Realistic and Scalable Floating Car Observer Detection in SUMO Derived from Co-Simulation

Jeremias Gerner[1], Klaus Bogenberger[2], Stefanie Schmidtner[1]

1 Aimotion Bavaria, Department of Electrical Engineering, , Technische Hochschule Ingolstadt, Ingolstadt, Germany

2 School of Engineering and Design, Technical University of Munich, Munich, Germany

Motivation

Floating Car Observers (FCOs) — connected vehicles equipped with on-board perception — promise a scalable source of traffic-state observations for Intelligent Transportation Systems (ITS). Evaluating FCO-based applications in simulation, however, requires answering a hard question: which vehicles would a real on-board 3D object detector actually see, and how accurately would it localize them?

Existing approaches in SUMO model FCO detection geometrically via line-of-sight raytracing [Ilic et al. 2023; Gerner et al. 2023]. These yield only a binary visibility flag and fail to reproduce the failure modes of real sensor rigs and state-of-the-art 3D detectors — missed objects that are geometrically visible, spurious detections under occlusion, and localization errors that grow with range and depend on the sensor modality.

This work closes the gap with a co-simulation and emulation framework that produces realistic, sensor- and detector-specific FCO detection labels — including detection probability, position error, and confidence. A neural-network emulator distills the co-simulation into a compact model that runs ~100× faster and requires no CARLA map at deployment, making realistic FCO modeling tractable for large-scale studies and RL training loops.

3D Object Detection on Synthetic Data

We deploy three detectors spanning the dominant sensor modalities in modern autonomous-driving perception, all integrated through MMDetection3D and fine-tuned from public real-nuScenes weights on the synthetic CARLA corpus. Each predicted 3D box is matched to the SUMO ground truth via Intersection over Union IoU. The output for every FCO–vehicle pair carries the binary detection decision, the predicted bounding box, the confidence score, and the localization error — quantities that geometric raytracing fundamentally cannot produce. However, also during inference the CARLA server needs to run making it slow and not scalable to simulations, where Carla maps are not available.

Detector	Modality	In-domain val mAP	Ingolstadt-twin test mAP	Real nuScenes val mAP
CenterPoint [Yin et al. 2021]	LiDAR	0.756	0.555	0.564
PGD [Wang et al. 2022]	Camera	0.502	0.121	0.251
BEVFusion [Liu et al. 2023]	Cam + LiDAR	0.824	0.673	0.685

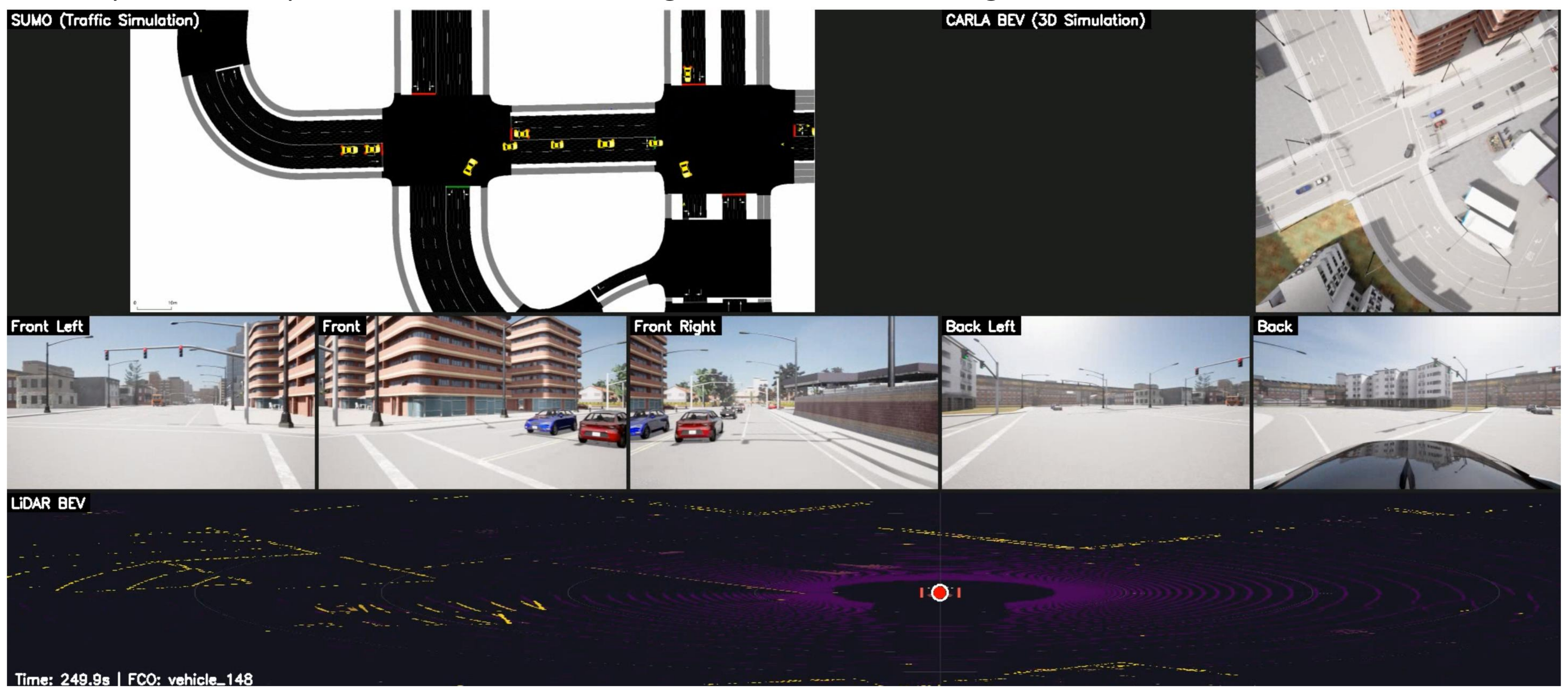
Co-Simulation Framework

Bridge. SUMO and CARLA are coupled through a bidirectional bridge [Dosovitskiy et al. 2027]. SUMO remains the authoritative source of demand, routes, and microscopic driving behavior; at every synchronized step its vehicle poses are mirrored into CARLA, where corresponding actors are spawned, updated, or removed.

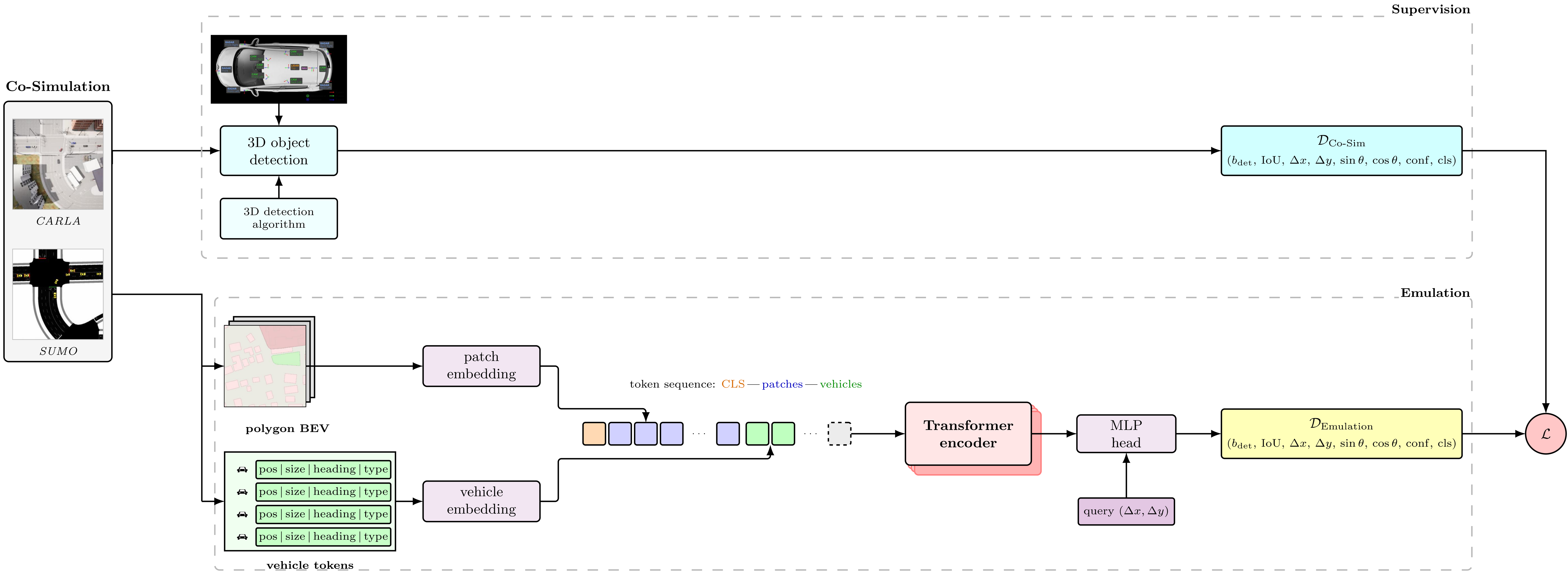
Sensor rig. Each FCO carries a sensor configuration adopted from the nuScenes benchmark [Caesar et al. 2020]: six surround RGB cameras (1600×900 px, 70°/110° FoV) and a roof-mounted 64-channel LiDAR. Camera positions match the real nuScenes vehicle exactly, so detectors trained on the public dataset transfer directly. The framework is sensor-agnostic — alternative rigs can be plugged in.

Maps. Two complementary sources: seven CARLA default Towns (Town01–05, Town07, Town10HD) cover diverse scale and density for pre-training; a photorealistic digital twin of Ingolstadt — ~18 km of road network built from real-world scans [Rössle et al. 2026] — provides a held-out test environment, with calibrated 24-hour demand [Harth et al. 2021] cropped to the twin's extent.

Output. The recording pipeline emits the full nuScenes scene/sample/sweep layout, with six camera images, one LiDAR sweep, ego pose, full sensor calibration, and per-annotation visibility tokens — drop-in compatible with public 3D-detection training and evaluation tooling.



Neural-Network-Based Detection Estimation



To avoid the co-simulation between SUMO and CARLA at runtime while still reproducing realistic detections, we distill the co-simulation into a compact neural network that emulates a specific 3D detector using only information available inside SUMO.
The emulator takes a bird's-eye-view of the local scene around an FCO together with a position vector identifying the queried vehicle, and returns the full detector output: detection probability, position offset, heading, confidence, and class. It learns not only *whether* a vehicle is detected, but how accurately the detector would localize it — information that geometric raytracing fundamentally cannot produce.
On unseen areas, the emulator reproduces detector decisions with over 90 % accuracy, while running 100× faster than co-simulation and on par with 2D raytracing.

Open source. We release the scalable detection-inference code together with trained emulator weights for all three 3D object detectors under the nuScenes sensor configuration — drop-in for any SUMO scenario. No CARLA, no MMDetection3D, no GPU rendering required.