Temporal Misalignment Attacks against Multimodal Perception in Autonomous Driving

Md Hasan Shahriar* , Md Mohaimin Al Barat*, Harshavardhan Sundar†, Ning Zhang‡,
Naren Ramakrishnan*, Y. Thomas Hou*, Wenjing Lou*
*Virginia Tech, Blacksburg, VA, USA
{hshahriar, barat, naren, thou, wjlou}@vt.edu
†Amazon.com, Inc., New York, NY, USA
hsundar427@gmail.com
‡Washington University in St. Louis, St. Louis, MO, USA
zhang.ning@wustl.edu

***Abstract*—Multimodal fusion (MMF) plays a critical role in the perception of autonomous driving, which primarily fuses camera and LiDAR streams for a comprehensive and efficient scene understanding. However, its strict reliance on precise temporal synchronization exposes it to new vulnerabilities. In this paper, we introduce DEJAVU, an attack that exploits the in-vehicular network to manipulate the integrity of time and create subtle temporal misalignments, severely degrading downstream MMF-based perception tasks. Our comprehensive attack analysis across different models and datasets reveals the sensors' task-specific imbalanced sensitivities: object detection is overly dependent on LiDAR inputs, while object tracking is highly reliant on the camera inputs. Consequently, with a single-frame LiDAR delay, an attacker can reduce the car detection mAP by up to 88.5%, while with a three-frame camera delay, multiple object tracking accuracy (MOTA) for car drops by 73%. We further demonstrated two attack scenarios using an automotive Ethernet testbed for hardware-in-the-loop validation and the Autoware stack for end-to-end AD simulation, demonstrating the feasibility of the DEJAVU attack and its severe impact, such as collisions and phantom braking. Our code and artifacts are publicly available at https://github.com/shahriar0651/DejaVu.**

***Index Terms*—multimodal sensor fusion, autonomous vehicles, temporal misalignment attacks**

I. INTRODUCTION

Autonomous driving (AD) is designed to navigate and interact with complex environments—a capability fundamentally reliant on a comprehensive understanding of its surroundings. The adoption of heterogeneous sensors, such as camera, LiDAR, and radar, that capture data from different modalities allows an accurate perception with enhanced accuracy and robustness [1]. Each individual modality has unique strengths; for example, cameras capture rich semantic details, LiDAR provides accurate depth measurements, and radar particularly excels in detecting speed, even in adverse weather conditions [2]. However, these sensors also face inherent limitations, such as a camera's sensitivity to lighting variations, LiDAR's lack of texture information, and radar's sparsity, which can compromise performance when used independently [3, 4, 5].

This work has been accepted for publication at the IEEE Conference on Secure and Trustworthy Machine Learning (SaTML 2026). The final version will be available on IEEE Xplore.

Multimodal fusion (MMF)—the process of integrating multiple unimodal sensor data into a single and comprehensive representation—compensates for such individual sensor weaknesses, ensures accurate and robust perception, and efficient downstream tasks in AD [6, 7]. MMF remains a fundamental research challenge, primarily due to the heterogeneity across sensing modalities, including differences in data formats and spatial [8, 9].

Another central aspect of this heterogeneity is *temporal*: in practice, multimodal sensors operate at inherently different sampling rates and are naturally asynchronous. Cameras typically capture frames at 30–60 Hz, LiDAR at 10–20 Hz, and radar at yet different rates. Each sensor samples the environment independently, so messages arrive at the fusion node at different times and with different temporal spacing. As a result, a camera frame captured at a given instant often has no LiDAR or radar measurement captured at exactly the same moment. Before fusion can take place, the system must therefore decide *which* observations from each modality correspond to the same physical instant—a problem known as *temporal alignment*. Getting this pairing right is critical. Fusing observations from different times, known as misalignments, yields semantically inconsistent representations, degrading perception accuracy and leading to object misdetection, localization errors, and scene misinterpretation. These inaccuracies propagate downstream to control, maneuver planning, and safety interventions, ultimately compromising vehicle reliability and safety [10]. However, the robustness of temporal alignment has received comparatively limited attention [11], despite being a fundamental enabler of MMF in AD. Moreover, it remains a complex problem due to the following challenges.

Clock synchronization. To ensure temporally aligned perception in AD, sensor data from all the modalities must be timestamped relative to a common global clock with minimal drift. In practice, the electronic controller units (ECUs) hosting the sensors must be tightly synchronized—often at sub-microsecond precision required by AUTOSAR [12])—to enable accurate fusion and downstream reasoning. To achieve that, automotive Ethernet (AE) equipped with Time-Sensitive Networking (TSN) has become the de facto backbone of

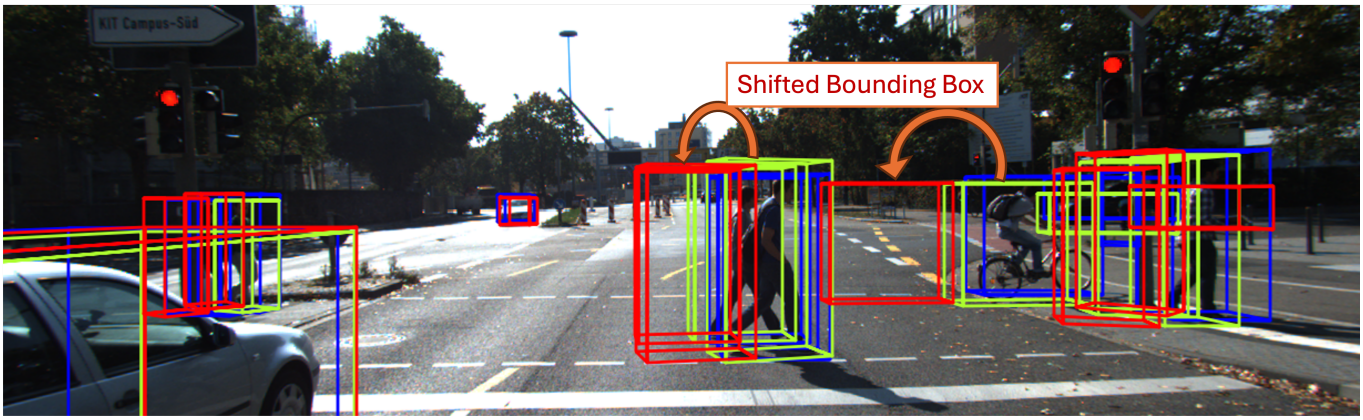


Fig. 1. Visualization of the impact of the DEJAVU attack on a camera–LiDAR fusion-based perception model. Ground-truth boxes are shown in blue, benign predictions in green, and predictions under DEJAVU in red. MMF-based fused predictions are overlaid on the current camera frame. While benign predictions closely track the ground truth, delaying the LiDAR stream by five frames under DEJAVU induces cross-sensor temporal misalignment, shifting predicted bounding boxes toward objects' past locations and causing some objects to be missed. This behavior suggests an overreliance on LiDAR inputs relative to the camera stream. A corresponding LiDAR-plane visualization is provided in the Appendix (Fig. 17).

modern in-vehicle networks. TSN leverages the generalized Precision Time Protocol (gPTP, IEEE 802.1AS) to maintain global clock synchronization among distributed ECUs [13, 14]. Consequently, a core security assumption is: A_1 ***The gPTP-based synchronization infrastructure is secure and maintains consistent global time across all ECUs.*** However, despite its widespread adoption in AE, gPTP was primarily designed to provide deterministic time synchronization—not to operate under adversarial conditions. Hence, gPTP lacks built-in security mechanisms, particularly cryptographic authentication, leaving it vulnerable to attacks such as grandmaster spoofing, delay injection, replay, and false time advertisement [15, 16, 17]. Such attacks can compromise the temporal alignment of sensor fusion, undermining the integrity of MMF-based perception, without directly manipulating the sensor ECU itself.

Timestamp Integrity. Even when sensors are nominally synchronized, they often capture data at slightly different times due to variations in sampling rates caused by inherent hardware limitations [11]. Consequently, the timestamps attached to sensor messages become critical in multimodal fusion pipelines, as they serve as the primary reference for aligning asynchronous data streams by identifying the most temporally proximal pairs from buffered queues. This reliance gives rise to a fundamental security assumption: A_2 ***Sensor ECUs are trusted entities that always provide accurate timestamps.*** However, this assumption can be violated. In vehicular systems, sensor ECUs may be compromised through remote exploits of unpatched software vulnerabilities [18, 19, 20], insecure Over-The-Air (OTA) updates [21, 22], or direct physical access via the OBD-II port [18, 23, 24]. Once compromised, an attacker-controlled ECU can inject messages containing legitimate sensor data but with forged timestamps. Since the fusion ECU uses timestamps for temporal alignment, this can lead to selecting data pairs that appear temporally aligned but are semantically misaligned, thereby degrading the integrity of the sensor fusion process.

Middleware Integrity. Sensor data exchange in production-grade AD (such as Autoware[1]) is often facilitated by robotic middleware frameworks, such as robot operating system (ROS[2]), where the data distribution service (DDS) serves as the underlying communication backbone. DDS enables a real-time and scalable publish-subscribe communication model for data sharing among distributed ECUs, making it a widely adopted solution for managing the high-bandwidth, low-latency communication demands of multimodal perception and control pipelines. Consequently, a third foundational assumption arises: A_3 ***The ROS-based DDS infrastructure is secure and ensures the integrity, authenticity, and freshness of shared data.*** However, the design of ROS prioritizes performance and scalability over robust security guarantees. Since ROS does not enforce strong authentication, encryption, or source verification by default, ROS is vulnerable to a wide range of network-level and application-level attacks, including message spoofing, replay, and impersonation attacks [25]. As a result, an attacker with access to the ROS communication graph, for example, can impersonate legitimate nodes (e.g., a LiDAR publisher) and can publish fabricated sensor messages with targeted timestamps (i.e., replay attacks), which can be considered for fusion and mislead downstream perception tasks.

Moreover, prior work has demonstrated that existing MMF-based perception systems in AD lack *temporal robustness*—the ability to maintain reliable outputs in the presence of temporal inconsistencies—making them vulnerable even to benign, non-malicious misalignments [11, 26, 27]. In particular, perception pipelines have been shown to degrade significantly under small delays in just one modality. Existing studies, however, are limited in both scope and threat modeling: they primarily focus on random, system-induced delays and evaluate only a narrow subset of perception tasks—most commonly 3D object detection, without realizing them on any end-to-end

[1]https://github.com/autowarefoundation/autoware
[2]https://github.com/ros2/ros2

AD software stacks. To investigate the true fragility of MMF-based perception under adversarial conditions, we present a comprehensive study of temporal misalignment attacks, which we term DEJAVU. These attacks are realized by violating one or more of the core trust assumptions. Unlike prior work [28], we target both 3D object detection and multi-object tracking (MOT), and evaluate the effects of different delay distributions across multiple perception models and datasets. Fig. 1 illustrates how temporal delays due to DEJAVU attacks in one of the modalities can degrade the performance of the MVXNet model [29]—a MMF-based 3D object detection model, potentially leading to unsafe driving conditions in AD. In summary, we make the following key contributions:

- We propose DEJAVU, a temporal misalignment attack against MMF in AD that exploits vulnerabilities of in-vehicle networks and the fragility of multimodal fusion by selectively delaying sensor streams to disrupt perception in safety-critical tasks.
- We conduct a comprehensive empirical evaluation of DEJAVU on state-of-the-art 3D object detection models (MVXNet [29], BEVFusion [30]) using the KITTI [31] and nuScenes [32] datasets, and a multi-object tracking model (MMF-JDT [33]) under various misalignment scenarios using the KITTI [31] dataset. Our findings reveal distinct modality-specific vulnerabilities: object detectors are highly sensitive to LiDAR delays, while the tracking model is significantly impacted by camera timing disruptions. A single-frame LiDAR delay reduces 3D detection mAP by up to 88.5%, and a three-frame camera delay drops multiple object tracking accuracy (MOTA) by 73%.
- To further validate our findings in a realistic autonomous driving setting, we built an automotive Ethernet testbed that models the sensor data acquisition and fusion pipeline. Using this platform, we implemented the DEJAVU attack by violating both A_1 and A_2, demonstrating its feasibility in a hardware-in-the-loop environment. Additionally, to assess the end-to-end consequences beyond perception—specifically on planning and control—we integrated DEJAVU into Autoware, a production-grade, full-stack autonomous driving simulator, by breaking A_3. In both environments, our experiments show that DEJAVU is highly practical and can result in severe safety violations, including direct collisions and phantom braking events.

The remainder of the paper is organized as follows. The remainder of the paper is organized as follows. Section II reviews related work. Section III formalizes the AD system model. Section IV presents the threat model and introduces the DEJAVU attack. Section V describes the realization of the attack on an HIL testbed. Section VI details the datasets and evaluation settings. Section VII reports the evaluation results. Section VIII discusses the findings and potential defenses. Section IX concludes the paper.

II. RELATED WORK

The fundamental research direction has been in the direction of spoofing sensors from a single modality, such as LiDAR [34, 35, 36] and camera [37] through different means of physical perturbation. Moreover, advanced attacks have demonstrated to even compromise multiple modalities together [38, 39, 40]. These sensor spoofing attacks demonstrate that simply having multiple sensors is not sufficient. With carefully constructed inputs, an adversary can simultaneously mislead camera and LiDAR sensors, defeating the very redundancy meant to ensure safety.

Unlike sensor spoofing attacks, time delay attacks have not been extensively studied within AD. They have created significant attention in other cyber-physical systems (CPS) domains, such as power systems [41, 42], wireless networks [43], unmanned aerial vehicles (UAVs) [15, 44, 45], and time-sensitive networks [46]. Software timing interference is also exploited to cause system destabilization in CPS [47]. Moreover, multimodal temporal misalignment in sensor fusion has been shown to degrade the accuracy of simultaneous localization and mapping (SLAM) [48], which was limited only to the fusion between IMU and camera data. Contrary to the existing works, we comprehensively study the impact of a temporal misalignment attack on task-agnostic MMF-based perception.

In the realm of AD security, various defense mechanisms have been proposed to counteract sensor spoofing [49] and multimodal fusion attacks. These defenses can be broadly categorized into spatiotemporal consistency checks, specification-aware recovery strategies, and hardware-based techniques [50]. PercepGuard [51] detects misclassification attacks by enforcing consistency between object tracks and class labels, but it does not examine the temporal validity of sensor readings and thus cannot detect replayed LiDAR scans whose semantic trajectories remain plausible. Connecting the Dots [52] employs class-specific autoencoders to uncover context violations introduced by adversarial perturbations, yet time-shifted data aligns perfectly with learned scene co-occurrence statistics and evades its checks. PhyScout [53] formalizes cross-modal conflict detection to identify gross spoofing, but subtle timestamp manipulations within the synchronizer's tolerance window introduce no overt spatial or modality discrepancies. These approaches overlook *timestamp integrity* and data freshness as a security property.

III. AUTONOMOUS DRIVING SYSTEM MODEL

This section provides a formal end-to-end model of AD systems, covering data collection, temporal alignment, and MMF-based perception.

A. Data Collection and Temporal References

Modern AD systems rely on heterogeneous sensor arrays to perceive the environment. We focus on camera (S_C) and LiDAR (S_L) as representative modalities, though the model extends to additional sensors (e.g., radar, IMU). Each sensor $S \in \{S_C, S_L\}$ transmits measurements over in-vehicle networks to a fusion node and produces a sequence of messages $m_S^{(i)} = \left(x_S^{(i)},\ t_S^{(i)}\right)$ indexed by $i \in \mathbb{N}$. A critical modeling detail is the distinction between two time references

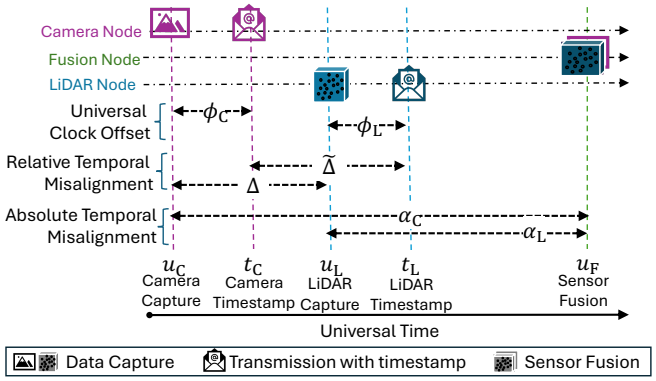


Fig. 2. Illustration of temporal events in MMF-based perception for autonomous driving systems. Under benign operating conditions, the UCO (ϕ_S), the reported and actual RTMs ($\tilde{\Delta}$ and Δ), and the S-ATM values (α_S) are all approximately negligible, up to the system's end-to-end pipeline latency.

associated with each measurement. The *universal capture time* $u_S^{(i)}$ denotes the true physical time when a data sample $x_S^{(i)}$ is captured in the world. As an estimation of $u_S^{(i)}$, the sensor attaches a *timestamp* $t_S^{(i)}$ using its local clock to the message header. The time reference $t_S^{(i)}$ is used by the fusion node for sorting, buffering, and temporal alignment of the message queue. Figure 2 illustrates such temporal events at different sensor nodes. For analysis purposes, we quantify the discrepancy between these two time references by the *universal clock offset* (UCO) of sensor S:

$$\phi_S^{(i)} = \left| t_S^{(i)} - u_S^{(i)} \right|. \tag{1}$$

In practice, sensors operate under nominal conditions where clock synchronization and minimal processing delays keep the universal clock offset small (i.e., $\phi_S^{(i)} \approx 0$). We formalize benign operation via the following assumption.

Assumption 1 (Timestamp fidelity)**.** *For each sensor message, the local timestamp approximates the universal capture time, i.e., $t_S^{(i)} \approx u_S^{(i)}$ (equivalently, $\phi_S^{(i)} \approx 0$).*

B. Temporal Alignment

Due to asynchronous sampling and transmission, temporal alignment is a critical step for MMF. Each modality maintains a finite FIFO buffer $\mathcal{Q}_S$ containing the most recent N messages, ordered by $t_S^{(\cdot)}$ (the only time reference available to the system). The fusion node employs a *temporal alignment* mechanism to identify temporally proximal cross-modal pairs that correspond to approximately the same physical instant.

A widely used policy is *approximate-time synchronization*, where the fusion node pairs messages from different modalities by comparing their header timestamps and accepting pairs whose timestamp difference is within a tolerance τ (the *slop*). This mechanism is implemented in production AD stacks such as ROS 2's `ApproximateTimeSynchronizer`[3] and similar middleware. Concretely, for each unpaired camera message $m_C^{(i)} \in \mathcal{Q}_C$, the aligner finds the LiDAR index:

$$j^\star(i) = \arg \min_{k:\, m_L^{(k)} \in \mathcal{Q}_L} \left| t_C^{(i)} - t_L^{(k)} \right|. \tag{2}$$

If $\left| t_C^{(i)} - t_L^{(j^\star(i))} \right| \le \tau$, the fusion node selects the pair $(i, j^\star(i))$ for fusion and remove both messages from their buffers. Symmetrically apply the rule for unpaired LiDAR messages $m_L^{(j)}$. For an selected pair (i, j), we quantify temporal alignment in two complementary ways:

- Reported relative temporal misalignment (R-RTM): $\tilde{\Delta}^{(i,j)} = |t_C^{(i)} - t_L^{(j)}|$, which must satisfy $\tilde{\Delta}^{(i,j)} \le \tau$ for the aligner to accept the pair.
- Semantic relative temporal misalignment (S-RTM): $\Delta^{(i,j)} = |u_C^{(i)} - u_L^{(j)}|$, which captures the true capture-time separation between the paired observations.

Under benign synchronization and Assumption 1, the aligned pairs satisfy $|t_C^{(i)} - t_L^{(j)}| \ll \tau$ and holds ($t_S^{(i)} \approx u_S^{(i)}$), this implies that the paired payloads are also close in capture time, so fusion inputs are semantically aligned and the reported and semantic RTM coincide, i.e., $\tilde{\Delta}^{(i,j)} = \Delta^{(i,j)}$. This alignment policy bases pairing exclusively on reported timestamps $t_S^{(\cdot)}$ and queue state, *not* on message content. Thus, while the above-mentioned assumptions are effective under benign conditions, it creates a vulnerability: an adversary who can manipulate timestamps can force the alignment mechanism to pair semantically inconsistent observations.

C. Multimodal Fusion-based Perception

As each heterogeneous sensor generates data samples with unique formats and dimensions, modality-specific unimodal feature encoders E_S convert raw data $x_S^{(i)}$ to intermediate representations $f_S^{(i)} = E_S(x_S^{(i)})$. These encoders map raw data to a common feature format, enabling various fusion techniques such as concatenation, attention mechanisms, or tensor-based fusion [54]. For each aligned pair (i, j) selected by the alignment mechanism, the fusion module computes a fused representation $z^{(i,j)}$ using a fusion operator $\mathcal{F}_\theta$:

$$z^{(i,j)} = \mathcal{F}_\theta\big(f_C^{(i)}, f_L^{(j)}\big).$$

We associate each selected pair (i, j) with a *universal fusion time* $u_F^{(i,j)}$, when the fusion actually happens, and the system perceives the environment. For a fused pair (i, j), we define the semantic absolute temporal misalignment (S-ATM) $\alpha_S^{(i,j)}$ as the age of each data sample relative to the fusion event:

$$\alpha_C^{(i,j)} = |u_F^{(i,j)} - u_C^{(i)}|, \qquad \alpha_L^{(i,j)} = |u_F^{(i,j)} - u_L^{(j)}|. \tag{3}$$

[3] `TimeSynchronizer` and `ApproximateTimeSynchronizer` are commonly used message filtering utilities in ROS 2 that align multiple sensor message streams based on their timestamps. While `TimeSynchronizer` performs strict timestamp matching, `ApproximateTimeSynchronizer` allows messages with slight temporal differences—within a specified tolerance window—to be synchronized.

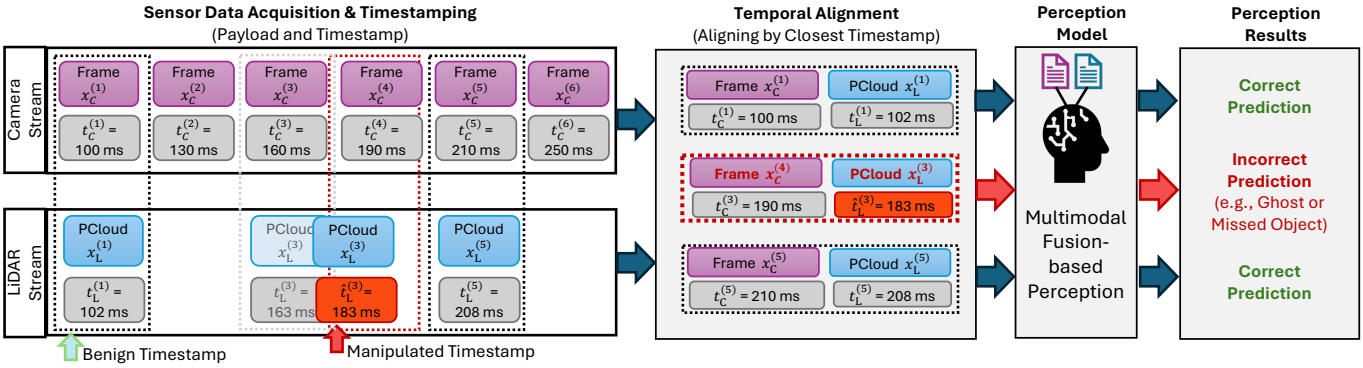


Fig. 3. **Overview of the DejaVu Attack.** Camera and LiDAR sensors produce asynchronous data streams $\{x_C^{(i)}\}$ and $\{x_L^{(j)}\}$. The adversary maliciously modifies a LiDAR packet by manipulating its timestamp $t_L^{(3)}$ (i.e., 183 ms instead of 163 ms) while keeping the data payload benign. During approximate temporal synchronization, the corresponding LiDAR point cloud is incorrectly paired with a temporally proximate camera frame (e.g., $(x_C^{(4)}, x_L^{(3)})$). This temporally misaligned pair is then fused by the perception model, leading to degraded predictions such as ghost objects or missed detections.

With a reliable communication infrastructure, this leads to the following assumptions:

Assumption 2 (Fusion-time Coherence)**.** *For an selected pair (i, j), the universal fusion time $u_F^{(i,j)}$ is close to the capture time of each sensor, so that $u_F^{(i,j)} \approx u_C^{(i)} \approx u_L^{(j)}$ and $\alpha_C^{(i,j)} \approx \alpha_L^{(i,j)} \approx 0$.*

The corresponding perception head $\mathcal{H}_\theta$ then produces the final prediction: $\mathcal{Y}^{(i,j)} = \mathcal{H}_\theta\big(z^{(i,j)}\big)$. The perception output $\mathcal{Y}^{(i,j)}$ typically includes object detections, semantic segmentation, or tracking states, and is then consumed by downstream planning and control modules.

IV. Temporal Misalignment Attack : DejaVu

In this section, we formalize the DejaVu attack, where an adversary maliciously introduces temporal misalignment by manipulating the local timestamps used for temporal alignment. The attack exploits the timestamp-based pairing mechanism, which can keep pairs aligned in the reported timestamp space while making them semantically misaligned in universal capture time, degrading fusion quality. Figure 3 illustrates an example of DejaVu attack.

A. Threat Model and Attack Surface

1) Adversary Objective and Scope: The adversary's goal is to coerce the temporal alignment mechanism to emit *semantically inconsistent* (misaligned) pairs so that the fused embedding $z^{(i,j)}$ combines observations from different physical times, degrading downstream performance (e.g., missed or spurious detections, localization errors). The adversary is *timing-only*: they do not modify payloads $x_S^{(i)}$ or model parameters, ensuring the attack remains stealthy against content-based detectors. The attack breaks benign operation via two linked steps. First, the adversary injects a universal clock offset (UCO) so that Assumption 1 ($t_S^{(i)} \approx u_S^{(i)}$) no longer holds and $\phi_S^{(i)}$ becomes large. Second, because the aligner pairs messages using timestamps only, the poisoned queues can yield pairs that satisfy the reported tolerance window ($\tilde{\Delta}^{(i,j)} \leq \tau$) yet violate Assumption 2 by producing large $\Delta^{(i,j)}$ and/or large $\alpha_S^{(i,j)}$.

2) Attacker Capabilities: The adversary can gain access to the in-vehicle network through one or more realistic entry points [55]. *Physical Access:* The attacker connects to the vehicle's OBD-II port or directly to Ethernet/CAN interfaces, either through malicious maintenance personnel or via physical compromise [18]. *Remote Exploitation:* The attacker exploits vulnerabilities in externally exposed interfaces, such as telematics units, infotainment systems, or over-the-air (OTA) update mechanisms [18, 20, 22]. *Supply Chain Attacks:* The attacker implants malicious code or hardware during manufacturing, allowing persistent access post-deployment [56].

Once access is established, the attacker can monitor, intercept, and inject messages on the in-vehicle AE backbone and associated sub-networks. We assume the attacker operates under any of the following three distinct capabilities:

$\mathcal{C}_1$ *Disruption of Clock Synchronization.* With this capability, the adversary targets the clock synchronization mechanism in AE—specifically, the gPTP (IEEE 802.1AS). Rather than altering timestamps directly, the attacker compromises the synchronization process itself, thereby inducing *actual* temporal misalignment between sensor streams. This can be accomplished by impersonating the grandmaster clock or tampering with synchronization messages via selective delay, replay, or man-in-the-middle attacks [57, 58, 59]. Although individual timestamps are not directly manipulated, they no longer correspond to a consistent global time due to induced drift, impairing the performance of perception tasks.

$\mathcal{C}_2$ *Manipulation of Timestamp Integrity.* In this scenario, the attacker is much powerful and has direct control over the ECU. Under this attack, the attacker preserves the actual flow of data but alters the timestamps embedded in transmitted packets [55]. This results in *seemingly* temporally misaligned messages, even though the underlying data remain *actually* temporally aligned. Consequently, the fusion node introduces semantic misalignment during the alignment process. This capability is practically plausible because many ROS2 deploy-

ments are not configured with authentication-by-default [25], and ECUs frequently run third-party or legacy software that enlarges the attack surface [18, 19, 20, 21, 22].

C_3 *Impersonation of a Legitimate Node in ROS2.* In this scenario, the attacker is a participant in the ROS2 network. As the default ROS2 implementation lacks built-in security mechanisms—a configuration commonly adopted in industry-grade AD software stacks, including Autoware—the attacker can instantiate a malicious node that impersonates a legitimate sensor publisher. By subscribing to target sensing topics[4], the attacker gains access to the data stream and records historical sensor messages. The malicious node then re-publishes previously captured messages with updated, genuine-looking timestamps, potentially while the original publisher is still active. By repeatedly injecting such delayed-but-valid messages, the attacker can pollute the input queue of time-based synchronizers, increasing the likelihood that a forged message is selected during the fusion process.

B. Attack Formalization

For any message $m_S^{(k)} = (x_S^{(k)}, t_S^{(k)})$, the adversary injects a timestamp perturbation $\delta_S^{(k)}$ so that, while the payload $x_S^{(k)}$ remains unchanged, the local timestamp used by the aligner becomes inconsistent with the universal capture time:

$$\hat{t}_S^{(k)} = t_S^{(k)} + \delta_S^{(k)}, \qquad \phi_S^{(k)} = \left|\hat{t}_S^{(k)} - u_S^{(k)}\right| \gg 0.$$

For concreteness, consider a unimodal DEJAVU where only the camera stream is compromised, i.e., $\hat{t}_C^{(i)} \neq t_C^{(i)}$ while $\hat{t}_L^{(j)} = t_L^{(j)}$. The temporal alignment mechanism then computes pairings using the (potentially compromised) timestamps:

$$\hat{j}^\star(i) = \arg \min_{k:\, m_L^{(k)} \in \mathcal{Q}_L} |\hat{t}_C^{(i)} - t_L^{(k)}|,$$

and emits the pair $(i, \hat{j}^\star(i))$ if $\left|\hat{t}_C^{(i)} - t_L^{(\hat{j}^\star(i))}\right| \leq \tau$. A stealthy adversary ensures the reported RTM stays within tolerance (i.e., $\tilde{\Delta}^{(i,\hat{j}^\star(i))} \leq \tau$), so each emitted pair appears valid to the aligner. Meanwhile, by increasing the UCO and poisoning the timestamp-ordered queues, the attacker can inflate the actual S-RTM $\Delta^{(i,\hat{j}^\star(i))}$ and/or the S-ATM values $\alpha_S^{(i,\hat{j}^\star(i))}$, causing the fused representation to combine observations from distinct physical instants. These semantically inconsistent fused inputs yield corrupted perception outputs $\hat{\mathcal{Y}}^{(i,\hat{j}^\star(i))} \neq \mathcal{Y}^{(i,j)}$ (e.g., missed detections, false positives, localization drift), which can propagate to planning and control. We quantify these effects empirically in Section VI and Section VII.

C. Attack Strategies and Delay Distributions

The adversary can craft different attack strategies by controlling the injected timestamp perturbation schedule $\delta_S^{(k)}$ (and hence the induced $\phi_S^{(k)}$), which can vary over time depending on the attacker's intent. Based on the attacker's capabilities, the adversary can compromise either a single modality (unimodal DEJAVU, Uni-DEJAVU) or multiple modalities simultaneously (multimodal DEJAVU, Mul-DEJAVU). Moreover, based on the distribution of the delay $\delta_S^{(k)}$, we consider two representative types of attack as summarized in Table I and described as follows:

TABLE I
TWO TYPES OF DEJAVU ATTACK STRATEGY

Attack Name	Attack Type	Timestamp Perturbation $\delta_S^{(k)}$
Constant Delay	Constant	$\delta_S^{(k)} = \Delta_{\text{lag}}$ (constant offset)
Random Delay	Random	$\delta_S^{(k)} \sim \text{Uniform}(-\Delta_{\max}, \Delta_{\max})$

1) Constant Delay Attack: This strategy applies a fixed timestamp offset $\delta_S^{(k)} = \Delta_{\text{lag}}$ to all messages from a target sensor within a time window. The attacker can achieve this through capabilities C_1–C_3, such as creating clock desynchronization, tampering with timestamps, or replaying messages with forged timestamps. **Impact:** The alignment mechanism pairs messages with a consistent temporal lag, causing the fusion model to receive temporally consistent but shifted data. This leads to delayed (misaligned) perception, manifesting as missing or shifted bounding boxes in object detection, where real-time perception is essential.

2) Random Delay Attack: In this strategy, each message experiences a different timestamp perturbation, randomly sampled from $\delta_S^{(k)} \sim \text{Uniform}(-\Delta_{\max}, \Delta_{\max})$. This strategy not only disrupts real-time requirements but also disrupts the temporal sequence of sensor data, which is crucial for object tracking. **Impact:** The fusion system struggles to maintain proper ordering of sensor inputs, leading to degraded perception accuracy. This causes erratic behavior in time-sensitive sequential applications, particularly for object tracking and autonomous navigation.

V. REALIZATION OF DEJAVU

This section realizes DEJAVU on a hardware-in-the-loop (HIL) Automotive Ethernet testbed and empirically characterizes the induced temporal misalignment on this HIL testbed.

A. Automotive Ethernet HIL Testbed

1) Testbed description: We evaluate DEJAVU on a hardware-in-the-loop Automotive Ethernet testbed that mirrors the system model of Section III and serves as a proxy for an in-vehicle AD perception network. As illustrated in Fig. 4, the testbed comprises four Raspberry Pi nodes: a camera node, a LiDAR node, a fusion node, and a fourth node that is within the network but not part of the MMF pipeline. For reproducibility and controlled conditions, the camera and LiDAR nodes replay the *KITTI Tracking Dataset* instead of live sensors. Each node is equipped with a RealTime TSN HAT and a media converter; the nodes are interconnected via an AE switch to provide TSN and AE connectivity.

ROS 2 over DDS is used for message distribution. The camera and LiDAR nodes publish ROS 2 messages (images and point clouds) at a fixed rate (default $\approx$ 10 Hz). Each message carries (i) the sensor payload and (ii) a header

[4]Representative topic names from Autoware are `/sensing/camera/traffic_light/image_raw`, `/sensing/lidar/top/pointcloud_raw`, etc.

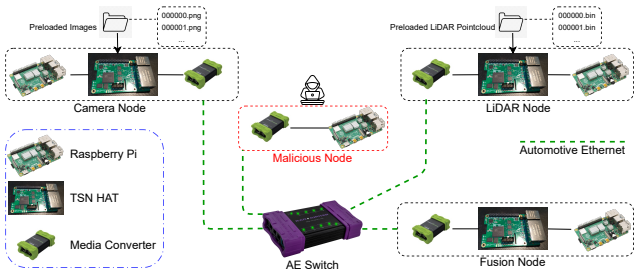
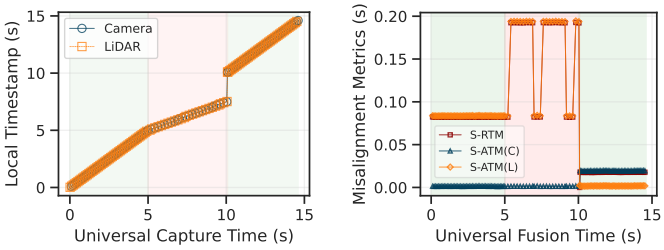


Fig. 4. Schematic diagram of hardware-in-the-loop testbed.



(a) Local timestamps under attack (b) Temporal misalignment metrics

Fig. 5. Impact of the PTP-based DEJAVU attack on timestamped sensor streams and temporal alignment in the HIL testbed. Following an initial benign phase, the attack is launched at t=5,s, during which the adversary repeatedly steps the GM clock backward (δ_S=0.20,s every 0.22,s). Panel (a) illustrates the resulting distortion in the camera and LiDAR local (header) timestamps. Panel (b) reports the corresponding alignment metrics for selected sensor pairs: while remaining within the acceptable slope threshold (< 0.10) under benign conditions, the S-RTM and S-ATM values exhibit pronounced spikes during the attack, indicating semantically misaligned fusion inputs.

timestamp set from the node's wall clock at publish time. All nodes run in a single PTP domain and are synchronized with `ptpd`. The fusion node subscribes to both topics and runs ROS 2's `ApproximateTimeSynchronizer`: it maintains bounded per-topic queues and emits a camera–LiDAR tuple when the header timestamps of the two messages fall within a configurable tolerance (the slop $\tau = 0.10\,\text{s}$). The fusion node then runs MMF-based perception on each emitted pair.

This setup instantiates the pipeline of Section III: the header timestamp is the local timestamp $t_S^{(i)}$, and the universal capture time at which the payload was captured (or replayed) is $u_S^{(i)}$. Under benign PTP synchronization, Assumption 1 holds, the UCO $\phi_S^{(i)}$ stays near zero, and the synchronizer's reported R-RTM $\tilde{\Delta}^{(i,j)}$ is a faithful proxy for the actual S-RTM $\Delta^{(i,j)}$. In the remainder of this section, we realize DEJAVU under different attacker capabilities and empirically characterize the induced temporal misalignment (UCO, R-RTM, S-RTM, and S-ATM) on this HIL testbed.

We realize the DEJAVU attack under capability $\mathcal{C}_1$ (disruption of clock synchronization) as described below. An instantiation under capability $\mathcal{C}_2$ (manipulation of timestamp integrity) is detailed in the Appendix B.

2) PTP-based DEJAVU *Attack:* We instantiate the threat model of Section IV-A with capability $\mathcal{C}_1$: the attacker does not modify message payloads or timestamps directly but compromises the *time-distribution plane*—the PTP-based clock synchronization used by all nodes. As shown in Fig. 4, we introduce a fourth, malicious node (Raspberry Pi) into the same Ethernet/PTP domain alongside the camera, LiDAR, and fusion nodes. The malicious node does not publish sensor data; it participates only in PTP. By exploiting the Best Master Clock Algorithm (BMCA), the attacker advertises a superior clock quality, is elected as Grandmaster (GM), and thus dictates the global time reference to which all legitimate nodes (and hence their local timestamps $t_S^{(i)}$) are bound. This takeover is protocol-compliant and transparent to the fusion stack.

a) Attack Mechanism: After assuming the GM role, the attacker periodically forces the GM's advertised time to step *backward* by a controlled amount δ_S (e.g., from t_{attack} to $t_{\text{attack}} - \delta_S$), then lets time advance again naturally. In our ROS 2 pipeline, sensors stamp each message with their local wall clock at publish time; that value is exactly the *local timestamp* $t_S^{(i)}$ used by the fusion node (Section III). A backward GM step propagates to slave nodes via PTP, so their system clocks—and thus the timestamps they attach—jump backward. Because ROS 2 disallows publishing messages with timestamps in the past relative to the last published time, sensor nodes effectively stop publishing for the real-time duration δ_S (a transient denial-of-service). When clocks later advance again and "catch up" to real time, sensors resume publishing with local timestamps that have moved forward. Critically, the *universal capture times* $u_S^{(i)}$ of newly produced messages are now at least δ_S seconds ahead of the last pre-step message, yet the *local timestamps* $t_S^{(i)}$ can lie in a narrow window that overlaps earlier timestamps. Thus, measurements from *distinct physical moments* (separated in universal time by $\geq \delta_S$) are mapped into a *narrow* local-timestamp window—*temporal compression* on the reported timeline. The UCO $\phi_S^{(i)}$ becomes large, violating Assumption 1. Because camera and LiDAR nodes follow their own PTP servo and scheduler dynamics, they recover at different rates and instants; the aligner can therefore form pairs (i, j) whose reported R-RTM $\tilde{\Delta}^{(i,j)}$ remains within the slop τ, while the actual S-RTM $\Delta^{(i,j)}$ and the S-ATM $\alpha_C^{(i,j)}$, $\alpha_L^{(i,j)}$ are large. The attacker can tune δ_S and the step cadence to trade off transient DoS duration against the magnitude of induced misalignment and stealth.

b) Attack Impact: We report results for a benign phase followed by an attack phase starting at $t = 5\,\text{s}$ and lasting 5 s, with the GM clock stepped backward by $\delta_S = 0.20\,\text{s}$ every 0.22 s. Figure 5a shows the effect on message transmission and on the local timestamps (header timestamps) reported by the camera and LiDAR nodes. Figure 5b shows the effect on temporal alignment at the fusion node: while the S-RTM and S-ATM values remained within the slop $\tau = 0.10$ under the benign period, their value spiked under the attack period, indicating that the aligner is emitting pairs that are semantically misaligned and/or stale. Fusing such pairs degrades perception; we quantify this in the following section.

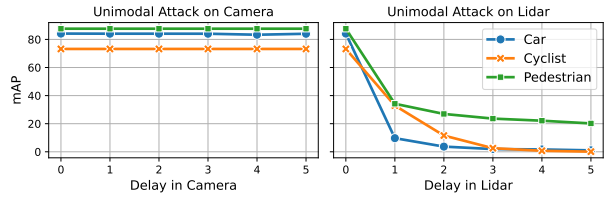


Fig. 6. Uni-DEJAVU attack impacts on 3D object detection performance of *MVXNet* on KITTI dataset for different object classes.

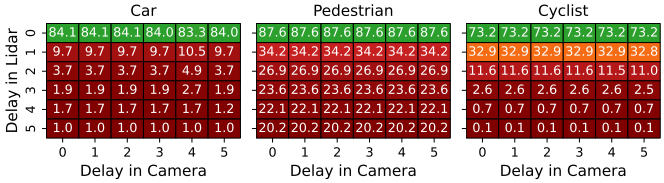


Fig. 7. Mul-DEJAVU attack impacts on 3D object detection performance of *MVXNet* on KITTI dataset for different object classes.

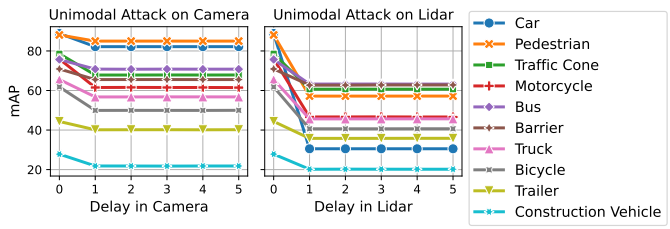


Fig. 8. Uni-DEJAVU attack impacts on 3D object detection performance of *BEVFusion* on nuScenes dataset for different object classes.

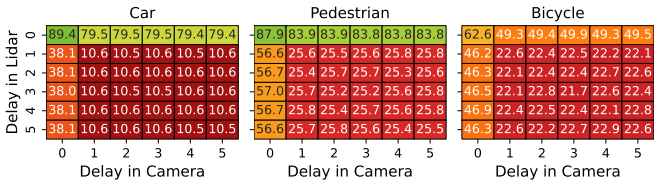


Fig. 9. Mul-DEJAVU attack impacts on 3D object detection performance of *BEVFusion* on nuScenes dataset for three object classes.

VI. EVALUATING DEJAVU ON MMF MODELS AND END-TO-END AD

This section evaluates the impact of DEJAVU-induced temporal misalignment on state-of-the-art MMF-based perception models across KITTI and NuScenes, and further realizes DEJAVU in an end-to-end AD simulator.

A. Datasets

We evaluate our approach on two widely used multimodal AD datasets for 3D object detection and multi-object tracking:

KITTI Tracking Dataset. The KITTI Tracking Dataset [31] was collected in Karlsruhe, Germany, across urban, suburban, and highway scenes. It provides a forward-facing RGB camera (1242×375 resolution) and a Velodyne HDL-64E LiDAR operating at ∼10 Hz. The dataset contains 21 training and 29 test sequences with frame-level 3D bounding boxes and identity annotations for three main classes: cars, pedestrians, and cyclists.

NuScenes Dataset. The NuScenes Dataset [32] was collected in Boston (USA) and Singapore, focusing on dense urban traffic under diverse conditions. It provides six surround-view RGB cameras, a Velodyne HDL-32E LiDAR (20 Hz), and five radars, with all annotations sampled at 2 Hz. The dataset consists of 1000 driving scenes, each 20 seconds long, with 3D bounding boxes and tracking IDs for 23 classes, including vehicles, pedestrians, bicycles, and traffic barriers.

B. Models

To systematically evaluate the effect of DEJAVU on MMF with different downstream tasks, we consider the following:

3D Object Detection: We evaluate DEJAVU on two representative MMF-based 3D object detection models. i) *MVXNet* [29], trained on the KITTI dataset, is an early fusion-based architecture that projects LiDAR point clouds into pseudo-image space and fuses them with camera image features at the voxel level. ii) *BEVFusion* [30] is a more advanced architecture, trained on the NuScenes dataset, that unifies multi-modal sensor inputs in the bird's eye view (BEV) representation space. BEVFusion is widely used in industry-grade AD software stacks, including Autoware.

Multi-Object Tracking (MOT): For evaluating tracking performance under DEJAVU, we consider *MMF-JDT* [33], trained on the KITTI tracking dataset, is a joint detection and tracking model that incorporates early and mid-level fusion strategies to align image and point cloud features for improved object association over time.

C. Attack Settings

To evaluate the impact of DEJAVU attack on MMF models, we introduce controlled delays in one or both sensor modalities (camera and LiDAR), as mentioned in Table I, and analyze the corresponding degradation in model performance. We systematically assess how different degrees and types of temporal misalignment affect the 3D object detection and multi-object tracking. Specifically, we use the S-ATM per modality as the delay parameter $\alpha \in \{0, 1, 2, 3, 4, 5\}$, where $\alpha = 0$ represents a perfectly synchronized sensor, and $\alpha = 5$ denotes a maximum delay of five frames for the affected modality. We consider both Uni-DEJAVU and Mul-DEJAVU attacks. In Uni-DEJAVU attack, delay is introduced in either the camera or the LiDAR input while keeping the other modality synchronized. In Mul-DEJAVU attacks, both camera and LiDAR streams are delayed independently, leading to varying degrees of temporal misalignment. We use the pretrained weights for the target model provided with the official implementations. We focus exclusively on attacking the test dataset by applying the defined temporal delays.

D. Evaluation Metrics

To assess the impact of DEJAVU, we analyze the model performance using task-specific evaluation metrics. For 3D object detection, we evaluate the models using mean average precision (mAP), which quantifies the accuracy of detected objects, as well as nuScenes Detection Score (NDS), particularly for the NuScenes dataset. For MOT, we use standard tracking metrics such as higher order tracking accuracy (HOTA), detection accuracy (DetA), association accuracy (AssA), multiple object tracking accuracy (MOTA), and Identity Switches (IDSW), which measure the effectiveness of object association.

E. Software Implementation

We implement and evaluate DEJAVU using Python 3.8 and PyTorch, utilizing open-source frameworks including MMDetection3D [60] and OpenPCDet [61]. Experiments were conducted on a server running Ubuntu 20.04.6 LTS with an Intel Xeon Gold 5520 (16 cores, 2.20GHz), 128GB RAM, and three NVIDIA RTX 6000 Ada GPUs.

VII. EVALUATION RESULTS

This section presents the DEJAVU attack impact on different MMF models and datasets, and discusses the key findings from the evaluation.

A. Impact of DEJAVU Attack on 3D Object Detection

We investigate the effectiveness of the proposed DEJAVU attack on multimodal 3D object detection using *MVXNet* and *BEVFusion*, evaluated on the KITTI and nuScenes datasets, respectively. We analyze the detection performance of 3D object detection across different object classes under varying levels of unimodal and multimodal sensor delay. Object detection models do not process sequential information; instead, their performance is affected only by frame-wise delays in each modality at any particular time, whether the delays are constant or random. Therefore, for simplicity and consistency, we only analyze the attack under the constant delay setting.

1) MVXNet on KITTI Dataset: Fig. 6 presents the 3D object detection performance of *MVXNet* under Uni-DEJAVU attacks, where either the camera or LiDAR input is delayed independently. Under benign (zero-delay) conditions, *MVXNet* achieves strong mAP across most object classes: approximately 84.1 for cars, 87.6 for pedestrians, and 73.2 for cyclists. The left plot shows that delaying the camera input alone–under Uni-DEJAVU camera attacks—has minimal effect on performance across all classes, with nearly constant mAPs. In contrast, the right plot highlights the model's high sensitivity to LiDAR delays: a 1-frame delay causes the car mAP to collapse from 84.1 to 9.7 (↓88.5%), pedestrian mAP from 87.6 to 34.2 (↓60.9%), and cyclist mAP from 73.2 to 32.9 (↓55.1%), with further degradation as delay increases. This indicates that LiDAR data is significantly more critical than camera input in *MVXNet*'s perception pipeline; hence, *MVXNet*'s high vulnerability against Uni-DEJAVU attack against LiDAR. However, Fig. 7 shows the mAP heatmaps across combinations of camera and LiDAR delays under Mul-DEJAVU attacks. Although the 1-frame LiDAR delay drops mAP from 55.1% to 88.5% for different objects, camera delay has almost no effect, indicating the dominance of LiDAR in *MVXNet*.

Key Findings. *MVXNet* heavily depends on LiDAR input for 3D object detection. Camera delay, under both Uni- or Mul-DEJAVU attack, has almost no effect, but even minor LiDAR misalignment leads to severe performance degradation. When both modalities are delayed, the impact is still only dominated by the LiDAR stream.

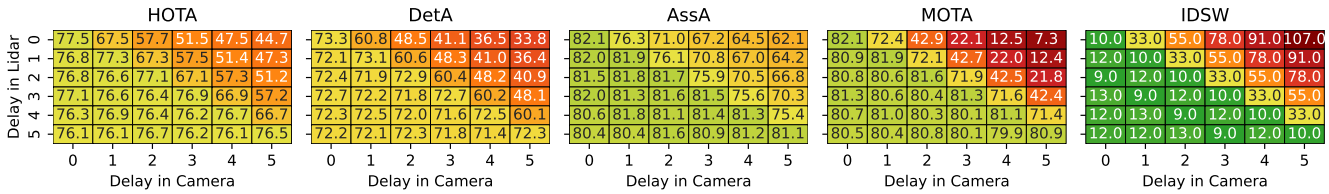
2) BEVFusion on nuScenes Dataset: Fig. 8 shows *BEVFusion*'s 3D detection performance under Uni-DEJAVU attacks. With no delay, the model achieves high mAP across most classes—approximately 88 for car and pedestrian, and slightly lower for the rest. With a 1-frame camera delay, mAP drops slightly across all classes (for instance, car mAP drops by 7.4%), but remains stable with further delays, showing *BEVFusion* is slightly vulnerable against Uni-DEJAVU camera attacks. In contrast, introducing a 1-frame LiDAR delay results in a substantial reduction in mAP for specific object classes, with performance dropping by 65.6% for cars (from 88.8 to 30.5), 35.1% for pedestrians (from 88.1 to 57.2), and 38.6% for motorcycles (from 75.9 to 46.6). Although performance remains stable with further LiDAR delays, this finding shows *BEVFusion*'s higher vulnerability against Uni-DEJAVU LiDAR attacks. Similarly, Fig. 9 illustrates *BEVFusion*'s performance under Mul-DEJAVU attacks for the three different objects (complete list of object is in Fig. 14). Although delaying the camera alone has a minimal impact on performance, combining it with a delay in the LiDAR stream results in a significant drop in mAP. For instance, a 1-frame delay in the camera or LiDAR stream reduces car mAP by 7.4% and 65.5%, respectively. However, when both modalities are delayed simultaneously by one frame, the mAP drops dramatically by 89.2% (from 88.8 to 9.6), highlighting a substantial compounding effect. This trend is consistently observed across all object classes.

Key Findings. BEVFusion is slightly affected by camera delays but is highly affected by LiDAR delays, further underscoring LiDAR's dominating role in 3D object detection. However, the impact becomes significant if both sensors are delayed simultaneously, even just by one frame.

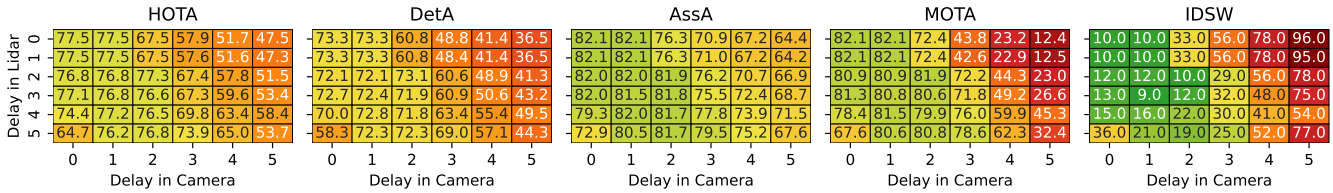
B. Impact of DEJAVU Attack on Multi Object Tracking

We investigate the effectiveness of the proposed DEJAVU attack on MOT algorithm using *MMF-JDT* on the KITTI dataset. We analyze the tracking performance for cars under varying levels of delays under Uni- and Mul-DEJAVU attack scenarios.

1) MMF-JDT on KITTI Dataset: This part studies the impact of DEJAVU attacks on *MMF-JDT* evaluated on the KITTI tracking dataset. For this evaluation, we consider both types of delays: constant (Fig. 10a) and random (Fig. 10b). Across both



(a) Constant Delay Scenario



(b) Random Delay Scenario

Fig. 10. DEJAVU attack impacts on multi-object (car) tracking performance of *MMF-JDT* on KITTI tracking dataset with respect to different metrics.

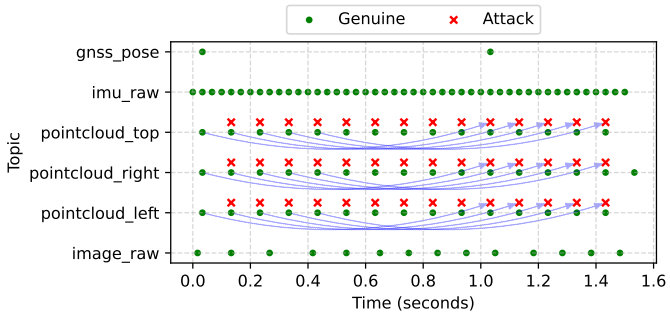


Fig. 11. Demonstration of replay-style DEJAVU attacks where malicious LiDAR messages closely follow genuine ones, increasing alignment likelihood with the malicious ones.

attack scenarios, the tracking performance declines as camera delay increases. Along with other tracking metrics, MOTA and IDSW suffer noticeable degradation as camera delays increase under Uni-DEJAVU attacks. For instance, IDSW—a metric that captures identity switches and ideally should be low, increases dramatically under increasing camera delays, underscoring the disruption in tracking consistency caused by DEJAVU attacks.

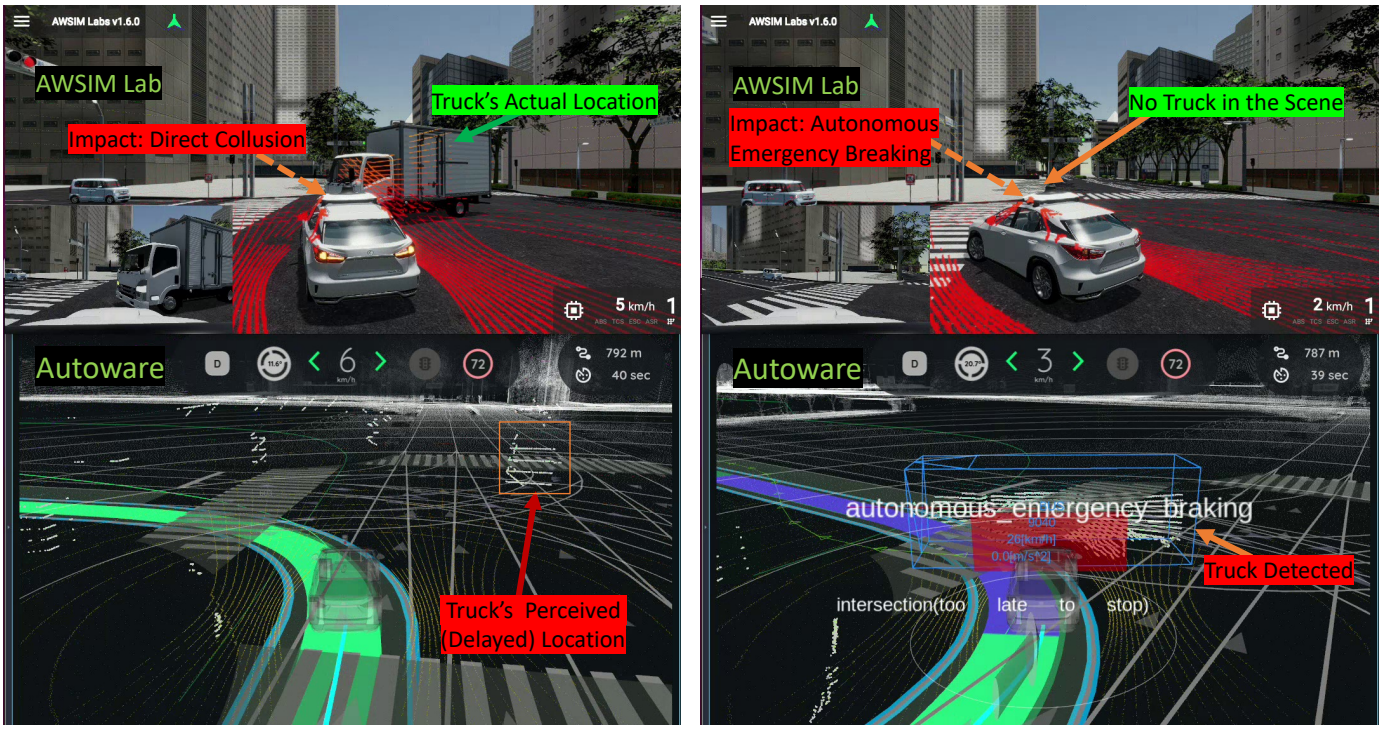
Furthermore, the values of different metrics across the heatmaps of Fig. 10a suggest that while camera delay under Uni-DEJAVU deteriorates the performance, delaying both the camera and LiDAR by exactly the same delay (*i.e.,* constant delay scenario) under Mul-DEJAVU attack diminishes the attack impact and mostly retains the performance. On the other hand, heatmaps in Fig. 10b show that Mul-DEJAVU attacks with random delays remain effective as different delays in both modalities break the sequence, making object tracking considerably more difficult. For instance, under the Mul-DEJAVU constant attack scenario with a five-frame delay, MOTA decreases by only 1.4%, whereas the random attack scenario results in a 60.5% drop.

Key Findings. In contrast to 3D object detection tasks, MOT appears to rely more heavily on camera inputs. This may be attributed to the fact that MOT does not require precise 3D bounding boxes; instead, the rich texture information in camera images may offer more effective contrastive representations than sparse point clouds. As a result, DEJAVU attacks can substantially impair MOT performance, particularly under Uni-DEJAVU (camera delay) and Mul-DEJAVU with random delay scenarios.

C. Simulation-Based End-to-End AD Setup

We conduct our experiments using Autoware, an open-source full-stack autonomous driving framework. Autoware is widely adopted in both commercial and public-sector deployments, including Level 4 autonomy trials and government-funded programs (e.g., the U.S. Department of Transportation's CARMA^{SM} initiative). Its extensive use in real-world systems makes it a realistic and representative platform for evaluating the safety and robustness of autonomous driving pipelines. To simulate real-world driving scenarios in a reproducible and controllable setting, we integrate Autoware with AWSIM Lab—a Unity-based, open-source simulator developed as part of the Autoware ecosystem—that provides high-fidelity urban environments and realistic sensor simulation. The simulated vehicle is equipped with a representative sensor suite including *GNSS*, *IMU*, three *Velodyne VLP-16 LiDARs*, and a *traffic light camera*. ROS 2 (Humble) works as the middleware, enabling seamless and real-time communication between AWSIM's simulated sensors and Autoware full autonomy stack, allowing us to evaluate the impact of DEJAVU attack in a safe yet realistic environment. Our experiments are conducted on Tokyo's Nishishinjuku district road map.

DEJAVU Attack Impact on AD Simulation. We demonstrate the DEJAVU attack in a full-stack autonomous driving pipeline, targeting the LiDAR sensor under attacker capability C_3 (see Section IV-A). Given LiDAR's dominant role in 3D



(a) False negative scenario.

(b) False positive scenario.

Fig. 12. Impact of delayed yet valid LiDAR data on Autoware. a) The ego misses the oncoming truck, causing a head-on collision, and b) the ego vehicle brakes for a non-existent object from delayed data.

perception, the attacker impersonates three legitimate LiDAR nodes by subscribing to their respective ROS 2 topics and monitoring inter-frame intervals (10 Hz) to predict the next transmission times. The attacker stores recent point cloud messages and, at the time of attack, publishes forged messages with previously captured data with updated timestamps—just before the expected legitimate message. This increases the likelihood that the forged message is selected by the time-based synchronizer if it aligns more closely with the timestamps of other modalities (e.g., IMU, camera). Figure 11 illustrates how these delayed messages are positioned and transmitted on the same topics, effectively impersonating genuine LiDAR messages in real time.

When the forged messages are utilized in the downstream tasks, in the most severe case, the system completely misses the presence of an actual oncoming vehicle, resulting in a head-on collision at an intersection—despite nearby objects being within sensor range (Fig. 12a). This constitutes a false negative perception failure with life-threatening implications. In another scenario, delayed LiDAR data causes the ego vehicle to perceive a non-existent obstacle—a vehicle that has already passed—leading to unnecessary emergency braking (Fig. 12b). This false positive event can create rear-end collision risks. In both cases, the outdated sensor data was used to repeatedly overwrite fresh messages, degrading the temporal integrity of the perception pipeline.

Beyond object-level failures, we observe broader impacts across the autonomy stack. The tracker may assign separate IDs to genuine and delayed instances of the same object, interpreting them as distinct entities. Similarly, temporal inconsistencies introduced in the LiDAR stream desynchronize SLAM modules, leading to localization drift and control failures such as veering off-lane or collisions with curbs and roadside objects (as shown in Fig. 18 in the Appendix).

VIII. Discussion

A. Attack Limitations

While the DejaVu attack demonstrates the vulnerability of multimodal perception systems to temporal misalignment, there are several limitations that constrain its applicability in real-world settings. First, the attack has not been validated on a physical vehicle platform, where additional practical challenges such as sensor noise, actuator delays, and system integration issues may affect both the feasibility and effectiveness of the attack. Besides, the attack assumes a high level of knowledge about the target system, including the sensors and network architecture. In practice, the attack requires adversarial access to the in-vehicle network. Although not trivial, prior work shows that physical access through the OBD-II port, remote exploitation of infotainment systems, or supply-chain insertion can provide such capabilities [18]. However, the adversary would still need to perform an exploration phase to gather this information, which introduces additional complexity and may limit the ability to execute the attack stealthily.

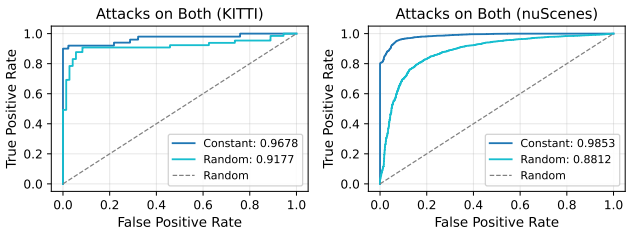


Fig. 13. ROC performance of the OC-SVM–based temporal misalignment detector under mul-DEJAVU attacks on KITTI (left) and nuScenes (right). Detection uses 3×3 cross-modal similarity matrices constructed from historic shared camera–LiDAR embeddings.

Moreover, the evaluation does not account for real-world network-induced delays and variability, which could impact the targeted timing manipulation strategies. In actual deployment, the presence of stochastic latency and jitter may reduce the precision with which an attacker can control temporal misalignment, potentially diminishing the attack's effectiveness. Finally, the evaluation was conducted exclusively on an offline dataset with relatively low frame rates. Real-time systems typically operate at higher frame rates, and the metrics and thresholds used in this offline evaluation may not directly translate to real-world performance. Consequently, the practical impact of the attack on deployed autonomous systems may differ from the results observed in the experimental study.

B. Defense Strategies Against DEJAVU

Hardening defenses can reduce the attack surface by securing sensor timestamps before fusion. This can be achieved via authenticated, hardware-anchored timestamps with cryptographic signatures, monotonic sequence numbers, and hardware-level clocks, such as network interface controller/SoC real-time clock (RTC). Combining multiple time sources—PTP, GNSS, and local RTCs—further strengthens temporal integrity. While these measures add overhead, they significantly raise the barrier to adversarial manipulation.

Detection defenses can identify temporal misalignments in real time. Techniques include intermodality temporal-consistency analysis of embeddings, kinematic cross-checks using IMU, odometry, and other controller area networks (CAN) signals, and statistical monitoring of monotonicity, jitter, and freshness counters, before fusion. These mechanisms can detect out-of-order or replayed frames, allowing rapid response to potential attacks.

Mitigation techniques can limit the impact of detected misalignments on vehicle control. Delay-aware adaptive fusion compensates for inter-modal lags, weighs sensor inputs, and computes confidence scores. If confidence is low, conservative measures, such as slowing down, increasing spacing, or lowering autonomy, can be applied, ensuring safety even under temporal attacks.

1) One-Class SVM Detection Defense: To detect DEJAVU attack, we design an unsupervised defense based on cross-modal similarity modeling and One-Class SVM (OC-SVM) classification. We first learn a *shared representation* that projects camera and LiDAR features into a common embedding space. For each of the fusion pairs, we compute the pairwise cross-modal similarity and form a temporal similarity matrix over a sliding window of w consecutive messages ($w = 3$). These $w \times w$ similarity snapshots are flattened and used as inputs to the OC-SVM. To prevent data leakage, the detector is trained exclusively on the first half of benign (attack-free) test data and evaluated on the remaining half, which contains injected attacks.

Figure 13 reports the ROC curves of the OC-SVM detector under mul-DEJAVU attacks (as conducted on Section V-A) on KITTI and nuScenes. Under the *constant attack*, the detector achieves strong performance on both datasets (AUC $=$ 0.9678 on KITTI and 0.9853 on nuScenes), as the prolonged freezing of timestamps introduces persistent degradation in cross-modal similarity that deviates from the learned benign manifold. In contrast, detection under the *random attack* is noticeably weaker (AUC $=$ 0.9177 on KITTI and 0.8812 on nuScenes). This gap arises because random forward and backward clock perturbations intermittently preserve short-term alignment structure, allowing portions of the similarity matrices to remain consistent with benign temporal patterns.

These results expose fundamental limitations of learning-based detectors under temporally irregular attacks. While constant offsets induce persistent distortions that are easier to detect, random offsets produce transient, stochastic misalignments that are difficult to distinguish from benign timing jitter. More broadly, detectors trained on limited temporal patterns risk bias toward the training distribution and may fail to generalize to unseen driving contexts and adaptive attacks. This motivates the need for more robust and generalized defenses, as well as the integration of additional modalities such as CAN bus and IMU data to improve sensing fidelity and strengthen cross-modal integrity under realistic and adversarial conditions.

IX. CONCLUSION

This work presents DEJAVU, a temporal misalignment attack that exploits synchronization vulnerabilities in multimodal perception systems for autonomous driving. Through extensive evaluations on state-of-the-art 3D object detection and multi-object tracking models, we uncover modality-specific vulnerabilities: 3D detection models are predominantly reliant on LiDAR and suffer severe degradation—with up to 88.5% drop in mAP—from even a single-frame LiDAR delay, while MOT model exhibits heightened sensitivity to camera stream disruptions, with MOTA dropping by 73% under just three-frame camera delays. These findings highlight the critical need for synchronization-aware design in perception architectures and emphasize the importance of robust temporal consistency checks in safety-critical autonomous systems.

ACKNOWLEDGMENT

This work was supported in part by the Office of Naval Research under grants N00014-24-1-2730, and the National

Science Foundation under grants 2235232, 2312447, 2247560, 2154929, 2154930, 2238635, 2403758, 2509636, 2312794, and a fellowship from the Amazon-Virginia Tech Initiative for Efficient and Robust Machine Learning.

LLM Usage Disclosure.

LLMs were used for editorial purposes in this manuscript, and all outputs were inspected by the authors to ensure accuracy and originality.

References

[1] X. Zhang, Y. Gong, J. Lu, J. Wu, Z. Li, D. Jin, and J. Li, "Multi-modal fusion technology based on vehicle information: A survey," *IEEE Transactions on Intelligent Vehicles*, vol. 8, no. 6, pp. 3605–3619, 2023.

[2] E. Marti, M. A. De Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE intelligent transportation systems magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[3] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, "I can see the light: Attacks on autonomous vehicles using invisible lights," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 1930–1944.

[4] L. de Paula Veronese, F. Auat-Cheein, F. Mutz, T. Oliveira-Santos, J. E. Guivant, E. De Aguiar, C. Badue, and A. F. De Souza, "Evaluating the limits of a lidar for an autonomous driving localization," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1449–1458, 2020.

[5] F. Engels, P. Heidenreich, M. Wintermantel, L. Stäcker, M. Al Kadi, and A. M. Zoubir, "Automotive radar signal processing: Research directions and practical challenges," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 865–878, 2021.

[6] R. Bramon, I. Boada, A. Bardera, J. Rodriguez, M. Feixas, J. Puig, and M. Sbert, "Multimodal data fusion based on mutual information," *IEEE Transactions on Visualization and Computer Graphics*, vol. 18, no. 9, pp. 1574–1587, 2012.

[7] Z. Cheng, H. Choi, J. Liang, S. Feng, G. Tao, D. Liu, M. Zuzak, and X. Zhang, "Fusion is not enough: Single modal attacks on fusion models for 3d object detection," *arXiv preprint arXiv:2304.14614*, 2023.

[8] K. Huang, B. Shi, X. Li, X. Li, S. Huang, and Y. Li, "Multi-modal sensor fusion for auto driving perception: A survey," *arXiv preprint arXiv:2202.02703*, 2022.

[9] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[10] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[11] D. Kuhse, N. Holscher, M. Gunzel, H. Teper, G. Von Der Bruggen, J.-J. Chen, and C.-C. Lin, "Sync or sink? the robustness of sensor fusion against temporal misalignment," in *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2024, pp. 122–134.

[12] *Specification of Time Synchronization for Adaptive Platform*, Release r20-11 ed., AUTOSAR, 2020.

[13] K. B. Stanton, "Distributing deterministic, accurate time for tightly coordinated network and software applications: Ieee 802.1 as, the tsn profile of ptp," *IEEE Communications Standards Magazine*, vol. 2, no. 2, pp. 34–40, 2018.

[14] L. Deng, G. Xie, H. Liu, Y. Han, R. Li, and K. Li, "A survey of real-time ethernet modeling and design methodologies: From avb to tsn," *ACM Computing Surveys (CSUR)*, vol. 55, no. 2, pp. 1–36, 2022.

[15] S. Shi, Y. Xiao, C. Du, M. H. Shahriar, A. Li, N. Zhang, Y. T. Hou, and W. Lou, "Ms-ptp: protecting network timing from byzantine attacks," in *Proceedings of the 16th ACM conference on security and privacy in wireless and mobile networks*, 2023, pp. 61–71.

[16] A. Finkenzeller, O. Butowski, E. Regnath, M. Hamad, and S. Steinhorst, "Ptpsec: Securing the precision time protocol against time delay attacks using cyclic path asymmetry analysis," in *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*. IEEE, 2024, pp. 461–470.

[17] A. Finkenzeller, A. Fucks, E. Regnath, M. Hamad, and S. Steinhorst, "Securing the precision time protocol with sdn-enabled cyclic path asymmetry analysis," *ACM Transactions on Cyber-Physical Systems*, 2025.

[18] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *20th USENIX security symposium (USENIX Security 11)*, 2011.

[19] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*, 2015.

[20] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, no. S 91, pp. 1–91, 2015.

[21] S. Yeasmin and A. Haque, "A multi-factor authenticated blockchain-based ota update framework for connected autonomous vehicles," in *2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*. IEEE, 2021, pp. 1–6.

[22] A. Ghosal, S. Halder, and M. Conti, "Secure over-the-air software update for connected vehicles," *Computer Networks*, vol. 218, p. 109394, 2022.

[23] O. Avatefipour and H. Malik, "State-of-the-art survey on in-vehicle network communication (can-bus) security and vulnerabilities," *arXiv preprint arXiv:1802.01725*, 2018.

[24] M. H. Eiza and Q. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 45–51, 2017.

[25] G. Deng, G. Xu, Y. Zhou, T. Zhang, and Y. Liu, "On the (in) security of secure ros2," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, 2022, pp. 739–753.

[26] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[27] T. Huck, A. Westenberger, M. Fritzsche, T. Schwarz, and K. Dietmayer, "Precise timestamping and temporal synchronization in multi-sensor fusion," in *2011 IEEE intelligent vehicles symposium (IV)*. IEEE, 2011, pp. 242–247.

[28] A. Finkenzeller, A. Roberts, M. Bellone, O. Maennel, M. Hamad, and S. Steinhorst, "Sensor fusion desynchronization attacks," in *37th Euromicro Conference on Real-Time Systems (ECRTS 2025)*. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025, pp. 6–1.

[29] V. A. Sindagi, Y. Zhou, and O. Tuzel, "Mvx-net: Multimodal voxelnet for 3d object detection," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7276–7282.

[30] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, and S. Han, "Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation," in *2023 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2023, pp. 2774–2781.

[31] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[32] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[33] X. Wang, C. Fu, J. He, M. Huang, T. Meng, S. Zhang, H. Zhou, Z. Xu, and C. Zhang, "A multi-modal fusion-based 3d multi-object tracking framework with joint detection," *IEEE Robotics and Automation Letters*, 2024.

[34] Y. Cao, S. H. Bhupathiraju, P. Naghavi, T. Sugawara, Z. M. Mao, and S. Rampazzi, "You can't see me: Physical removal attacks on {lidar-based} autonomous vehicles driving frameworks," in *32nd USENIX security symposium (USENIX Security 23)*, 2023, pp. 2993–3010.

[35] Z. Jin, Q. Jiang, X. Lu, C. Yan, X. Ji, and W. Xu, "Phantomlidar: Cross-modality signal injection attacks against lidar," *arXiv preprint arXiv:2409.17907*, 2024.

[36] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic, "Security analysis of {Camera-LiDAR} fusion against {Black-Box} attacks on autonomous vehicles," in *31st USENIX Security Symposium (USENIX Security 22)*, 2022, pp. 1903–1920.

[37] N. Wang, S. Xie, T. Sato, Y. Luo, K. Xu, and Q. A. Chen, "Revisiting physical-world adversarial attack on traffic sign recognition: A commercial systems perspective," *arXiv preprint arXiv:2409.09860*, 2024.

[38] Y. Zhu, C. Miao, H. Xue, Y. Yu, L. Su, and C. Qiao, "Malicious attacks against multi-sensor fusion in autonomous driving," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 436–451.

[39] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *2021 IEEE symposium on security and privacy (SP)*. IEEE, 2021, pp. 176–194.

[40] E. Bagdasaryan, R. Jha, V. Shmatikov, and T. Zhang, "Adversarial illusions in {Multi-Modal} embeddings," in *33rd USENIX Security Symposium (USENIX Security 24)*, 2024, pp. 3009–3025.

[41] K. Xiahou, Y. Liu, and Q. Wu, "Robust load frequency control of power systems against random time-delay attacks," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 909–911, 2020.

[42] X.-C. Shangguan, Y. He, C.-K. Zhang, W. Yao, Y. Zhao, L. Jiang, and M. Wu, "Resilient load frequency control of power systems to compensate random time delays and time-delay attacks," *IEEE Transactions on Industrial Electronics*, vol. 70, no. 5, pp. 5115–5128, 2022.

[43] H. Song, S. Zhu, and G. Cao, "Attack-resilient time synchronization for wireless sensor networks," *Ad hoc networks*, vol. 5, no. 1, pp. 112–125, 2007.

[44] W. Zhai, L. Liu, Y. Ding, S. Sun, and Y. Gu, "Etd: an efficient time delay attack detection framework for uav networks," *IEEE transactions on information forensics and security*, vol. 18, pp. 2913–2928, 2023.

[45] W. Zhai, S. Sun, L. Liu, Y. Ding, and W. Lu, "Hotd: A holistic cross-layer time-delay attack detection framework for unmanned aerial vehicle networks," *Journal of Parallel and Distributed Computing*, vol. 177, pp. 117–130, 2023.

[46] F. Luo, Z. Wang, and B. Zhang, "Impact analysis and detection of time-delay attacks in time-sensitive networking," *Computer Networks*, vol. 234, p. 109936, 2023.

[47] A. Li, J. Wang, and N. Zhang, "Chronos: Timing interference as a new attack vector on autonomous cyber-physical systems," in *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021, pp. 2426–2428.

[48] A. Li, H. Liu, J. Wang, and N. Zhang, "From timing variations to performance degradation: Understanding and mitigating the impact of software execution timing in slam," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 13 308–13 315.

[49] T. Sato, R. Suzuki, Y. Hayakawa, K. Ikeda, O. Sako, R. Nagata, R. Yoshida, Q. A. Chen, and K. Yoshioka, "On the realism of lidar spoofing attacks against autonomous driving vehicle at high speed and long distance," in *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2025.

[50] C. Gao, G. Wang, W. Shi, Z. Wang, and Y. Chen, "Autonomous driving security: State of the art and challenges," *IEEE Internet of Things Journal*, vol. 9, no. 10, pp. 7572–7595, 2021.

[51] Y. Man, R. Muller, M. Li, Z. B. Celik, and R. Gerdes, "That person moves like a car: Misclassification attack detection for autonomous systems using spatiotemporal consistency," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 6929–6946.

[52] S. Li, S. Zhu, S. Paul, A. Roy-Chowdhury, C. Song, S. Krishnamurthy, A. Swami, and K. S. Chan, "Connecting the dots: Detecting adversarial perturbations using context inconsistency," in *European Conference on Computer Vision*. Springer, 2020, pp. 396–413.

[53] Y. Xu, G. Deng, X. Han, G. Li, H. Qiu, and T. Zhang, "Physcout: Detecting sensor spoofing attacks via spatiotemporal consistency," in *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024, pp. 1879–1893.

[54] C. Xiang, C. Feng, X. Xie, B. Shi, H. Lu, Y. Lv, M. Yang, and Z. Niu, "Multi-sensor fusion and cooperative perception for autonomous driving: A review," *IEEE Intelligent Transportation Systems Magazine*, vol. 15, no. 5, pp. 36–58, 2023.

[55] M. De Vincenzi, G. Costantino, I. Matteucci, F. Fenzl, C. Plappert, R. Rieke, and D. Zelle, "A systematic review on security attacks and countermeasures in automotive ethernet," *ACM Computing Surveys*, vol. 56, no. 6, pp. 1–38, 2024.

[56] D. Yang, M. Tang, and Y. Ni, "Robustness of automotive supply chain networks based on complex network analysis," *Electronic Commerce Research*, pp. 1–28, 2024.

[57] C. DeCusatis, R. M. Lynch, W. Kluge, J. Houston, P. A. Wojciak, and S. Guendert, "Impact of cyberattacks on precision time protocol," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 5, pp. 2172–2181, 2019.

[58] W. Alghamdi and M. Schukat, "Precision time protocol attack strategies and their resistance to existing security extensions," *Cybersecurity*, vol. 4, pp. 1–17, 2021.

[59] R. Annessi, J. Fabini, F. Iglesias, and T. Zseby, "Encryption is futile: Delay attacks on high-precision clock synchronization," *arXiv preprint arXiv:1811.08569*, 2018.

[60] M. Contributors, "MMDetection3D: OpenMMLab next-generation platform for general 3D object detection," https://github.com/open-mmlab/mmdetection3d, 2020.

[61] O. D. Team, "Openpcdet: An open-source toolbox for 3d object detection from point clouds," https://github.com/open-mmlab/OpenPCDet, 2020.

APPENDIX A
EXTENDED RESULTS ON BEVFUSION MODEL

Fig. 14 shows the Mul-DEJAVU attack impacts on 3D object detection performance of *BEVFusion* on the nuScenes dataset for all different object classes. Figure 15 further shows the impact of Mul-DEJAVU on the performance of *BEVFusion* on the nuScenes dataset regarding average mAP and NDS score.

APPENDIX B
DEJAVU ATTACK TO MANIPULATE TIMESTAMP INTEGRITY.

In this experiment, we assume that the attacker is targeting the LiDAR sensor with the attacker capability C_2 (see Section IV-A). As shown in Fig. 16(top), the attacker first publishes six LiDAR messages with benign timestamps; hence, they are very close to the camera messages' timestamps. The attacker then deliberately introduces a constant offset of 5 seconds to the timestamps of the next eleven LiDAR messages, while continuing to transmit genuine, real-time data. As a result, the timestamps of these LiDAR messages are abruptly increased, as depicted in the figure.

In the bottom panel of Fig. 16, the message content is accurately synchronized for both camera and LiDAR pairs, for all benign timestamps, which are indicated by green rectangles. When the attacker sends manipulated timestamps, the synchronizer forces camera messages to align with LiDAR messages inaccurately, which are shown using red arrows in the figure. This demonstrates that, although the contents of camera and LiDAR messages were sequential, the synchronizer fails to align them accurately during the attack, as it prioritizes the timestamps of the messages. The impact of this attack is similar to the attack we see in Fig. 1 and Fig. 17.

APPENDIX C
IMPACT OF DEJAVU ON SLAM

Fig. 18 illustrates the impact of DEJAVU attack on SLAM due to the delay in the LiDAR stream.

APPENDIX D
HARDWARE-IN-THE-LOOP TESTBED DESCRIPTION

A. Hardware Architecture

Our testbed emulates an in-vehicle automotive Ethernet network using three Raspberry Pi 4B units, each equipped with Time-Sensitive Networking (TSN) Network Interface Cards (NICs), connected through RAD-Moon media converters to a RAD-Jupiter TSN-capable switch. This architecture forms a realistic automotive Ethernet backbone where sensor data streams traverse the same network path as they would in a production vehicle, enabling controlled studies of temporal misalignment attacks and their impact on multimodal fusion.

The three Raspberry Pi units serve distinct roles in the data pipeline. Two source nodes replay prerecorded camera and LiDAR datasets, publishing them as ROS 2 `sensor_msgs/Image` and `sensor_msgs/PointCloud2` messages, respectively. The third unit acts as an ECU fusion node, subscribing to both sensor streams, performing temporal synchronization, and logging timing metrics for post-processing analysis. Each Raspberry Pi is connected to a TSN-capable Ethernet NIC that provides hardware timestamping and 100BASE-T1/TSN functionality, bridging conventional Ethernet interfaces with the automotive Ethernet domain. A fourth node is used to take part in PTP communication and to advertise a superior clock quality, and eventually, to be selected as Grandmaster (GM).

Three RAD-Moon media converters serve as transparent Layer 1/Layer 2 bridges, converting between 100BASE-T1 automotive Ethernet and conventional 10/100BASE-TX Ethernet. Two converters connect the source nodes to the network backbone, while a third converter bridges the ECU node. Each converter operates in full-duplex mode, automatically forwarding traffic between its two PHY interfaces while preserving link timing characteristics. The RAD-Jupiter switch, based on the Marvell 88Q5050 ASIC, serves as the central aggregation point with multiple 100BASE-T1 ports and an integrated AVB/TSN stack, routing time-sensitive traffic with deterministic latency guarantees. The physical connectivity follows a star topology centered on the switch, with camera and LiDAR data streams converging at the switch before being routed to the fusion ECU.

B. Software Stack

All Raspberry Pi units in the MMF pipeline run Raspberry Pi OS (64-bit) with vendor-provided TSN NIC drivers that enable correct PHY mode selection, hardware timestamping, and kernel-level optimizations for deterministic network behavior. The software stack uses ROS 2 (rclpy) for inter-node communication, employing standard message types (`sensor_msgs/Image` for camera frames and `sensor_msgs/PointCloud2` for LiDAR point clouds) with timestamps embedded in message headers via ROS 2's clock abstraction. Network configuration uses static IP addressing for deterministic routing, while the media converters and switch operate transparently at Layer 2, requiring minimal configuration beyond PHY mode selection.

C. ROS 2 Node Architecture

1) Sensor Replay Nodes: The camera and LiDAR nodes abstract prerecorded datasets as live sensor streams, enabling reproducible experiments with controlled timing characteristics. Both nodes employ a deterministic, time-based indexing mechanism that maps wall-clock time to frame indices through a shared seed function, ensuring synchronized frame selection across sensors while decoupling frame selection from ROS timer callbacks. This design makes frame ID selection a function of absolute time rather than message count, providing temporal consistency across experimental runs.

The camera node publishes images at a configurable rate (typically 10 Hz), embedding the current ROS 2 timestamp in each message header and encoding the frame index in the `frame_id` field. The LiDAR node follows an identical pattern, publishing point cloud messages with synchronized timing. Additionally, the LiDAR node includes built-in support

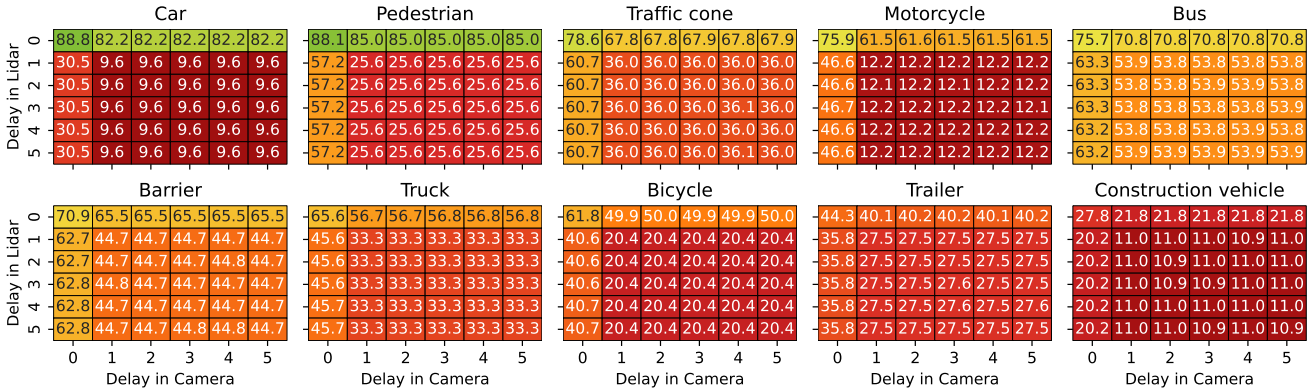


Fig. 14. Mul-DejaVu attack impacts on 3D object detection performance of *BEVFusion* on nuScenes dataset for all different object classes.

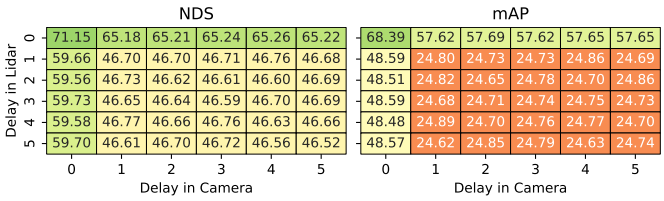


Fig. 15. Overall Mul-DejaVu attack impacts on 3D object detection performance of *BEVFusion* on nuScenes dataset for all the object classes regarding i) NDS, ii) mAP.

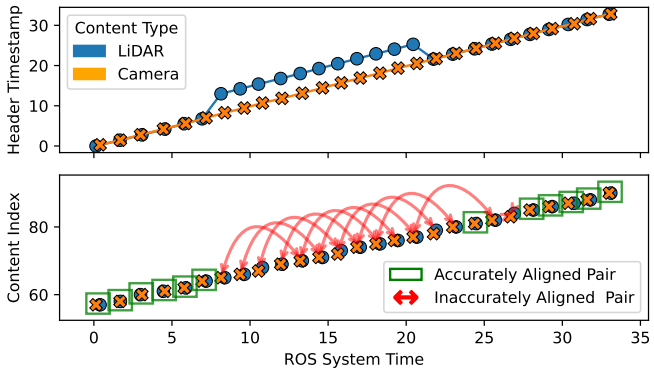


Fig. 16. ROS System time vs (top) Timestamp of Camera and LiDAR messages, and (bottom) Message content of Camera and LiDAR messages with

for temporal offset injection, allowing controlled studies of timestamp manipulation effects. This capability enables systematic injection of temporal misalignment without modifying the fusion node, supporting modular attack experimentation.

2) Fusion and Synchronization Node: The fusion node implements the ECU fusion logic, subscribing to both camera and LiDAR streams and aligning them temporally using ROS 2's `ApproximateTimeSynchronizer`. The synchronizer employs configurable queue buffering and temporal slop tolerance (typically ± 0.1 second) to match messages with similar timestamps, handling out-of-order arrivals caused by network jitter or temporal attacks. Separate raw subscriptions are maintained for per-message logging, recording all received messages independently of the fusion timing logic to capture complete arrival statistics.

When the synchronizer identifies a matching camera-LiDAR pair within the temporal tolerance window, the fusion callback is triggered with both messages, which involves optional MMF-based perception for visualization. In that case, the synchronized image and point cloud data are then passed to the MVXNet perception model, which performs multimodal fusion and 3D object detection, enabling evaluation of how temporal misalignment affects fusion quality and detection accuracy.

Appendix E
End-to-End Autoware Testbed Description

A. Simulation and System Configuration

Our end-to-end evaluation is conducted using the Unity-based AWSIM Lab simulator tightly integrated with the open-source Autoware autonomous driving stack. AWSIM provides a high-fidelity digital twin of the ego vehicle, featuring photorealistic three-dimensional urban environments, realistic vehicle dynamics, and configurable automotive-grade sensors, including LiDAR, cameras, IMUs, and GNSS. Sensor extrinsics and vehicle geometry are specified through a URDF model, while a unified simulation clock ensures synchronized multi-sensor data generation. Native ROS 2 integration enables closed-loop operation by publishing simulated sensor streams and vehicle states and subscribing to drive-by-wire control commands from Autoware, thereby coupling perception, planning, and control in real time.

We employ a high-definition urban map of Tokyo's Nishishinjuku district, which combines precise lane-level geometry, semantic Lanelet2 annotations, and stochastic traffic agents. The environment includes static assets (roads, buildings, and infrastructure) as well as dynamic actors (vehicles and pedestrians), enabling systematic evaluation across diverse traffic densities, intersection layouts, and environmental conditions. Semantic map annotations enforce traffic rules and directly

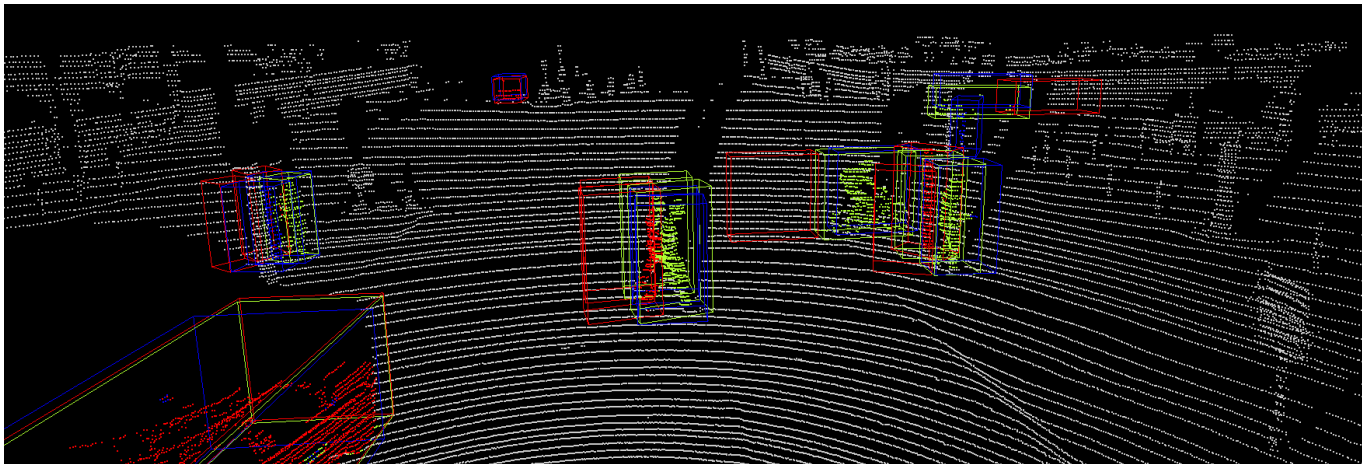


Fig. 17. Visualization of the impact of the DEJAVU attack on a camera–LiDAR fusion-based perception model. Ground-truth boxes are shown in blue, benign predictions in green, and predictions under DEJAVU in red. MMF-based fused predictions are overlaid on the current LiDAR frame. While benign predictions closely track the ground truth, delaying the LiDAR stream by five frames under DEJAVU induces cross-sensor temporal misalignment, shifting predicted bounding boxes toward objects' past locations and causing some objects to be missed. This behavior suggests an overreliance on LiDAR inputs relative to the camera stream. A corresponding Camera-plane visualization is provided in the Appendix (Fig. 1).

Fig. 18. Impact of delayed but valid LiDAR data on Autoware: the induced SLAM drift propagates to the planning module, ultimately causing the vehicle to collide with the curb.

support localization, prediction, and behavior planning, yielding a realistic urban driving workload.

B. Perception

Perception is primarily driven by LiDAR-based deep object detection. Raw point clouds are processed by a 3D neural network detector based on the CenterPoint/PointPillars family of architectures, which performs bird's-eye-view object detection and semantic classification of surrounding traffic participants, including vehicles, pedestrians, and cyclists.

The detected objects are temporally associated and filtered by a multi-object tracking module based on Kalman filtering and probabilistic data association, yielding stable object identities and refined kinematic state estimates. In parallel, LiDAR returns are used to construct a dynamic occupancy grid that encodes free space and obstacles for downstream collision checking and trajectory optimization. Traffic-signal handling is performed through map-based stop-line association. Overall, this perception pipeline reflects a production-style LiDAR-centric sensing architecture widely used in modern autonomous driving systems.

C. Planning

The planning subsystem follows a hierarchical decision-making architecture. A mission planner computes a global

route over the lane-level map using semantic navigation constraints. A behavior planner subsequently enforces traffic rules and selects high-level maneuvers such as lane following, lane changes, yielding, and stopping based on the current traffic context.

A motion planner then optimizes a dynamically feasible trajectory and velocity profile that minimizes tracking error and collision risk while satisfying kinematic and dynamic vehicle constraints. A centralized planner manager arbitrates among planning modules and integrates predicted object trajectories, occupancy grids, and traffic-signal states to ensure safe, lawful, and continuous navigation in complex urban environments.

D. Control

Vehicle control is realized through a combined lateral and longitudinal control architecture operating in closed loop with the planner. Lateral control is implemented using a linear Model Predictive Control (MPC) formulation that minimizes path-tracking error under steering and stability constraints by solving a constrained quadratic program at each control cycle.

Longitudinal control employs a feed-forward and feedback structure with PID compensation to regulate vehicle speed relative to the planned velocity profile, incorporating delay and slope compensation for improved tracking fidelity. Together, these controllers generate smooth, stable, and physically realistic vehicle motion in simulation.

By integrating AWSIM's high-fidelity digital twin with Autoware's production-grade LiDAR-based perception, hierarchical planning, and MPC-based control, our testbed enables rigorous, fully reproducible end-to-end evaluation of autonomous driving pipelines under realistic urban conditions while maintaining complete experimental safety and controllability.