Enhancing the Interconnectedness of Open Data, Traffic Engineering, and City Planning Through Utilization of Available Data

Dr. Xuesong (Simon) Zhou (xzhou74@asu.edu)

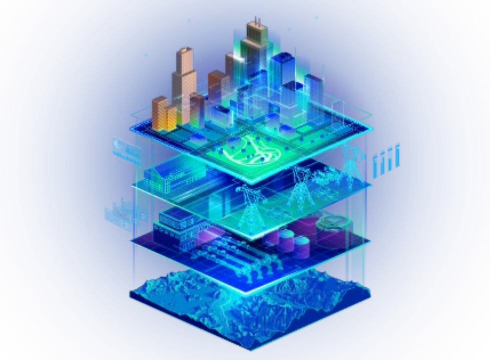
Graduate Student: Mustafa Gadah (mgadah1@asu.edu)

School of Sustainable Engineering and the Built Environment
https://github.com/asu-trans-ai-lab
Arizona State University

Prepared for TRB Webinar: State-of-the-Art Traffic Signal Simulation Tools and Platforms
Sponsored by TRB's Standing Technical Committee on Traffic Signal Systems
March 15, 2023

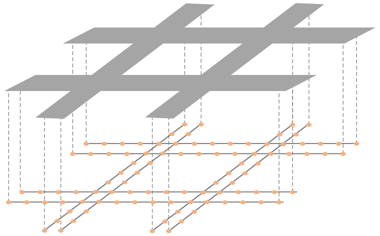
Outline

I. Open Data Specification

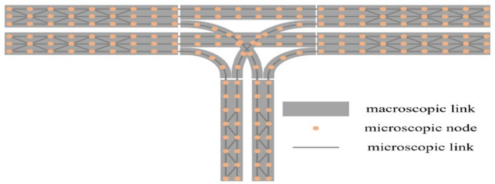


II. Open-Source Tools and MRM Community

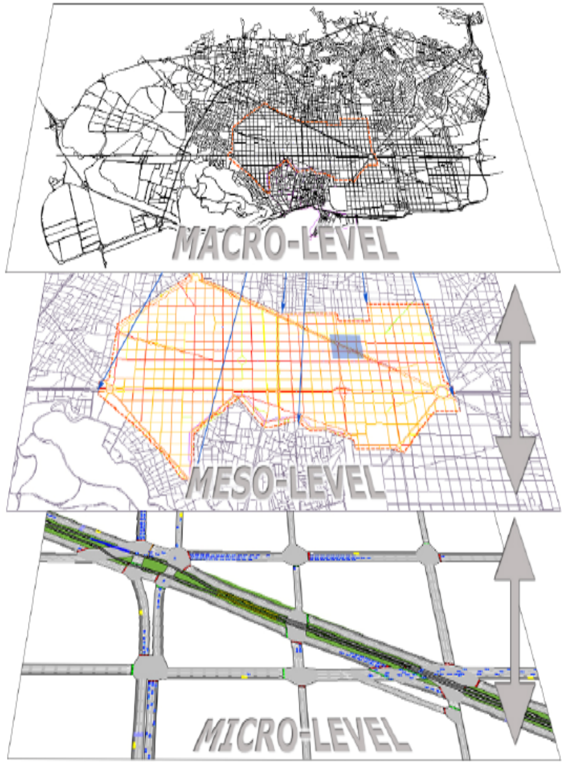
Macroscopic Layer



Microscopic Mesh Layer

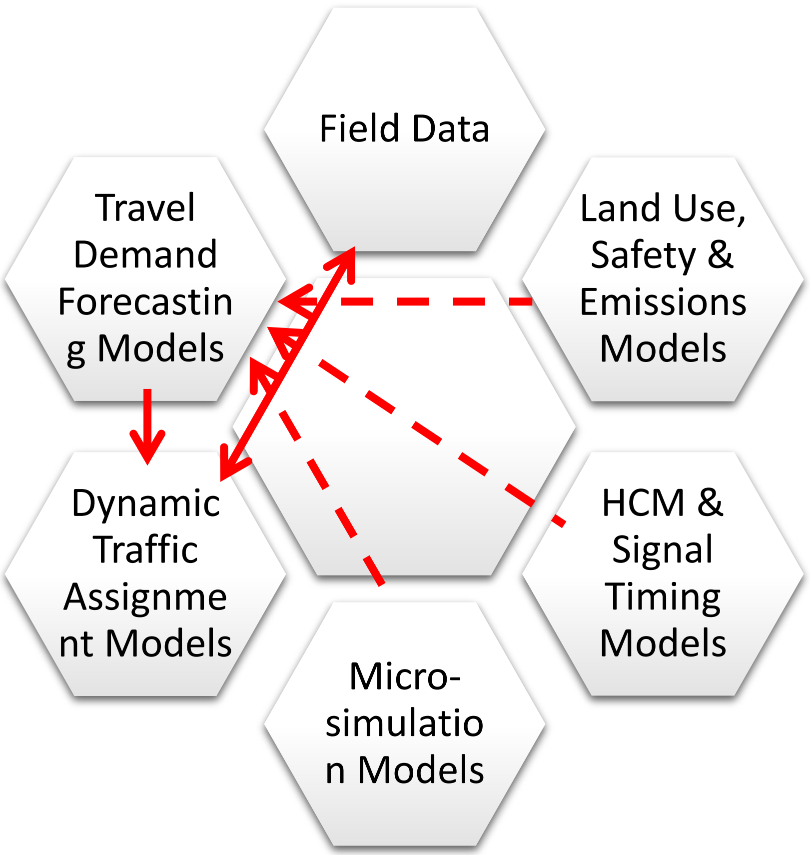


III. Network Simulation Prospective and Case Studies

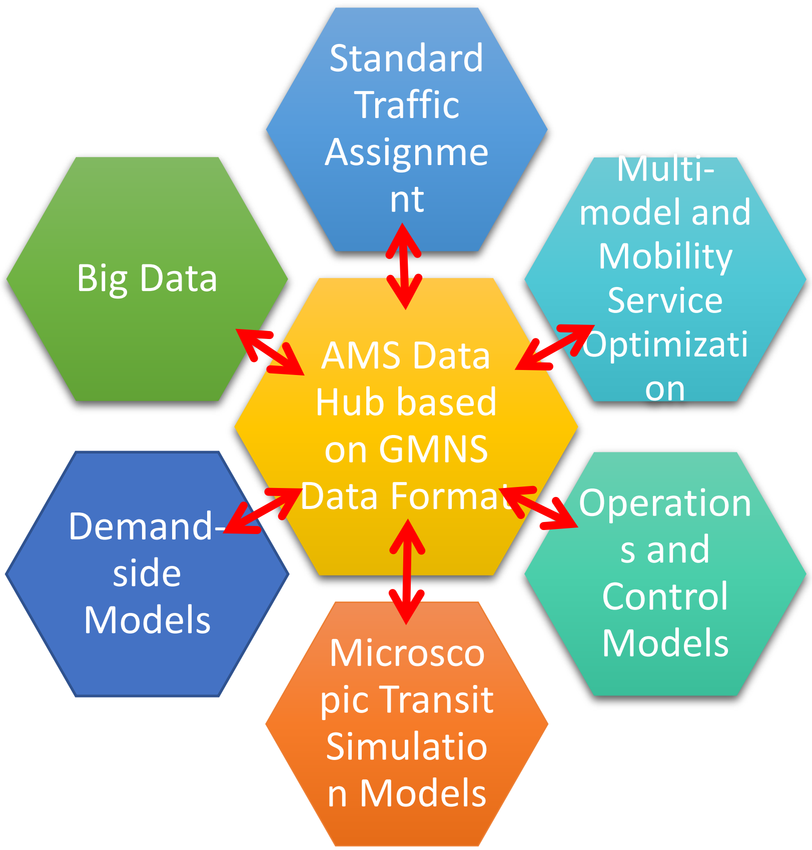


Critical Challenges for Transportation Analysis Modeling and Simulation (AMS) Data Hub

Ad Hoc Connection



Systematic Coupling

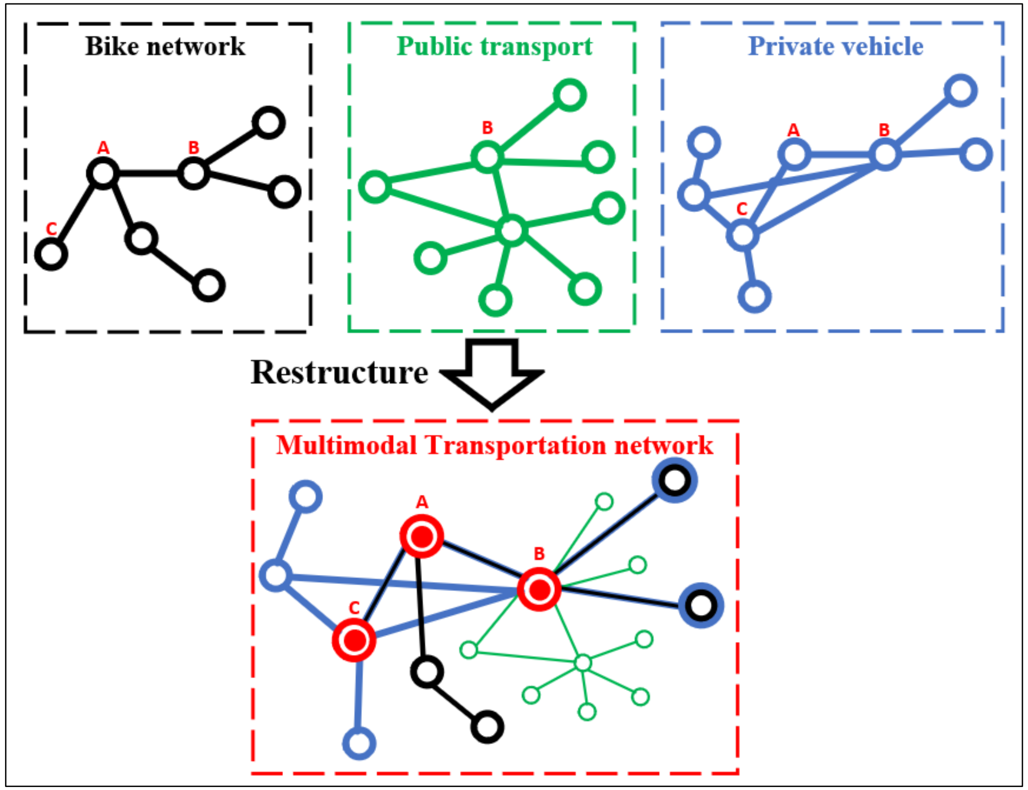


AMS Data Hub for Transportation Planning Applications

Source: Nevers, Brandon L., et al. *The effective integration of analysis, modeling, and simulation tools.* No. FHWA-HRT-13-036. United States. Federal Highway Administration. Office of Operations Research and Development, 2013.

Specific Challenges in Building Model Specifications

1. Requiring more specific information
2. A wide range of ways to code multimodal network with real-world requirements
3. Non-trivial data structure even for essential information
4. Network manipulation is not being standardized
5. Different data format options
6. Limited recent activities in bicycle or pedestrian network model standardization
7. Needs for cross-mode standards to ensure complete trip interoperability

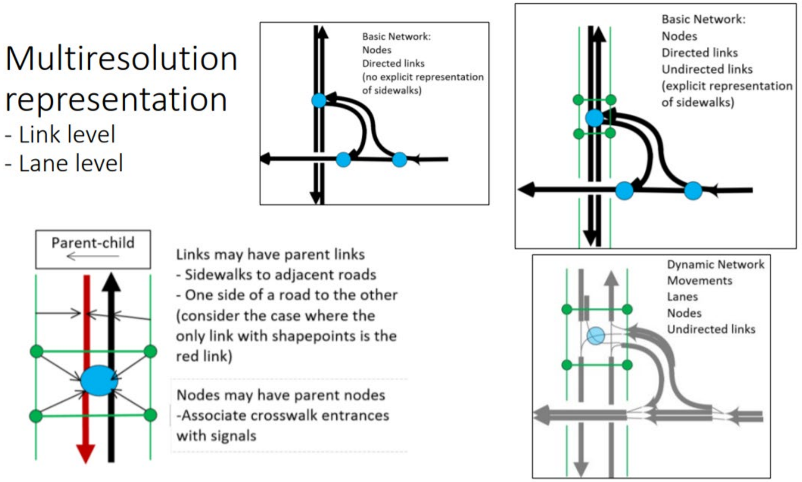


Multimodal network representation

Introducing General Modeling Network Specification (GMNS)

- The objective of the GMNS is to provide **a common human and machine-readable format** for sharing routable road network files.
- The project is overseen by a project management group, with MPO, city, industry, academic and US DOT participation.

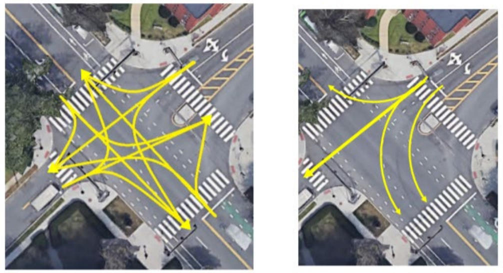
github.com/zephyr-data-specs/GMNS



Source: Volpe Center GMNS team
Scott.Smith@dot.gov
Ian.Berg@dot.gov

Movements at an intersection

- Link Level
- Lane Level (for the northeast approach)



Permitted movements may also be time-of-day specific

- Movement attributes
  - Node
  - Inbound link and lane(s)
  - Outbound link and lane(s)
  - Type of Movement
    - left, right, thru, merge, uturn
  - Type of control
    - no_control, yield, stop, stop_2_way, stop_4_way, signal
  - Optionally
    - Right-turn-on-red
    - Penalty
    - Capacity
    - Pct Green Time

Governance

This project is overseen by a board-approved Project Management Group (PMG) as follows:

- Joe Castiglione, SFCTA (chair, board representative)
- Michael Mahut, INRO
- Wu Sun, SANDAG
- Guy Rousseau, ARC
- Chetan Joshi, PTV
- Jeff Frkonja, Portland Metro
- Scott Smith, Volpe
- Natalia Ruiz Juri, University of Texas Center for Transportation Research
- Song Gao, UMass Amherst

GMNS Format (Node and Link)

Basic Data Elements

- node
- link
- geometry
- zone

name	node_id	osm_node_	osm_highw	zone_id	ctrl_type	node_type	activity_typ	is_boundar	x_coord	y_coord	main_node	poi_id	notes
	0	41459438			0				-111.928	33.4245			
	1	41520512			0				-111.944	33.42547			
	2	41520515			0				-111.944	33.42432			
	3	41520518			0				-111.944	33.42318			
	4	41520521	traffic_signals		1				-111.944	33.42189			
	5	41520523			0				-111.944	33.42064			
	6	41520525			0				-111.944	33.42035			
	7	41520528			0				-111.944	33.4194			
	8	41520531			0				-111.944	33.41892			
	9	41520533			0				-111.944	33.41778			

node.csv

name	link_id	osm_way_i	from_node	to_node_ic	dir_flag	length	lanes	free_speed	capacity	link_type_n	link_type	geometry	allowed_us	from_biwa
South Farn	0	5590095	13	14	1	81.57798	1	25		residential	6	LINESTRIN	auto	1
South Farn	1	5590095	14	13	1	81.57798	1	25		residential	6	LINESTRIN	auto	1
South Farn	2	5590095	14	15	1	80.16146	1	25		residential	6	LINESTRIN	auto	1
South Farn	3	5590095	15	14	1	80.16146	1	25		residential	6	LINESTRIN	auto	1
South Farn	4	5590095	15	16	1	240.2824	1	25		residential	6	LINESTRIN	auto	1
South Farn	5	5590095	16	15	1	240.2824	1	25		residential	6	LINESTRIN	auto	1
South Farn	6	5590095	16	17	1	84.15426	1	25		residential	6	LINESTRIN	auto	1
South Farn	7	5590095	17	16	1	84.15426	1	25		residential	6	LINESTRIN	auto	1
South Farn	8	5590095	17	18	1	83.10715	1	25		residential	6	LINESTRIN	auto	1
South Farn	9	5590095	18	17	1	83.10715	1	25		residential	6	LINESTRIN	auto	1

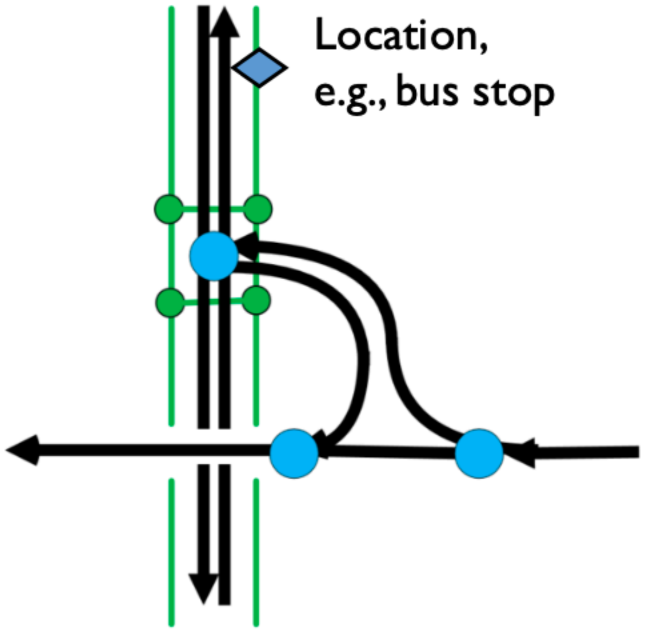
link.csv

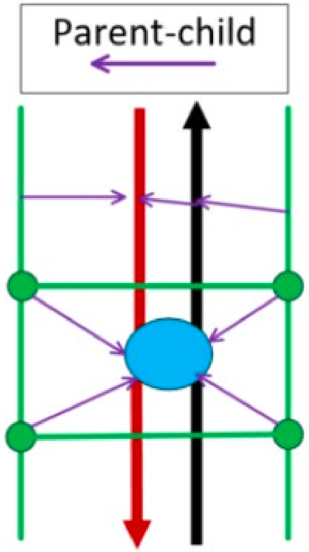
Source: https://github.com/zephyr-data-specs/GMNS

Multimodal Accommodation in GMNS

The **allowed_uses** field indicates what may flow on a **link** or **lane** (e.g., walk, bike, bus, truck, auto, hov2, hov3+), as well as non-travel uses (shoulder, parking)

Location—a point that is associated with a specific location along a link, using a linear reference



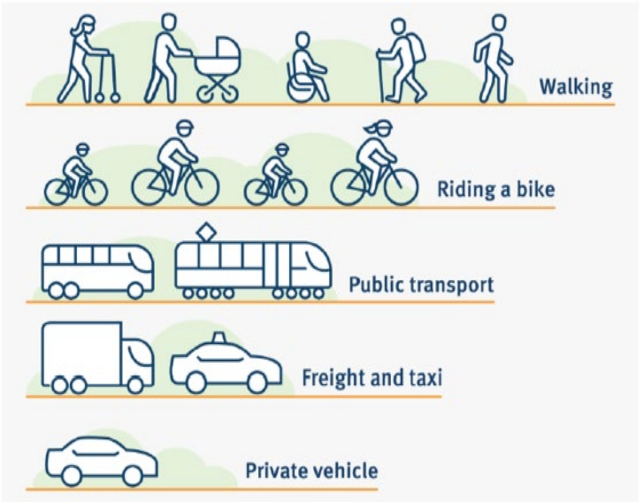


Links include fields for **ped_facility,bike_facility.**

Sidewalks and crosswalks may optionally be handled via their own undirected links. Parent-child relationships:

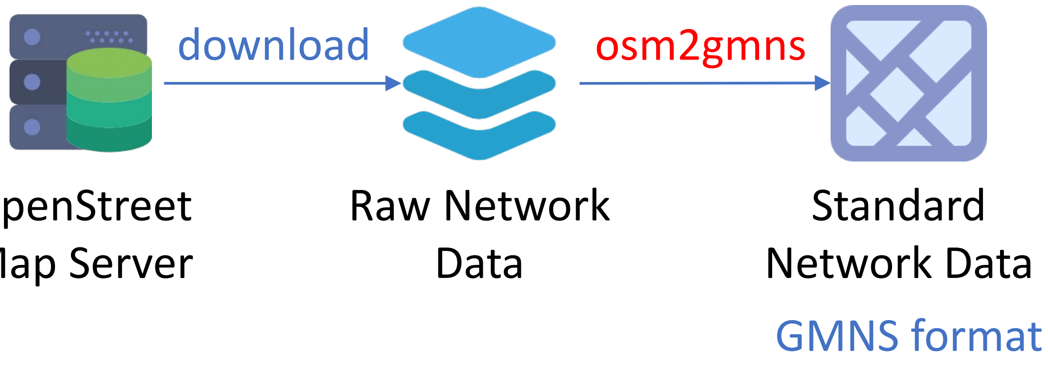
- Sidewalk with associated road
- Crosswalk and intersection nodes

Separated bike facilities may also be handled as their own pairs of directed links

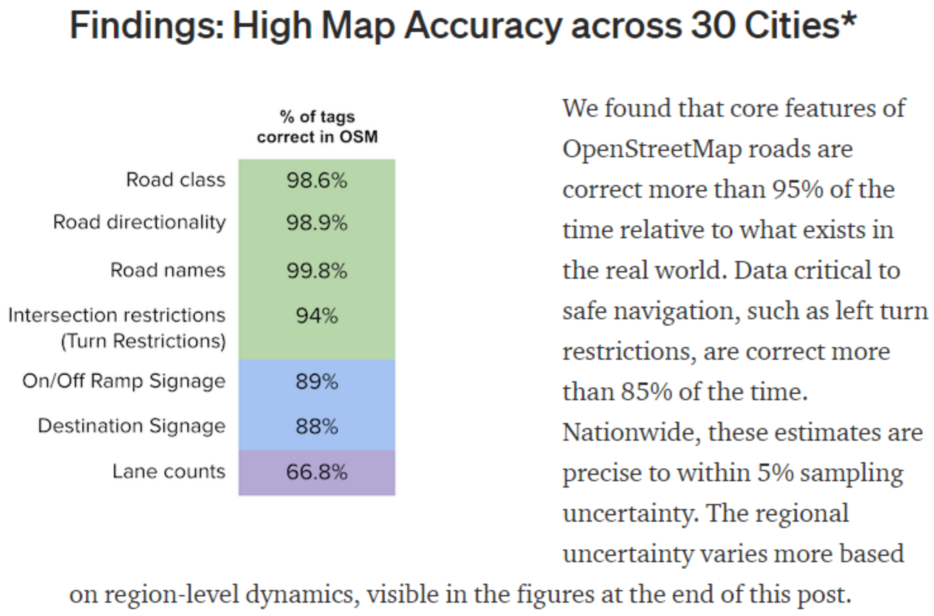


Open Data Source: OpenStreetMap

OpenStreetMap (OSM) is a free, open-source, editable map website that can provide free downloads. osm2gmns, as a data conversion tool, can directly convert the OSM map data to node and link network files in GMNS format. Users can convert and model drivable, walkable, railway, or aero way networks with a few lines of Python code.



Source: https://osm2gmns.readthedocs.io/



Source: https://eng.lyft.com/how-lyft-discovered-openstreetmap-is-the-freshest-map-for-rideshare-a7a41bf92ec

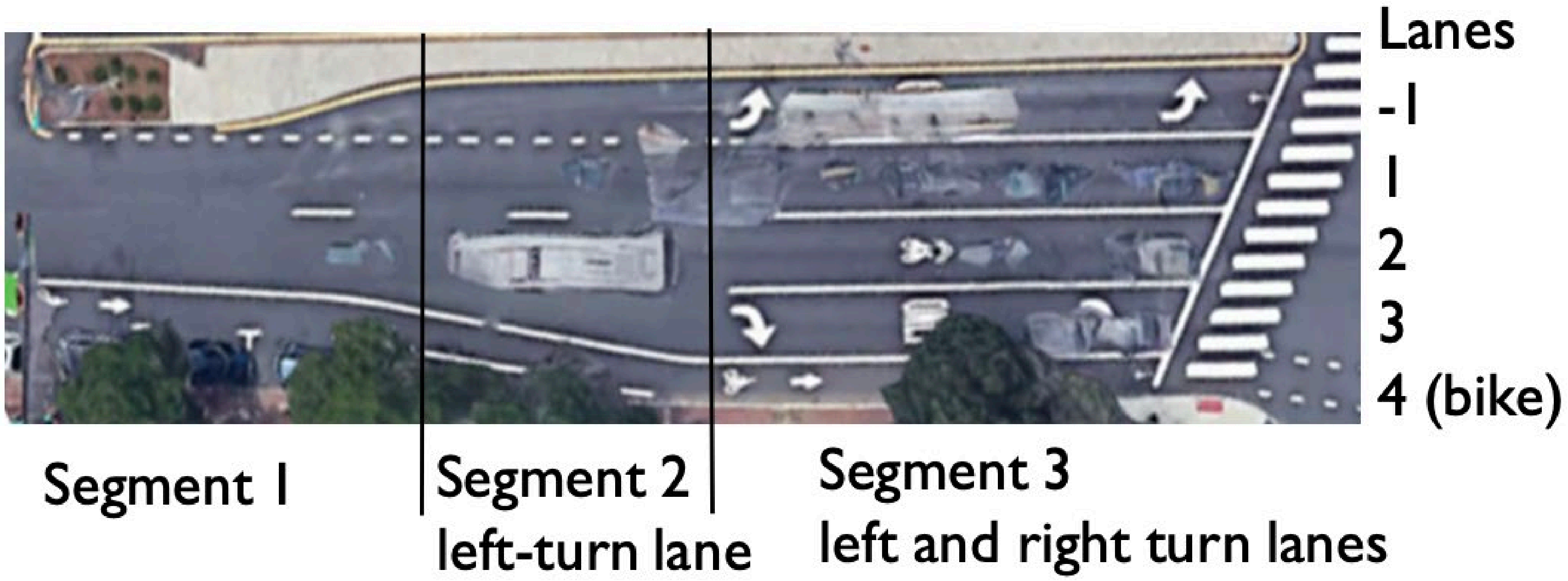
GMNS Format (Segment, Lane, Signal Timing)

Segments and Lanes:

Segment—portion of a link defined by linear references

Lane— Lanes are numbered left to right with 1 as the left-most through lane. Left turn lane is –1. A bike lane is a lane with allowed uses = BIKE

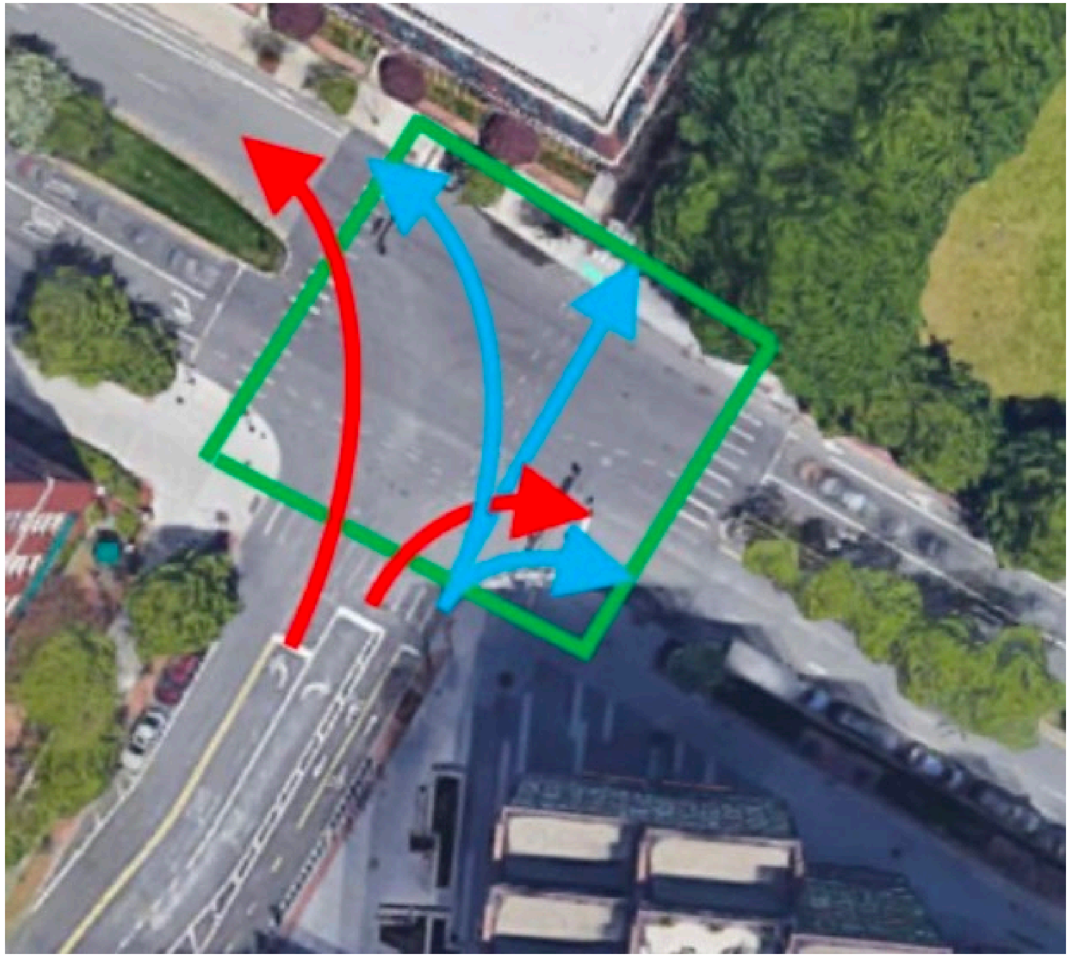
Turn pockets are defined via segments.



GMNS Format (Movement)

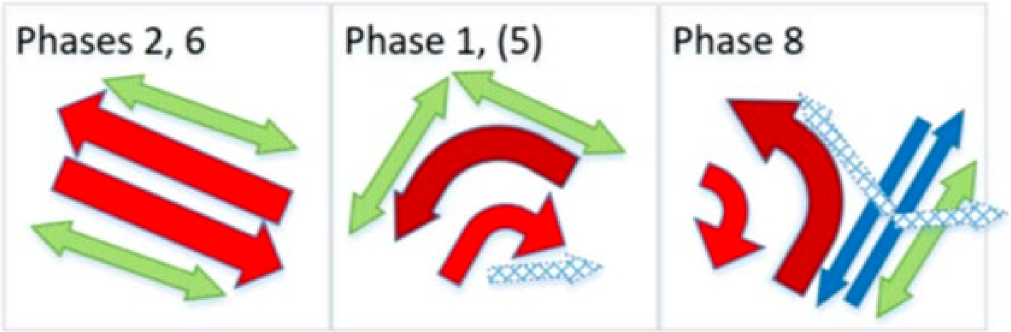
- Movements define connections and traffic control types (none, yield, stop, signal) between inbound and outbound links or lanes at an intersection.

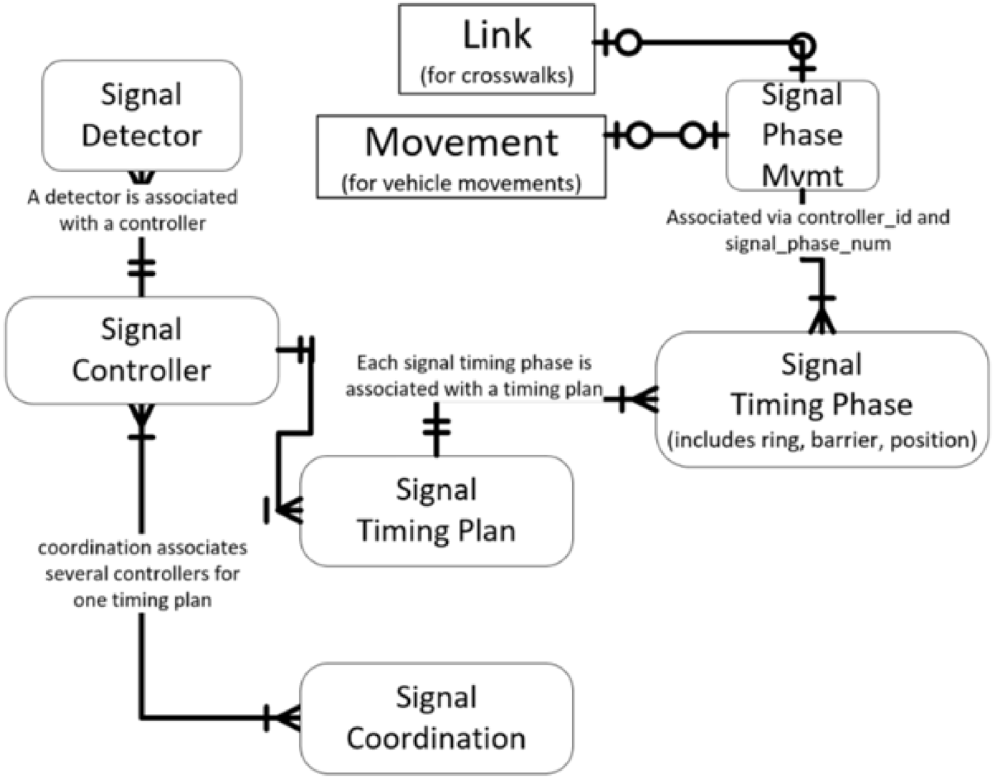
- Example:
  - Pedestrian movements in green
  - General traffic (red) and bike (blue) movements are shown from the south



GMNS Format (Traffic signal)

- **Signal_controller**— association of one or more intersections whose signals use the same controller
- **Signal_phase_mvmt**— signal_phase mapped to its associated traffic movements and pedestrian links (e.g., crosswalks)
- **Signal_timing_phase**— timing and concurrency information for each signal phase
- **Signal_timing_plan**— timing plan for the signal, by controller, time period
- **Signal_coordination**— coordination for several signal controllers, associated with a timing plan
- **Signal_detector**— traffic detector associated with a controller, a phase and a group of lanes

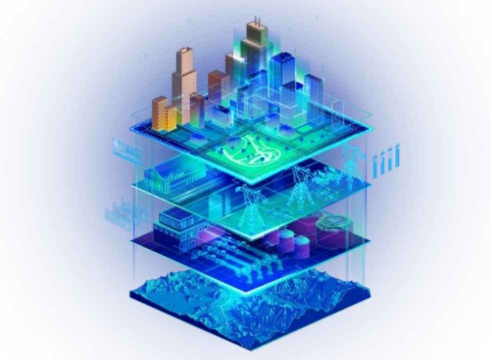




Source: GMNS.

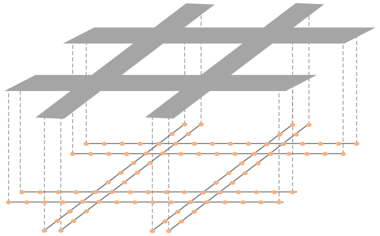
Part II: Open-Source Tools and MRM Community

I. **Open Data Specification**

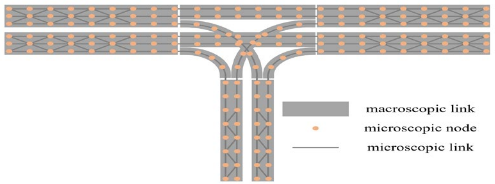


II. **Open-Source Tools and MRM Community**

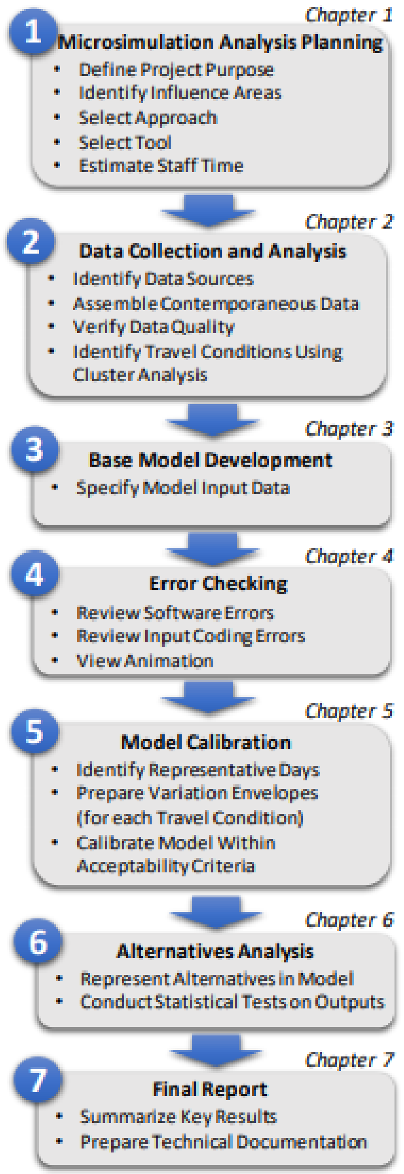
Macroscopic Layer



Microscopic Mesh Layer



General Modeling Microsimulation Modeling Procedure



Assess MRM cost-effectiveness, stakeholders, spatiotemporal scope, risks, performance measures, data requirements, data availability, and level of effort.

Obtain MRM input data, origin-destination (O-D) data, and bottleneck data.

Develop workflow for data importing and conversion, disaggregate trips, improve the O-D matrix, specify traffic control, specify transportation system management and operations strategies and emerging technologies, and develop multiscenario datasets.

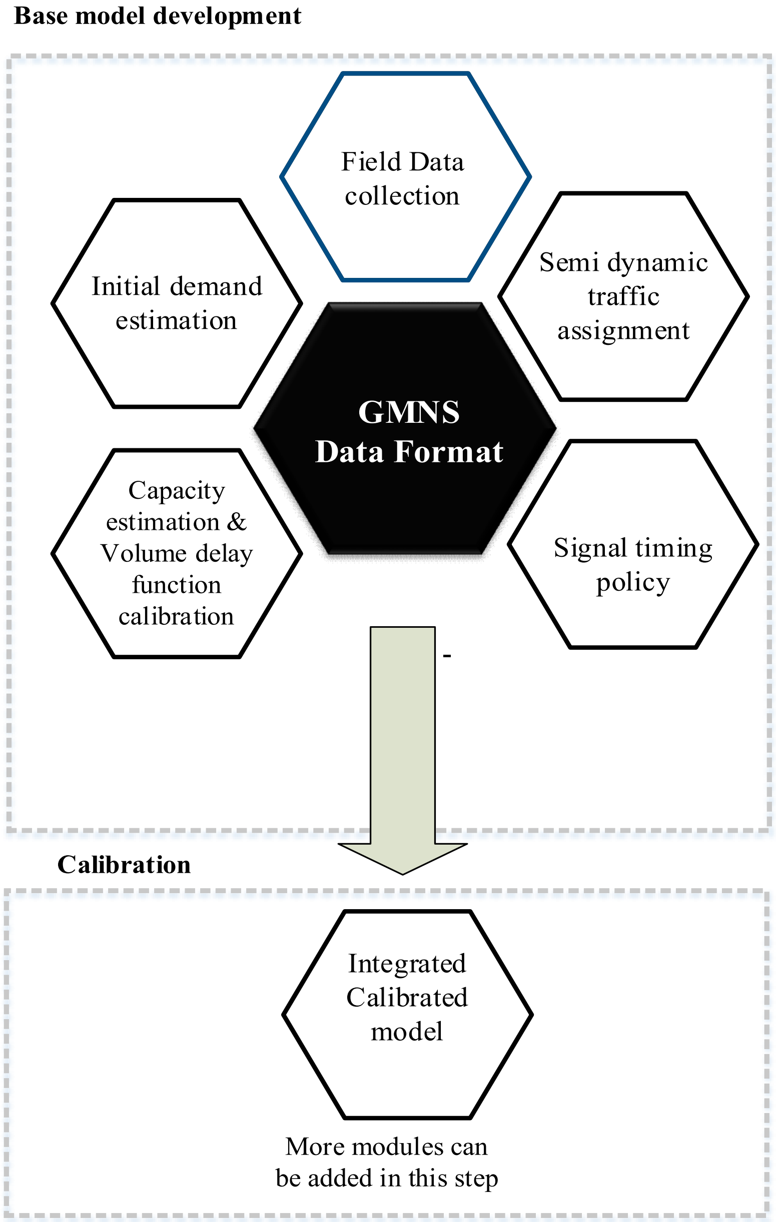
Calibrate demands, bottlenecks, and traffic flow models at all resolutions.
Assess convergence to ensure quality of traffic assignment.
Ensure consistency between ABM, DTA, and all resolutions.

Estimate future demands and signal timings.
Perform sensitivity analysis and account for model stochasticity.

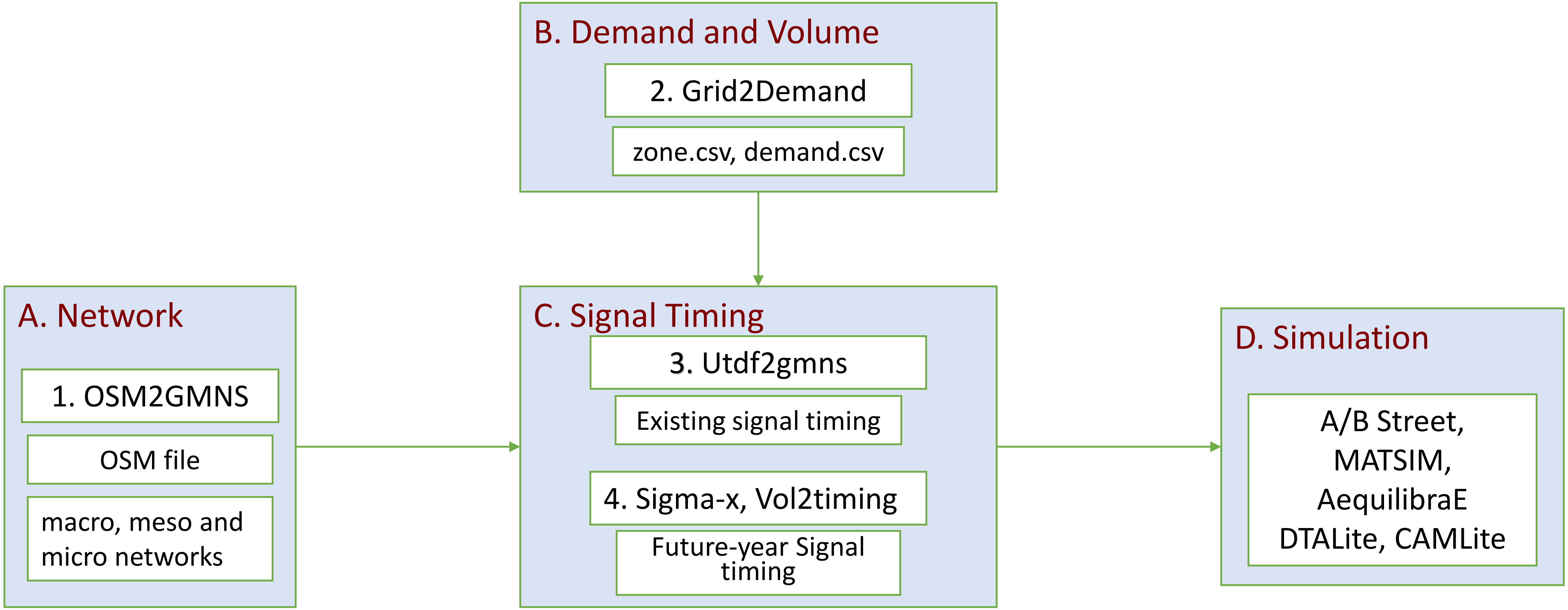
Source: FHWA. (Wunderlich et al. 2019)

MRM Benefits and Applications for Traffic Signal Simulation

- Use of different resolutions to supplement each other.
- Improved dynamic traffic assignment via mesoscopic modeling.
- Overall results more defensible.
- Meso-to-microscopic model consistency
- Unique insights into strategic driver behavior.
  - Important for emerging technologies and transportation system management and operations (TSMO) strategies.
- Robust analysis of subnetwork interaction.



It Takes a Community For Building Tools That Work with GMNS



- **Sigma-X: Excel-based computational engine for signalized intersections Vol2Timing as Python API**

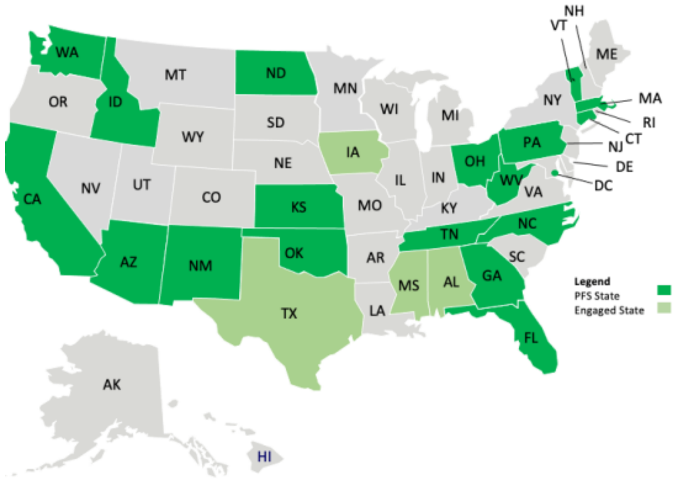
Excel->GMNS, Synchro UTDF -> GMNS

https://github.com/milan1981/Sigma-X

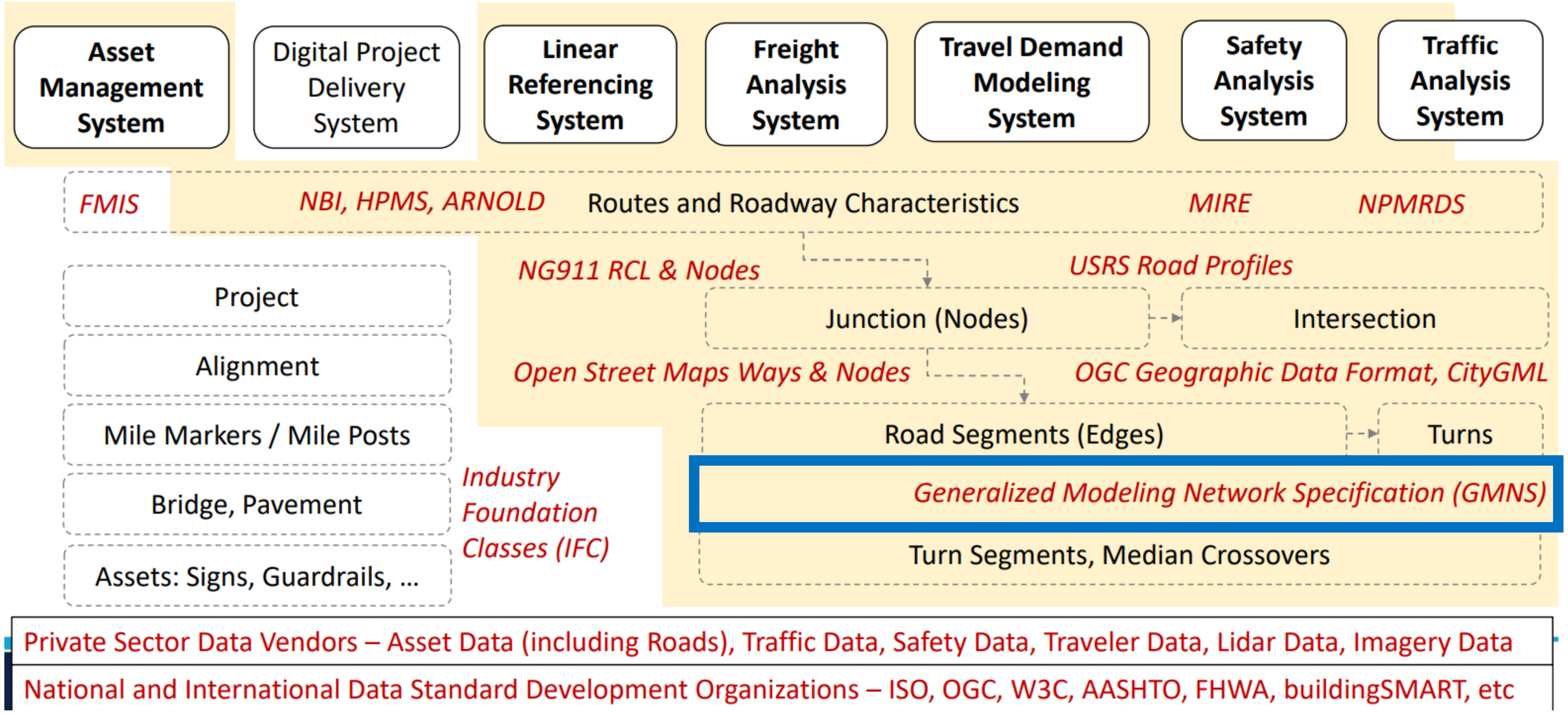
Contributor: Milan Zlatkovic

Broader Effort for Incorporating GMNS Into Applications Of Enterprise GIS In Transportation (AEGIST) Framework

Pooled Fund Study (PFS):
FHWA and States
Enterprise Data Management and Governance
Standards, Processes, Tools and Technology



www.gisintransportation.com



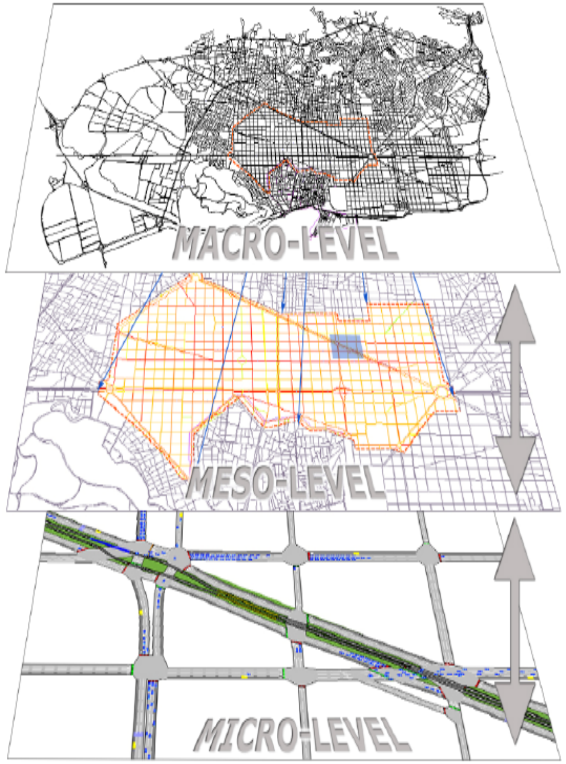
Source: AEGIST Pooled Fund Study (PFS)
FHWA Contact: Joe Hausman, Joseph.Hausman@dot.gov

Part III: Case Study of MRM

I. Open Data Specification

II. Open-Source Tools and MRM Community

III. Network Simulation Prospective and Case Studies

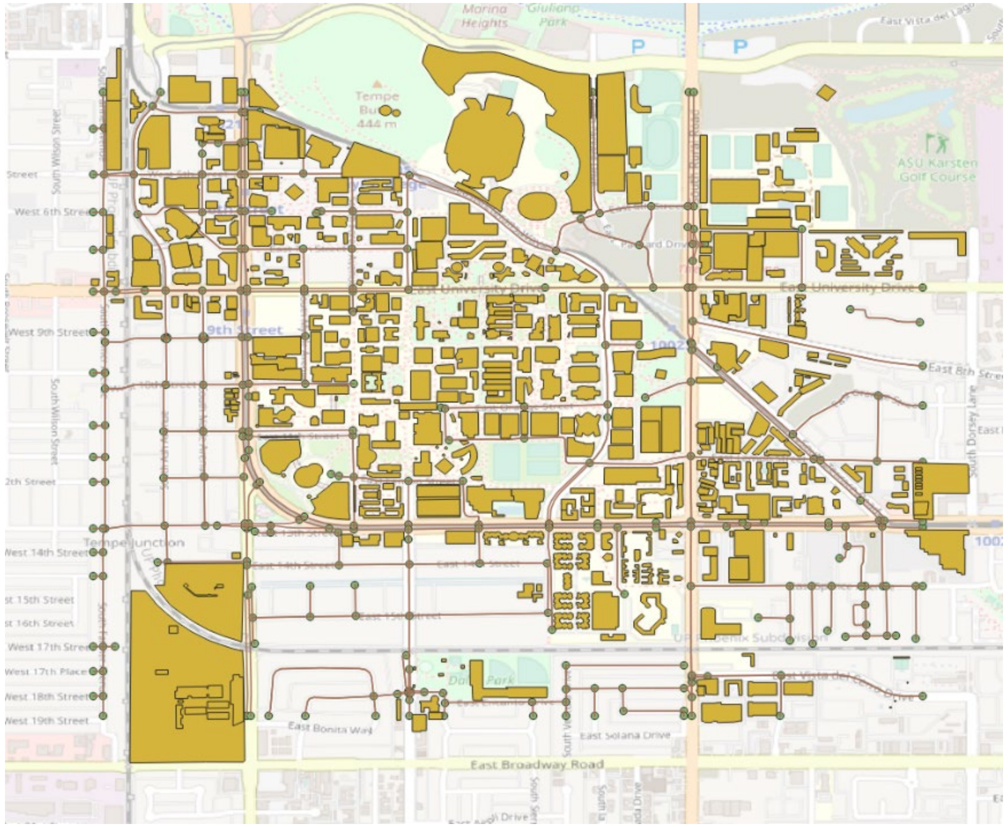


Streamlined workflow to be demonstrated using open-source tools

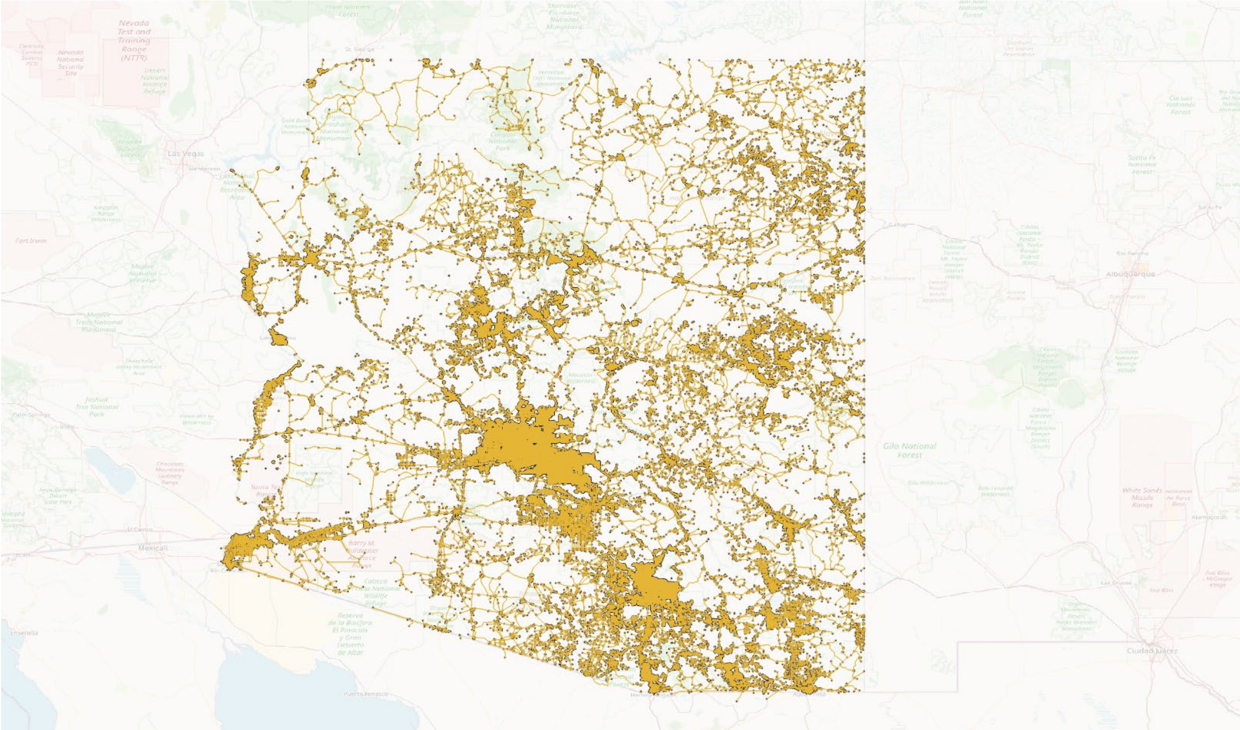
Step	Open-source tool(s)
1. Data download from online mapping database.	OSM
2. Convert OSM downloaded data into the GMNS format.	OSM2GMNS
3. Define zone-to-zone travel demand.	Grid2demand
4. Establish base-year traffic signal timing inventory	Utdf2gmns
5. Generate future-year traffic signal timing	Vol2timing, Sigma-X
6. Perform traffic simulation.	A/B Street, DTALite, CAMLite
7. Visualize results.	QGIS, NeXTA, A/B Street

A/B Street = traffic simulation game (GitHub 2021).
DTALite = queue-based mesoscopic traffic simulator (Zhou and Taylor 2014).
GMNS = General Modeling Network Specification (Zephyr Foundation 2021).
Grid2demand = data conversion tool (PyPI 2021). Utdf = data conversion tool (PyPI 2021).
NeXTA = Network EXplorer for Traffic Analysis (Zhou and Taylor 2014).
OSM = OpenStreetMap®.
OSM2GMNS = data conversion tool (PyPI 2021).
QGIS = geographic information system (qgis.org 2021).
Sigma-X = spreadsheet-based computational engine for signalized intersections (Zlatkovic 2021).
Vol2timing = GMNS-based signal timing generation tool for multi-resolution modeling (PyPI 2021).

1: Network from OSM: with POIs and Activity Locations



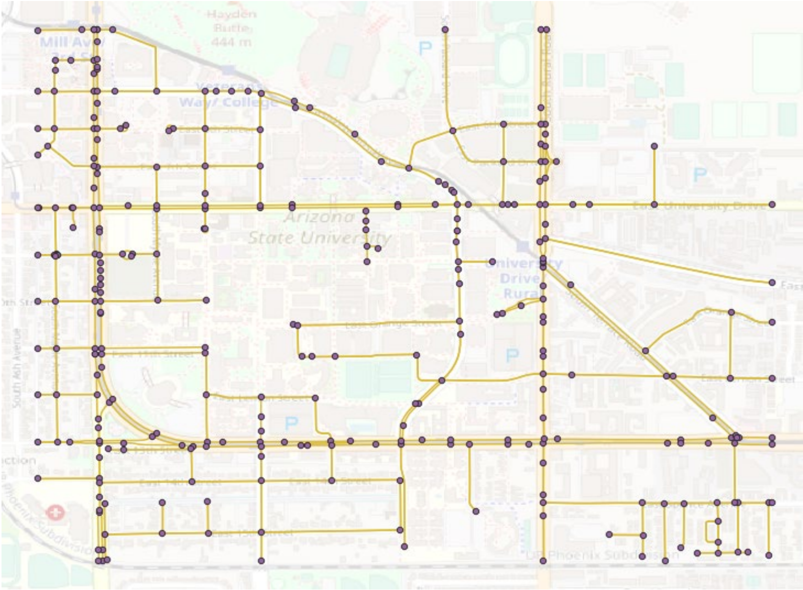
Network with POIs (Point of Interest)



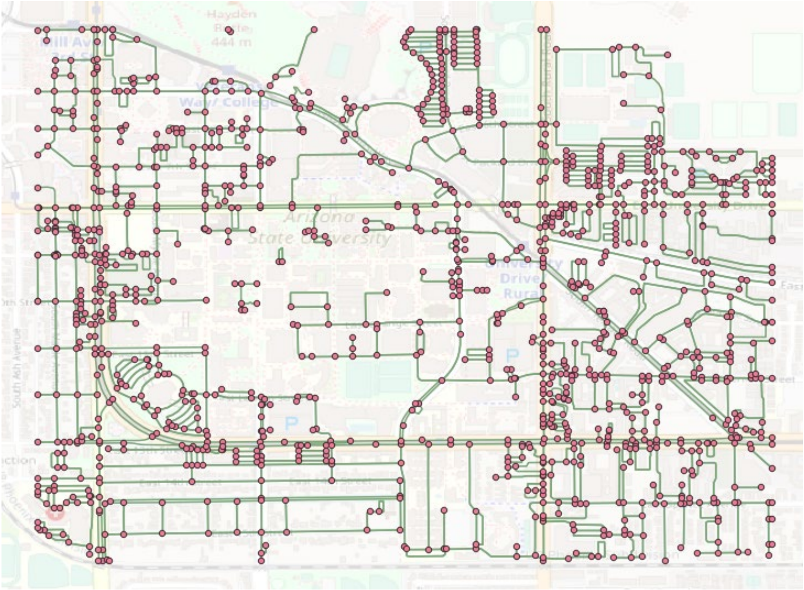
Arizona, U.S.

More sample networks. Contributor: Jiawei Lu at ASU
https://osm2gmns.readthedocs.io/en/latest/sample-net.html

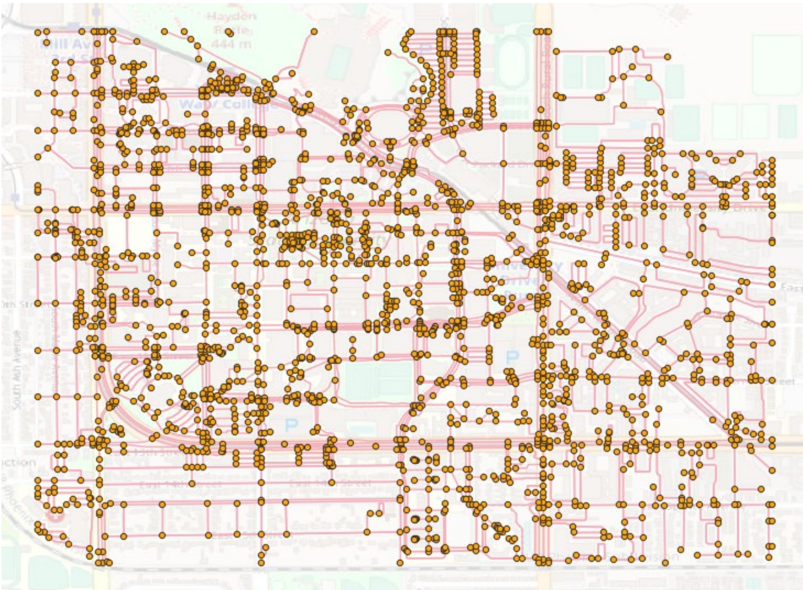
Multimodal Networks



Drivable network
320 nodes, 619 links



Bikeable network
1282 nodes, 2989 links



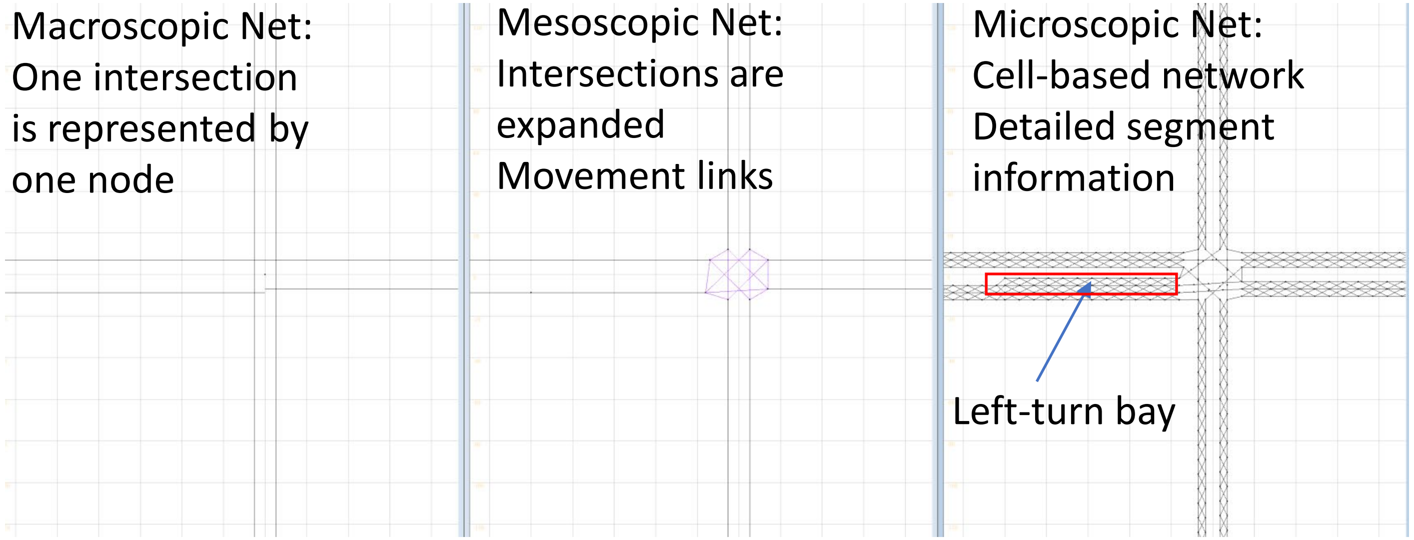
Walkable network
2426 nodes, 5019 links

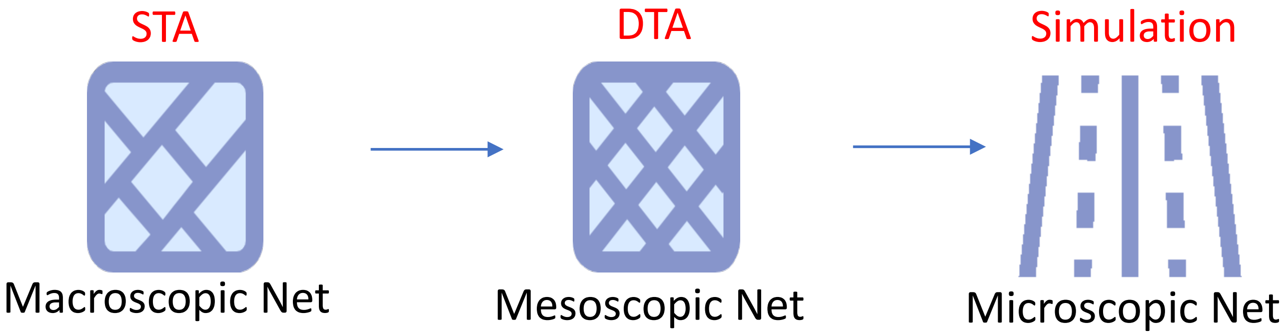
More sample networks

https://osm2gmns.readthedocs.io/en/latest/sample-net.html

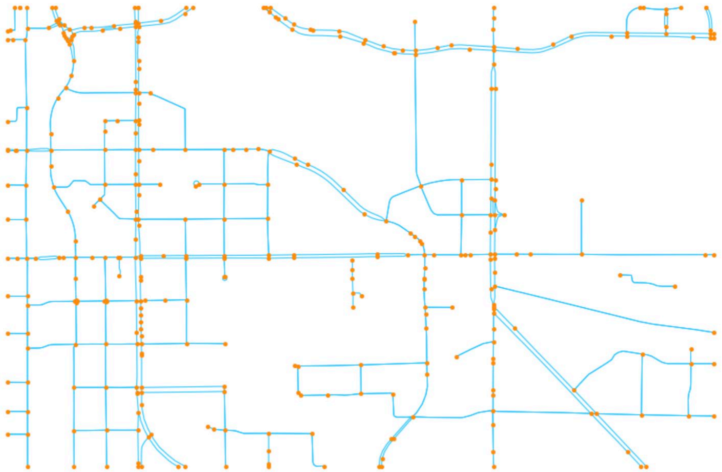
2. Convert OSM Downloaded Data into The GMNS Format (Multi-resolution Network)

- OSM2GMNS helps users automatically generate hybrid (macroscopic, mesoscopic and microscopic) transportation networks to accommodate different modelling needs.





- OSM2GMNS Tool, ASU Network



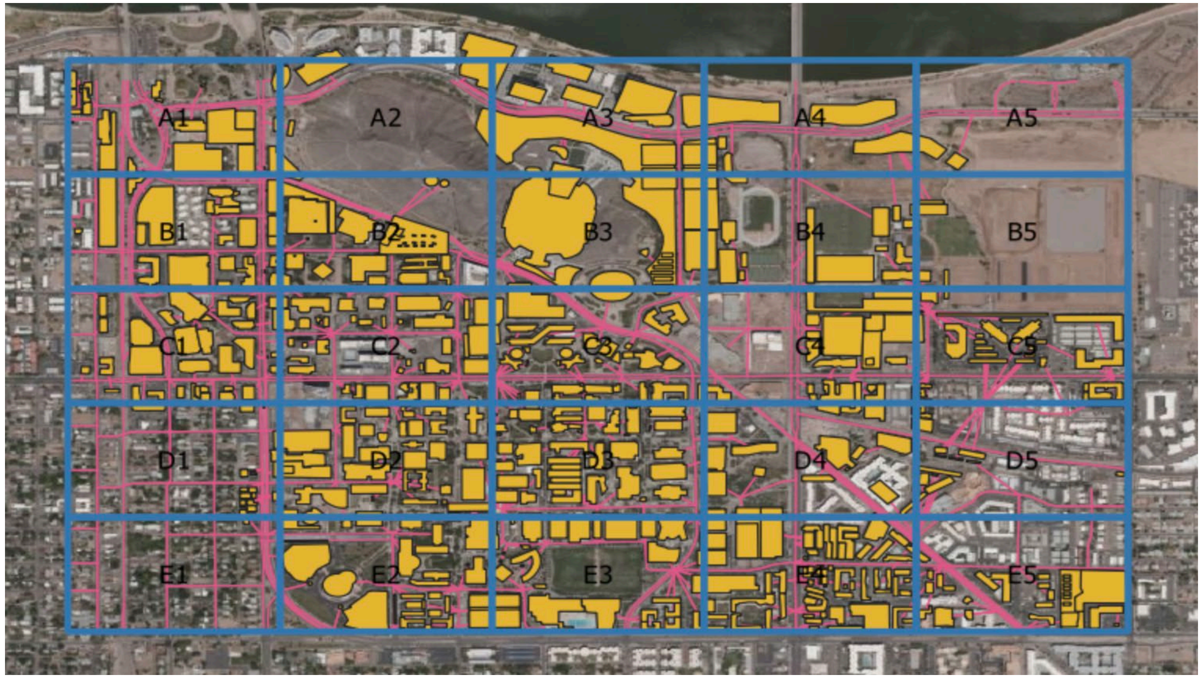
Macroscopic Network

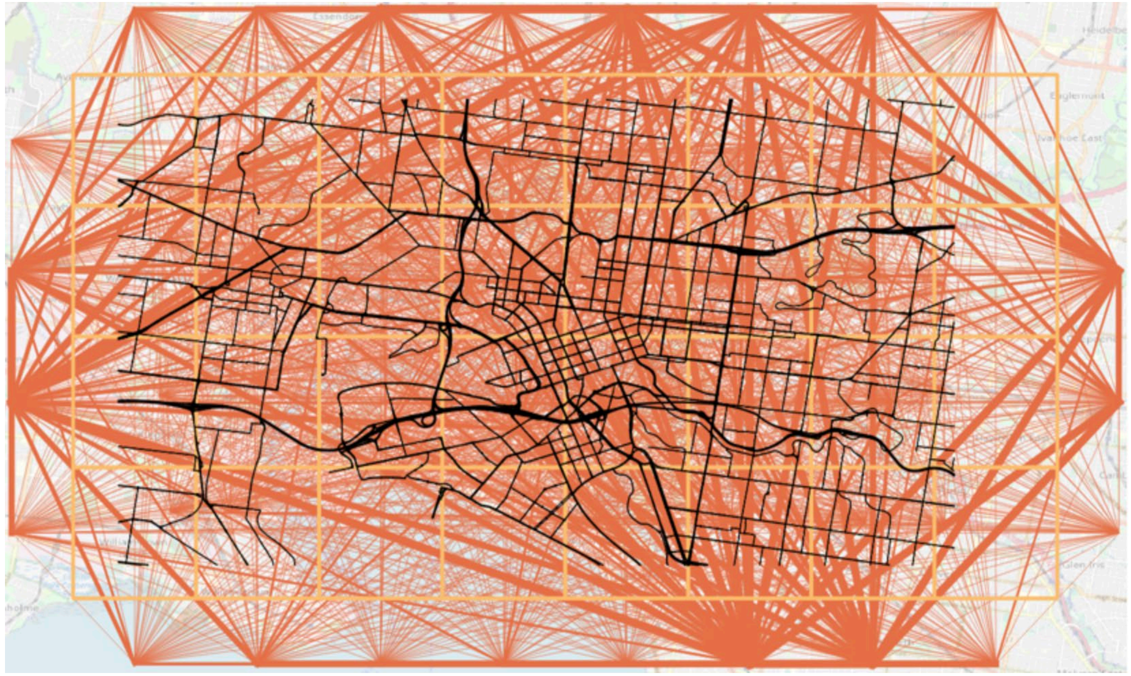
Microscopic, virtual-track based Network

2. Define Zone-to-zone Travel Demand

- Creating grid zones, based on POI data from OpenStreetMap to analyze the resident locations and other land use properties

- Quickly generate initial origin-destination transportation demand based on land use propertaies for engaging traffic simulation games such as A/B street and DTALite

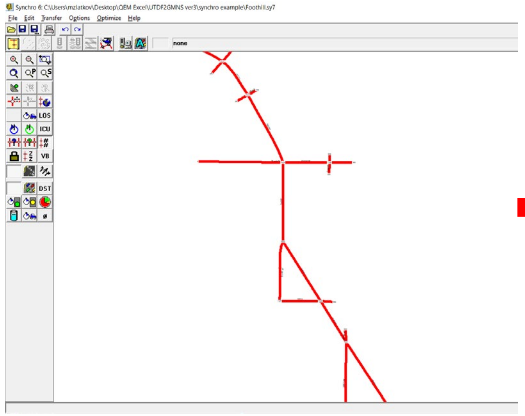




- Grid2demand Tool, ASU Network

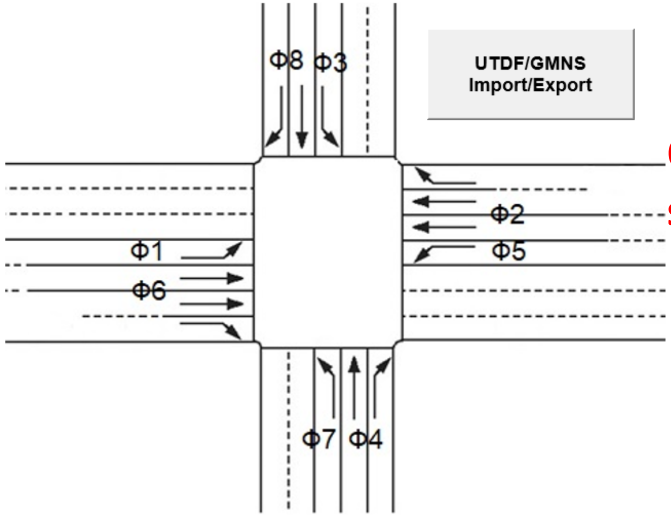
4. Establish Base-year Traffic Signal Timing Data Inventory

- **Meso and micro network**
- Convert UTDF file data into GMNS format (turn movement)
  UTDF Input-GMNS Output
- Turning Volume Inventory (e.g. from Gridsmart, MS2 traffic counts and from existing synchro files)



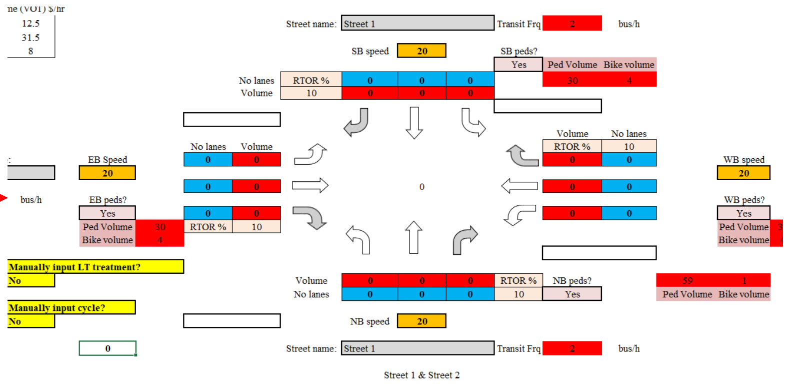
Synchro

UTDF



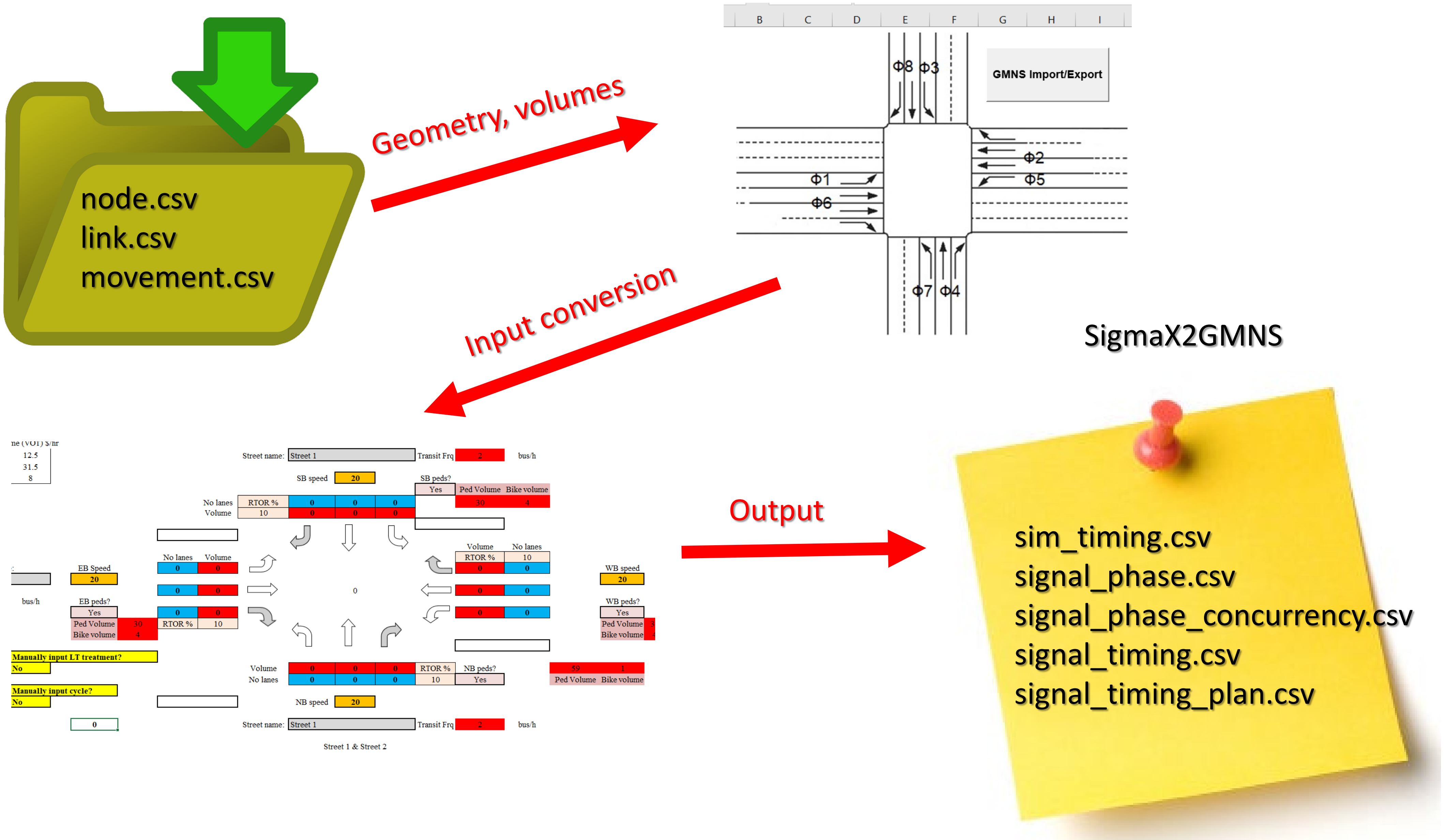
UTDF2GMNS

Output for simulation

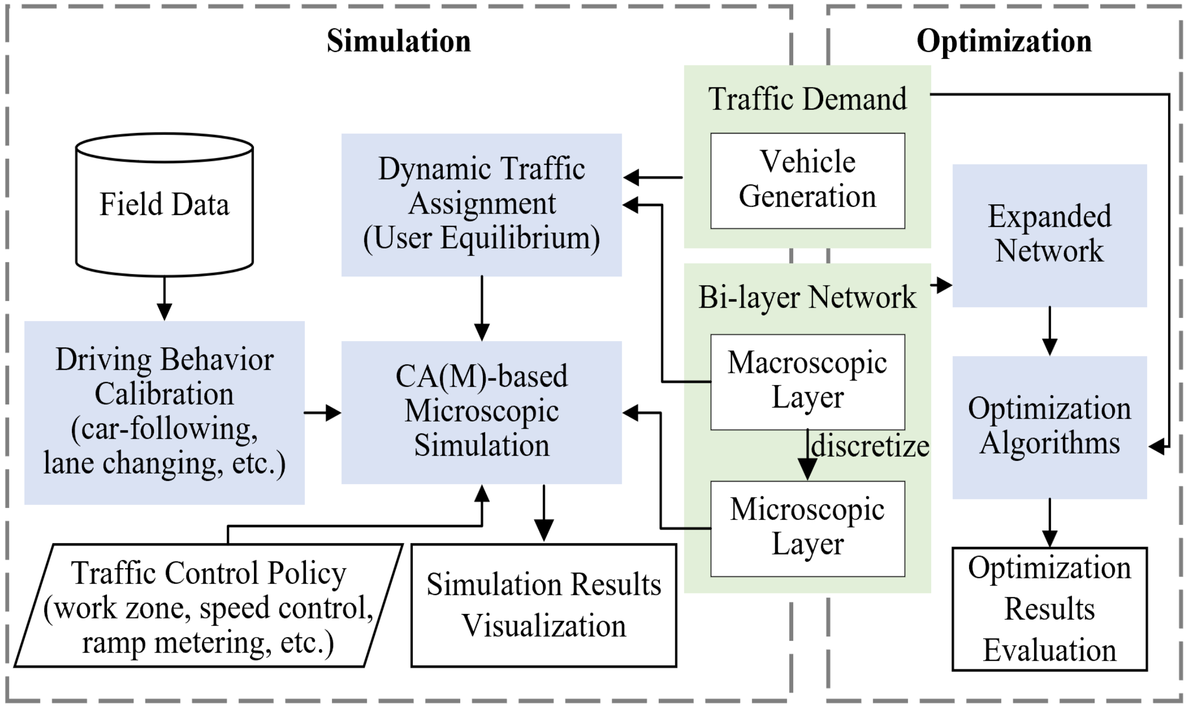


Movement_utdf.csv

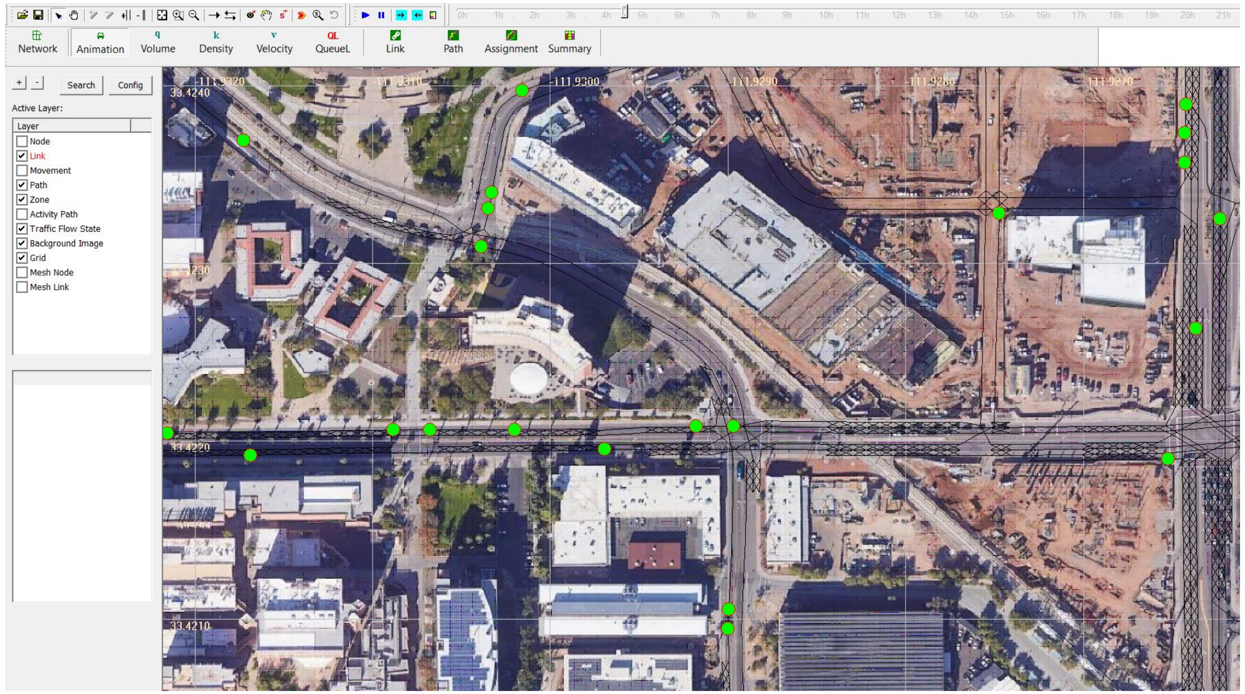
5. Establish Future-year Traffic Signal Timing Scenario



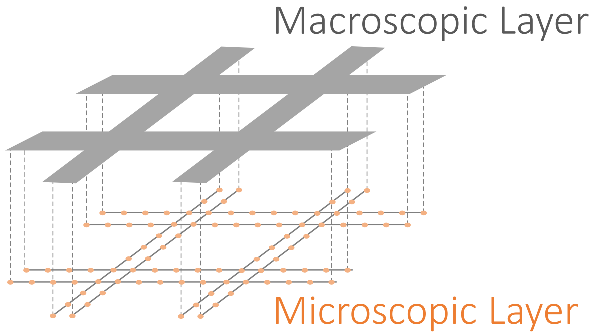
6. Perform Traffic Simulation,
Open-source CAMLite: Virtual-Track-based Modeling Framework:

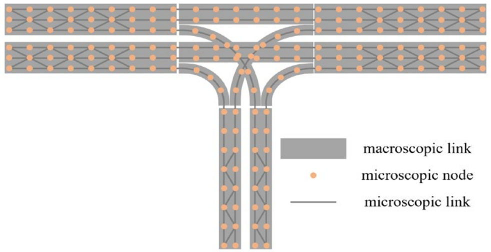


System architecture overview



Simulation Results from CAMLite





- CAMLite Tool, ASU Network

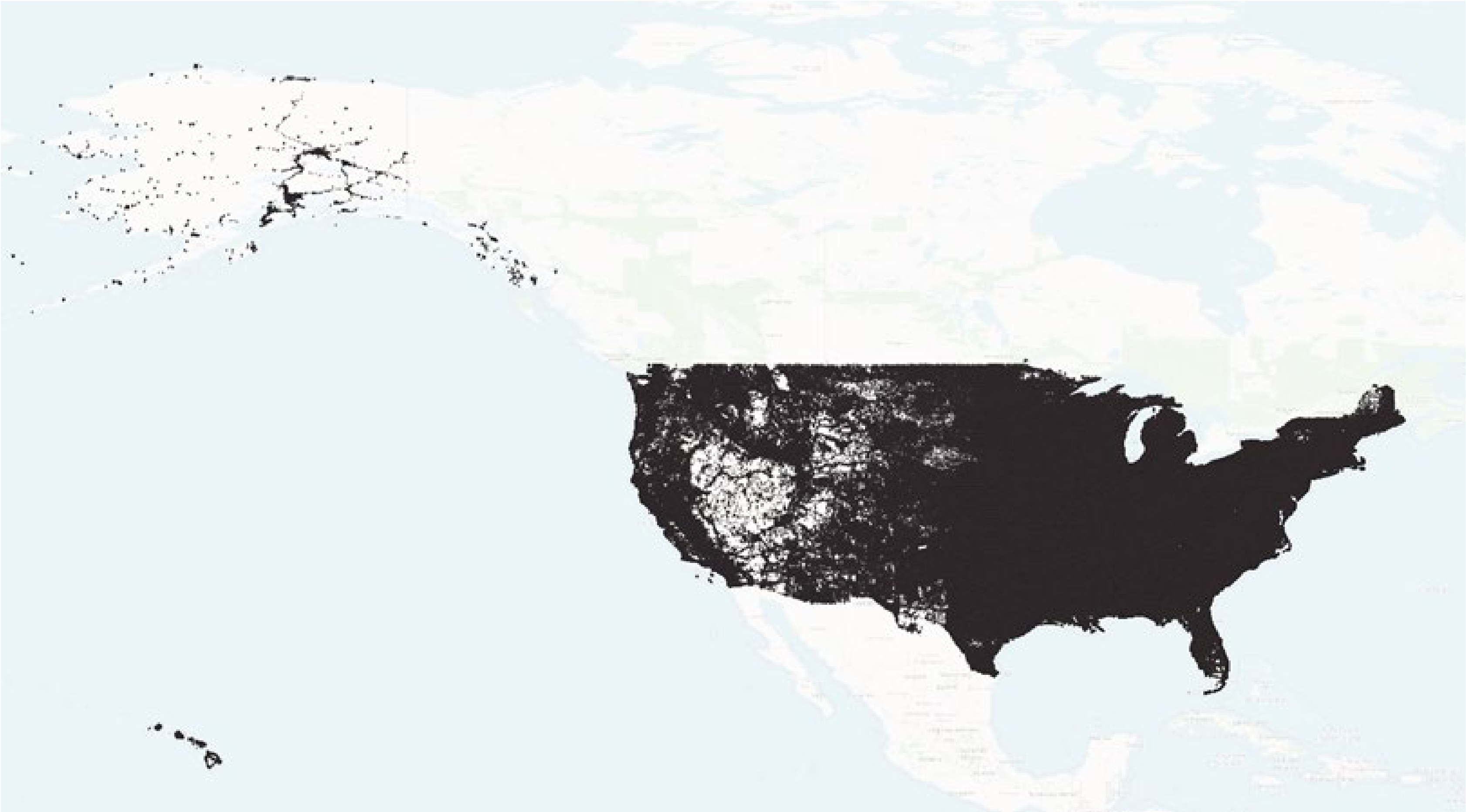
7. Visualization through AB Street

https://github.com/a-b-street/abstreet

Source: https://github.com/a-b-street/abstreet, Special thanks to Dustin Carlino (dabreegster@gmail.com)

Challenges Addressed	Opportunities for state DOTs and MPO stakeholders
1. Map intersections to OpenStreetMap GIS layer	- Utilize machine learning algorithms to automatically map intersections to OSM, instead manually.
2. Manage # of links and counts for each movement in a unified data base	- Develop a data validation process to ensure accuracy of counts and links.
3. Integrate Synchro files in one GIS database, through UTDF reading interface	- Use a transportation management software to generate UTDF files automatically.
4. . Automate the signal timing update process using HCM methodology	- Implement a standard signal timing update process to ensure consistency and efficiency.
5. Future data connection with TranSync for corridor and city levels	- Define the data requirements for corridor-based trajectory-based performance assessment.

To bridge connections between open data, traffic engineers, and city planners, we need the collective effort of a community coming together to build the necessary tools.



Using ASU research computing facilities, we are able to produce the entire U.S. driving network from OpenStreetMap with 20 million nodes.

https://github.com/asu-trans-ai-lab/asu-trans-ai-lab.github.io

PyPi link	https://pypi.org/project/osm2gmns/
Total downloads	55,000
Total downloads - 30 days	2,747
Total downloads - 7 days	538

Integrating ATC Controller Software into the SUMO Microsimulation Traffic Control Interface

Dustin DeVoe

VP, Embedded ITS Product Management

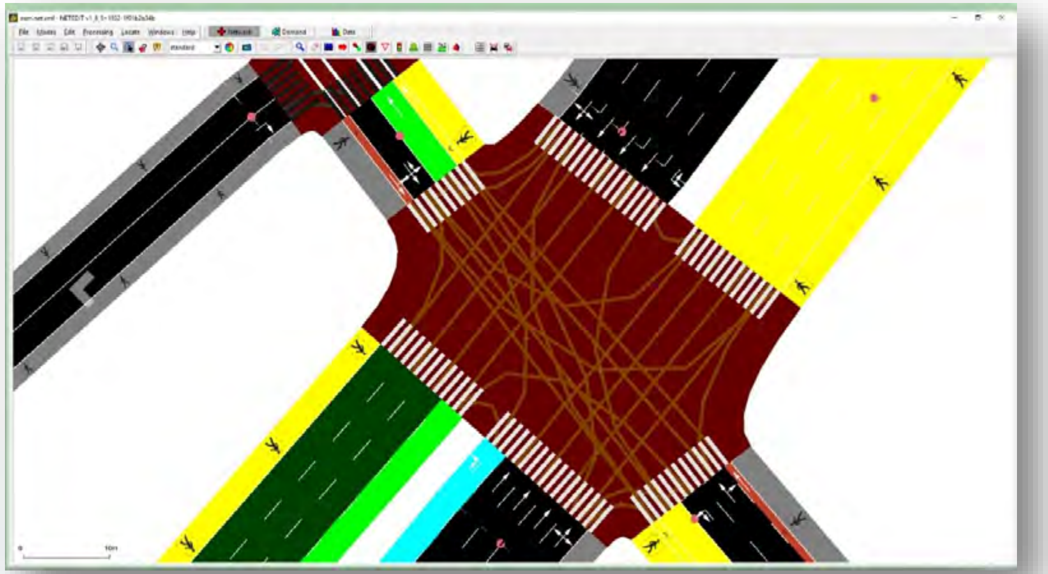
ECONOLITE

Saving Lives Through Improved Mobility

Background

What is SUMO and the Econolite ATC software?

- SUMO (**S**imulation of **U**rban **MO**bility), an open source, highly portable, microscopic and continuous multi-modal traffic simulation package designed to handle large networks.
- Used in FHWA VOICES and Carma Streets
- https://www.eclipse.org/sumo/



EOS
Controller Software

- Latest ATC Controller software
- Familiar ASC3 features
- Deployed in over 12,000 Intersections
- Coordination Visualization
- Adaptive splits and COS commands
- NTCIP 1202v3
- NTCIP 1211 Priority Request Server
- SAE SPaT and Map broadcast
- 32 Phase, 32 Overlaps, 128 Detectors
- ATSPM

Integrating into SUMO

How can they work together?

Connecting the Time Domain and IO

Getting Started

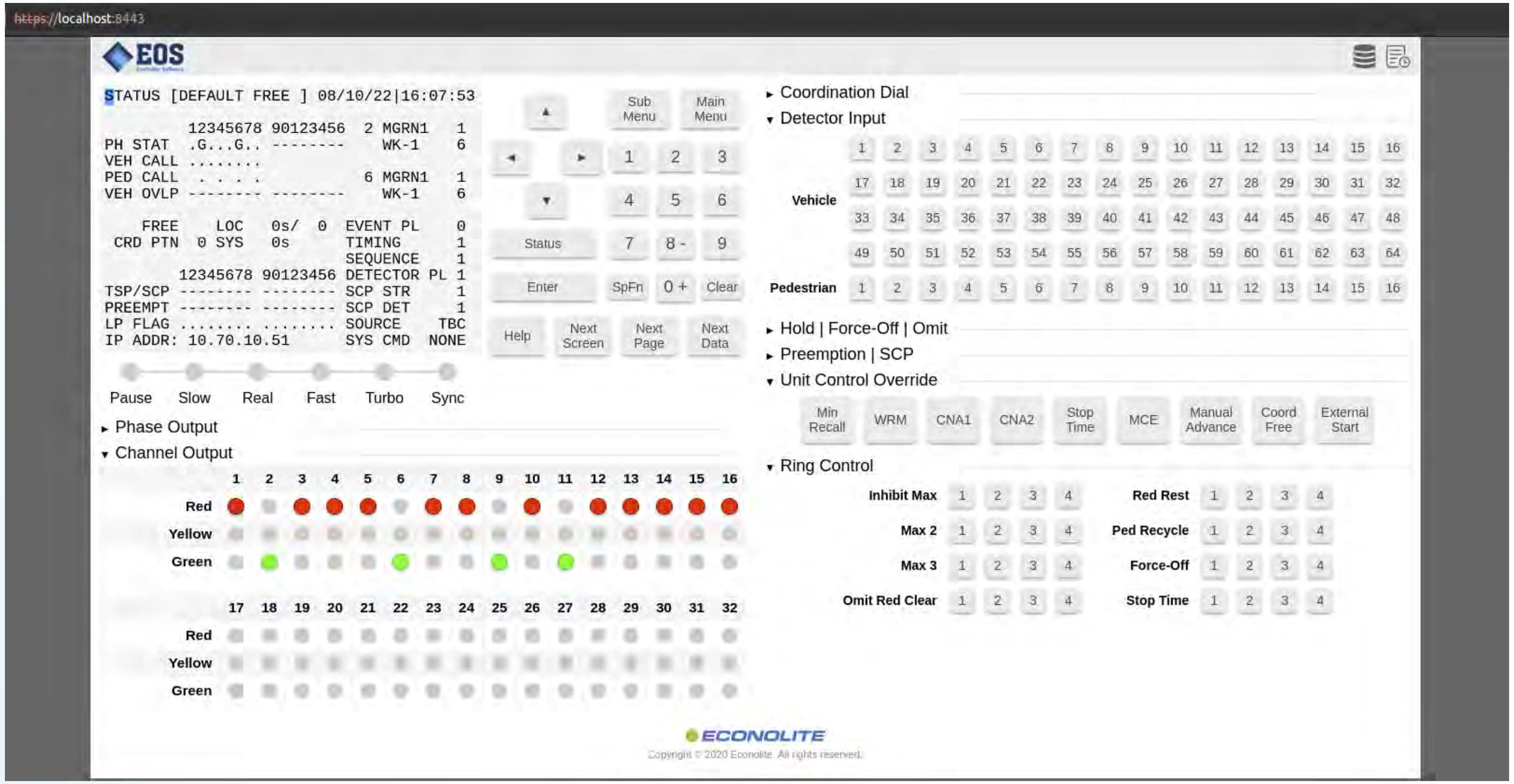
- **PyEOS** is an open-source Python library created by Econolite to spawn virtual EOS Traffic Controllers
- **Installation Requirements**:
  - Linux operating system – ideally Ubuntu 20.04 LTS or higher
  - 32-bit GCC/G++ libraries
  - Python 3.8 or higher
  - Pip package manager
    - **pip3 install pyeos**
    - **pip3 install traci**
  - EOS Traffic application version 3.2.18 or higher

Controller Terminology

- **CIB** = An acronym for Controller Input Buffer.
  - This represents the generic set of inputs that EOS controllers support.
  - For SUMO usage, the most used are vehicle and pedestrian detectors.
- **COB** = An acronym for Controller Output Buffer.
  - This represents the generic set of outputs that EOS controllers support.
  - For SUMO usage, the most used are vehicle, overlap, and pedestrian colors.

The Controller

- Controller class holds meta related features
- Process controls
  - run()
  - stop()
  - is_running()
- Open the web front panel of EOS with *watch()*




```
with virtual_factory(eos_dir) as eos_factory:
    with eos_factory.run_controller() as eos:
        assert eos.is_running()
        eos.watch()
```

Controller Options

- An input for the Controller class that preconfigures EOS
- Controller configuration object is available for spawning EOS under specific conditions such as:
  - EOS Configuration
  - Start Time *(using Python datetime object)*
  - Designated Port

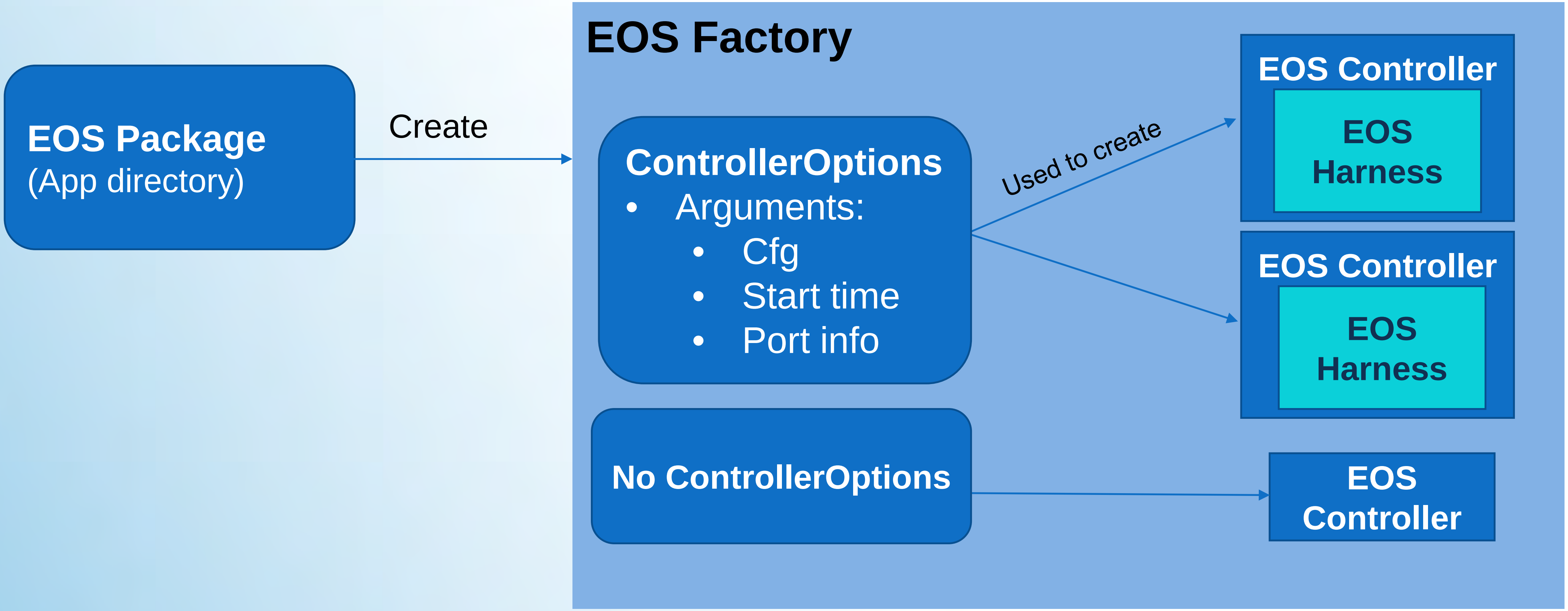
```
start_time = datetime.datetime.now() - datetime.timedelta(days=1)
options = ControllerOptions(eos_cfg_path, start_time=start_time, https_port=9001)

with virtual_factory(eos_dir) as eos_factory:
    with eos_factory.run_controller(options) as eos:
        # . . . code
```

The EOS Harness

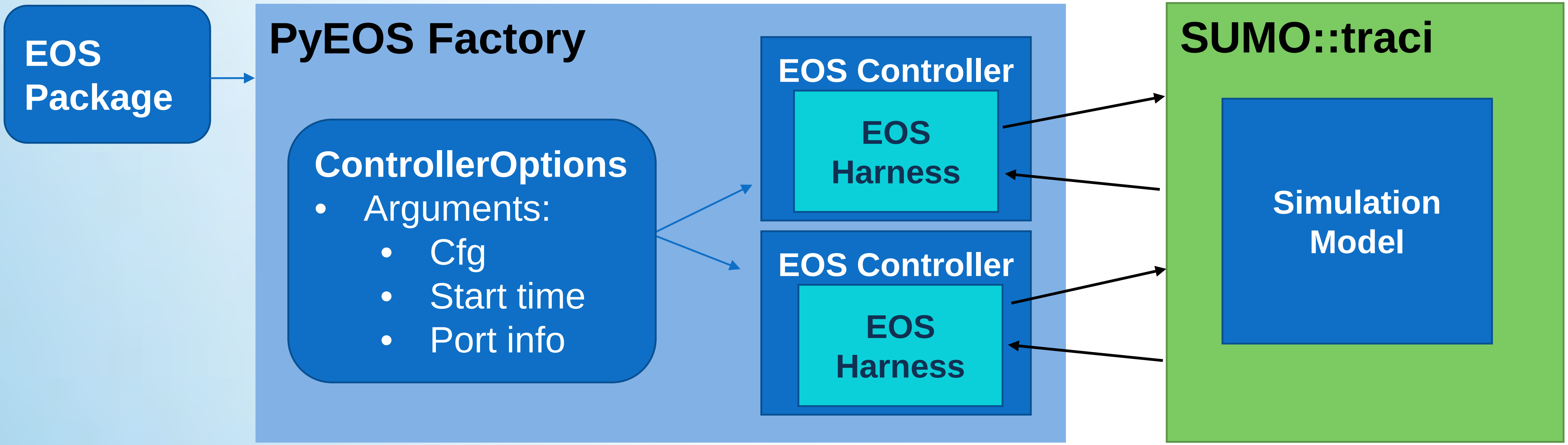
- A "Utility belt" for EOS virtual controller
- Manages I/O for the following:
  - CIB / COB Communication
  - Clock controls
- Use the controller's *harness()* method to create:

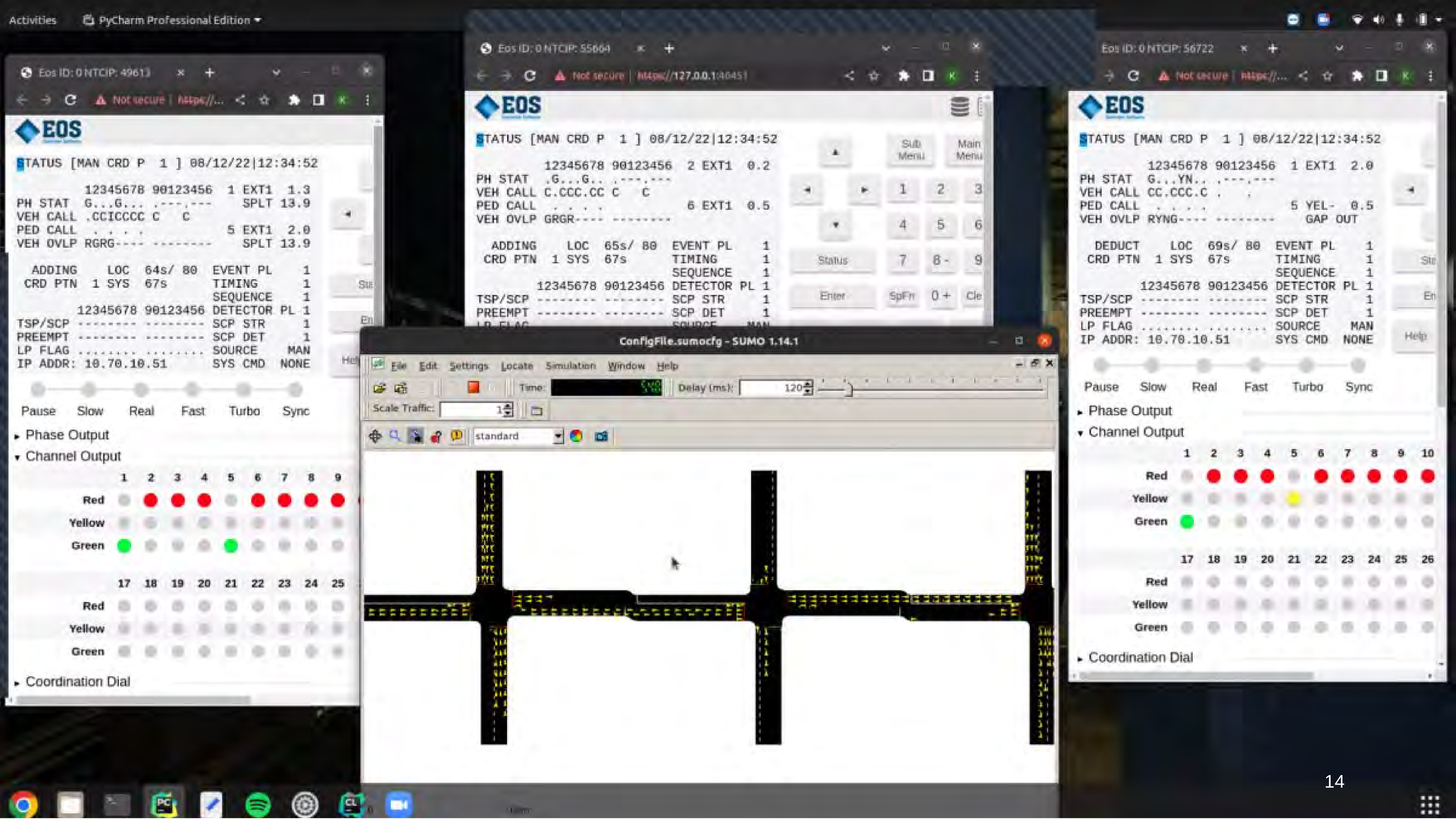
```
with virtual_factory(eos_dir) as eos_factory:
    with eos_factory.run_controller() as eos:
        with controller.harness() as eos_harness:
            # . . . code
```

PyEOS Factory Model



PyEOS Demo with SUMO





Activities
PyCharm Professional Edition
Eos ID: 0 NTCIP: 49613
Eos ID: 0 NTCIP: 55664
Eos ID: 0 NTCIP: 56722
Not secure
EOS
STATUS [MAN CRD P 1] 08/12/22|12:34:52
12345678 90123456 1 EXT1 1.3
PH STAT G...G... .---.--- SPLT 13.9
VEH CALL .CCICCCC C C
PED CALL 5 EXT1 2.0
VEH OVLP RGRG---- -------- SPLT 13.9
ADDING LOC 64s/ 80 EVENT PL 1
CRD PTN 1 SYS 67s TIMING 1
SEQUENCE 1
12345678 90123456 DETECTOR PL 1
TSP/SCP -------- -------- SCP STR 1
PREEMPT -------- -------- SCP DET 1
LP FLAG .... SOURCE MAN
IP ADDR: 10.70.10.51 SYS CMD NONE
Pause
Slow
Real
Fast
Turbo
Sync
Phase Output
Channel Output
Red
Yellow
Green
Coordination Dial
STATUS [MAN CRD P 1] 08/12/22|12:34:52
12345678 90123456 2 EXT1 0.2
PH STAT .G...G.. .---.---
VEH CALL C.CCC.CC C C
PED CALL 6 EXT1 0.5
VEH OVLP GRGR---- --------
ADDING LOC 65s/ 80 EVENT PL 1
CRD PTN 1 SYS 67s TIMING 1
SEQUENCE 1
12345678 90123456 DETECTOR PL 1
TSP/SCP -------- -------- SCP STR 1
PREEMPT -------- -------- SCP DET 1
Sub Menu
Main Menu
Status
Enter
SpFn
STATUS [MAN CRD P 1] 08/12/22|12:34:52
12345678 90123456 1 EXT1 2.0
PH STAT G...YN.. .---.---
VEH CALL CC.CCC.C . .
PED CALL . . . . . 5 YEL- 0.5
VEH OVLP RYNG---- -------- GAP OUT
DEDUCT LOC 69s/ 80 EVENT PL 1
CRD PTN 1 SYS 67s TIMING 1
SEQUENCE 1
12345678 90123456 DETECTOR PL 1
TSP/SCP -------- -------- SCP STR 1
PREEMPT -------- -------- SCP DET 1
LP FLAG .... SOURCE MAN
IP ADDR: 10.70.10.51 SYS CMD NONE
Help
ConfigFile.sumocfg - SUMO 1.14.1
File
Edit
Settings
Locate
Simulation
Window
Help
Time:
Delay (ms):
120
Scale Traffic:
standard

Available Soon

- Continue Support of Active Project
  - CARMA Streets (soon)
- Finish development of Commercial Usage license mechanism in First Half of 2023
- Share Source and Binaries Through GitHub

Thanks!

Dustin DeVoe

ddevoe@econolite.com

A multi-resolution Simulation Platform for Perception-Based Traffic Signal Control

TRB Webinar: State-of-the-Art Traffic Signal Simulation Tools and Platforms

Dr. Yiheng Feng

Assistant Professor

Connected Automated and Resilient Transportation (CART) Lab

Lyles School of Civil Engineering, Purdue University

feng333@purdue.edu

PURDUE UNIVERSITY | Lyles School of Civil Engineering

Technical Path of the Existing Traffic Signal Control

Infrastructure-based detection

- High installation and maintenance cost
- Fixed location sensors
- Limited in congestion management
- Limited in performance measures

Single Intersection

Communication

Coordination

Detection

Vehicle Actuated

More Detection

Adaptive

Vehicle Trajectories

CV Based Signal Control

Perception Data

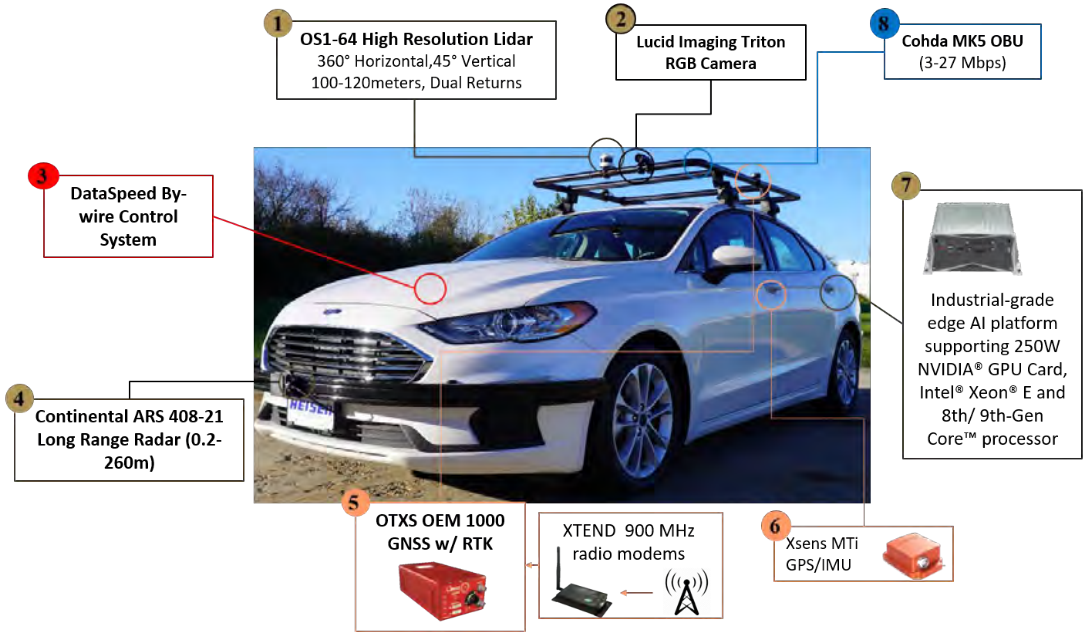
CAV Based Signal Control

Trajectory-based detection

- Temporal-spatial continuous data
- Crowdsourcing and low-cost
- Require certain penetration rate
- Inaccurate positioning

Cooperative Perception Environment

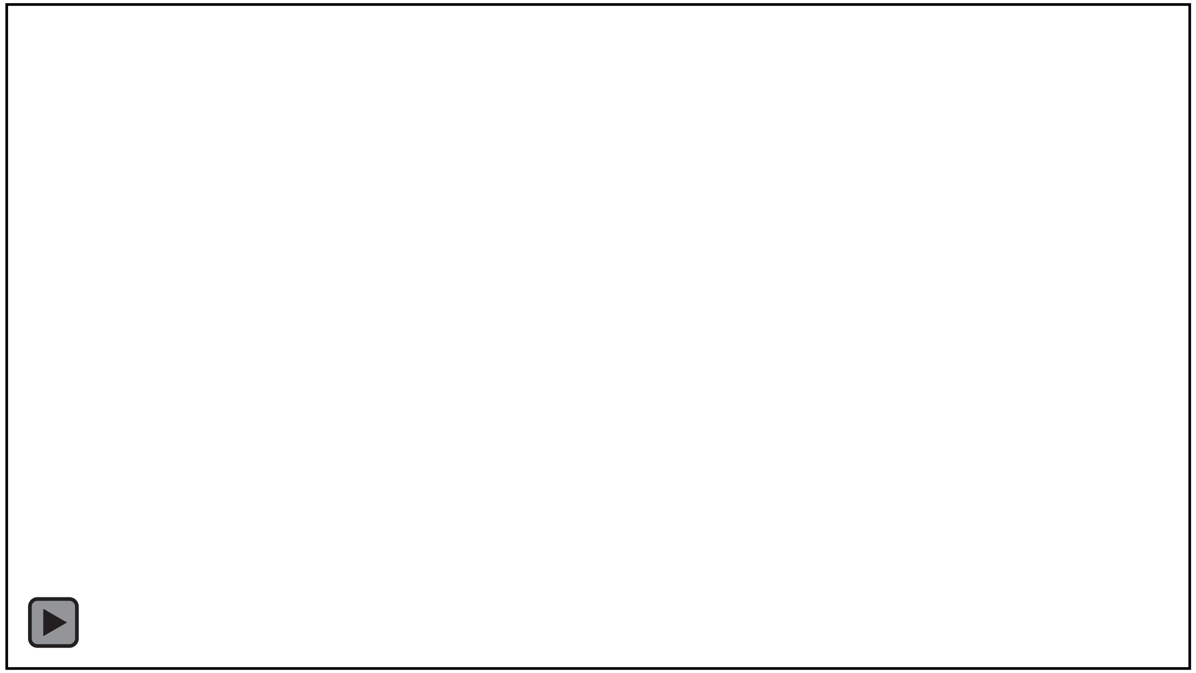
- Perception sensors (Lidar, Radar, Camera) on vehicles and Infrastructure



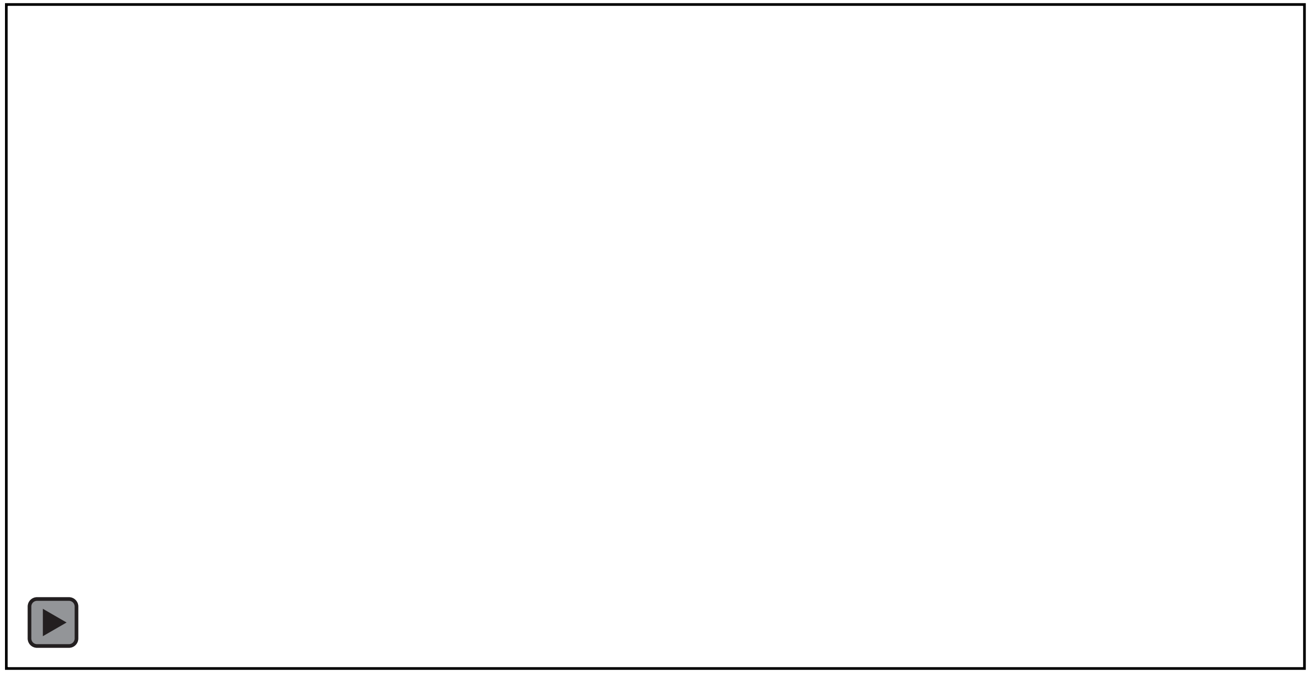
Level 4 AV at Purdue CART Lab

Equipped infrastructure at State and Ellsworth roundabout, Ann Arbor, Michigan

Cooperative Perception Environment



Vehicle Detection and Tracking from the CAV

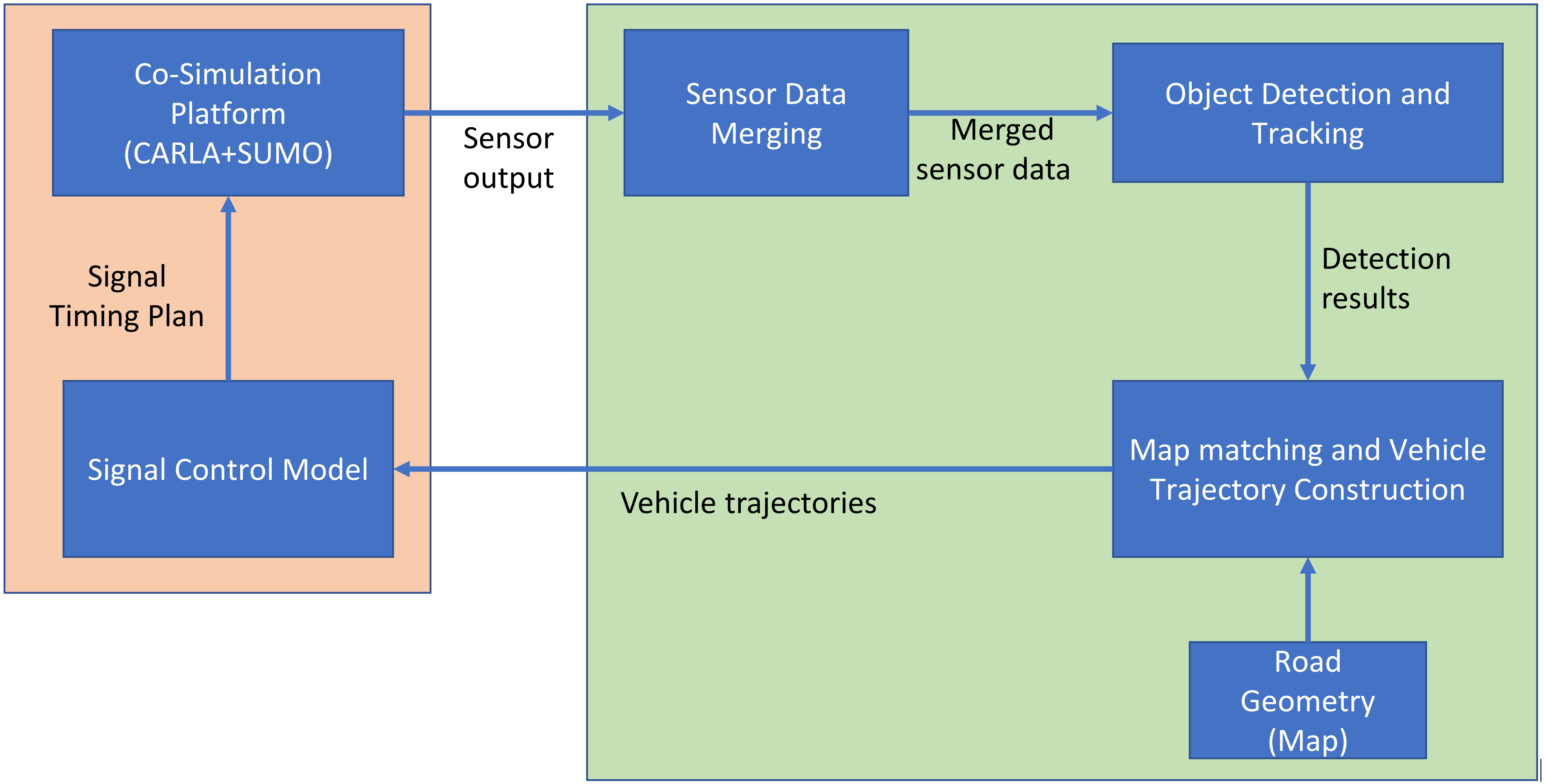


Vehicle Detection and Tracking from Infrastructure Cameras (Credit: Michigan Traffic Lab)

Cooperative Perception for Traffic Signal Control

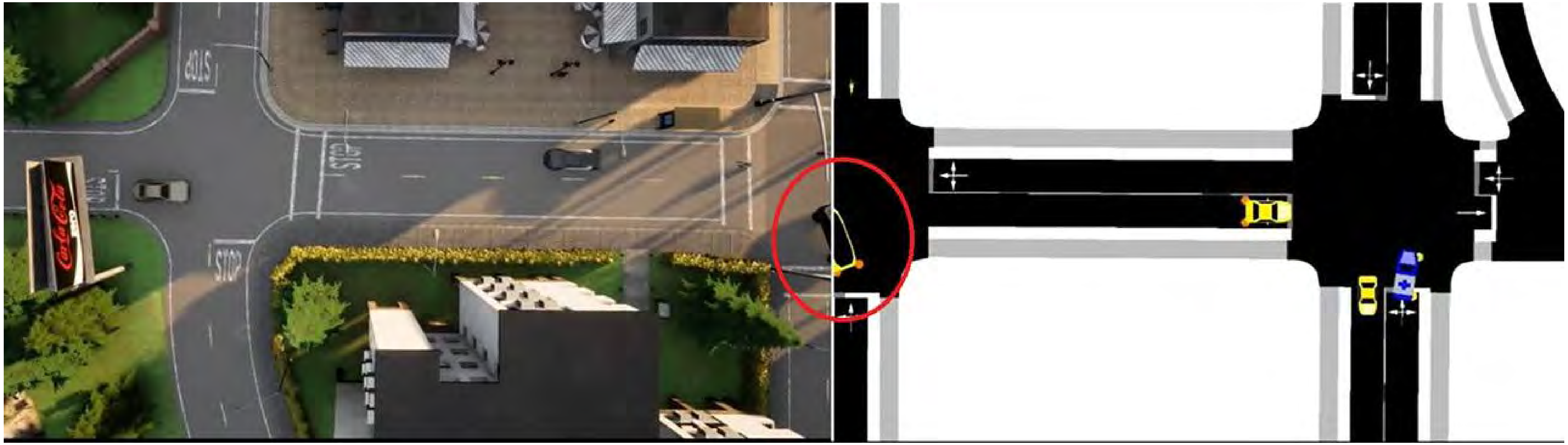
- A new data source (video and/or point cloud)
  - Richer information than both detectors and CV trajectories
  - High fidelity and high accuracy
  - Only need very low penetration rate of CAVs
- Sensor fusion from multiple sources
  - Raw data level: directly fuse image or point cloud
  - Future level: fuse features extracted from the raw data
  - Decision level: fuse detection results (e.g., bounding boxes)
- Develop a simulation environment to test signal control algorithms with cooperative perception

Simulation System Design



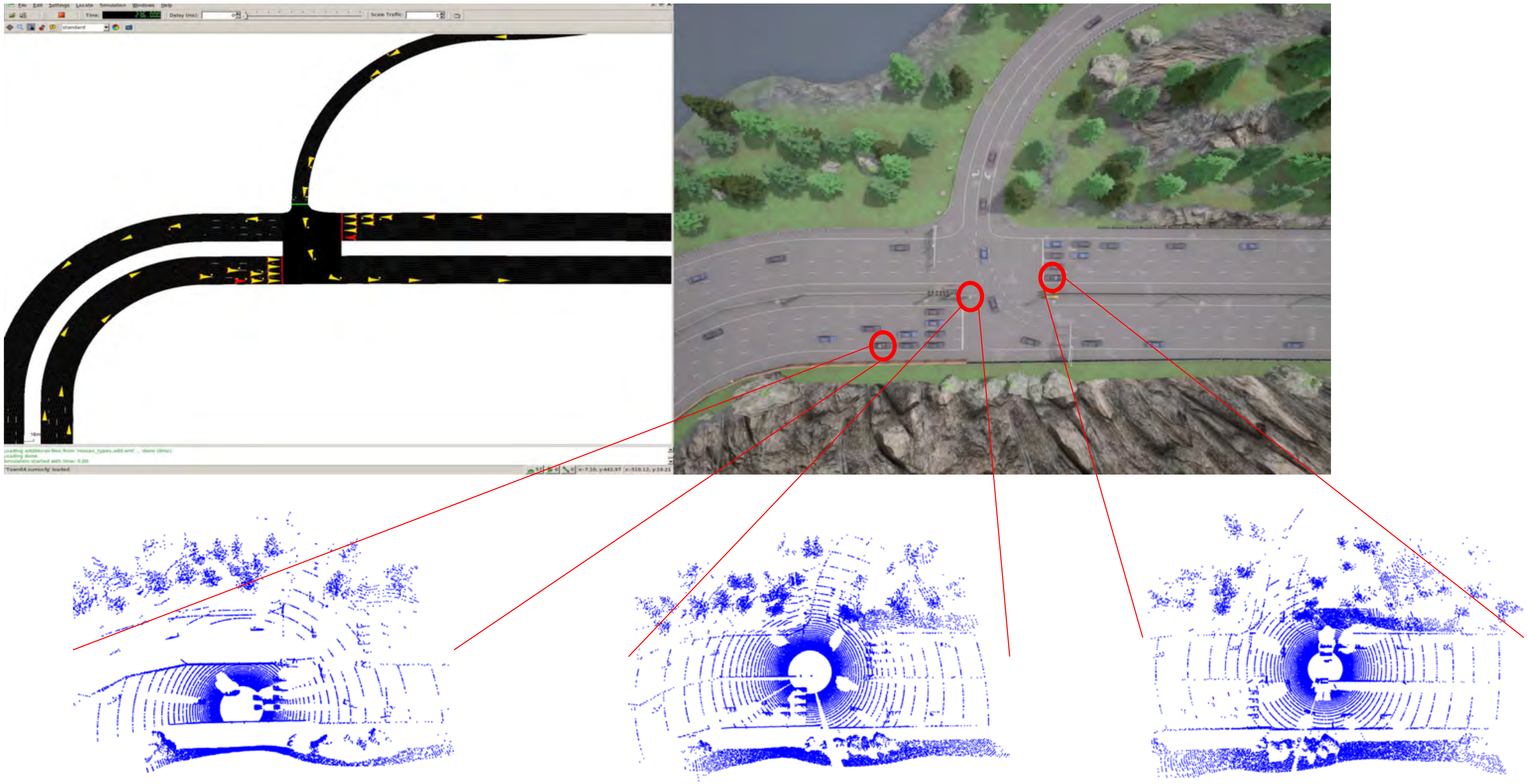
Co-Simulation Platform

- Cooperative Driving Automation (CDA) multi-resolution simulation platform
  - Developed by the CDA program at FHWA
  - Integrated with CARMA, CARLA, SUMO, MOASIC, and NS3



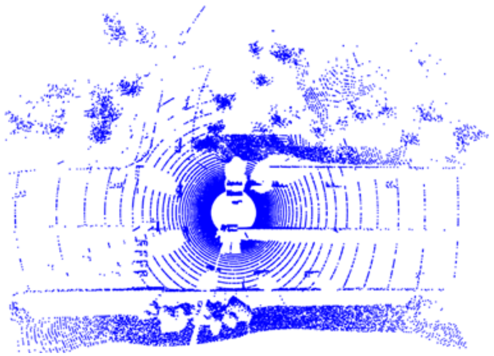
(Source: https://usdot-carma.atlassian.net/wiki/spaces/CRMSIM/overview)

Cooperative Perception from Multiple Lidars

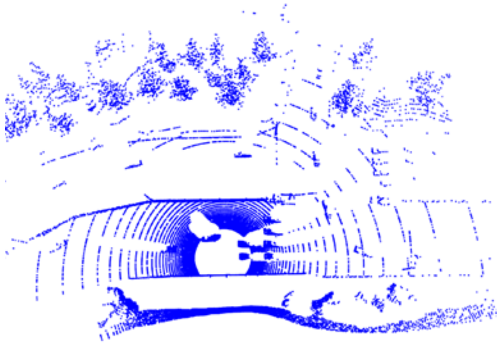


CAV 2 | Infrastructure Lidar | CAV 1

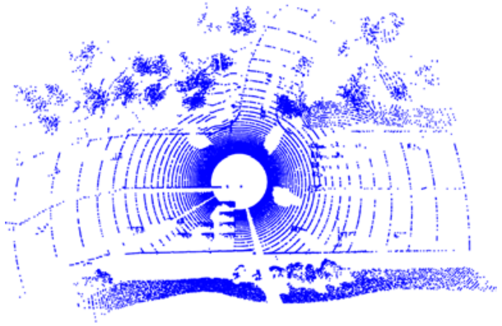
Cooperative Perception from Multiple Lidars



CAV 1



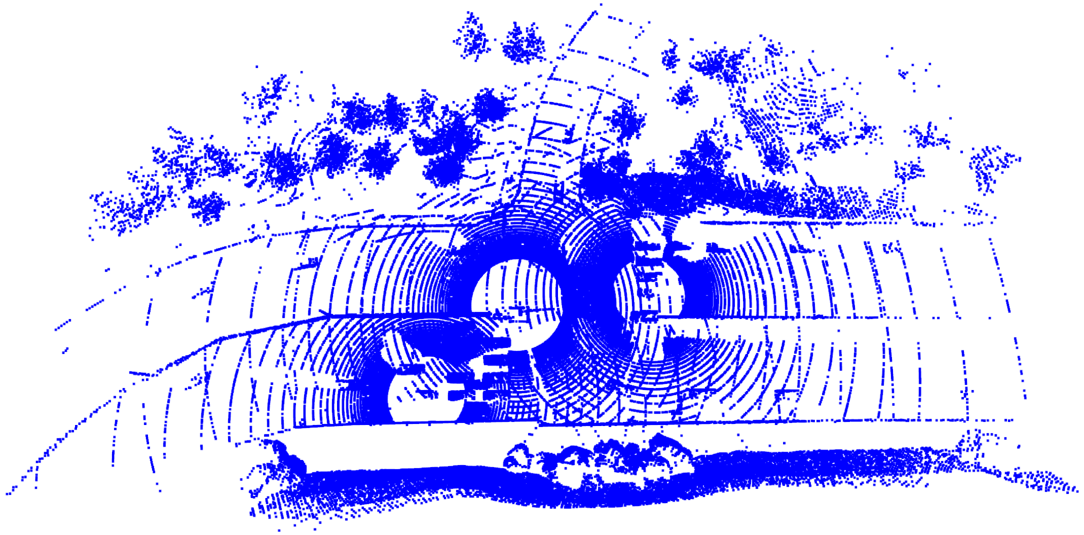
CAV 2



Infrastructure

Merging Algorithm: EMP: Edge-assisted Multi-vehicle Perception

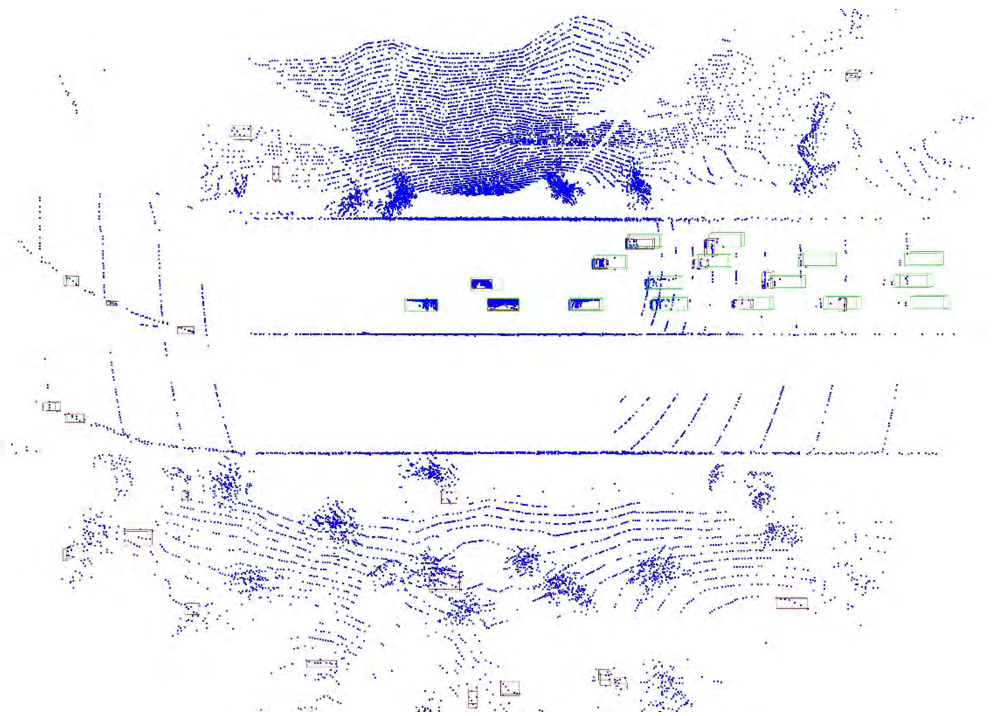
Zhang, X., Zhang, A., Sun, J., Zhu, X., Guo, Y. E., Qian, F., & Mao, Z. M. (2021, October). EMP: Edge-assisted multi-vehicle perception. In *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking* (pp. 545-558).



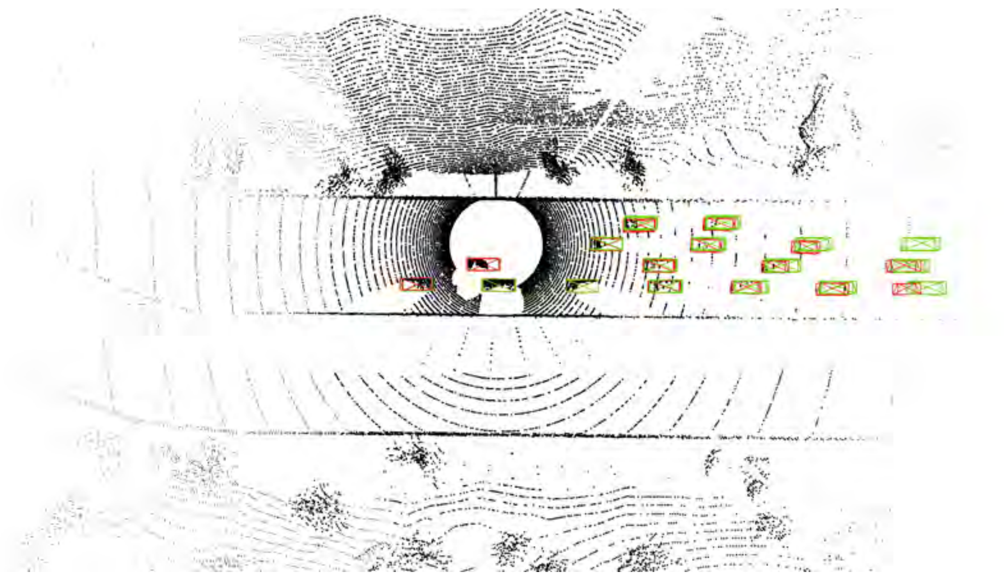
Merged Point Cloud Data

Vehicle Detection Model

- Clustering-based 3D Detection (RANSAC + DBSCAN)

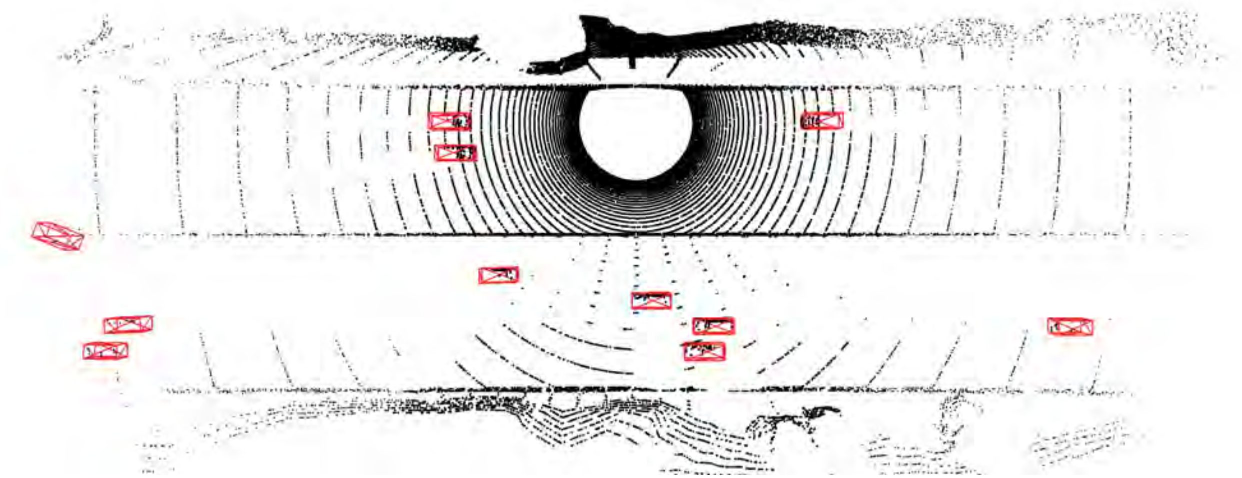


- Deep learning-based 3D Detection (PointPillar)
- Trained with Carla data

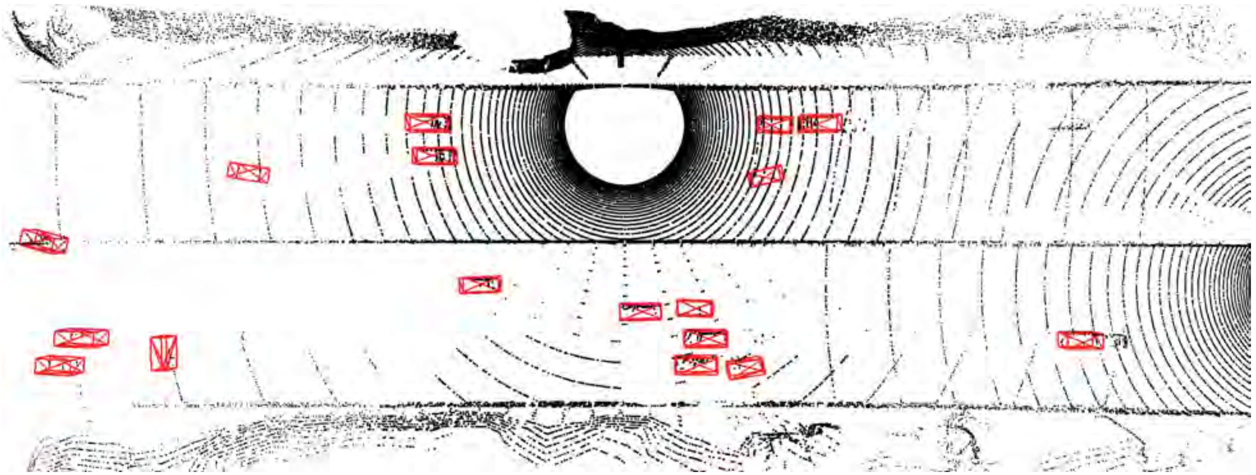


	$AP40_{0.01}$ BEV	$AP40_{0.01}$ 3D	$AP40_{0.1}$ BEV	$AP40_{0.1}$ 3D
DBSCAN	23.3318	16.7142	15.1852	7.8422
PointPillar	84.9086	84.9144	84.8281	84.8151

Comparison between single Lidar and merged Lidar point Cloud



Detection results from single Lidar point cloud



Detection results from merged point cloud

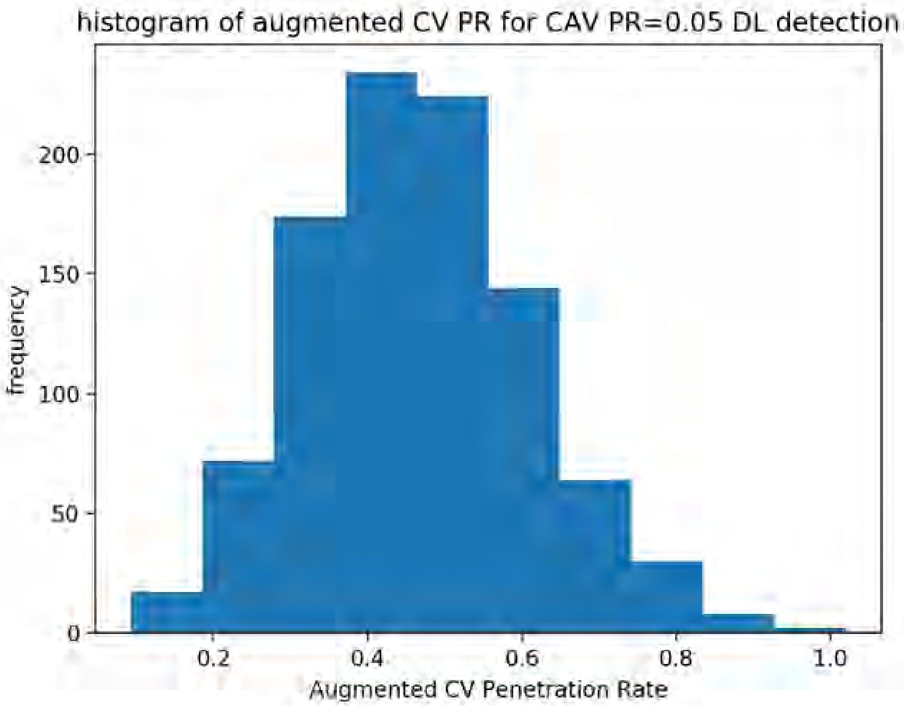
Data Collection Efficiency

- Equivalent connected vehicle penetration rate (E-CVPR)

$$ECVRP = \frac{detected\ \#\ vehicles}{total\ \#\ vehicles}$$

- E-CVPR by the PointPillar detection model

CAV Penetration Rate	E-CVPR (mean and std.)
0%	31.41% (10.12%)
1%	35.68% (10.94%)
2%	37.42% (11.89%)
5%	46.59% (14.64%)



Cooperative Perception Based Adaptive Signal Control

- Signal Control Algorithm: Max Pressure[1]
  - Calculate pressure based on number of vehicles at the incoming and outgoing approaches
  - Actuate traffic signals based on the highest pressure
- Settings
  - Varying penetration rates of CAVs from 0% to 5% with an infrastructure Lidar
  - Vehicles that are within 200m of the intersections are considered
  - Simulation time: 2000s

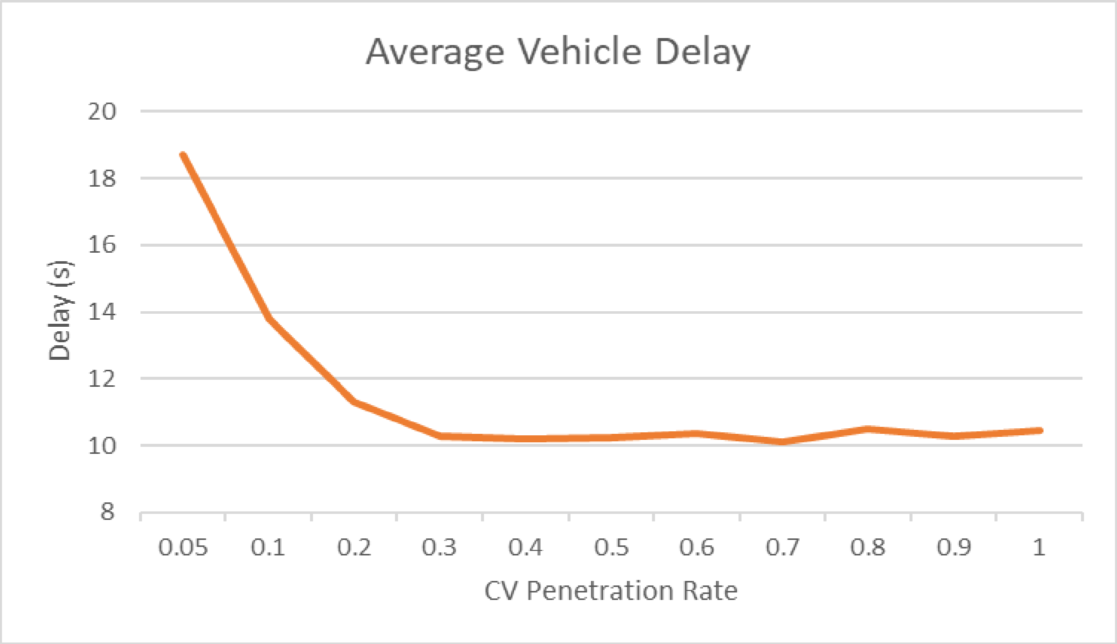
[1] Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies*, *36*, 177-195.

Demo Video

Performance Comparison

- Performance metric: average vehicle delay

CV-based Max Pressure

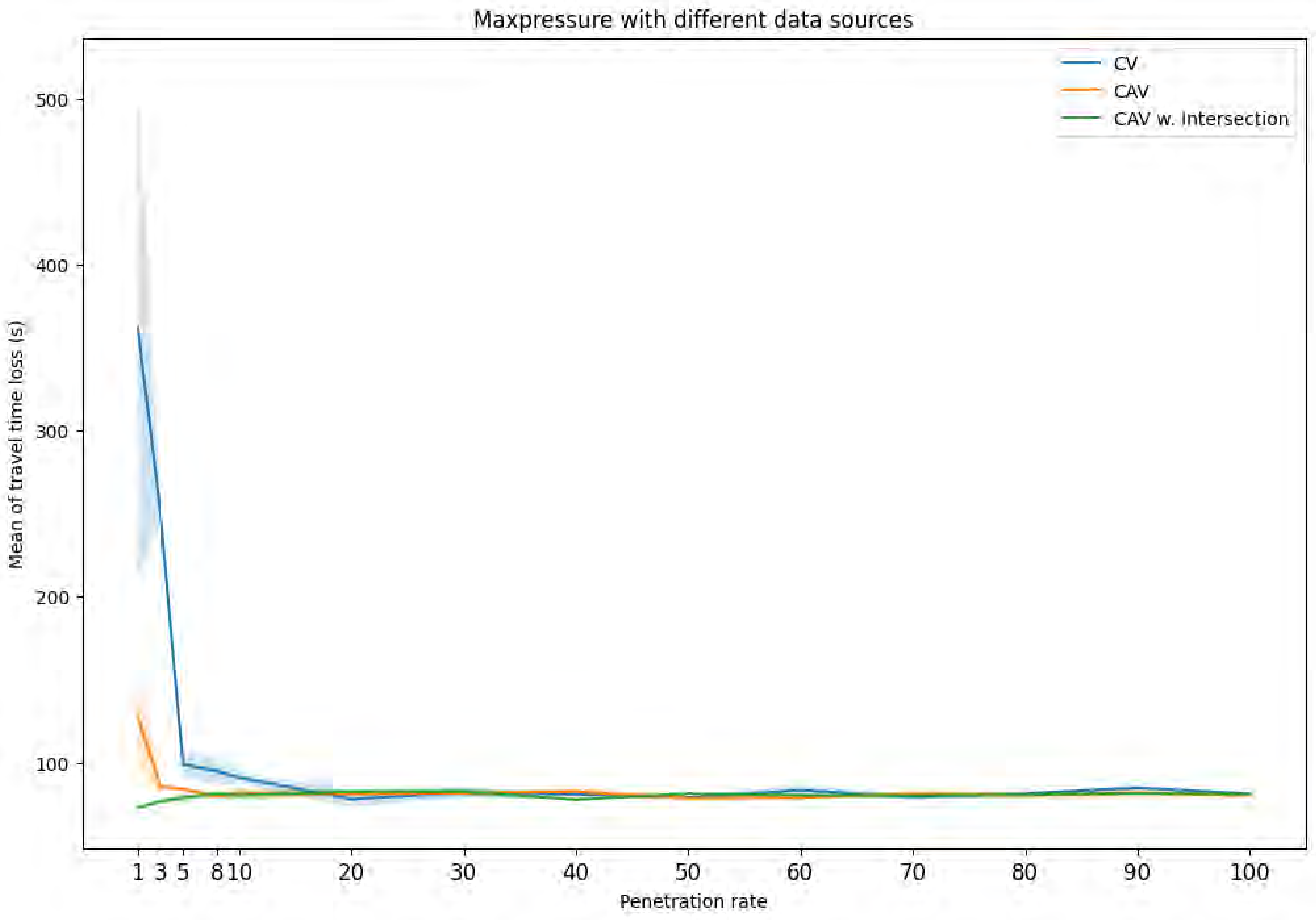


Cooperative Perception Based Max Pressure

CAV Penetration Rate	Average Vehicle Delay
0%	10.57
1%	10.86
2%	10.63
5%	11.01

Performance Comparison

- Eight-phase intersection (Plymouth and Green, Ann Arbor, MI)
- Signal Control Algorithm: Max pressure
- Three settings
  - Connected Vehicle
  - Cooperative perception with infrastructure sensor
  - Cooperative perception without infrastructure sensor



Summary

- Multi-resolution simulation platform
  - Based on CDASim from FHWA
  - SUMO + CARLA
  - Integrate with perception sensors and data
  - Integrate with perception models
  - Create a cooperative perception environment
- Cooperative perception based traffic signal control
  - Implement to very low penetration rates of CAVs
  - Infrastructure sensors play an critical role in data acquisition
- Other traffic applications

Acknowledgement

NSF

CENTER FOR CONNECTED AND AUTOMATED TRANSPORTATION

Reference:

Chen, H., Liu, B., Zhang, X., Qian, F., Mao, Z.M. and Feng, Y., 2022. A Cooperative Perception Environment for Traffic Operations and Control. *arXiv preprint arXiv:2208.02792*.

THANK YOU

QUESTIONS?

Contact: Yiheng Feng (feng333@purdue.edu)