Closing the Temporal Gap: A Deterministic Simulation Framework for Physics-Aware Automotive Intrusion Detection

Liron Ahmeti*, Klara Dolos*, Conrad Meyer*, Andreas Attenberger*, Sebastian Fischer† Rudolf Hackenberg‡
*Research Unit, Central Office for Information Technology in the Security Sector
Munich, Germany
Email: poststelle@zitis.bund.de
†Dept. Informatics and Mathematics, OTH Regensburg
Regensburg, Germany
Email: sebastian.fischer@oth-regensburg.de
‡Dept. Informatics and Mathematics, OTH Regensburg
Regensburg, Germany
Email: rudolf.hackenberg@oth-regensburg.de

***Abstract*—Validating Intrusion Detection Systems (IDS) for autonomous vehicles requires simulation environments that are not only visually realistic but forensically deterministic. Current real-time simulators often couple network arbitration to rendering framerates, introducing non-deterministic timing artifacts (jitter) that obscure the signatures of cyber-attacks, such as bus flooding or replay injections. This paper presents SimDAT-AV, a simulation framework designed to close this temporal gap. By decoupling simulation time from host execution speed via a Synchronous Lockstep architecture, we achieve bit-exact reproducibility of Controller Area Network (CAN) bus traffic (validated via SHA-256 checksums), enabling the precise reconstruction of attack scenarios regardless of computational load. Furthermore, we introduce a Physics-Aware IDS that leverages a high-fidelity Virtual Powertrain Model to detect semantic anomalies. We demonstrate that correlating internal vehicle states (e.g., engine engine Revolutions Per Minute (RPM) vs. wheel speed) allows for the detection of sophisticated spoofing attacks that satisfy protocol-level checks but violate physical consistency. The proposed framework eliminates execution jitter (collapsing timing-induced variance to a single deterministic outcome), providing a robust foundation for verifying automotive security mechanisms under reproducible, forensic conditions.**

Keywords-Automotive Security; Intrusion Detection; Digital Forensics; Deterministic Simulation

I. Introduction

Safety validation in autonomous driving is not a monolith. While kinematic simulation excels at testing planning logic, for security validation—particularly for Intrusion Detection Systems (IDS)—it often solves the wrong problem. In a forensic setting, the decisive evidence is rarely the vehicle's trajectory. Instead, it is the integrity of the underlying signals: bus arbitration timing, message inter-arrival patterns, and the causal consistency between physical measurements and digital messages [1]. Current validation approaches force a trade-off. Physical testing captures essential sensor variability, but safety regulations prevent the systematic injection of severe cyber-attacks on public roads. Simulation removes this constraint, yet standard real-time environments, such as CARLA [2] or AirSim [3], typically abstract away the very effects an IDS monitors. Because many detectors rely on micro-timing regularities as their definition of 'normal,' even small simulation-induced distortions become indistinguishable from attacks. In many simulators, critical forensic artifacts—clock drift, arbitration latency, weather-driven signal degradation—are either smoothed out or become entangled with the host computer's performance [4]. For a detector monitoring inter-arrival times, this breaks the baseline; the distribution of normal traffic shifts simply because the GPU is busy. This mismatch creates a practical reality gap between synthetic traces and vehicle logs. Trained on such regularized data, models learn decision boundaries that look crisp because the world they see is crisp. On the road, that crispness vanishes: rain attenuation can be flagged as malicious, while a protocol-correct spoofing attack might look perfectly normal. To bridge this gap, we introduce *SimDAT-AV*, a simulation framework engineered for forensic readiness, i.e., the capability to produce data that supports causal reconstruction and allows observed anomalies to be attributed to specific causes rather than simulator artifacts. Where others pursue visual realism, we prioritize two properties essential for forensics: explicit parametric modeling of degradations and tick-synchronous determinism, meaning that state evolution is strictly decoupled from wall-clock time so that rerunning a configuration with a fixed seed yields bit-identical logs independent of host performance. This lets us rerun a scenario with the exact same rain profile and determine whether an IDS alarm tracks the weather or only appears when we inject an attack. By doing so, artifacts become reproducible and attributable, allowing us to isolate an alarm's true cause instead of blaming simulator noise.

A. Motivation

Our work confronts three specific deficits in current simulation technology. For a simulator to be forensically sound, it must answer three questions: Does it reproduce realistic signal variance? Does it preserve deterministic timing? And can it expose causal ground truth?

1) Inadequate signal variance in synthetic data: Statistical intrusion detection works by learning the envelope of normal operation; that envelope is the detector's definition of reality. Most simulators approximate it using simple additive Gaussian

noise, but for safety-critical sensors, this is a poor substitute. Holder et al. [5] demonstrated that simple radar noise models fail to reproduce the coherent clutter distributions seen in physical sensors, and LiDAR attenuation behaves nonlinearly with rain intensity [6]. If 'normal' is modeled incorrectly, anomaly detection degrades into a calibration error, not a security guarantee [7]. An IDS validated on 'perfect attacks + white noise' may succeed in simulation, only to fail when faced with structured, weather-dependent noise in the real world.

2) Lack of temporal determinism: Forensic interpretation hinges on precise timing; it is often the only clue separating benign system jitter from a malicious attack. Yet many real-time simulators implicitly couple their simulation stepping to rendering performance. As the GPU is busy or CPU load fluctuates, scheduling jitter appears—jitter that is often indistinguishable from an attack's timing effects. This is a fundamental failure mode for any detector that monitors message Inter-Arrival Times (IAT). A forensic simulator must therefore guarantee tick-synchronous determinism, replaying scenarios bit-identically from a fixed seed, regardless of host system load.

3) Weak causal attribution without internal vehicle states: Without ground truth, it is difficult to determine whether an Autonomous Emergency Braking (AEB) event was triggered by a software fault, a phantom return from weather, or malicious CAN injection. To resolve this uncertainty, the simulation must expose its causal structure, not just the final output. IDS logic requires access to the evolution of internal physical states. Without it, a physically impossible bus trace can be labeled benign simply because the simulator lacks the vocabulary to express the violation. We therefore constrain bus signals by physically reachable state trajectories such as pressure build-up limits and drivetrain dynamics.

B. Problem Definition and Research Objectives

The central problem is the absence of a simulation environment built for strict forensic requirements: deterministic replay, physically meaningful timing, and controllable sensor degradation. To close this gap, we engineer the simulator around these constraints rather than real-time rendering. In this context, the actuation gap denotes the physical latency between a logical command, such as a brake request, and the corresponding mechanical response, such as pressure build-up; we use this irreducible latency floor as a discriminator to distinguish physically plausible actuation from instantaneous spoofing signals. The following three objectives are our minimum requirements for making IDS validation genuinely falsifiable.

1) Spectrally Consistent Vehicle Dynamics: We start with dynamics, because plausibility checks are only meaningful if the underlying physical state is credible. Game-engine kinematics often smooth out high-frequency transients that form the operational signature of a vehicle, such as drivetrain oscillations or shift shock. A gear shift, for example, leaves a characteristic burst in the torque signal that a detector can learn to recognize. If we validate against a model that suppresses these cues, the IDS learns to verify physics against a physics-free world. Accordingly, our dynamics model reconstructs internal states (e.g., engine speed, torque, hydraulic pressure) to ensure the resulting CAN signals exhibit spectral characteristics consistent with physical vehicle logs.

2) Parametric, Deterministic Sensor Degradation: Visually convincing data from neural renderers often comes at the cost of control, making counterfactual analysis impossible. An analyst must be able to ask: "Would this attack still succeed if rain intensity were 10 mm/h lower?" To answer this, we implement phenomenological Radar and LiDAR models where effects like attenuation and clutter become controlled levers for such counterfactual tests [8] [9]. This allows us to vary one condition at a time—rain, fog, or an attack—and know precisely which change triggered an IDS response.

3) Cross-Domain Validity of Security Mechanisms: Our final acceptance criterion is cross-domain performance. An intrusion detection system trained exclusively on *SimDAT-AV* data must achieve a comparable True Positive Rate (TPR) and latency robustness when evaluated against real-world datasets, such as ROAD [10] and RADIATE [11]. This demonstrates that the synthetic data is useful not just in theory, but in practice.

C. Overview

The remainder of this paper is structured as follows. Section II reviews existing work. Section III details the *SimDAT-AV* architecture and its synchronous lockstep mechanism. Section IV describes the cyber-physical modeling approach. Section V verifies simulation fidelity. Section VI evaluates forensic utility and IDS transferability. Finally, Section VII discusses limitations, and Section VIII concludes the paper.

II. Related Work

Validation methodologies rely on synthetic data to bridge the gap between nominal driving and edge cases. While datasets like nuScenes [12] or KITTI [13] provide high-fidelity sensor data, they often lack the intersection of adverse weather and active cyber-attacks. Recent work by Dološ et al. [14] emphasizes *Forensic Readiness* in autonomous mobility, but its validation implicitly assumes a simulation environment with deterministic timing and physically grounded signal generation. Existing approaches frequently trade forensic controllability for visual fidelity or statistical convenience. We therefore organize related work along three axes: temporal determinism, signal realism, and data generation paradigm.

A. Temporal Determinism and Reproducibility

Forensic analysis depends on deterministic reconstruction. Many standard simulation frameworks undermine this by tying simulation speed to rendering frame rates [2]. The resulting host-dependent jitter creates uncertainty over whether delayed CAN frames stem from an attack or from rendering load. A simulator that cannot guarantee bit-identical replay offers no stable baseline for timing-based IDS features.

B. Signal Variance and Environmental Noise

A second failure point occurs when 'normal' is mis-modeled. This happens frequently when sensor noise is approximated with

simple additive Gaussian distributions rather than structured, temporally correlated weather effects. Holder et al. [5] showed that basic radar noise models fail to reproduce coherent clutter, while Linnhoff et al. [6] found that LiDAR attenuation depends non-linearly on rain intensity.

C. Paradigms of Synthetic Data Generation

Forensic integrity also depends on the data generation paradigm itself. Attack injection without physically consistent signal propagation can produce traces that are protocol-correct but causally implausible. Frameworks such as SimuTack [15] and DriveFI [16] enable attack injection, but often focus on the logical layer rather than the physical propagation needed for forensic plausibility.

a) Statistical Models: Methods like CTGAN [17] scale well for tabular augmentation but cannot represent protocol timing. They are therefore useful for static risk assessment, but not for validating stateful timing-sensitive IDSs.

b) Implicit vs. Explicit Modeling: Neural simulation systems such as UniSim [18] and SHIFT [19] achieve high visual fidelity, but often rely on implicit latent representations that hinder counterfactual analysis. Forensic evaluation instead requires explicit parametric knobs so that a single changed factor can be linked to a changed IDS response.

III. Architecture for Forensic Determinism

Standard game engines operate on a heuristic that conflicts with forensic requirements: they prioritize visual fluidity over temporal precision. To keep the display smooth, an engine will often decouple physics steps or drop frames, effectively injecting host-dependent jitter into both the rendering and the network event stream. Under load, this makes it difficult to disentangle host-induced artifacts from malicious timing manipulations. To establish a forensically sound ground truth, we must remove the host system as a hidden variable. We achieve this by stepping outside the real-time paradigm and employing an architecture based on Synchronous Lockstep and Inverted Control to ensure the simulation behaves strictly as a deterministic function of its model parameters.

A. Inverted Control Strategy

Lockstep alone is insufficient if the simulator still owns the clock; the engine might still interpolate or skip work under load, forcing clients to chase a moving time base. We therefore invert the traditional control hierarchy. Where a standard setup has the simulator (e.g., CARLA) dictating the timeline, we assign timing authority to a central *Supervisor* module. This demotes the simulator to a passive backend. At any given tick, the Supervisor requests a world snapshot, executes all subsystem updates, and holds the clock until completion. The simulation advances only because computation for the current step, t_k, is finished, not because a certain amount of real time has passed. We utilize a fixed logical step of 20 ms (50 Hz) to align with standard Electronic Control Unit (ECU) cycle times.

1) Formal Scheduling Model: A fixed timestep is inadequate as long as subsystems can lag behind asynchronously. The core architectural challenge is preventing a slow module from being skipped or approximated when the host is under stress. We model the Supervisor as a state machine cycling through REQUEST, COMPUTE, and COMMIT states. The invariant guaranteeing determinism is the condition for advancing time: the system transitions to the COMMIT state if and only if every active client has explicitly acknowledged completion of the current step, t_k. We formalize this dependency as:

$$S_{next} = \text{COMMIT} \iff \forall c \in C_{clients} : \text{Ack}(c, t_k) \quad (1)$$

The practical effect of this barrier is that system overload increases the wall-clock time per tick but preserves the sequence and content of events within those ticks. Modules do not rely on wall-clock timestamps; the shared monotonic tick counter serves as the sole time base. If the host stalls, the virtual clock pauses rather than skipping ahead. This effectively converts overload into latency while maintaining causal correctness, as argued by Kopetz for real-time systems [20]. The empirical validation of this mechanism is presented in Section V.

B. Synchronous Execution Control and Virtual Arbitration

This barrier-based scheduling model necessitates a synchronous execution flow. The Supervisor coordinates sensors, the Virtual Forensic Bus, and recording modules through blocking synchronization. Crucially, this extends to network arbitration. To satisfy conflicting requirements (hardware compatibility vs. reproducibility), we introduce a Dual-Mode Interconnect Architecture:

- **Live Mode:** Utilizes standard Operating System (OS)-level SocketCAN interfaces. This preserves the stochastic behavior of the kernel scheduler, which is necessary when interfacing with physical ECUs (Hardware-in-the-Loop).
- **Forensic Mode:** Bypasses the OS kernel entirely using a user-space broadcast hub. This mode enforces strictly sequential packet delivery governed solely by the logical tick.

For the validation of forensic determinism presented in this paper, the system operates strictly in *Forensic Mode*. This ensures that packet ordering is defined by the simulation step, preventing event reordering and data loss that occur in asynchronous environments under adversarial workloads.

C. High-Performance Data Ingestion

With a deterministic timeline, the data serialization process itself becomes the next potential source of error. Forensic analysis depends on precise residuals—the mathematical differences between predicted and observed states. Text-based formats like CSV are unsuitable for this task because conversion between decimal strings and binary floating-point values is not guaranteed to be round-trip stable. This can introduce rounding artifacts that, in a residual-based IDS, could be misclassified as anomalies. To eliminate this risk, our setup mandates lossless floating-point precision and columnar access. We log directly

to Apache Parquet with Snappy compression. This ensures the storage layer cannot perturb the data and prevents logging from becoming a bottleneck. This setup satisfies two mandatory forensic criteria:

- **Binary Fidelity:** Sensor channels are stored as typed float32 columns, preserving IEEE-754 values without decimal conversion.
- **State Separation:** Ground-truth columns from the simulator API are explicitly segregated from perceived columns generated by the sensor model.

This allows any observed error to be attributed entirely to the sensor model or an injected attack, rather than to serialization artifacts.

IV. Modeling Cyber-Physical Systems

To validate an IDS, the simulation must expose not only the final vehicle trajectory but also the internal states of the electronic control units (ECUs) and the network arbitration layer. An anomaly often hides not in where the car moves, but in the causal chain that explains *how* it achieves that motion. We therefore model the system layer by layer, starting with the physics of actuation.

A. Virtual ECUs: Powertrain & Transmission

We move beyond simple kinematic bicycle models to include high-fidelity Virtual ECUs. As shown in Figure 1, we implement a backward-facing kinematics model because the scenario engine dictates the vehicle's target trajectory. By inverting physical causality, the model derives engine RPM and torque from vehicle velocity and current gear ratio. This inversion is critical for IDS validation, giving us a physically consistent reference signature against which CAN traces become falsifiable. Without it, physically impossible torque/RPM jumps can still produce the correct trajectory, leaving an IDS with no violation to detect.

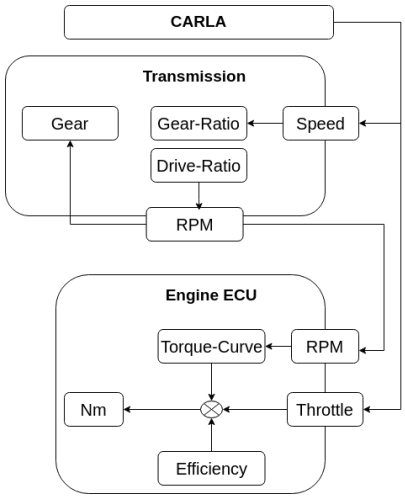

Figure 1: Data flow of the Virtual Powertrain Model.

1) Propulsion Dynamics: The engine speed ω_{eng} is derived kinematically from vehicle velocity and gear ratio. This kinematic link alone, however, leaves a gap that spoofing attacks can exploit. An attacker can craft a protocol-correct message that still violates the engine's feasible torque envelope. To close this gap, we add a second validation step using a parametric efficiency model. It synthesizes the necessary torque using a nonlinear lookup table based on engine speed and throttle position α. This forces the model to account for the physical cost of acceleration, making attacks that fake instantaneous torque changes detectable as violations against the engine's known performance map.

2) Transmission Logic: Standard simulations often rely on static shift maps, resulting in perfectly predictable gear changes. For a physics-aware IDS, however, such regularity is not to be smoothed away; it is part of the signature that helps separate plausible driver behavior from overly clean, spoofed signals. To replicate this, the transmission logic solves a cost minimization problem to select the optimal gear g^*:

$$g^* = \arg\min_{g \in G} \left(w_1 \cdot P_{eff}(g) + w_2 \cdot P_{smooth}(g) \right) \tag{2}$$

where P_{eff} represents fuel efficiency penalties and P_{smooth} penalizes frequent shifting, with weights w_1, w_2 calibrated to match observed shift rates in vehicle logs. The resulting behavior includes realistic gear oscillation and shift-shock artifacts, providing a richer baseline for an IDS to learn.

3) Hydraulic Brake Dynamics (Actuation Gap): In forensic reconstruction, physical delay is often a key discriminator. Kinematic models assume instantaneous deceleration, but real brakes are bound by fluid dynamics. We model brake hydraulics to enforce an irreducible actuation latency, making an instant-stop trace testably implausible. Simulating gradual pressure build-up in the master cylinder alongside fluid compression and valve flow limits creates this mandatory physical latency in the simulation loop.

B. Network Topology and SecOC

We simulate the CAN bus with non-destructive bitwise arbitration because many IAT-based detection features depend on the micro-delays that arbitration introduces, which simpler bus models erase. To support analysis of authenticated traffic, we integrate a *Secure Onboard Communication* (SecOC) model. A common failure mode in replay experiments is counter desynchronization, where volatile freshness counters drift across runs, causing Message Authentication Code (MAC) failures that look like attacks even when the traffic is benign. To fix this, we decouple the Freshness Value (FV) from volatile state and bind it directly to the simulation's monotonic tick, t_k:

$$FV(t_k) = \text{Hash}(\text{Seed} \parallel t_k) \tag{3}$$

By anchoring the FV to this rigid time-step, MAC verification becomes deterministic relative to the simulation clock, provided the replay preserves the exact tick sequence.

C. Physics-Aware Intrusion Detection

With this deterministic physical baseline established, we can deploy a specification-based IDS. The system enforces four rule sets that together cover the minimum needed for falsifiable plausibility: Periodicity (timing regularity), Range (field bounds), Burst (transient structure), and Kinematic Consistency (cross-signal coupling). This methodology adopts the reliability assessment principles proposed by Lohre et al. [21] for unmanned aircraft systems. Specifically, we implement their concept of Sensor Cross-Validation by correlating logically dependent states. Instead of relying on a single source of truth, we validate the trustworthiness of the CAN bus data by checking the coherence between interacting physical components. While the first three check for low-level violations, the kinematic rule serves as the central physical constraint against more sophisticated spoofing.

This final rule posits that the engine speed must remain mathematically coupled to the wheel speed and the current transmission ratio:

$$|\omega_{eng} - (\omega_{wheel} \cdot i_{gear})| \leq \epsilon \tag{4}$$

An IDS applying this rule can therefore detect attacks where an adversary injects a valid RPM signal (correct protocol, correct checksum) that contradicts the vehicle's physical velocity—an anomaly that purely digital firewalls would miss.

V. Verification of Simulation Fidelity

Before evaluating forensic utility, we must confirm that the simulation satisfies its two core requirements: bit-exact reproducibility (causal invariance) and physical plausibility at the dynamics level.

A. Bit-Exact Reproducibility

Forensic utility hinges on the guarantee that a scenario S executed with seed σ produces an identical set of artifacts across repeated runs, regardless of host computational load. To validate this, we executed the same urban scenario twice in sequence on the same host.

a) Experimental Setup:: Run A (Baseline) was executed under nominal system load with seed $\sigma = 42$. Run B (Stressed) used the same seed while the host CPU was saturated via `stress-ng` to induce wall-clock delays and thread-scheduling jitter.

We verified bit-exactness by calculating the SHA-256 checksums of the recorded sensor data streams (velocity, Global Navigation Satellite System (GNSS), CAN logs) and by computing the pointwise delta between the time-series.

b) Results:: As illustrated in Figure 2, despite the large difference in wall-clock execution time, the artifacts from Run B remained bit-exact to the baseline. Both runs produced the identical SHA-256 hash (see plot legend), confirming binary integrity. Moreover, the delta $\Delta = |Run_A(t) - Run_B(t)|$ remained strictly at 0.0 m/s for every simulation tick. This confirms that the proposed architecture (Synchronous Lockstep coupled with the Virtual Forensic Bus) successfully decouples the generation of forensic evidence from host performance, eliminating the "Observer Effect" common in real-time simulators.

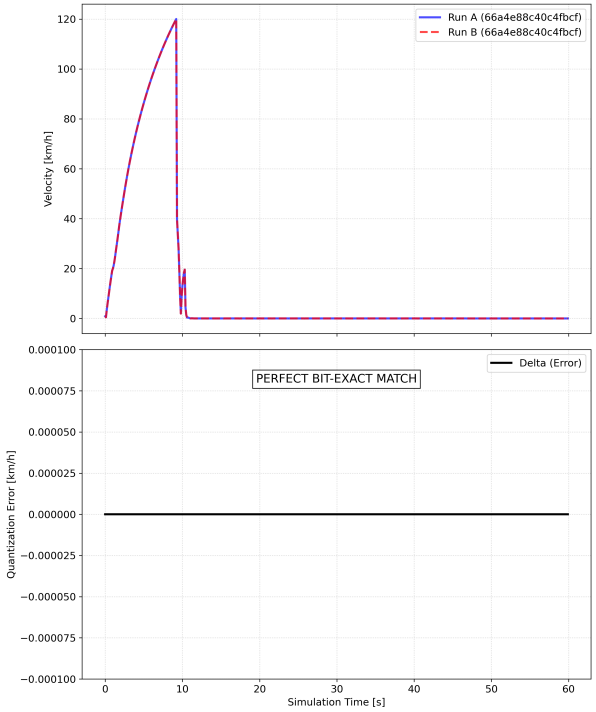

Figure 2: Verification of Causal Invariance.

B. Physical Validity: Dynamics & Latency

Beyond byte-level identity, the simulation must adhere to physical constraints to serve as a ground truth for an IDS. We focus on braking dynamics because they are both safety-critical and a common attack surface where naive spoofing attacks fail due to missing actuation latency. We validated the hydraulic brake model against real-world traces from $N = 19$ distinct braking events. The events were aligned at the brake-command onset to compare pressure build-up dynamics. The simulated pressure response tracks the statistical profile of the physical vehicle (Figure 3) with a Root Mean Square Error (RMSE) of 0.179, where pressure is normalized by the peak value of each event. This result rules out the "instant-stop" artifacts common in kinematic simulators and establishes a validated baseline where implausible braking traces (e.g., impossible pressure gradients) remain falsifiable.

VI. Forensic Utility & Security Evaluation

Having verified the simulation's determinism and physical validity, we now evaluate its specific utility for security analysis and intrusion detection.

A. Quantifying the "Temporal Gap": Async vs. Sync

To demonstrate why standard asynchronous simulation is insufficient for security validation, we conducted a test using a *Ghost Target Injection* attack. An adversary injected a false obstacle into the radar stream to trigger an Autonomous Emergency Braking (AEB) maneuver. We define reaction latency as

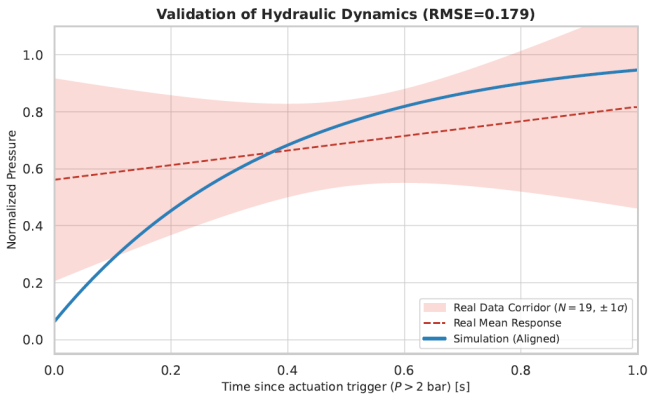

Figure 3: Statistical validation of the Hydraulic Brake Dynamics ($N = 19$ events).

the time from injection onset to the first AEB CAN command. We compared $N = 50$ runs in standard Asynchronous Mode against $N = 50$ runs in our Synchronous Lockstep Mode.

TABLE I: IMPACT OF EXECUTION MODE ON REACTION STABILITY ($N = 50$)

Metric	Async (Baseline)	Sync (Ours)
AEB trigger rate	100%	100%
Mean latency (μ)	17.53 s	17.53 s
Standard deviation (σ)	2.26 s	0.00 s
Range (min–max)	15.63–26.74 s	17.53 s

The results (Table I) reveal a critical forensic discrepancy. In the asynchronous baseline, scheduler-induced jitter caused a reaction time variance of $\sigma = 2.26$,s, with a spread of over 11 seconds between the fastest and slowest reaction to the identical attack script. In a forensic context, this noise makes it impossible to distinguish attack-induced delays from simulation lag.

In contrast, the synchronous mode collapsed this variance to a single deterministic outcome (no brake event). Consequently, the AEB command was consistently withheld in all 50 trials, proving that the erratic braking in the asynchronous mode was merely an artifact of scheduling jitter. This stability is a prerequisite for forensic analysis, as it ensures that any observed timing deviation in a trace is attributable to the attack vector, not the simulation engine's scheduler.

B. IDS Transferability (Sim-to-Real)

Finally, we assessed whether the physics-aware data improves the robustness of an IDS in real-world scenarios. We trained two versions of a specification-based IDS and evaluated them on the CAN spoofing scenarios within the ROAD dataset [10]. Two IDS variants are evaluated. The Baseline IDS is trained on idealized synthetic data with perfect timing behavior. In contrast, the SimDAT-AV IDS is trained on synthetic data generated by our framework, including physically modeled powertrain dynamics and environment-induced sensor noise. The kinematic consistency rule's tolerance (ϵ) was set for each IDS to the 99th percentile of benign timing residuals in its respective training set. On the ROAD evaluation set, the SimDAT-AV-trained model achieved a 100% True Positive Rate (TPR). Its tolerance threshold, derived from physically consistent training data, settled at 0.53,s, a value that closely matched the real vehicle's observed timing variability. The baseline model, in contrast, derived a tighter, artificial tolerance of 0.42,s from its overly clean training data. Consequently, it misclassified normal environmental noise in the ROAD logs as attacks (i.e., produced false positives). This indicates that integrating physical constraints in the data generation process prevents the IDS from overfitting to idealized simulation artifacts.

VII. Discussion & Limitations

While the synchronous lockstep architecture guarantees the evidential stability required for forensics, this stability comes at a cost. We trade nondeterminism for reduced execution throughput and higher sensitivity to model parameters.

A. The Cost of Determinism: Real-Time Factor

Lockstep inherently serializes execution: the clock cannot advance until the slowest component finishes its step. In complex scenarios, this often drops the Real-Time Factor (RTF) below 1.0. Consequently, *SimDAT-AV* is designed for *offline* tasks such as forensic replay and model training, rather than Hardware-in-the-Loop (HiL) integration with rigid wall-clock deadlines. For forensics, logical causal consistency matters more than wall-clock time.

B. Sensitivity to Physical Parameters

The utility of our kinematic consistency checks depends heavily on parameter accuracy. Such checks are only meaningful if the modeled friction coefficients and hydraulic time constants (Section IV) match the real vehicle and road surface. In our evaluation, we relied on static calibration. Future work must therefore focus on automated system identification from vehicle logs.

C. Scope of Attacks

Our current evaluation focused on spoofing and timing attacks. We demonstrated that enforcing physical plausibility is effective against these vectors. However, attacks on scene semantics, such as adversarial patches on traffic signs that confuse the camera classifier without violating kinematic constraints, remain out of scope for the current physics-aware IDS and require complementary defense layers.

D. Kernel Abstraction vs. Stack Fidelity

A deliberate architectural trade-off involves the abstraction level of the network stack. In "Live Mode" (using SocketCAN), the simulation includes stochastic OS-kernel behavior needed for hardware integration and interrupt-level timing tests, but this precludes reproducibility. In "Forensic Mode" (Virtual Bus), we bypass the kernel to achieve the zero-jitter results shown in Section V, at the cost of abstracting away kernel-space race conditions and driver-specific buffer overflows. *SimDAT-AV* therefore allows analysts to choose OS-fidelity or forensic precision, but not both simultaneously.

VIII. Conclusion and Future Work

For forensic replay, temporal determinism is a prerequisite. Our tests showed that standard best-effort architectures produce load-dependent timing shifts that can mimic cyber-physical attack signatures. We address this by decoupling simulation time from host performance through synchronous lockstep and a user-space virtual bus. Under identical seeds and induced CPU load, the resulting telemetry remained byte-identical. This makes each CAN frame a reproducible function of the model parameters rather than a byproduct of host scheduling. The resulting determinism enables a physically grounded IDS that can detect spoofing traces inconsistent with wheel-speed evolution, commanded torque, and gear state.

References

[1] K. Dološ, C. Meyer, A. Attenberger, and J. Steinberger, "Driver identification using in-vehicle digital data in the forensic context of a hit and run accident," *Forensic Science International: Digital Investigation*, vol. 35, p. 301 090, 2020. DOI: 10.1016/j.fsidi.2020.301090.

[2] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, *Carla: An open urban driving simulator*, [retrieved: March, 2026], 2017. arXiv: 1711 . 03938 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1711.03938.

[3] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "Airsim: High-fidelity visual and physical simulation for autonomous vehicles," *CoRR*, vol. abs/1705.05065, 2017, [retrieved: March, 2026]. arXiv: 1705.05065. [Online]. Available: http://arxiv.org/abs/1705.05065.

[4] F. Sezgin, D. Vriesman, D. Steinhauser, R. Lugner, and T. Brandmeier, "Safe autonomous driving in adverse weather: Sensor evaluation and performance monitoring," in *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2023.

[5] M. F. Holder, J. R. Thielmann, P. Rosenberger, C. Linnhoff, and H. Winner, "How to evaluate synthetic radar data? lessons learned from finding driveable space in virtual environments," in *FAS Workshop on Future Automotive Safety Technology*, 2020.

[6] C. Linnhoff, K. Hofrichter, L. Elster, P. Rosenberger, and H. Winner, "Measuring the influence of environmental conditions on automotive lidar sensors," *Sensors*, vol. 22, no. 14, p. 5266, 2022. DOI: 10.3390/s22145266.

[7] A. Khan, B. Malik, and C. Chen, "Intrusion detection system for can-bus in-vehicle networks based on statistical characteristics of attacks," *Sensors*, vol. 23, no. 3554, pp. 1–22, 2023.

[8] R. H. Rasshofer, M. Spies, and H. Spies, "Influences of weather phenomena on automotive laser radar systems," *Advances in Radio Science*, vol. 9, pp. 49–60, 2011.

[9] S. Teufel, G. Volk, A. von Bernuth, and O. Bringmann, "Simulating realistic rain, snow, and fog variations for comprehensive performance characterization of LiDAR perception," in *Proc. IEEE VTC-Spring*, 2022.

[10] M. E. Verma et al., "A comprehensive guide to can ids data and introduction of the road dataset," *PLOS ONE*, vol. 19, no. 1, pp. 1–32, Jan. 2024, [retrieved: March, 2026]. DOI: 10.1371/journal.pone.0296879. [Online]. Available: https://doi.org/10.1371/journal.pone.0296879.

[11] M. Sheeny et al., "Radiate: A radar dataset for automotive perception in bad weather," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 1–7. DOI: 10.1109/ICRA48506.2021.9562089.

[12] H. Caesar et al., "Nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020.

[13] 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*, 2012, pp. 3354–3361. DOI: 10.1109/CVPR.2012.6248074.

[14] K. Dološ et al., "Forensic readiness for autonomous mobility: The forensic incident recorder and information system concept," *Forensic Science International: Digital Investigation*, vol. 56, p. 302 044, 2026.

[15] A. Finkenzeller, A. Mathur, J. Lauinger, M. Hamad, and S. Steinhorst, "Simutack - an attack simulation framework for connected and autonomous vehicles," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, 2023, pp. 1–7. DOI: 10.1109/VTC2023-Spring57618.2023.10200555.

[16] S. Jha et al., "Ml-based fault injection for autonomous vehicles: A case for bayesian fault injection," in *2019 49th annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, IEEE, 2019, pp. 112–124.

[17] L. Xu, M. Skoularidou, A. Cuesta-Infante, and K. Veeramachaneni, "Modeling tabular data using conditional gan," in *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[18] Z. Yang et al., "Unisim: A neural closed-loop sensor simulator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 1389–1399.

[19] T. Sun et al., "Shift: A synthetic driving dataset for continuous multi-task domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 21 371–21 382.

[20] H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications*, 2nd ed. New York, NY: Springer, 2011, ISBN: 978-1-4419-8236-0.

[21] K. Lohre, H. Baier, L. Hardi, and A. Attenberger, "Towards reliable data in the scope of unmanned aircraft systems," *Forensic Science International: Digital Investigation*, vol. 53, p. 301 914, 2025, [retrieved: March, 2026], ISSN: 2666-2817. DOI: 10.1016/j.fsidi.2025.301914. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666281725000538.