TRL | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

Theme Poster: **99P Deploying Data Driven Digital Twins to Develop Innovative Solutions**

Start / End: Aug 2023 – Apr 2027
Director of Project: Duane Detwiler
End-User: HRI HGPU HGRX ADC

PIC: Nithin Santhanam, Rajeev Chhajer
Collaborators: Prof Sean Qian (CMU), Keisuke Suda (HGPU), Kenta Fukami (HGPU), Sebastian Brulin (HRI-EU)

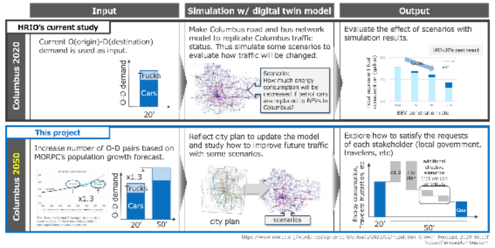
A00 Integrate large-scale digital-twin modeling, data-driven simulations to quantify impacts, expose vulnerabilities, and prototype transformative solutions

Targets / Minimum Viable Prototype

1) Open-source toolkit: simulation, data, and visual components to accelerate development
2) Develop a Digital Twin for mobility of Columbus Transportation Network
3) Use Digital Twin to test mobility and behavior hypotheses
4) Analytically Identify Network vulnerabilities
5) Heterogenous Adaptive Collaborative Object Detection through Networking Digital Twin
6) Enhance digital twin models to incorporate networking, compute and nuanced behaviors

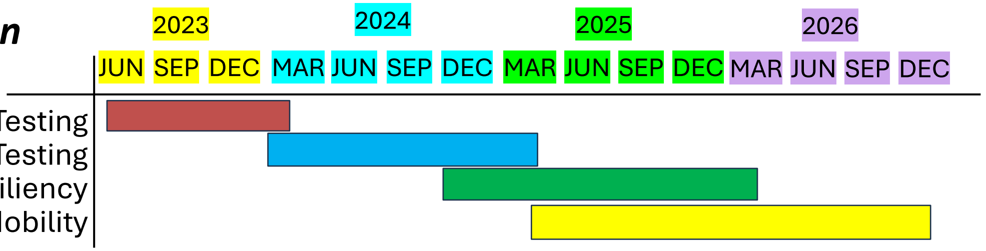
Recent Achievements

- Illustrate Effects of EV charging behavior on Transportation network & Environment
- Collaboration with HGPU to use Digital Twin to inspire Research innovation
- S. Thornton, N. Santhanam, R. Chhajer and S. Dey, "Real-Time Heterogeneous Collaborative Perception in Edge-Enabled Vehicular Environments," in IEEE Open Journal of Vehicular Technology, vol. 6, pp. 471-486, 2025, doi: 10.1109/OJVT.2025.3533368.



Papers: x2 submitted, x1 planning **Patent**: NA **HTF**: x2

Project Plan



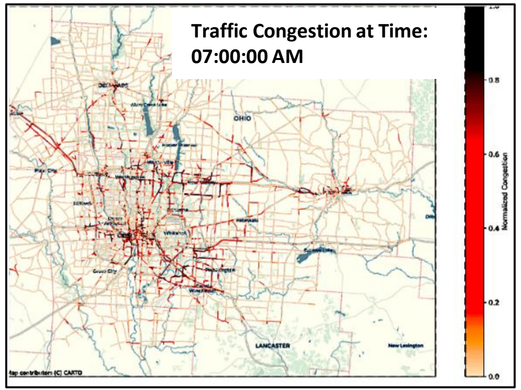
Background, Current Challenges and Barriers

Understanding Complex Transportation Network dynamics is critical to effective Interventions


Francis Scott Key Bridge collapse, Baltimore, 2024

- **Informed Decision Making:** Enables city planners to simulate and evaluate scenarios for informed decision-making.
- **Scenario Testing:** Allows exploration of hypothetical futures without real-world consequences.
- **System Optimization:** Facilitates optimization of traffic flow, energy usage, general behaviors for efficiency and sustainability.
- **Environmental Impact:** Supports modeling of transportation's environmental footprint and testing strategies to reduce emissions.
- **Scenario Variety:** Testing impacts from agent level decisions and network level decisions.

This Work:
Development & Application of Data Driven Digital Twins to better understand complex network dynamics



Challenges/ Barriers:
- Sparse/ Data Access
- Balance accuracy & performance
- Scalability
- Interdisciplinary Collaboration
- ROI uncertainties

99P Deploying Data Driven Digital Twins to Develop Innovative Solutions

This project develops data-driven Digital Twins to simulate and analyze complex transportation networks, enabling better decision-making, scenario testing, and system optimization. By modeling real-time traffic and environmental dynamics, it helps identify vulnerabilities and design more resilient, efficient mobility solutions.

99P Labs Team:
Nithin Santhanam
Rajeev Chhajer

HGPU Team:
Keisuke Suda
Kenta Fukami

Honda Research Institute, EU Team:
Sebastian Brulin

Carnegie Mellon University Team:
Sean Qian

1. Background

Modern transportation networks are complex systems with evolving challenges in traffic flow, environmental impact, and infrastructure resilience. To effectively address these issues, city planners and engineers need tools that support informed decision-making, scenario testing, and system optimization without risking real-world consequences.
This project focuses on leveraging Data-Driven Digital Twins, which are virtual models of physical systems that continuously update using real-time data. By simulating real-world transportation dynamics, Digital Twins can help expose vulnerabilities, assess environmental impacts, and test mobility scenarios under different conditions. The aim is to improve transportation resilience, efficiency, and sustainability across urban networks.

2. Methods

The project integrates large-scale digital twin modeling with data-driven simulation tools. The team is building an open-source toolkit that incorporates simulation modules, data pipelines, and visual components to accelerate development.
Key efforts include:

- Developing a Digital Twin of the Columbus transportation network.
- Simulating traffic and environmental conditions to analyze the effects of variables like electric vehicle charging behavior.
- Using Digital Twins to test mobility hypotheses and evaluate system performance under stress.
- Enhancing models with adaptive object detection and heterogeneous data fusion for realistic and scalable applications.
- Collaborating with industry and academic partners (for example, HGPU, CMU, HRI-EU) to drive interdisciplinary research and implementation.

3. Findings

Early outcomes demonstrate the potential of Digital Twins to model real-time behavior and guide smarter transportation policies. Highlights include:

- Identification of network vulnerabilities under various mobility scenarios.
- Visualization of traffic dynamics during peak hours, aiding scenario planning.
- Publications and planned patents addressing cooperative vehicle environments and scalable digital modeling.
- Demonstrated collaboration benefits, such as shared insights on EV impacts on traffic and environmental systems.

Despite these advancements, challenges remain. The team continues to tackle issues like data access limitations, model scalability, and balancing accuracy with performance. Addressing these will be crucial for translating research into impactful, real-world applications.

TRL 1 2 3 4 5 6 7 8 9

Theme Poster:
99P Cloud Edge Emulator

Start / End: 5/2024 – 5/2026 +
Director of Project: Duane Detwiler
End-User: HGRX

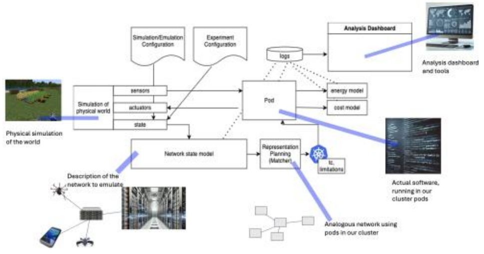
PIC: Tony Fontana, Rajeev Chhajer
Collaborators: Majd Sakr, Anthony Rowe, Greg Ganger, Chris Bogart – CMU
Deepak Ganesan - UMass Amherst (Potential)

A00 Develop a cloud-edge emulator that can facilitate streamlined cloud-edge development in software, effectively manage compute system resource allocation, optimize performance, and evaluate complex compute, networking and real-world simulation systems.

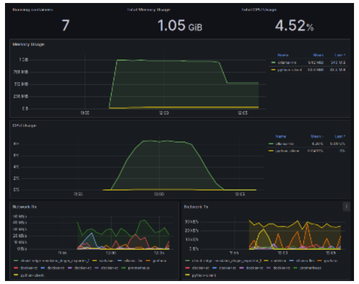
Targets / Minimum Viable Prototype

- Internally develop MVP for the emulation environment. Create toolset that can constrain compute and networking accurately to represent complex compute systems.
- Create evaluation metrics for a successful compute and application system, trade off study development.
- Develop more detailed network emulation system. (Network type considerations, real world interference, more realism).

Recent Achievements

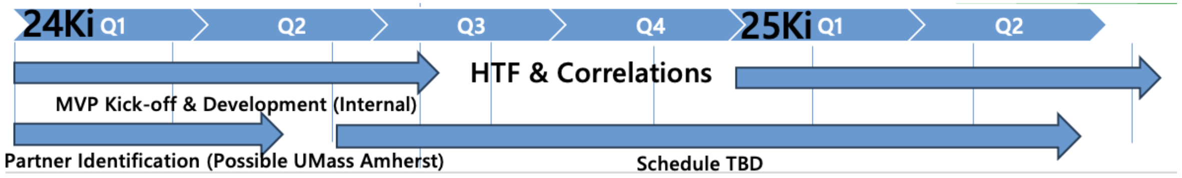


Emulator design creation



CPU + GPU + Network constraints developed in MVP, Test case developed, testing began

Project Plan



Background, Current Challenges and Barriers

With on-demand computing penetrating across the hardware and cloud continuum in a smart city context, many architectures and full-stack software paradigms are emerging such as Centralized, Decentralized, Distributed and Mesh as a few examples to enable "Systems to Cooperate"

- Embedding systems and edge compute are increasingly important for managing latency, throughput, performance, cost, and scale.
- Careful consideration needed in system design before deployment.
- These systems are becoming so complex with hundreds of IoT components and devices all needing to communicate and collaborate.
- <u>There is no current way to benchmark, measure or evaluate these complex systems</u>

Lack of benchmarking metrics adds complexity to creating optimal systems. Emulation environments help design, test, and speed up system development.

<u>Areas of Interest</u>
Multi-modal Network Emulation
Workload Placement Optimization
- Study where workloads should run: cloud, edge server, or device.

Resource-Constrained Compute Models
Real-World Stimuli
- Trigger edge cases like network drop, power loss, interference.

99P Cloud Edge Emulator

The 99P Cloud Edge Emulator is a software-only sandbox that lets engineers rehearse cloud-to-edge deployments, dynamically modeling CPU, GPU, and network constraints. By surfacing bottlenecks and quantifying trade-offs before any hardware is purchased, it cuts design risk, cost, and time-to-deploy.

99P Labs Team:
Tony Fontana
Rajeev Chhajer

CMU Team:
Majd Sakr
Anthony Rowe
Greg Ganger
Chris Bogart

1. Background

Digital services increasingly straddle “the cloud” and the physical world. Smart-city cameras, factory sensors and autonomous vehicles all push data to nearby edge servers for rapid processing, while still relying on remote cloud clusters for heavier analytics. Designing these split-layer systems is tricky: developers must juggle CPU, GPU and networking limits, decide where each workload should run, and be confident the final design will survive real-world glitches such as congestion or a sudden power cut. Today there is no standard way to benchmark or rehearse such complex, distributed deployments before expensive hardware is purchased or deployed in the field.

2. Methods

The 99P Cloud Edge Emulator tackles this gap by building a purely software testbed that can:

- **Constrain resources on demand** – The platform dials CPU, GPU and network capacity up or down, reproducing everything from a lightweight IoT device to a 5G backhaul link.
- **Inject realistic network conditions** – Latency spikes, packet loss, radio interference and power outages can be scripted to mirror harsh, real-world scenarios.
- **Guide workload placement** – Built-in analytics explore “what-if” mappings (cloud vs. edge vs. device) to spot the trade-offs among latency, cost and energy.
- **Define objective metrics** – A new scorecard quantifies overall system performance, resource efficiency and quality of experience, making designs comparable across projects.

The internal team has produced a **minimum viable prototype (MVP)** and toolset, collaborating with researchers at Carnegie Mellon and (potentially) UMass Amherst. A detailed roadmap runs from Q1 2024 through mid-2026, adding multi-modal network emulation, richer compute models and expanded real-world stimuli.

3. Findings

Early tests show the emulator can already replicate combined CPU + GPU limits and network constraints for a sample application, allowing engineers to detect bottlenecks before hardware purchase. The approach promises three tangible benefits:

- **Faster design cycles** – Teams iterate in software rather than re-cabling lab gear or shipping devices to the field.
- **Better-performing deployments** – Quantitative metrics highlight the most efficient workload split, trimming latency and energy use.
- **Shared benchmarks for the community** – A common yardstick will let industry and academia compare edge-cloud architectures head-to-head, accelerating research on decentralized, distributed and mesh systems.

As the project matures, the 99P Cloud Edge Emulator aims to become the go-to “flight simulator” for edge computing, helping designers build trustworthy, scalable systems before a single sensor is screwed into place.

TRL 1 2 3 4 5 **6** 7 8 9

Theme Poster:
99P Embedded system design for hardware agnostic-software patterns

Start / End: Jan 2024 – Dec 2025
Director of Project: Duane Detwiler
End-User: HG/HM/ADC

PIC: Brian Coy
Collaborators: Ben Davis, Luka Brkljacic (Internal)

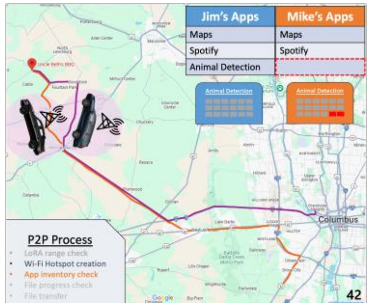
A00 Develop a software defined ecosystem with flexibility and scale utilizing the latest technologies

Targets / Minimum Viable Prototype

1) Develop Over The Air (OTA) to deliver updates to devices
2) Create cloud development environment
3) Cluster compute for embedded systems
4) Containerized applications with Robotic Operating System (ROS)
5) Peer to Peer (P2P) Communication

Recent Achievements

- P2P communication
- Cluster computation for embedded systems



Papers: x1 submitted / x1 planning **Patents**: Not Applicable **HTF**: x2

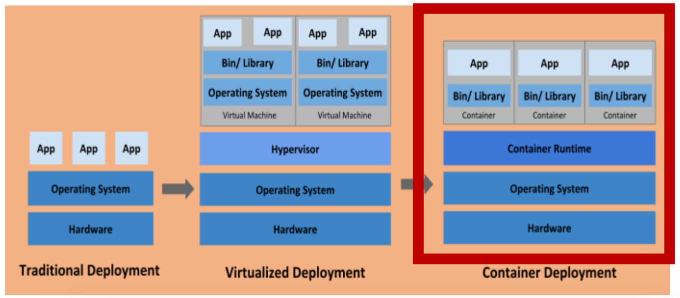
Project Plan

May Jun Jul Aug Sep Oct Nov Dec

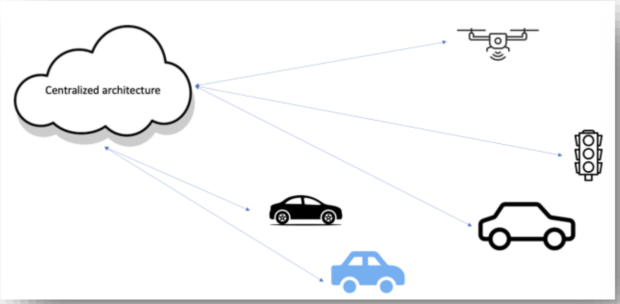
Task 3
Task 4
Task 5
HTF

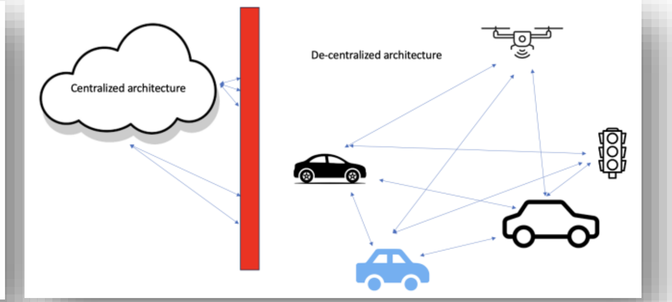
Background, Current Challenges and Barriers

Containerization to enable the flexibility to only affect the application(s) that is intended to affect



VMs	Containers
Heavyweight.	Lightweight.
Limited performance.	Native performance.
Each VM runs in its own OS.	All containers share the host OS.
Hardware-level virtualization.	OS virtualization.
Startup time in minutes.	Startup time in milliseconds.
Allocates required memory.	Requires less memory space.
Fully isolated and hence more secure.	Process-level isolation, possibly less secure.





Migration from a Client/Server (Centralized) architecture to a Peer to Peer (Decentralized) to distribute software.

Challenges:
1. Long range communication
2. File transfer with intermittently connected devices
3. Combining multiple devices to act as a centralized system of individual compute

99P Embedded system design for hardware agnostic-software patterns

This project is building a flexible, hardware-agnostic software ecosystem for embedded systems using containerization, cloud tools, and peer-to-peer communication. It enables scalable, over-the-air updates and decentralized computing across diverse devices.

99P Labs Team:
Brian Coy
Ben Davis
Luka Brklijacic

1. Background
Embedded systems are the backbone of many smart devices, from vehicles to industrial machines. However, traditional embedded software is often tightly coupled with specific hardware, making it hard to update, scale, or repurpose across platforms. This project tackles that challenge by designing a flexible, software-defined ecosystem that is hardware-agnostic. The goal is to decouple applications from underlying hardware through modern technologies like containerization, cloud environments, and peer-to-peer (P2P) communication.
The team is also addressing a shift from centralized (client-server) architectures to decentralized, peer-based systems, enabling more resilient and scalable networks for connected devices.

2. Methods
The project is using a container-based approach, where each software component runs in isolated environments, allowing specific applications to be updated or managed without affecting the entire system. Key technologies and design methods include:

- Over-The-Air (OTA) updates to remotely deliver new features or fixes.
- Cloud-based development environments to streamline testing and deployment.
- Cluster computing for embedded systems, enabling devices to pool their computing resources.
- Integration with Robot Operating System (ROS) for containerized robotic applications.
- Implementation of Peer-to-Peer (P2P) communication to enhance decentralization and reduce single points of failure.

A phased project timeline runs from January 2024 to December 2025, with recent accomplishments including working prototypes for P2P communication and cluster computing.

3. Findings
So far, the team has demonstrated working P2P communication and cluster computation capabilities among embedded systems, confirming the feasibility of this decentralized, container-based architecture. These advances pave the way for scalable deployment across varied hardware platforms.
Notably, the migration from centralized to decentralized systems has allowed:

- Better system resilience.
- Flexibility in updates and deployment.
- Support for dynamic, distributed computing models.

However, challenges remain, including:

- Managing long-range communication efficiently.
- Handling intermittent connectivity between devices.
- Coordinating multiple edge devices as part of a coherent, centralized logic model.

TRL 1 2 3 4 **5** 6 7 8 9

Theme Poster:
99P Modular and Adaptable Output Decomposition in Large Language Models

Start / End: Apr 2025 – July 2025
Director of Project: Duane Detwiler
End-User: HG/HM (anticipate tools/infra for HG/HM)

PIC: Ryan Lingo, Martin Arroyo, Luka Brkljacic, Ben Davis, Rajeev Chhajer, Nithin Santhanam
Collaborators:

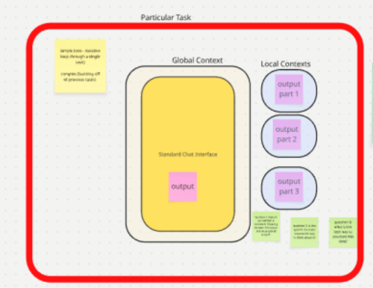
A00 Develop a framework that turns LLM outputs into clear, modular pieces, making them easy to understand, refine, and reuse.

Targets / Minimum Viable Prototype

1) Modular output decomposition
2) Node-level editing
3) UI – Browser-based workspace
4) Dependency-aware updates
5) Selective reruns

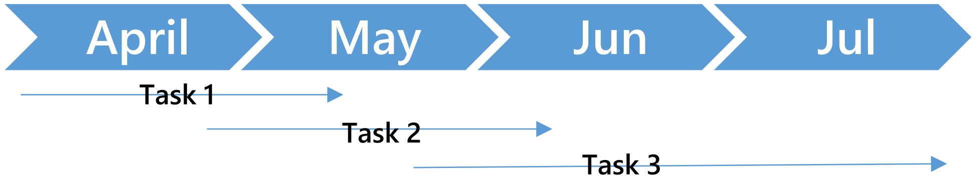
Recent Achievements

- Mapped out the key workflow from prompt to editable modules.
- Built a starter prompt set that encourages the model to label each section.
- Sketched interface wireframes and gathered initial feedback from researchers.



Papers: x1 planning x5 blogs **Patents**: Not Applicable **HTF**: Not Applicable

Project Plan



Background, Current Challenges and Barriers

The Current Challenge: Monolithic & Inflexible LLM Outputs
LLM outputs for complex tasks are often monolithic. This structure hinders:

- Understanding and verification
- Iterative refinement
- Adaption to new requirements

This research addresses the critical need for LLMs to produce inherently modular and adaptable outputs.

Our Focus: Enabling Modular & Adaptable Outputs

We are investigating methods to guide Large Language Models (LLMs) to generate outputs as a series of distinct, interconnected, and adaptable modules. Our goal is to develop methods for structured, editable, and reusable LLM responses.

This will be achieved by:

- Developing effective prompting and interaction strategies to structure outputs into logical modules.
- Exploring how LLMs can manage dependencies between these output modules.
- Employing a system framework that includes:
  - Context isolation for focused module generation.
  - Persistent memory for component management.
  - Collaborative interfaces for user interaction and refinement.

Expected Outcomes & Impact:

- Enhanced Clarity: Outputs become easier to understand, dissect, and debug.
- Improved Iterative Refinement: Users and LLMs can focus on specific parts of a solution more efficiently.
- Increased Adaptability: Solutions can be readily adjusted to new requirements by modifying or replacing modules.
- Better Human-AI Collaboration: Facilitates teamwork on distinct and verifiable components.
- Greater Reusability & Transparency: Output modules may be reused, leading to more controllable and maintainable AI solutions.

99P Modular and Adaptable Output Decomposition in Large Language Models

This project develops a framework that breaks down LLM outputs into clear, modular components, making them easier to understand, edit, and reuse. By enabling structured and adaptable responses, it improves clarity, flexibility, and collaboration in AI-assisted work.

99P Labs Team:
Ryan Lingo
Martin Arroyo
Luka Brkljacic
Ben Davis
Rajeev Chhajer
Nithin Santhanam

1. Background

Large Language Models (LLMs) are powerful tools, but their outputs are often delivered as large, monolithic text blocks. This makes it difficult for users, especially researchers and developers, to understand, refine, or adapt the model's responses. These rigid outputs hinder critical tasks like verifying information, making iterative improvements, and adapting responses to changing needs.

This project tackles the need for LLMs to produce more modular and adaptable outputs by breaking down complex answers into manageable, structured components. The goal is to help users better understand, edit, and reuse model outputs.

2. Methods

The team is developing a framework that guides LLMs to generate outputs as a collection of clear, editable modules. These methods include:

- Prompting strategies to encourage the model to structure its responses into logical sections
- Dependency management tools to track relationships between modules
- A browser-based interface for users to interact with and refine individual parts of the output
- System components such as:
    - Context isolation for generating focused modules
    - Persistent memory to manage shared information across modules
    - Collaborative tools for refining outputs with user feedback

Initial prototypes include a modular output decomposition system, node-level editing features, and selective reruns of specific sections. Early work has produced wireframes, prompt templates, and initial user feedback.

3. Findings

Although still in development (through July 2025), early progress shows promise in transforming how LLM outputs are handled:

- Clarity: Modular outputs are easier to read, understand, and debug
- Efficiency: Users can refine specific sections without reworking the entire response
- Flexibility: Outputs can be adapted to new requirements by simply updating modules
- Collaboration: Clearly defined modules make it easier for humans and AI to work together
- Reusability: Modular design enables better versioning, tracking, and control of AI outputs

This work lays the foundation for more maintainable and human-aligned AI systems, particularly valuable in research environments that demand transparency and precision.

TRL 1 2 3 4 5 **6** 7 8 9

Theme Poster:
99P Multimodal Data Management for Mobility

Start / End: Jan 2024 – Dec 2025
Director of Project: Duane Detwiler
End-User: HG/HM (anticipate tools/infra for HG/HM)

PIC: Rajeev Chhajer, Ryan Lingo
Collaborators: Hojin Yoo(PhD student), Prof Arnab Nandi

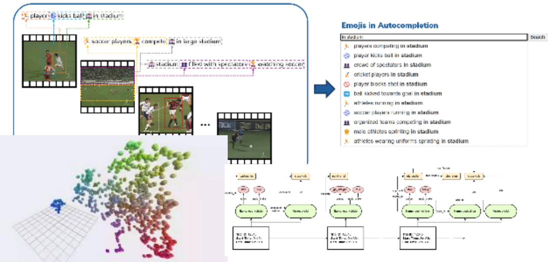
A00 Investigate methods to unify diverse mobility data using a multi-modal learning approach

Targets / Minimum Viable Prototype

1) Multimodal mobility embeddings
2) Guided querying over Videos
3) Dataset summarization
4) Embeddings Explorer for Videos
5) Multi-modal data to knowledge graphs generation

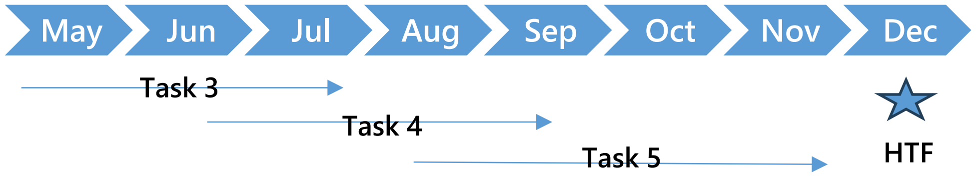
Recent Achievements

- Search phrase with emoji generation
- Agent-Based Trip Summarization
- Development of Embeddings explorer



Papers: x1 submitted / x1 planning **Patents**: Not Applicable **HTF**: x2

Project Plan



Background, Current Challenges and Barriers

Time series data
CAN bus, V2X sensors, radar

Spatial and positional data
GPS, accelerometer

Vehicle

Video and image data
Dashcam, surround cameras, LiDAR

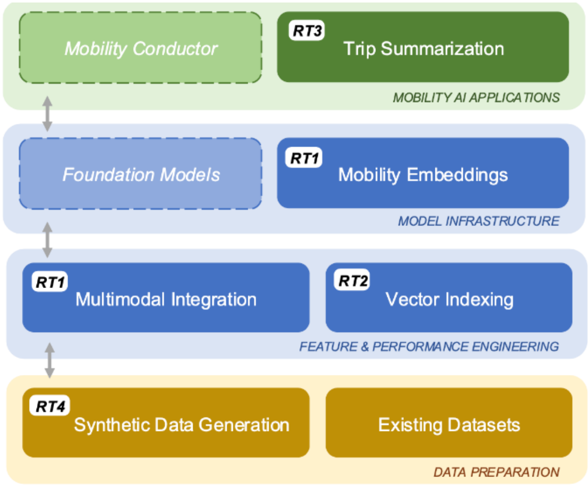
Text data
Location description, user profiles

Heterogeneity in nature of data: difficult to integrate and use together .

Extracting insights & intelligence from multi-modal data is hard, limiting a sustainable way to build AI applications in Mobility leveraging a rich data ecosystem

A foundation model (LLM) for mobility will enable the distillation of observed transportation knowledge into a powerful artifact. Such a model can be used to power a variety of use cases ranging from vehicle design, mobility analytics, to multi-vehicle traffic operations.

The overarching vision of a foundation model will require infrastructure for preparation of training, fine-tuning, and prompting data. This project seeks to develop building blocks of such a data infrastructure.



99P Multimodal Data Management for Mobility

This project develops AI methods to unify diverse mobility data such as video, sensor, and GPS using a multi-modal learning approach, enabling smarter and more integrated transportation analytics. By building foundational tools for data integration, it paves the way for scalable AI applications in mobility, from trip summarization to multi-vehicle coordination.

99P Labs Team:
Rajeev Chhajer
Ryan Lingo

The Ohio State University Team:
Hojin Yoo
Arnad Nandi

1. Background

Modern transportation systems generate vast and varied data, including GPS and accelerometer readings, video footage, vehicle sensor data, and user-generated text. While rich in insights, this multi-modal data is extremely heterogeneous, which makes it difficult to integrate, analyze, and apply effectively in AI-driven mobility solutions. This project addresses that challenge by aiming to unify these diverse data sources to support more intelligent and efficient mobility systems.

The goal is to lay the groundwork for AI tools and infrastructure that can operate effectively across this complex data landscape, enabling better decision-making, planning, and mobility analytics.

2. Methods

To address the data fragmentation problem, the project is developing a foundation model (LIM: Large-scale Integrated Model) tailored for mobility applications. This model will be capable of learning from multiple data types and will enable several downstream tasks, such as:

- Multimodal mobility embeddings, which create unified representations of data from video, sensors, text, and more
- Guided querying over videos, allowing for interactive and context-aware exploration
- Dataset summarization, which compresses large-scale data into more understandable formats
- Embeddings Explorer for Videos, a visual tool for interpreting how AI models understand video content
- Graph-based knowledge generation, which turns raw data into structured knowledge

The architecture includes an integrated pipeline for data ingestion, synthetic data generation, embedding creation, vector indexing, and search. Tools like the "Trip Summarization" and "Mobility Conductor" modules are also being developed to showcase practical applications.

3. Findings

Although the project is still in progress (Jan 2024 to Dec 2025), several milestones have already been achieved:

- A search phrase system with emoji generation for intuitive querying
- An Agent-Based Trip Summarization method that simplifies mobility event data
- An initial version of the Embeddings Explorer, which helps visualize and interpret AI-generated representations

These early results show that integrating multi-modal mobility data into a unified learning framework is not only feasible but can significantly enhance transportation analytics and planning. With one paper submitted and another in planning, the project is making strong research contributions while also focusing on tools that will be useful for a wide range of mobility applications. The outcomes are expected to support smarter mobility systems, improve data accessibility, and enable scalable AI infrastructure for transportation.