TOWARD A CONSTRUCTIVE AND INTERACTIVE DIGITAL TWIN FOR NEXT-GENERATION ADVANCED TRAFFIC MANAGEMENT SYSTEM

by

Rei Tamaru

A dissertation submitted in partial fulfillment of
the requirements for the degree of

Doctor of Philosophy

(Civil and Environmental Engineering)

at the

UNIVERSITY OF WISCONSIN–MADISON

2026

Date of final oral examination: 04/10/2026

The dissertation is approved by the following members of the Final Oral Committee:
Soyoung Ahn, Professor, Civil and Environmental Engineering
Sikai Chen, Assistant Professor, Civil and Environmental Engineering / Mechanical Engineering
David A. Noyce, Professor, Civil and Environmental Engineering / Industrial and Systems Engineering
Bin Ran, Professor, Civil and Environmental Engineering

Dedicated to my mother, Noriko & father, Masashi

Thank you for inspiring a life filled with curiosity, ambition, and wisdom.

ACKNOWLEDGMENTS

My deepest appreciation belongs to my advisor, Professor Bin Ran. His ability to envision the future of transportation engineering gave my research a clear sense of direction and purpose. I am especially grateful for his willingness to integrate me into his professional network, which allowed me to connect with prominent figures across the field. Working alongside him taught me the practical realities of making a meaningful contribution to the academic community.

I also wish to recognize my committee members: Professor David Noyce, Professor Sue Ahn, and Professor Sikai Chen. Serving under their guidance has been a distinct privilege. The critical feedback they provided from my preliminary exam through my final defense fundamentally shaped my understanding of advanced traffic management systems and digital twin technologies.

Much of my practical development as a transportation engineer is owed to the TOPS lab. I want to thank Steven Parker, Steve Rau, Andrew McFadden, Sam Noyce, and my other colleagues for their daily insights and professional perspectives. Collaborating on active Wisconsin DOT projects provided me with invaluable, real-world engineering experience. I must also acknowledge former lab members Joseph Tuburan, Jon Reihl, Pei Li, and Yang Chen; their mentorship helped me navigate complex, state-of-the-art projects and broadened the scope of my research interests.

The demands of a Ph.D. are made manageable by the people you share the experience with. I will always look back fondly on the first two years spent with my 2021 cohort: Jingwen Zhu, Xinzhi Zhong, Jiwan Jiang, Stephen Kocmoud, and Suravich Khoobaramee. I consider it a great stroke of luck that Cesar Andriola and I were paired together in our first term; our subsequent collaborations, office discussions, and kayaking trips were a highlight of my time in Madison. For keeping the lab environment both productive and enjoyable, I thank Yifan Yao and Keshu Wu, as well as Xiaotian Li, Shuoxuan Dong, Haotian Shi, and Sicheng Fu for our pickup basketball games. I also extend my thanks to the Institute of Transportation Engineers and transportation communities for cultivating my love for the field.

Thank you to Amy Koike for standing by me, for our endless discussions, and for embarking on this life together. Through her, I gained wonderful friends: thank you to Laura Stegnar for the hikes and her cat, Gwen, for being such a calming presence, Callie Kim for the movie outings, Irene Ho and Simon Li for taking me flying and letting me be your co-pilot, and Tugce Pekcetin and Serkan Pekcetin for inviting me to your heartwarming home parties. Their presence truly enriched my life in Madison.

Lastly, I owe everything to my brother, Rion Tamaru, and my parents, Noriko Nakashima and Masashi Tamaru. Their unconditional love, boundless energy, and wholehearted encouragement in everything I do have been the foundation that made all of this possible.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Advanced Traffic Management Systems (ATMSs) are becoming increasingly data-rich through advances in sensing, simulation, and communication technologies. Yet the digital infrastructures adopted to support traffic management remain limited in their ability to construct roadway state from observations, operationalize procedural transportation knowledge, and represent traffic behavior in a form that supports higher-level reasoning and scenario-based evaluation. This dissertation argues that next-generation traffic digital twins must be both *constructive*, in their ability to build, modify, and evaluate roadway state in a virtual environment, and *interactive*, in their ability to support knowledge-grounded reasoning between infrastructure, models, and operators.

First, this dissertation develops a constructive digital twin framework for lane-level roadway synchronization from distributed traffic observations. Through weakly supervised lane detection and federated meta-learning, **Geo-ORBIT** enables scalable and privacy-preserving reconstruction of roadway geometry across heterogeneous locations, providing the structural basis for a virtual traffic-management testbed. Second, this dissertation investigates how procedural transportation knowledge can be embedded into the digital twin through an open and reproducible knowledge-management framework. **CrossTraffic** codifies transportation procedures into interoperable computational interfaces and extends them with semantic validation, enabling more reliable and consistent interaction between users, models, and transportation rules. Third, this dissertation introduces **GeoLaneRep**, a behavior-grounded lane representation framework that jointly encodes structural geometry, aggregated trajectory dynamics, and operational descriptors into a shared latent space. This representation captures lane-level functional identity and supports downstream tasks such as lane comparison, anomaly detection, and conditional geometry generation.

Together, these contributions establish the foundational layers of an interoperable traffic digital twin that can construct roadway state, interpret traffic semantically, and support virtual reasoning and intervention testing for traffic-management

strategies. Rather than treating the digital twin as a passive descriptive replica, this dissertation positions it as a structured and self-optimizing virtual testbed for infrastructure-aware traffic management.

1.1 Motivation

Traffic Management System (TMS) has long been central to maintaining safety and efficiency across urban transportation networks. As cities grow more complex, active management has become a cornerstone of modern practice (Kuhn et al. (2013)). This is the strategy by which an agency can improve system performance by operating and managing transportation systems in a way that is dynamic and adaptive to current and future conditions. The digital twin concept, a simulated, virtual counterpart of real traffic and road networks, has been recognized as a transformative enabler of such active management, allowing predictive analysis and decision-making for transportation network development (Schroeder et al. (2016); Rudskoy et al. (2021)). Yet, current systems, built around static infrastructure models and manual updates, struggle to cope with the dynamic nature of urban traffic networks (Zhang et al. (2020); Feng et al. (2023)). Specifically, they lack the inherent mechanisms to unfold road scenarios intrinsically because of their complex urban system (Moroni (2025)). Furthermore, current TMS deployments prioritize capacity expansion over adaptive management (Kuciemba et al. (2023)). This mismatch between the physical nature and its digital representation limits the responsiveness and scalability of Advanced Traffic Management strategies.

Another critical limitation of current TMS lies in their closed characteristic architecture, which often overlooks the systemic implications for coordination, transparency, and responsiveness. A particularly salient issue is the lack of cross-system interoperability and human-in-the-loop interactivity in how these systems are architected and operated. Most traffic decision-making frameworks remain embedded in isolated control environments, whether in simulation platforms, planning tools, or real-world TMS software, where procedural logic is fixed and inaccessible to other systems or stakeholders (**?**). This architectural closure manifests in two key ways: (1) the detachment of infrastructure logic from domain expertise codified in formal manuals into TMS pipelines and digital twins; and

(2) the inability to translate system-level decisions into semantically meaningful updates interpretable by downstream applications or users. As a result, traffic operations remain opaque and fragmented, stifling adaptive policy integration and the broader goal of intelligent and collaborative traffic management (Jacobson and McLaughlin (2024)).

Taken together, these shortcomings underscore the need for a fundamental architectural shift toward a *constructive and interactive digital twin*. To overcome current limitations, the digital twin needs to be capable of continuously and autonomously reasoning about road scenarios from its perception while remaining highly interactive, functioning as a semantic translation layer that bridges raw perception with diverse downstream tasks.

1.2 Problem Statement

This dissertation addresses three interdependent limitations that prevent current traffic management digital twins from transforming into constructive and interactive systems:

1. Lack of scalable lane-level geometry synchronization prevents digital twins from maintaining accurate, up-to-date roadway geometry, undermining their reliability for operational decision-making at the corridor scale.

2. Domain knowledge of transportation systems remains fragmented from digital twin platforms, and requires additional manual lookup or offline processing, limiting the system's comprehension of the scenario.

3. Semantic interoperability is essential for the interaction in the observed traffic scenario and delivering context-aware traffic information across digital twin systems and operators.

1.3 Research Objective

The primary objective of this dissertation is to establish a **constructive and interactive digital twin framework** for Advanced Traffic Management Systems (ATMS). Unlike traditional digital twins that passively mirror roadway conditions, this digital twin is envisioned as both a controllable system variable and a collaborative decision interface. By integrating real-time geometrical sensing, structured transportation knowledge, and semantic reasoning, this framework aims to enable ATMS that are scalable, knowledge-augmented, and capable of reasoning traffic operations for adaptive traffic management. To this end, this dissertation addresses the three research problems identified in 1.2.

1. **Scalable Digital Twin Synchronization**: Develop methods to automatically infer and maintain lane-level roadway geometry from traffic camera data, enabling digital twins to adapt across heterogeneous urban traffic scenarios.

2. **Knowledge-Augmentation**: Integrate procedural transportation knowledge into a modular, continuously deployable framework that supports automated reasoning, validation of roadway geometries, and simulation-driven operational decision-making.

3. **Semantic Structuring for Interoperability**: Design a semantic layer and natural language interface that can bridge infrastructure, digital twin, and operators, laying the foundation for interactive digital twins and human-centered decision support.

2 GEO-ORBIT: A FEDERATED DIGITAL TWIN FRAMEWORK FOR SCENE-ADAPTIVE LANE GEOMETRY DETECTION

As established in the preceding chapter, the foundation of an interactive Advanced Traffic Management System (ATMS) is the ability to continuously synchronize the physical roadway with its digital counterpart. While digital twins offer a transformative approach to traffic management, enabling real-time sensing, analysis, and decision support, their large-scale deployment remains hindered by a reliance on static maps, costly high-fidelity sensors, and pervasive data privacy concerns. To overcome these infrastructural bottlenecks, this chapter introduces **Geo-ORBIT**, a scalable, privacy-preserving framework for scene-adaptive lane geometry detection.

Instead of relying on static, pre-compiled topological maps, Geo-ORBIT dynamically extracts lane geometries directly from observed roadside vehicle trajectories. These extracted structures are subsequently personalized and adapted across heterogeneous camera deployments via meta-learning. Crucially, the framework mitigates the inherent privacy risks of centralized video processing by employing a federated learning architecture. This distributed approach allows the model to collaboratively learn dynamic geometric features across multiple edge devices without ever transmitting sensitive raw video feeds to a central server. Validated through real-world traffic observations and integrated directly into virtual traffic simulators, Geo-ORBIT establishes a computationally efficient, privacy-preserving pipeline for continuous synchronization. Essentially, this framework provides the critical physical perception layer of the digital twin, successfully grounding the operational reasoning (Chapter 3) and behavioral representations (Chapter 4) developed later in this dissertation.

2.1 Introduction

As established, digital twins enable proactive traffic management by maintaining a live connection between physical roadways and their virtual counterparts. In

modern transportation systems, these architectures replicate elements such as vehicle dynamics, pedestrian behavior, and signal statuses by leveraging data from diverse sensor networks (VanDerHorn and Mahadevan (2021); Kritzinger et al. (2018); Jones et al. (2020); Fuller et al. (2020)). While this paradigm offers immense promise for data-driven optimization, its practical performance relies heavily on high-quality data and robust modeling schemas. This reliance introduces two critical bottlenecks that severely hinder deployment in real-world settings: (1) a lack of accurate, real-time sensing for lane-level roadway geometry, and (2) a strict dependence on centralized processing architectures that provoke concerns regarding data privacy, communication overhead, and system scalability.

Transportation systems are commonly conceptualized through a five-layer model (Fig. 2.1a). Recent progress in the field predominantly targets the upper layers, successfully replicating dynamic entities such as vehicles, pedestrians, signals, and communication networks (Wang et al. (2021); Hui et al. (2022); Dong et al. (2023); Tan et al. (2023); Liao et al. (2024b); Wang et al. (2024b); Cai et al. (2023); Zelenbaba et al. (2022); Liu et al. (2022); Cazzella et al. (2024); Wágner et al. (2023); Dasgupta et al. (2023); Fu et al. (2024b)). In stark contrast, the lower layers—specifically the precise, lane-level understanding of physical infrastructure and its temporal modifications—remain underexplored. Most existing frameworks extract structural data from separate, static sources like OpenStreetMap (OSM) (Dasgupta et al. (2023); Wang et al. (2024b, 2021); Wágner et al. (2023)) or rely on expensive hardware such as LiDAR (Pan et al. (2024); Jiang et al. (2022); Zhang et al. (2019); Davletshina et al. (2024)). While functional in highly constrained environments, these approaches are either profoundly labor-intensive to maintain or prohibitively costly. This effectively cripples the scalability of lane-level digital twins across expansive urban networks.

The second major challenge stems from the underlying system architecture. Conventional digital twins demand extensive data aggregation, inherently necessitating centralized processing (Fig. 2.1b). This centralization triggers severe vulnerabilities regarding data privacy, immense communication costs, and network scalability constraints. To circumvent these architectural flaws, decentralized machine learning

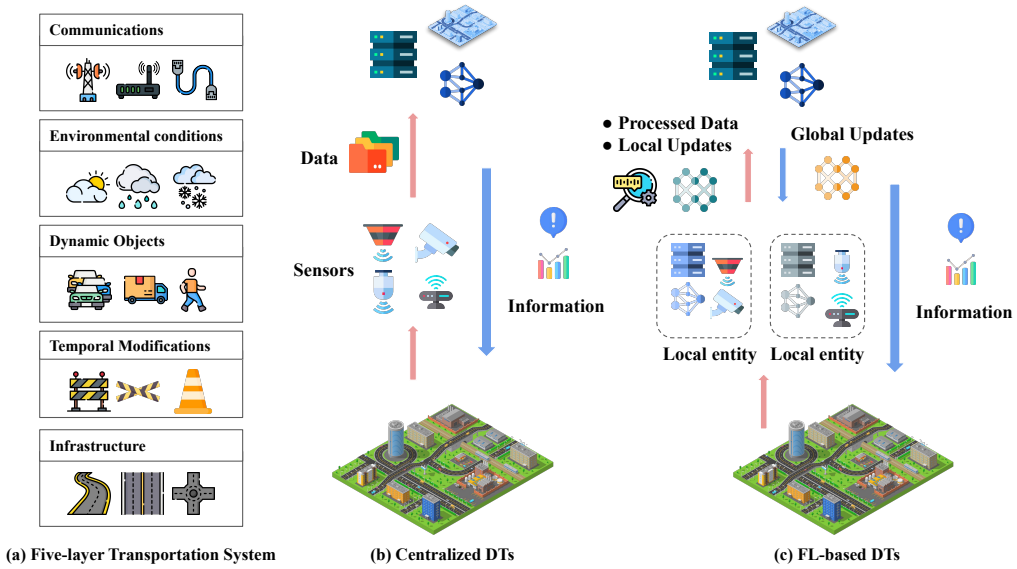


Figure 2.1: Conceptualization of five-layer transportation systems and digital twins. (a) Five-layer transportation system model adopted from Scholtes et al. (2021), with exemplary entities on the different layers in order of interference amplitude. (b) Centralized digital twins collect data from the physical space and create global models and digital twins in the central server. (c) Federated learning-based digital twins utilize local entities and the global server communicating simultaneously to update the model without sharing the raw data.

paradigms, particularly Federated Learning (FL), present a highly effective alternative (Kairouz et al. (2021)). FL is an established machine learning technique that allows multiple edge entities to collaboratively train a shared model while keeping all raw data strictly localized (Zhang et al. (2024)). By adapting this paradigm for transportation (Fig. 2.1c), local entities like roadside units or connected vehicles process raw camera feeds locally. They transmit only processed model updates to a central server. The server aggregates these weight updates to refine a global model, which is then redistributed to the edge. This methodology resolves privacy constraints and minimizes bandwidth usage, rendering large-scale deployment feasible.

To further contextualize these architectural and perceptual gaps, Fig. 2.2 categorizes representative studies across various application domains. The distribution

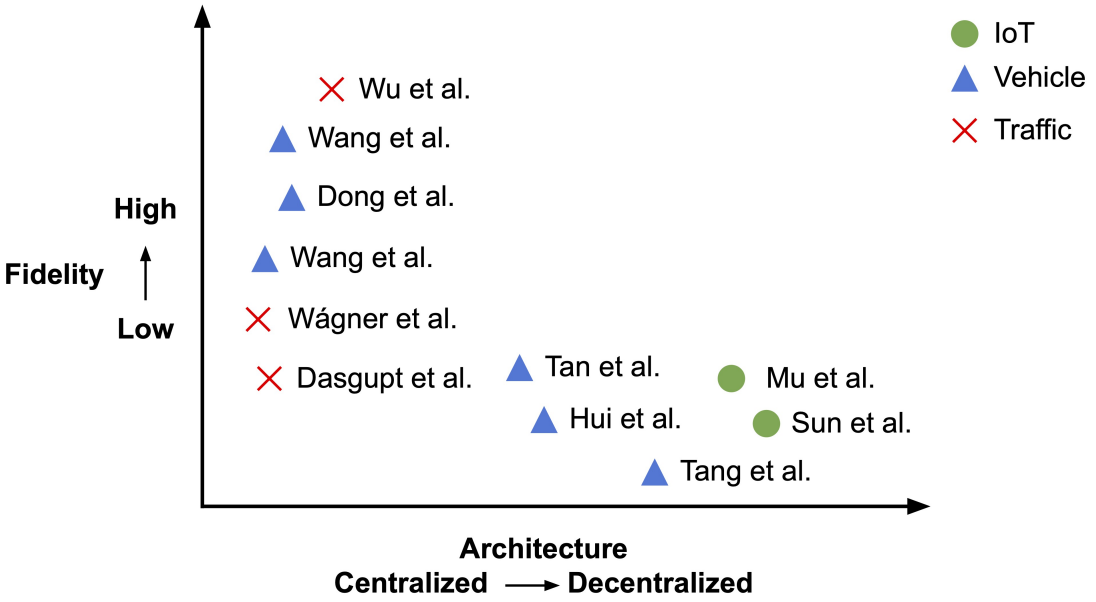


Figure 2.2: Comparison of existing digital twin studies based on system architecture (centralized to decentralized) and fidelity (low to high), categorized by domains including IoT, vehicle, and traffic.

reveals a distinct void in systems capable of both decentralized operation and lane-level microscopic resolution. While several frameworks explore decentralized architectures (Hui et al. (2022); Zelenbaba et al. (2022); Liu et al. (2022)), they overwhelmingly rely on low-to-moderate resolution models. These are often limited to pure mathematical abstractions or macroscopic traffic state replications. Conversely, systems that achieve high-resolution microscopic tracking (Pan et al. (2024); Liao et al. (2024b); Dong et al. (2023)) remain stubbornly centralized, exacerbating the aforementioned privacy and communication bottlenecks. A unified framework that seamlessly integrates microscopic, lane-level digital representations with a decentralized, privacy-preserving architecture is critically absent from the literature.

To directly address these intersecting challenges, this chapter introduces the Geometrical Operational Roadway Blueprint with Integrated Twin (Geo-ORBIT).

Geo-ORBIT is a decentralized, microscopic traffic framework that connects real-world lane geometry and vehicle trajectories to a simulated environment, actively preserving privacy while minimizing communication overhead. The framework employs a meta-learning approach to sense lane-level infrastructure, enabling local edge devices to extract contextual geometric knowledge directly from standard roadside camera data. An FL-based optimization strategy trains a global meta-learner on a central server by aggregating parameters from these local entities, strictly avoiding the transmission of private video feeds. This global meta-learner generalizes to unseen locations without requiring site-specific retraining, massively enhancing overall system scalability. By connecting this continuous perception loop to simulation engines (SUMO and CARLA), the central server constructs a live environment for continuous traffic monitoring and decision-making. The primary contributions of this research include:

- The formulation of a meta-learning framework designed to extract lane geometry information across highly varied camera locations, offering structural scalability and compatibility with diverse detection pipelines.
- The implementation of an FL-based optimization strategy to train the meta-learner. Experimental results on real-world camera data confirm the strategy's capacity to preserve privacy, slash communication costs, and improve geometric generalization on unseen datasets.
- The introduction of comprehensive metrics to assess global shape similarity, local geometry accuracy, and semantic alignment, establishing a robust benchmark for dynamic lane detection tasks.
- The development of an integrated digital twin pipeline that connects real-world lane geometry and vehicle behavior with simulated environments. This pipeline supports the scalable, real-time validation of detection models without requiring manual infrastructure updates.

2.2 Related Works

2.2.1 Infrastructure Sensing in Digital Twins

Existing studies have successfully developed digital twins to replicate vehicles, pedestrians, signal phases, and communication networks (Wang et al. (2021); Hui et al. (2022); Dong et al. (2023); Tan et al. (2023); Liao et al. (2024b); Wang et al. (2024b); Cai et al. (2023); Zelenbaba et al. (2022); Liu et al. (2022); Cazzella et al. (2024); Wágner et al. (2023); Dasgupta et al. (2023); Adarbah et al. (2024, 2023); Wang et al. (2023); Fu et al. (2024b)). Driven by the rapid advancement of onboard sensing technologies, vehicle digital twins receive the majority of research attention, supporting critical applications like cooperative driving, crash avoidance, and behavior prediction. For instance, Wang et al. (2021) developed a cooperative driving twin for non-signalized intersections that schedules crossing sequences based on telemetry from intelligent vehicles. Similarly, Dong et al. (2023) constructed a digital twin to test cooperative driving automation via a scaled physical sand table, while Wang et al. (2024b) validated a campus-environment twin that aggregates onboard and roadside data to provide active crash-avoidance advisories.

Despite these advances in tracking dynamic agents, infrastructure sensing typically remains a disjointed, secondary process. The foundational road networks in most digital twins rely heavily on public databases such as OpenStreetMap (OSM) or Google Maps (Dasgupta et al. (2023); Wang et al. (2024b, 2021); Wágner et al. (2023)). Sourcing map data externally introduces significant data collection and synchronization overhead, fundamentally hindering the scalability and real-time applicability of the digital twin across expansive transportation systems.

To capture accurate, localized infrastructure information, researchers have explored dedicated hardware sensors like LiDAR and roadside cameras (Pan et al. (2024); Jiang et al. (2022); Zhang et al. (2019); Davletshina et al. (2024)). While LiDAR provides highly precise 3D spatial measurements, excellent for tracking moving objects, its hardware remains prohibitively expensive for ubiquitous deployment. Furthermore, the immense computational overhead required to pro-

cess dense point clouds introduces significant latency, even when the sensors are mounted directly on roadside infrastructure. Conversely, vision-based sensors offer a highly cost-effective alternative for real-time roadway monitoring. Extensive literature has explored camera-based lane detection (Qiu et al. (2024); Ren et al. (2014)). However, these studies predominantly analyze isolated static images. They fail to extract continuous, trajectory-based geometric readings that accurately reflect how drivers actually navigate the physical space. Crucially, these vision-based models operate in isolation and do not attempt to synchronize their detections with a live digital twin environment.

2.2.2 Federated Learning for Digital Twins

Federated Learning (FL) has emerged as a transformative decentralized machine learning paradigm across diverse domains, including mobile edge computing, industrial engineering, and healthcare (Li et al. (2020)). In these fields, FL allows multiple institutions to collaboratively train robust predictive models without ever sharing sensitive raw data, thereby complying with strict regulatory and privacy constraints. In transportation specifically, FL powers advancements in vehicular edge computing, traffic flow prediction, and multi-object detection (Zhang et al. (2024)). By processing video feeds and telemetry locally at the edge, these approaches mitigate severe transmission bottlenecks, enhance data diversity, and rigorously protect driver privacy.

Because digital twins inherently rely on massive data aggregation, centralized architectures (Fig. 2.1b) provoke severe concerns regarding data privacy, communication overhead, and network scalability. FL-based digital twins address these structural flaws by aggregating refined model weights from local entities instead of transmitting raw data. While developing these FL-based twins, most literature focuses heavily on optimizing the decentralized network algorithms. For example, Sun et al. (2021) proposed a reinforcement learning strategy to optimize global aggregation frequency in industrial IoT networks, utilizing a digital twin to map the physical state and energy consumption of the edge devices. To improve target

detection accuracy, Mu et al. (2023) deployed a centralized federated transfer learning framework to pretrain local models, alongside a blockchain-based mechanism to identify malicious entities before model aggregation. Similarly, Tang et al. (2024) designed a semi-asynchronous FL framework to predict vehicle collisions, dynamically adjusting the number of communicating vehicles based on local network conditions. Additionally, Khan et al. (2024) integrated digital twins and FL within vehicular networks, addressing core algorithmic challenges in edge caching and resource management.

While these studies successfully innovate on specific FL transmission and security techniques, they overwhelmingly evaluate their FL-based digital twins using abstract numerical simulations or macroscopic traffic state replications. Relying on these low-resolution mathematical abstractions severely limits the applicability of the resulting frameworks. To effectively support proactive traffic management operations, an FL-based digital twin should successfully synchronize highly localized, lane-level topological configurations without compromising its decentralized architecture.

2.3 The Framework for Lane Detection Algorithm with Federated Learning Integration into Digital Twin

The proposed framework, Geo-ORBIT, enhances real-time traffic management by dynamically sensing, modeling, and synchronizing roadway geometry and vehicle behavior within a digital twin environment. As illustrated in Fig. 2.3, the framework operates across multiple roadside sensing units. Each edge unit locally processes vehicle and roadway observations to infer lane-level geometry. A meta-learning approach dynamically adapts the detection parameters to diverse scene contexts, while Federated Learning (FL) enables collaborative model optimization across locations without transmitting raw video data. The resulting lane geometries and trajectory streams are subsequently synchronized with microscopic simulation

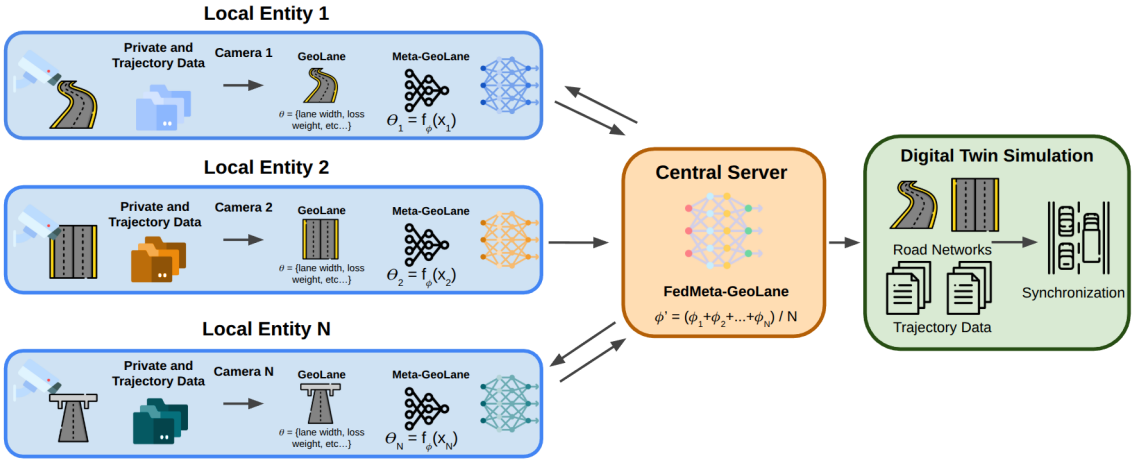


Figure 2.3: Architecture of the federated meta-learning framework. The framework detects roadway geometry at local entities with local GeoLane models. The central server collects parameters from local entities with federated learning. The digital twin synchronizes road geometry and trajectories in a simulated environment.

environments, supporting dynamic digital twin updates for traffic analysis.

2.3.1 Federated Meta-Learning for Lane Detection Framework

To enable the scalable, privacy-preserving deployment of lane detection algorithms across highly diverse roadside environments, this framework utilizes a federated meta-learning architecture. This approach adapts the detection process to local scene characteristics entirely at the edge. Within this framework, each roadside camera constitutes a distinct learning task. A shared meta-learner is trained to output optimal, scene-specific detection parameters. By merging FL with black-box meta-learning (Hu et al. (2023)), this architecture accommodates non-differentiable components within the detection pipeline while ensuring highly efficient model personalization.

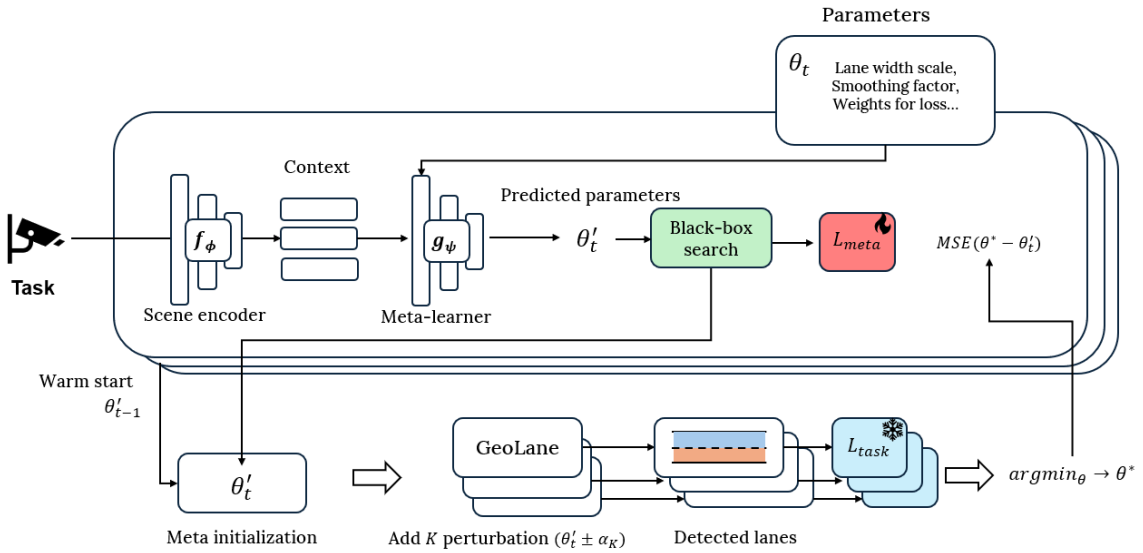


Figure 2.4: Meta-GeoLane Framework.

Task Definition and Motivation

Each camera entity is defined as a discrete task $\mathcal{T}_i$, characterized by unique camera perspectives, intersection topologies, and traffic dynamics. Traditional lane detection models, typically trained centrally on static datasets, suffer severe performance degradation when deployed across such heterogeneous contexts. Their reliance on fixed, hardcoded thresholds renders them suboptimal for real-world variability.

To resolve this, the framework employs a meta-learning strategy. A meta-model learns to infer optimal lane detection parameters θ_i for each specific task based on high-level contextual features x_i (e.g., aggregate trajectory speed, vehicle density). In contrast to gradient-based meta-learning approaches like MAML (Finn et al. (2017)), which demand full end-to-end differentiability, this framework adopts a black-box formulation. It operates exclusively at the parameter level, making it fully compatible with heuristic and spatial detection pipelines.

Meta-Learner Design

As shown in Fig. 2.4, the meta-learner $f(\phi)$ functions as a two-layer multi-layer perceptron (MLP). It features a shared hidden representation coupled with multiple parameter-specific output heads. The network ingests the vector of scene-specific features and outputs a precise dictionary of detection parameters customized for the current task $\mathcal{T}_i$.

Each output head maps the shared latent representation to a scalar value using a linear layer and a bounding activation function (e.g., sigmoid). These bounded outputs are then scaled to the required numeric ranges. For instance, the network predicts specific structural constraints, such as the spatial smoothing factor ($\theta_{\text{smoothing}}$) and peak angle thresholds (θ_{angle}). This architecture ensures interpretable, task-specific parameter estimation directly from scene features, completely bypassing the need to backpropagate through the subsequent spatial detection algorithm.

Federated Optimization Strategy

To guarantee data privacy and facilitate decentralized learning, the meta-learner is trained via a federated optimization protocol. During each federated round:

- A designated subset of client nodes (edge units) receives the current global meta-learner weights ϕ.
- Each client extracts local scene features x_i, utilizes the meta-learner to generate pipeline parameters $\theta_i = f_\phi(x_i)$, and executes the local lane detection algorithm.
- The client computes the local task loss $\mathcal{L}_i$ and its analytical gradient with respect to ϕ, evaluating geometric similarity without backpropagating through the non-differentiable spatial model.
- These parameter updates are transmitted to a central server, which aggregates the gradients to refine the global meta-learner.

This protocol ensures that absolutely no raw trajectory coordinates or image frames are transmitted over the network, strictly adhering to the privacy constraints demanded by intelligent transportation systems. Once deployed to a new edge camera, the meta-learner immediately outputs customized detection parameters $\theta_j = f_\phi(x_j)$ based on the new context, enabling rapid scene adaptation without requiring localized retraining.

2.3.2 Trajectory-Driven Lane Detection Algorithm

Each local client executes a trajectory-driven lane detection procedure. This procedure serves as the non-differentiable pipeline parameterized by the meta-learner's output θ_i. It reconstructs highly accurate lane geometries by conducting spatial analysis on vehicle trajectories captured from edge camera feeds, generating structured boundaries suitable for simulation.

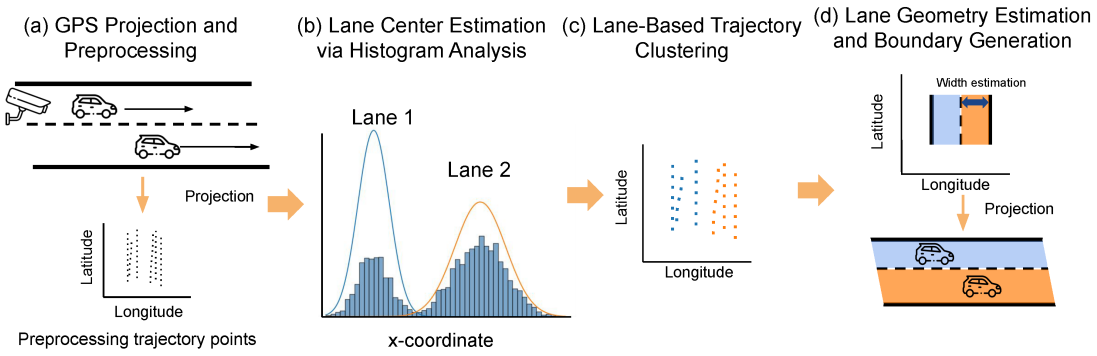


Figure 2.5: Overview of the Trajectory-Driven Lane Detection Algorithm. (a) Video detection and trajectory projection to GPS coordinates. (b) Lane center estimation using histogram analysis. (c) Lane-based trajectory clustering with K-Means. (d) Lane geometry estimation and boundary generation.

Initially, vehicles within the camera's field of view are tracked using the YOLOv11 object detection model (Jocher et al. (2023)) over a defined observation window (e.g., 60 seconds). This raw kinematic data is isolated into distinct *lane

groups*—defined as directional clusters of adjacent lanes sharing a single roadway cross-section.

Raw trajectory detections are first projected from pixel space into global GPS coordinates using homography calibration matrices. This transformation mitigates perspective distortion caused by the camera angle. For each isolated lane group, trajectories are filtered to remove tracking noise and aggregated into a spatial matrix containing mean positions and heading directions.

To dynamically determine the precise number of lanes present within a segment, the system conducts histogram-based peak detection on the distribution of lateral vehicle coordinates. Let $X = \{x_i\}$ define the set of mean lateral positions across all trajectories within a given lane group. A spatial histogram is constructed over X and smoothed via a Gaussian filter.

Crucially, the meta-learned parameters $\theta_{\text{smoothing}}$ and θ_{angle} dynamically dictate the histogram bin count and peak prominence thresholds. This allows the algorithm to adapt perfectly to varying intersection widths. The resulting peaks $\{p_1, p_2, \ldots, p_k\}$ serve as preliminary lane center estimates.

Based on this estimated lane count k, K-Means clustering is applied to the lateral coordinates to definitively assign each trajectory to a specific lane:

$$\hat{y} = \text{K-Means}(X, \text{clusters} = k). \tag{2.1}$$

This mapping discretizes the continuous trajectory space. It assigns each vehicle strictly to the lane corresponding to the nearest detected center, establishing the foundation for localized geometry modeling.

For each isolated lane, the assigned trajectory points are sorted by their longitudinal GPS coordinate y_{gps}. A univariate spline $x = f(y)$ is then fitted to the data points to construct the centerline:

$$x(y) = \text{Spline}(y; s), \tag{2.2}$$

where the flexibility of the curve, s, is governed directly by the meta-learned $\theta_{\text{smoothing}}$ parameter. To estimate the physical lane width, the lateral spread of the

assigned trajectories is calculated:

$$w = 2 \cdot \sigma_x, \tag{2.3}$$

where σ_x represents the standard deviation of the x-coordinates. Assuming a Gaussian distribution of lateral vehicle placement, this captures approximately 95% of the lane's practical width. Finally, physical boundary lines are generated by computing the unit normals along the spline curve and offsetting the centerline:

$$\text{Left}(y) = (x(y), y) + \frac{w}{2} \cdot \vec{n}, \quad \text{Right}(y) = (x(y), y) - \frac{w}{2} \cdot \vec{n}, \tag{2.4}$$

where $\vec{n} = (-\frac{dy}{\sqrt{dx^2+dy^2}}, \frac{dx}{\sqrt{dx^2+dy^2}})$ acts as the normalized perpendicular vector. This continuous process dynamically refines lane center positions and boundaries as new trajectory batches are processed by the edge unit.

2.3.3 Digital Twin Environment Setup

To enable robust validation of the lane detection algorithm under scalable conditions, a trajectory-synchronized microscopic digital twin was constructed by integrating SUMO and the CARLA autonomous driving simulator. This integration provides an environment capable of replicating both large-scale traffic dynamics and highly detailed spatial interactions.

The digital twin construction follows a strict pipeline. The foundational road network is initially extracted from OpenStreetMap (OSM) and converted into the SUMO network format. This network is subsequently converted into the OpenDRIVE (XODR) format, an industry-standard high-definition mapping schema. This conversion guarantees geometric consistency between the 2D traffic simulation (SUMO) and the 3D perception environment (CARLA), facilitating accurate spatial rendering.

To align real-world vehicle movements with this virtual environment, observed trajectories are piped through a synchronization module. GPS coordinates generated by the local camera's homography calibration are translated into SUMO's

local coordinate grid using the `sumolib` network model. This module maps each observed trajectory to the nearest valid network edge, generates specific route files based on the empirical motion sequences, and actively injects these vehicles into the live SUMO simulation. Vehicles are repositioned at every time step to match their real-world counterparts, forcing strict temporal alignment.

During experimentation, the base road geometry within this digital twin remains deliberately static. While dynamic infrastructure updating is a theoretical capability of digital twins, locking the simulation geometry provides a stable, controlled baseline. This configuration allows the system to rigorously evaluate the geometric accuracy and lane assignment consistency of the proposed federated detection model against an established, static map, supporting rapid error diagnosis and trajectory-grounded supervision across heterogeneous urban contexts.

2.4 Lane Geometry Quantification and Evaluation

To rigorously evaluate the quality of the detected lane geometries, this section introduces a comprehensive set of loss functions and evaluation metrics. These metrics quantify both geometric alignment with simulation-based references and mathematical consistency with the learned meta-parameters. The evaluation framework encompasses global shape similarity, local structural accuracy, topological consistency (e.g., absolute lane counts), and predictive parameter fidelity. Furthermore, the framework assesses communication efficiency within the federated network context.

2.4.1 Geometric and Structural Alignment Metrics

The consistency of the detected lane shapes is first assessed against reference map geometries derived from the static SUMO simulation network. This is achieved using the Fréchet distance, a metric that captures the overall shape discrepancy between two continuous curves, accounting for the ordering and flow of points. Let

$P(t)$ and $S(t)$ serve as continuous parameterizations of the reference and detected centerlines, respectively. The consistency loss is defined as:

$$\mathcal{L}_{\text{consistency}} = d(S, P) = \inf_{\alpha,\beta} \max_{t\in[0,1]} \Big\| P(\alpha(t)) - S\Big(\beta(t)\Big) \Big\|_2, \tag{2.5}$$

where α and β are continuous, non-decreasing reparameterizations mapping the unit interval to the curve lengths. This ensures the distance metric evaluates the continuous path structure rather than strictly pointwise Euclidean proximity.

To measure local geometric agreement, the system computes the variance in physical lane widths between the detected lanes and their corresponding SUMO references. The geometry loss is expressed as:

$$\mathcal{L}_{\text{geometry}} = \sum_{M} \Big\| s_{m_w} - c_{m_w} \Big\|_2^2, \tag{2.6}$$

where s_{m_w} and c_{m_w} denote the detected and reference widths of matched lane segments, respectively.

A centerline embedding loss is also incorporated, utilizing a standard triplet structure. This mechanism mathematically forces the latent embedding of the detected lane center s_m closer to the true reference center c_m than to an arbitrary negative sample c'_m:

$$\begin{aligned} \mathcal{L}_{\text{center}} = \sum_{\mathcal{E}} \max\Big\{ & \Big\| f(c_m) - f(s_m) \Big\|_2^2 \\ & - \Big\| f(c_m) - f(c'_m) \Big\|_2^2, 0 \Big\}, \end{aligned} \tag{2.7}$$

where $f(\cdot)$ denotes a learned spatial feature mapping function.

Finally, to enforce strict lane grouping and intersection logic, a topological consistency loss is defined based on the absolute scalar difference between the detected and reference lane counts:

$$\mathcal{L}_{\text{count}} = |N_{\text{det}} - N_{\text{ref}}|. \tag{2.8}$$

2.4.2 Composite Objective Function

The localized, task-level objective function aggregates all previously defined spatial and topological terms into a single formulation, strictly weighted by hyperparameters λ_i:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{consistency}} + \lambda_2 \mathcal{L}_{\text{geometry}} + \lambda_3 \mathcal{L}_{\text{center}} + \lambda_4 \mathcal{L}_{\text{count}}. \tag{2.9}$$

During evaluation, continuous trajectory processing proceeds iteratively until this total loss falls below a predefined tolerance threshold ϵ. At this point, the dynamically detected lane geometry is considered operationally aligned with the static simulation baseline.

2.4.3 Meta-Learning Parameter Alignment

Within the federated meta-learning framework, each edge client predicts a discrete set of geometric detection parameters $\hat{\theta}$. These predictions must be compared against pseudo-optimal reference parameters θ^*, derived from weak supervision via the SUMO environment. The parameter alignment loss is defined utilizing the mean squared error across all parameters in the task set $\mathcal{P}$:

$$\mathcal{L}_{\text{param}} = \sum_{p \in \mathcal{P}} \left\| \hat{\theta}_p - \theta_p^* \right\|_2^2. \tag{2.10}$$

The separation of this parameter loss from the spatial composite loss is a strict architectural necessity. Because the downstream spatial lane detection pipeline relies on non-differentiable operations (e.g., histogram peak detection, discrete K-Means clustering), gradients generated by the spatial composite loss ($\mathcal{L}_{\text{total}}$) cannot be mathematically backpropagated to the meta-learner.

To resolve this bottleneck, $\mathcal{L}_{\text{total}}$ is utilized independently to identify the pseudo-optimal reference parameters θ^* for a given scene. Once established, the continuous meta-learner is optimized strictly against these target parameters using $\mathcal{L}_{\text{param}}$. This surrogate loss dictates the parameter updates transmitted back to the global server during federated synchronization.

2.4.4 Communication Cost

Finally, the framework systematically evaluates the communication overhead inherent to the federated meta-learning architecture. For each distinct round of communication between the edge clients and the global server, the system measures the total volume of transmitted parameter updates and the corresponding duration of the network session. The communication cost is formally quantified as Bits Per Second (BPS):

$$\mathrm{BPS} = \frac{\text{Total bits transmitted}}{\text{Total communication time (seconds)}}. \tag{2.11}$$

This metric provides a rigorous baseline for assessing the real-world scalability of the federated architecture when deployed across bandwidth-constrained roadside networks.

2.5 Experiments and Results

2.5.1 Data Preparation

The Wisconsin Department of Transportation (WisDOT) maintains a perceptual network of over 400 roadside cameras, providing continuous live traffic streams via the 511 Wisconsin infrastructure[1]. To evaluate the proposed framework, this study isolates four specific camera deployments. As illustrated in Fig. 2.6, these sites capture traffic conditions across diverse topological configurations, including multi-lane corridors, complex diverging geometries, and high-density merge zones.

To synthesize the federated learning architecture, trajectory-based lane data is actively aggregated from these discrete camera feeds. Each client operates as an independent geographic edge node. The data generated by each node is strictly characterized by its unique local road geometry and empirical vehicle behavior patterns. This synthesis ensures a comprehensive evaluation of the model's adaptive

[1] `https://511wi.gov/`

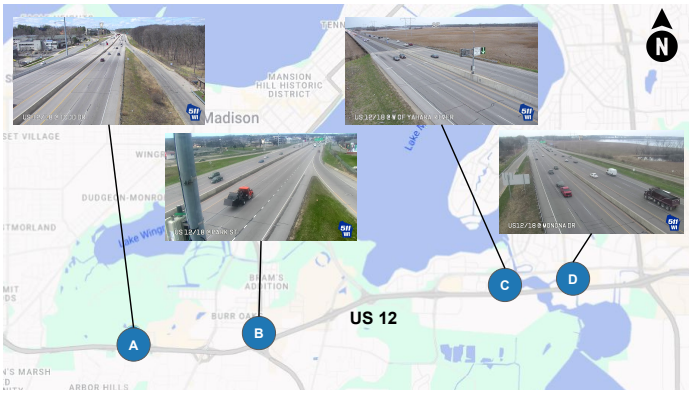


Figure 2.6: Geospatial distribution of the four roadside camera deployments along the US 12/18 corridor in Madison, Wisconsin. Evaluation sites include: (A) Todd Dr., (B) Park St., (C) West of Yahara River, and (D) Monona Dr.

capabilities without requiring physical edge-hardware deployment. The specific data matrix for each client includes:

- Detected trajectory points extracted from processed video frames, representing the dynamic kinematic presence of vehicles.
- OSM-derived pseudo-ground-truth coordinates for lane centerlines, serving as the static spatial reference for validation and alignment.

2.5.2 Performance Comparison

The effectiveness of the FedMeta-GeoLane framework is evaluated utilizing both quantitative spatial metrics and qualitative visual analysis. The models are rigorously assessed on their capacity to accurately infer physical lane boundaries directly from empirical traffic streams. Unless otherwise specified, all continuous validation losses are reported in meters, excluding the discrete lane count metric.

Quantitative Evaluation

Table 2.1 presents a strict quantitative breakdown of the validation loss components for three distinct model configurations across seen and unseen locations. "Seen"

Table 2.1: Validation loss component comparisons of each model on seen and unseen locations.

Model	$\mathcal{L}_{\text{consistency}} \downarrow$	$\mathcal{L}_{\text{geometry}} \downarrow$	$\mathcal{L}_{\text{center}} \downarrow$	$\mathcal{L}_{\text{lane\_num}} \downarrow$	$\mathcal{L}_{\text{total}} \downarrow$
Seen					
Baseline	5.45	15.12	6.78	5.00	77.84
Meta	7.04	11.76	4.73	2.67	12.16
FedMeta	0.0	2.65	3.16	2.67	**6.94**
Unseen					
Meta	18.51	105.35	34.60	12.00	69.61
FedMeta	0.0	12.82	21.39	12.00	**32.38**

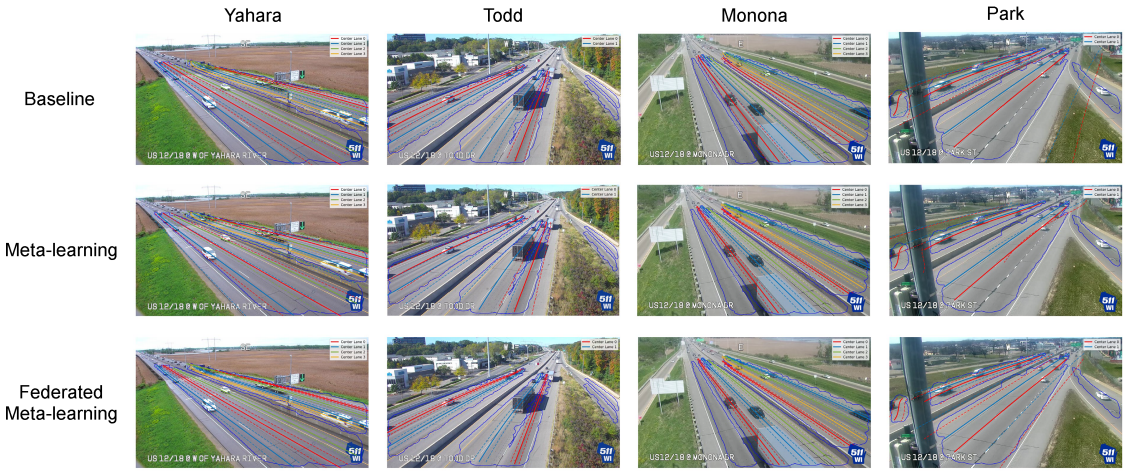


Figure 2.7: Qualitative comparison across multiple locations. The camera at Park St. is treated as an unseen location for Meta-GeoLane and FedMeta-GeoLane. Blue lines represent trajectory contours, and each lane is colored accordingly to denote its assigned lane group.

locations define camera sites explicitly included during the training phase. The metrics trace consistency loss ($\mathcal{L}_{\text{consistency}}$), geometry loss ($\mathcal{L}_{\text{geometry}}$), centerline deviation ($\mathcal{L}_{\text{center}}$), lane count error ($\mathcal{L}_{\text{lane\_num}}$), and the aggregate composite loss ($\mathcal{L}_{\text{total}}$). The evaluated architectures are defined as follows:

- **GeoLane (Baseline)**: A monolithic lane detection model utilizing fixed parameters trained globally across all locations. It applies a uniform geometric configuration and fundamentally lacks any meta-learning or local scene adaptation capability.
- **Meta-GeoLane**: A black-box meta-learning model trained via centralized data aggregation. This model supports local context-conditional parameter generation but absolutely requires the centralized pooling of all training trajectories at the server.
- **FedMeta-GeoLane**: The proposed federated architecture. It merges black-box parameter adaptation with decentralized federated optimization. Each edge client learns site-specific configurations without ever transmitting raw kinematic data, ensuring structural scalability and strict privacy.

Analysis of seen locations indicates that centralized meta-learning substantially outperforms the static baseline. It reduces the total validation loss from 77.84 to 12.16 (an 84.4% improvement). The most significant gains occur within geometric and centerline spatial extraction.

The FedMeta-GeoLane architecture further enhances this performance. It achieves a geometry loss of 2.65 and an exceptional total loss of **6.94** on seen locations. Compared to the baseline, this constitutes a massive 91.1% reduction in total error, while simultaneously outperforming the centralized meta-learning approach by 42.9%. This severe drop in geometry-related loss proves the federated model extracts vastly more precise lane boundaries. Notably, the consistency loss is eliminated entirely. This suggests the learned representation achieves highly stable temporal predictions, rendering continuous shape discrepancy negligible once spatial alignment is locked.

The architectural advantage of FedMeta-GeoLane becomes overwhelmingly clear on unseen locations. The centralized Meta-GeoLane yields a total loss of 69.61 when exposed to a novel environment. The federated variant suppresses this error to **32.38**, marking a 53.5% improvement in zero-shot transfer. This confirms that federated meta-learning not only optimizes performance at trained sites but successfully extracts highly generalizable topological features for completely unseen deployments.

However, lane count estimation remains an outstanding analytical challenge. Despite massive improvements in spatial boundary extraction, the lane number loss ($\mathcal{L}_{\text{lane\_num}}$) remains non-zero across all configurations. This indicates that kinematic geometry-based clustering alone is mathematically insufficient to guarantee the correct inference of active lane cardinality. Future iterations must incorporate stronger topological priors or end-to-end visual reasoning to conclusively resolve lane counting anomalies.

Qualitative Evaluation

Figure 2.7 visualizes the inferred lane geometries across the four highway corridors (Yahara, Todd, Monona, and Park). Centerlines and computed boundaries are overlaid directly onto real-world video frames, providing qualitative proof of the model's spatial accuracy.

At Yahara and Todd, all models successfully capture the overarching structure of the multi-lane environment. Yet, lane count classification remains volatile. At the Todd deployment, all models either improperly merge adjacent lanes or under-segment them, generating visible boundary overlaps.

The Monona deployment exposes a critical data-driven limitation. None of the models successfully identifies the leftmost lane. This physical lane correlates with a trajectory contour possessing extremely sparse vehicle detections within the training set. This omission proves that the meta-learned configuration actively deprioritizes low-frequency spatial zones. The optimization process is inherently biased toward high-traffic structures, ignoring regions with insufficient kinematic

evidence.

The Park scene serves as the ultimate unseen deployment test, characterized by complex curvature and diverging off-ramp geometry. Predictably, no model achieves flawless alignment. While both FedMeta-GeoLane and the baseline successfully infer visually plausible lanes along the primary flow vector, structural inaccuracies emerge immediately within the divergence areas. The centralized Meta-GeoLane performs particularly poorly here, failing to generalize to topological splits it was never trained to recognize.

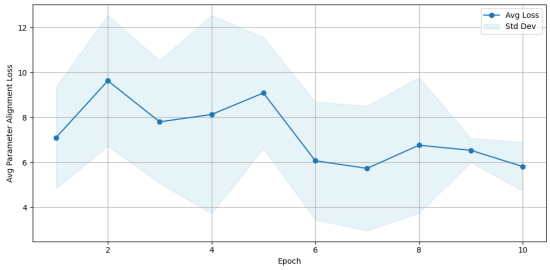


Figure 2.8: FedMeta-GeoLane: the average and standard deviation of parameter alignment losses across ten epochs.

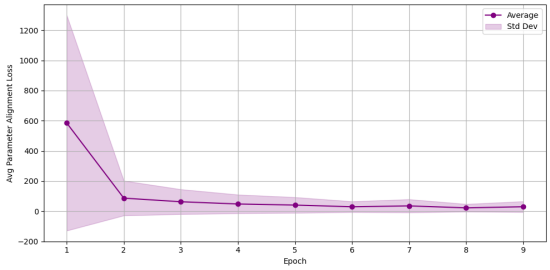


Figure 2.9: Meta-GeoLane: the average and standard deviation of parameter alignment losses across ten epochs.

Finally, Fig. 2.8 traces the parameter alignment loss across federated training rounds for the FedMeta-GeoLane architecture. The average loss steadily decreases, accompanied by a rapid reduction in variance. This proves the federated meta-learner quickly stabilizes its ability to generate task-adaptive geometric parameters. Figure 2.9 plots the identical metric for the centralized Meta-GeoLane. While initial loss spikes higher due to severe client-specific discrepancies, it also converges. However, its overall variance remains significantly wider, indicating much weaker generalization stability when compared directly to the federated architecture.

2.5.3 Transmission Cost Analysis

To rigorously evaluate architectural scalability, transmission costs are quantified in terms of raw data upload/download volumes and the required bit-per-second (BPS)

Table 2.2: Bit Per Second Performance Comparison for All Clients

Parameters	**Baseline**	**Meta**	**FedMeta**
Model size (MB)	0	0	0.2
Clients	4	4	4
Rounds	1	20	20
Model Upload (MB)	0	0	0.01
File Upload (MB)	427.3	427.3	5.6
Download (MB)	0	0	0.018
BPS (Mbps)	3418	3418	**47.2**

network throughput (Table 2.2). Both the static baseline and the centralized Meta-GeoLane models require each client to transmit full video streams or uncompressed trajectory logs to the server. This results in a crippling communication cost of 3418 Mbps. While functionally straightforward, this monolithic approach presents an insurmountable bottleneck for distributed edge deployments due to extreme bandwidth saturation and severe data privacy violations.

In stark contrast, the FedMeta-GeoLane framework completely bypasses this bottleneck. It crushes communication overhead by over 98%, requiring a total BPS of merely 47.2 Mbps. This efficiency is achieved by exchanging only highly compressed, lightweight model weights over 10 training rounds. No raw kinematic or video data ever leaves the edge device, definitively preserving driver privacy while enabling massive system scalability.

2.5.4 Digital Twin Synchronization

Figure 2.10 illustrates a primary capability of the Geo-ORBIT framework: the continuous, real-time synchronization between physical vehicle trajectories and their virtual representations within an integrated SUMO-CARLA environment. In this configuration, CARLA generates a visually rich 3D rendering of the road segments from the perspective of the simulated roadside cameras, while SUMO rigorously drives the underlying vehicle kinematics based on extracted trajectory logic.

Figure 2.10: Digital twin synchronization with SUMO and CARLA at multiple locations, driven by real-time empirical vehicle trajectories.

The system securely links these two environments. It maps the visually inferred vehicle trajectories—processed at the edge by the perception module, directly onto their SUMO counterparts. This creates a side-by-side analytical environment. Researchers can directly compare visually perceived vehicle behavior against the simulation-grounded mathematical truth, providing an indispensable mechanism for validating edge perception models and ensuring absolute digital twin consistency.

2.5.5 Discussions

While the proposed lane detection system demonstrates highly robust empirical performance, specific operational constraints outline clear trajectories for future architectural refinement.

Data-Driven Approach to Lane Detection

The detection pipeline relies exclusively on empirical vehicle trajectories for lane center estimation and spatial clustering. This design deliberately filters out unused pavement, capturing only active operational flow. However, in regions experiencing extremely sparse traffic, such as the leftmost lane in the Monona deployment (Fig. 2.7). This strict kinematic dependence occasionally yields underrepresented

lane boundaries. Furthermore, purely trajectory-based lane count estimation remains a complex challenge (Qiu et al. (2024)). This is particularly true in scenes dominated by partial occlusion, low vehicular throughput, or complex weaving behaviors. Resolving these edge cases necessitates the future integration of secondary contextual priors. Fusing historical traffic patterns or static scene semantics with the dynamic kinematic stream will heavily stabilize lane classification under highly ambiguous geometric conditions.

Calibration Sensitivity

To maintain strict computational efficiency at the edge, the current pipeline employs homography-based transformations to map pixel coordinates into the global GPS space. This mathematical approach is highly effective for standard roadside deployments. Nevertheless, it remains sensitive to severe camera misalignment, imperfect manual calibration, and an absence of ground control points. Upgrading the physical sensory infrastructure would quickly alleviate this spatial uncertainty. Integrating known camera intrinsics, LiDAR-equipped sensor nodes, or survey-grade GNSS modules would drastically enhance coordinate transformation reliability. These hardware advances are particularly vital for securing precise geometric alignment in topographies featuring steep elevation changes or extreme curvature.

Environmental and Simulation Constraints

The digital twin integration successfully isolates and synchronizes trajectory-level realism utilizing the combined SUMO-CARLA architecture. However, the current simulation environment deliberately omits auxiliary physical contexts, such as dynamic weather conditions, dense vegetation, and occluding roadside structures. These environmental variables heavily dictate human driving behavior. Expanding the digital twin to ingest and render these complex, dynamic environmental elements constitutes a critical future milestone. Additionally, evaluating the framework's responsiveness to sudden, temporary infrastructure modifications, such as

active work zones or emergency lane closures, will further validate the system's real-world resilience.

2.6 Conclusions

This chapter presented Geo-ORBIT, a federated meta-learning framework designed to bridge the fundamental gap between real-time roadside perception and dynamic digital twin modeling. By treating each camera deployment as an independent edge entity, the architecture learns optimal spatial parameters via a privacy-preserving meta-learner. This guarantees robust performance across highly diverse traffic topographies. The framework dynamically adapts spatial detection pipelines without requiring local backpropagation. Consequently, it operates effectively in both seen and unseen environments, strictly preserves data privacy, and slashes network transmission costs by orders of magnitude.

Quantitative metrics validate the superiority of this distributed approach, highlighting massive gains in centerline accuracy and cross-camera generalization. Qualitative evaluations confirm that while baseline models extract only coarse geometries, FedMeta-GeoLane consistently locks onto fine-grained spatial alignments. Despite these successes, kinematic lane count estimation and sparse traffic handling remain active research challenges. Future iterations will fortify model robustness by fusing supplementary contextual cues into the perception pipeline. Concurrently, the digital twin environment will be expanded to support highly complex, multi-scenario simulations enriched with dynamic environmental constraints.

By leveraging distributed vehicle trajectories and federated optimization, Geo-ORBIT establishes the critical physical perception layer required for a fully scalable digital twin ecosystem. However, raw geometry alone carries zero operational meaning. To evaluate whether these perceived physical states mathematically conform to established traffic engineering principles, the next chapter introduces a unified knowledge management framework. This subsequent architecture is specifically designed to formally codify regulatory logic and validate infrastructure capacity (Chapter 3).

3 CROSSTRAFFIC: A UNIFIED, OPEN-SOURCE FRAMEWORK FOR REPRODUCIBLE TRANSPORTATION ANALYSIS AND KNOWLEDGE MANAGEMENT

Building upon the scalable, lane-level scene understanding established in Chapter 2, a digital twin requires further conceptual comprehension to semantically interpret its operational capacity. While Geo-ORBIT provides the foundational physical representation of the roadway, a robust Advanced Traffic Management System (ATMS) demands a validation layer. This layer determines whether inferred physical configurations conform to established traffic engineering procedures. Currently, foundational transportation knowledge remains siloed within static, non-reciprocal tools. This fragmentation actively prevents dynamic integration into digital twins, making it difficult to automatically verify representations or quantify their precise operational implications.

To address this gap, this chapter introduces **CrossTraffic**, a unified and extensible framework for intelligent transportation knowledge management. Designed to serve as this critical validation layer, CrossTraffic codifies established transportation methodologies into versioned, testable computational modules. The framework provides the structured facility schemas and algorithmic rules necessary to evaluate whether a digital twin's operating conditions conform to standard engineering procedures. By establishing a dynamic, open ecosystem, CrossTraffic ensures consistent capacity and Level of Service (LOS) computations. This directly enables reproducible scenario testing and collaborative model improvement.

Beyond structural computations, CrossTraffic introduces a novel semantic decision support layer powered by Large Language Models (LLMs). This chat-based interface transforms user interaction with complex Highway Capacity Manual (HCM) methods, facilitating the intelligent retrieval of methodologies and the natural-language orchestration of analytical workflows. Early benchmarks and co-simulation case studies demonstrate the framework's ability to uncover discrepancies between simulation outputs and manual-based calculations, yielding

highly consistent, reproducible results to support human decision-making. While the current implementation is scoped to the HCM and AASHTO Green Book, the underlying architecture is manual-agnostic. It is fundamentally designed to consolidate transportation knowledge across the broader digital twin ecosystem.

3.1 Introduction

Designing safe and efficient transportation systems has long been a central tenet of transportation engineering. This discipline is supported by a shared, foundational body of knowledge, ranging from the empirical equations of the Highway Capacity Manual (HCM) to the regulatory definitions of the American Association of State Highway and Transportation Officials (AASHTO). However, the current ecosystem for managing this knowledge remains overwhelmingly closed and fragmented. Critical methodologies are typically locked within proprietary, black-box software tools or buried within unstructured PDF documents (Oman et al. (2009)). This lack of machine-interpretable representation severely disperses transportation knowledge. It makes baseline assumptions difficult to verify, inherently limiting opportunities for reuse, collaborative innovation, and rigorous analytical reproducibility (Kleinsteuber et al. (2024)).

In response, earlier efforts emphasized the creation of Transportation Knowledge Networks, decentralized infrastructures designed to connect information providers across jurisdictions (Oman et al. (2009); Jenks et al. (2015)). While these initiatives successfully improved document access and facilitated technology transfer, true computational interoperability remains elusive. Consequently, the absence of a unified, machine-readable representation of these methodologies continues to impede automated validation, cross-tool integration, and the reliable deployment of simulation-driven decision support systems within Advanced Intelligent Transportation Systems (ITS) (Harrison et al. (2025)).

To bridge this gap, recent research has increasingly adopted ontology-driven architectures (Bai et al. (2025)). Representing domain knowledge as a structured Knowledge Graph (KG) enables the direct transformation of static manuals into ex-

ecutable standards. These standards actively constrain and validate computational workflows. This paradigm is particularly critical in the era of Generative AI. Large Language Models (LLMs) offer highly capable natural-language interfaces, yet they lack the inherent deterministic reliability required to perform safety-critical engineering without structured mathematical guardrails (Bubeck et al. (2023)). However, a KG alone is insufficient. It must be embedded within an open, auditable software ecosystem that explicitly supports community maintenance and continuous peer review of the encoded logic.

To realize this paradigm, this chapter introduces **CrossTraffic**, an open-source knowledge management system (KMS) designed to represent and execute transportation engineering logic. Within CrossTraffic, regulatory constraints, analytical procedures, and validation rules are encoded as auditable, reusable digital artifacts rather than being obscured within monolithic software applications. The system deliberately separates semantic validation, computational execution, and user interaction into independent, modular components, preserving absolute analytical correctness and reproducibility.

From its inception, CrossTraffic has evolved significantly beyond a simple calculator into a robust, comprehensive KMS. This chapter documents the framework's architecture, its ontological underpinnings, and its empirical validation as a platform for modern transportation systems. The primary contributions of this work are:

1. **A Framework for Open Knowledge Maintenance:** The introduction of a modular architecture that decisively decouples regulatory logic from computational execution, significantly lowering the barrier for community contribution and continuous maintenance.

2. **Ontological Standardization of Terminology:** A demonstration of how a domain-specific KG effectively manages and enforces the complex, hierarchical relationships between transportation terminologies.

3. **Cross-Platform Reproducibility & Reliability:** By centralizing knowledge

within a strict computational core, model-agnostic fidelity experiments prove that integrating this structured knowledge with LLMs reduces computational error by $> 94\%$. This effectively resolves the black-box reproducibility crisis currently inherent in generative AI engineering applications.

3.2 Related Works

3.2.1 Knowledge Management for Open Transportation Science

Recent work in knowledge management has proposed structured, open, and collaborative approaches for organizing transportation information (Pangaribuan and Satrya (2024)). Effective practices in this domain rely on capturing, curating, and sharing expertise across stakeholders to support informed decision-making and enable organizational learning (Bolisani and Bratianu (2018); Olan et al. (2022); Jarrahi et al. (2023)). However, transportation agencies face significant institutional barriers to achieving this vision. Siloed information systems often dominate state departments and municipal agencies. This leads to incompatible systems for traffic data, manuals, and analytical tools, ultimately crippling inter-agency interoperability (Jenks et al. (2015)). Furthermore, proprietary software frequently encapsulates fundamental methodologies, limiting transparent access and preventing the collaborative refinement of analytical procedures (Weerakkody et al. (2021)). To address these challenges, research has proposed curation-oriented models, such as the Digital Curation Center model (Higgins (2008)) and the framework of Irani et al. (2023), which emphasize the systematic description, representation, preservation, and reuse of informational resources.

Building upon these curation efforts, modern transportation practice increasingly relies on data-driven and simulation-based decision support systems. These require the seamless integration of heterogeneous data sources with formal analytical models (Chowdhury et al. (2024)). In parallel, digital twin initiatives have expanded the use of real-time data coupled with microscopic simulation for operational planning and system optimization (Kušić et al. (2023)). Creating a robust

middleware system for cross-platform interoperability is a strict prerequisite for a cohesive smart city testbed (Goumopoulos (2024)). At the organizational level, sustained analytical practice depends entirely on effective knowledge sharing and cross-disciplinary collaboration across safety, mobility, and infrastructure planning (Kleinsteuber et al. (2024); Irfan et al. (2022)). Despite these advances, existing infrastructures primarily support rudimentary data and document exchange rather than providing executable representations of analytical procedures. Consequently, the absence of a unified, machine-executable representation of codified transportation knowledge continues to limit reproducibility and scalable interoperability for traffic operations (Santana et al. (2017); Riehl et al. (2025)).

3.2.2 Semantic Intelligence and Decision Support Layers

Within this chapter, the term *semantic intelligence* refers to language-centered interfaces and representation learning techniques that support the interpretation, retrieval, and explanation of transportation knowledge via natural language interaction.

As transportation systems grow increasingly complex, there is surging demand for artificial intelligence-based decision support tools. These tools help practitioners navigate and query massive bodies of domain knowledge (K. and F. (2018); Zhang et al. (2025)). Recent advances in LLMs and retrieval-augmented generation (RAG) (Lewis et al. (2020)) have enabled powerful natural language interfaces for transportation knowledge access (Tupayachi et al. (2024); Ye et al. (2025)). These systems allow users to pose queries naturally and retrieve relevant methodological descriptions, reports, or documentation, drastically improving usability for non-expert users (Lu et al. (2025)).

In the transportation domain, such language-driven semantic interfaces have been explored for numerous applications, including traffic management support (Masri et al. (2025)), interaction with digital twin platforms (Yang et al. (2024)), and the adaptive control of connected and automated vehicles (Cui et al. (2024)). Despite these advances, LLM-based semantic layers remain fundamentally proba-

bilistic. They operate primarily over unstructured or weakly structured text. As a result, existing studies report persistent limitations regarding analytical reliability, completeness of domain coverage, and the standardization of tool integration (Nie et al. (2025)).

A central challenge is that most current LLM and RAG systems operate completely independent of authoritative engineering manuals and regulatory procedures. This isolation makes it nearly impossible to guarantee that retrieved or generated analytical workflows actually comply with established standards. This frequently leads to semantic hallucinations and severe procedural inconsistencies when models infer or omit required steps.

3.2.3 Ontology-Based Knowledge Management in Transportation

In contrast to probabilistic language-centered interfaces, ontology-based knowledge management formally represents domain concepts, rules, and relationships in strict, machine-interpretable structures that support deterministic reasoning.

Ontology-based Knowledge Management Systems (OKMS) provide a formal specification of shared conceptualizations, enabling software to accurately reason about entities, attributes, and relationships (Gruber (1993); Studer et al. (1998); Hogan et al. (2021)). As reviewed by Mora et al. (2022), OKMS offers a highly structured framework for transforming tacit domain knowledge into explicit, machine-executable logic. By leveraging W3C standards such as the Resource Description Framework and the Web Ontology Language (McBride (2004)), these systems support formal reasoning and the structured integration of heterogeneous datasets across domains. Large-scale infrastructures like KnowWhereGraph (Janowicz et al. (2022)) illustrate the successful application of ontology-based integration in geospatial contexts.

Within the transportation sector, ontologies have successfully supported data interoperability and semantic integration. For example, Fernandez et al. (2016) proposed an ontology-driven architecture based on the Semantic Sensor Network ontology to interpret real-time traffic situations from varied sensor streams. Zhang

et al. (2025) employed KGs to represent mobility-related entities, combining rule-based and neural models to infer semantic trip purposes from numerical data. Similarly, Gan et al. (2025) demonstrated the use of KGs for underground infrastructure management, applying entity disambiguation techniques to consolidate redundant data records.

Despite these successes, a critical gap remains regarding the use of ontologies for normative and regulatory reasoning in transportation engineering. Existing works primarily support descriptive interoperability, such as data integration (Gan et al. (2025)) or general semantic decision support (Moradi et al. (2013)). However, these approaches fail to establish an explicit connection between ontology-based knowledge representations and the actual execution of regulatory validation procedures. This work directly addresses this gap by introducing an executable regulatory ontology. This ontology links formal domain concepts to concrete analytical procedures, rigorously verifying multi-source regulatory constraints prior to any computational execution.

3.3 System Architecture

To support the continuous deployment and advancement of transportation knowledge, this chapter presents the architecture of CrossTraffic. The framework provides a structured approach that strictly decouples the definition of engineering rules from their computational execution.

As illustrated in Fig. 3.1, the framework consists of three tightly connected components: **(1) The Computational Core**, a high-performance Rust library that serves as the authoritative repository for transportation equations and data structures; **(2) The Middleware Layer**, which exposes this core to multiple programming environments through Python bindings and WebAssembly (WASM); and **(3) The Interaction Ecosystem**, which delivers user-facing applications such as web calculators, desktop clients, and intelligent agents. Spanning these components, CrossTraffic incorporates an active, ontology-driven semantic validator that inspects all analytical requests and parameters prior to execution. The mechanics of this validation

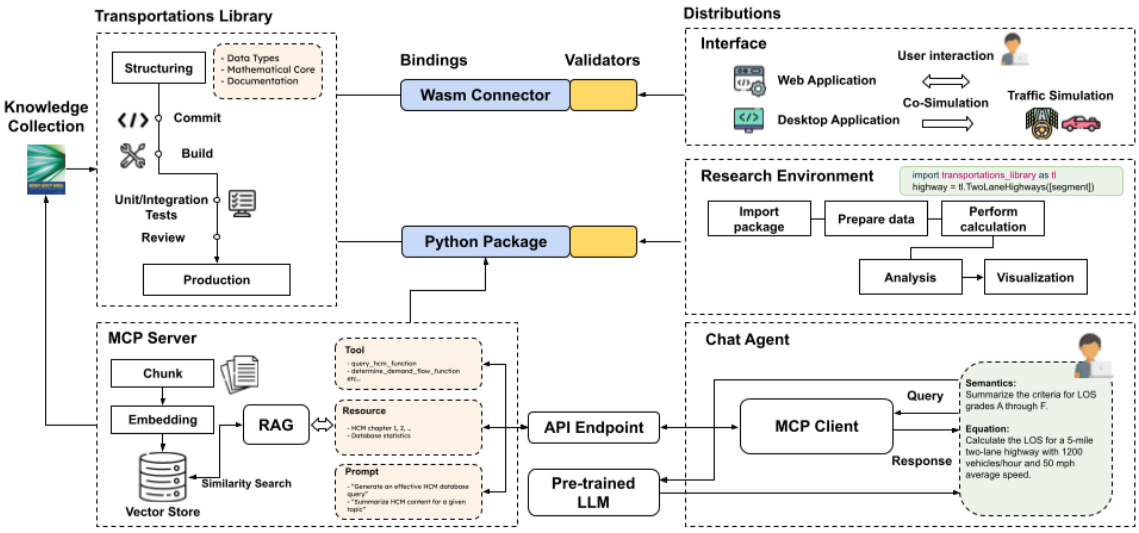


Figure 3.1: Modular architecture of the CrossTraffic platform. The system is stratified into three integrated layers: the Computational Core (Transportations Library) for authoritative calculation, Middleware for providing bindings, and the Interaction Ecosystem for distributions of capabilities across multi-platforms. An ontology-driven semantic validation layer (yellow highlighted area) operates across these components to validate analytical requests and parameters.

layer are detailed in Section 3.4.

3.3.1 Computational Core: Transportations Library

The `Transportations Library` serves as the foundational layer of CrossTraffic, providing the authoritative implementation of transportation analysis procedures. The library is authored in Rust, utilizing a strongly typed and memory-safe programming model (Matsakis and K. (2014)). This enables the compile-time enforcement of parameter structures and dimensional consistency. Rather than treating engineering inputs as unconstrained floating-point numbers, domain parameters are rigidly represented as explicit data types.

Each analytical function and equation from established transportation manuals is encapsulated as a discrete, modular component with precisely defined interfaces. To maintain the absolute integrity of this source of truth, the library adheres to a

strict Test-Driven Development paradigm. A dedicated Continuous Integration and Continuous Development (CI/CD) pipeline runs comprehensive unit tests against official manual examples. It concurrently runs regression tests against previous builds on every commit, mathematically guaranteeing that the equations remain consistent and accurate over time.

3.3.2 Middleware: Python Bindings and WASM Wrappers

To ensure the Transportations Library is broadly accessible across diverse computing environments, a middleware layer compiles the Rust kernel into portable targets. This layer exposes the core computational capabilities directly through Python bindings and WebAssembly (WASM) (WebAssembly Project and Contributors (2022)) wrappers.

The Python interface, engineered via PyO3 (PyO3 Project and Contributors (2017–2025)), empowers researchers and practitioners to seamlessly integrate the library's authoritative calculations into their custom data analysis workflows. This enables high-throughput analysis without the dangerous need to reimplement underlying equations. For web and cross-platform deployment, CrossTraffic utilizes a WASM wrapper to compile the library into portable bytecode. Crucially, this ensures that browser-based calculators execute the exact same mathematical binary as server-side analyses, guaranteeing perfectly consistent results across all operating systems without requiring native installation.

3.3.3 Cross-Platform User Interfaces

The user-facing layer of CrossTraffic grants practitioners direct access to transportation methodologies while preserving absolute computational fidelity across diverse platforms (Fig. 3.2). The primary interface, the HCM Calculator, is distributed simultaneously as a Progressive Web App and a native desktop client via Tauri.

Beyond static, single-scenario analysis, the framework facilitates dynamic validation through hybrid co-simulation. The calculator natively supports exporting standardized inputs as SUMO configuration files (Lopez et al. (2018)). Utilizing

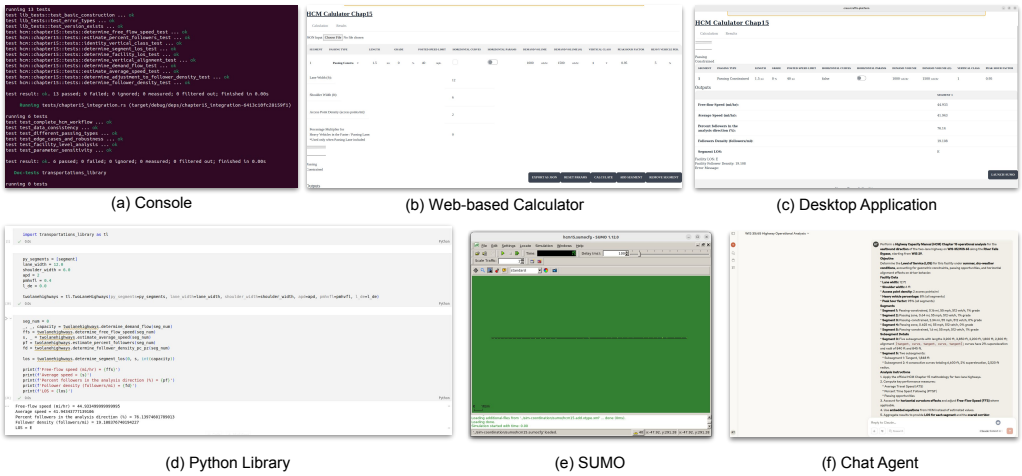


Figure 3.2: CrossTraffic deployed across multiple platforms. (a) Linux console. (b) Web-based calculator interface. (c) Desktop application packaged with Tauri. (d) Python library integration. (e) Hybrid co-simulation with SUMO. (f) MCP connection with the Claude Desktop. Each platform is validated using a consistent scenario setup to ensure reproducibility across environments.

the TraCI interface, the system can launch SUMO simulations directly from the desktop client, establishing an immediate, verifiable connection between the static analytical framework and dynamic microscopic traffic simulation.

Generative AI Integration (MCP)

The final component of the interaction ecosystem is the Model Context Protocol (MCP) server, which securely connects the computational core to generative AI agents. The MCP server explicitly exposes the functions of the Transportations Library as callable tools for LLMs. Consequently, this interface successfully merges probabilistic language-based retrieval (RAG) with highly structured, deterministic tool invocation.

To support language-centered semantic interaction with transportation manuals, the interface adopts the RAG architecture (Lewis et al. (2020)). Official documentation (e.g., HCM chapters) is preprocessed and embedded into a ChromaDB vector database. This facilitates semantic similarity-based retrieval of highly rele-

vant passages in response to user queries. The system then renders an interactive decision-support interface. Users may request complex explanations or full analysis workflows (e.g., level-of-service computations), safe in the knowledge that all numerical results are deterministically generated exclusively through the verified computational core.

3.4 The Ontology-Driven Architecture

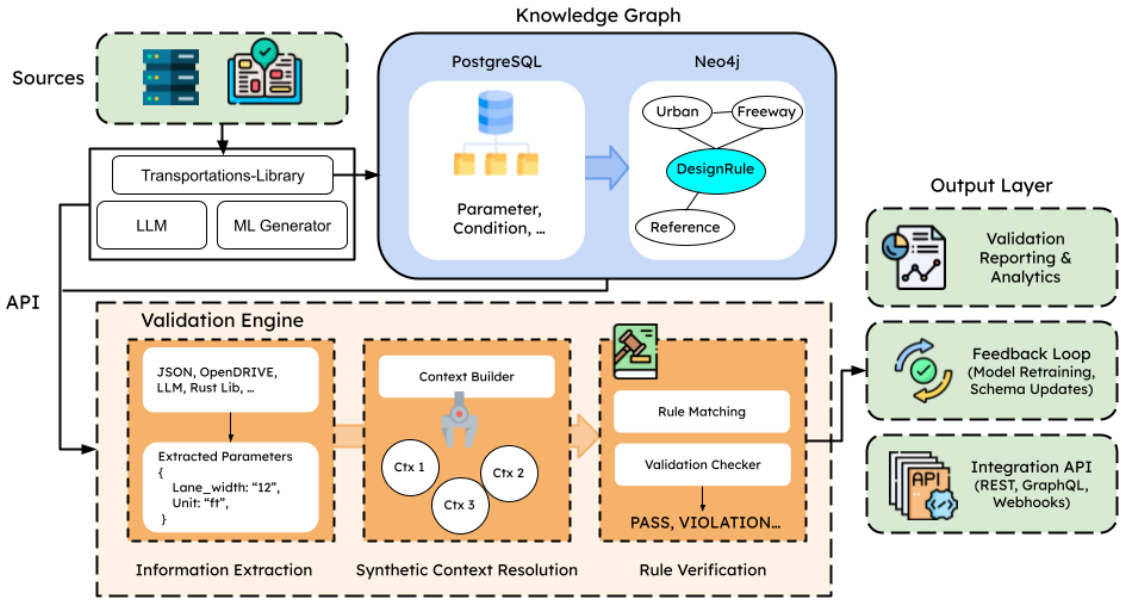


Figure 3.3: The validation execution pipeline. The pipeline converts heterogeneous design inputs into normalized parameters, resolves their regulatory context, retrieves applicable design rules from the KG, and evaluates all constraints through the Validation Engine. It then allows downstream computation and simulation to proceed.

A fundamental challenge in modern transportation analytics lies not in numerical computation itself, but in the strict verification of contextual and regulatory validity. Standard software tools treat engineering inputs (e.g., lane width, grade) as generic numbers. In stark contrast, the CrossTraffic framework treats inputs as

distinct semantic entities. They are relentlessly constrained by formal regulatory and design rules derived directly from the HCM.

While Section 3.3 detailed the computational infrastructure, this section outlines the internal mechanics of the **Semantic Validator**. This specific component transforms the system from a passive data conduit into an active validation layer. It achieves this by validating all inputs against a formal, external Knowledge Graph (KG) prior to execution.

As illustrated in Fig. 3.3, the validator enforces a strict domain-specific ontology. By structuring transportation knowledge as a formal graph $G = (\mathcal{V}, \mathcal{E})$, the system seamlessly executes graph queries to detect complex, multi-variable constraint violations (e.g., speed and radius incompatibility) that simplistic arithmetic validators routinely miss. The following subsections define the ontological structure, the formal constraint definitions, and the algorithmic execution pipeline.

3.4.1 Domain-Specific Ontology Structure

Unlike traditional database schemas designed merely to enforce data types, the Semantic Validator explicitly enforces *engineering logic*. The KG schema is organized into five distinct node classes ($\mathcal{V}$) connected by semantic relationships ($\mathcal{E}$). This structure transcends simple look-up tables by explicitly encoding the complex dependencies between parameters, rules, and their authoritative sources.

Node Entities ($\mathcal{V}$)

The ontology defines the following entity types:

- **Parameter (P):** Represents atomic engineering variables. Each node contains metadata linking it via a foreign key to the corresponding Rust `struct` field, guaranteeing absolute type safety.
- **DesignRule (R):** Encodes the mathematical validation logic. These nodes possess a `rule_type` attribute and a `severity` level, allowing for nuanced, tiered feedback to the user.

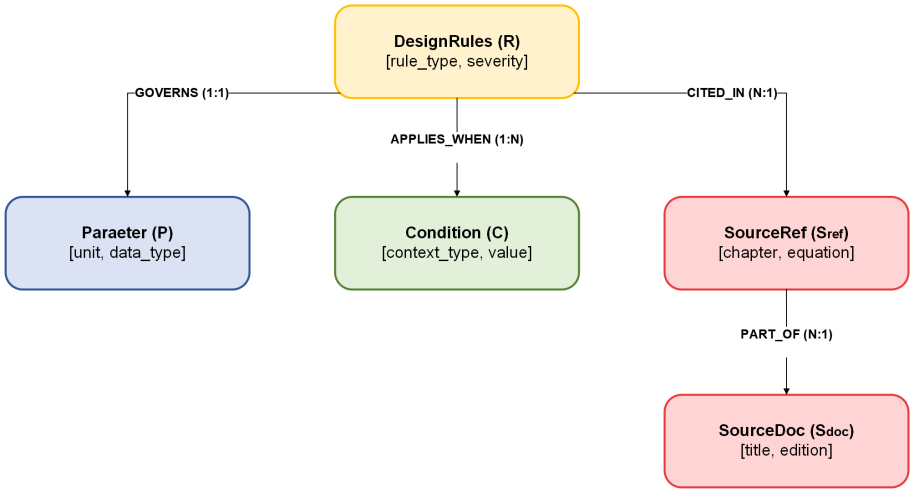


Figure 3.4: The KG Ontology. Rules (Yellow) act as the central connectors, linking engineering Parameters (Blue) to their contextual Conditions (Green) and authoritative Sources (Red).

- **Condition (C):** Represents the environmental or operational context. These nodes act as logical gates, dynamically determining when specific rules are active.
- **Provenance (S):** Composed of `SourceDoc` (e.g., HCM 7th Ed.) and `SourceRef` (e.g., Chapter 15, Eq 15-3). These nodes guarantee auditability, ensuring every validation error can be instantly traced to a specific legal or technical citation.

Semantic Relationships ($\mathcal{E}$)

The edges define the inferential capabilities of the system:

- `(R)-[:VALIDATES]->(P)`: Links a logic rule to the specific parameter it governs.
- `(R)-[:REQUIRES]->(C)`: Establishes conditional logic. A rule node possessing this edge is evaluated only if the input scenario perfectly matches the connected Condition.

- `(P)-[:AFFECTS]->(P)`: Defines downstream dependencies. This allows the system to instantly invalidate derived metrics the moment a root parameter changes.
- `(R)-[:CITED_IN]->(S)`: Provides the definitive audit trail, linking mathematical logic directly to the text of the official manual.

Formal Constraint Definitions

Within this ontological structure, the graph is populated with specific **Semantic Validator constraints (SF-)** derived directly from the HCM (7th Edition) and AASHTO design standards. These constraints function as the instance layer of the ontology, ensuring the logical robustness of the entire framework. They are categorized into three distinct classes:

- **Geometric Validity (SF-001, SF-002)**: Enforces physical dimensional limits calibrated to the empirical scope of the manual (e.g., $9 \leq$ Lane Width ≤ 12 ft).
- **Operational Logic (SF-003, SF-004)**: Validates discrete categorical variables. This ensures that facility classifications (e.g., Passing Zones) strictly align with their surrounding geometric context.
- **Physics-Based Safety (SF-005)**: A complex relational constraint enforcing the non-linear relationship between design speed (V) and minimum curve radius ($R_{\min}$), dictated by centripetal force and pavement friction.

3.4.2 The Validation Execution Pipeline

The operational core of the Semantic Validator is the pipeline itself. Unlike conventional input validation, which is typically hardcoded directly into procedural scripts (e.g., `if x < 0`), CrossTraffic delegates all validation to a dynamic, graph-based rule evaluation process.

The pipeline executes four distinct phases for every single computation request:

Table 3.1: Five Semantic Validator Constraints (Two-Lane Highway)

ID	Parameter	Valid Range	Source
SF-001	Lane Width	9–12 ft	HCM 7th Ed.
SF-002	Shoulder Width	0–8 ft	HCM 7th Ed.
SF-003	Horizontal Class	$0, 1, 2, 3, 4, 5$	HCM 7th Ed.
SF-004	Passing Type	Constrained, Zone, Lane	HCM 7th Ed.
SF-005	Design Radius	$R \geq R_{min}(V_{design})$	AASHTO Green Book

1. **Semantic Mapping:** The incoming JSON payload is parsed, and all keys are mapped to `Parameter` nodes (P) within the ontology. Unrecognized parameters are immediately flagged.

2. **Context Resolution:** The system evaluates the global state (e.g., `FacilityType`) against `Condition` nodes (C) to mathematically determine the active rule set $R_{active} \subset R_{total}$.

3. **Predicate Evaluation:** For each active rule $r \in R_{active}$, the system evaluates the logical predicate $f_r(p)$ against the provided input value p.

4. **Enforcement:** If $\forall r, f_r(p) =$ True, the validated inputs are safely serialized into Rust structs. If $\exists r, f_r(p) =$ False, the pipeline halts instantly and returns a highly structured `SemanticException`.

Algorithm 1 details this logic. Notably, the retrieval function GetActiveRule effectively pre-filters the graph. It returns only those rules where the connected `Condition` node matches the input context **I**, drastically optimizing the verification process for real-time execution.

Algorithm 1: Semantic Validator Execution Logic

Input: User Input Vector **I**, Knowledge Graph $G(\mathcal{V}, \mathcal{E})$

$Errors \leftarrow \emptyset$;
foreach *parameter* $p \in \mathbf{I}$ **do**
    $Node_p \leftarrow$ FindParameterNode$(G, p.key)$;
    $ActiveRules \leftarrow$ GetActiveRules$(Node_p, \mathbf{I})$;
    foreach *rule* $r \in ActiveRules$ **do**
        if $\neg$*Evaluate*$(r, p.value)$ **then**
            $Citation \leftarrow$ TraverseSource(r);
            $Errors$.push(FormatError$(r, Citation)$);
if $Errors \neq \emptyset$ **then**
    return Reject(400 Bad Request, $Errors$);
else
    return Proceed(200 OK, ExecuteRustCore(**I**));

3.5 Experimental Validation and Results

To evaluate the operational feasibility and effectiveness of CrossTraffic as a unified knowledge management framework, four distinct validation studies were conducted. These studies rigorously assess the framework's ability to ensure operational consistency, logical robustness, digital-twin scalability, and generative-AI reliability.

3.5.1 Ground Truth and Operational Consistency

A ground-truth baseline was first established to verify that the framework consistently executes HCM equations accurately across entirely heterogeneous user interfaces.

Case Study: River Falls Bypass

A real-world corridor was selected for evaluation: WIS 35/WIS 65 along the River Falls Bypass (Wisconsin). This corridor is characterized by moderate geometric complexity and highly variable passing opportunities (Fig. 3.5). The analysis strictly follows HCM Chapter 15 guidelines. It models five consecutive segments featuring mixed passing-constrained and passing zones, alongside horizontal curves requiring superelevation adjustments.

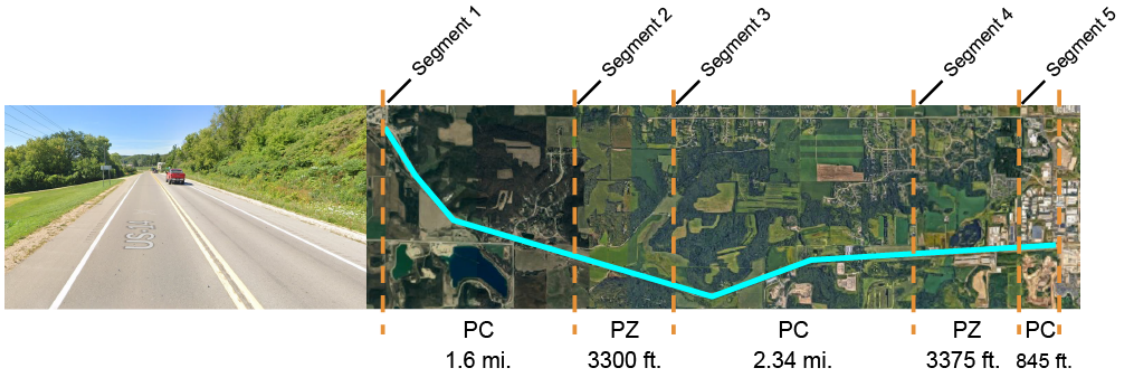


Figure 3.5: Case study visualization on WIS 35/WIS 65 along the River Falls Bypass

Inference Consistency Verification

To evaluate cross-interface consistency and reproducibility, the exact same operational analysis was performed using both the deterministic Web Calculator (WASM) and the probabilistic LLM-based semantic interface integrated via MCP.

As summarized in Table 3.2, the results overwhelmingly confirm the efficacy of the single source-of-truth architecture. Despite utilizing fundamentally different interaction modalities, both interfaces produced an identical overall level of service (LOS C). Minor discrepancies in average speed (AS) and follower density (FD) (all within 0.05) are safely attributed to standard floating-point rounding differences between the browser-based runtime and the server-side execution environment. These results confirm that the shared computational core and semantic

Table 3.2: Segment-Level and Overall LOS Comparison between Web Calculator (WASM) and LLM Interface

Segment	Web Calculator (WASM)				LLM Interface			
	AS (mph)	PF (%)	FD (fol/mi)	LOS	AS (mph)	PF (%)	FD (fol/mi)	LOS
1	59.17	59.36	5.41	C	59.33	53.8	5.35	C
2	59.90	47.98	4.32	C	54.14	43.0	4.90	C
3	56.70	56.59	5.38	C	56.56	51.3	5.15	C
4	56.90	48.04	4.32	C	54.19	43.0	4.92	C
5	56.76	55.10	5.23	C	55.07	50.0	5.02	C
Overall			**5.09**	**C**			**5.05**	**C**

interface successfully produce consistent, replicable analytical results across the entire ecosystem.

3.5.2 Logical Robustness and Constraint Enforcement

To quantify the regulatory capacity of the semantic validator, a severe stress test was conducted utilizing a stochastic adversarial generator. Instead of relying on randomized standard testing, this generator employed aggressive boundary value analysis. It specifically targeted the edges of geometric validity where software systems most frequently fail.

The generator produced $N = 1000$ synthetic test vectors. This dataset included severe boundary attacks (e.g., 12.01 ft lanes) and complex combinatorial conflicts (e.g., exceptionally high speeds paired with impossibly tight radii). The resulting payload contained a mix of perfectly compliant designs alongside diverse violations, including negative dimensions, excessive grades, and physics-incompatible speed-radius pairs.

As detailed in Table 3.3, the framework successfully rejected 100% of invalid engineering inputs with absolutely zero observed false positives. Furthermore, the median execution overhead was logged at an astonishing 0.002 ms per check. This explicitly confirms that the KG-based validator can be safely integrated directly into high-frequency digital twin simulation loops without introducing any measurable computational latency.

Table 3.3: Confusion Matrix of Semantic Validator Stress Test

Metric	Value	Implications for Engineering Safety
True Positives	740	The system effectively stops dangerous designs
True Negatives	260	The system does not hinder valid workflows
False Negatives	0	Critical safety success without illegal designs passed
F1 Score	1.00	Perfect deterministic reliability

3.5.3 Scalability and Heterogeneous Data Validation

To verify scalability, the evaluation scope was vastly expanded to include industry-standard digital-twin assets. The evaluation utilized the complete OpenDRIVE asset suite from the CARLA Simulator (Town01 through Town07)[1]. This specific dataset comprises $1,347$ unique road segments and demands $14,846$ individual parameter checks.

Table 3.4: Digital Twin Validation Results: CARLA Asset Suite ($N = 7$ Towns)

Asset ID	**Roads**	**Params**	**Valid**	**Invalid**	**Pass Rate**
Town01 (Urban)	98	715	409	306	57.20%
Town02 (Urban)	68	824	452	372	54.85%
Town03 (Mixed)	279	2,440	2,094	346	85.82%
Town04 (Highway)	242	2,708	2,433	275	**89.84%**
Town05 (Urban/Hwy)	259	3,734	3,407	327	**91.24%**
Town06 (Highway)	167	2,093	1,927	166	**92.07%**
Town07 (Rural)	234	2,332	2,009	323	86.15%
TOTAL	**1,347**	**14,846**	**12,731**	**2,115**	**85.75%**

The validation results (Table 3.4) effectively demonstrate the ability of the semantic validator to cleanly distinguish between merely geometrically valid road layouts and specific facilities that actually comply with strict U.S. design and operational standards.

[1]CARLA Assets on GitHub: `https://github.com/carla-simulator/opendrive-test-files/tree/master/OpenDrive`

- **Urban Non-Compliance:** Town01 and Town02 exhibited surprisingly low pass rates ($\approx 55\%$) due to the liberal use of $4.0\,\text{m}$ ($13.12\,\text{ft}$) lane widths designed to provide generous safety buffers for autonomous vehicle agents. While perfectly acceptable for theoretical robotics simulation, these dimensions flagrantly violate HCM Constraint `SF-001` (Max $12\,\text{ft}$). Under HCM logic, excessive width induces unsafe, high-speed driving behaviors. Additionally, the dense urban grid layout introduces exceptionally tight curve radii ($< 200\,\text{ft}$). These are inherently incompatible with the default $55\,\text{mph}$ highway classification logic, triggering immediate `SF-005` (physics-based safety) violations.

- **Highway Compliance:** Conversely, highway-focused maps (Town03-Town07) achieved exceptionally high compliance ($> 90\%$). The standard European highway lane width ($3.5\,\text{m} \approx 11.5\,\text{ft}$) falls perfectly within the required HCM valid range ($9 - 12\,\text{ft}$), and the much larger curve radii effortlessly satisfy AASHTO safety envelopes.

This large-scale experiment proves that the KG does not merely ingest static data. It actively evaluates data within a strict regulatory and operational context. For example, the system correctly and automatically identified that while Town01 represents a structurally valid road layout for a robot, it legally does not constitute a valid two-lane highway facility under U.S. design standards.

3.5.4 Model-Agnostic Performance Validation

Finally, a comparative evaluation assessed whether the framework truly mitigates hallucination and analytical inconsistency in generative-AI-based transportation analysis. Four distinct, state-of-the-art agent architectures (GPT-5.2, Claude 4.5 Sonnet, Gemini 3.0 Pro, and Claude 4.5 Opus) were tasked with performing the complex River Falls case study.

In this study, *context-based agents* refer to models augmented exclusively with uploaded PDF content. *MCP-integrated agents* represent models combining RAG

methodologies with direct tool invocation through the CrossTraffic MCP server and the external semantic validation layer.

Quantitative Convergence

Performance was quantified by measuring the Mean Absolute Error (MAE) of computed traffic metrics (e.g., flow rate, density, speed) strictly against the ground truth established in Section 3.5.1.

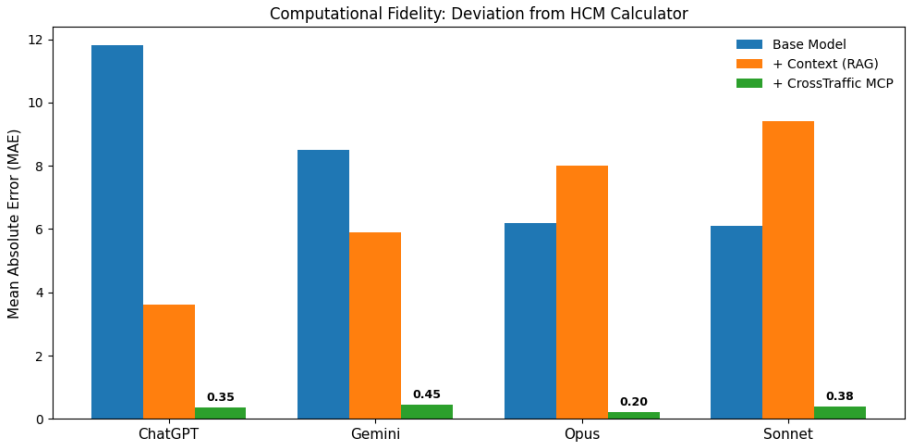


Figure 3.6: Impact of framework augmentation on computational error. Blue bars: base computational model with Rust-based calculator. Orange bars: PDF context augmentation for agents. Green bars: CrossTraffic MCP integration with agents.

As illustrated in Fig. 3.6, PDF-based context augmentation exhibited highly unstable performance. Specifically, providing an unstructured PDF context to Claude 4.5 Sonnet actually increased the mathematical error from 6.1 to 9.4. This proves that foundation models struggle massively to reliably interpret and execute multi-step procedural engineering text. Conversely, all MCP-integrated agents converged to near-zero numerical error (MAE < 0.50). This success was largely independent of the specific underlying language model, proving the architecture is model-agnostic.

Qualitative Methodological Fidelity

Beyond raw numerical accuracy, the reasoning quality of the engineering analysis was assessed. A strict rubric (Table 3.5) was defined to score agent responses on a scale of 1 (Poor) to 3 (Excellent) when provided with identical prompts.

Table 3.5: Qualitative Assessment Criteria for Engineering Agents

Dimension	Definition of Excellent (Score 3)
Methodological Fidelity	Explicitly references specific HCM chapters/equations rather than generic rules of thumb.
Parameter Extraction	Precisely maps user text to strongly-typed structs, capturing secondary details (e.g., superelevation).
Analytical Orchestration	Provides a rigorous trace of the sequential analysis, linking specific library functions directly to results.
Hallucination Mitigation	Strictly refuses to estimate missing data, forcefully requesting clarification instead of guessing.
Scientific Interpretability	Provides key technical findings clearly explaining why the facility performs at its current LOS.

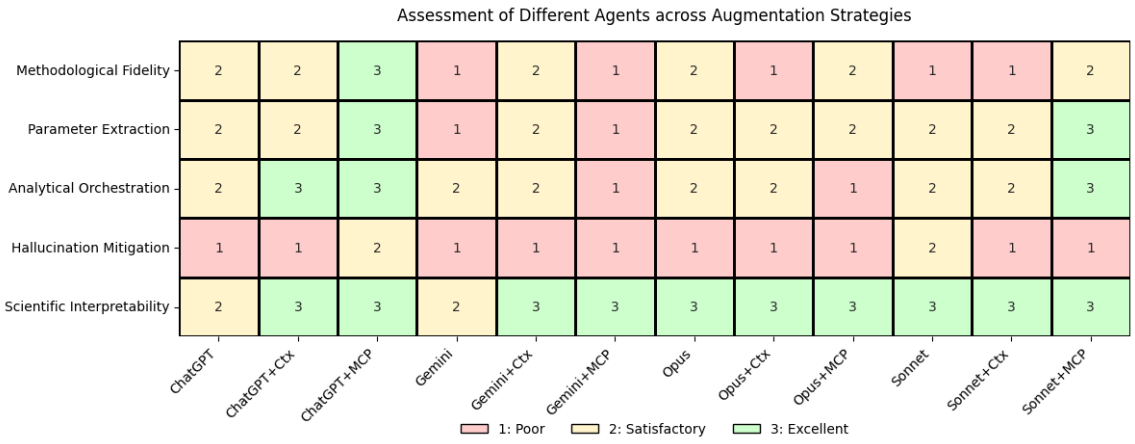


Figure 3.7: Qualitative scoring heatmap across multiple agents with different augmentation strategies. Lower numbers colored in red indicate poor performance on the metric, and higher numbers with green indicate better performance.

The heatmap results (Fig. 3.7) reveal a sharp performance contrast. While PDF-based context-augmented agents showed slightly improved parameter extraction,

they failed disastrously in analytical orchestration. They frequently hallucinated numerical answers without providing any explicit computational trace. In complete contrast, MCP-integrated agents consistently achieved exceptional scores in methodological fidelity. They reliably generated highly structured execution traces perfectly aligned with HCM procedures.

A minor trade-off was observed regarding contextual understanding. Because MCP-integrated agents prioritize structured tool execution over creative inference, they occasionally underutilized implicit or underspecified user inputs when compared to context-only agents. In a small number of edge cases, the strict schema requirements forced models to aggressively infer missing parameters rather than politely requesting user clarification. Nevertheless, scientific interpretability remained incredibly strong. This was particularly true when agents were explicitly prompted to provide step-by-step validation of their intermediate assumptions.

3.6 Discussion

The development and deployment of CrossTraffic represents a concrete, verifiable step toward modernizing transportation analysis workflows. It explicitly demonstrates the feasibility of transforming complex, static engineering manuals into a unified, continuously deployable knowledge management system. Beyond validating the sheer technical feasibility of the architecture, these findings reveal fundamental engineering trade-offs regarding API design, agent orchestration, and organizational adoption that will directly dictate the practical deployment of future AI-assisted engineering systems.

3.6.1 API Granularity and the Agentic Trade-off

Decomposing monolithic HCM procedures into reusable, discrete computational endpoints brilliantly enables modularity and cross-platform deployment. However, it introduces an inherent trade-off between endpoint granularity and orchestration complexity. During this evaluation, both baseline and MCP-integrated agents

occasionally stumbled during analytical orchestration (Fig. 3.7). These failures were primarily caused by incorrect endpoint sequencing, the skipping of necessary intermediate steps, and the inconsistent handling of intermediate states across highly complex, multi-step workflows.

The effective deployment of AI agents requires a robust extraction layer. This layer must translate deeply heterogeneous user inputs into perfectly structured parameters. It simultaneously requires an evaluation layer that presents analytical results in a professionally interpretable, engineering-grade format (Shankar et al. (2024); Polak and Morgan (2024)). In the current implementation, parameter extraction and result presentation are handled entirely through prompted LLM roles. While generally effective, observed failure modes included implicit facility misclassification, unit ambiguity, and incorrect parameter binding when semantically similar variables coexisted in the prompt. These findings strongly suggest that future agent controllers must employ rigorous, dependency-aware planning over the API and Knowledge Graph, rather than relying naively on prompt-based endpoint chaining (Huang et al. (2024)).

3.6.2 The Necessity of Semantic Guardrails

A central, highly critical empirical finding of this research is that RAG does absolutely not guarantee engineering validity. As proven in Fig. 3.6, providing unstructured PDF context to foundation models frequently degraded computational accuracy and actively increased numerical error. This produced procedurally plausible but mathematically invalid execution paths. This phenomenon is defined here as *semantic hallucination*, where agents misapply equations with extreme confidence or completely omit mandatory analytical steps.

In stark contrast, MCP-integrated agents consistently converged to near-zero error across every evaluated foundation model ($\text{MAE} < 0.50$). This proves definitively that executable mathematical constraints and KG-based validation are an absolute requirement for procedural reliability. While RAG remains highly valuable for document retrieval and high-level conceptual explanation, it is entirely

insufficient for enforcing normative analytical workflows (Gupta et al. (2024)). These results, therefore, strongly advocate for hybrid context-infused architectures. In these systems, LLMs flexibly interpret user intent, while highly structured computational and ontological layers strictly constrain and regulate all permissible mathematical operations.

A corresponding, minor limitation of this architecture is a phenomenon termed the *reading gap*. Schema-constrained agents occasionally fail to exploit implicit or underspecified user context (Fig. 3.7). Because only explicitly provided or formally represented parameters can be passed to the validated computational tools, contextual assumptions expressed loosely in natural language are frequently ignored, as they cannot be safely translated into executable mathematical inputs.

3.6.3 Open Source Significance for Transportation

Finally, this work explicitly highlights the persistent, damaging tension between proprietary black-box software and the fundamental scientific requirement for reproducibility (Wood and Schalkwyk (2025)). Although the base equations underlying most transportation analysis tools are publicly documented, their actual operational implementations are overwhelmingly embedded within closed-source, proprietary systems. This practice creates massive, insurmountable barriers for researchers and practitioners seeking to inspect, verify, or safely extend analytical methodologies.

CrossTraffic does not single-handedly resolve the broader institutional and standardization challenges that currently paralyze open transportation knowledge infrastructures. Organizations responsible for developing official manuals must constantly balance the protection of proprietary design processes with the urgent need for open, community-driven representations of foundational methodologies. Furthermore, widely adopted, machine-readable standards for representing transportation equations, procedural rules, and regulatory logic are still entirely lacking across the industry. Consequently, achieving large-scale interoperability and sustained knowledge exchange across disparate tools and institutions remains

a massive open challenge, sitting just beyond the scope of the current framework.

3.7 Conclusions

This chapter presented **CrossTraffic**, a unified framework for intelligent transportation knowledge management designed specifically to address the persistent dangers of fragmented, static, and siloed engineering tools. By establishing an executable architecture where transportation procedures are rigorously represented as both validated computational functions and formal semantic constraints, CrossTraffic serves as a single, unassailable source of analytical logic.

Furthermore, the integration of Large Language Models (LLMs) into this validated execution environment demonstrated massive, quantifiable improvements in computational fidelity and methodological correctness when compared against context-only, retrieval-based approaches. This ontology-driven decision-support layer successfully bridges probabilistic natural-language interaction with strict, normative engineering workflows. It ensures fully reproducible and perfectly auditable results across entirely heterogeneous interfaces, including web calculators, microscopic simulation pipelines, and conversational AI agents.

While the current implementation successfully proves the feasibility of this architecture strictly within the scope of the Highway Capacity Manual, sustainable progress in digital twin ecosystems requires expanding this foundational logic to encompass much broader regulatory manuals. Establishing standardized, open-source representations of procedural logic driven by deep collaboration among software engineers and transportation researchers will remain absolutely critical for fostering innovation while preserving strict regulatory accountability as these digital systems scale.

Therefore, CrossTraffic equips the digital twin with the codified rules necessary to formally validate inferred infrastructure states against established engineering principles. With the physical roadway perceived (Chapter 2) and the operational rules codified (Chapter 3), the system now possesses both structural observational capability and rigorous methodological logic. However, translating these static,

descriptive assessments into proactive, dynamic traffic management requires a mathematical representation of how the infrastructure fundamentally functions. To achieve a fully interactive and constructive environment, the subsequent chapter introduces a behavior-grounded representation learning framework. This framework enables the system to actively manipulate semantic lane roles and evaluate dynamic infrastructure modifications (Chapter 4).

4 BEHAVIOR-GROUNDED LANE REPRESENTATION FOR DIGITAL TWIN SUPPORTED TRAFFIC MANAGEMENT

The dissertation has thus far established two fundamental capabilities for next-generation traffic digital twins. The capacity to observe roadway conditions through sensing, and the ability to reason over transportation operations through structured computational logic. However, observation and reasoning alone are insufficient to sustain an interactive, constructive traffic management paradigm. The critical missing link is a semantic representation layer that bridges the system's perceptual capabilities with its capacity for interpretation, comparison, and infrastructural modification. Without this semantic grounding, lanes remain mere geometric abstractions in perception modules and arbitrary symbolic entities within operational models, preventing their utility as functional components. To resolve this, this chapter introduces **GeoLaneRep**, a behavior-grounded representation framework that models each lane as an interpretable operational entity, defined through the joint encoding of its spatial geometry, empirical traffic behavior, and contextual role. Consequently, this chapter provides the semantic interface required to monitor, compare, and ultimately reshape lane-level traffic scenes within a closed-loop digital twin ecosystem.

4.1 Introduction

Digital twins have emerged as a promising system for advanced traffic management, integrating sensing, simulation, and decision-support capabilities within a cohesive virtual environment (Xu et al. (2023); Gu et al. (2026)). These systems can continuously monitor live traffic states, evaluate control strategies, and reduce manual overhead in scenario generation. However, early traffic digital twins operated primarily as descriptive platforms for visualization and state synchronization, and recent traffic management requires more than passive observation. It demands constructive digital twins capable of interpreting operational shifts and supporting

rapid, intervention-oriented reasoning (Irfan et al. (2024); Dasgupta et al. (2024)).

This requirement is particularly critical at the lane level, the operational layer at which traffic management interventions are defined and executed. Although contemporary digital twins replicate road geometries and vehicle trajectories at high fidelity, structural reproduction alone is insufficient for operational reasoning (Zhou et al. (2025); Miller et al. (2021)). Strategic interventions, such as dynamic lane reassignment, merge coordination, turn regulation, and work-zone configuration, depend intrinsically on understanding the specific functional role each lane plays within the broader traffic ecosystem (Feng et al. (2025)). Therefore, a constructive digital twin must conceptualize each lane not merely as a physical boundary but as a dynamic, interpretable component whose operational utility can be analyzed and compared.

Recent advancements in representation learning present a viable methodology for bridging this semantic gap. A growing body of literature demonstrates that learned neural embeddings have been shown to extract spatiotemporal structures from high-dimensional transportation data. For instance, Trajectory Representation Learning (TRL) improves spatial similarity computations and travel-time estimation (Wang et al. (2024a)), while graph-based architectures encode higher-order topological dependencies across macroscopic road networks (Mao et al. (2022); Zhang and Long (2023)). Furthermore, integrated embedding strategies model multi-agent interaction ego-trajectory predictions (Hou et al. (2023)). These efforts confirm that neural representations capture rich traffic dynamics, but their focus has remained disproportionately bounded to moving agents or macro-scale network topologies, largely bypassing the lane itself as the unit of representation.

This oversight is particularly detrimental to the advancement of traffic digital twins, which require robust lane-level operational reasoning to complement geometric reconstruction. Prevailing lane representations are strictly topological or geometric, encoding centerlines, boundaries, and static connectivity. Such features are indispensable for high-definition map generation and autonomous navigation, but do not capture the functional semantics of a lane during live traffic operations (Arman and Tampère (2021)). Two lanes possessing identical geometric parameters

may serve profoundly disparate operational purposes, ranging from uninterrupted through-movement to auxiliary flow or merge assistance. Crucially, these functional roles are not static as they emerge dynamically from the surrounding lane topology, fluctuating traffic demand, and the continuous trajectory interactions of individual vehicles (Poggenhans et al. (2018); Naumann et al. (2023)). Consequently, current methodologies lack a representation that synthesizes static geometric structure with observed traffic behavior, thereby precluding rigorous semantic comparison across lanes and across scenes.

The contribution of this work is a behavior-aware lane representation that closes this gap, together with the training objectives and downstream tasks that exercise it:

- We introduce **GeoLaneRep**, a representation learning framework that produces a single shared lane embedding by jointly encoding spatial geometry, observed trajectories, and operational descriptors with three parallel encoders fused via cross-lane multi-head attention.
- We propose a joint training objective that combines contrastive cross-camera alignment, auxiliary role supervision, and temporal anomaly detection, and we show empirically that this joint formulation outperforms two-stage and single-objective variants on every metric we measure.
- We demonstrate that the resulting embedding supports three downstream tasks through the same encoder weights, including zero-shot cross-camera lane matching, per-window anomaly detection, and behavior-conditioned geometry generation, establishing a semantic interface between roadside observations and lane-level digital twin tasks.

4.2 Related Work

4.2.1 Digital Twins for Traffic Management

Digital twins in transportation are conceptualized as dynamic virtual counterparts of physical mobility networks, integrating sensing, communication, and computational modeling to support monitoring and decision making (Wang et al. (2022, 2024b)). Recent methodological advances have improved dynamic traffic synchronization (Kušić et al. (2023)) and simulation-based analysis (Perna et al. (2025)). Nonetheless, contemporary transportation digital twins remain anchored in state alignment, visual replication, and macroscopic calibration (Luo et al. (2025); Bongomin et al. (2025)).

This paradigm presents a critical limitation: effective traffic management relies not merely on continuous observation, but on interpreting the functional behavior of the infrastructure as it evolves. Existing frameworks, including those designed to estimate operational measures (Xu et al. (2025b)) and synchronize lane geometry (Tamaru et al. (2025)), are predominantly observational. They maintain parity between physical and virtual states, but they lack a formal mechanism to encode lanes as actionable semantic units. Advancing toward a constructive digital twin requires a cohesive representation layer that consolidates structural geometry, traffic dynamics, and operational semantics. Addressing this methodological gap forms the primary motivation for this study.

4.2.2 Semantic Understanding of Traffic Operations

While conventional traffic models capture vehicle kinematics, they rarely yield a learned semantic representation of lane-level operations. Recent work has sought to augment traffic comprehension by deploying Large Vision-Language Models (VLMs) (Rivera et al. (2025); Luo et al. (2024)). For instance, CityLLaVA (Duan et al. (2024)) leverages roadside camera feeds for question-answering on urban scenes, whereas MAPLM (Cao et al. (2024)) combines multi-view imagery, bird's-eye-view projections, and high-definition maps for spatial reasoning. From the

ego-vehicle perspective, Liao et al. (2024a) fuses LiDAR and video within a VLM architecture to parse driving environments. Despite their formidable descriptive capabilities, these frameworks predominantly generate textual or symbolic outputs, such as natural language narratives, bounding boxes, and scene descriptions, which remain computationally prohibitive to integrate into the continuous mathematical control loops required for precise traffic management.

Concurrently, recent work argues that actionable traffic semantics must extend beyond visible geometry. The PAMR framework (Liang et al. (2026)) jointly models geometric configurations and semantic traffic rules for robust navigation. Similarly, Fu et al. (2024a) demonstrates that explicit reasoning over lane structure enhances scene analysis, particularly in pedestrian-heavy scenarios. At a macroscopic scale, Xu et al. (2025a) incorporates road connectivity and regional proximity to refine cross-city traffic forecasting. Building upon these insights, this research pivots from descriptive, VLM-driven scene interpretation toward infrastructure-centric representation learning that embeds behavioral semantics directly into the representation.

4.2.3 Representation Learning for Transportation and Lane Structure

Representation learning is increasingly used in transportation engineering to distill latent structure from high-dimensional traffic data (Alahi (2026)). Trajectory Representation Learning (TRL) frameworks such as START (Jiang et al. (2023)) jointly encode trajectories and road segments for self-supervised recovery, while GTR (Wang et al. (2025)) and TRACK (Han et al. (2025)) leverage spatio-temporal encoders and co-attentional transformers to capture intrinsic trajectory dynamics. These methods operate primarily at the network scale or in unconstrained motion settings, treating the lane as a contextual feature rather than the unit of representation.

At the lane level, several studies use representation learning to infer lane topology. Li et al. (2025a) models lanes as nodes within a unified spatial topology to pre-

dict nonlinear fluctuations in traffic flow. Complementing this, advanced vectorized mapping techniques, such as MapTR (Liao et al. (2022)), extract structured lane networks directly from onboard sensors. Furthermore, generative approaches like CDSTE (Lei et al. (2024)) and RoadDiff (Li et al. (2025b)) employ diffusion-based modules to infer fine-grained lane traffic states from macroscopic observations.

While these frameworks demonstrate that lane-specific topological reasoning improves predictive accuracy, their learned embeddings are optimized for narrow, task-dependent objectives. Consequently, they fail to achieve the broader mandate of representation learning: establishing a generalized methodology that uncovers underlying data structures to support versatile processing (Alahi (2026)). To resolve this methodological shortcoming, GeoLaneRep synthesizes geometric frameworks, trajectory behaviors, and operational descriptors into a single shared embedding, transforming lanes into semantic entities that are computationally retrievable, comparable, and amenable to behavior-conditioned synthesis.

4.3 Problem Setup

The GeoLaneRep framework is structured as a three-stage pipeline: *observe*, *encode*, and *generate*. It represents a lane not as a physical shape alone but as a functional component defined jointly by its observed traffic behavior, geometric structure, and operational context. This unified representation is what lets the framework support multiple downstream tasks, zero-shot cross-camera matching, per-window anomaly detection, and behavior-conditioned lane generation, through a single shared embedding.

4.3.1 Problem Formulation and Variable Definition

We formulate each lane as a composite of its geometry, observed trajectories, summary traffic statistics, and structural role. For a given lane i,

$$x_i = (g_i,\ \mathcal{T}_i,\ s_i,\ r_i), \tag{4.1}$$

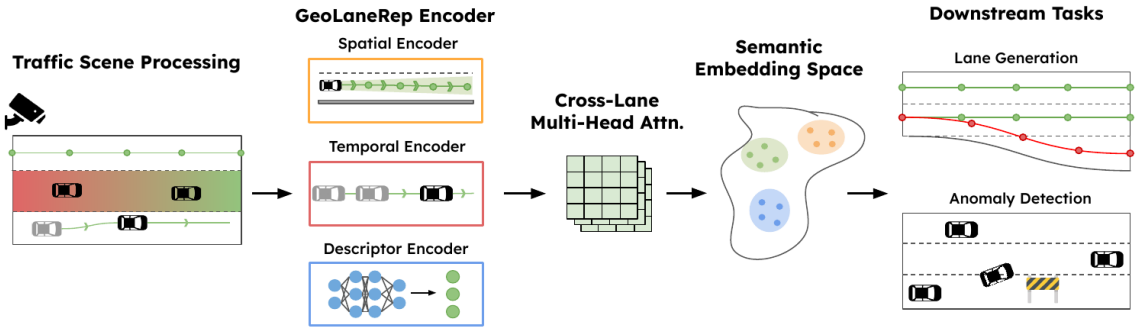


Figure 4.1: Overview of the GeoLaneRep pipeline. Roadside observations are converted into per-lane geometry, trajectories, and descriptors (left). Three parallel encoders, spatial, temporal, and descriptor, produce per-lane representations that are fused via cross-lane multi-head attention into a shared semantic embedding space that supports multiple downstream tasks.

where g_i is the physical lane geometry. $\mathcal{T}_i$ is the set of vehicle trajectories assigned to the lane by roadside sensing, s_i holds aggregated traffic statistics, and r_i encodes the lane's structural role within its lane group.

The trajectory set is

$$\mathcal{T}_i = \{\boldsymbol{\tau}_i^{(n)}\}_{n=1}^{N_i}, \qquad \boldsymbol{\tau}_i^{(n)} = \{(x_t, y_t)\}_{t=1}^{T_n}, \tag{4.2}$$

i.e., each lane carries N_i tracklets, and each tracklet is a sequence of 2-D image-normalized waypoints over T_n frames. Speed and heading are not stored per timestep; they are derived during preprocessing and enter the model only through the aggregate descriptor s_i. To capture short-term traffic variation without overwhelming the model with noise, we further partition each lane's trajectory stream into discrete temporal windows before feature construction.

Beyond raw movement data, the geometric component g_i provides a structural prior, formatted as a centerline polyline $g_i \in \mathbb{R}^{K\times 2}$, a sequence of K waypoints in the same image-normalized coordinate frame. The descriptor s_i summarizes the traffic properties derived from the assigned tracklets. The role vector r_i captures interpretable structural signals such as the lane's lateral rank within its group and

whether it sits at an edge of the roadway.

For a temporal window w, the per-window trajectory tensor is denoted as $\mathcal{T}_{i,w} \in \mathbb{R}^{N_{i,w} \times K \times 2}$, consisting of $N_{i,w}$ tracklets that have been arc-length resampled to K spatial points each. The corresponding traffic descriptor $s_{i,w} \in \mathbb{R}^4$ records four lane-level quantities computed from those tracklets: mean speed, mean curvature, mean lateral offset from the lane centerline, and a normalized trajectory count used as a density proxy. The role vector $r_i \in \mathbb{R}^5$ encodes lateral rank, leftmost and rightmost edge flags, a successor flag indicating whether the lane continues into a downstream segment, and the group's lane count.

Our objective is to learn a mapping

$$f : x_i \;\mapsto\; z_i, \qquad z_i \in \mathbb{R}^d, \tag{4.3}$$

with $d = 128$ in our implementation. The latent embedding z_i is intended to preserve behavioral semantics, properly trained, it places lanes with similar operational roles close together even when their raw geometries or camera viewpoints differ substantially. This makes the representation portable across the downstream tasks introduced in the experiments.

4.3.2 Input Construction and Preprocessing

Each lane instance is constructed from three primary information streams: static geometry, dynamic vehicle movement, and interpretable descriptors. The geometric stream provides the physical backbone of the lane. The trajectory stream captures how drivers actually interact with that structure, isolating realized behavior from idealized lane shape. The descriptor stream summarizes operational signals derived from those interactions, aggregate kinematics (s_i), and the lane's structural role within its group (r_i).

Constructing these inputs requires assigning trajectories to candidate lanes through spatial association with the known geometry within each lane group. We define a *lane group* as a set of adjacent lanes that share the same cross-section and local flow context (e.g., the four lanes of a single direction at a single intersection

approach). The group provides the basis for determining lateral rank, edge status, and the neighbor-aware features used by the cross-lane attention.

Observations are then organized into discrete temporal windows, preserving short-term traffic fluctuations while simultaneously filtering extraneous noise. Geometry is normalized into a lane group's shared image-normalized coordinate frame, and trajectory statistics are aggregated per window. Prior to encoding, the anchor geometry g_i is broadcast across the time dimension of the per-window stream, and the per-window dynamic statistics are concatenated with the static role vector to form a single unified descriptor:

$$x_{i,w}^{\text{stat}} = [\, s_{i,w} \parallel r_i \,] \in \mathbb{R}^9. \tag{4.4}$$

The preprocessing pipeline, therefore, yields aligned, multi-modal inputs in which each lane is represented as a coherent semantic unit – geometry, behavior, and structural context bound together – rather than as an isolated geometric line on a map.

4.4 GeoLaneRep Encoder

GeoLaneRep encodes the preprocessed lane inputs into a shared embedding space that captures lane-level behavioral semantics. To make latent similarity reflect operational similarity, the encoder is trained with structurally mined contrastive supervision across disparate camera views, supplemented by auxiliary role regression and temporal anomaly objectives that prevent representational collapse during long training runs.

4.4.1 Input Streams and Fusion

For lane i and temporal window w, the encoder consumes three inputs: (i) a static lane geometry polyline $g_i \in \mathbb{R}^{K\times 2}$, (ii) the set of trajectory polylines $\mathcal{T}_{i,w} \in \mathbb{R}^{N_{i,w}\times K\times 2}$ together with a validity mask $m_{i,w} \in \{0,1\}^{N_{i,w}}$ that flags which slots correspond

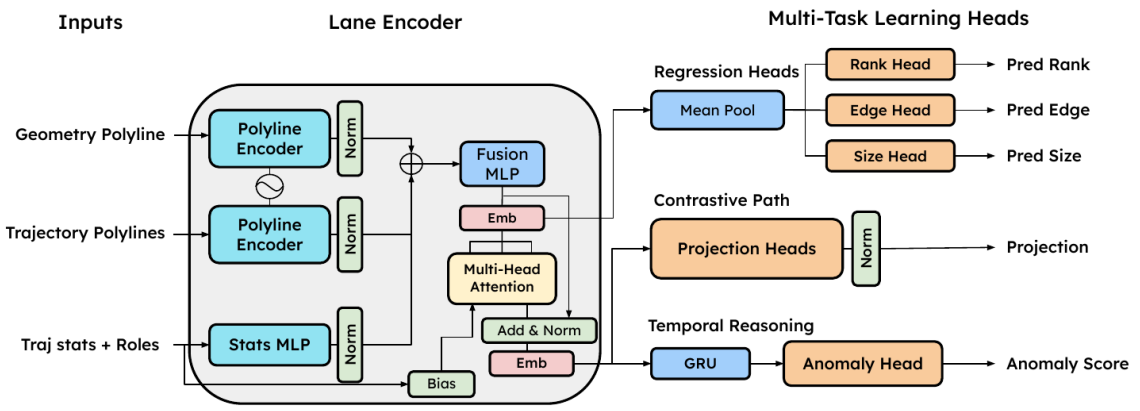


Figure 4.2: GeoLaneRep encoder architecture. The static lane geometry, the per-window set of assigned trajectories, and the fused stats–role descriptor pass through three parallel encoders that each emit a 64-dim embedding. The three embeddings are concatenated and passed through a fusion MLP to produce the per-window lane embedding, then averaged over valid temporal windows to obtain a per-lane embedding.

to real tracklets, and (iii) a fused descriptor vector $x_{i,w}^{\text{stat}}$ formed by concatenating the four trajectory statistics $s_{i,w} \in \mathbb{R}^4$ with the five-dim structural role descriptor $r_i \in \mathbb{R}^5$.

The architecture processes these inputs through a geometry stream, a trajectory stream, and a descriptor stream, each emitting a 64-dim embedding. The three embeddings are then fused into a single per-window lane representation.

Geometry Stream. The geometry branch maps the static lane shape into a structural embedding,

$$f_i^g = f_{\text{geom}}(g_i) \in \mathbb{R}^{64}. \tag{4.5}$$

f_{geom} first applies a per-waypoint linear projection $\mathbb{R}^2 \to \mathbb{R}^{64}$, adds a fixed sinusoidal positional encoding PE_K to preserve spatial ordering, and runs the resulting sequence through a 2-layer Transformer encoder. The token sequence is mean-pooled across the K waypoitns to a single vector, after which a per-stream Batch Normalization is applied externally.

Trajectory Stream. The trajectory branch models how vehicles traverse the lane,

$$f^x_{i,w} = f_{\text{traj}}(\mathcal{T}_{i,w}, m_{i,w}) \in \mathbb{R}^{64}. \tag{4.6}$$

f_{traj} shares the geometry stream's per-polyline encoding scheme: each tracklet's K waypoints are projected positionally encoded, and Transformer-encoded to a 64-dim vector. The resulting per-tracklet embeddings are subsequently mean-pooled across the $N_{i,w}$ slots with validity mask $m_{i,w}$ used to zero out padded slots and to divide by the true valid count, so padding does not bias the mean. Per-stream Bach Normalization is applied externally.

Descriptor Stream. The third branch processes the interpretable descriptor, which carries operational signals, aggregate kinematics, edge membership, and lateral ordering, not easily recovered from raw polylines. The descriptor passes through a two-layer MLP,

$$f^s_{i,w} = f_{\text{desc}}\Big([s_{i,w} \parallel r_i]\Big) \in \mathbb{R}^{64}. \tag{4.7}$$

where f_{desc} has the form of a two-layer network mapping from $\mathbb{R}^9 \to \mathbb{R}^{64}$, with per-stream Batch Normalization and GELU activations so that all three streams share the same output normalization scheme.

Fusion. The three branch outputs are concatenated to form an intermediate representation $c_{i,w} = [\, f^g_i \parallel f^x_{i,w} \parallel f^s_{i,w} \,] \in \mathbb{R}^{192}$, which is projected to the final per-window embedding,

$$z_{i,w} = f_{\text{fuse}}(c_{i,w}) \in \mathbb{R}^{128}. \tag{4.8}$$

f_{fuse} expands the 192-dim concatenation to 256-dim intermediate space via a linear projection, applies GELU activation and dropout, and projects down to the final $\mathbb{R}^{128}$ embedding.

A global, per-lane embedding is obtained by averaging across W_i valid temporal windows of lane i,

$$\bar{z}_i = \frac{1}{W_i} \sum_{w=1}^{W_i} z_{i,w}, \tag{4.9}$$

This fusion design lets every lane be summarized by a single $\mathbb{R}^{128}$ vector while preserving per-window granularity for the temporal anomaly head, which consumes the per-window sequence $\{z_{i,w}\}_{w=1}^{W_i}$ directly.

4.4.2 Cross-Lane Attention

In multi-lane road segments, a lane's operational meaning is inherently relative. A passing lane is defined by its speed relative to adjacent traffic; an edge lane derives its role from the physical boundary. To capture these group-relative semantics, GeoLaneRep optionally routes the per-lane embeddings through a cross-lane attention module before downstream use.

Lanes that share a common group identifier are packed together for joint processing. For every ordered lane pair (i, j) within a group, three relative features are combined:

$$\phi_{ij} = \Big[\Delta_{\text{lateral}}(i,j),\ \Delta_{\text{speed}}(i,j),\ \rho_{\text{density}}(i,j)\Big], \tag{4.10}$$

where $\Delta_{\text{lateral}}(i,j) = s_i^{\text{lat}} - s_j^{\text{lat}}$ and $\Delta_{\text{speed}}(i,j) = s_i^{\text{spd}} - s_j^{\text{spd}}$ are signed differences between the per-lane mean lateral offset and mean speed of the assigned trajectories, while $\rho_{\text{density}}(i,j) = \frac{s_i^{\text{cnt}}}{s_j^{\text{cnt}}+\varepsilon}$ is the ratio of their normalized trajectory counts, clamped to $[-10, 10]$ to prevent numerical explosion.

The pairwise feature stack $\Phi \in \mathbb{R}^{|\mathcal{G}|\times|\mathcal{G}|\times 3}$ is projected linearly to a per-head bias of dimension equal to the number of attention heads, and added directly to the scaled dot-product attention scores. A multi-head self-attention block with a residual connection and layer normalization then yields the final group-aware embedding,

$$\tilde{z}_i = \text{LayerNorm}\Big(\bar{z}_i + f_{\text{MHA}}(\bar{z}_i, \mathcal{Z}_{\text{group}(i)}, \phi_{i,\cdot})\Big) \in \mathbb{R}^{128}. \tag{4.11}$$

where $\mathcal{Z}_{\text{group}(i)}$ denotes the set of per-lane embeddings for all lanes in the group containing lane i, and $\phi_{i,\cdot}$ collects the relative features between lane i and every group member. Padded slots produced during batched group packing are masked

out of both the attention scores and the output, so they exert no influence on the attended representation.

4.4.3 Multi-Task Learning

The encoder is trained under a multi-task regimen in which several heads supervise distinct facets of the learned representation. Rather than acting as isolated task-specific outputs, these heads shape the shared embedding space toward structural consistency, semantic interpretability, and temporal sensitivity.

Contrastive Alignment

To enforce cross-camera structural alignment, the group-aware embedding $\tilde{z}_i$ is mapped through projection head f_{proj} before being ℓ_2-normalized:

$$p_i = f_{\text{proj}}(\tilde{z}_i), \qquad \hat{p}_i = \frac{p_i}{\|p_i\|_2}. \tag{4.12}$$

f_{proj} operates as a two-layer multi-layer perceptron mapping $\mathbb{R}^{128} \to \mathbb{R}^{64}$. The trailing BatchNorm decorrelates batch statistics before ℓ_2 normalization, which lets cosine similarity reduce cleanly to the dot product $\hat{p}_i^\top \hat{p}_j$.

The normalized projection $\hat{p}_i$ is then optimized with the standard InfoNCE objective (Oord et al. (2018)), following the contrastive representation learning paradigm (**??**), with positive pairs mined structurally across cameras rather than through input augmentation. This pulls structurally similar lanes from distinct cameras together in the embedding space while pushing apart pairs that disagree in role. Following standard representation learning practice, this projection head is discarded at inference, so downstream tasks operate directly on the encoder embeddings.

Role Supervision

A purely contrastive objective leaves the embedding free to ignore structurally meaningful attributes, an outcome typically called *representation collapse*, where the

encoder finds a shortcut that minimizes the contrastive loss without preserving useful geometry. Three auxiliary heads provide direct supervision against this failure mode. They attach to the pre-attention per-lane embedding $\bar{z}_i$ rather than to the post-attention $\tilde{z}_i$, so the supervision targets a per-lane signal that is not yet contaminated by within-group mixing:

- **Lateral rank head** $f_{\text{rank}} \colon \mathbb{R}^{128} \to \mathbb{R}$, a two-layer MLP through a 32-dim hidden layer, predicts the lane's normalized lateral position within the lane group.
- **Edge flag head** $f_{\text{edge}} \colon \mathbb{R}^{128} \to \mathbb{R}^2$, a single linear projection, outputs logits for the leftmost and rightmost flags.
- **Group size head** $f_{\text{size}} \colon \mathbb{R}^{128} \to \mathbb{R}$, a linear projection, estimates the group's normalized lane count.

All three heads emit raw logits and are trained with BCE-with-logits against bounded $[0, 1]$ targets, which provides stronger gradients at extremes than MSE on a sigmoid-squashed prediction would.

The combined role loss is

$$\mathcal{L}_{\text{role}} = \mathcal{L}_{\text{rank}} + \mathcal{L}_{\text{edge}} + 0.5\,\mathcal{L}_{\text{size}}. \tag{4.13}$$

When cross-lane attention is active, an additional group-rank consistency term enforces that, within each group, the predicted ranks form a monotonic and roughly uniformly spaced sequence:

$$\mathcal{L}_{\text{group}} = \frac{1}{G} \sum_{g=1}^{G} \left\| \text{sort}\big(\sigma(\hat{r}_g)\big) - \text{linspace}(0, 1, n_g) \right\|_2^2, \tag{4.14}$$

where $\hat{r}_g$ collects the predicted rank logits for the n_g lanes in group g and σ is the sigmoid function.

Joint Training Objective

In the contrastive stage, the encoder loss combines the InfoNCE term with the role losses under a three-phase epoch schedule:

$$\mathcal{L}_{\mathrm{E}}(e) = w_{\mathrm{ctr}}(e)\,\mathcal{L}_{\mathrm{ctr}} + w_{\mathrm{role}}(e)\,\mathcal{L}_{\mathrm{role}}. \tag{4.15}$$

The schedule (with e/E the training fraction) is

$$(w_{\mathrm{ctr}}, w_{\mathrm{role}})(e) = \begin{cases} (0.3,\ 2.0) & e/E < 0.3 \quad \text{(role-dominant warm-up)} \\ (1.0,\ 1.0) & 0.3 \le e/E < 0.7 \quad \text{(balanced)} \\ (2.0,\ 0.5) & e/E \ge 0.7 \quad \text{(contrastive fine-tune)}. \end{cases} \tag{4.16}$$

The schedule prevents the contrastive and role objectives from colliding early in training: role supervision establishes an interpretable per-lane signal first, contrastive alignment then pulls cross-camera matches together, and a final contrastive-heavy phase sharpens retrieval quality.

The temporal branch learns from synthetic anomaly injection. A fraction of valid training windows are corrupted by simulated speed reductions, trajectory dropouts, or lateral deviations. The anomaly head is supervised by a validity-weighted binary cross-entropy loss,

$$\mathcal{L}_{\mathrm{temp}} = \frac{\sum_{i,w} v_{i,w}\,\mathrm{BCE}\big(a_{i,w}, y_{i,w}\big)}{\sum_{i,w} v_{i,w}}, \tag{4.17}$$

where $v_{i,w} \in \{0,1\}$ is the window validity indicator and $y_{i,w} \in \{0,1\}$ is the ground-truth anomaly label, so padded windows contribute neither numerator nor denominator, and the loss remains comparable across batches with different valid-window counts.

Under the joint training configuration, the encoder remains trainable through

both streams, and the total loss is

$$\mathcal{L} = \alpha\,\mathcal{L}_{\text{temp}} + \beta\,\mathcal{L}_{\text{E}}, \tag{4.18}$$

where α, β control the relative influence of the temporal and structural objective. Empirically, this joint formulation is what makes the encoder useful for both spatial retrieval and downstream temporal reasoning rather than excelling at one at the expense of the other (Sec. 4.6.2).

4.4.4 Geometry Dropout for Zero-Shot Transfer

Cross-camera generalization is a defining requirement for GeoLaneRep. A lane observed by a previously unseen camera may lack reliable geometric annotation, yet it must still be matched against a reference lane bank. To prevent the encoder from overfitting to static structural geometry, we apply geometry dropout during training.

With probability p_{drop}, the geometry-stream embedding f_i^g is zeroed out. This forces the network to derive lane semantics from the trajectory and descriptor streams alone. To preserve expected activation magnitudes, non-dropped embeddings are rescaled by $1/(1 - p_{\text{drop}})$, the standard inverted-dropout correction. At inference, geometry can be leveraged for established reference lanes that carry annotations while safely omitted for new query lanes. This technique enables trajectory-driven zero-shot retrieval without retraining.

4.5 Downstream Tasks

Following the training phase, the GeoLaneRep encoder facilitates three downstream tasks. These operate directly on the learned embeddings rather than on raw geometry or trajectories. This section details three primary applications: zero-shot cross-camera lane matching, temporal anomaly detection, and behavior-conditioned geometry generation. All three reuse the encoder weights from the contrastive/joint

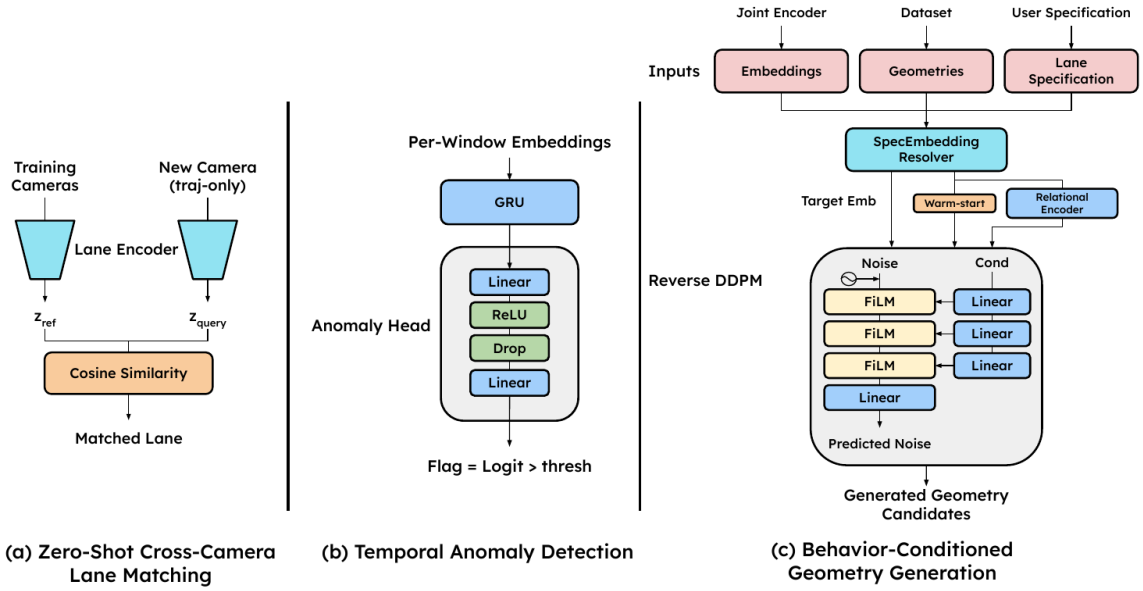


Figure 4.3: Downstream tasks built on the GeoLaneRep encoder. (a) Zero-shot cross-camera lane matching. (b) Temporal anomaly detection. (c) Behavior-conditioned geometry generation.

training described in Sec. 4.4.3. Only the generation task adds a separately trained head.

4.5.1 Zero-Shot Cross-Camera Lane Matching

Given a query lane from a previously unseen camera, the matching task retrieves the most similar lane from a reference bank constructed over the training cameras. Both query and reference lanes are encoded with the trained encoder, and the match is the reference whose embedding has the highest cosine similarity to the query:

$$\mathrm{match}(q) = \arg\max_{r \in \mathcal{R}} \frac{\bar{z}_q^\top \bar{z}_r}{\|\bar{z}_q\| \, \|\bar{z}_r\|}. \tag{4.19}$$

This setup tests whether the encoder captures viewpoint-invariant structural semantics rather than scene-specific shortcuts. In deployment, reference lanes are

typically encoded with all modalities available, while a query lane may be encoded under restricted conditions – most importantly, with the geometry stream dropped (Sec. 4.4.4) when the new camera has no attention. The matched reference's structural attributes (lateral rank, edge flags, group size) transfer to the query lane, providing the operational labels needed by the rest of the pipeline.

4.5.2 Temporal Anomaly Detection

To facilitate continuous temporal reasoning, the framework processes each lane as a sequence of per-window embeddings:

$$Z_i = [z_{i,1}, z_{i,2}, \ldots, z_{i,W_i}]. \tag{4.20}$$

and feeds the sequence through GRU to obtain a hidden state at each window:

$$h_i^{(w)} = \mathrm{GRU}\Big(z_{i,w}, h_i^{(w-1)}\Big). \tag{4.21}$$

A small MLP anomaly head maps each hidden state to a scalar logit $a_{i,w} \in \mathbb{R}$, supervised at training time by the validity-weighted BCE loss $\mathcal{L}_{\mathrm{temp}}$ defined in Eq. 4.17 against synthetically injected anomalies.

At inference, the logits are passed through a sigmoid to obtain per-window anomaly probabilities, and a threshold (chosen by Youden's J on the validation split) converts them to binary anomaly flags. Because the GRU consumes the full per-window sequence, the decision at window w has access to the lane's preceding behavioral context rather than only its current trajectory snapshot.

4.5.3 Behavior-Conditioned Geometry Generation

To connect the learned representation to digital twin interventions, the framework incorporates a behavior-conditioned geometry generator. Given a target embedding $z^\star \in \mathbb{R}^{128}$ that specifies a desired operational profile, supplied by the user or retrieved from a reference lane with the desired role, the generator synthesizes a

candidate centerline $\hat{g}$ whose behavioral semantics, when re-encoded, are close to $z^\star$.

The module is a denoising diffusion probabilistic model (DDPM) (Ho et al. (2020)) with a linear β schedule of length T_{diff}. Rather than initializing from pure Gaussian noise, the generative process applies a warm-start forward diffusion from a canonicalized anchor geometry g_{anchor} flattened into $w_s = \text{vec}(g_{\text{anchor}}) \in \mathbb{R}^{32}$ ($K = 16$ waypoints $\times$ 2 coordinates) by running the forward diffusion only up to an intermediate step t_0:

$$x_{t_0} = \sqrt{\bar{\alpha}_{t_0}}\, w_s + \sqrt{1 - \bar{\alpha}_{t_0}}\, \epsilon, \qquad \epsilon \sim \mathcal{N}(0, I), \tag{4.22}$$

where $\bar{\alpha}_t = \prod_{s=1}^{t}(1 - \beta_s)$.

The reverse denoising process is conditioned on the target embedding $z^\star$ via Feature-wise Linear Modulation (FiLM) (Perez et al. (2018)). At each denoiser layer l, two linear projections of the conditioning vector produce a scale and shift,

$$\gamma^l = f_{\text{scale}}(z^\star), \tag{4.23}$$

$$\beta^l = f_{\text{shift}}(z^\star), \tag{4.24}$$

which modulate the layer's hidden state after a linear transform and LayerNorm:

$$h_l = \text{GELU}\Big(\text{LayerNorm}\Big(f_{\text{hidden}}(h_{l-1})\Big) \cdot (1 + \gamma^l) + \beta^l\Big). \tag{4.25}$$

The denoiser also consumes a sinusoidal timestep embedding concatenated with the noisy geometry; from this it predicts the noise to remove at the current step.

Iterating the FiLM-conditioned denoiser from t_0 down to 0 yields a candidate geometry $\hat{g}$ that has been continuously nudged toward the target embedding's behavioral semantics. The module produces an ensemble of candidates per specification, which are scored and ranked by re-encoding $\hat{g}$ and measuring its similarity to $z^\star$ in the encoder space.

4.6 Experiments and Results

This section evaluates the GeoLaneRep framework across both encoder performance and downstream task execution. The evaluation encompasses lane-centered dataset construction, controlled preprocessing, and a series of targeted experiments. These experiments are explicitly designed to test representation quality, temporal consistency, and generation utility.

4.6.1 Experiment Setup

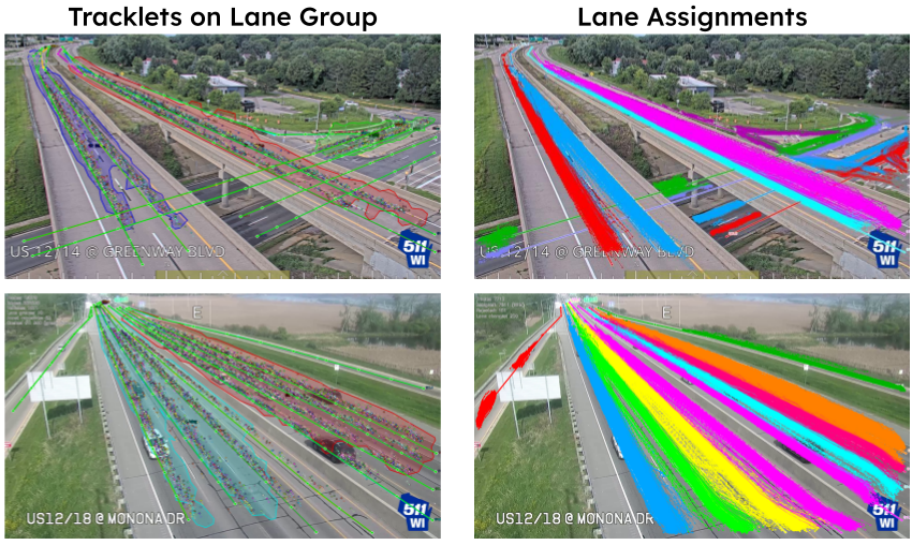


Figure 4.4: (Left) Tracklets extraction for each lane group depicted by transparent boundaries. (Right) Tracklets assignments to each annotated lanelet. Within a lane group, each color marks a distinct lane.

The dataset is engineered to expose variations in lane geometry, traffic behavior, and cross-site operating conditions. It aggregates feeds from 16 roadside cameras served by 511 Wisconsin, covering 132 lanes organized into 38 lane groups and a total of 104,415 vehicle trajectories. The structural composition reflects diverse operational functionalities: 38 leftmost (28.8%), 38 rightmost (28.8%), and 38

interior (28.8%) lanes, with the remaining 18 lanes (13.6%) functioning as merges. All 16 cameras contribute to both training and cross-camera evaluation.

To prepare the raw video feeds, the framework uses a deterministic preprocessing pipeline. Raw vehicle kinematics are first extracted using YOLOv11n (Khanam and Hussain (2024)) with a confidence threshold of ≥ 0.25, coupled with persistent tracking. Tracklets surviving a 5-frame minimum are spatially normalized to a $[0, 1]$ coordinate space. As illustrated in Fig. 4.4, these dynamic tracklets are then assigned to manually annotated static lane boundaries. Assignment computes the mean point-to-polyline distance between each tracklet and every candidate lane, binds the tracklet to the nearest lane within a 60-pixel threshold, and discards tracks that fail the threshold as out-of-bound tracking errors.

Crucially, this pipeline converts raw kinematic data into a structured lane-level representation. Rather than treating each vehicle in isolation, inputs are aggregated so that every sample describes holistic lane behavior, including temporal fluctuations and the lane's operational role (leftmost, rightmost, interior, or merge). Each lane sample is summarized by four aggregate trajectory statistics: mean speed, mean curvature, mean lateral offset from the lane centerline, and a normalized trajectory count used as a density proxy, computed over the tracklets assigned to that lane.

All static geometries and dynamic trajectories are arc-length resampled to $K = 16$ points via linear interpolation. To enable contrastive learning, positive pairs are strictly mined across different cameras. A candidate pair is accepted only when the two lanes satisfy three structural alignment criteria: lateral rank difference ($|\Delta| < 0.15$), identical edge-type flags (leftmost/rightmost), and role cosine similarity (≥ 0.8). The final collated batches supply the encoder with standardized tensors: static geometry $(B, K, 2)$, dynamic trajectories $(B, T_{\max}, K, 2)$ with a boolean validity mask $(B, T_{\max})$ that prevents zero-padded slots from biasing the mean-pooled trajectory embedding, and the 9-dim fused stats vector $(B, 9)$ concatenating four trajectory statistics with the five-dim structural role descriptor.

4.6.2 Leave-One-Camera-Out Cross-Camera Evaluation

Cross-camera generalization is evaluated using a leave-one-camera-out (LOCO) protocol. For any given camera c, all lanes from the remaining cameras $\mathcal{C} \setminus \{c\}$ are encoded to form a reference bank. Lanes from the held-out camera then serve as queries, each matched to its nearest reference by cosine similarity in the learned embedding space. This procedure is repeated with every camera held out in turn. This configuration tests whether the representation captures lane identity in a manner that transfers across unseen viewpoints and novel roadway layouts.

The evaluation reports four complementary metrics. *Mean cosine similarity* measures overall nearest-neighbor affinity. However, this metric alone can be misleading, because a representation that collapses every lane into a narrow region of feature space will also produce uniformly high similarities. Greater analytical weight is therefore placed on *lateral rank difference*, $|r_q - r_{\mathrm{ref}}|$, which verifies that the matched reference shares the query's relative lateral position rather than an arbitrary nearby embedding. This evaluation also tracks *edge F1*, which checks preservation of leftmost and rightmost boundary roles, alongside *anomaly accuracy*, measured on the same LOCO splits for methods that expose a temporal detection module.

Comparison Across Models

Table 4.1 benchmarks the proposed methodology against representative non-learning baselines, supervised references, and anomaly-detection alternatives. The compared models operate as follows:

- **traj-stats** uses a minimal set of aggregate trajectory statistics per lane and performs nearest-neighbor matching without learned embeddings.
- **stats-oracle** serves as the matching upper bound. It reads ground-truth lane-role annotations directly as input, so its high matching scores are not achievable at deployment time. It has no temporal module, hence no anomaly result.

Table 4.1: Comparison across cross-camera matching and anomaly detection baselines under the LOCO protocol. Lower lateral-rank difference is better; higher edge-role F1 and anomaly accuracy are better. "—" denotes metrics not applicable to a given method. The *Is Generalize* column summarizes whether the method is designed to transfer to unseen cameras at test time: *yes* for methods trained without per-site labels, *no* for supervised or oracle baselines, *partial* for `traj-stats`, which uses no learned representation but requires the target site's own trajectories at query time.

Method	Supervision	Match sim. ↑	Lat. diff ↓	Edge F1 ↑	Anomaly ↑	Is Generalize
traj-stats	none	0.991	0.398	0.356	0.875	partial
stats-oracle (GT)	role labels required	0.990	0.018	1.000	—	no
per-camera-sup	per-camera labels	—	0.227 / 0.476	0.481 / 0.182	—	no
SVM (One-Class)	none	—	—	—	0.776	yes
LSTM (traj-stats)	none	—	—	—	0.823	yes
GeoLaneRep (Ours)						
two-stage (frozen)	none	—	—	—	0.821	yes
geometry-only	none	**1.000**	0.448	0.262	—	yes
trajectory-only	none	0.996	0.387	0.358	—	yes
contrastive	none	0.932	0.013	0.994	—	yes
joint	none	0.962	**0.004**	**1.000**	**0.979**	yes

- **per-camera-sup** trains supervised lane-role predictors independently for each camera. Reporting both the within-camera validation and the held-out-camera test scores isolates whether camera-specific supervision transfers to unseen views.
- **contrastive** uses the static encoder trained purely for cross-camera alignment, with no temporal modeling.
- **two-stage (frozen)** establishes cross-camera alignment first, freezes the encoder weights, and subsequently trains a GRU-based anomaly detector on those fixed embeddings.
- **One-Class SVM** and **LSTM (traj-stats)** are anomaly-focused baselines. They indicate whether joint representation learning is required for temporal detection.
- **geometry-only** and **trajectory-only** are ablated architecture variants that isolate static shape cues and dynamic motion cues, respectively.

- **joint** constitutes the full proposed model. The encoder and temporal anomaly module are optimized simultaneously end-to-end.

The core finding is that high raw matching similarity does not imply meaningful cross-camera correspondence. The geometry-only, trajectory-only, and traj-stats baselines all yield extremely high nearest-neighbor similarity scores $(0.991–1.000)$, yet their lateral-rank errors remain large $(0.387–0.448)$, and their edge-role F1 scores remain low $(0.262–0.358)$. The three representations evidently place lanes into narrow clusters that admit high cosine similarity to the nearest neighbor without preserving the lane's functional role.

Applying the contrastive encoder sharpens cross-camera discriminability: lateral-rank error drops to 0.013 and edge F1 rises to 0.994, at the cost of a slightly lower raw similarity (0.932). The full joint model refines this further, reaching a lateral-rank error of 0.004 and an edge F1 of 1.000, while simultaneously securing the highest anomaly detection accuracy (0.979).

These results lead to two conclusions. First, cross-camera lane correspondence requires the combination of geometric structure, trajectory observation, and contrastive alignment. No single modality alone is sufficient. Second, joint temporal- and contrastive training strengthens the embedding for both spatial retrieval and downstream temporal reasoning, rather than sacrificing one for the other.

Training Behavior: Joint vs. Two-Stage Optimization

Analyzing the training strategies exposes a clear gap between end-to-end joint optimization and the two-stage approach. Figure 4.5 (Left) shows the frozen-encoder variant plateauing near 81% anomaly accuracy by epoch ~ 25 and making negligible further progress over the remaining epochs of that run. The joint model crosses the same 80% threshold at epoch ~ 25 and continues improving, stabilizing around 97% by epoch ~ 75. This sustained divergence indicates that anomaly detection capability cannot be merely "bolted on" to a fixed representation. The encoder itself must adapt to the temporal objective.

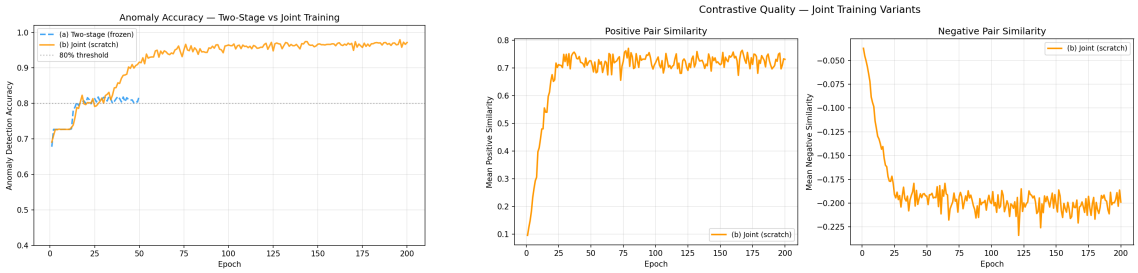


Figure 4.5: Training behavior of the proposed model. Left: anomaly detection accuracy for two-stage training with a frozen encoder versus end-to-end joint training. Joint optimization continues improving well beyond the plateau reached by the frozen-encoder variant. Right: contrastive quality during joint training, showing rapid convergence of positive-pair similarity and stable separation from negative pairs.

Figure 4.5 (Right) confirms that the contrastive structure remains intact during joint training. Mean positive-pair similarity rises from ~ 0.10 to ~ 0.72 within the initial 25 epochs. Simultaneously, mean negative-pair similarity drops to ~ -0.20 over the same interval. Both curves remain stable across the full 200-epoch run, with variance at the scale of mini-batch noise rather than systematic drift. End-to-end optimization, therefore, heightens temporal sensitivity without triggering representation collapse (positives and negatives remaining well-separated) or eroding cross-camera alignment.

4.6.3 Representation Quality

Lateral Rank Prediction

The encoder produces an embedding space in which cross-camera nearest-neighbor retrieval preserves lateral position: for each of the 132 lanes, the embedding-nearest lane drawn from a different camera shares the query's ground truth lateral rank with mean absolute error 0.010 (Fig. 4.6). Matches align with the diagonal across the full $0 \rightarrow 1$ rank spectrum. This confirms the encoder captures the relative lateral ordering of lanes independent of the raw camera viewpoint under which a lane was observed.

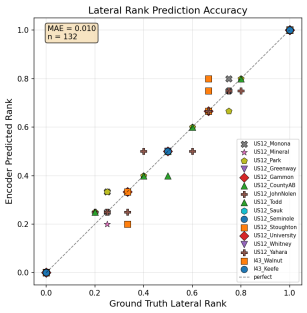


Figure 4.6: Cross-camera lateral rank alignment across 132 lanes from 16 cameras. Each point plots a query lane's ground-truth lateral rank against the ground-truth rank of its embedding-nearest neighbor retrieved from a different camera.

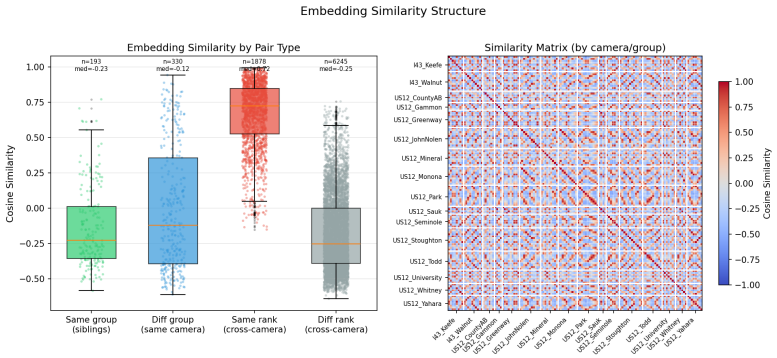


Figure 4.7: Embedding similarity structure. Left: cosine similarity of projection pairs grouped into four categories. Right: per-lane cosine similarity matrix, organized by camera and group.

Embedding Space Structure

Probing the learned embedding space reveals a distinct geometry driven by the contrastive training objective (Fig. 4.7). Lanes sharing identical lateral ranks across disparate cameras generate a median cosine similarity of $+0.72$ ($n = 1{,}878$ pairs). This reflects strong cross-scene generalization at the semantic level. Conversely, sibling lanes situated within the exact same physical group, which share spatial proximity but execute different roles, are pushed apart to a median similarity of -0.23 ($n = 193$), essentially matching the cross-camera diff-rank baseline. The embedding, therefore, does not use camera or group co-occurrence as a similarity shortcut; rank identity dominates. Different groups within the same camera sit at an intermediate -0.12 ($n = 330$), indicating only a small residual camera bias.

The per-camera similarity matrix (right panel) corroborates this structure; block-diagonal intensity captures within-camera within-group identity, while sparse off-diagonal bright cells mark the cross-camera same-rank matches that drive the $+0.72$ distribution.

4.6.4 Temporal Consistency and Anomaly Sensitivity

This evaluation isolates whether the learned embedding maintains temporal stability under normal operation while reacting dynamically to operational disruptions. The embedding is assessed as an anomaly signal across three dimensions: absolute detection accuracy, robustness to temporal aggregation scale, and structural localization.

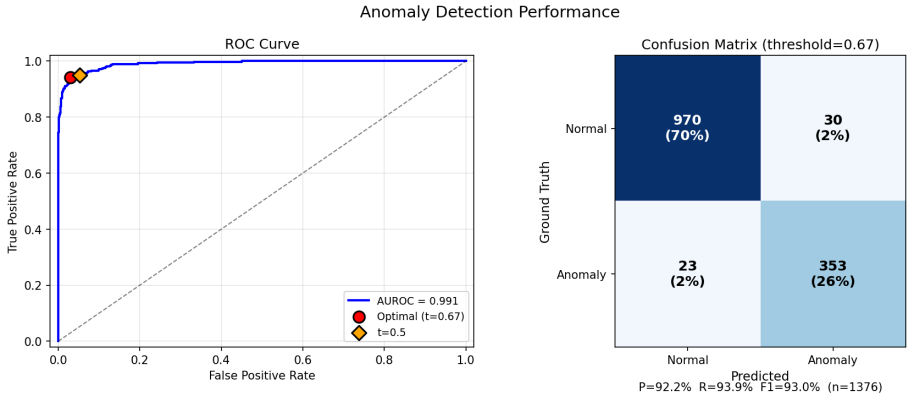


Figure 4.8: Overall anomaly detection performance on 5-minute detection window. Left: ROC curve with AUROC = 0.991 and operating points for the default and optimal thresholds. Right: confusion matrix at the optimal threshold ($t = 0.67$).

Overall Detection Performance

The embedding-based anomaly detector displays strong discriminative performance. Operating at a 5-minute window, the receiver operating characteristic yields an AUROC of 0.991 (Fig. 4.8). Utilizing an optimal threshold of $t = 0.67$, the detector achieves 92.2% precision, 93.9% recall, and a 93.0% F1 score across 1,376

evaluation windows. It triggers only 30 false positives and misses a mere 23 events. These margins show that embedding deviations provide a reliable mathematical signal for isolating anomalous behavior from standard traffic flow.

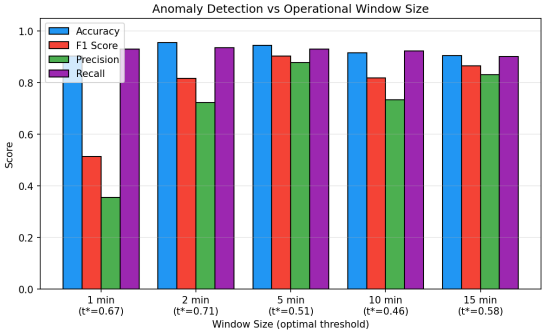


Figure 4.9: Anomaly detection performance across operational window sizes.

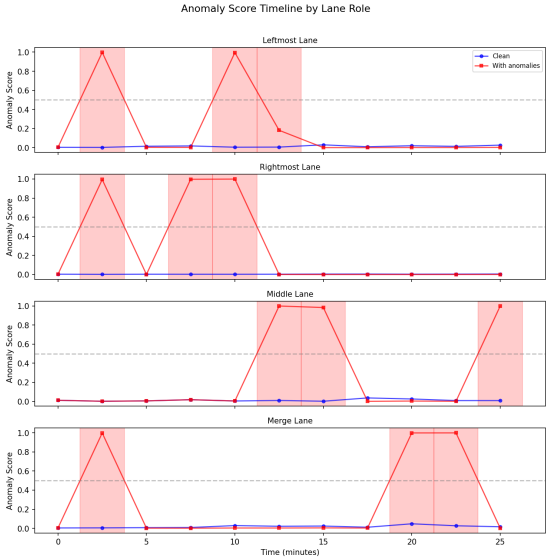


Figure 4.10: Anomaly score timelines across four representative lane roles. Each panel plots the detector's predicted anomaly probability on the clean window sequence (blue) and on the same sequence with injected corruptions (red). Red shaded bands mark windows where a synthetic anomaly was injected into the trajectory input.

Effect of Operational Window Size

Detection fidelity intrinsically depends on the volume of temporal context available. However, performance stabilizes rapidly (Fig. 4.9). At a severe 1-minute restriction, precision drops to 0.36, and the F1 score falls to 0.51, while recall remains high at 0.93. The detector still catches most injected anomalies, but the per-window trajectory statistics are averaged over too few samples to separate corrupted windows from

noisy clean ones, driving false positives up. This lack of history prevents the formation of a stable lane-level baseline. Expanding to a 2-minute window triggers a substantial recovery (F1 ≈ 0.82). The detector finds its optimal operational balance at the 5-minute mark (F1 ≈ 0.90, precision ≈ 0.88, recall ≈ 0.93). Performance remains consistent through 10- and 15-minute intervals. Consequently, a 2-to-5-minute contextual horizon provides maximum anomaly resolution while remaining highly practical for near-real-time digital twin monitoring.

Temporal Localization Across Lane Roles

Figure 4.10 maps anomaly score trajectories across standard leftmost, rightmost, middle, and complex merge lanes. Each panel overlays the predicted probability on the clean sequence (blue baseline) against the corrupted sequence (red), with shaded bands marking the injected windows. Under nominal conditions, these scores stay near zero. This proves the embedding holds temporal stability during standard operations. Upon anomaly injection, the scores spike sharply above the 0.5 threshold inside the injected windows, with magnitude varying by anomaly type. They return cleanly to baseline the moment the event concludes. Crucially, this reactive behavior is uniform across all structural lane roles. The anomaly signal is not biased toward a specific lane archetype.

4.6.5 Behavior-Conditioned Geometry Generation

The final experiment evaluates whether the learned representation can physically guide the generation of new lane geometries that adhere to targeted semantic specifications. The generator receives a target specification and a spatial anchor and outputs candidate centerlines, which are re-encoded and scored for semantic compliance.

Spec-Conditioned and Relational Generation

Figure 4.11 showcases generation performance at the `US12_Yahara` site. The module produces plausible centerlines for rightmost, leftmost, and merge roles when

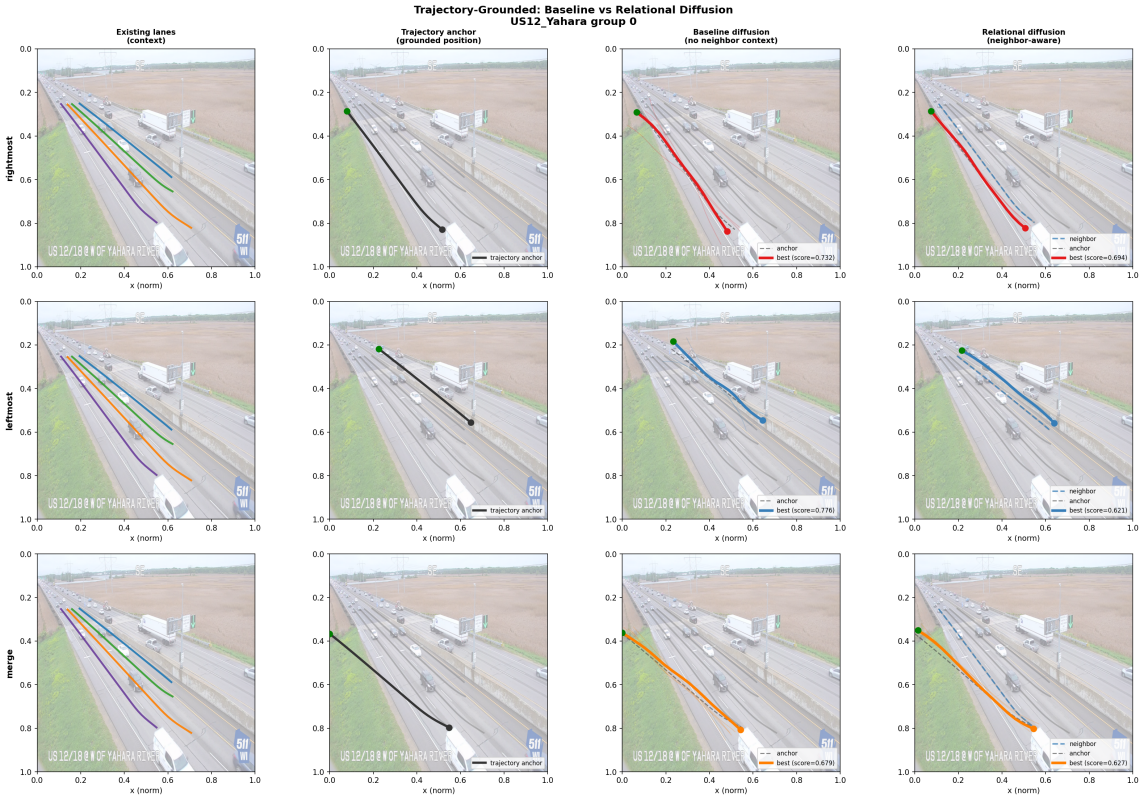


Figure 4.11: Qualitative examples of behavior-conditioned lane generation at `US12_Yahara` (group 0), comparing independent and relational conditioning for rightmost, leftmost, and merge lane generation.

conditioned on lateral role and anchor alone. For standard (edge) lanes, both independent and relational variants yield geometries that visually align with the real lane structure, with rightmost and leftmost candidates exhibiting the lowest semantic reconstruction errors. This confirms the generated lines remain faithful to the target embedding.

Relational conditioning explicitly exposes the generator to the neighboring-lane context. This constrains the output toward locally consistent spacing. The generated geometries track the surrounding lane bundle rather than drifting independently, as is visible in the no-neighbor variant.

Generation Quality Across Sites

Table 4.2: Quantitative evaluation of behavior-conditioned lane generation (38 lane groups, 570 candidates, 5 candidates per spec per group). FGD and curvature smoothness are reported in both raw and filtered forms; the filtered forms exclude candidates flagged by the curvature-outlier criterion.

Category	Metric	Value
Specification Accuracy	Rightmost Lane	98.4%
	Leftmost Lane	99.5%
	Merge Lane	65.8%
	Overall Accuracy	**87.9%**
Per-spec Chamfer	Rightmost	0.039
	Leftmost	0.040
	Merge	0.052
Geometric & Spatial	Candidate Diversity (Mean L2)	0.1737
	Spatial Coherence	0.1003
	Chamfer Distance (overall)	0.0450
	Fréchet Geometry Distance (raw / filt.)	0.0179/0.0254
	Curvature Smoothness (raw / filt.)	2.727/0.0220
	Smoothness Outliers	7.0% (38/570)

Quantitative evaluation across 38 camera groups and 570 generated candidates (Table 4.2) supports these observations. The generator attains 99.5% lateral specification accuracy for leftmost lanes and 98.4% for rightmost lanes, with an overall accuracy of 87.9%. Merge lanes account for most of the residual error (65.8% accuracy), examining where merge-specified candidates actually land shows that failures do not scatter randomly but bifurcate into leftmost or rightmost geometries, suggesting that the model has learned "merge" as a region of interpolation between the two edge-lane modes rather than as a distinct topological class. Per-spec Chamfer distances tell the same story: merge candidates sit at 0.052, roughly 30% higher than the 0.039–0.040 seen on rightmost/leftmost, so the gap is not only categorical but also geometric.

Aggregate geometric fidelity is strong. A mean Chamfer distance of 0.0450 indicates close pointwise alignment with the real roadway geometry. Fréchet Geometry Distance (FGD) is 0.0179 on the full candidate set and 0.0254 after excluding candidates flagged as curvature outliers; in both regimes, the distributional shape of the generated lanes closely matches the reference. The generator avoids representation collapse: mean pairwise L_2 distance is 0.1737, so the system produces a diverse ensemble of candidates rather than repeating a single solution. Finally, the curvature-smoothness analysis flags 7.0% (38 / 570) of candidates as rough outliers – high variance in heading-change angles along the polyline – leaving 93% whose curvature profiles fall within the smooth range defined by the threshold.

4.7 Conclusions

This research introduces **GeoLaneRep**, a behavior-grounded representation framework that maps static lane geometry, observed vehicle trajectories, and operational descriptors into a single shared embedding. Trained jointly with contrastive cross-camera alignment, auxiliary role supervision, and temporal anomaly detection, the encoder achieves a 0.004 lateral-rank error and an edge-role F1 of 1.000 under a leave-one-camera-out protocol, and an AUROC of 0.991 on per-window anomaly detection. Consequently, it produces a highly transferable embedding that generalizes seamlessly across heterogeneous camera deployments. Furthermore, a FiLM-conditioned diffusion module uses these embeddings to synthesize candidate lane geometries that satisfy targeted operational specifications at 87.9% overall accuracy. The same encoder weights drive all three tasks, supporting the central claim that a single behavior-aware representation can serve heterogeneous lane-level digital-twin applications.

The evaluation also surfaces an asymmetry between representation and generation quality. The encoder produces a stable, highly discriminative embedding space, but the generated lane geometry trails this fidelity, particularly for merge specifications (65.8% accuracy versus 98–99% for edge lanes). This gap likely reflects compounded upstream uncertainty rather than a generator-side defect alone:

imperfect trajectory extraction, occlusion, lane-marking ambiguity, and limited coverage of rare lane types each weaken the geometric supervision the diffusion module receives. A second limitation is that the generation scheme operates lane-wise. It produces one plausible lane at a time, conditioned on a target embedding, and does not yet model the mutual dependence among neighboring lanes that real roadway geometry imposes. Third, the framework prioritizes semantic consistency over engineering-grade geometric fidelity — a deliberate trade that allows training on noisy roadside observations without curated HD-map labels, but one that limits direct deployability for tasks that need vertical alignment, curvature continuity, or precise lane-boundary reconstruction. Finally, anomaly detection performance degrades sharply at the shortest evaluated window (precision drops to 0.36 at 60-second windows), reflecting variance in per-window trajectory statistics when only a handful of tracklets are available.

Each of these limitations points to a concrete extension. Closing the representation–generation gap calls for richer geometric supervision: multi-view reconstruction, depth estimation, LiDAR fusion, or BIM-derived lane references would give the diffusion module a cleaner target without disturbing the behavioral embedding. Moving from lane-wise to corridor-level synthesis would require coupled multi-lane generation under topological and geometric constraints. For example, graph-structured decoders over lane groups, or relational denoisers conditioned on the neighboring-lane embeddings already produced by the cross-lane attention module. A third direction is interfacing the embedding with connected-vehicle infrastructure: V2X MAP messages provide a machine-readable lane topology that could either supervise the representation or, conversely, be inferred or repaired by it when annotations are incomplete. Together, these extensions would carry GeoLaneRep beyond lane-level interpretation toward the coordinated, geometrically faithful interventions that constructive digital twins ultimately require.

Therefore, GeoLaneRep establishes the semantic interface required for next-generation Advanced Traffic Management Systems (ATMS). It directly bridges the physical perception capabilities developed in Chapter 2 with the codified reasoning schemas established in Chapter 3. This chapter completes the central architectural

layer of the dissertation. It provides the mechanism for an interactive digital twin ecosystem to actively observe, interpret, and reshape traffic infrastructure in a human-centered and operationally grounded manner. The final chapter synthesizes these overarching contributions and outlines the roadmap for automated, human-in-the-loop traffic management.

5 USE CASE IMPLICATIONS OF CONSTRUCTIVE AND INTERACTIVE DIGITAL TWINS

Final chapter synthesizes the dissertation contributions and discusses their implications for automated, human-in-the-loop traffic management. The preceding chapters established the three core layers required for an interoperable digital twin. Chapter 2 provided a perception layer by reconstructing lane-level infrastructure geometry from observed traffic trajectories. Chapter 3 provided a knowledge layer by exposing transportation standards through an executable and validated semantic framework. Chapter 4 introduced a lane representation that bridges geometry, traffic behavior, and operational implications.

This chapter brings these components together through an explicit application context as **GeoLane-Twin**. The goal is not to introduce a completely separate model, but to demonstrate how the dissertation contributions can be integrated into a constructive and interactive digital twin workflow for traffic management. The chapter proceeds in three sections. First, it introduces GeoLane-Twin as a lane-centric use case that synthesizes the constructive and interactive perspectives developed throughout the dissertation. Second, it evaluates the integrated framework through a closed-loop digital twin experiment based on generation, simulation, and re-encoding. Third, it presents a Rust-based prototype system that exposes this workflow through a user-facing interface.

5.1 GeoLane-Twin Architecture

A representative use case of the proposed framework is the **GeoLane-Twin**, a lane-centric digital twin in which each lane is represented not only as a geometric object, but as an operational unit associated with interpretable semantic attributes. As illustrated in Fig. 5.2(a), the GeoLane-Twin has basic traffic monitoring features on the corridor levels. In Fig. 5.2(b), the roadside observation is made on one camera, where vehicle detections and trajectories are extracted from live traffic

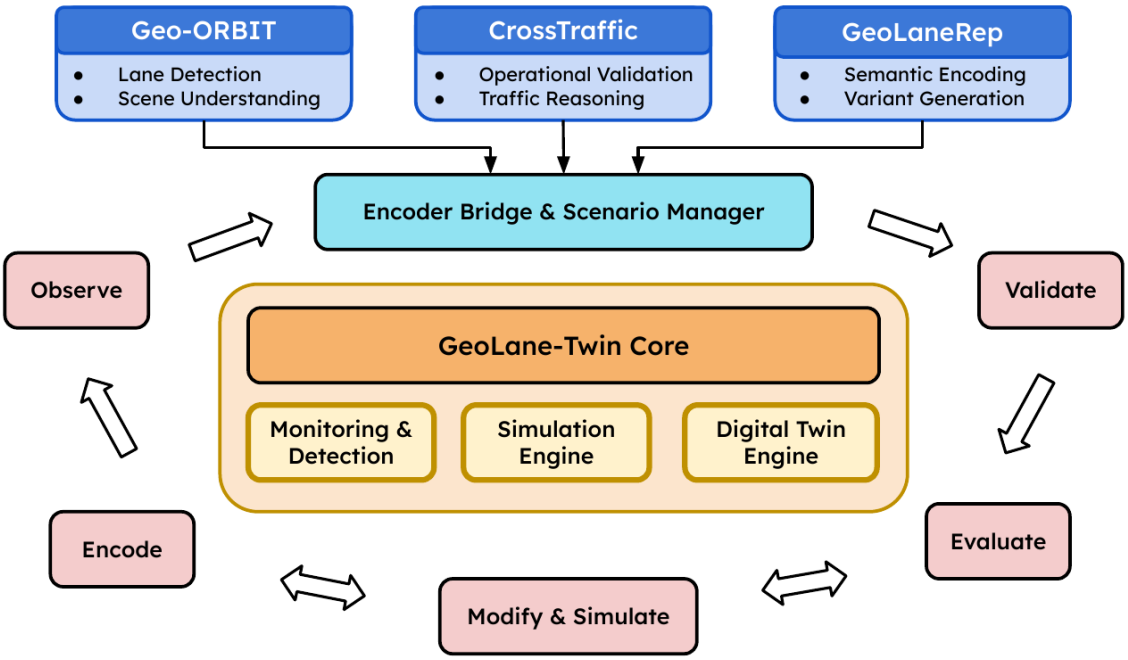
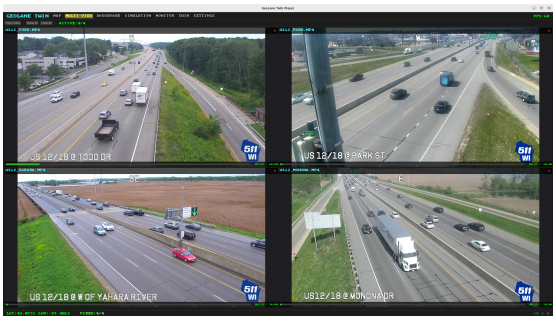


Figure 5.1: Conceptual overview of GeoLane-Twin as a lane-centric digital twin for traffic management.

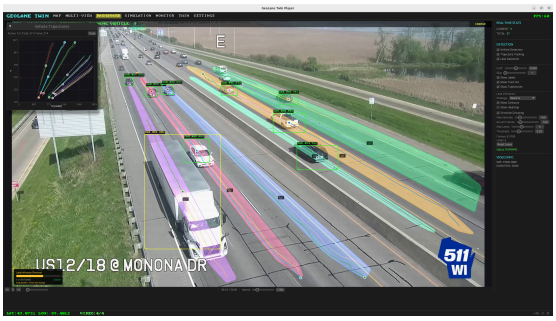
scenes. These observations are then synchronized into a virtual environment, as shown in Fig. 5.2(c), so that the twin can maintain an updated representation of roadway structure and ongoing traffic conditions.

Figure 5.2(d) illustrates the role of the lane embedding within this process. The embedding serves as an intermediate semantic state between observation and intervention. By combining lane geometry, motion patterns, and interpretable descriptors into a unified representation, this approach provides a basis for comparison, retrieval, anomaly detection, and generation, while remaining grounded in observed traffic operations. GeoLane-Twin extends passive synchronization of digital twin toward constructive reasoning in which lane-level roles can be interpreted, evaluated, and systematically modified within a closed-loop traffic management framework.

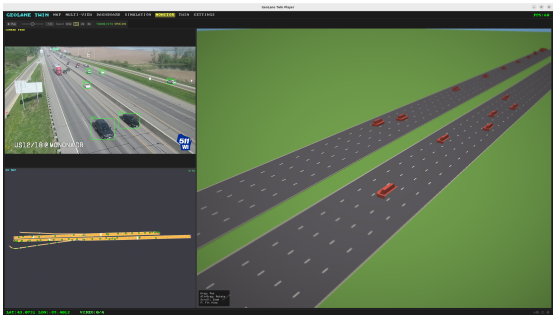
This use case also clarifies the dissertation-level distinction between *constructive* and *interactive* digital twins. The constructive dimension concerns the system's



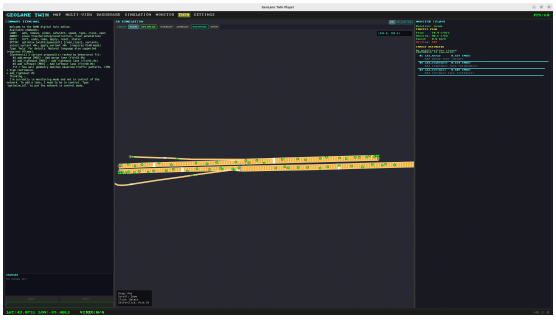
(a) Traffic monitoring on corridor



(b) Trajectory monitoring with object detection, tracking, and lane detection



(c) Descriptive digital-twin synchronization



(d) Constructive and interactive digital twin interface

Figure 5.2: Constructive and interactive digital twins capabilities on top of the basic traffic management system features.

ability to build, update, and modify lane-level infrastructure representations from sensed traffic data. The interactive aspect lies in the presentation of these representations in a form that supports human interpretation, semantic validation, and user-facing intervention workflows. GeoLane-Twin may therefore be understood not simply as a demonstration environment, but as a synthesis of the constructive and interactive principles developed throughout this dissertation.

5.2 Digital Twin Integration and Closed-Loop Evaluation

The final component of the framework connects the learned lane representation to digital twin-based evaluation. Once a candidate lane geometry $\hat{g}$ is generated, it is inserted into a microscopic simulation environment, where traffic behavior is simulated under the modified configuration. The resulting lane-linked traffic observations are then reprocessed and re-encoded into the same latent space.

This closed-loop procedure enables semantic consistency checking. If the generated lane faithfully reflects the intended operational role, the re-encoded embedding $\bar{z}_i'$ should remain highly similar to the target embedding z^*. If not, the discrepancy suggests that the generated geometry failed to preserve the intended behavioral meaning under simulated traffic demand. The re-encoding step, therefore, provides an evaluation criterion based not only on geometric similarity, but also on whether the lane behaves as intended after simulation.

5.2.1 Objective and Experimental Setup

The objective of the experiment is to examine whether behavior-conditioned lane generation preserves semantic lane-role identity through a closed-loop digital twin pipeline. The study uses eight camera locations along the US-12 and I-43 corridors in Wisconsin, covering 132 annotated lanes across four role classes: *rightmost*, *leftmost*, *merge*, and *through*. The pipeline combines a pre-trained lane encoder from Chapter 4, the corresponding diffusion-based lane generator, CrossTraffic with an HCM-based semantic firewall, and SUMO microscopic simulation.

Three intervention types are considered:

1. **Lane Shift**: shoulder conversion with a rightmost-role target,
2. **Left Lane**: HOV or passing-lane insertion with a leftmost-role target,
3. **Merge Adjust**: acceleration-lane modification with a merge-role target,

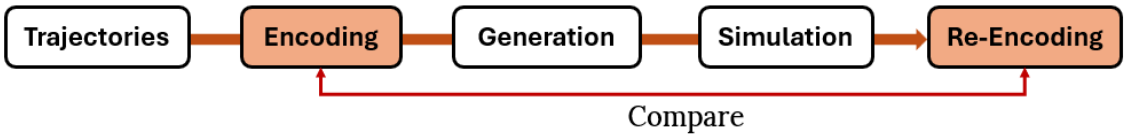


Figure 5.3: Process of re-encoding experiment.

Applying these three interventions across the eight camera locations yields 32 total test cases. For each case, the following protocol is applied as shown in Fig. 5.3. First, the observed real-world lane is encoded into a target semantic embedding. Second, the intervention parameters are validated against engineering constraints using CrossTraffic. Third, lane geometry is generated conditioned on the target embedding. Fourth, the generated geometry is inserted into SUMO, and traffic is simulated. Finally, the simulated lane observations are re-encoded and compared against embeddings of real lanes with the corresponding role. Encode the observed real-world lane into a target semantic embedding,

The primary evaluation metrics are as follows:

- **In-class similarity**: cosine similarity between the re-encoded intervention and real lanes with the same role,
- **Cross-role similarity**: cosine similarity to lanes with different roles,
- **Role gap**: in-class minus cross-role similarity,
- **Generation quality**: geometric validity and coherence of the generated lane.

Table 5.1 summarizes the overall evaluation setup and the role of CrossTraffic in constraining candidate interventions before simulation. All 44 proposed interventions pass CrossTraffic validation. The final semantic evaluation is reported on the subset of cases that complete the full simulation and re-encoding loop.[1]

Table 5.1: Closed-loop evaluation setup for GeoLane-Twin.

Item	Value	Description
Camera locations	8	US-12 and I-43 corridors
Annotated lanes	132	Four lane-role categories total
Lane roles	4	Rightmost, leftmost, merge, through
Intervention types	4	Lane shift, merge adjust, work zone, left lane
Total test cases	44	11×4 interventions
Lane encoder	–	Pre-trained GeoLaneRep encoder (Chapter 4)
Generator	–	Diffusion-based lane generator (Chapter 4)
Validator	–	CrossTraffic semantic firewall (HCM 7th)
Simulator	–	SUMO microsimulation
CrossTraffic validated	44/44	All cases passed engineering constraints
With SUMO network	8	All cameras had a SUMO network
Successfully simulated	32	Final successful closed-loop cases
Excluded	12	Three cameras with no SUMO edge match

5.2.2 Role-Aware Semantic Preservation

The main result is shown in Fig. 5.4. Across the 32 successful simulations, the closed-loop pipeline achieves an overall in-class similarity of 0.899 ± 0.054, a cross-role similarity of 0.057 ± 0.162, an average role gap of 0.758, and a mean generation quality of 0.905. Taken together, these values indicate that the generated and re-encoded lanes remain substantially closer to their intended role than to alternative role classes after the full re-encoding loop.

Three intervention types show especially strong semantic preservation. Lane shift achieves an in-class similarity of 0.915 ± 0.021 and a role gap of 0.933. Left-lane insertion achieves 0.905 ± 0.032 and a role gap of 0.893. Work-zone intervention achieves 0.882 ± 0.043 and a role gap of 0.923. In all three cases, cross-role similarity remains near zero, and for lane shift and work zone, it becomes slightly negative. Because cosine similarity ranges from -1 to $+1$, these negative values indicate that the embeddings are not merely weakly related, but are separated slightly beyond orthogonality in the latent space. It indicates that the encoder distinguishes positional lane roles strongly enough that rightmost-lane interventions become weakly

[1]The exact filtering details from 44 validated cases to 32 successful semantic evaluation cases can be inserted here once the final accounting is fixed.

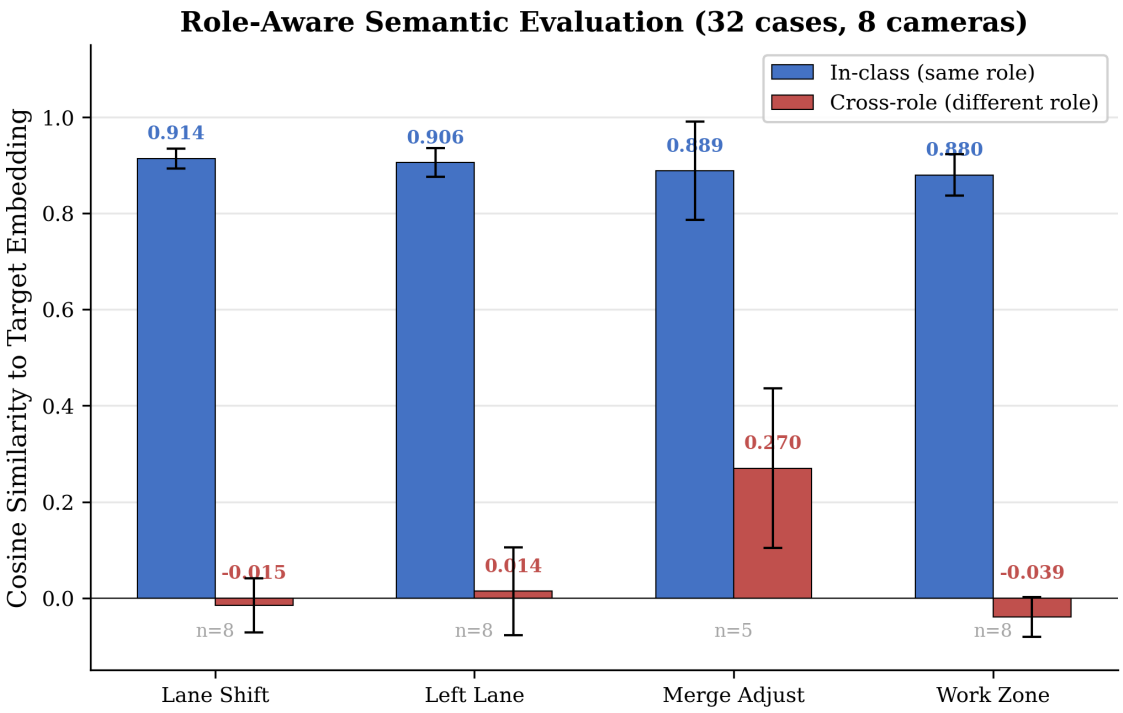


Figure 5.4: Role-aware semantic evaluation of closed-loop interventions. In-class similarity measures alignment with real lanes of the intended role, while cross-role similarity measures overlap with non-target roles.

anti-correlated with non-target roles. This suggests that GeoLane-Twin generates lane interventions whose simulated behavior remains semantically consistent with the intended operational role after re-encoding.

5.2.3 Failure Analysis of Merge Lane

Merge-adjustment interventions are the main outlier. Although their in-class similarity remains relatively high at 0.888 ± 0.103, their cross-role similarity increases to 0.272 ± 0.168, reducing the role gap to 0.283. This is substantially weaker than the corresponding values observed for lane shift, left lane, and work zone interventions.

Figure 5.5 clarifies this pattern. The per-camera analysis shows that semantic preservation for merge lanes varies considerably across sites. The main cause

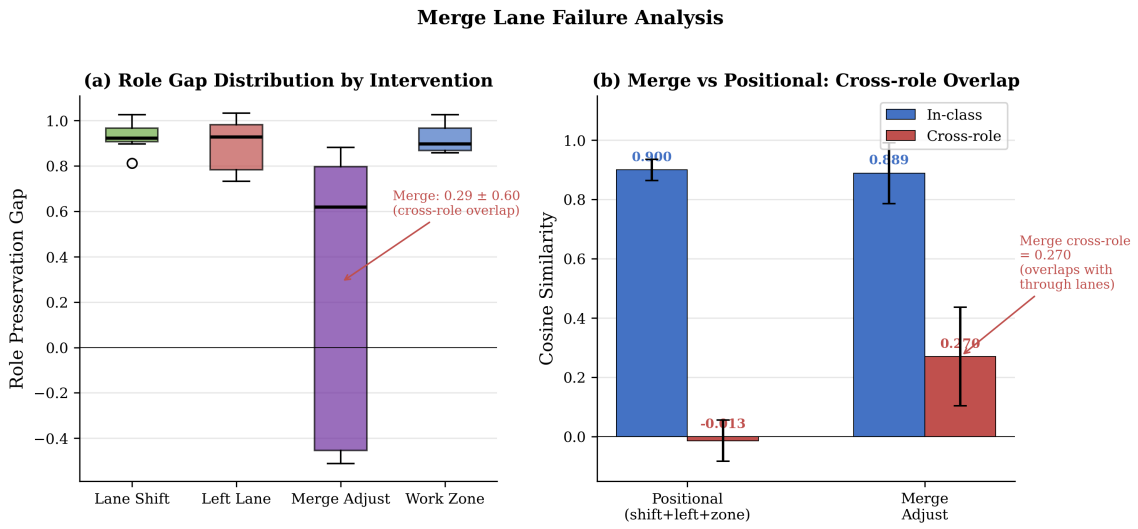


Figure 5.5: Failure analysis for merge-lane interventions. Merge cases show substantially greater cross-role overlap and higher variance than positional lane interventions.

(Right) is shown that positional interventions have cross-role similarity near zero, whereas merge interventions exhibit substantial positive overlap with non-merge lanes, especially through lanes.

This behavior is plausible from a traffic perspective. Merge lanes share many local trajectory characteristics with adjacent through lanes over most of their length, especially speed and heading, and become clearly distinct only near the convergence region. Their identity is therefore more topological and context-dependent than purely positional. This makes them more difficult both to generate and to recover through re-encoding. The lower generation-quality score for merge adjustment (0.809) reinforces this interpretation. Accordingly, the primary limitation of the current GeoLane-Twin loop is reduced robustness for topologically ambiguous lane roles.

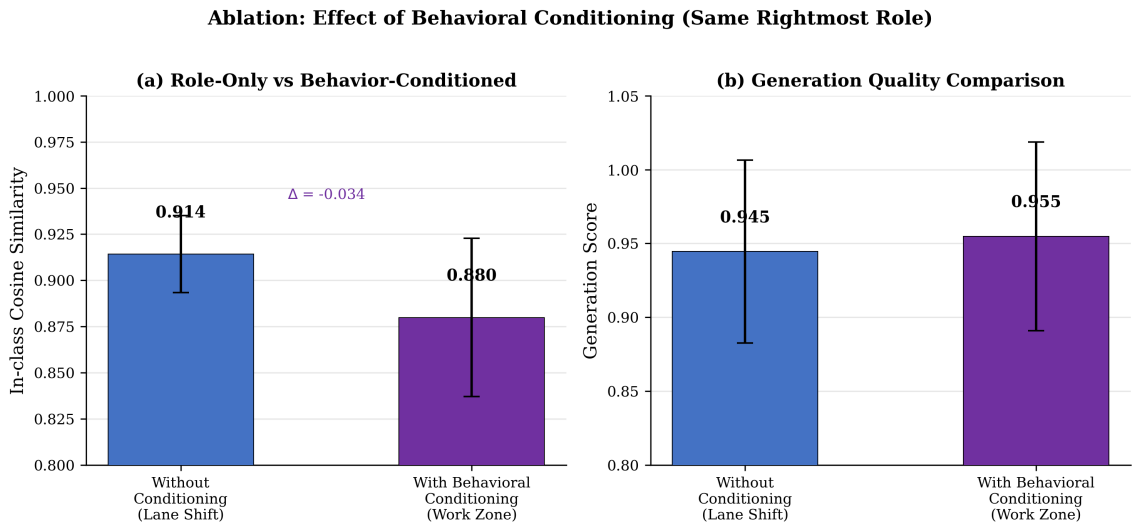


Figure 5.6: Ablation on behavioral conditioning. Comparing standard rightmost-lane conditioning with work-zone conditioning shows that behavioral prefixes shift the target embedding within the role cluster.

5.2.4 Effect of Behavioral Conditioning

To examine whether behavioral conditioning meaningfully influences generation, the analysis compares a standard rightmost-lane intervention (lane shift) against a behavior-conditioned rightmost intervention (work zone). The work-zone case achieves a lower in-class similarity (0.880 versus 0.914), but slightly higher generation quality (0.955 versus 0.945).

At first glance, the reduction in in-class similarity may appear counterintuitive. However, this behavior is consistent with the role of the behavioral prefix. The work-zone embedding is not intended to match a generic rightmost lane. Instead, it shifts the target toward a more specific region within the rightmost-role cluster, corresponding to slower and more behaviorally constrained traffic conditions. Since such specialized conditions are less common than standard rightmost-lane examples, average similarity to the larger generic cluster becomes slightly lower.

At the same time, the improved generation quality suggests that the diffusion model benefits from the more specific conditioning signal. The result is therefore

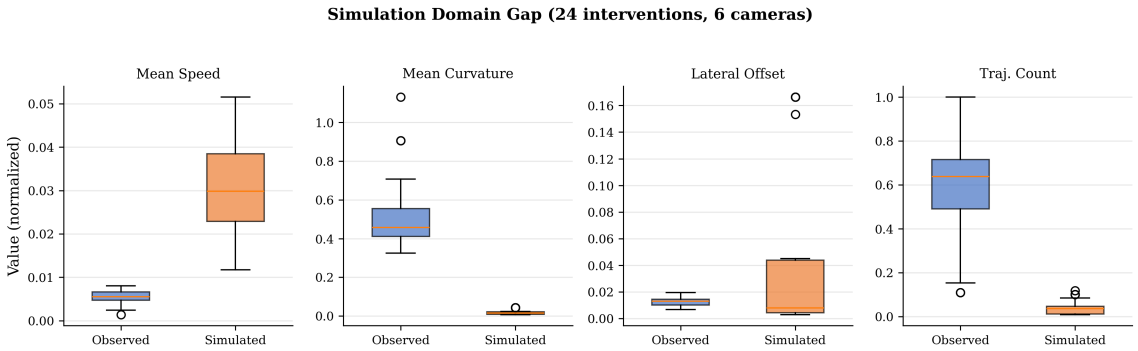


Figure 5.7: Behavioral gap between observed and simulated trajectories. Differences in speed, curvature, lateral offset, and trajectory count indicate that the current simulation loop remains behaviorally simplified.

best interpreted as a semantic precision and coverage. Behavioral conditioning makes the target more selective, which slightly lowers broad in-class similarity while improving the specificity of the generated lane.

5.2.5 Simulation Domain Gap and Current Limitation

The simulation-domain-gap analysis in Fig. 5.7 should be treated as an explicit limitation of the current system. The observed and simulated trajectory distributions differ substantially in mean speed, curvature, lateral offset, and trajectory count. In other words, the SUMO demand used in the current loop does not yet fully reproduce the role-specific traffic behavior observed in the field.

This limitation is important for interpretation. The results presented above demonstrate that semantic lane-role identity can be preserved through a closed-loop process of validation, generation, simulation, and re-encoding. They do not yet demonstrate behaviorally calibrated traffic prediction under intervention. In this sense, the present chapter provides evidence for the semantic consistency of the digital-twin loop, rather than full operational realism of the traffic simulation.

5.3 Prototype System: User-Facing GeoLane-Twin Interface

Beyond the offline evaluation, I implemented a Rust-based prototype application that exposes the GeoLane-Twin workflow through an operator-facing interface. This prototype serves as a system instantiation of the dissertation architecture, showing how the closed-loop pipeline can be surfaced as an interactive traffic-management tool.

5.3.1 Current Scope

The current prototype supports four core functions. First, it loads and visualizes roadway network structure derived from the digital-twin pipeline. Second, it allows lane-level interventions to be staged through an interactive command interface. Third, it previews the modified network in either a 2D or 3D simulation view. Fourth, it displays basic traffic performance summaries, such as vehicle count, speed, flow, density, waiting vehicles, and selected edge-level statistics.

Together, these functions provide a concrete interface for inspecting detected infrastructure, proposing modifications, previewing their effect on the virtual network, and monitoring the resulting traffic state.

5.3.2 Interface Design

The prototype is organized into three main panels:

1. **Command terminal**: a left-side interaction panel where the user enters lane-editing commands, reviews system feedback, and stages pending modifications;

2. **Simulation view**: a central panel that displays the roadway network and vehicles in either 2D or 3D, enabling before-and-after inspection of the twin;

3. **Performance panel**: a right-side panel that summarizes traffic conditions using aggregate metrics such as flow, density, speed, waiting vehicles, and vehicle mix.

This layout reflects the intended role of the digital twin as both a constructive and interactive system. The operator can modify the infrastructure representation while simultaneously observing the evolving traffic state.

5.3.3 Supported Interventions

At its current stage, the interface supports several lane-level editing operations, including lane addition, lane removal, lane widening, speed modification, lane closure, lane reopening, and work-zone style annotations. These edits are staged before application, allowing the user to inspect pending modifications and compare the modified network against the original layout. The current implementation therefore supports an intervention workflow based on propose, preview, and apply, rather than immediate irreversible editing. The prototype also includes an early natural-language interaction pathway that maps free-form input into structured lane-editing commands.

5.3.4 Relationship to the Closed-Loop Framework

The prototype provides a system-level realization of the GeoLane-Twin loop developed in this dissertation. In the broader architecture, lane structure is first observed from real-world data and encoded into a machine-readable semantic representation. Candidate interventions are then checked against engineering constraints through CrossTraffic, translated into modified lane geometry, and reflected within the virtual network. The prototype exposes this logic through an interface that allows users to inspect the current twin, stage lane modifications, visualize the updated road structure, and monitor simulation outputs.

5.3.5 Use Case

One representative use case is a lane-closure or work-zone scenario. A user first loads a corridor segment into the interface and inspects the detected lane structure. The user then stages a rightmost-lane closure or work-zone modification through the command terminal. The updated geometry is previewed in the simulation panel, and the performance panel displays how traffic indicators change after the intervention is applied. This use case mirrors the logic of the closed-loop experiments discussed earlier; lane semantics are not only encoded and analyzed, but also used to guide infrastructure edits within an interactive digital-twin environment.

5.3.6 Current Limitations

The current prototype remains under active development and has several limitations. First, it should be regarded as a research prototype rather than a production-ready traffic management system. Second, the operational usefulness of the interface has not yet been evaluated through a user study with practicing traffic engineers or operators. Third, simulation realism remains constrained by the same behavioral domain gap discussed earlier in this chapter, since the simulated demand does not yet fully reproduce observed role-specific traffic patterns. Finally, parts of the broader data-processing pipeline still depend on manual calibration and data preparation.

Despite these limitations, the prototype demonstrates that the dissertation contributions can be assembled into a unified, operator-facing digital-twin interface. It therefore provides a concrete bridge from representation learning and standards-aware validation toward next-generation advanced traffic management applications.

5.4 Discussion and Implications

The results of this chapter suggest that constructive and interactive digital twins can be connected through a lane-centric semantic representation. On the constructive side, the system can detect, encode, generate, and modify lane-level infrastructure

under engineering constraints. On the interactive side, the same semantic representation can be exposed through a user-facing interface that supports inspection, intervention staging, and monitoring.

The closed-loop experiment provides evidence that this integration is semantically coherent. Generated interventions remain substantially closer to their intended lane role than to competing roles after simulation and re-encoding, especially for more clearly defined positional roles such as rightmost, leftmost, and work-zone interventions. At the same time, the failure analysis identifies an important limitation: roles with stronger topological ambiguity, such as merge lanes, remain more difficult to preserve robustly.

Taken together, these findings suggest that semantic lane representations can serve as a practical intermediate layer between roadway sensing and traffic-engineering action. In addition to treating lanes as geometric primitives or as rule-defined objects, GeoLane-Twin shows how they can be represented as operational units that support comparison, anomaly detection, generation, and guided intervention.

5.5 Future Directions

Several directions follow from this chapter.

First, the simulation component should be calibrated more carefully against observed traffic behavior. The current behavioral gap indicates that semantic preservation and operational realism should be treated as related but distinct objectives. Future work should incorporate role-aware demand calibration so that post-intervention traffic patterns more closely reflect field observations.

Second, merge-lane handling should be strengthened by incorporating richer topological context. Because merge semantics depend heavily on downstream connectivity and local convergence structure, future models should include more explicit graph-level or connectivity-aware conditioning.

Third, the operator-facing prototype should be extended and evaluated through human-subject studies or feedback from practitioners. Such studies would help determine whether semantically guided lane suggestions and standards-aware

validation improve traffic-engineering reasoning and intervention workflows in practice.

Finally, GeoLane-Twin could be extended toward broader corridor-level or network-level traffic management applications. The present dissertation focuses on lane-level reasoning, but the same semantic interface may support more scalable digital twins that reason across interacting links, intersections, and control strategies.

5.6 Chapter Summary

This chapter demonstrated how the dissertation contributions can be synthesized into a constructive and interactive digital twin use case. GeoLane-Twin was introduced as a lane-centric digital twin in which each lane is represented not only by geometry, but also by behavioral and semantic meaning. A closed-loop experiment showed that generated lane interventions generally preserve their intended role after validation, simulation, and re-encoding, with the strongest performance observed for positional roles and clear limitations in merge-lane scenarios. The chapter also presented a Rust-based prototype interface that exposes this workflow through lane visualization, intervention staging, modified-network preview, and traffic monitoring. Taken together, these results show that the dissertation also contributes a coherent path toward interoperable digital twins for next-generation advanced traffic management systems.

6 CONCLUSION AND FUTURE WORK

This dissertation centers on advancing the integration of roadway geometry comprehension into digital twin environments and their semantic representations for Advanced Traffic Management Systems (ATMS). Toward the first objective, I proposed a scalable digital twin synchronization framework, **Geo-ORBIT**, designed to enable scalable and privacy-preserving roadway adaptation across a distributed roadside deployment framework. The proposed approach implements lane-level real-time roadway geometry perception and digital twin modeling. Specifically, the primary advantages are that it provides (1) a capability of learning roadway geometric features from heterogeneous camera perspectives and (2) strategies in preserving privacy by adopting federated learning to TMS and applying the model on unseen locations. I found that our model consistently reconstructs fine-grained lane-level alignment. With learned parameters from weak-supervised training, our model infers lane detection pipelines operating effectively in both seen and unseen environments. It is also noteworthy that the whole detection pipeline preserves data confidentiality by limiting shared data and significantly reducing data transmission costs.

Building upon the scalable digital twin framework for TMS, I further investigated the unification of TMS capabilities from the perspective of knowledge sharing and transportation knowledge management, with a particular focus on codifying the transportation manuals, such as the Highway Capacity Manual and American Association of State Highway and Transportation Operations (AASHTO) Green Book. To address this, I developed a framework **CrossTraffic**, an open, modular platform designed to enhance computational procedures for consistent and transferable knowledge management. Central to this platform is the development of a domain-specific ontology that structures explicit engineering rules into a comprehensive knowledge graph. The formal representation supports reproducibility, interoperability, and collaborative advancement in transportation analysis. Extending the CrossTraffic framework, I introduced a semantic decision-support layer,

powered by Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG). This layer enables natural language interaction with transportation manuals, facilitating intelligent retrieval of methodologies and dynamic orchestration of analytical workflows. In addition, to apply codified knowledge to constrain the digital twin's behavior, an ontology-based semantic validator is used to interpret, check, and connect applications to explicit engineering knowledge. Collectively, these efforts contribute to the foundation of a responsive and consistent knowledge management ecosystem, with early prototypes demonstrating the feasibility of co-simulation and natural language-based interaction.

Leveraging these foundations of scalable perception and codified knowledge, I finally introduce **GeoLaneRep** to achieve the primary objective of a constructive and interactive digital twin. While conventional digital twins often remain descriptive, observational replicas, GeoLaneRep encodes each lane as a functional, behavior-grounded traffic unit. By jointly encapsulating structural geometry, dynamic vehicle trajectories, and interpretable operational descriptors into a shared latent space, it provides a queryable semantic interface for digital twin operation. This interoperable layer transforms the digital twin into a constructive environment for controlled infrastructure modification and an interactive environment for semantically informed decision-making. By enabling behavior-conditioned geometry generation and closed-loop analyses, this dissertation envisions a digital twin-enabled ATMS that not only senses and synchronizes but actively reasons, adapts, and translates roadway infrastructure changes into system-verifiable intents for proactive traffic management.

6.1 Main Contributions

The main contributions of this dissertation can be categorized into three key areas:

1. **Road Geometry Sensing and Scalable Digital Twin Synchronization**

   This study presents Geo-ORBIT, a scalable and privacy-preserving framework for real-time lane-level road geometry perception and digital twin syn-

chronization. Leveraging federated meta-learning, the framework enables adaptive geometry understanding across heterogeneous roadside camera deployments without sharing raw data, thereby preserving confidentiality while significantly reducing communication overhead. Geo-ORBIT achieves fine-grained lane reconstruction and robust generalization to unseen locations, establishing the scalable perception foundation needed for continuously updated roadway digital twins.

2. **Knowledge Management and Semantic Reasoning for Transportation Analysis**

   This study introduces CrossTraffic, an open and modular framework designed to codify and operationalize the Highway Capacity Manual for improved transparency, adaptability, and reproducibility in traffic analysis. The framework integrates a structured transportation knowledge stack, interoperable middleware, and a multi-platform interface that enables researchers and practitioners to access, extend, and validate computational procedures. At its semantic layer, CrossTraffic incorporates LLMs and RAG to support natural language interaction with transportation manuals, enabling efficient knowledge retrieval and dynamic analytical workflows. Demonstration on the real-world highway showed the effectiveness and consistent performance of the proposed framework across interfaces and LLM agents. Semantic validator validates highway scenarios with more than 80% + threshold tests, completely extinguishes the false inputs.

3. **Behavior-Grounded Lane Representation for Constructive Digital Twins**

   GeoLaneRep is a cohesive representation learning framework that elevates traffic digital twins from descriptive replicas to constructive and interactive management tools with semantic awareness. Rather than modeling lanes merely as static spatial coordinates, GeoLaneRep encodes each lane as a functional traffic unit by jointly fusing structural geometry, dynamic vehicle trajectories, and interpretable operational descriptors into a shared latent

space. This behavior-grounded embedding serves as a queryable and editable semantic interface between raw observation, digital twin, and traffic simulation. By enabling behavior-conditioned geometry generation and temporal reasoning as downstream tasks, the framework supports controlled infrastructure modification and semantically informed human-in-the-loop intervention, translating low-level roadway changes into system-verifiable intents.

6.2 Limitations and Future Work

Despite the capabilities introduced in constructive and interactive digital twins, several technical and operational boundaries remain. Addressing these limitations presents clear pathways for future research.

1. **Predictive Safety and Bounded Geometrical Control**

   While GeoLaneRep successfully establishes a behavior-grounded representation for constructive analyses, it currently lacks forward-looking conflict prediction. Future extensions must integrate trajectory forecasting and surrogate safety assessment models directly into the geometry generation loop. Ensuring that any algorithmically proposed lane modification is constrained by mathematical safety bounds and collision risk thresholds is essential before deployment in active traffic management.

2. **V2X Integration and Multi-Agent Data Fusion**

   The current frameworks rely predominantly on infrastructure-mounted vision sensors to construct the digital twin. A critical next step is integrating Vehicle-to-Everything (V2X) communication streams. Fusing macro-level roadside camera perception with high-fidelity, micro-level telematics and intended agent trajectories will resolve occlusion issues, reduce latency, and create a comprehensive, multi-modal semantic representation of the roadway.

3. **Generalization to Diverse Traffic Scenes via Self-Supervised Learning**

The findings and framework validations presented in this dissertation are primarily highway-centric. Scaling these models to handle complex urban topologies remains an open challenge, largely due to the reliance on annotated training data. Achieving network-level scalability and adaptability across novel road geometries will require shifting toward self-supervised or unsupervised representation learning, allowing the system to learn lane semantics purely from continuous observation without the labeling bottleneck.

4. **Human Computer Interaction and Operational Credibility**

   As the digital twin becomes more interactive and automated, the human-in-the-loop experience must be refined. Future research should prioritize transparent semantic reasoning, ensuring that when the system proposes a geometry or operational shift, the underlying logic is explainable to traffic engineers. Building intuitive, natural language interfaces will be key to building operational credibility and practical adoption.

5. **Real-Time Performance under Edge Conditions**

   The proposed frameworks have not yet been heavily constrained by the computational limits of field hardware. Deploying these models natively on roadside units requires further optimization of the neural architectures to ensure low-latency, real-time inference and synchronization under constrained edge-computing environments.

REFERENCES

Adarbah, Haitham Y., Mehdi Sookhak, and Mohammed Atiquzzaman. 2023. A Digital Twin Environment for 5G Vehicle-to-Everything: Architecture and Open Issues. In *Proceedings of the Int'l ACM Symposium on Performance Evaluation of Wireless Ad Hoc, Sensor, & Ubiquitous Networks*, 115–122. Montreal Quebec Canada: ACM.

———. 2024. A digital twin-based traffic light management system using birch algorithm. *Ad Hoc Networks* 164:103613.

Alahi, Alexandre. 2026. Chapter 9 - foundational principles for representation learning: applications in autonomous mobility. In *Mobility patterns, big data and transport analytics (second edition)*, ed. Constantinos Antoniou, Loukas Dimitriou, and Francisco Câmara Pereira, second edition ed., 265–286. Elsevier.

Arman, Mohammad Ali, and Chris M.J. Tampère. 2021. Lane-level routable digital map reconstruction for motorway networks using low-precision gps data. *Transportation Research Part C: Emerging Technologies* 129:103234.

Bai, X., S. He, Y. Li, X. Yabo, Z. Xin, D. Wenli, and L. Jian-Rong. 2025. Construction of a knowledge graph for framework material enabled by large language models and its application. *npj Computational Materials* 11(51).

Bolisani, E., and C. Bratianu. 2018. *The elusive definition of knowledge*, vol. None of *Knowledge Management and Organizational Learning*. None ed. Springer.

Bongomin, Ocident, Mwewa Chikonkolo Mwape, Nonsikelelo Sheron Mpofu, Brendah Kembabazi Bahunde, Richard Kidega, Ibrahim Luqman Mpungu, Godias Tumusiime, Cynthia Awuor Owino, Yannick Marnaigue Goussongtogue, Aregawi Yemane, Proscovia Kyokunzire, Clement Malanda, Jimmy Komakech, Dan Tigalana, Onesmas Gumisiriza, and George Ngulube. 2025. Digital twin technology advancing industry 4.0 and industry 5.0 across sectors. *Results in Engineering* 26:105583.

Bubeck, S., V. Chandrasekaran, R. Eldan, J. Gehrke, E. Horvitz, E. Kamar, P. Lee, Y. T. Lee, Y. Li, S. Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Cai, G., B. Fan, Y. Dong, T. Li, Y. Wu, and Y. Zhang. 2023. Task-efficiency oriented v2x communications: Digital twin meets mobile edge computing. *IEEE Wireless Communications* 1–8.

Cao, Xu, Tong Zhou, Yunsheng Ma, Wenqian Ye, Can Cui, Kun Tang, Zhipeng Cao, Kaizhao Liang, Ziran Wang, James M. Rehg, and Chao Zheng. 2024. Maplm: A real-world large-scale vision-language benchmark for map and traffic scene understanding. In *2024 ieee/cvf conference on computer vision and pattern recognition (cvpr)*, 21819–21830.

Cazzella, Lorenzo, Francesco Linsalata, Maurizio Magarini, Matteo Matteucci, and Umberto Spagnolini. 2024. A multi-modal simulation framework to enable digital twin-based v2x communications in dynamic environments. In *2024 ieee 100th vehicular technology conference (vtc2024-fall)*, 1–6. IEEE.

Chowdhury, M., K. Dey, and A. Apon. 2024. *Data analytics for intelligent transportation systems*. Elsevier.

Cui, C., Y. Ma, X. Cao, W. Ye, Y. Zhou, K. Liang, J. Chen, J. Lu, Z. Yang, K. Liao, T. Gao, E. Li, K. Tang, Z. Cao, T. Zhou, A. Liu, X. Yan, S. Mei, J. Cao, Z. Wang, and C. Zheng. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the ieee/cvf winter conference on applications of computer vision (wacv) workshops*, 958–979.

Dasgupta, Sagar, Mizanur Rahman, and Steven Jon. 2024. Harnessing digital twin technology for adaptive traffic signal control: Improving signalized intersection performance and user satisfaction. *IEEE Internet of Things Journal* 11(22):36596–36618.

Dasgupta, Sagar, Mizanur Rahman, Abhay D. Lidbe, Weike Lu, and Steven Jones. 2023. A Transportation Digital-Twin Approach for Adaptive Traffic Control Systems. ArXiv:2109.10863 [physics].

Davletshina, Diana, Varun Kumar Reja, and Ioannis Brilakis. 2024. Automating construction of road digital twin geometry using context and location aware segmentation. *Automation in Construction* 168:105795.

Dong, Jianghong, Qing Xu, Jiawei Wang, Chunying Yang, Mengchi Cai, Chaoyi Chen, Yu Liu, Jianqiang Wang, and Keqiang Li. 2023. Mixed cloud control testbed: Validating vehicle-road-cloud integration via mixed digital twin. *IEEE Transactions on Intelligent Vehicles* 8(4):2723–2736.

Duan, Zhizhao, Hao Cheng, Duo Xu, Xi Wu, Xiangxie Zhang, Xi Ye, and Zhen Xie. 2024. Cityllava: Efficient fine-tuning for vlms in city scenario. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, 7180–7189.

Feng, Hailin, Haibin Lv, and Zhihan Lv. 2023. Resilience towarded digital twins to improve the adaptability of transportation systems. *Transportation Research Part A: Policy and Practice* 173:103686.

Feng, Yuxiang, Leah Camarcat, Paraskevi Koliou, Fahmy Adan, Paraskevi Michalaki, Nicolette Formosa, Georgios Zacharopoulos, Panagiotis Angeloudis, and Mohammed Quddus. 2025. Innovative interventions for transforming road capacity, safety and emissions on the england's strategic road network. *Research in Transportation Business & Management* 61:101398.

Fernandez, S., R. Hadfi, T. Ito, I. Marsa-Maestre, and J. R Velasco. 2016. Ontology-based architecture for intelligent transportation systems using a traffic sensor network. *Sensors* 16(8):1287.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Fu, Yanping, Wenbin Liao, Xinyuan Liu, Hang Xu, Yike Ma, Yucheng Zhang, and Feng Dai. 2024a. Topologic: An interpretable pipeline for lane topology reasoning on driving scenes. In *The thirty-eighth annual conference on neural information processing systems*.

Fu, Yongjie, Mehmet K Turkcan, Vikram Anantha, Zoran Kostic, Gil Zussman, and Xuan Di. 2024b. Digital twin for pedestrian safety warning at a single urban traffic intersection. In *2024 ieee intelligent vehicles symposium (iv)*, 2640–2645. IEEE.

Fuller, Aidan, Zhong Fan, Charles Day, and Chris Barlow. 2020. Digital twin: Enabling technologies, challenges and open research. *IEEE access* 8:108952–108971.

Gan, B., D. Zhang, Z. Huang, F. Zheng, R. Zhu, and W. Zhang. 2025. Ontology-driven knowledge graph for decision-making in resilience enhancement of underground structures: Framework and application. *Tunnelling and Underground Space Technology* 163:106739.

Goumopoulos, C. 2024. Smart city middleware: A survey and a conceptual framework. *IEEE Access* 12:4015–4047.

Gruber, T. R. 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition* 5(2):199–220.

Gu, Xiaohui, Wei Duan, Guoan Zhang, Jia Hou, Limei Peng, Miaowen Wen, Feifei Gao, Min Chen, and Pin-Han Ho. 2026. Digital twin technology for intelligent vehicles and transportation systems: A survey on applications, challenges and future directions. *IEEE Communications Surveys & Tutorials* 28:3235–3271.

Gupta, S., R Ranjan, and S. N. Singh. 2024. A comprehensive survey of retrieval-augmented generation (rag): Evolution, current landscape and future directions. *arXiv preprint arXiv:2410.12837*.

Han, Chengkai, Jingyuan Wang, Yongyao Wang, Xie Yu, Hao Lin, Chao Li, and Junjie Wu. 2025. Bridging traffic state and trajectory for dynamic road network and trajectory representation learning. In *Proceedings of the thirty-ninth aaai conference on*

artificial intelligence and thirty-seventh conference on innovative applications of artificial intelligence and fifteenth symposium on educational advances in artificial intelligence. AAAI'25/IAAI'25/EAAI'25, AAAI Press.

Harrison, F. D., C. Brown, K. Admas, and T. Hall. 2025. Knowledge management at state departments of transportation research roadmap. Research Report 1134, National Academies of Sciences, Engineering, and Medicine.

Higgins, S. 2008. The dcc curation lifecycle model. *International Journal of Digital Curation* 3(1):134–140.

Ho, Jonathan, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *Advances in neural information processing systems* 33:6840–6851.

Hogan, A., E. Blomqvist, M. Cochez, C. d'Amato, G. D. Melo, C. Gutierrez, S. Kirrane, J. E. L. Gayo, R. Navigli, S. Neumaier, et al. 2021. Knowledge graphs. *ACM Computing Surveys (Csur)* 54(4):1–37.

Hou, Lian, Shengbo Eben Li, Bo Yang, Zheng Wang, and Kimihiko Nakano. 2023. Integrated graphical representation of highway scenarios to improve trajectory prediction of surrounding vehicles. *IEEE Transactions on Intelligent Vehicles* 8(2): 1638–1651.

Hu, Zixuan, Li Shen, Zhenyi Wang, Baoyuan Wu, Chun Yuan, and Dacheng Tao. 2023. Learning to learn from apis: Black-box data-free meta-learning. In *International conference on machine learning*, 13610–13627. PMLR.

Huang, H., Y. Tang, X. Shi, and X. Mao. 2024. Dependency-uware neural topic model. *Information Processing & Management* 61(1):103530.

Hui, Y., X. Ma, Z. Su, N. Cheng, Z. Yin, T. Luan, and Y. Chen. 2022. Collaboration as a service: Digital-twin-enabled collaborative and distributed autonomous driving. *IEEE Internet Of Things Journal* 9:18607–18619.

Irani, Z., R. M. Abril, V. Weerakkody, A. Omar, and U. Sivarajah. 2023. The impact of legacy systems on digital transformation in european public administration:

Lesson learned from a multi case analysis. *Government Information Quarterly* 40(1): 101784.

Irfan, I., M. S. U. K. Sumbal, F. Khurshid, and F.TS Chan. 2022. Toward a resilient supply chain model: critical role of knowledge management and dynamic capabilities. *Industrial management & data systems* 122(5):1153–1182.

Irfan, Muhammad Sami, Sagar Dasgupta, and Mizanur Rahman. 2024. Toward transportation digital twin systems for traffic safety and mobility: A review. *IEEE Internet of Things Journal* 11(14):24581–24603.

Jacobson, Les, and Katie McLaughlin. 2024. Next generation of traffic management systems and centers: A primer. report FHWA-HRT-24-081, Federal Highway Administration.

Janowicz, K., P. Hitzler, W. Li, D. Rehberger, M. Schildhauer, R. Zhu, C. Shimizu, C. Fisher, L. Cai, G. Mai, et al. 2022. Know, know where, knowwheregraph: A densely connected, cross-domain knowledge graph and geo-enrichment service stack for applications in environmental intelligence. *AI Magazine* 43(1):30–39.

Jarrahi, M. H., D. Askay, A. Eshraghi, and P. Smith. 2023. Artificial intelligence and knowledge management: A partnership between human and ai. *Business Horizons* 66(1):87–99.

Jenks, C. W., C. Hedges, A. C. Lemer, S. Moore, E. P. Delaney, and S. E. Hitchcock. 2015. A guide to agency-wide knowledge management for state departments of transportation. NCHRP Report 813, National Academies of Sciences, Engineering, and Medicine.

Jiang, Feng, Ling Ma, Tim Broyd, Weiya Chen, and Hanbin Luo. 2022. Building digital twins of existing highways using map data based on engineering expertise. *Automation in Construction* 134:104081.

Jiang, Jiawei, Dayan Pan, Houxing Ren, Xiaohan Jiang, Chao Li, and Jingyuan Wang. 2023. Self-supervised trajectory representation learning with temporal

regularities and travel semantics. In *2023 ieee 39th international conference on data engineering (icde)*, 843–855.

Jocher, Glenn, Jing Qiu, and Ayush Chaurasia. 2023. Ultralytics YOLO.

Jones, David, Chris Snider, Aydin Nassehi, Jason Yon, and Ben Hicks. 2020. Characterising the digital twin: A systematic literature review. *CIRP journal of manufacturing science and technology* 29:36–52.

K., Megan, and Mark F. 2018. Ontologies for transportation research: A survey. *Transportation Research Part C: Emerging Technologies* 89:53–82.

Kairouz, Peter, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning* 14(1–2):1–210.

Khan, Latif U., Ehzaz Mustafa, Junaid Shuja, Faisal Rehman, Kashif Bilal, Zhu Han, and Choong Seon Hong. 2024. Federated learning for digital twin-based vehicular networks: Architecture and challenges. *IEEE Wireless Communications* 31(2):156–162.

Khanam, Rahima, and Muhammad Hussain. 2024. Yolov11: An overview of the key architectural enhancements. *arXiv preprint arXiv:2410.17725*.

Kleinsteuber, E., T. Al Mustafa, F. Zander, B. König-Ries, and S. Babalou. 2024. Managing provenance data in knowledge graph management platforms. *Datenbank Spektrum* 24:43–52.

Kritzinger, Werner, Matthias Karner, Georg Traar, Jan Henjes, and Wilfried Sihn. 2018. Digital twin in manufacturing: A categorical literature review and classification. *Ifac-PapersOnline* 51(11):1016–1022.

Kuciemba, Steve, Les Jacobson, Arianne Mizuta, and David Nguyen. 2023. Review of traffic management systems — current practice. report FHWA-HRT-23-051, Federal Highway Administration.

Kuhn, B., D. Gopalakrishna, and E. Schreffler. 2013. The active transportation and demand management program (atdm): Lessons learned. Tech. Rep. FHWA-HOP-13-018, Federal Highway Administration.

Kušić, K., R. Schumann, and E. Ivanjko. 2023. A digital twin in transportation: Real-time synergy of traffic data streams and simulation for virtualizing motorway dynamics. *Advanced Engineering Informatics* 55:101858.

Lei, Da, Min Xu, and Shuaian Wang. 2024. A conditional diffusion model for probabilistic estimation of traffic states at sensor-free locations. *Transportation Research Part C: Emerging Technologies* 166:104798.

Lewis, P., E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33:9459–9474.

Li, Li, Yuxi Fan, Mike Tse, and Kuo-Yi Lin. 2020. A review of applications in federated learning. *Computers & Industrial Engineering* 149:106854.

Li, Shuhao, Yue Cui, Jingyi Xu, Libin Li, Lingkai Meng, Weidong Yang, Fan Zhang, and Xiaofang Zhou. 2025a. Unifying lane-level traffic prediction from a graph structural perspective: Benchmark and baseline. *IEEE Transactions on Knowledge and Data Engineering*.

Li, Shuhao, Weidong Yang, Yue Cui, Xiaoxing Liu, Lingkai Meng, Lipeng Ma, and Fan Zhang. 2025b. Fine-grained traffic inference from road to lane via spatio-temporal graph node generation. In *Proceedings of the 31st acm sigkdd conference on knowledge discovery and data mining v.2*, 1529–1540. KDD '25, New York, NY, USA: Association for Computing Machinery.

Liang, Shiyi, Xinyuan Chang, Changjie Wu, Huiyuan Yan, Yifan Bai, Xinran Liu, Hang Zhang, Yujian Yuan, Shuang Zeng, Mu Xu, et al. 2026. Persistent autoregressive mapping with traffic rules for autonomous driving. In *Proceedings of the aaai conference on artificial intelligence*, vol. 40, 6862–6870.

Liao, Bencheng, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. 2022. Maptr: Structured modeling and learning for online vectorized hd map construction. *arXiv preprint arXiv:2208.14437*.

Liao, Guibiao, Jiankun Li, and Xiaoqing Ye. 2024a. Vlm2scene: Self-supervised image-text-lidar learning with foundation models for autonomous driving scene understanding. *Proceedings of the AAAI Conference on Artificial Intelligence* 38(4): 3351–3359.

Liao, Xiwen, Supeng Leng, Yao Sun, Ke Zhang, and Muhammad Ali Imran. 2024b. A digital twin-based traffic guidance scheme for autonomous driving. *IEEE Internet of Things Journal*.

Liu, Qian, Xiaoqing Qi, Shanshan Liu, Xiuju Cheng, Xin Ke, and Feng Wang. 2022. Application of lightweight digital twin system in intelligent transportation. *IEEE Journal of Radio Frequency Identification* 6:729–732.

Lopez, P. A., M. Behrisch, L. Bieker-Walz, J. Erdmann, Y. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner. 2018. Microscopic traffic simulation using sumo. In *2018 21st international conference on intelligent transportation systems (itsc)*, 2575–2582.

Lu, Y., B. Yao, H. Gu, J. Huang, Z. J. Wang, Y. Li, J. Gesi, Q. He, T. J. Li, and D. Wang. 2025. Uxagent: An llm agent-based usability testing framework for web design. In *Proceedings of the extended abstracts of the chi conference on human factors in computing systems*. CHI EA '25, New York, NY, USA: Association for Computing Machinery.

Luo, Junjie, Pengyuan Liu, Wenhui Xu, Tianhong Zhao, and Filip Biljecki. 2025. A perception-powered urban digital twin to support human-centered urban planning and sustainable city development. *Cities* 156:105473.

Luo, Sheng, Wei Chen, Wanxin Tian, Rui Liu, Luanxuan Hou, Xiubao Zhang, Haifeng Shen, Ruiqi Wu, Shuyi Geng, Yi Zhou, et al. 2024. Delving into multi-modal multi-task foundation models for road scene understanding: From learning paradigm perspectives. *IEEE Transactions on Intelligent Vehicles*.

Mao, Zhenyu, Ziyue Li, Dedong Li, Lei Bai, and Rui Zhao. 2022. Jointly contrastive representation learning on road network and trajectory. In *Proceedings of the 31st acm international conference on information & knowledge management*, 1501–1510.

Masri, S., H. I. Ashqar, and M. Elhenawy. 2025. Large language models (llms) as traffic control systems at urban intersections: A new paradigm. *Vehicles* 7(1).

Matsakis, N. D., and Felix S. K. 2014. The rust language. *Proceedings of the 2014 ACM SIGAda annual conference on High integrity language technology*.

McBride, B. 2004. *The resource description framework (rdf) and its vocabulary description language rdfs*, 51–65. Berlin, Heidelberg: Springer Berlin Heidelberg.

Miller, Kevin, John Horner, Jeff Adler, and Robert Sanchez. 2021. Decision support for traffic management systems—current practices. Final Report FHWA-HRT-21-108, Federal Highway Administration, U.S. Department of Transportation, Washington, DC. September 2017–November 2021.

Mora, M., F. Wang, J. M. Gómez, and G. Phillips-Wren. 2022. Development methodologies for ontology-based knowledge management systems: A review. *Expert Systems* 39(2):e12851.

Moradi, M., A. Aghaie, and M. Hosseini. 2013. Knowledge-collector agents: Applying intelligent agents in marketing decisions with knowledge management approach. *Knowledge-Based Systems* 52:181–193.

Moroni, Stefano. 2025. Insurmountable limitations of city-scale digital twins? on urban knowledge and planning. *Computational Urban Science* 5(17).

Mu, Junsheng, Wenjiang Ouyang, Tao Hong, Weijie Yuan, Yuanhao Cui, and Zexuan Jing. 2023. Digital twins-enabled federated learning in mobile networks: From the perspective of communication-assisted sensing. *IEEE Journal on Selected Areas in Communications* 41(10):3230–3241.

Naumann, Alexander, Felix Hertlein, Daniel Grimm, Maximilian Zipfl, Steffen Thoma, Achim Rettinger, Lavdim Halilaj, Juergen Luettin, Stefan Schmid, and

Holger Caesar. 2023. Lanelet2 for nuscenes: Enabling spatial semantic relationships and diverse map-based anchor paths. In *2023 ieee/cvf conference on computer vision and pattern recognition workshops (cvprw)*, 3248–3257.

Nie, T., J. Sun, and W. Ma. 2025. Exploring the roles of large language models in reshaping transportation systems: A survey, framework, and roadmap. *Artificial Intelligence for Transportation* 1:100003.

Olan, F., O. E. Arakpogun, J. Suklan, F. Nakpodia, N. Damij, and U. Jayawickrama. 2022. Artificial intelligence and knowledge sharing: Contributing factors to organizational performance. *Journal of Business Research* 145:605–615.

Oman, L., A. J. Wilson, and F. D. Harrison. 2009. Implementing transportation knowledge networks. Research Report 20-75, National Academies of Sciences, Engineering, and Medicine.

Oord, Aaron van den, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Pan, Yuandong, Mudan Wang, Linjun Lu, Ran Wei, Stefano Cavazzi, Matt Peck, and Ioannis Brilakis. 2024. Scan-to-graph: automatic generation and representation of highway geometric digital twins from point cloud data. *Automation in Construction* 166:105654.

Pangaribuan, L., and A. Satrya. 2024. The role of knowledge management, transformational leadership, and organizational commitment on employee performance: empirical study in public sector. *Journal of Theory and Applied Management* 17: 355–371.

Perez, Ethan, Florian Strub, Harm De Vries, Vincent Dumoulin, and Aaron Courville. 2018. Film: Visual reasoning with a general conditioning layer. In *Proceedings of the aaai conference on artificial intelligence*, vol. 32.

Perna, Mariana, Bernardo Pinto, José Mendes, Rafaela Dias, Filipe Obrist, Pedro Rito, Susana Sargento, Duarte Raposo, and Filipe Pinto. 2025. Innovating urban

mobility with digital twins: Data-driven traffic visualization and testing. In *2025 ieee wireless communications and networking conference (wcnc)*, 1–6.

Poggenhans, Fabian, Jan-Hendrik Pauls, Johannes Janosovits, Stefan Orf, Maximilian Naumann, Florian Kuhnt, and Matthias Mayr. 2018. Lanelet2: A high-definition map framework for the future of automated driving. In *2018 21st international conference on intelligent transportation systems (itsc)*, 1672–1679.

Polak, M.P., and D. Morgan. 2024. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nat Commun* 15(1569).

PyO3 Project and Contributors. 2017–2025. PyO3. `https://github.com/PyO3/pyo3`.

Qiu, Mei, Lauren Christopher, Stanley Yung-Ping Chien, and Yaobin Chen. 2024. Intelligent highway adaptive lane learning system in multiple rois of surveillance camera video. *IEEE Transactions on Intelligent Transportation Systems* 25(8):8591–8601.

Ren, Jianqiang, Yangzhou Chen, Le Xin, and Jianjun Shi. 2014. Lane detection in video-based intelligent transportation monitoring via fast extracting and clustering of vehicle motion trajectories. *Mathematical Problems in Engineering* 2014(1):156296.

Riehl, K., Kouvelas A., and Michail A. M. 2025. Revisiting reproducibility in transportation simulation studies. *European Transport Research Review* 17(22).

Rivera, Esteban, Jannik Lübberstedt, Nico Uhlemann, and Markus Lienkamp. 2025. Scenario understanding of traffic scenes through large visual language models. In *Proceedings of the winter conference on applications of computer vision*, 1037–1045.

Rudskoy, Andrey, Igor Ilin, and Andrey Prokhorov. 2021. Digital twins in the intelligent transport systems. *Transportation Research Procedia* 54:927–935. International Scientific Siberian Transport Forum - TransSiberia 2020.

Santana, E. F. Z., A. P. Chaves, M. A. Gerosa, F. Kon, and D. S. Milojicic. 2017. Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture. *ACM Comput. Surv.* 50(6).

Scholtes, Maike, Lukas Westhofen, Lara Ruth Turner, Katrin Lotto, Michael Schuldes, Hendrik Weber, Nicolas Wagener, Christian Neurohr, Martin Herbert Bollmann, Franziska Körtke, et al. 2021. 6-layer model for a structured description and categorization of urban traffic and environment. *IEEE Access* 9:59131–59147.

Schroeder, Greyce N., Charles Steinmetz, Carlos E. Pereira, and Danubia B. Espindola. 2016. Digital twin data modeling with automationml and a communication methodology for data exchange. *IFAC-PapersOnLine* 49(30):12–17. 4th IFAC Symposium on Telematics Applications TA 2016.

Shankar, S., J.D. Zamfirescu-Pereira, B. Hartmann, A. Parameswaran, and I. Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th annual acm symposium on user interface software and technology*. UIST '24, New York, NY, USA: Association for Computing Machinery.

Studer, R., V.R. Benjamins, and D. Fensel. 1998. Knowledge engineering: Principles and methods. *Data & Knowledge Engineering* 25(1):161–197.

Sun, Wen, Shiyu Lei, Lu Wang, Zhiqiang Liu, and Yan Zhang. 2021. Adaptive federated learning and digital twin for industrial internet of things. *IEEE Transactions on Industrial Informatics* 17(8):5605–5614.

Tamaru, Rei, Pei Li, and Bin Ran. 2025. Geo-orbit: A federated digital twin framework for scene-adaptive lane geometry detection. *arXiv preprint arXiv:2507.08743*.

Tan, Xiaobin, Qiushi Meng, Mingyang Wang, Quan Zheng, Jun Wu, and Jian Yang. 2023. Digital twin-based cloud-native vehicular networks architecture for intelligent driving. *IEEE Network*.

Tang, Lun, Mingyan Wen, Zhenzhen Shan, Li Li, Qinghai Liu, and Qianbin Chen. 2024. Digital twin-enabled efficient federated learning for collision warning in intelligent driving. *IEEE Transactions on Intelligent Transportation Systems* 25(3): 2573–2585.

Tupayachi, J., H. Xu, O. A. Omitaomu, M. C. Camur, A. Sharmin, and X. Li. 2024. Towards next-generation urban decision support systems through ai-powered construction of scientific ontology using large language models—a case in optimizing intermodal freight transportation. *Smart Cities* 7(5):2392–2421.

VanDerHorn, E., and S. Mahadevan. 2021. Digital twin: Generalization, characterization and implementation. *Decision Support Systems* 145:113524.

Wang, Chao, Jiahui Huang, Yongheng Wang, Zhengxuan Lin, Xiongnan Jin, Xing Jin, Di Weng, and Yingcai Wu. 2024a. A deep spatiotemporal trajectory representation learning framework for clustering. *IEEE Transactions on Intelligent Transportation Systems* 25(7):7687–7700.

Wang, Kui, Zongdian Li, Kazuma Nonomura, Tao Yu, Kei Sakaguchi, Omar Hashash, and Walid Saad. 2024b. Smart mobility digital twin based automated vehicle navigation system: A proof of concept. *IEEE Transactions on Intelligent Vehicles* 9(3):4348–4361.

Wang, Xiangheng, Ziquan Fang, Chenglong Huang, Danlei Hu, Lu Chen, and Yunjun Gao. 2025. GTR: A general, multi-view, and dynamic framework for trajectory representation learning. In *Forty-second international conference on machine learning*.

Wang, Zijin, Ou Zheng, Liangding Li, Mohamed Abdel-Aty, Carolina Cruz-Neira, and Zubayer Islam. 2023. Towards next generation of pedestrian and connected vehicle in-the-loop research: A digital twin co-simulation framework. *IEEE Transactions on Intelligent Vehicles* 8(4):2674–2683.

Wang, Ziran, Rohit Gupta, Kyungtae Han, Haoxin Wang, Akila Ganlath, Nejib Ammar, and Prashant Tiwari. 2022. Mobility digital twin: Concept, architecture, case study, and future challenges. *IEEE Internet of Things Journal* 9(18):17452–17467.

Wang, Ziran, Kyungtae Han, and Prashant Tiwari. 2021. Digital twin-assisted cooperative driving at non-signalized intersections. *IEEE Transactions on Intelligent Vehicles* 7(2):198–209.

WebAssembly Project and Contributors. 2022. WebAssembly core specification. Tech. Rep., W3C.

Weerakkody, V., M. Janssen, and R. El-Haddadeh. 2021. The resurgence of business process re-engineering in public sector transformation efforts: Exploring the systemic challenges and unintended consequences. *Inf Syst E-Bus Manage* 19: 993–1014.

Wood, J., and I. Schalkwyk. 2025. Reproducibility in transportation research: Importance, best practices, and dealing with protected and sensitive data. *Journal of Transportation Technologies* 15:179–202.

Wágner, T., T. Ormándi, T. Tettamanti, and I. Varga. 2023. Spat/map v2x communication between traffic light and vehicles and a realization with digital twin. *Computers And Electrical Engineering* 106:108560.

Xu, Dongwei, Yufu Tang, Jingfei Ju, Zefeng Yu, Jiaying Zheng, Tongcheng Gu, and Haifeng Guo. 2025a. Cross-city traffic state prediction based on knowledge transfer framework. *Expert Systems with Applications* 272:126747.

Xu, G., J. Chen, Z. Wang, et al. 2025b. Enhancing traffic safety analysis with digital twin technology: integrating vehicle dynamics and environmental factors into microscopic traffic simulation. *Scientific Reports* 15:44404.

Xu, Haowen, Andy Berres, Srikanth B. Yoginath, Harry Sorensen, Phil J. Nugent, Joseph Severino, Sarah A. Tennille, Alex Moore, Wesley Jones, and Jibonananda Sanyal. 2023. Smart mobility in the cloud: Enabling real-time situational awareness

and cyber-physical control through a digital twin for traffic. *IEEE Transactions on Intelligent Transportation Systems* 24(3):3145–3156.

Yang, H., M. Siew, and C. Joe-Wong. 2024. An llm-based digital twin for optimizing human-in-the loop systems. In *2024 ieee international workshop on foundation models for cyber-physical systems & internet of things (fmsys)*, 26–31.

Ye, S., Q. Wu, P. Fan, and Q. Fan. 2025. A survey on semantic communications in internet of vehicles. *Entropy* 27(4).

Zelenbaba, Stefan, Benjamin Rainer, Markus Hofer, and Thomas Zemen. 2022. Wireless digital twin for assessing the reliability of vehicular communication links. In *2022 ieee globecom workshops (gc wkshps)*, 1034–1039. IEEE.

Zhang, Kai, Ting Qu, Dajian Zhou, Hongfei Jiang, Yuanxin Lin, Peize Li, Hongfei Guo, Yang Liu, Congdong Li, and George Q Huang. 2020. Digital twin-based opti-state control method for a synchronized production operation system. *Robotics and Computer-Integrated Manufacturing* 63:101892.

Zhang, Liang, and Cheng Long. 2023. Road network representation learning: A dual graph-based approach. *ACM Trans. Knowl. Discov. Data* 17(9).

Zhang, Q., Z. Ma, and P. et al. Zhang. 2025. Mobility knowledge graph: Review and its application in public transport. *Transportation* 52:1119–1145.

Zhang, Rongqing, Jingxin Mao, Hanqiu Wang, Bing Li, Xiang Cheng, and Liuqing Yang. 2024. A survey on federated learning in intelligent transportation systems. *IEEE Transactions on Intelligent Vehicles*.

Zhang, Zongliang, Jonathan Li, Yulan Guo, Chenhui Yang, and Cheng Wang. 2019. 3d highway curve reconstruction from mobile laser scanning point clouds. *IEEE Transactions on Intelligent Transportation Systems* 21(11):4762–4772.

Zhou, Jian, Minghao Yu, Yuan Guo, Bijun Li, Shen Ying, and Zhijiang Li. 2025. A high-definition map architecture for transportation digital twin system con-

struction. *International Journal of Applied Earth Observation and Geoinformation* 144: 104822.