

CADET: A Modular Platform for Evaluating Distributed Cooperative Autonomy in Connected Autonomous Vehicles

Pragya Sharma, Brian Wang, Mani Srivastava[†]
University of California Los Angeles
{pragyasharma, wangbri1, mbs}@ucla.edu

Abstract—Deep learning models are increasingly central to autonomous vehicle (AV) pipelines, yet their integration has traditionally followed a monolithic design where perception, planning, and control execute on a single onboard computer. This design overlooks the emerging paradigm of cooperative autonomy, where vehicles interact with roadside units (RSUs), edge servers, and cloud-hosted intelligence through vehicle-to-everything (V2X) connectivity. Cooperative perception and control improve safety and efficiency, but also introduce systems-level challenges: network latency, compute heterogeneity, and multi-tenant contention, all critically affect real-time decision-making. These challenges are further amplified by the increasing reliance on large foundation models, whose scale necessitates cloud deployment.

We present CADET (Cooperative Autonomy through Distributed Experimentation Toolkit), a modular platform for systematic and reproducible evaluation of distributed cooperative autonomy systems under realistic deployment conditions. CADET decouples the AV stack into composable modules that can be flexibly deployed across vehicles, infrastructure, and edge/cloud tiers. The framework integrates state-of-the-art models, incorporates trace-driven network and workload emulation, and provides synchronized model-, system-, and task-level instrumentation. Through V2V and V2I experiments, we show that distributed deployment choices fundamentally shape safety, with V2V intent packets outperforming cloud-based perception and RSU-assisted perception sustaining safety until overloaded by concurrent requests. Although designed for AV pipelines, CADET also supports dataset-driven experimentation, enabling systems and ML researchers to benchmark distributed inference workloads independently of full vehicle simulation. CADET is open source, with code and demo available at <https://nesl.github.io/cadet-web>.

I. INTRODUCTION

Autonomous vehicles (AV) increasingly rely on deep learning models for perception, planning, and control. As these models have grown in scale and capability, they have been integrated into platforms such as Apollo [1], AutoWare [2], and Pylot [3] to create comprehensive AI-driven AV pipelines. While modular in design, enabling researchers to swap components and experiment with novel algorithms, these platforms remain monolithic in deployment. They execute entire pipelines on single onboard computers, constraining experimentation to fixed hardware configurations and isolated vehicle intelligence.

[†]Author holds concurrent appointments as a Professor of ECE and CS (joint) at UCLA, and as an Amazon Scholar. This paper describes work performed at the UCLA, and is not associated with Amazon.

Recent advances in vehicle-to-everything (V2X) connectivity are reshaping the assumption of siloed AV stack execution. The emerging paradigm of cooperative autonomy (CA) allows vehicles to exchange sensor data, predictions, and control advisories with other vehicles, roadside units (RSUs), and edge/cloud infrastructure [4], [5]. Such cooperation has been shown to improve safety under occlusion [6], [7], improve traffic efficiency [8], and reduce collision probability in dense environments. However, realizing these benefits depends not merely on the algorithmic accuracy of individual deep learning models but on complex systems-level interactions. The same cooperative perception algorithm can yield different safety outcomes depending on computational resources and communication reliability.

The performance of cooperative autonomy is shaped by three key design-time choices: (i) model architecture selection which determines which models to use in the AV pipeline (ranging from lightweight CNN detectors [9] to large transformer-based predictors [10]); (ii) deployment topology which specifies where these models are executed (on-vehicle, at RSUs, or on edge/cloud servers, each with distinct latency–accuracy trade-offs [11]); and (iii) network characteristics which govern how these components communicate (wireless jitter, congestion, and multi-client interference that introduce non-deterministic delays and directly affect safety [5], [12]). Optimizing application outcomes therefore requires system–AI co-design: reasoning jointly about models, placement, and networks, rather than any single component in isolation.

Existing simulation platforms address only fragments of this challenge. Pylot provides modular autonomy pipelines but assumes single-vehicle execution without cooperation. OpenCDA [13], OpenCAMS [14], and Plexe [15] incorporate aspects of V2X and network modeling, but do not capture the holistic interaction of autonomy stacks with heterogeneous computing infrastructure. As a result, critical questions remain unanswered: How does network jitter affect cooperative perception when the AV pipeline is split across the RSU and the ego vehicle? Does safety degradation occur when edge or cloud servers incorporated in the loop face workload spikes? How should perception tasks be redistributed between vehicles and infrastructure as network conditions fluctuate? No platform currently enables systematic exploration of these systems-autonomy interactions that determine real-world cooperative driving safety.

Platform	V2X	Platform Heterogeneity	Network Emulation	Orchestration	System-Level Metrics	Modularity
Plexe [15]	●	○	◐	○	○	◐
Eclipse MOSAIC [16]	●	○	●	◐	◐	●
V2XVerse [17]	●	○	◐	○	○	●
OpenCAMS [14]	●	○	●	◐	○	●
OpenCDA [13]	●	○	◐	◐	○	●
VaN3Twin [18]	●	○	●	○	○	◐
CADET [ours]	●	●	●	●	●	●

TABLE I: Comparison of capabilities across V2X simulation platforms. CADET uniquely integrates platform heterogeneity, orchestration, system-level metrics, and modularity, distinguishing it from prior simulators.

This paper introduces CADET (Cooperative Autonomy through Distributed Experimentation Toolkit), a modular platform for reproducible evaluation of cooperative autonomy under realistic distributed systems constraints. Unlike monolithic AV stacks that execute fixed pipelines on single platforms, CADET disaggregates perception, planning, and control into composable modules deployable across vehicles, RSUs, edge nodes, and cloud servers. This architecture enables systematic exploration of distributed execution strategies and their impact on cooperative driving safety in V2V, V2I, and broader V2X scenarios.

CADET provides four core capabilities. First, a modular AV pipeline architecture supports flexible deployment of state-of-the-art models (YOLO [9], MPC [19]) and cooperation policies. Second, NetWaggle, a network emulation layer built on Mininet [20], injects realistic network dynamics modeling 4G/5G/WiFi variability and replayable traces. Third, distributed deployment abstractions enable seamless execution across heterogeneous resources, from resource-constrained vehicles to GPU clusters. Fourth, CADET incorporates an orchestration layer that, through a single configuration file, transparently manages model and policy selection, deployment, and network configuration. It further integrates multi-level instrumentation and records model-level (precision, recall, inference latency), system-level (queuing delay, bandwidth utilization, energy consumption), and task-level (time-to-collision, braking smoothness) metrics. Using such metrics, CADET makes it possible to trace how specific model, deployment, and network decisions translate into downstream application outcomes. Importantly, these components can be used individually or jointly, allowing not only the AV community, but also systems and ML researchers to evaluate their models in realistic cooperative settings.

In summary, this work makes the following contributions:

- We present CADET, the first modular platform that disaggregates the AV stack and supports distributed cooperative autonomy experiments with hardware-in-the-loop validation.
- We develop NetWaggle, a reusable emulation layer that captures realistic V2X delay distributions and device heterogeneity.
- We demonstrate CADET’s ability to compare cooperative autonomy policies, showing that V2V intent outperforms cloud-hosted perception under adverse network tails, and that RSU-assisted perception improves safety

until compute saturation.

- We release CADET as open-source software with documentation, benchmarks, and reference implementations to accelerate research at the intersection of distributed systems and cooperative autonomy.

II. RELATED WORK

A. Cooperative Autonomy

Cooperative autonomy has emerged as a systems-level paradigm in which vehicles and infrastructure share state, intent, and sensor data to overcome the inherent limits of single-agent autonomy. Isolated AVs remain constrained by occlusion, restricted sensor range, and reaction delays [8], whereas vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and broader vehicle-to-everything (V2X) cooperation extend perception and coordination. V2V enables cooperative adaptive cruise control (CACC) and platooning, yielding 10–30% fuel savings [21], [8]. Studies show that connected intersection managers can achieve 2–3× increase in intersection capacity and 90% reduction in delays [22].

However, these advantages hinge critically on the behavior of the underlying network and computational substrate. Empirical evaluations consistently show that latency, asynchrony, and packet loss degrade cooperative perception accuracy, with a sharp drop beyond 100 ms [23], [24]. Algorithms such as V2X-ViT try to address such challenges, yet temporal mismatches from communication delays continue to compromise perception outcomes [25]. Heterogeneous sensing and communication stacks exacerbate this, requiring bandwidth-aware scheduling and selective transmission to sustain performance under load [26], [27], [28].

B. V2X Simulation Platforms

A diverse ecosystem of V2X simulation platforms has emerged to enable cooperative autonomy. Network-centric simulators such as Veins [29] pioneered SUMO-OMNeT++ coupling, while Plexe [15] added realistic vehicle dynamics for platooning and VaN3Twin [18] integrated ray tracing to achieve 70% accuracy improvements. Multi-domain platforms such as Eclipse MOSAIC [16] enable runtime exchange across traffic, communication, and application domains, while OpenCAMS [14] synchronizes SUMO-CARLA-OMNeT++. Cooperative Driving Automation systems including V2XVerse [17] demonstrate 62% driving improvements through collaborative perception, and

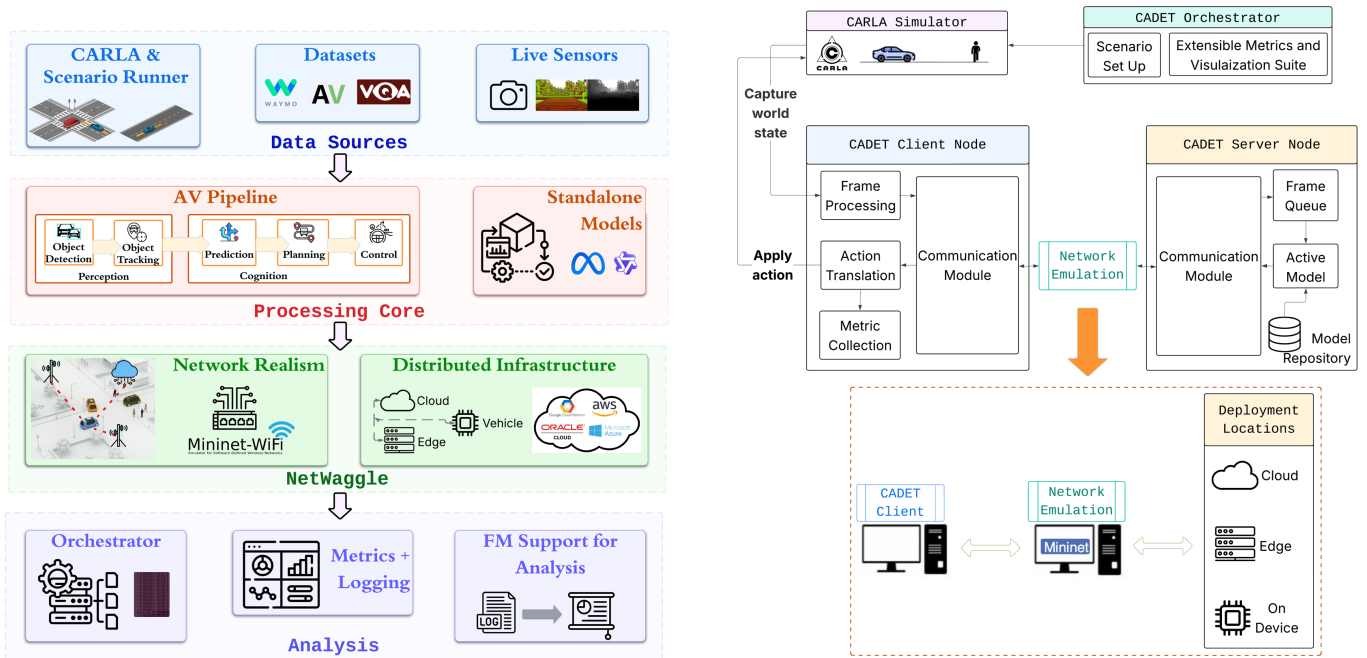


Fig. 1: **(left)** (a) CADET implements a modular four-layer architecture allowing components to be evaluated in isolation or integrated for end-to-end cooperative autonomy experiments. **(right)** (b) NetWaggle enables distributed inference under device heterogeneity and realistic network dynamics. It adopts a client–server design and mimics real-world network profiles.

OpenCDA [13] provides Python pipelines for rapid prototyping. These platforms excel within their domains, achieving high-fidelity network modeling and realistic traffic dynamics.

However, existing simulators converge on a critical limitation: they overlook cooperative autonomy as a distributed systems problem that requires compute platform heterogeneity modeling and system-level orchestration. They fail to capture how AV pipelines are partitioned across heterogeneous compute tiers (e.g., perception on edge GPUs, planning on vehicle), each with distinct performance profiles. Additionally, real deployments face scenarios where multiple vehicles compete for limited cloud GPU cycles and processing rates drop from 30 FPS on datacenter hardware to 5 FPS on embedded accelerators. CADET addresses this gap (shown in Table I) by providing multi-fidelity simulation and instrumentation that explicitly models computational resources across devices, hardware-in-the-loop testing for real deployment validation, and orchestration-aware metrics that capture end-to-end performance across distributed pipelines.

III. DESIGN GOALS

The central goal of CADET is to enable systematic evaluation of cooperative autonomy algorithms under realistic V2X communication and computational constraints. To achieve this objective, CADET must fulfill four key requirements:

Modularity. Cooperative autonomy research spans a wide spectrum of pipelines and scenarios, and CADET must support modularity at both the pipeline and component level. Researchers should be able to execute complete AV pipelines for end-to-end evaluation or isolate individual models (such as perception) for benchmarking in controlled settings. This

flexibility allows components to be interchanged or assessed independently without re-implementation of the entire stack.

Reproducibility. Experiments involving distributed computation and network dynamics require careful orchestration to ensure deterministic outcomes. CADET must ensure that identical network, platform, and scenario configurations yield consistent results across repeated runs, enabling validation and comparison across users.

Heterogeneity. Cooperative autonomy operates across asymmetric sensing and computational resources, from on-board processors to edge/cloud servers. CADET should capture the performance profiles of this continuum while also integrating diverse data sources into a unified workflow.

Multi-level Analysis. Evaluating cooperative autonomy requires visibility across multiple layers of the system. CADET must provide synchronized instrumentation spanning model-level metrics, system-level behavior, and application-level outcomes. This fine-grained view enables researchers to attribute the observed performance to specific algorithmic, communication, or resource-level factors.

IV. CADET ARCHITECTURE

CADET implements a four-layer architecture that systematically addresses the design goals (Sec. III) while maintaining the flexibility required for diverse cooperative autonomy research scenarios. The stratified design, shown in Fig. 1(a) establishes clear abstraction boundaries that facilitate both isolated component testing and integrated system evaluation.

A. Data Source Layer

The Data Source Layer provides scenario generation and sensing inputs integration via three interfaces:

CARLA simulator, datasets, and real-world sensor inputs.

At the core of this layer is CARLA [30], an open-source simulator widely adopted in the AV community. CARLA couples a physics-based vehicle dynamics engine with sensor-accurate rendering, enabling controlled evaluation of perception, planning, and control algorithms in urban environments. Augmented with the Scenario Runner library, it supports configurable V2X scenarios such as cooperative perception and platooning maneuvers. Researchers can instantiate road networks, spawn vehicles or pedestrians with scripted behaviors, and configure environmental conditions (e.g., rain, night driving, occlusion). Integration with Scenario Runner ensures modularity: new actors, pre-defined or cooperative behaviors can be introduced without altering the core simulation. This framework also supports failure-mode injection, such as dynamic changes in actors' behaviors or RSU disconnections, providing realistic stressors for cooperative autonomy scenarios.

CADET's Dataset Mode enables deterministic replay on pre-recorded inputs (e.g., Waymo [31], COCO [32]), maintaining full compatibility with network emulation and metrics collection for reproducible evaluation without real-time simulation. Finally, the data source layer can incorporate Live Sensor inputs, enabling direct evaluation with physical hardware (e.g., RGB camera, LiDAR) or hybrid setups that combine real and simulated streams.

B. Processing Core Layer

The Processing Core Layer embodies the modularity design goal through its configurable AV pipeline architecture and standalone model interfaces.

The AV Pipeline is decomposed into two primary stages: Perception and Cognition, each comprising multiple sub-stages that can be independently configured, swapped, and deployed across the computational continuum. For instance, object detection may be executed on a cloud server with high compute capacity, while trajectory planning runs on an RSU at the edge.

The Perception stage integrates object detection, tracking, and localization across local and V2V-shared observations, fusing asynchronous detections from multiple agents. The Cognition stage uses this enhanced awareness for prediction, planning, and control, enabling more accurate trajectory anticipation and safer motion plans.

Beyond the full AV pipeline, CADET provides an interface for evaluating Standalone Models in isolation. Users can benchmark perception, prediction, or emerging foundation models [33], [34] without running the entire AV pipeline, enabling fine-grained analysis of how network-induced delays or resource constraints affect specific algorithms.

C. NetWaggle

Distributed cooperative autonomy requires components of the AV pipeline to execute across heterogeneous devices (onboard processors, roadside units (RSUs), edge servers, and cloud platforms) connected through real networks. While

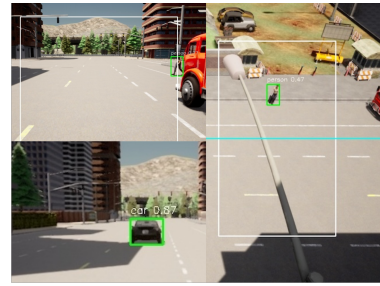


Fig. 2: Example CADET simulation captures: (a) (**bottom-left**) V2V scenario showing a platooning leader-follower interaction; (b) (**top-left**) ego vehicle camera view. (**right**) infrastructure camera view illustrating V2I perception.

such deployments are essential for realism, they are notoriously difficult to evaluate reproducibly. Actual cloud and network environments introduce variability from multi-tenancy, time-of-day fluctuations, congestion, and hidden prioritization policies [28]. These factors make it impossible to obtain deterministic outcomes: the same cooperative braking scenario might succeed in one run and fail in another purely due to transient network or resource conditions. To fulfill CADET's design goals of reproducibility and heterogeneity, we introduce NetWaggle, a network emulation layer that combines device heterogeneity with network realism in a controlled and repeatable setting.

Device heterogeneity. NetWaggle supports deployment across a continuum of computational platforms, from embedded vehicle-like devices to GPU-equipped RSUs and cloud servers. To enable this flexibility, CADET adopts a client-server architecture: the client hosts the simulation environment and manages vehicle actuation, while servers execute perception, prediction, or planning modules. This separation, shown in Fig 1(b), allows the AV pipeline to run entirely on the client, fully offloaded to the server, or distributed across multiple nodes. The architecture naturally extends to multi-vehicle scenarios, where each vehicle operates as a client while cooperating via shared infrastructure or peer-to-peer channels. To better reflect real deployments, CADET departs from CARLA's synchronous execution model and supports an asynchronous mode in which the simulation progresses even while awaiting delayed commands, mimicking the timing behavior of real-world V2X systems. Furthermore, CADET experiments can incorporate real devices, with modules deployed directly to local GPUs, Jetson boards, or cloud VMs, enabling hardware-in-the-loop testing alongside simulation. Data exchange between nodes is implemented over WebSockets [35], supporting both frame-by-frame and streaming modes to match the requirements of different cooperative autonomy pipelines.

Network realism. Alongside heterogeneous devices, cooperative autonomy depends critically on the properties of the connecting networks. Real-world V2X links are affected by network dynamics which influences safety outcomes [12]. Using uncontrolled networks directly in experiments undermines reproducibility. NetWaggle addresses this challenge by interposing Mininet-based emulation between distributed

components. Researchers can inject latency distributions, jitter patterns, packet loss rates, and bandwidth caps that match measured traces from 4G-LTE, 5G-NR, DSRC, or congested Wi-Fi links [36], or synthetically construct profiles to stress-test cooperative behaviors under adverse conditions. Mininet’s low overhead (typically <10 ms) ensures that emulation does not distort experimental outcomes while preserving repeatability. NetWaggle can connect both local heterogeneous devices and remote servers, enabling experiments that are simultaneously realistic and reproducible.

D. Analysis Layer

The `Analysis Layer` operationalizes evaluation by coordinating experiments, collecting metrics, and enabling reproducible analysis across distributed cooperative autonomy scenarios. It comprises three components:

`Orchestrator`. The Orchestrator manages the full life-cycle of an experiment, unifying configuration and execution. A single configuration file specifies scenario parameters, model variants, deployment mappings (cloud, onboard), and network traces. The orchestrator parses this configuration and launches all components in sequence: CARLA with Scenario Runner, client nodes (ego and cooperative vehicles), and distributed inference servers. Synchronization ensures that V2X communication channels are active before scenario execution begins. Runtime orchestration supports remote deployment through SSH-based execution, allowing servers to run on heterogeneous devices or cloud resources while remaining under centralized control. Batch execution mode enables systematic sweeps over parameters such as vehicle density, network delay profiles, or concurrent workloads.

`Metrics and Logging`. CADET provides synchronized metrics across three dimensions: (i) model-level metrics capturing algorithmic accuracy (e.g., cooperative object detection mAP, tracking consistency) and inference latency; (ii) performance outcomes measuring task-level behavior relevant to safety, including time-to-collision and braking efficiency; and (iii) system-level metrics recording CPU/GPU utilization, memory footprint, and energy consumption, characterizing the feasibility of deploying cooperative autonomy policies on diverse platforms. Logs are timestamped at critical points in the pipeline (frame capture, transmission, inference start/end, actuation) to attribute latency between computation, communication, and coordination overheads. Time synchronization libraries (e.g., NTP [37]) align distributed logs across clients, RSUs, and servers, supporting consistent analysis of end-to-end behavior. The logger outputs structured records (CSV/JSON) that can feed both real-time dashboards and post-hoc statistical evaluation, facilitating reproducible benchmarking across experiments.

`Foundation model interface`. As an extension, CADET can route logs through vision-language or language models [34] to provide semantic summarization and exploratory analysis. While not central to CADET’s architecture, this demonstrates extensibility toward foundation-model-based cooperative autonomy.

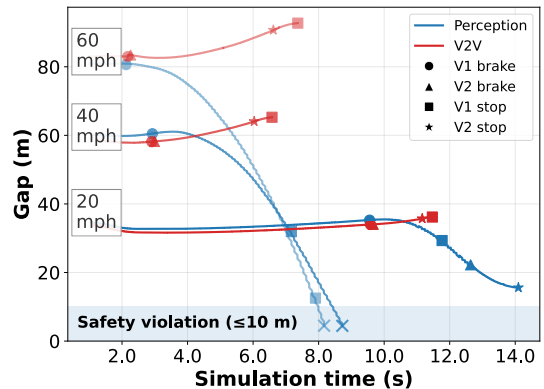


Fig. 3: (V2V) Leader-follower gap over time at different speeds under perception-only and V2V-based braking.

V. EVALUATION

We evaluate CADET through experiments that validate its design goals and demonstrate its unique capability to reveal how systems-level decisions fundamentally alter cooperative autonomy outcomes. Our evaluation establishes that CADET transforms complex distributed experimentation, that traditionally require thousands of lines of specialized code, into concise, reproducible studies accessible through simple configuration files. Each experiment isolates specific dimensions of the cooperative autonomy design space (e.g., model architecture, deployment topology, concurrent load) to attribute safety outcomes to individual factors rather than aggregate system behavior.

A. Implementation and Platform Validation

CADET comprises ~ 30 k lines of Python code implementing the architecture described in Sec. IV, with NetWaggle enabling distributed coordination across the vehicle-cloud continuum. Despite this implementation complexity, researchers interact with CADET through declarative YAML configuration files that specify scenario parameters, model deployments, and network profiles. For instance, a complete V2I perception-sharing experiment requires only 8-10 lines of code for scenario configuration.¹

To validate CADET’s modularity and reproducibility, we repeated identical cooperative autonomy scenarios 20 times under a fixed deployment configuration and network profile. Across repetitions, we observed $<5\%$ variance in safety metrics (e.g., time-to-collision and collision outcomes) and <3 ms standard deviation in end-to-end latency, confirming CADET’s ability to deliver reproducible results despite the inherent nuances of distributed execution.

We next demonstrate that CADET orchestrates cooperative autonomy scenarios, including V2V, V2I, and dataset-driven experiments, each designed to stress-test different dimensions of distributed autonomy.

B. V2V Cooperative Braking

We begin with a two-vehicle platoon scenario as a representative case of V2V safety-critical coordination (Fig. 2(a)).

¹Further implementation, configuration, and evaluation details can be found at <https://nesl.github.io/cadet-web>

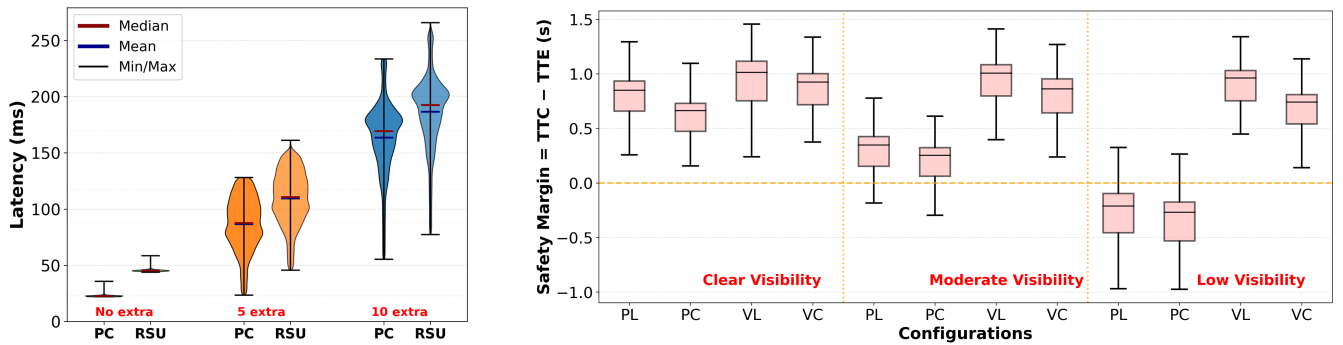


Fig. 4: V2I evaluation results. **(left)** (a) End-to-end latency distribution under increasing concurrent load (no extra, +5, +10 clients) for perception on cloud (PC) and RSU. **(right)** (b) Safety margin across visibility conditions for different pipelines: PL (perception-only local), PC (perception-only cloud), VL (V2I local), and VC (V2I cloud). Results show stable positive margins in clear visibility, with margins shrinking and turning negative under occlusion.

The leader executes a single emergency braking event, and the follower must respond quickly enough to maintain a safe inter-vehicle distance. This scenario captures a fundamental trade-off in cooperative autonomy: whether safety is better served by explicit V2V intent packets or by high-capacity perception models operating under networked inference.

Experimental Setup. The safety threshold for this experiment is set to 10 m, approximately two car lengths. The follower operates under two modes: (i) perception-only, using cloud-hosted YOLO11x inference deployed on a datacenter-class GPU (NVIDIA A5000), and (ii) V2V intent communication, where the leader transmits a braking packet directly. Cloud-based perception is modeled with a 30 ms baseline (p_{50}) communication latency consistent with 5G connectivity, with delay distributions extending to 100 ms at the tail (p_{99}). V2V packets are modeled after DSRC links with a baseline of 20 ms and a maximum tail of 50 ms. While Fig. 3 highlights YOLO11x and V2V for clarity, the full evaluation was run across all YOLO variants from nano to extra-large, with consistent patterns observed.

Results. Fig. 3 plots the inter-vehicle gap across speeds of 20, 40, and 60 mph. At 20 mph, perception-only control maintains a safe margin, but at 40 mph and 60 mph, the follower approaches within 10 m of the leader even before braking, violating the safety threshold. In contrast, V2V policy maintains a substantially larger gap, with margins exceeding 12 m even at higher speeds. The timing of brake initiation (denoted by ● and ▲) confirms that V2V followers react nearly instantaneously to leader intent, whereas perception-only followers accumulate additional delay from frame transmission and cloud inference.

Table II further quantifies collision outcomes under varying network delay percentiles. Perception-only control fails at p_{90} and p_{99} , with both larger models resulting in collisions despite their higher perception accuracy. The root cause is compounding delay: 30–50 ms inference latencies added to 100 ms tail delays exceed the safe response window. Moreover, larger models impose higher energy costs (up to 1.66 J per inference) without improving safety under adverse network conditions. In contrast, V2V communication remains collision-free across all delay profiles, as intent

Policy	Object Detector	Energy (J)	Network Delays			
			p50	p80	p90	p99
V2V	–	–	Green	Green	Green	Green
Perception (Cloud)	YOLO11n	0.73	Green	Green	Green	Green
	YOLO11s	0.76	Green	Green	Green	Green
	YOLO11m	0.92	Green	Green	Green	Green
	YOLO11l	1.27	Green	Green	Red	Red
	YOLO11x	1.66	Green	Green	Red	Red

TABLE II: V2V Collision outcomes across various deployment configurations. (Red : Collision, Green : No Collision)

packets are small and efficient to transmit.

CADET’s Impact. This experiment illustrates how CADET operationalizes reproducible evaluation of cooperative autonomy. With a single configuration file, CADET automatically records fine-grained timing traces, such as braking onset, stopping distance, and inter-vehicle gap, while simultaneously capturing system-level metrics including per-inference energy.

C. V2I-Assisted Braking

The second demonstration evaluates infrastructure-assisted perception in a braking scenario where roadside units extend the sensing capabilities of the ego vehicle beyond its line of sight, as shown in Fig. 2(b).

Experimental Setup. An ego vehicle travels at 20 mph when a pedestrian emerges from behind an occluding truck and the ego must brake in time to avoid a collision. We define the safety margin as the difference between the ego vehicle’s time to reach the collision area (TTC) and the pedestrian’s time to enter it (TTE), quantifying the temporal buffer available at detection. We use it as a post-hoc measure of timing safety, indicating whether detection and braking occurred with sufficient buffer to avoid overlap. We evaluate four configurations: (i) onboard perception using YOLO11x, (ii) cloud-hosted perception with additional 30 ms network delay, (iii) RSU-local V2I, where an infrastructure-mounted camera is co-located with the RSU and inference is performed locally (no additional communication latency), and (iv) RSU-cloud V2I, where the infrastructure camera is networked with the RSU, incurring an extra 30 ms hop before inference. Visibility conditions vary from clear to moderate

to low, simulating weather-related sensor degradation. We additionally examine scalability by adding concurrent clients (simulating additional load by 5-10 vehicles) generating requests to the RSU and Cloud server.

Results. Fig. 4(b) shows that perception-only policies achieve sufficient safety margins under clear visibility but deteriorates rapidly as conditions worsen; under low visibility, the median safety margin drops below zero, indicating unsafe operation even at 20 mph. In contrast, both RSU-local and RSU-cloud V2I configurations maintain positive margins across all visibility settings, demonstrating the advantage of infrastructure-based sensing for occlusion handling. Additionally, Figure 4(a) shows RSUs exceed 100 ms inference latency with 5-10 extra clients, and degrade gradually as load increases, becoming compute-bound beyond 10 vehicles. Cloud-based deployments, while elastically scalable, still exhibit long-tailed latency distributions: p_{99} latencies exceed 200 ms with 10 clients due to queuing under concurrent load. These results reveal complementary failure modes: perception-only policies fail under degraded visibility, while V2I policies fail under heavy compute load.

CADET’s Impact. This experiment highlights CADET’s ability to couple safety evaluation with scalability analysis in a single framework. CADET orchestrated multi-node deployments spanning vehicle, RSU, and cloud, while injecting additional load profiles.

D. Dataset Mode with Foundation Models

Our final demonstration uses CADET’s dataset mode to explore the feasibility of integrating vision-language models (VLMs) into cooperative autonomy pipelines. While current cooperative autonomy relies on specialized DNNs for perception and control, the growing capability of VLMs suggests future interest in semantic V2X communication, where vehicles or infrastructure may exchange natural language descriptions of complex scenes.

Experimental Setup and Results. We evaluate ten VLMs across three GPU servers (NVIDIA A6000, A100, and A16) using the VQA2 dataset. This evaluation serves as a proxy for cooperative autonomy scenarios where sensor or scene data is sent to a foundation model and queried in natural language, rather than being decomposed into narrowly scoped perception or planning tasks. Fig. 5 shows non-intuitive accuracy-latency profile that do not seem to improve with increasing model size or number of parameters. Further, inference latencies exceeding 200ms even for smaller models lie far beyond the deadlines of safety-critical coordination. We also measure the corresponding resource demands: 13B-scale models require more than 25GB of GPU memory and deliver throughput below 10 tokens/s, while some 3B–7B models remain efficient, sustaining >12 tokens/s with manageable memory footprints. These findings indicate that, although semantic queries are promising for higher-level coordination or advisory tasks, current models are too slow and resource-intensive for low-level safety decisions.

CADET’s Impact. By enabling dataset-driven evaluation, CADET makes it possible to benchmark standalone models

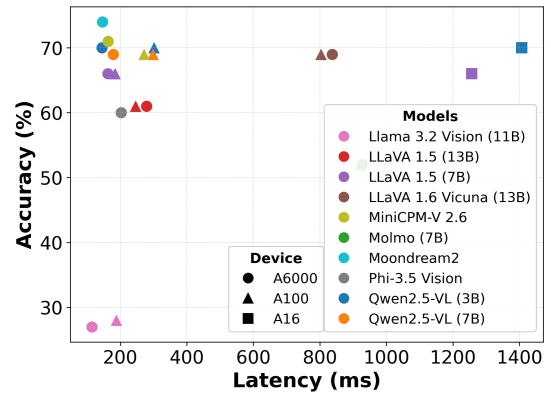


Fig. 5: Accuracy–latency tradeoffs for different VLMs across heterogeneous deployment platforms.

such as LLMs and VLMs alongside conventional DNNs using the same unified metrics and infrastructure. As these models continue to scale, CADET provides the infrastructure to systematically test when, where, and how such models could augment cooperative autonomy.

VI. CONCLUSION AND DISCUSSION

This work introduced CADET, a modular platform that disaggregates the AV stack into composable modules deployable across the vehicle–edge–cloud continuum. By coupling modular pipelines with NetWaggle’s reproducible network emulation and unified orchestration, CADET enables controlled evaluation of cooperative autonomy under heterogeneous deployment conditions. Our experiments show that safety outcomes hinge as much on *where* and *how* models are executed as on their algorithmic accuracy. V2V braking scenarios reveal that low-latency intent packets consistently outperform cloud-based perception despite the latter’s higher fidelity, while V2I experiments highlight that RSU-assisted perception preserves positive safety margins under occlusion yet collapses when overloaded by concurrent clients. Dataset-driven evaluations further demonstrate that large vision–language models, though attractive for semantic scene understanding, currently incur latencies and resource demands incompatible with safety-critical decision loops. Together, these results establish that cooperative autonomy cannot be advanced through algorithmic sophistication alone, but requires systematic exploration of the joint design space of models, placement, and networks.

Beyond these empirical findings, CADET contributes a methodological foundation for distributed robotics research. By reducing complex multi-node experiments to concise configuration files, the platform transforms previously ad-hoc engineering into reproducible systems experimentation. Its architecture generalizes beyond AVs: drone swarms coordinating via edge nodes, warehouse robots sharing world models through local clusters, and agricultural robots leveraging intermittent satellite connectivity, all face the same entanglement of computation, communication, and coordination. CADET makes these entanglements visible by providing synchronized model-, system-, and task-level instrumentation, enabling causal attribution from design choices to safety

outcomes. By releasing CADET as open source, we seek to democratize access to rigorous, systems-aware evaluation of cooperative autonomy, accelerating progress toward robotic systems that are not only intelligent in isolation but robust when distributed across heterogeneous, networked infrastructure. Future work includes extending evaluations to larger multi-vehicle deployments and incorporating additional safety metrics beyond time-to-collision.

ACKNOWLEDGMENT

This research was funded in part by DEVCOM ARL under the cooperative agreement W911NF1720196, and by the NSF under awards CNS-2211301 and CNS-2325956.

REFERENCES

- [1] Baidu Apollo, "Apollo: Open source autonomous driving." <https://github.com/ApolloAuto/apollo>. Accessed: 2025-09-15.
- [2] "Autoware." <https://github.com/autowarefoundation/autoware>. Accessed: 2025-09-15.
- [3] I. Gog, S. Kalra, P. Schafhalter, M. A. Wright, J. E. Gonzalez, and I. Stoica, "Pylot: A modular platform for exploring latency-accuracy tradeoffs in autonomous vehicles," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8806–8813, IEEE.
- [4] A. Caillot, S. Ouerghi, P. Vasseur, R. Boutteau, and Y. Dupuis, "Survey on cooperative perception in an automotive context," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14204–14223, 2022.
- [5] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17252–17262, 2022.
- [6] S. Aoki and R. Rajkumar, "V2v-based synchronous intersection protocols for mixed traffic of human-driven and self-driving vehicles," in *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pp. 1–11, IEEE.
- [7] I. Soto, M. Calderon, O. Amador, and M. Urueña, "A survey on road safety and traffic efficiency vehicular applications based on c-v2x technologies," *Vehicular Communications*, vol. 33, p. 100428, 2022.
- [8] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.
- [9] G. Jocher and J. Qiu, "Ultralytics yolo11," 2024.
- [10] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," in *Proceedings of the IEEE/CVF CVPR*, pp. 7577–7586, 2021.
- [11] H. Li, C. Hu, J. Jiang, Z. Wang, Y. Wen, and W. Zhu, "Jalad: Joint accuracy-and latency-aware deep structure decoupling for edge-cloud execution," in *2018 IEEE 24th international conference on parallel and distributed systems (ICPADS)*, pp. 671–678, IEEE, 2018.
- [12] R. Ernst, D. Stöhrmann, A. Bendrick, and A. Kostrzewa, "Application-centric network management-addressing safety and real-time in v2x applications," *ACM Transactions on Embedded Computing Systems*, vol. 22, no. 2, pp. 1–25, 2023.
- [13] R. Xu, Y. Guo, X. Han, X. Xia, H. Xiang, and J. Ma, "OpenCDA: An open cooperative driving automation framework integrated with co-simulation," in *Proceedings of the IEEE International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1155–1162, 2021.
- [14] M. U. Ahmad, A. Abrar, S. Dasgupta, and M. Rahman, "Opencams: An open-source connected and automated mobility co-simulation platform for advancing next-generation intelligent transportation systems research," *arXiv preprint arXiv:2507.09186*, 2025.
- [15] M. Segata, S. Joerer, B. Bloessl, C. Sommer, F. Dressler, and R. L. Cigno, "Plexe: A platooning extension for veins," in *2014 IEEE Vehicular Networking Conference (VNC)*, pp. 53–60, IEEE, 2014.
- [16] K. Schrab, M. Neubauer, R. Protzmann, I. Radusch, S. Manganiaris, P. Lytrivis, and A. J. Amditis, "Modeling an its management solution for mixed highway traffic with eclipse mosaic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, pp. 6575–6585, 2022.
- [17] G. Liu, Y. Hu, C. Xu, W. Mao, J. Ge, Z. Huang, Y. Lu, Y. Xu, J. Xia, Y. Wang, *et al.*, "Towards collaborative autonomous driving: Simulation platform and end-to-end system," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.
- [18] R. Pegurri, D. Gasco, F. Linsalata, M. Rapelli, E. Moro, F. Raviglione, and C. Casetti, "Van3twin: the multi-technology v2x digital twin with ray-tracing in the loop," *arXiv preprint arXiv:2505.14184*, 2025.
- [19] C. E. Garcia, D. M. Prett, and M. Morari, "Model predictive control: Theory and practice—a survey," *Automatica*, vol. 25, no. 3, pp. 335–348, 1989.
- [20] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks," in *2015 11th International conference on network and service management (CNSM)*, pp. 384–389, IEEE, 2015.
- [21] B. van Arem, C. J. van Driel, and R. Visser, "The impact of cooperative adaptive cruise control on traffic-flow characteristics," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, 2006.
- [22] J. Lioris, R. Pedarsani, F. Y. Tascikaraoglu, and P. Varaiya, "Platoons of connected vehicles can double throughput in urban roads," *Transportation Research Part C: Emerging Technologies*, vol. 77, 2017.
- [23] D. C. Nguyen, M. R. Azghadi, Y. Xia, Q.-L. Han, and S. Sun, "V2X cooperative perception for autonomous driving: Recent advances and challenges," *arXiv preprint arXiv:2310.03525*, 2024.
- [24] T. ElBatt, S. K. Goel, G. Holland, H. Krishnan, and J. Parikh, "Cooperative collision warning using dedicated short range wireless communications," in *Proceedings of the 3rd ACM International Workshop on Vehicular Ad Hoc Networks (VANET)*, pp. 1–9, 2006.
- [25] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2X-ViT: Vehicle-to-everything cooperative perception with vision transformer," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 107–124, 2022.
- [26] P. Sharma and M. B. Srivastava, "Impact of delays and computation placement on sense-act application performance in iot," in *MILCOM 2023-2023 IEEE Military Communications Conference (MILCOM)*, pp. 133–138, IEEE, 2023.
- [27] P. Sharma, B. Wang, X. Ouyang, R. Nanayakkara, B. Balaji, P. Tabuada, and M. B. Srivastava, "Towards a performance-driven device-edge-cloud relationship," in *Proceedings of the 26th International Workshop on Mobile Computing Systems and Applications*, pp. 125–125, 2025.
- [28] P. Sharma, H. Qiu, and M. Srivastava, "Cloud is closer than it appears: Revisiting the tradeoffs of distributed real-time inference," in *2025 34th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–9, IEEE, 2025.
- [29] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins: The open source vehicular network simulation framework," in *Recent advances in network simulation: the OMNeT++ environment and its ecosystem*, Springer, 2019.
- [30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*, pp. 1–16, PMLR, 2017.
- [31] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2446–2454, 2020.
- [32] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740–755, Springer, 2014.
- [33] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang, *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.
- [34] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.
- [35] I. Fette and A. Melnikov, "The websocket protocol," tech. rep., 2011.
- [36] B. Coll-Perales, M. C. Lucas-Estañ, T. Shimizu, J. Gozalvez, T. Higuchi, S. Avedisov, O. Altintas, and M. Sepulcre, "End-to-end v2x latency modeling and analysis in 5g networks," *IEEE Transactions on Vehicular Technology*, vol. 72, no. 4, pp. 5094–5109, 2022.
- [37] D. L. Mills, *Computer network time synchronization: the network time protocol*. CRC press, 2006.