

SDAV: OpenCOOD SmartData Model

Dataset

OpenCOOD is an Open COoperative Detection framework for autonomous driving. More details on the dataset can be found at OpenCOOD [Github](#) .

Dataset at LISHA's IoT

- `domain`: opencood
- `username`: opencood
- `password`: A7B64D415BD3E9AA
- `version`: 1.2 (Mobile)

Meta-Data

- Data-set type: "simulation"
- Simulator: "CARLA-0.9.15"
 - Map: "Town06"
 - Description: "Town 6 is a low density town set into a coniferous landscape exhibiting a multitude of large, 4-6 lane roads and special junctions like the Michigan Left."
 - random seed: 11 # seed for numpy and random
 - weather:
 - sun_altitude_angle: 15 # 90 is the midday and -90 is the midnight
 - cloudiness: 0 # 0 is the clean sky and 100 is the thickest cloud
 - precipitation: 0 # rain, 100 is the heaviest rain
 - precipitation_deposits: 0 # Determines the creation of puddles. Values range from 0 to 100, being 0 none at all and 100 a road completely capped with water.
 - wind_intensity: 0 # it will influence the rain
 - fog_density: 0 # fog thickness, 100 is the largest
 - fog_distance: 0 # Fog start distance. Values range from 0 to infinite.
 - fog_falloff: 0 # Density of the fog (as in specific mass) from 0 to infinity. The bigger the value, the more dense and heavy it will be, and the fog will reach smaller heights
 - wetness: 0
 - Map specificities:
 - Road modifications: [None]
 - Includes Traffic Lights: Yes
 - Includes Traffic Signals: Yes
 - Traffic Management:
 - Number of vehicles: 83
 - Range Background Vehicle Spawn:
 - x_max :470
 - y_max: 610
 - x_min: -30
 - y_min: -9
 - x_step: 10
 - y_step: 3.5

Vehicle Attributes

- Model: "vehicle.lincoln.mkz_2017"

- Color: '0, 0, 255' # green
- Extent:
 - length: 4.901684 # meters
 - width: 2.128324 # meters
 - height : 1.510746 # meters
- Torque Curve: # Curve that indicates the torque measured in Nm for a specific RPM of the vehicle's engine.

RPM	Torque
1000	347
1500	523
2000	606
3000	695
4000	677
5300	607
6500	466

- Max RPM: 6500 # The maximum RPM of the vehicle's engine.
- moi: 1 # kg*m². The moment of inertia of the vehicle's engine.
- damping_rate_full_throttle: 0.05 # Damping ratio when the throttle is maximum.
- damping_rate_zero_throttle_clutch_engaged: 2 # Damping ratio when the throttle is zero with clutch engaged.
- damping_rate_zero_throttle_clutch_disengaged: 0.35 # Damping ratio when the throttle is zero with clutch disengaged.
- use_gear_autobox: True # If True, the vehicle will have an automatic transmission.
- gear_switch_time: 0.1 # Switching time between gears.
- clutch_strength: 10 # kg*m²/s Clutch strength of the vehicle.
- final_ratio: 3.21 # Fixed ratio from transmission to wheels.
- forward_gears: # Down = Quotient between current RPM and MaxRPM where the autonomous gear box should shift down. Up = Quotient between current RPM and MaxRPM where the autonomous gear box should shift up.

Ratio	Down Ratio	Up Ratio
4.584000	0.230000	0.540000
2.964000	0.230000	0.500000
1.912000	0.230000	0.460000
1.446000	0.230000	0.460000
1.000000	0.230000	0.460000
0.746000	0.230000	0.460000

- Mass: 1696.0 # Mass of the vehicle in kg
- drag_coefficient: 0.300000 # Drag coefficient of the vehicle's chassis.
- center_of_mass: [0.600000, 0.000000, -0.300000] # Center of mass of the vehicle.
- steering_curve: # Curve that indicates the maximum steering for a specific forward speed.

speed	maximum steering
0.000000	1.000000

20.000000	0.900000
60.000000	0.800000
120.000000	y=0.700000

- wheels:
 - tire_friction: 3.500000 # A scalar value that indicates the friction of the wheel.
 - damping_rate: 0.250000 # Damping rate of the wheel.
 - max_steer_angle: 69.999992 (front), 0 (back) # Maximum angle that the wheel can steer in degrees.
 - radius: 35.500000 # Radius of the wheel in centimeters
 - max_brake_torque: 800.000000 # Maximum brake torque in N*m
 - max_handbrake_torque: 0.000000 (front), 1600.000000 (back) # Maximum handbrake torque in N*m
 - lat_stiff_max_load: 3.000000 # Maximum normalized tire load at which the tire can deliver no more lateral stiffness no matter how much extra load is applied to the tire. Each vehicle has a custom value.
 - lat_stiff_value: 20.000000 # Maximum stiffness per unit of lateral slip. Each vehicle has a custom value.
 - long_stiff_value: 3000.000000 # kg/rad. Tire longitudinal stiffness per unit gravitational acceleration. Each vehicle has a custom value.
 - Wheel position: # World position of the wheel.
 - front left wheel: [10872.801758, 5181.030762, 65.987663]
 - front right wheel: [11029.766602, 5182.101074, 65.987663]
 - back left wheel: [10870.842773, 5468.136719, 65.987663]
 - back right wheel: [11027.808594, 5469.207520, 65.987663]

Vehicle Driving Agent

- Controller: PID
 - lat: k_p: 0.75, k_d: 0.02, k_i: 0.4
 - lon: k_p: 0.37, k_d: 0.024, k_i: 0.032
 - max_brake : 1.0
 - max_throttle: 1.0
 - max_steering: 0.3
- Agent: "carla.agents.BehaviorAgent"
 - Collision_time_ahead: 1.5 # used for collision checking
 - emergency_param: 0.4 # used to identify whether a emergency stop needed
 - ignore_traffic_light: true
 - local_planner:
 - buffer_size: 12 # waypoint buffer size
 - min_dist: 3 # used to pop out the waypoints too close to current location
 - trajectory_dt: 0.2 # for every dt seconds, we sample a trajectory point from the trajectory
 - trajectory_update_freq: 15 # used to control trajectory points updating frequency
 - waypoint_update_freq: 9 # used to control waypoint updating frequency
 - max_speed: 41 # The maximum speed in km/h your vehicle will be able to reach.
 - overtake_allowed: false
 - overtake_counter_recover: 35
 - safety_time: 4 # Time-to-collision; an approximation of the time it will take for your vehicle to collide with one in front if it brakes suddenly.
 - sample_resolution: 4.5 # the unit distance between two adjacent waypoints in meter

- speed_decrease: 15 # How quickly in km/h your vehicle will slow down when approaching a slower vehicle ahead.
- speed_lim_dist: 3 # Value in km/h that defines how far your vehicle's target speed will be from the current speed limit (e.g., if the speed limit is 30km/h and speed_lim_dist is 10km/h, then the target speed will be 20km/h)
- tailgate_speed: 51 # when a vehicles needs to be close to another vehicle asap

Sensor Attributes

- IMU:
 - Type: "carla.sensor.other.imu"
 - Location: [0,0,1] # meters, relative to vehicle center of mass

Blueprint attribute	Type	Value	Description
noise_accel_stddev_x	float	0.0	Standard deviation parameter in the noise model for acceleration (X axis).
noise_accel_stddev_y	float	0.0	Standard deviation parameter in the noise model for acceleration (Y axis).
noise_accel_stddev_z	float	0.0	Standard deviation parameter in the noise model for acceleration (Z axis).
noise_gyro_bias_x	float	0.0	Mean parameter in the noise model for the gyroscope (X axis).
noise_gyro_bias_y	float	0.0	Mean parameter in the noise model for the gyroscope (Y axis).
noise_gyro_bias_z	float	0.0	Mean parameter in the noise model for the gyroscope (Z axis).
noise_gyro_stddev_x	float	0.0	Standard deviation parameter in the noise model for the gyroscope (X axis).
noise_gyro_stddev_y	float	0.0	Standard deviation parameter in the noise model for the gyroscope (Y axis).
noise_gyro_stddev_z	float	0.0	Standard deviation parameter in the noise model for the gyroscope (Z axis).
noise_seed	int	0	Initializer for a pseudorandom number generator.
sensor_tick	float	0.0	Simulation seconds between sensor captures (ticks).

- GNSS:
 - Type: "carla.sensor.other.gnss"
 - Description: calculated by adding the metric position to an initial geo reference location defined within the OpenDRIVE map definition
 - Location: [0,0,1] # meters, relative to vehicle center of mass

Blueprint attribute	Type	Value	Description
noise_alt_bias	float	0.0	Mean parameter in the noise model for altitude.
noise_alt_stddev	float	0.005	Standard deviation parameter in the noise model for altitude.
noise_lat_bias	float	0.0	Mean parameter in the noise model for latitude.
noise_lat_stddev	float	1e-6	Standard deviation parameter in the noise model for latitude.
noise_lon_bias	float	0.0	Mean parameter in the noise model for longitude.
noise_lon_stddev	float	1e-6	Standard deviation parameter in the noise model for longitude.

noise_seed	int	0	Initializer for a pseudorandom number generator.
sensor_tick	float	0.0	Simulation seconds between sensor captures (ticks).

- Camera:
 - Type: "Carla.RGB"
 - Description: "Frontal Camera"
 - Lens: "carla.lens" # configuration of lens is given by blueprint attribute
 - Location: [2.5, 0, 1] # meters, relative to vehicle center of mass
 - Rotation: [Pitch : -35, Yaw : 0, Roll: 0] # Degrees, from -180 to 180

Blueprint attribute	Type	Value	Description
bloom_intensity	float	0.675	Intensity for the bloom post-process effect, 0.0 for disabling it.
fov	float	45.0	Horizontal field of view in degrees.
fstop	float	1.4	Opening of the camera lens. Aperture is 1/fstop with typical lens going down to f/1.2 (larger opening). Larger numbers will reduce the Depth of Field effect.
width	int	4096	Image width in pixels.
height	int	4096	Image height in pixels.
iso	float	100.0	The camera sensor sensitivity.
gamma	float	2.2	Target gamma value of the camera.
lens_flare_intensity	float	0.1	Intensity for the lens flare post-process effect, 0.0 for disabling it.
sensor_tick	float	0.0	Simulation seconds between sensor captures (ticks).
shutter_speed	float	200.0	The camera shutter speed in seconds (1.0/s).
lens_circle_falloff	float	5.0	Range: [0.0, 10.0]
lens_circle_multiplier	float	0.0	Range: [0.0, 10.0]
lens_k	float	-1.0	Range: [-inf, inf]
lens_kcube	float	0.0	Range: [-inf, inf]
lens_x_size	float	0.08	Range: [0.0, 1.0]
lens_y_size	float	0.08	Range: [0.0, 1.0]
min_fstop	float	1.2	Maximum aperture.
blade_count	int	5	Number of blades that make up the diaphragm mechanism.
exposure_mode	str	histogram	Can be manual or histogram. The Auto Exposure Histogram mode first calculates a histogram of the log luminance scene. Then, the histogram is analyzed to determine the average luminance value.

exposure_compensation	float	+0.75	Logarithmic adjustment for the exposure. 0: no adjustment, -1:2x darker, -2:4 darker, 1:2x brighter, 2:4x brighter.
exposure_min_bright	float	10.0	In exposure_mode: "histogram". Minimum brightness for auto exposure. The lowest the eye can adapt within. Must be greater than 0 and less than or equal to exposure_max_bright.
exposure_max_bright	float	12.0	In `exposure_mode: "histogram"`. Maximum brightness for auto exposure. The highest the eye can adapt within. Must be greater than 0 and greater than or equal to `exposure_min_bright`.
exposure_speed_up	float	3.0	In exposure_mode: "histogram". Speed at which the adaptation occurs from dark to bright environment.
exposure_speed_down	float	1.0	In exposure_mode: "histogram". Speed at which the adaptation occurs from bright to dark environment.
calibration_constant	float	16.0	Calibration constant for 18% albedo.
focal_distance	float	1000.0	Distance at which the depth of field effect should be sharp. Measured in cm (UE units).
blur_amount	float	1.0	Strength/intensity of motion blur.
blur_radius	float	0.0	Radius in pixels at 1080p resolution to emulate atmospheric scattering according to distance from camera.
motion_blur_intensity	float	0.45	Strength of motion blur [0,1].
motion_blur_max_distortion	float	0.35	Max distortion caused by motion blur. Percentage of screen width.
motion_blur_min_object_screen_size	float	0.1	Percentage of screen width objects must have for motion blur, lower value means less draw calls.
slope	float	0.88	Steepness of the S-curve for the tonemapper. Larger values make the slope steeper (darker) [0.0, 1.0].
toe	float	0.55	Adjusts dark color in the tonemapper [0.0, 1.0].
shoulder	float	0.26	Adjusts bright color in the tonemapper [0.0, 1.0].
black_clip	float	0.0	This should NOT be adjusted. Sets where the crossover happens and black tones start to cut off their value [0.0, 1.0].
white_clip	float	0.04	Set where the crossover happens and white tones start to cut off their value. Subtle change in most cases [0.0, 1.0].

temp	float	6500.0	White balance in relation to the temperature of the light in the scene. White light: when this matches light temperature. Warm light: When higher than the light in the scene, it is a yellowish color. Cool light: When lower than the light. Blueish color.
tint	float	0.0	White balance temperature tint. Adjusts cyan and magenta color ranges. This should be used along with the white balance Temp property to get accurate colors. Under some light temperatures, the colors may appear to be more yellow or blue. This can be used to balance the resulting color to look more natural.
chromatic_aberration_intensity	float	0.0	Scaling factor to control color shifting, more noticeable on the screen borders.
chromatic_aberration_offset	float	0.0	Normalized distance to the center of the image where the effect takes place.
enable_postprocess_effects	bool	True	Post-process effects activation.

- LIDAR:

- Type: "carla.sensor.lidar.ray_cast"
- Description: "Frontal LIDAR. sensor simulates a rotating LIDAR implemented using ray-casting. The points are computed by adding a laser for each channel distributed in the vertical FOV. The rotation is simulated computing the horizontal angle that the Lidar rotated in a frame. The point cloud is calculated by doing a ray-cast for each laser in every step. $\text{points_per_channel_each_step} = \text{points_per_second} / (\text{FPS} * \text{channels})$. A LIDAR measurement contains a package with all the points generated during a 1/FPS interval. During this interval the physics are not updated so all the points in a measurement reflect the same "static picture" of the scene."
- Location: [2.5, 0, 1] # meters, relative to vehicle center of mass
- Rotation: [Pitch : -35, Yaw : 0, Roll: 0] # Degrees, from -180 to 180

Blueprint attribute	Type	Value	Description
channels	int	64	Number of lasers.
range	float	120.0	Maximum distance to measure/raycast in meters (centimeters for CARLA 0.9.6 or previous).
points_per_second	int	1300000	Points generated by all lasers per second.
rotation_frequency	float	20.0	LIDAR rotation frequency.
upper_fov	float	2.0	Angle in degrees of the highest laser.
lower_fov	float	-25.0	Angle in degrees of the lowest laser.
horizontal_fov	float	360.0	Horizontal field of view in degrees, 0 - 360.
atmosphere_attenuation_rate	float	0.004	Coefficient that measures the LIDAR intensity loss per meter. Check the intensity computation above.
dropoff_general_rate	float	0.1	General proportion of points that are randomly dropped.

dropoff_intensity_limit	float 0.7	For the intensity based drop-off, the threshold intensity value above which no points are dropped.
dropoff_zero_intensity	float 0.15	For the intensity based drop-off, the probability of each point with zero intensity being dropped.
sensor_tick	float 0.0	Simulation seconds between sensor captures (ticks).
noise_stddev	float 0.02	Standard deviation of the noise model to disturb each point along the vector of its raycast.

Simulations

There are two simulations available with multiple vehicles. Each vehicle operates as EGO and collects frontal camera and lidar samples as described further below. Each vehicle is accompanied by a collection of vehicle ground truths. The vehicles are selected at each timestamp whenever they are hit by more than 3 lidar rays. This means that the time series of each ground truth has a different size. If a vehicle does not have a sample at a specific timestamp it could not be identified through perception.

- Simulation 1:
 - Vehicles:
 - `signature`: 641 , t0: 1629145617450000 , tf: 1629145659250000 , length= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 650, 659, 669, 673, 677, 682, 684, 685, 689, 690, 691, 692, 693, 695, 696, 698, 706, 708, 710, 712, 714, 720, 727, 728, 729, 730, 732, 740, 744, 746, 747, 748, 751, 688, 679, 750, 671, 737
 - `signature`: 650 , t0: 1629145617450000 , tf: 1629145659250000 , length= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 641, 659, 671, 677, 682, 684, 685, 687, 688, 689, 691, 692, 693, 698, 705, 708, 710, 712, 714, 720, 727, 728, 730, 732, 737, 740, 744, 746, 747, 748, 750, 751, 695, 669, 676, 738, 755, 706
 - `signature`: 659 , t0: 1629145617450000 , tf: 1629145659250000 , length= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 641, 650, 671, 677, 685, 687, 689, 691, 692, 693, 695, 698, 705, 706, 708, 712, 714, 727, 730, 732, 737, 738, 740, 746, 747, 748, 750, 751, 755, 688, 676, 669, 733, 739, 743, 718, 682, 717, 736, 713, 683, 742, 710
- Simulation 2:
 - Vehicles:
 - `signature`: 1188 , t0: 1629306665500000 , tf: 1629306677600000 , len= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 1197, 1206, 1225, 1227, 1228, 1229, 1231, 1226, 1232, 1230, 1215, 1233
 - `signature`: 1197 , t0: 1629306665500000 , tf: 1629306677600000 , len= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 1188, 1206, 1225, 1229, 1231, 1227, 1228, 1226
 - `signature`: 1206 , t0: 1629306665500000 , tf: 1629306677600000 , len= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 1188, 1197, 1215, 1225, 1227, 1228, 1229, 1231, 1226, 1232, 1230, 1233
 - `signature`: 1215 , t0: 1629306665500000 , tf: 1629306677600000 , len= 4.901683330535889 , width= 2.128324270248413 , height= 1.5107464790344238
 - Ground Truth of Perception: Signatures: 1206, 1226, 1227, 1228, 1230, 1232, 1233,

Vehicle Classes

Vehicle Motion Vectors are stored with a unit containing the class of the vehicle ([ETSI](#)). The class represents is related to the dimensions of the vehicle:

Class	SmartData Unit	Dimensions (Length, Width, Height in meters)
Unknown	0x40000001	N.A.
Moped	0x40010001	2.35 x 0.88 x 1.50
Motorcycle	0x40020001	2.30 x 0.90 x 0.60
Passenger Car	0x40030001	4.42 x 1.98 x 1.55
Bus	0x40040001	10.20 x 2.46 x 3.23
Light Truck	0x40050001	6.90 x 2.90 x 3.40
Heavy Truck	0x40060001	15.00 x 2.45 x 4.00
Trailer	0x40070001	12.00 x 2.50 x 4.30
Special Vehicle	0x40080001	N.A.
Tram	0x40090001	16.00 x 3.00 x 4.00
Emergency Vehicle	0x400a0001	5.20 x 2.30 x 2.74
Agricultural	0x400b0001	4.36 x 2.01 x 2.49

The ground truth motion vectors have all been stored as Passenger Cars.

IMU Data

- **Period:** 100ms

Name	Original Quantity	SmartData Unit	SmartData Quantity	Conversion	Semantics
IMU Acceleration	m/s ²	0xC4962924 (F32)	m/s ²		Accelerometers' readings of the AV, longitudinal (dev=0), lateral (dev=1), vertical (dev=2).
IMU Rotation	Degree	0xC4B24924 (F32)	rad	value * PI/180	Roll, Yaw, Pitch angle in radians (dev=0, 1, 2).
IMU Angular Velocity	rad/s	0xC4B23924 (F32)	rad/s		Gyroscopes Roll, Yaw, Pitch rate in rad/s (dev=0, 1, 2).
Compass (Heading)	rad	0xC4B24924 (F32)	rad		The heading of the speed vector of the object in relation to the true north (WGS84) in radians (dev=3)

Control Data

- **Period:** 100ms

Name	Original Quantity	SmartData Unit	SmartData Quantity	Conversion	Semantics
Steering	Degree	0xC4B24924 (F32)	rad	value * PI/180	The angle of the steering wheel (dev=4)

LIDAR Data

- **Period:** 100ms

Description	Unit	Desc	Semantics
Point Cloud LIDAR	0x03020001	type=3, subtype=2 (xyz,intensity), l=length of configuration (1)	Central 360 degrees lidar (dev=0)

- Original format: ".pcd" with x, y, z, and color as columns (values in F32)
- Convert blob returned query back to PCD: base64_decode twice.

Camera Data

- **Period:** 100ms

Description	Unit	Desc	Semantics
Image	0x02240003	type=2, subtype=PNG (36), l=config 3, between 4MB and 20MB	front center (dev=0)

- Original format PNG 4096x4096 pixels
- Conversion: None, image is read as binary and sent to platform as a blob
- Convert blob returned query back to PNG: base64_decode twice.

Getters

A set of getter examples in multiple different languages is available in the [IoT Platform documentation - Example Scripts](#).

Important Note: On the retrieval of large data (digital data with size > 8 bytes e.g., camera, lidar or MV Global), a single base64 decodes is required to recover the original sample.

Note that a mobile series is represented with at least:

□□□□□□□□

```
"series" : Object { "unit" : unsigned long "t0" : unsigned long long "tf" : unsigned long long "signature" : unsigned int }
```

Note that the [Global Motion Vectors](#) from the ground truth can be read with the following Python code:

```
speed, heading, yawrate, acceleration= struct.unpack("ffff",value)
```

Construction and Usage of CARLA Vehicle Telemetry Dashboard in Grafana

This Grafana dashboard is designed to provide a structured visualization of vehicle telemetry and motion parameters within the CARLA simulator, utilizing the OpenCood SmartData Model (<https://lisha.ufsc.br/Opencood+-+SmartData+Model>). By organizing data within this model, the dashboard enhances the monitoring of vehicle dynamics and environmental interactions in real time.

Description of Variables and Data Collection

The dashboard visualizes several key parameters, which are structured as follows:

IMU Data: The IMU acceleration and rates are essential for understanding vehicle motion. The IMU acceleration graph tracks three-dimensional acceleration—longitudinal, lateral, and vertical—which are crucial to gauge how the vehicle accelerates and decelerates relative to its axes. Meanwhile, the yaw, roll, and pitch rates measure rotational movements, indicating changes in orientation that are critical for assessing stability and control.

Telemetry Data: This section includes variables such as engine RPM, gear, longitudinal speed, and drag force. The drag graph displays the force exerted on the vehicle due to air resistance, measured in Newtons, offering insights into how vehicle speed and environmental factors impact fuel efficiency and performance. The telemetry graph consolidates engine RPM, longitudinal speed, vertical speed, and gear into a cohesive visualization, allowing for simultaneous tracking of the vehicle's speed, power output, and transmission state.

GNSS Data: Geographic data is represented here, including latitude, longitude, and altitude. These GNSS parameters help with precise location tracking within the simulation, providing spatial context necessary for navigation and route analysis.

Structure of the Grafana Dashboard

Following the principles of SmartData's **Series Semantics** (<https://epos.lisha.ufsc.br/Series+Semantics>), the dashboard organizes and filters data based on predefined regions of interest. In version 1.2 of SmartData, this region is defined as a spherical area with a center at "**(x, y, z)**" and a radius "**r**". By limiting data to this spherical region, the dashboard improves data retrieval efficiency, ensuring only relevant telemetry data is displayed.

For example, data on **drag force** is only visualized when within the region of interest, which helps to streamline the analysis by focusing on areas where drag is most likely to affect vehicle dynamics.

Disambiguation and Signature

SmartData uses a "**dev**" identifier as a disambiguator for variables that may share similar units but represent different phenomena. For example, drag force and other forces are measured in Newtons but

serve different purposes in telemetry analysis; the "**dev**" label allows Grafana to distinguish between them without ambiguity. This ensures that each graph on the dashboard clearly represents distinct data types, enhancing interpretability for simulation analysts.

Additionally, each data entry has a unique **signature** associated with the specific vehicle in the simulation. This signature serves as a unique identifier, allowing data tracking across sessions and enabling a reliable historical analysis of vehicle behavior.



Workflow Process

The **workflow** in this dashboard setup involves transforming raw telemetry data from CARLA into the structured visualizations seen on Grafana. Data is initially queried from the SmartData model, then processed and mapped to fit the dashboard's layout. Each graph uses processed data to render real-time visuals based on axis scaling, color schemes, and other parameters tailored for clarity and consistency.

For instance, the **IMU Rates** graph displays yaw, roll, and pitch rates based on continuously updated IMU values. This setup allows for a clear depiction of the vehicle's rotational dynamics, providing real-time insights into its orientation and stability. Similarly, the GNSS graph filters the vehicle's location data to display coordinates within the specified "region of interest," ensuring the focus remains on relevant spatial data.

Summary

This CARLA telemetry dashboard, supported by SmartData version 1.2, offers a structured, efficient approach to monitoring vehicle performance in real time. Using SmartData features like the "dev" disambiguator and region of interest filtering, the dashboard delivers accurate, organized data essential for assessing and optimizing vehicle behavior within simulation scenarios. This setup provides a valuable resource for technical analysis and performance tuning in CARLA-based testing.