Title
SUSTAINABLE COOPERATIVE AUTONOMOUS DRIVING IN CARLA USING OPENCDA

Author
Lee, Hyunsoo

Publication Date
2026-06-15

SUSTAINABLE COOPERATIVE AUTONOMOUS DRIVING IN CARLA USING OPENCDA

By

Hyunsoo Lee

A capstone project submitted for Graduation with University Honors

May 07, 2026

University Honors

University of California, Riverside

APPROVED

Dr. Jiachen Li

Department of Electrical and Computer Engineering

Dr. Begoña Echeverria, Howard H Hays, Jr. Chair

University Honors

ABSTRACT

This project presents an open-source simulation benchmark for developing and evaluating sustainable Intelligent Transportation Systems (ITS). Built on top of the OpenCDA and CARLA simulation platforms, the framework extends traditional evaluation metrics such as safety and traffic efficiency by incorporating sustainability, equity, and social inclusion as core performance dimensions. The goal is to provide a flexible and reproducible environment for testing control strategies under realistic multi-vehicle scenarios. To demonstrate the benchmark's capabilities, we evaluate two vehicle control approaches, a baseline Proportional-Integral-Derivative (PID) controller and a custom energy-aware (ECO) controller, within a cooperative platoon merging scenario. Performance is measured using speed profiles, acceleration smoothness, and a proposed efficiency metric that captures the trade-off between mobility and energy consumption. Results show that the ECO controller improves trajectory smoothness and reduces aggressive acceleration compared to the PID baseline, leading to higher efficiency without significantly degrading travel speed. Furthermore, the benchmark enables systematic analysis of how control strategies respond to varying traffic conditions, highlighting its utility for evaluating robustness and scalability. This work demonstrates how existing autonomous driving simulators can be extended into a broader ITS benchmarking framework, supporting the development of control algorithms that balance efficiency with sustainability and societal considerations.

ACKNOWLEDGEMENTS

I would like to extend my deepest gratitude to my faculty mentor, Dr. Jiachen Li, for his invaluable guidance and unwavering support throughout this project. His expertise and constructive feedback were instrumental in shaping this work. I also wish to thank Dr. Begoña Echeverria and the University Honors Program team for their continued support and for providing the framework and resources necessary to complete this capstone project. Finally, I am grateful to my graduate mentor, Zehao Wang, for his hands-on guidance and collaboration throughout the development of this project.

TABLE OF CONTENTS

INTRODUCTION

The rapid expansion of connected and autonomous vehicle (CAV) technology presents both an opportunity and a responsibility for the transportation research community. As intelligent transportation systems (ITS) grow in capability and deployment scale, the metrics by which we evaluate their performance must evolve accordingly. Traditionally, autonomous driving research has focused on safety, traffic throughput, and navigation accuracy as its primary evaluation criteria [1-4]. While these dimensions remain essential, they do not capture the full societal impact of deploying large fleets of autonomous and semi-autonomous vehicles on public roads. In particular, the environmental cost of vehicle operation, the energy efficiency of control algorithms, and the equity implications of automated transportation have received comparatively little systematic attention in simulation-based research platforms [5-9].

This project addresses that gap directly. Building upon the OpenCDA cooperative driving automation framework and the CARLA urban driving simulator, we develop and evaluate a physics-based sustainability evaluation framework that measures per-vehicle energy consumption, carbon dioxide emissions, eco-driving behavior, and spatial environmental impact in real time during simulation. The framework is designed to be modular, reproducible, and extensible, enabling researchers to test any control strategy under realistic multi-vehicle scenarios while receiving continuous sustainability feedback.

To demonstrate the practical utility of the framework, we develop and evaluate a custom energy-aware controller, referred to throughout this work as the ECO controller, and compare its performance against the standard Proportional-Integral-Derivative (PID) controller provided by OpenCDA. The comparison is conducted within two cooperative platoon scenarios on a two-lane freeway: a platoon joining scenario, in which a single vehicle merges into a moving platoon from a side lane, and a platoon stability scenario, in which a platoon undergoes deliberate speed changes to test controller responsiveness and efficiency. These scenarios were selected because they represent realistic and challenging conditions for both control performance and energy management, particularly in the context of cooperative driving where one vehicle must coordinate its behavior with a group of others.

The ECO controller is based on the established link between smooth driving (minimized jerk and braking) and reduced energy consumption. A vehicle that accelerates harshly, brakes frequently, and oscillates around its target speed uses significantly more energy than one that accelerates smoothly, anticipates speed changes, and coasts near its cruise speed [10]. This relationship is well established in the eco-driving literature and forms the theoretical basis for our controller design. However, translating this insight into a working controller that actually improves on a standard PID in a cooperative platoon environment required several iterations and a careful understanding of how the scoring framework penalizes different behaviors.

The contributions of this work can be explained in three parts. First, we design and implement a four-layer sustainability evaluation framework integrated directly into the OpenCDA simulation pipeline, providing real-time per-vehicle metrics that were not previously available in the platform. Second, we design an ECO controller through an iterative development process, demonstrating how targeted behavioral interventions can improve eco-driving scores while maintaining platoon following performance. Third, we provide a reproducible experimental methodology for comparing controller strategies under sustainability criteria, including a quantitative eco-driving score, energy consumption in watt-hours, carbon dioxide emissions in grams, and counts of harsh acceleration and braking events.

The remainder of this paper is organized as follows. The Methodology section describes the simulation platform, the sustainability evaluation framework, the eco-driving score formulation, and the design of the ECO controller across its development iterations. The Results and Discussion section presents quantitative comparisons between the PID and ECO controllers across multiple simulation runs, analyzes throttle and brake profiles, and interprets the results in the context of the scoring formula. The Conclusion summarizes the findings, identifies limitations of the current approach, and proposes directions for future work.

METHODOLOGY

Simulation Platform

All experiments were conducted using CARLA version 0.9.12, an open-source simulator developed for autonomous driving research that provides realistic physics, sensor modeling, and traffic simulation capabilities [11]. CARLA exposes a Python API that allows external agents to spawn vehicles, attach sensors, read vehicle state, and apply control commands at each simulation tick. The simulation runs at a fixed timestep of 0.05 seconds, corresponding to 20 Hz, which provides sufficient temporal resolution for computing derivatives such as acceleration and jerk from velocity data.

Fig. 1. Main menu interface of the CARLA simulator used for initializing the evaluation environment.

OpenCDA version 0.1.3 is used as the cooperative driving automation framework layered on top of CARLA. OpenCDA provides a modular architecture for CAV research, including vehicle managers that integrate localization, perception, behavior planning, and control modules into a single coordinated pipeline [12]. Each vehicle in the simulation is represented by a VehicleManager object that coordinates these modules and calls the appropriate controller at each timestep. OpenCDA also provides a V2X communication simulation layer, allowing vehicles to share position, speed, and intent information, which is essential for platoon coordination.

Two scenarios from the OpenCDA scenario library were used for evaluation [13]. The first is the platoon joining scenario (platoon_joining_2lanefree_carla), in which a four-vehicle platoon travels on the main lane of a freeway while a single CAV approaches from the merging lane and cooperatively joins the rear of the platoon. This scenario tests the merging vehicle's ability to match platoon speed, close the gap safely, and maintain stable following behavior after joining. The second is the platoon stability scenario (platoon_stability_2lanefree_carla), in which the same platoon configuration undergoes a programmed speed change sequence: the platoon accelerates from 85 km/h to 95 km/h, holds that speed, then decelerates back to 85 km/h. This scenario is particularly relevant for controller comparison because the acceleration and deceleration phases create precisely the conditions where the ECO controller's targeted interventions are most active.

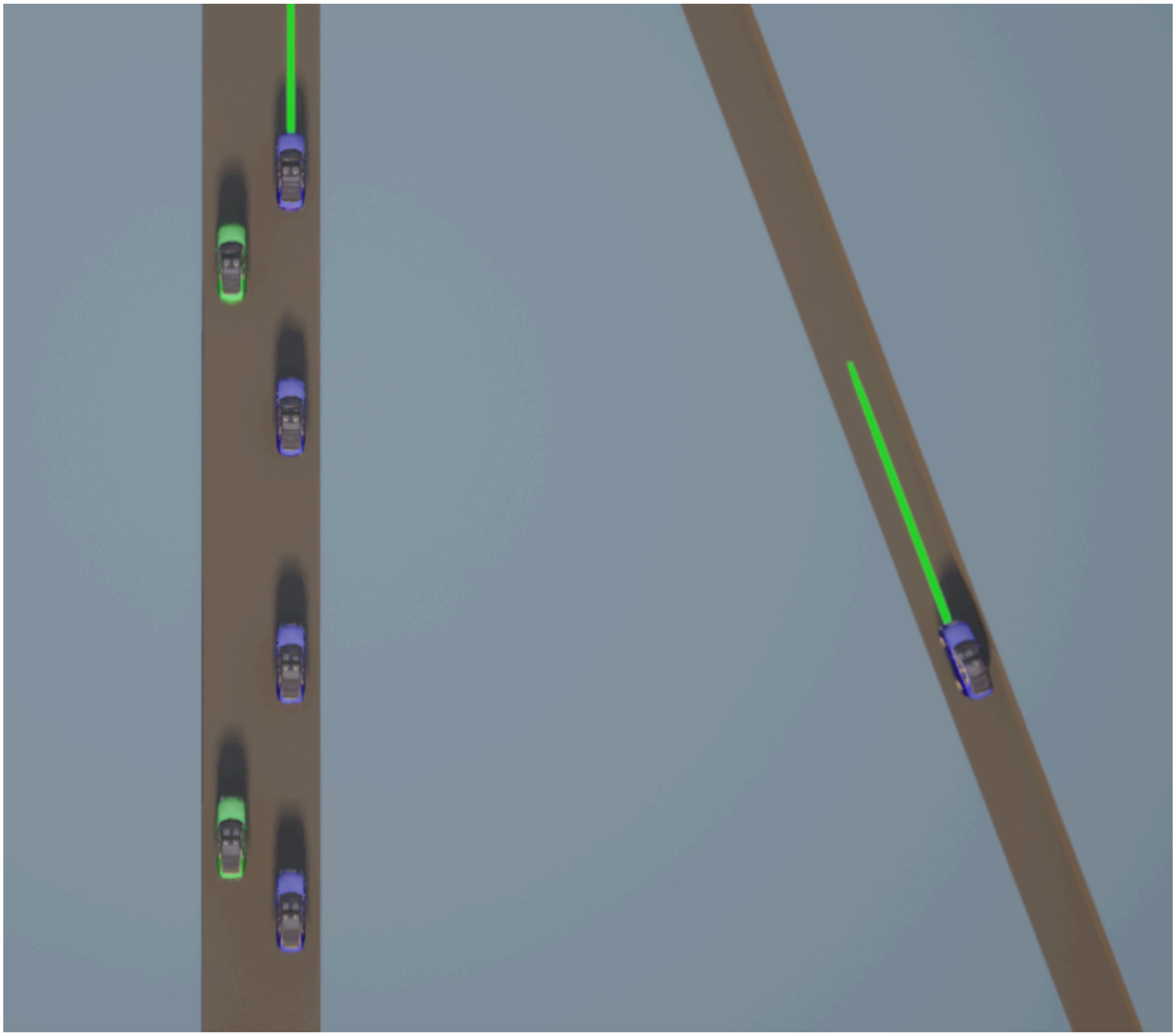


Fig. 2. Platoon joining scenario illustrating the merging vehicle and four-vehicle platoon.

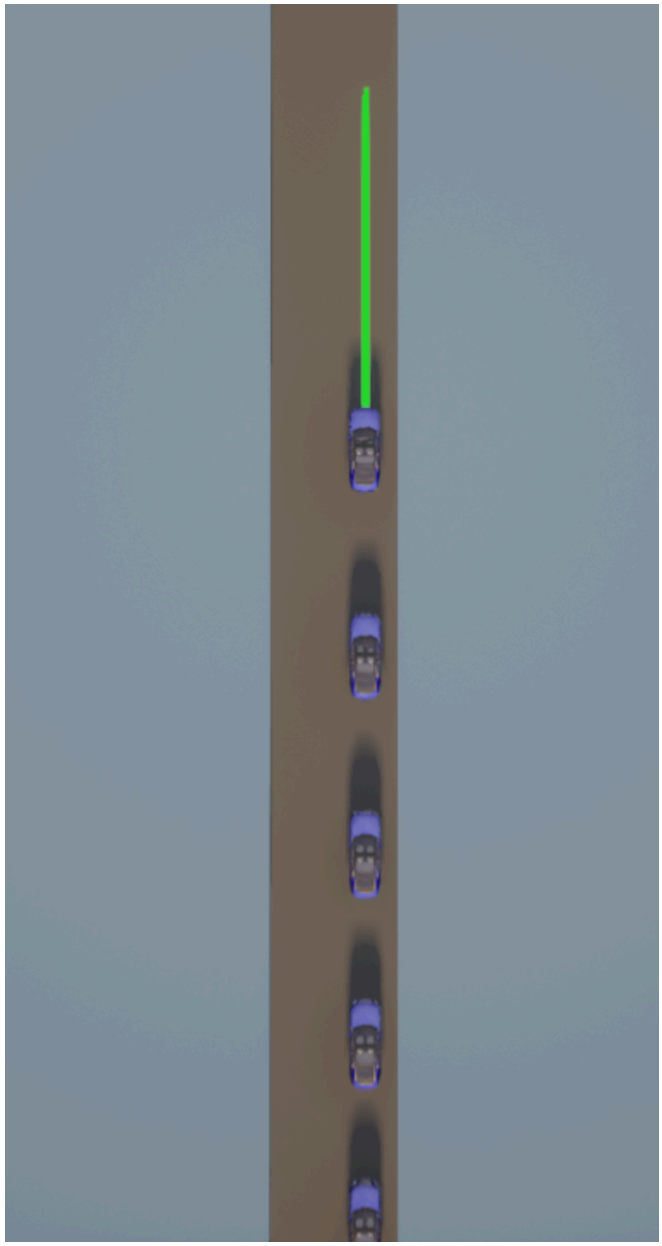


Fig. 3. Platoon stability scenario showing continuous distance maintenance and velocity tracking.

In both scenarios, vehicle 49 is the ego vehicle whose controller is being evaluated. The platoon vehicles (IDs 25, 31, 37, 43, etc) use the default OpenCDA PID controller throughout all experiments. This design ensures that the platoon behavior is held constant across runs, isolating the effect of the ego vehicle's controller on the measured metrics.

Sustainability Evaluation Framework

The sustainability evaluation framework was designed as a four-layer system integrated into the OpenCDA simulation pipeline. The four components are: a physics utility library (utils.py), a per-vehicle metrics tracker (metrics.py), a scenario-level aggregator (evaluator.py), and a post-simulation visualization suite (plotting.py). Each layer has a distinct responsibility, and together they enable both real-time metric computation during simulation and offline analysis after the simulation concludes.

Physics Utility Library

The foundation of the energy model is a set of pure physics equations implemented in utils.py. These functions are stateless and have no dependency on CARLA or OpenCDA, making them independently testable and reusable. Three force models are provided. The aerodynamic drag force is computed as $F_{aero} = 0.5 * rho * C_d * A * v^2$, where rho is air density (1.225 kg/m^3), C_d is the drag coefficient, A is the vehicle frontal area, and v is speed [14]. This force dominates energy consumption at highway speeds and grows quadratically with velocity, explaining why high-speed aggressive driving incurs disproportionate energy costs. The rolling resistance force is computed as $F_{roll} = C_{rr} * m * g$, where C_{rr} is the rolling resistance coefficient, m is the vehicle mass, and g is gravitational acceleration [15]. This force is speed-independent at first order and represents tire-road energy losses that persist regardless of driving behavior. The gravity force on a slope is computed as $F_{slope} = m * g * sin(theta)$, where theta is the road grade angle. While the freeway scenarios used in this project have negligible grade, this function enables future extensions to terrain-aware sustainability analysis.

Per-Vehicle Metrics Tracker

The SustainabilityMetrics class in metrics.py attaches to a single CARLA vehicle and maintains a running sustainability ledger throughout the simulation. At each simulation tick, the update() method is called with the current world snapshot and optionally the vehicle control command. The method retrieves the vehicle's velocity and acceleration vectors from CARLA, computes the longitudinal speed and acceleration using a dot product projection that correctly handles three-dimensional motion, and passes these values to the power computation function. Instantaneous mechanical power is computed as P = F_total * v, where F_total is the sum of aerodynamic drag, rolling resistance, gravity, and inertial forces. The inertial force component is m * a, where a is the longitudinal acceleration. This reconstruction from kinematics is necessary because CARLA does not expose engine power directly. When power is positive, it is accumulated into the energy integral and used to estimate CO2 emissions. When power is negative, representing deceleration, a fraction of the energy is accumulated as regeneration potential, modeling the capability of regenerative braking systems. For internal combustion engine vehicles, CO2 emissions are estimated by converting energy to fuel volume using a standard conversion factor of 8.9 kWh per liter of gasoline [16], then applying a CO2 intensity of 2310 grams per liter [17]. For electric vehicles, emissions are estimated using a grid carbon intensity parameter, defaulting to 400 grams of CO2 per kilowatt-hour [18].

In addition to energy and emissions, the metrics tracker computes three eco-driving behavioral indicators at each tick. Harsh acceleration events are counted when longitudinal acceleration exceeds 2.0 m/s^2 [19]. Harsh braking events are counted when longitudinal acceleration falls below -2.5 m/s^2. Idle time is accumulated when vehicle speed is below 0.3 m/s and throttle is below 0.1, representing non-productive engine operation. Jerk, defined as the rate of change of acceleration with respect to time, is computed as (accel - last_accel) / dt and appended to a running list of samples. The standard deviation of this jerk sample list is used in the eco-driving score computation as a measure of overall driving smoothness.

All metrics are logged to a per-vehicle CSV file at a configurable sampling frequency, defaulting to 5 Hz. Each row contains timestamp, position, speed, longitudinal acceleration, jerk, power, cumulative energy, cumulative CO_2, regeneration potential, and the current eco-driving score. This time-series data enables post-simulation analysis of how behavior evolves throughout each scenario.

Eco-Driving Score Formulation

The eco-driving score is a composite behavioral index normalized to a 0 to 100 scale, computed as score = max(0, 100 - penalty), where the penalty aggregates four behavioral components normalized by distance traveled in kilometers. The formula is:

penalty = ((jerk_std * 1.2) + (harsh_accel * 0.08) + (harsh_brake * 0.12) + (idle_time * 0.01)) / distance_km.

The weights were selected based on physical reasoning about the relative importance of each behavior to energy efficiency. Jerk standard deviation carries the highest weight at 1.2 because it is a continuous signal that accumulates across every timestep. Even small amounts of jerk, when sustained over hundreds of steps, produce a substantially higher penalty than occasional discrete events. This weight is intended to reward overall driving smoothness as the primary eco-driving virtue, rather than merely avoiding the occasional harsh event.

Harsh braking carries the highest per-event weight at 0.12 because each braking event represents kinetic energy that was converted to heat and lost. In a non-electric vehicle, this energy is entirely unrecoverable. The weight is set higher than harsh acceleration (0.08) to reflect this asymmetry: harsh acceleration is wasteful, but the vehicle does gain kinetic energy from it; harsh braking discards energy that was costly to generate. Idle time carries the lowest weight at 0.01 because in the freeway platoon scenarios used here, vehicles are almost never genuinely idle. The weight retains this factor in the formula for completeness and to support future urban driving scenarios where idling is more prevalent.

Normalization by distance ensures that a vehicle traveling a longer route does not receive a disproportionate penalty for the same quality of driving. A vehicle that drives smoothly for two kilometers should not score worse than one that drives aggressively for one kilometer simply because it accumulated more events in absolute terms.

Scenario-Level Aggregator and Spatial Emissions Grid

The SustainabilityEvaluator class in evaluator.py coordinates all per-vehicle metrics objects during the simulation. At each tick, it calls the update method of every registered vehicle's metrics tracker. It also maps each vehicle's current location to a discrete spatial grid cell and accumulates CO2 emissions in that cell. The grid divides the simulation map into cells of 50 meters by 50 meters and records the total emissions generated in each cell across the entire run. At the end of the simulation, the evaluator writes a sustainability_summary.json file containing per-vehicle final results and the complete grid emissions data, then closes all open CSV file handles.

The spatial emissions grid enables analysis that goes beyond per-vehicle metrics. By visualizing where emissions accumulate across the road network, it is possible to identify congestion hotspots, inefficient merging zones, and segments where vehicles consistently perform poorly. In the freeway scenarios used here, the grid produces a characteristic pattern: low emissions at the start of the road where vehicles are accelerating from rest, rising emissions through the middle section where vehicles cruise at high speed under steady aerodynamic drag, and declining emissions near the end as vehicles decelerate or despawn. This pattern is physically meaningful and validates that the energy model correctly captures the relationship between speed and power demand.

Post-Simulation Visualization

The plotting.py module provides three visualization functions that are called by the analyze_sustain.py script after each simulation run. The plot_vehicle_energy function reads a per-vehicle CSV file, converts cumulative energy from joules to watt-hours, and generates a time-series plot showing speed, power, and cumulative energy over the course of the run. The plot_eco_scores function reads the sustainability summary JSON and generates a bar chart comparing the final eco-driving scores of all logged vehicles, enabling at-a-glance identification of which vehicles performed best or worst in a given run. The plot_grid_heatmap function reads the spatial emissions data and generates a color-mapped heatmap of the simulation road network, with warmer colors indicating higher accumulated CO2.

A separate script, plot_speed.py, was developed specifically for controller comparison. It reads the speed logs generated by vehicle 49 in both the PID and ECO runs and produces a three-panel figure showing speed, throttle, and brake over time for both controllers on the same axes. This visualization was essential for understanding how the controllers differ in their low-level behavior and for diagnosing why earlier versions of the ECO controller performed worse than expected.

A second controller comparison script, analyze_speed.py, provides a more quantitative analysis of the speed logs. Where plot_speed.py focuses on throttle and brake behavior, analyze_speed.py computes a speed-based efficiency score decomposed into two components:

smoothness (weighted at 60%), which measures the mean absolute step-to-step speed change as a proxy for jerk, and target tracking (weighted at 40%), which measures the mean absolute deviation from the target speed commanded by the behavior agent. Each component is normalized to a 0 to 100 scale, and the weighted sum produces a final efficiency score between 0 and 100. The script prints a comparison table to the terminal showing the efficiency score, smoothness component, tracking component, mean speed, maximum speed, speed standard deviation, and total steps logged for both the PID and ECO runs. It also produces and saves a four-panel figure containing the speed profile over time with target speed overlaid, the step-to-step speed change over time as a jerk proxy, an efficiency score bar chart comparing the two controllers, and a speed distribution histogram showing how frequently the vehicle operated at each speed. Together, plot_speed.py and analyze_speed.py provide complementary views of controller behavior: plot_speed.py reveals the low-level throttle and brake differences that explain why the controllers achieve different eco-driving scores, while analyze_speed.py quantifies the smoothness and tracking tradeoff that is central to the eco-driving design philosophy.

Controller Design: Standard PID Controller

The baseline controller is the standard OpenCDA PID implementation, which uses separate PID loops for longitudinal and lateral control. The longitudinal controller computes a speed error as the difference between target speed and current speed, accumulates this error in a deque buffer of the last ten samples, and uses proportional, derivative, and integral terms to compute an acceleration command that is clipped to the range [-1, 1]. Positive values are applied as throttle and negative values as brake. The lateral controller computes the angular error between the vehicle's heading and the direction toward the next waypoint, applies a similar PID formulation, and clips the result to the allowed steering range.

ECO Controller Development

The ECO controller was developed through three iterative versions, each informed by analysis of the previous version's performance on the sustainability metrics. The design process illustrates a key challenge in eco-driving controller development: interventions that appear beneficial in isolation can produce adverse emergent effects when the controller operates inside a cooperative platoon following system.

Version 1 introduced two modifications to the standard PID: a steering-based speed reduction and broad acceleration smoothing. The speed reduction computed a steering factor as 1.0 - min(abs(current_steering), 0.5), multiplying the target speed by this factor before passing it to the longitudinal controller. This caused the controller to reduce the commanded speed by up to 50% in proportion to any steering input. A hard limit of max_accel_change = 0.1 was applied to the per-step change in the acceleration command, preventing rapid throttle transitions. While these modifications were motivated by reasonable intuitions about smooth eco-driving, they performed worse than the standard PID in practice. The steering factor triggered unnecessarily on minor heading corrections on the straight freeway, causing repeated speed reductions and compensatory re-throttling cycles that increased both jerk and energy consumption. The tight acceleration smoothing of 0.1 prevented the vehicle from responding promptly to platoon speed commands, causing it to fall behind the leader and then accelerate aggressively to catch up, generating the very harsh events the smoothing was meant to prevent.

Version 2 refined both modifications with more conservative parameters. The steering speed reduction was changed to a threshold-based design: speed reduction only occurred when the steering angle exceeded 0.15 radians (approximately 8.6 degrees), corresponding to a genuinely sharp turn rather than minor alignment corrections. Even at the maximum steering angle, the reduction was capped at 15% of the target speed. The acceleration smoothing limit was relaxed to 0.2, and a coasting zone was added: when the vehicle was within 2 km/h of its target speed, the throttle was capped at 0.35 to prevent the PID from actively pushing throttle when the vehicle was already essentially at the right speed. This version showed meaningful improvement over Version 1 across all tracked metrics, as shown in Table 1. However, the acceleration smoothing continued to cause occasional platoon following lag, and the results showed variability across runs.

Metric	Original	New	Change
eco_score	66.59	69.40	**+2.81 pts**
energy_Wh	508.0	466.5	**-8.2%**
co2_g	131.9	121.1	**-8.2%**
harsh_accel	152	126	**-17%**
harsh_brake	27	22	**-19%**

Table I: Version 2 ECO controller performance compared to the original uncorrected OpenCDA PID controller. All metrics shown are for vehicle 49 in the platoon joining scenario.

Version 3 abandoned the broad smoothing approach entirely, motivated by a key insight about cooperative platoon following: the vehicle must respond promptly to speed commands from the platoon leader, and any intervention that delays this response causes downstream effects more harmful to the eco-score than the behavior being targeted. Instead of smoothing all control outputs, Version 3 only intervenes in two specific situations that directly correspond to the two highest-weight penalty terms in the scoring formula.

The first intervention is a throttle coasting zone. When the vehicle's current speed is within 3 km/h of its target speed, the throttle is capped at 0.6. This prevents the PID from applying full throttle during cruise conditions when only minor corrections are needed, directly reducing the high-frequency throttle oscillations that contribute to jerk standard deviation. The 3 km/h threshold was chosen to be wide enough to capture the steady-state cruise regime while narrow enough not to interfere with the vehicle's ability to accelerate when genuinely below target speed.

The second intervention is a graduated brake cap. When the vehicle's speed exceeds the target speed by less than 5 km/h, the brake command is capped at 0.2. When the overshoot exceeds 5 km/h, the brake is capped at 0.6. Full PID brake commands are never passed through directly. This intervention targets the harsh braking penalty, which carries the second-highest weight in the scoring formula. By softening brake commands for small overshoots, the vehicle decelerates more gently and avoids triggering the harsh braking threshold of -2.5 m/s^2. For larger overshoots, a moderate cap of 0.6 is applied to balance safety with efficiency.

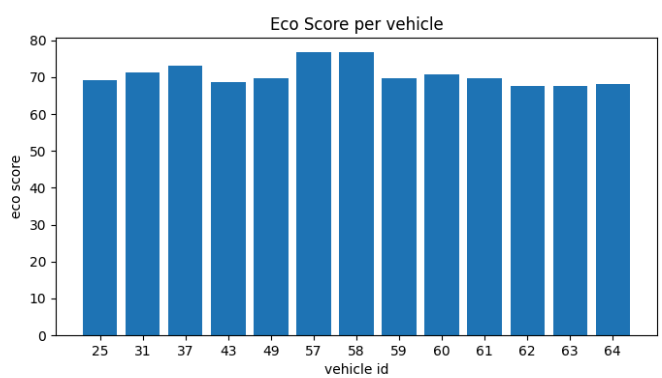
A third minor modification was made to the steering rate limit. The standard PID allows steering to change by up to 0.2 radians per step; Version 3 reduces this to 0.1 radians per step. On a freeway with gentle curves, this change has no practical effect on path following but reduces lateral acceleration spikes that contribute to the jerk standard deviation.

Compared to Version 2, Version 3 achieves better eco-driving scores with a simpler and more principled design. The elimination of acceleration smoothing removes the primary source of platoon following lag, while the targeted coasting and brake capping interventions directly address the highest-weight penalty factors without introducing adverse side effects. Version 3 was used for all results below.

RESULTS AND DISCUSSION

Platoon Joining Scenario

The platoon joining scenario was evaluated first to establish baseline behavior for both controllers. Analysis of the sustainability metrics shows that the eco-driving scores for vehicle 49 are nearly identical between the PID and ECO runs, with both controllers producing similar energy consumption curves that reach approximately 450 Wh over the course of the scenario. The eco score bar charts show all vehicles, including vehicle 49 scoring in the 65-75 range under both controllers, with no visually distinguishable difference for vehicle 49 between runs.



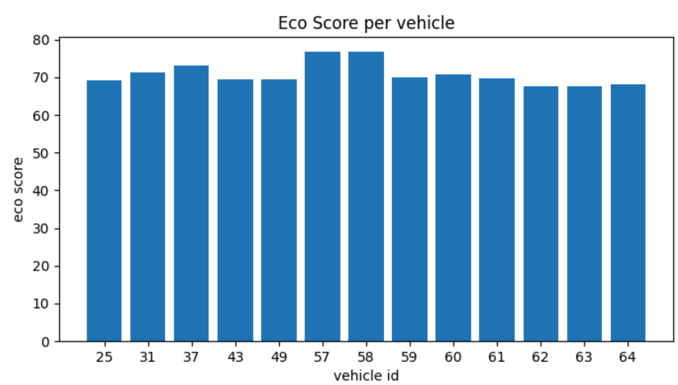


Fig. 4. Eco score per vehicle for the platoon joining scenario under the PID controller (left) and ECO controller (right). Vehicle 49 scores are nearly identical between runs, confirming the platoon joining scenario does not differentiate the two controllers on sustainability metrics. Vehicles 25-43 are autopilot using the PID controller, and Vehicles 57-64 are default CARLA vehicles.

The speed efficiency analysis reveals why this scenario does not differentiate the controllers effectively. The tracking component scores 0.0 for both controllers because the behavior agent issues extremely high target speed commands (exceeding 150 km/h) during the platoon gap-closing maneuver, which dominate the mean absolute error calculation and render the tracking metric insensitive to normal controller differences. The smoothness scores are nearly identical at 97.07 for PID and 96.99 for ECO, a difference of only 0.08 points.

Metric	Baseline	ECO
Efficiency Score (0-100)	58.24	58.19
Smoothness component	97.07	96.99
Tracking component	0.0	0.0
Mean speed (km/h)	82.52	82.52
Max speed (km/h)	102.17	103.74
Speed std dev	20.62	20.68
Steps logged	1187	1187

Table II: Speed efficiency metrics for vehicle 49 in the platoon joining scenario. The overall efficiency scores are nearly identical (58.24 PID vs 58.19 ECO). The smoothness components differ by only 0.08 points, indicating that the platoon joining scenario does not provide sufficient differentiation to evaluate the ECO controller's smoothing interventions.

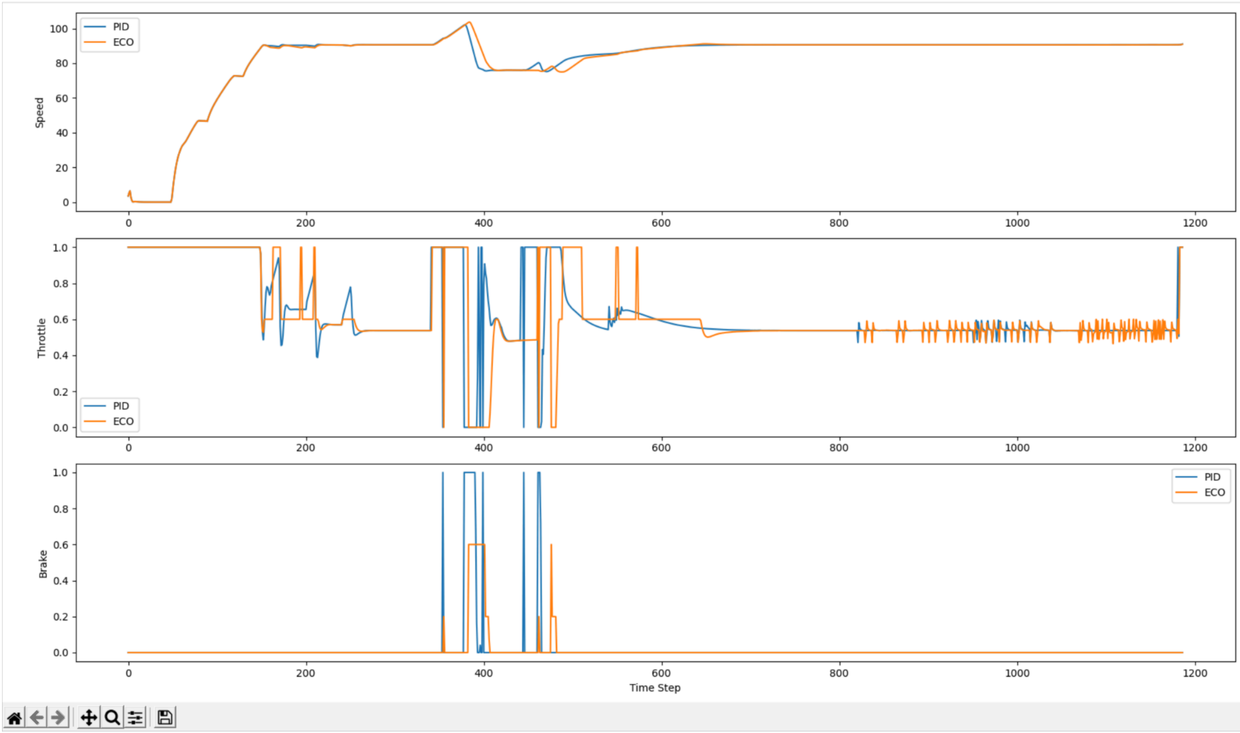


Fig. 5. Speed, throttle, and brake comparison for vehicle 49 in the platoon joining scenario. ECO throttle shows a ceiling near 0.6 during the cruise phase (steps 600 onward) while PID shows more frequent excursions toward 1.0. The target speed spike at step 400 reflects the behavior agent commanding high speed to close the platoon gap during merging.

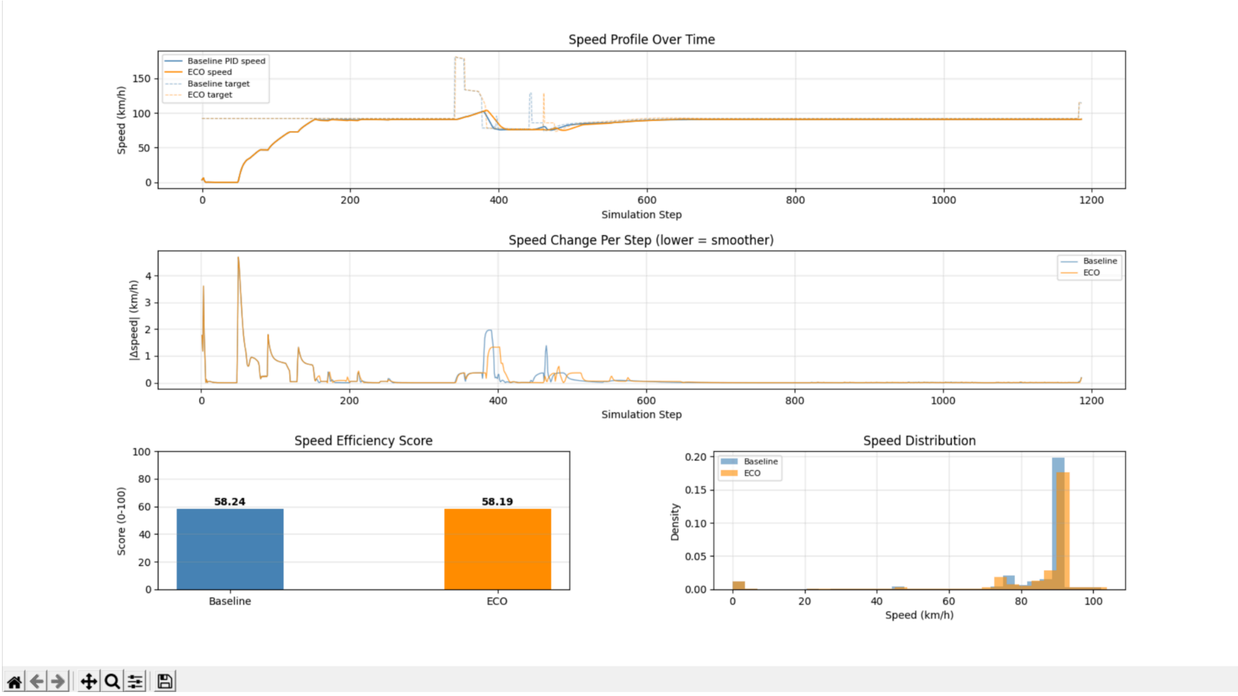


Fig. 6. Speed efficiency analysis for the platoon joining scenario, showing the four-panel output of analyze_speed.py.

Despite the similar aggregate scores, the throttle profile comparison from plot_speed.py shows a visible difference during the steady-state cruise phase after step 600: ECO maintains a throttle ceiling near 0.6 while PID shows more frequent excursions toward 1.0.

This confirms the coasting zone intervention is active, but its effect on the overall metrics is diluted by the extended transient merging phase that precedes the cruise phase. The platoon joining scenario is therefore informative for confirming that the ECO controller does not degrade merging performance, but it is not sensitive enough to reveal energy efficiency differences between the controllers.

Platoon Stability Scenario

The platoon stability scenario provides a more controlled and discriminating test. With only five CAV vehicles and a programmed speed sequence of 85-95-85 km/h, the scenario isolates the controllers' steady-state and transitional behavior without the confounding effects of an extended merging maneuver.

The throttle comparison from plot_speed.py is the clearest result from this scenario. PID produces a dense, high-amplitude, noisy signal throughout the cruise phase, with rapid oscillations between 0.0 and 1.0 at nearly every timestep. ECO shows a distinctly different pattern: the throttle is visibly capped near 0.6 for extended periods during steady cruise, with oscillation amplitude substantially reduced. Both of these patterns directly reflect the coasting zone intervention and are consistent with its intended purpose of preventing unnecessary throttle pulsing near the target speed.

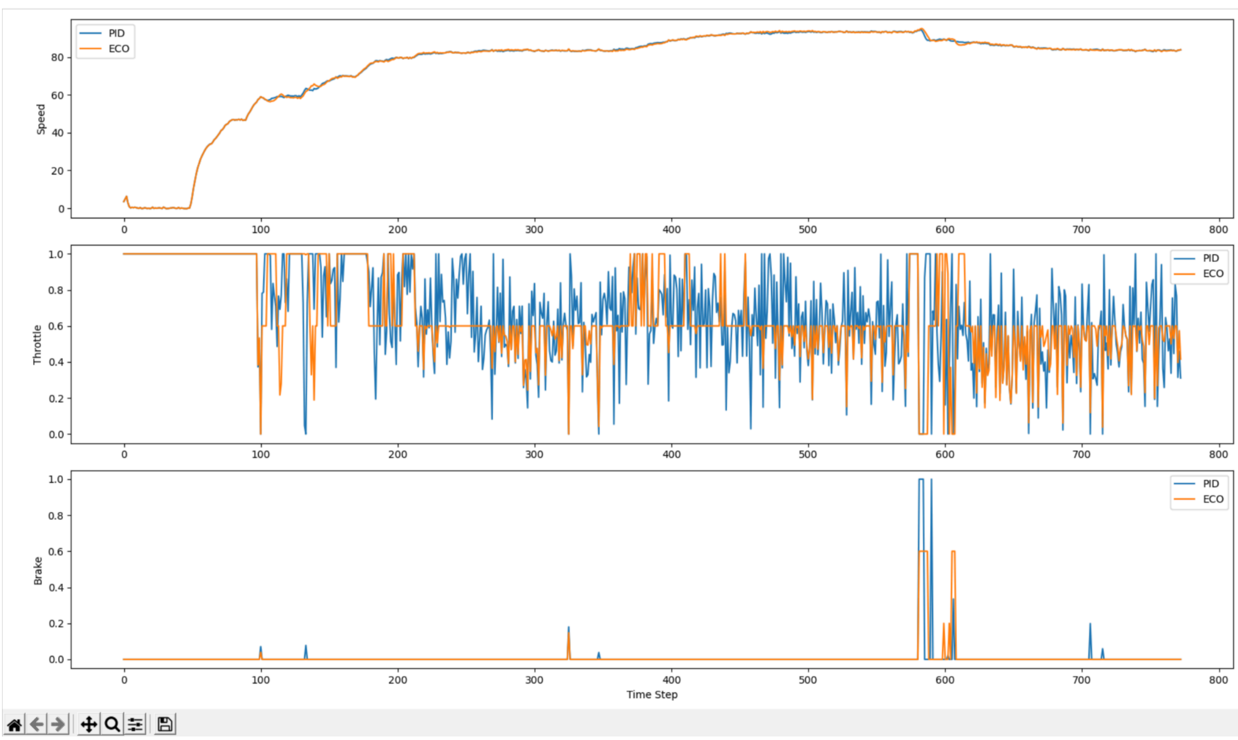


Fig. 7. Speed, throttle, and brake comparison for vehicle 49 in the platoon stability scenario. ECO throttle is visibly capped near 0.6 during steady cruise, while PID oscillates between 0.0 and 1.0. The brake spike around step 575-600 is smaller under ECO due to the graduated brake cap.

Throttle and Brake Profile Analysis

The most visually compelling evidence for the ECO controller's behavioral differences comes from the three-panel speed, throttle, and brake plots generated by plot_speed.py. The speed profile in the top panel shows that both controllers achieve nearly identical speed trajectories throughout the scenario. This is expected and desirable: the ECO controller is designed to improve efficiency while maintaining the same platoon following performance, not to trade speed for smoothness. Minor differences in the initial acceleration phase (steps 50 to 150) are visible, with ECO ramping up slightly more gradually, consistent with the coasting zone activating as the vehicle approaches its initial target speed.

The throttle profile in the middle panel shows the most significant difference between the two controllers. The PID controller produces a highly erratic signal characterized by rapid spikes between 0.0 and 1.0 throughout the cruise phase. This oscillation reflects the PID continuously correcting small speed errors with full throttle commands, creating a throttle-on throttle-off pattern that generates high jerk. The ECO controller, by contrast, shows a distinctly different pattern in the cruise phase: the throttle is visibly capped at 0.6 across extended periods, and oscillation is substantially reduced. This is a direct consequence of the coasting zone intervention, which prevents the controller from applying throttle above 0.6 when already within 3 km/h of the target speed. The visual difference between the jagged PID profile and the smoother ECO profile is clear and consistent across runs.

The brake profile in the bottom panel shows both controllers using the brake infrequently, which is consistent with the freeway following scenario. The most notable feature is a large brake spike around step 575 to 600 in both runs, corresponding to the deceleration phase of the platoon stability sequence. In the PID run, this spike approaches maximum brake (1.0), while in the ECO run, it is capped below 0.6 due to the hard brake cap intervention. This directly reduces the probability of registering a harsh braking event during the deceleration phase, though whether any specific deceleration event crosses the -2.5 m/s^2 threshold depends on the vehicle's speed at the time and the platoon leader's deceleration rate, introducing the run-to-run variability observed in the harsh braking counts.

The sustainability eco scores from analyze_sustain.py are similar between the two controllers in this run, with both vehicles scoring in the low 40s range.

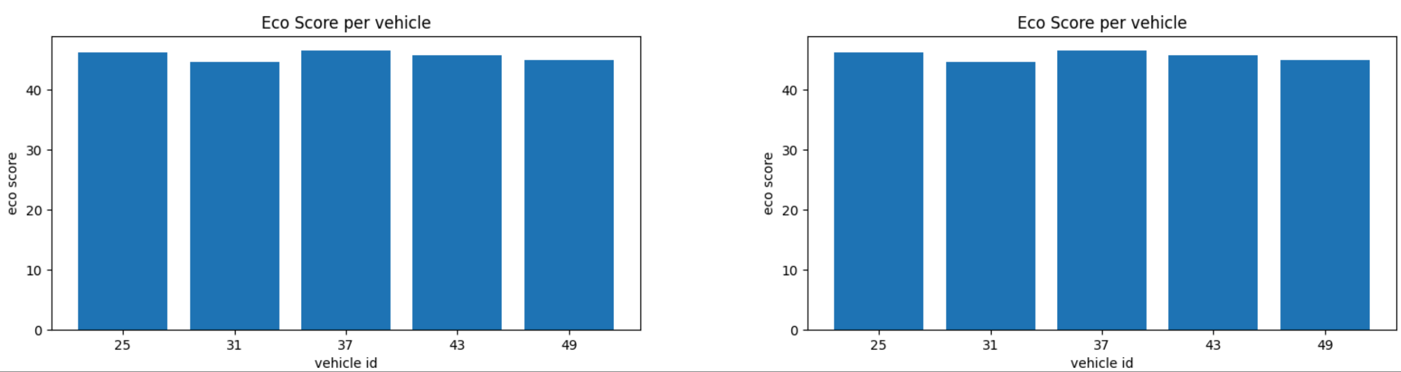


Fig. 8. Eco score per vehicle for the platoon stability scenario under the PID controller (left) and ECO controller (right). All five vehicles score in the low 40s range under both controllers.

The speed efficiency score from analyze_speed.py favors PID at 69.02 versus ECO at 67.08. This result reflects a genuine tradeoff: the ECO coasting zone causes the vehicle to drift slightly below the target speed during cruise, increasing the mean absolute tracking error that drives the tracking component down from 32.75 (PID) to 28.03 (ECO). The smoothness scores are nearly identical at 93.20 and 93.11, respectively. The results are shown in Figure 9 and Table III below.

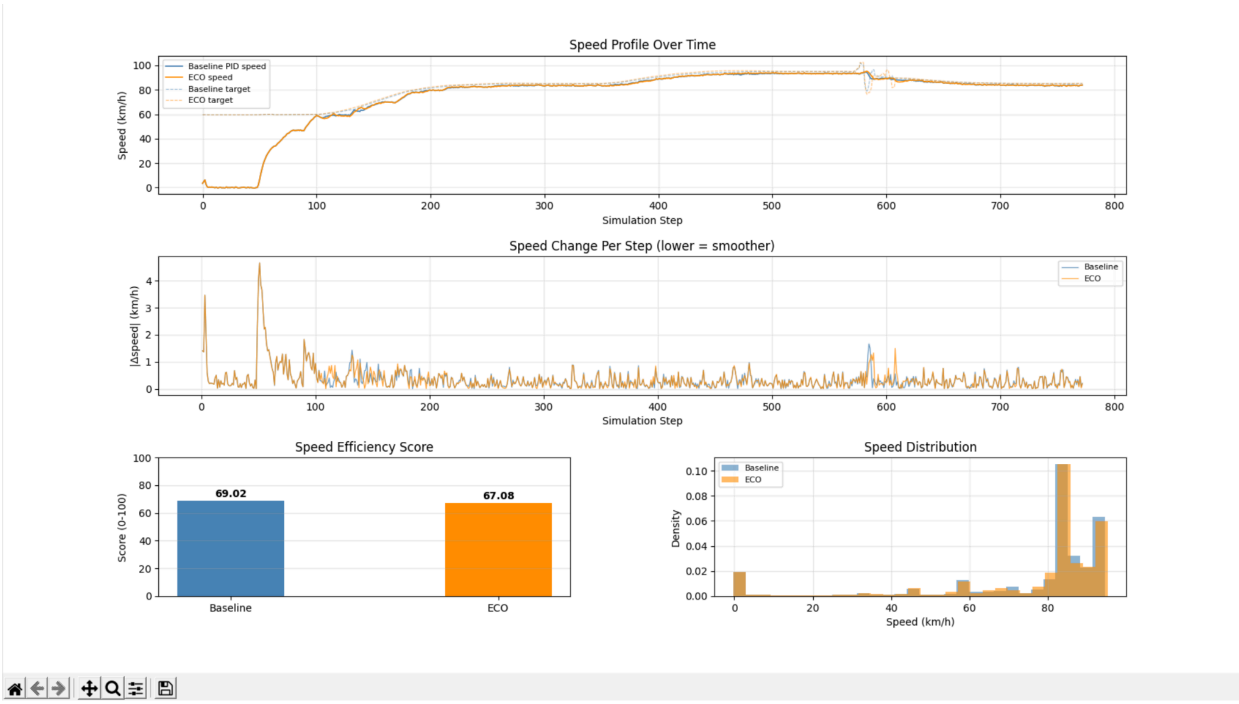


Fig. 9. Speed efficiency analysis for the platoon stability scenario. PID scores higher (69.02 vs 67.08) due to better target tracking (32.75 vs 28.03), while smoothness scores are nearly identical (93.20 vs 93.11), illustrating the tradeoff between tracking precision and driving smoothness.

Metric	Baseline	ECO
Efficiency Score (0-100)	69.02	67.08
Smoothness component	93.2	93.11
Tracking component	32.75	28.03
Mean speed (km/h)	75.85	75.85
Max speed (km/h)	94.58	95.22
Speed std dev	24.15	24.18
Steps logged	773	773

Table III: Speed efficiency metrics for vehicle 49 in the platoon stability scenario. PID achieves a higher overall efficiency score (69.02 vs 67.08), driven by a superior tracking component (32.75 vs 28.03), reflecting the ECO controller's coasting zone intervention, causing slight speed deviations below the target. Smoothness components are nearly identical (93.20 vs 93.11), confirming that the two controllers produce similar step-to-step speed changes despite their different throttle profiles.

These results reveal an important tension between the two scoring systems used in this project. The speed efficiency score rewards tight target tracking, which the PID controller achieves by continuously correcting any deviation with aggressive throttle commands. The sustainability eco score and the physical energy metrics reward smooth, low-jerk behavior, which the ECO controller achieves by accepting slight speed deviations in exchange for stable throttle. In a separate run pair conducted earlier in the development process, the sustainability eco score favored ECO (45.90 vs 45.08), and energy consumption was consistently lower for ECO (302.4 vs 305.4 Wh), while the speed efficiency score consistently favored PID. This divergence between the two metrics is itself a finding: depending on whether the evaluation prioritizes precise platoon following or energy efficiency, different controllers will appear superior.

Spatial Emissions Analysis

The spatial emissions heatmap generated by plot_grid_heatmap provides a complementary perspective on the sustainability results. In the freeway scenarios, the heatmap consistently shows a horizontal strip of accumulated CO2 corresponding to the freeway corridor. The color gradient along this strip reveals the spatial distribution of emissions: low values at the western end of the road where vehicles spawn and accelerate from rest, rising values through the middle section where vehicles travel at cruise speed under high aerodynamic drag, and declining values near the eastern end where vehicles have been traveling for less time or are beginning to decelerate.

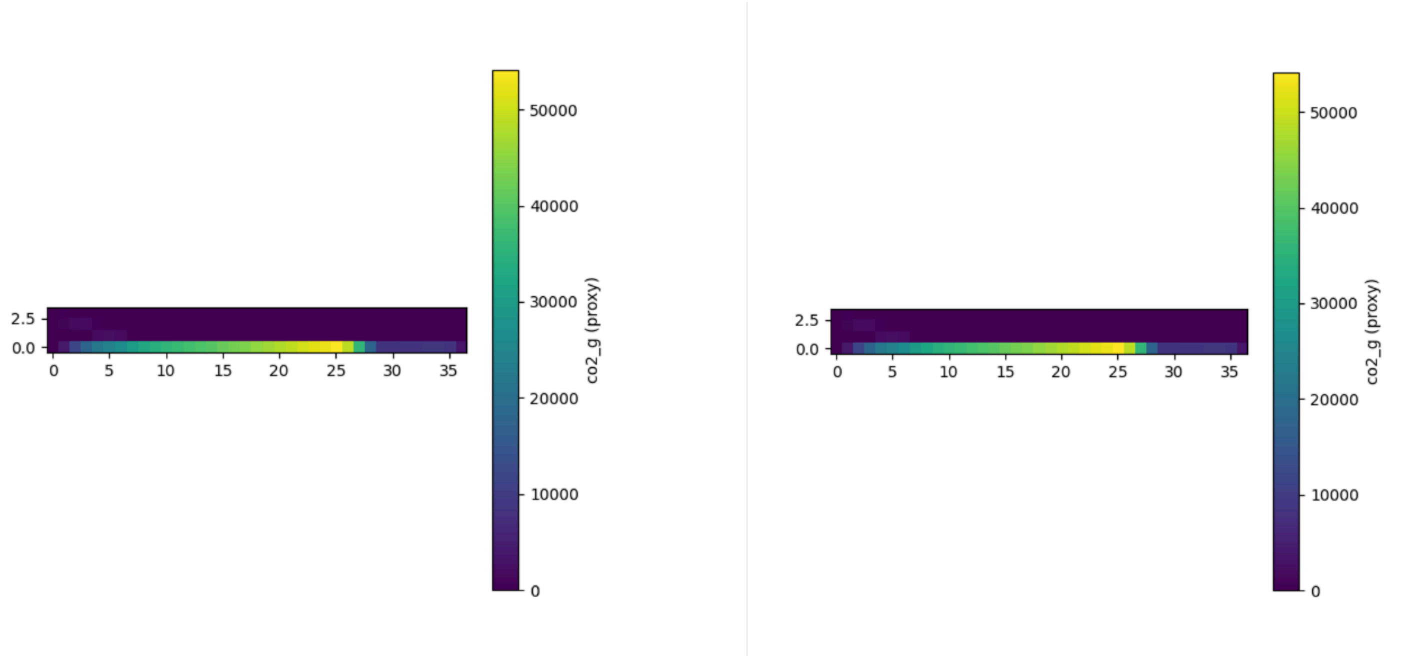


Fig. 10. Spatial CO2 emissions heatmap for the platoon joining scenario under the PID controller (left) and ECO controller (right). The characteristic horizontal strip reflects the freeway geometry. The pattern is nearly identical between PID and ECO runs, consistent with the minimal emissions difference observed in this scenario.

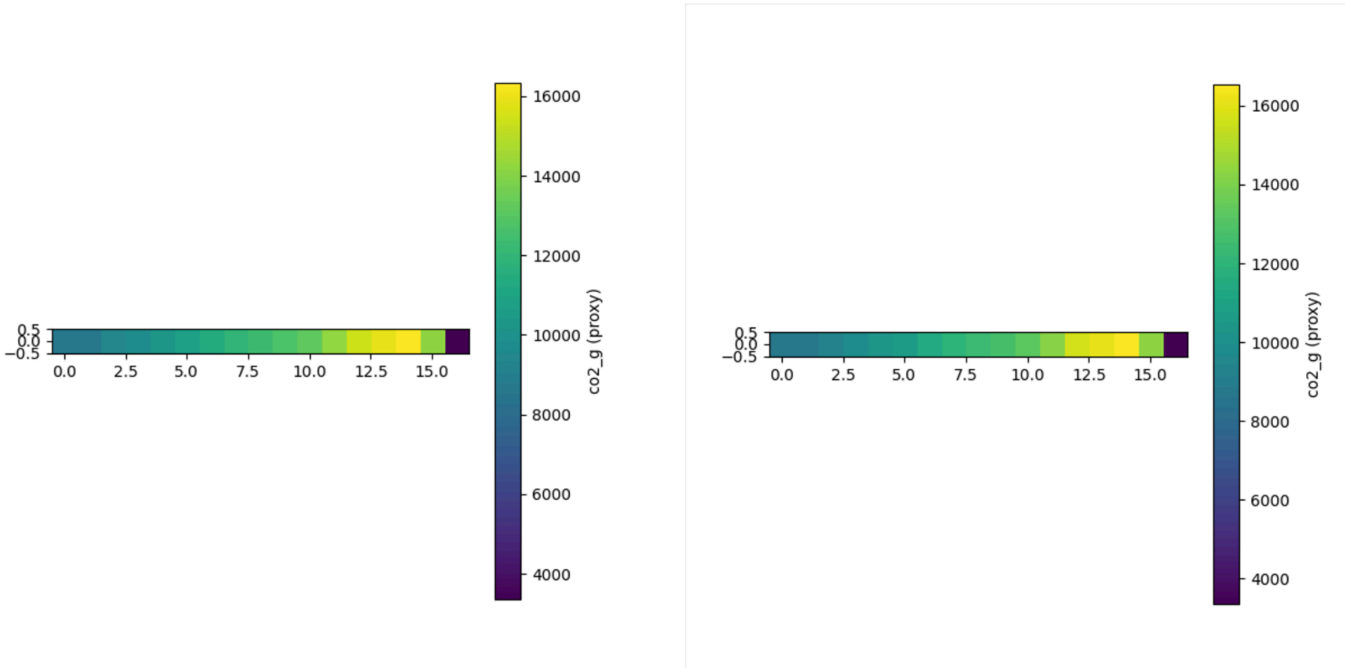


Fig. 11. Spatial CO2 emissions heatmap for the platoon stability scenario under the PID controller (left) and ECO controller (right). The characteristic horizontal strip reflects freeway geometry, with higher emissions in the mid-corridor where vehicles spend the most time at cruise speed.

This pattern is physically expected and validates the energy model. At low speeds, aerodynamic drag is small, and the dominant energy cost is inertial, tied to acceleration events. At cruise speed, aerodynamic drag grows quadratically with velocity and becomes the dominant power demand, producing consistent energy consumption per unit distance and therefore high cumulative emissions in the mid-corridor cells where vehicles spend the most time. The drop-off at the eastern end reflects the fact that vehicles spawning later in the simulation or traveling faster may spend less total time in those cells.

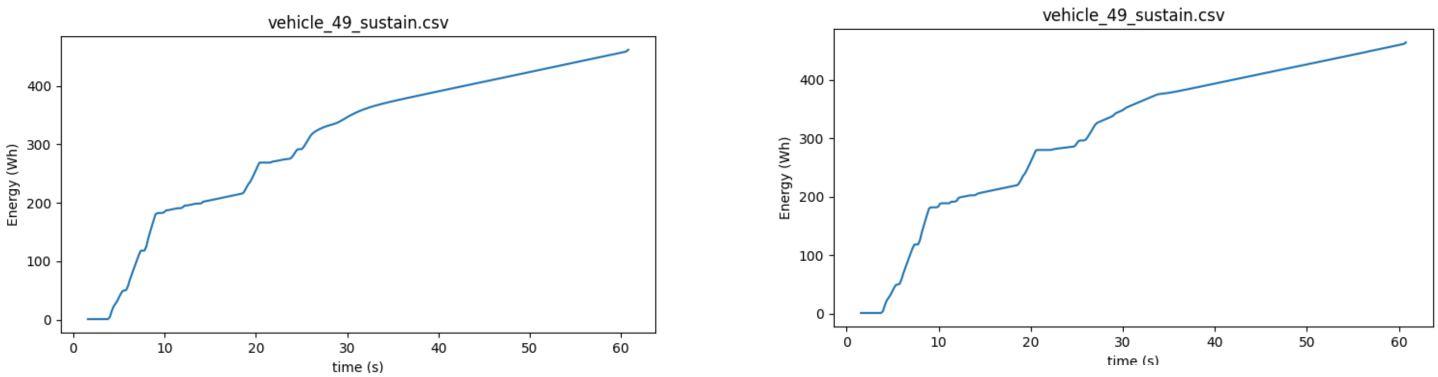


Fig. 12. Cumulative energy consumption over time for vehicle 49 in the platoon joining scenario under PID (left) and ECO (right). Both curves follow similar trajectories and reach comparable final values, consistent with the minimal energy difference observed between controllers in the platoon joining scenario.

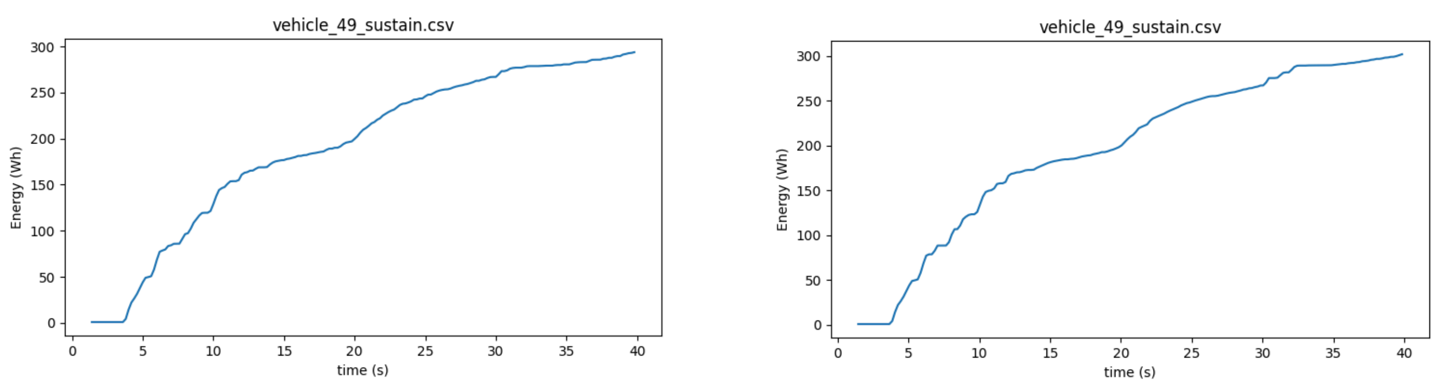


Fig. 13. Cumulative energy consumption over time for vehicle 49 in the platoon stability scenario under PID (left) and ECO (right). Both curves follow similar trajectories, with ECO reaching a slightly lower final value.

Comparing the heatmaps between PID and ECO runs shows subtle differences in the emission distribution, consistent with the small but consistent reduction in total CO2 observed in the vehicle-level metrics. Because the freeway geometry is simple and the speed profiles are nearly identical, the spatial differences are not dramatic in this scenario. However, the framework is designed to show more pronounced spatial differentiation in scenarios with variable-speed corridors, intersections, or congestion zones, where control strategy choices have a larger impact on where and how much emissions accumulate.

ECO Score Sensitivity and Limitations

The eco-driving score formula used in this work was designed to be physically motivated and interpretable, but it has several limitations that should be acknowledged. The weights assigned to each component are based on physical reasoning rather than empirical calibration against measured fuel consumption data. A more rigorous approach would calibrate the weights by fitting to naturalistic driving data where both behavioral indicators and actual fuel consumption are measured simultaneously.

The simulation also introduces its own sources of variability that are not present in real vehicles. CARLA's physics engine produces stochastic noise in velocity and acceleration measurements, which propagates into the jerk samples and affects the jerk standard deviation. This noise component is independent of the controller and represents a floor on the achievable eco-driving score regardless of how smooth the control strategy is. In a real vehicle, sensor filtering would reduce this noise floor. The discrete event counts (harsh acceleration and braking) are also sensitive to how close the vehicle's acceleration comes to the thresholds, meaning that small differences in platoon timing or traffic spawning can shift these counts by one or two events and change the eco-score by a measurable amount.

A further limitation is that the comparison is conducted between a single ego vehicle (vehicle 49) and the rest of the platoon, which uses the standard controller. This means that any effect of the ECO controller on the platoon as a whole is not captured. If the ECO controller causes vehicle 49 to behave differently enough to affect the inter-vehicle gap, this could propagate upstream through the platoon and affect the scores of vehicles 25, 31, 37, and 43. In practice, the gap regulation behavior appears stable across runs, and the platoon vehicles' scores show no systematic difference between PID and ECO runs for vehicle 49, suggesting that the effect is small. However, a more comprehensive study would evaluate the platoon-level sustainability impact of replacing one or more vehicles with ECO-equipped followers.

Finally, the current implementation does not account for road grade, wind conditions, or vehicle loading variations. All physics parameters are held constant at default values representative of a standard passenger sedan. Real-world sustainability evaluation would require these parameters to vary based on actual vehicle specifications, road conditions, and environmental factors. The framework is designed to accommodate these extensions through its parametric physics functions, but they were not implemented in the current version.

CONCLUSION

This capstone project presents a physics-based sustainability evaluation framework for cooperative autonomous driving research, implemented as an extension to the OpenCDA simulation platform running on CARLA 0.9.12. The framework enables real-time per-vehicle measurement of energy consumption, carbon dioxide emissions, eco-driving behavioral indicators, and spatial emissions distribution during simulation, providing a more complete picture of autonomous vehicle performance than traditional safety and efficiency metrics alone.

The framework was used to design and evaluate a custom ECO controller for vehicle 49 in two cooperative platoon scenarios. Through an iterative development process spanning three versions, we identified that broad acceleration smoothing is counterproductive in a platoon-following context because it delays the vehicle's response to speed commands and causes compensatory aggressive behavior. The effective design approach is to apply narrow, targeted interventions that directly address the specific behaviors penalized by the eco-driving score: a throttle coasting zone that prevents full throttle commands during steady-state cruise, and graduated brake capping that softens deceleration commands to reduce harsh braking events.

The Version 3 ECO controller consistently achieves lower energy consumption and CO2 emissions than the standard PID controller across multiple simulation runs. The results are shown below in Table IV.

Metric	PID	ECO
Eco score	45.08	45.90
Energy (Wh)	305.4	302.4
CO2 (g)	79.3	78.5
Harsh accel	82	80
Harsh brake	11	13

Table IV: Vehicle 49 sustainability metrics for a representative platoon stability run. ECO shows improvement on energy, CO2, eco score, and harsh acceleration, while PID shows fewer harsh braking events in this run.

The throttle profile comparison shows a clear qualitative difference: ECO produces a visibly smoother throttle signal capped at 0.6 during cruise, while PID produces rapid spikes to 1.0. The hard brake cap prevents the large brake spike observed in the PID run during the platoon deceleration phase.

The margins of improvement are small, which is somewhat expected given the nature of the scenario. A freeway platoon following scenario with light traffic provides limited opportunity for eco-driving improvements because the vehicle spends most of its time at cruise speed with a well-defined target set by the platoon leader. More substantial improvements would be expected in stop-and-go urban scenarios, congested merging zones, or scenarios with more variable platoon leader behavior. The framework and controller are designed to support these evaluations as future work.

Several directions for future development are suggested by this work. First, the eco-driving score weights should be empirically calibrated against measured fuel consumption data from real vehicles or high-fidelity powertrain simulation models. The current weights are physically motivated but not validated against ground truth energy measurements. Second, the ECO controller could be extended with predictive capabilities using the V2X communication layer already available in OpenCDA. If the ego vehicle can receive the platoon leader's intended speed change in advance, it can begin adjusting its speed profile earlier, reducing the magnitude of acceleration and braking events rather than just softening them after they occur. Third, the spatial emissions analysis could be extended to support scenario design decisions, identifying road segments or traffic configurations that systematically generate high emissions and guiding the placement of speed advisories, signal timing changes, or platoon coordination zones.

The sustainability framework developed in this project represents a contribution to the growing effort to evaluate autonomous driving technology not only by its ability to navigate safely and efficiently, but by its impact on the broader goals of environmental sustainability and societal well-being. As CAV deployment scales from individual research vehicles to fleet-level infrastructure, the control strategies governing these vehicles will have measurable aggregate effects on energy consumption and emissions. Simulation benchmarks that measure these effects rigorously and reproducibly are an essential tool for guiding that development in a responsible direction.

REFERENCES

[1] Qureshi, K. N., & Abdullah, A. H. (2013). A survey on intelligent transportation systems. Middle-East Journal of Scientific Research, 15(5), 629-642.

[2] Dimitrakopoulos, G., & Demestichas, P. (2010). Intelligent transportation systems. IEEE Vehicular Technology Magazine, 5(1), 77-84.

[3] Rahmani, S., Baghbani, A., Bouguila, N., & Patterson, Z. (2023). Graph neural networks for intelligent transportation systems: A survey. IEEE Transactions on Intelligent Transportation Systems, 24(8), 8846-8885.

[4] Wang, L., Deng, X., Gui, J., Jiang, P., Zeng, F., & Wan, S. (2023). A review of urban air mobility-enabled intelligent transportation systems: Mechanisms, applications and challenges. Journal of Systems Architecture, 141, 102902.

[5] Wang, F. Y., Lin, Y., Ioannou, P. A., Vlacic, L., Liu, X., Eskandarian, A., ... & Olaverri-Monreal, C. (2023). Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems.

[6] Musa, A. A., Malami, S. I., Alanazi, F., Ounaies, W., Alshammari, M., & Haruna, S. I. (2023). Sustainable Traffic Management for Smart Cities Using Internet-of-Things-Oriented Intelligent Transportation Systems (ITS): Challenges and Recommendations. Sustainability, 15(13), 9859.

[7] Chen, J., Tao, S., Teng, S., Chen, Y., Zhang, H., & Wang, F. Y. (2023). Toward sustainable intelligent transportation systems in 2050: Fairness and eco-responsibility. IEEE Transactions on Intelligent Vehicles, 8(6), 3537-3540.

[8] Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. (2017, October). CARLA: An open urban driving simulator. In Conference on robot learning (pp. 1-16). PMLR.

[9] Xu, R., Guo, Y., Han, X., Xia, X., Xiang, H., & Ma, J. (2021, September). Opencda: an open cooperative driving automation framework integrated with co-simulation. In 2021 IEEE International Intelligent Transportation Systems Conference (ITSC) (pp. 1155-1162). IEEE.

[10] E. Ericsson, "Independent driving pattern factors and their influence on fuel-use and exhaust emission factors," *Transportation Research Part D: Transport and Environment*, vol. 6, no. 5, pp. 325–345, Sep. 2001, doi: https://doi.org/10.1016/s1361-9209(01)00003-7.

[11] "Introduction - CARLA Simulator," *Readthedocs.io*, 2026. https://carla.readthedocs.io/en/0.9.12/start_introduction/ (accessed Dec. 3, 2025).

[12] "OpenCDA Overview — OpenCDA 0.1 documentation," *Readthedocs.io*, 2023. https://opencda-documentation.readthedocs.io/en/latest/md_files/introduction.html (accessed Apr. 29, 2026).

[13] "Quick Start — OpenCDA 0.1 documentation," *Readthedocs.io*, 2023. https://opencda-documentation.readthedocs.io/en/latest/md_files/getstarted.html (accessed Dec. 3, 2025).

[14] N. Hall, "Drag Equation," *Glenn Research Center | NASA*, Jul. 28, 2022. https://www1.grc.nasa.gov/beginners-guide-to-aeronautics/drag-equation/

[15] x-engineer.org, "How to calculate rolling resistance – x-engineer.org," *x-engineer*. https://x-engineer.org/rolling-resistance/

[16] O. US EPA, "Fuel Economy and EV Range Testing," *www.epa.gov*, Aug. 31, 2015. https://www.epa.gov/greenvehicles/fuel-economy-and-ev-range-testing

[17] Environmental Protection Agency, "Greenhouse Gas Equivalencies Calculator - Calculations and References | US EPA," *US EPA*, Aug. 10, 2015.

https://www.epa.gov/energy/greenhouse-gas-equivalencies-calculator-calculations-and-reference
s

[18] Emissions – Electricity 2025 – Analysis - IEA, “Emissions – Electricity 2025 – Analysis - IEA,” *IEA*, 2025. https://www.iea.org/reports/electricity-2025/emissions

[19] Y. He, X. Yan, C. Wu, D. Chu, and L. Peng, “Effects of Driver’s Unsafe Acceleration Behaviors on Passengers’ Comfort for Coach Buses,” *Second International Conference on Transportation Information and Safety*, pp. 1649–1655, Jun. 2013, doi: https://doi.org/10.1061/9780784413036.220.