (↑) | 0.3549 | **0.3683** |
| | | mAP (↑) | 0.1759 | **0.2051** |
| | 75% | NDS (↑) | 0.3361 | **0.3535** |
| | | mAP (↑) | 0.1381 | **0.1775** |
| | 99% | NDS (↑) | 0.3167 | **0.3337** |
| | | mAP (↑) | 0.1050 | **0.1389** |
| Back_right | 25% | NDS (↑) | 0.3802 | **0.3886** |
| | | mAP (↑) | 0.2310 | **0.2475** |
| | 50% | NDS (↑) | 0.3667 | **0.3819** |
| | | mAP (↑) | 0.2075 | **0.2359** |
| | 75% | NDS (↑) | 0.3548 | **0.3768** |
| | | mAP (↑) | 0.1849 | **0.2261** |
| | 99% | NDS (↑) | 0.3376 | **0.3613** |
| | | mAP (↑) | 0.1573 | **0.1988** |

Note: ↑ denotes that higher values are better.

To evaluate the robustness of our recovery framework, we simulate perception corruption by directly modifying the outputs of BEVFormer-Small. This mimics data tampering attacks in IVNs [50,51]. Specifically, we design a targeted partial information erasure strategy. For each selected camera view, the coordinates and category labels of all detected 3D objects are set to zero. Such a design simulates the attack situation where data is damaged by the data tampering attack. In experiments, we set three levels of attack intensity—25%, 50%, 75% and 99%—based on the proportion of corrupted frames in the latter half of each scene. At each intensity level, only one camera view is attacked, while the others are normal. We focus on continuous attack modeling as a representative evaluation scenario. This design reflects the realistic behavior of attackers who aim to maximize system impact through sustained interference, rather than isolated perturbations.

### *4.3 Experimental Results*

#### *4.3.1 Quantitative Analysis*

To quantitatively assess the effectiveness of the proposed recovery framework, we evaluate its performance under the simulated data tampering attack, where the detection map of one camera is invalid. We simulate four levels of attack intensity—25%, 50%, 75%, and an extreme case of 99%—by corrupting perception results from one specific-view camera in the latter half of each scene.

Table 1 compares the 3D object detection performance before attack, after attack and after using our recovery method, across different camera regions and attack intensities. We report NDS and mAP values in Table 1.

The results show that the proposed framework improves detection performance under all attack conditions. For example, in the Front_Left view under 50% attack intensity, NDS drops from 0.4504 to 0.3700 due to the attack. The proposed framework recovers it to 0.3831, improving its detection performance when getting an attack. Similarly, the mAP increases from 0.2084 (attacked) to 0.2351 (recovered), demonstrating the effectiveness of the proposed framework. Additionally, under the extreme 99% attack on the Back view—the most challenging case where mAP plummets to 0.1050—our method restores the performance to 0.1389. The performance gap in the "Back" view is primarily due to the nuScenes sensor setup, where the back camera has a larger Field of View (FOV) and smaller overlap regions with adjacent views, limiting the effectiveness of spatial corrections.

It is noticeable that the gains from the proposed recovery framework tend to increase along with the rising of the attack intensity, illustrating the superiority of the proposed method in high-risk scenarios. Furthermore, the results across different attack intensities reflect the sensitivity to history length m. As the intensity increases, the available valid history decreases, leading to a corresponding drop in recovery precision.

We also conduct a detailed analysis on the metrics mATE, mASE, mAOE, mAVE and mAAE to show the effectiveness of the recovery framework. Table 2 shows the various metric values under 50% attack intensity. From Table 2, it is observed that the attack significantly degrades localization accuracy. The increase of mATE values verifies that the loss of 3D object information weakens the ability to locate the objects. Interestingly, we observe that in some cases, certain metrics appear to improve after an attack. For example, the mAVE changes from 0.4681 to 0.4433 in the Front_Left. This is because the attack removes many high-error cases, such as fast-moving or hard-to-detect objects, leading to a lower average error. We count the number of detected objects on each attacked scenario. Taking the Front_Left as an example, the number of the detected objects decrease from 1,298,239 to 1,274,331 under an attack. After recovery, the number of the detected objects increases to 1,278,581. Thus, the proposed recovery method may cause slight rises in some metrics like mAVE, but achieves a more accurate detection results in real-world traffic scenarios.

**Table 2:** The performance of the recovery framework across various TP error metrics under the 50% attack intensity.

| Method | MATE (↓) | MASE (↓) | MAOE (↓) | MAVE (↓) | MAAE (↓) |
|---|---|---|---|---|---|
| Baseline | | | (Detections #: 1,298,239) | | |
| BEVFormer | 0.7586 | 0.2936 | 0.4284 | 0.4681 | 0.2311 |
| Front_Left | | | (Attack/Recovery #: 1,274,331/1,278,581) | | |
| Attack | **0.9457** | 0.2849 | 0.4384 | **0.4433** | 0.2290 |
| Recovery | 0.9471 | **0.2846** | **0.4382** | 0.4457 | **0.2287** |
| Front | | | (Attack/Recovery #: 1,272,830/1,278,990) | | |
| Attack | **0.9437** | **0.2845** | **0.4413** | **0.4440** | **0.2223** |
| Recovery | 0.9468 | 0.2846 | 0.4439 | 0.4484 | 0.2232 |
| Front_Right | | | (Attack/Recovery #: 1,273,161/1,278,905) | | |
| Attack | 0.9424 | **0.2844** | 0.4482 | 0.4473 | **0.2190** |
| Recovery | **0.9423** | 0.2845 | **0.4424** | **0.4468** | 0.2213 |
| Back_Left | | | (Attack/Recovery #: 1,271,319/1,279,946) | | |
| Attack | **0.9412** | 0.2855 | **0.4440** | **0.4396** | **0.2257** |
| Recovery | 0.9450 | **0.2853** | 0.4442 | 0.4460 | 0.2260 |

(Continued)

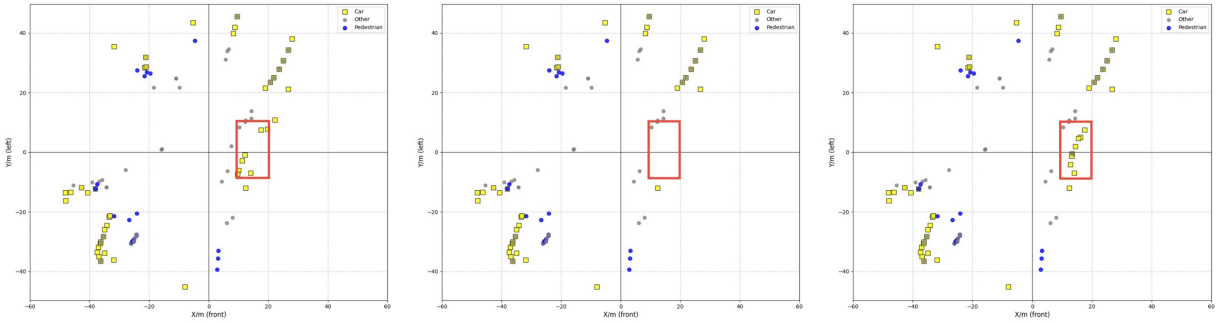
**Table 2 (continued)**

| Method | MATE (↓) | MASE (↓) | MAOE (↓) | MAVE (↓) | MAAE (↓) |
|---|---|---|---|---|---|
| Back | | (Attack/Recovery #: 1,265,547/1,281,458) | | | |
| Attack | **0.9399** | 0.2860 | **0.4082** | **0.4587** | 0.2374 |
| Recovery | 0.9448 | **0.2859** | 0.4144 | 0.4616 | **0.2355** |
| Back_Right | | (Attack/Recovery #: 1,272,510/1,278,939) | | | |
| Attack | 0.9506 | 0.2858 | 0.4440 | 0.4594 | 0.2301 |
| Recovery | **0.9506** | **0.2857** | **0.4396** | **0.4547** | **0.2296** |

Note: ↓ denotes that lower values are better.

### *4.3.2 Qualitative Results*

We visualize detection results to show the effect of the tampering attack and the recovery of our method in Fig. 5. We use yellow square markers to represent cars and blue circular markers to pedestrians, as they are the most important obstacle types in traffic. The other less common types of obstacles are uniformly classified as "other" and marked with gray circles. Fig. 5a shows the dense vehicle and pedestrian detection results of the standard BEVFormer model. After the tampering attack on the front view, most frontal objects disappear, creating a serious blind spot, as shown in the red box of Fig. 5b. Fig. 5c illustrates the recovery results where most of the missing 3D objects are recovered. Although there are some lost objects and position deviations of the recovered objects, most of the missing objects are successfully recovered, illustrating its application potential in the real-world autonomous driving task.




(**a**) Standard BEVFormer (**b**) Attack (Front View) (**c**) Recovery (Front View)

**Figure 5:** Qualitative results. (**a**) The detection results achieved by Standard BEVFormer. (**b**) The detection results under the tampering attack to the front view. (**c**) The detection results achieved by the recovery framework.

### *4.3.3 Ablation Study*

To evaluate the contribution of each component in the CRM framework, we conduct an ablation study, with results shown in Table 3. The experiments are performed under the 50% attack intensity, and the performance metrics are averaged across 6 camera views. The baseline, with no recovery mechanism, achieves an NDS of 0.3659. Introducing the basic spatiotemporal context improves the NDS to 0.3716, demonstrating that the context provides a solid foundation for the recovery task.

Next, integrating either TRB or SRB individually results in further improvements in both NDS and mAP, confirming their contributions to the recovery process. Finally, the full model, combining both TRB and SRB, achieves the highest performance with an NDS of 0.3783. This highlights the collaborative effect of

combining both components, confirming that the combination of the context, TRB, and SRB is essential for the overall effectiveness of the CRM.

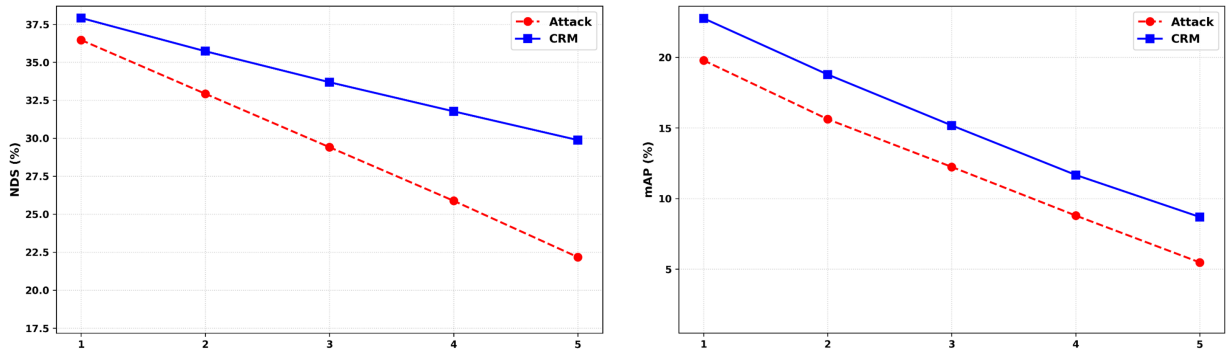
**Table 3:** Ablation study of each component of the CRM. We report the average NDS and mAP across 6 views under 50% attack intensity.

| Context | TRB | SRB | NDS (↑) | mAP (↑) |
|---|---|---|---|---|
| ✗ | ✗ | ✗ | 0.3659 | 0.1988 |
| ✓ | ✗ | ✗ | 0.3716 | 0.2180 |
| ✓ | ✓ | ✗ | 0.3720 | 0.2191 |
| ✓ | ✗ | ✓ | 0.3727 | 0.2206 |
| ✓ | ✓ | ✓ | **0.3783** | **0.2261** |

Note: ↑ denotes that higher values are better.

### *4.3.4 Multi-View Attack Analysis*

To further assess the robustness of our recovery framework in realistic scenarios, we simulate the scenario that multiple camera views are under the data tampering attack at the same time. We consider the cases where from one to five views are tampered with under the 50% attack intensity. This paper reports the average NDS and mAP along with the increase of the number of the attacked views in Fig. 6.




(**a**) NDS (**b**) mAP

**Figure 6:** Average NDS and mAP as the number of attacked views increases under the 50% attack intensity. (**a**) NDS, (**b**) mAP.

From Fig. 6, it is observed that both NDS and mAP degrade monotonically with the increase in the number of attacked views. Compared to the unrecoverable model, our model shows a lower degradation across all the attack scenarios. Notably, even under extreme conditions where five out of six camera views are attacked, our model maintains a considerable perception performance, preventing the failure of the autonomous driving system.

We also conduct a detailed comparison of recovery performance under different spatial attack patterns. There are 3 attack patterns compared in this paper, i.e., cross-view attack, consecutive attack and random attack. The cross-view attack stands for that the attack occurs every other view, while the consecutive views are attacked for the consecutive attack pattern. In experiments, we set 3 views that get the attack. For the random attack, we randomly mask 3 views to simulate the attack. The experimental results are shown in Table 4. It is observed that the consecutive attack pattern has the strongest damage power for the 3D object detection, where the recovery performance is worse. In contrast, the proposed recovery method shows the

best performance under the cross-view attack. That is because the adjacent views of the missing one provide more useful information for object recovery.

**Table 4:** Recovery performance comparison between different attack pattern under the 50% attack intensity.

| Attack Pattern | NDS | mAP |
|---|---|---|
| Random Attack | 0.3369 | 0.1518 |
| Consecutive Attack | 0.3264 | 0.1291 |
| Cross-View Attack | **0.3532** | **0.1819** |

*4.3.5 Temporal Resilience Analysis*

To further assess the temporal resilience of the proposed framework under sustained attack, we conduct a detailed analysis on a single representative scene. With a constant 50% attack intensity, this paper tracks the F1-Score on a frame-by-frame basis, as shown in Fig. 7. In Fig. 7, the black, red and blue lines represent the performance of the baseline method, the method under attack without recovery and the method with recovery separately. The baseline with the black line reflects the performance of the model in the environment without the attack, serving as an ideal benchmark for comparison. As the attack begins around timestamp 20, the red curve sharply declines and keeps at a very low F1 score. In contrast, the method with recovery with the blue curve, although it is still below the ideal baseline, consistently stays well above the attack curve throughout the sequence. Such a trend clearly demonstrates that the proposed framework delays complete failure when the autonomous driving system is under attack, leaving reaction time for the drivers to disengage the vehicle.

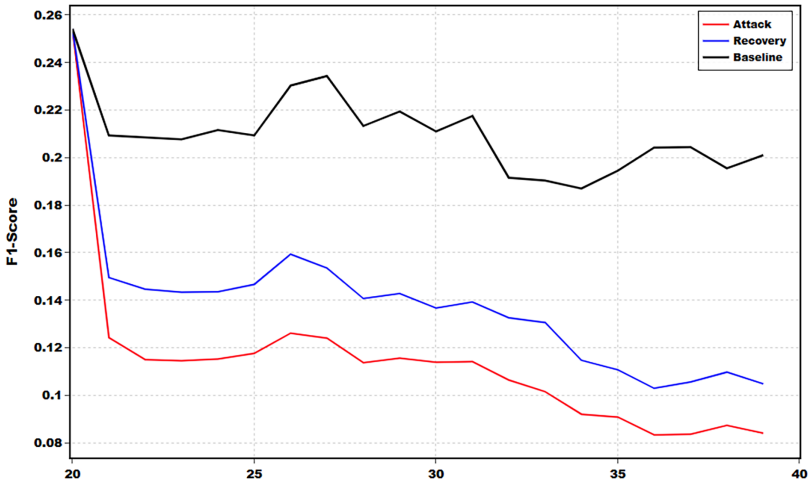



**Figure 7:** Temporal analysis of F1-Score within a single scene under 50% attack intensity, where the black, red and blue lines represent the performance of the baseline method, the method under attack without recovery and the method with recovery separately.

## 5 Conclusion and Future Work

In this work, we have designed a 3D object recovery framework for data tampering attacks in IVNs. TRB aims to learn the coordinate offsets of missing objects by modeling the complex and non-linear motion of the 3D objects along with the temporal dimension. To reduce the error accumulation, SRB is designed to provide a trajectory guidance for the movement of the missing objects. Experimental results on the nuScenes

benchmark demonstrate that the proposed framework effectively recovers the missing objects, especially under the severe attack scenario.

In the future, we will extend the framework to more complex and diverse attack patterns beyond the current simulation, such as progressive data corruption. In addition, we also plan to deploy the model on real-world autonomous driving systems to verify its effectiveness. Furthermore, we plan to explore the end-to-end integration of CRM with 3D detectors to further optimize recovery performance at the feature level.

**Acknowledgement:** Not applicable.

**Funding Statement:** This work was funded by the Program of Songshan Laboratory (241110210100); the National Natural Science Foundation of China (62301497); the Science and Technology Research Program of Henan (252102211024); and the Key Research and Development Program of Henan (231111212000).

**Author Contributions:** The authors confirm contribution to the paper as follows: Conceptualization, Gangtao Han and Song Wang; Methodology, Gangtao Han, Yurui Chen and Song Wang; Software, Yurui Chen; Validation, Yurui Chen; Formal analysis, Gangtao Han; Investigation, Gangtao Han and Yurui Chen; Resources, Song Wang; Data curation, Yurui Chen; Writing—original draft, Gangtao Han and Yurui Chen; Writing—review and editing, Song Wang, Lingling Li, Enqing Chen and Gaofeng Pan; Visualization, Yurui Chen; Supervision, Song Wang, Lingling Li, Enqing Chen and Gaofeng Pan; Project administration, Gangtao Han and Song Wang; Funding acquisition, Gangtao Han and Song Wang. All authors reviewed and approved the final version of the manuscript.

**Availability of Data and Materials:** The data that support the findings of this study include the publicly available nuScenes dataset, which can be accessed via its official website: https://www.nuscenes.org. This dataset was originally introduced in the work of Caesar et al.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Yen MH, Mishra N, Luo WJ, Lin CE. A novel proactive AI-based agents framework for an IoE-based smart things monitoring system with applications for smart vehicles. Comput Mater Contin. 2025;82(2):1839–55. doi:10.32604/cmc.2025.060903.
2. Hossain MN, Rahim MA, Rahman MM, Ramasamy D. Artificial intelligence revolutionising the automotive sector: a comprehensive review of current insights, challenges, and future scope. Comput Mater Contin. 2025;82(3):3643–92. doi:10.32604/cmc.2025.061749.
3. Sun C, Zhang R, Lu Y, Cui Y, Deng Z, Cao D, et al. Toward ensuring safety for autonomous driving perception: standardization progress, research advances, and perspectives. IEEE Trans Intell Transp Syst. 2024;25(5):3286–304. doi:10.1109/tits.2023.3321309.
4. Eraliev OMU, Lee KH, Shin DY, Lee CH. Sensing, perception, decision, planning and action of autonomous excavators. Autom Constr. 2022;141:104428. doi:10.1016/j.autcon.2022.104428.
5. Eze E, Eze J. Artificial intelligence support for 5G/6G-enabled Internet of Vehicles networks: an overview. ITU J Future Evol Technol. 2023;4(1):178–95. doi:10.52953/iezn8770.
6. Pipicelli M, Giallorenzo S, Liva G, Melis A, Sorniotti A. Architecture and potential of connected and autonomous vehicles. Vehicles. 2024;6(1):275–304. doi:10.3390/vehicles6010012.
7. Nanda A, Puthal D, Rodrigues JJPC, Pathan ASK. Internet of autonomous vehicles communications security: overview, issues, and directions. IEEE Wirel Commun. 2019;26(4):60–5. doi:10.1109/mwc.2019.1800503.
8. Tampuu A, Matiisen T, Semikin M, Nõmm S, Rõbens G. A survey of end-to-end driving: architectures and training methods. IEEE Trans Neural Netw Learn Syst. 2022;33(4):1364–84. doi:10.1109/tnnls.2020.3043505.

9. Limbasiya T, Ghosal A, Conti M. AutoSec: secure automotive data transmission scheme for in-vehicle networks. In: Proceedings of the 23rd International Conference on Distributed Computing and Networking; 2022 Jan 4–7; New Delhi, India. p. 208–16. doi:10.1145/3491003.3491024.
10. Lo Bello L, Patti G, Leonardi L. A perspective on Ethernet in automotive communications—current status and future trends. Appl Sci. 2023;13(3):1278. doi:10.3390/app13031278.
11. Bozdal M, Samie M, Aslam S, Jennions I. Evaluation of CAN bus security challenges. Sensors. 2020;20(8):2364. doi:10.3390/s20082364.
12. Fayyaz Khan O, Mubashir M, Iqbal J. Comprehensive review of CAN bus security: vulnerabilities, cryptographic and IDS approaches, and countermeasures. J Eng Technol Appl Phys. 2025;7(1):19–26. doi:10.33093/jetap.2025.7.1.4.
13. Gao X, Pan Y, Li J, Ji T, Yuan Y, Liu Z. Design of network security protection network architecture for intelligent networked vehicles. In: Proceedings of the 2024 3rd International Conference on Cyber Security, Artificial Intelligence and Digital Economy; 2024 Mar 1–3; Nanjing, China. p. 26–33. doi:10.1145/3672919.3672925.
14. Tripathi N, Hubballi N. Application layer denial-of-service attacks and defense mechanisms: a survey. ACM Comput Surv. 2022;54(4):1–33. doi:10.1145/3448291.
15. Dasgupta S, Rahman M, Islam M, Chowdhury M. A sensor fusion-based GNSS spoofing attack detection framework for autonomous vehicles. IEEE Trans Intell Transp Syst. 2022;23(12):23559–72. doi:10.1109/tits.2022.3197817.
16. Adly S, Salah A, El-Sayed A, El-Gazar M. Prevention of controller area network (CAN) attacks on electric autonomous vehicles. Appl Sci. 2023;13(16):9374. doi:10.3390/app13169374.
17. Duan X, Yan H, Tian D, Zhou J, Su J, Hao W. In-vehicle CAN bus tampering attacks detection for connected and autonomous vehicles using an improved isolation forest method. IEEE Trans Intell Transp Syst. 2023;24(2):2122–34. doi:10.1109/tits.2021.3128634.
18. Althunayyan M, Javed A, Rana OF. A survey of learning-based intrusion detection systems for in-vehicle network. arXiv:2505.11551. 2025. doi:10.48550/arXiv.2505.11551.
19. Rajapaksha S, Kalutarage H, Al-Kadri MO, Petrovski A, Madzudzo G, Cheah M. AI-based intrusion detection systems for in-vehicle networks: a survey. ACM Comput Surv. 2023;55(11):1–40. doi:10.1145/3570954.
20. Gu Q, Formby D, Ji S, Cam H, Beyah R. Fingerprinting for cyber-physical system security: device physics matters too. IEEE Secur Priv. 2018;16(5):49–59. doi:10.1109/msp.2018.3761722.
21. Lalouani W, Dang Y, Younis M. Mitigating voltage fingerprint spoofing attacks on the controller area network bus. Clust Comput. 2023;26(2):1447–60. doi:10.1007/s10586-022-03821-x.
22. Dong Q, Song Z, Shao H. Intrusion detection and defence for CAN bus through frequency anomaly analysis and arbitration mechanism. Electron Lett. 2024;60(3):e13112. doi:10.1049/ell2.13112.
23. Al-Jarrah OY, Maple C, Dianati M, Oxtoby D, Mouzakitis A. Intrusion detection systems for intra-vehicle networks: a review. IEEE Access. 2019;7:21266–89. doi:10.1109/access.2019.2894183.
24. Caesar H, Bankiti V, Lang AH, Vora S, Liong VE, Xu Q, et al. nuScenes: a multimodal dataset for autonomous driving. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2020 Jun 13–19; Seattle, WA, USA. p. 11618–28. doi:10.1109/cvpr42600.2020.01164.
25. Li T, Wang Z, Zou L, Chen B, Yu L. A dynamic encryption-decryption scheme for replay attack detection in cyber-physical systems. Automatica. 2023;151:110926. doi:10.1016/j.automatica.2023.110926.
26. Liu X, Jiang W, Li Z, Jin X, Ma Z, Li Q. FuzzAGG: a fuzzing-driven attack graph generation framework for industrial robot systems. Comput Secur. 2025;150:104223. doi:10.1016/j.cose.2024.104223.
27. Fuhrman S, Güngör O, Rosing T. CND-IDS: continual novelty detection for intrusion detection systems. arXiv:2502.14094. 2025. doi:10.48550/arXiv.2502.14094.
28. Wang K, Sun Z, Wang B, Fan Q, Li M, Zhang H. ATHENA: an in-vehicle CAN intrusion detection framework based on physical characteristics of vehicle systems. arXiv:2503.17067. 2025. doi:10.48550/arXiv.2503.17067.
29. Zhang M, Li J, Lai Y, Huan S, Shang W. A lightweight voltage-based ECU fingerprint intrusion detection system for in-vehicle CAN bus. IEEE Trans Veh Technol. 2025;74(10):15536–48. doi:10.1109/tvt.2025.3570961.

30. Young C, Olufowobi H, Bloom G, Zambreno J. Automotive intrusion detection based on constant CAN message frequencies across vehicle driving modes. In: Proceedings of the ACM Workshop on Automotive Cybersecurity; 2019 Mar 27; Richardson, TX, USA. p. 9–14. doi:10.1145/3309171.3309179.
31. Ahmad Khan J, Lim DW, Kim YS. A deep learning-based IDS for automotive theft detection for in-vehicle CAN bus. IEEE Access. 2023;11:112814–29. doi:10.1109/access.2023.3323891.
32. Shi J, Xie Z, Dong L, Jiang X, Jin X. IDS-DEC: a novel intrusion detection for CAN bus traffic based on deep embedded clustering. Veh Commun. 2024;49:100830. doi:10.1016/j.vehcom.2024.100830.
33. Neupane S, Ables J, Anderson W, Mittal S, Rahimi S, Banicescu I, et al. Explainable intrusion detection systems (X-IDS): a survey of current methods, challenges, and opportunities. IEEE Access. 2022;10:112392–415. doi:10.1109/access.2022.3216617.
34. Luo H, Zheng Y. Survey of research on image inpainting methods. J Comput Sci Explor. 2022;16:2193–218. doi:10.3778/j.issn.1673-9418.2204101.
35. Luo F, Wu X. Maximum a posteriori on a submanifold: a general image restoration method with GAN. In: Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN); 2020 Jul 19–24; Glasgow, UK. p. 1–7. doi:10.1109/ijcnn48605.2020.9207162.
36. Pang Y, Mao J, He L, Lin H, Qiang Z. An improved face image restoration method based on denoising diffusion probabilistic models. IEEE Access. 2024;12:3581–96. doi:10.1109/access.2024.3349423.
37. Wang L, Yang Q, Wang C, Wang W, Su Z. Coarse-to-fine mechanisms mitigate diffusion limitations on image restoration. Comput Vis Image Underst. 2024;248:104118. doi:10.1016/j.cviu.2024.104118.
38. Chen S, Ma Y, Qiao Y, Wang Y. M-BEV: masked BEV perception for robust autonomous driving. Proc AAAI Conf Artif Intell. 2024;38(2):1183–91. doi:10.1609/aaai.v38i2.27880.
39. Li X, Xie T, Liu D, Gao J, Dai K, Jiang Z, et al. Poly-MOT: a polyhedral framework for 3D multi-object tracking. In: Proceedings of the 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS); 2023 Oct 1–5; Detroit, MI, USA. p. 9391–8. doi:10.1109/iros55552.2023.10341778.
40. Karle P, Geisslinger M, Betz J, Lienkamp M. Scenario understanding and motion prediction for autonomous vehicles—Review and comparison. IEEE Trans Intell Transp Syst. 2022;23(10):16962–82. doi:10.1109/tits.2022.3156011.
41. Xie S, Kong L, Zhang W, Ren J, Pan L, Chen K, et al. RoboBEV: towards robust bird's-eye view perception under corruptions. arXiv:2304.06719. 2023. doi:10.48550/arXiv.2304.06719.
42. Liu Y, Yan J, Jia F, Li S, Gao A, Wang T, et al. PETRv2: a unified framework for 3D perception from multi-camera images. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 2023 Oct 1–6; Paris, France. p. 3239–49. doi:10.1109/iccv51070.2023.00302.
43. Xia Z, Lin Z, Wang X, Wang Y, Xing Y, Qi S, et al. HENet: hybrid encoding forEnd-to-end multi-task 3D perception fromMulti-view cameras. In: Computer Vision—ECCV 2024. Cham, Switzerland: Springer; 2025. p. 376–92. doi:10.1007/978-3-031-72973-7_22.
44. Unger D, Gosala N, Kumar VR, Borse S, Valada A, Yogamani SK. Multi-camera bird's eye view perception for autonomous driving. arXiv:2309.09080. 2023. doi:10.48550/arXiv.2309.09080.
45. Li Z, Wang W, Li H, Xie E, Sima C, Lu T, et al. BEVFormer: learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. arXiv:2203.17270. 2022. doi:10.48550/arXiv.2203.17270.
46. Loshchilov I, Hutter F. Decoupled weight decay regularization. arXiv:1711.05101. 2018. doi:10.48550/arXiv.1711.05101.
47. Alaba SY, Ball JE. Transformer-based optimized multimodal fusion for 3D object detection in autonomous driving. IEEE Access. 2024;12:50165–76. doi:10.1109/access.2024.3385439.
48. Song Y, Wang L. BiCo-fusion: bidirectional complementary LiDAR-camera fusion for semantic- and spatial-aware 3D object detection. IEEE Robot Autom Lett. 2025;10(2):1457–64. doi:10.1109/lra.2024.3518845.
49. Yin J, Shen J, Chen R, Li W, Yang R, Frossard P, et al. IS-fusion: instance-scene collaborative fusion for multimodal 3D object detection. In: Proceedings of the 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 2024 Jun 16–22; Seattle, WA, USA. p. 14905–15. doi:10.1109/cvpr52733.2024.01412.

50. Aliwa E, Rana O, Perera C, Burnap P. Cyberattacks and countermeasures for in-vehicle networks. ACM Comput Surv. 2022;54(1):1–37. doi:10.1145/3431233.
51. Guan T, Han Y, Kang N, Tang N, Chen X, Wang S. An overview of vehicular cybersecurity for intelligent connected vehicles. Sustainability. 2022;14(9):5211. doi:10.3390/su14095211.