VizAV: Visualizing Autonomous Vehicle Data

Students:
Michelle Cain 318005170
Hadar Daya 323831131
Shir Berko 208152405

Introduction to Autonomous Driving Data Ecosystems

The Technological Revolution of Mobility

The autonomous driving landscape represents one of the most complex and transformative technological frontiers of the 21st century. At its core, this technological revolution is powered by data – not just massive quantities, but incredibly nuanced, meticulously captured moments of mobility. These data ecosystems serve as the digital nervous system for autonomous vehicles, guiding decisions and enabling perception under ever-changing road conditions.

Autonomous driving involves the tight integration of multiple technological domains - robotics, artificial intelligence, computer vision, systems engineering, and high-definition mapping- into a single, intelligent agent capable of safe and efficient navigation.

The Lyft L5 Prediction Dataset: A Technological Milestone

The Lyft L5 Prediction Dataset, which contains over 100GB of semantic maps and time series motion data collected from real urban driving scenarios, represents a comprehensive library of mobility. It captures the intricate interactions between vehicles, pedestrians, and urban infrastructure.

Each entry within this dataset is a microcosm of real-world driving behaviour, offering researchers a lens into the complexity of urban navigation and enabling the development of machine learning models for motion forecasting, behaviour prediction, and intelligent path planning.

Dataset Architectural Framework

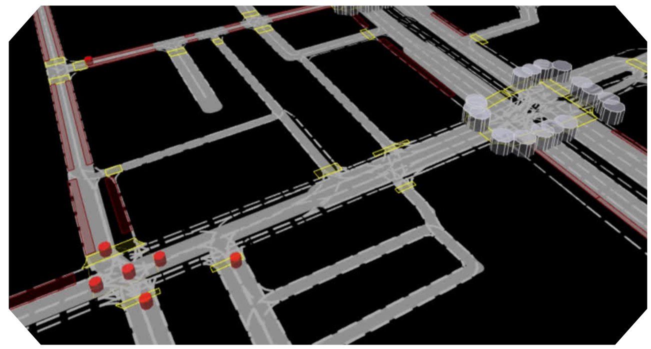
The Lyft L5 prediction dataset contains two main types of data:

1. Static Environment- Semantic map
2. Dynamic Environment- Scene database

This dual structure allows both context-aware navigation and dynamic motion prediction

Static Environment – SEMANTIC MAP

The semantic map is far more than a simple geographical representation. It is a 2D annotated map of roads, lanes, traffic control elements, speed limits, parking zones and more, that represents the fixed elements of the driving environment.
This map helps offload perception tasks from the AV by predefining the driving environment. without a semantic map, this static information would need to be continuously perceived and interpreted by the vehicle's sensors and CPUs.



Semantic Map Visualization – roads/lane lines, parking zones, crosswalks stop signs, traffic lights (gray cylinders)

Semantic Map Component Breakdown

Map Component	Detailed Elements	Precision Level	Purpose
Road Network	Lane Geometry, Intersection Topology	Centimeter-level	Navigation Planning
Traffic Control	Traffic Lights, Stop Signs, Crosswalks	Exact Positioning	Rule-based Decision Making
Regulatory Information	Speed Limits, Turn Restrictions	Precise Annotation	Compliance Modeling
Environmental Features	Parking Zones, Road Markings	Detailed Representation	Contextual Understanding

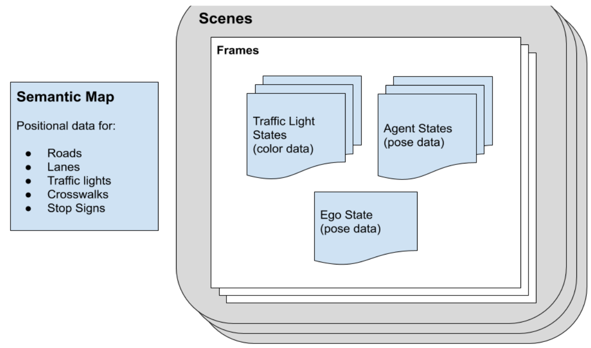
The following Python class defines the structure of a semantic map used in autonomous driving systems. It organizes road elements such as lanes, intersections, traffic controls, and regulatory zones into a structured and easily accessible format.

```
class SemanticMapArchitecture:
    def __init__(self):
        self.road_network = {
            'lanes': [],  # נתיבים – רשימה של כל הנתיבים
            'intersections': [],  # צמתים – רשימה של כל הצמתים
            'traffic_control_points': {  # נקודות בקרה תנועתית:
                'traffic_lights': [],  # רמזורים
                'stop_signs': [],  # תמרורי עצור
                'crosswalks': []  # מעברי חציה
            },
            'regulatory_zones': {  # אזורים עם רגולציות:
                'speed_limits': {},  # מגבלות מהירות (כמילון)
                'turn_restrictions': {}  # הגבלות פנייה (כמילון)
            }
        }
```

Dynamic Environment- SCENE DATABASE

The scene database captures time-varying elements in real-world driving.
It provides essential input for training machine learning models that predict vehicle behavior under dynamic conditions.

Key Characteristics:



Structural Diagram of the L5 Prediction Dataset

1. **Temporal Structure:**
    - Data is organized into frames (single timestamps) and scenes (sequences of frames).
    - Scenes are the fundamental training units for machine learning.
2. **Dynamic Driving Conditions Captured:**
    - Location and speed of surrounding vehicles, pedestrians, and cyclists.
    - Traffic light states (green, yellow, red).
    - Continuity of motion across time for agents and the ego vehicle.
3. **Each 25-second scene captures:**
    - Ego vehicle state – position, orientation, and motion.
    - Agent interactions – tracking of nearby road users.
    - Traffic signal behavior – changes and timing.
    - Probabilistic object classification – uncertainty in object labeling.
    - Temporal correlations – patterns across frames for consistent motion analysis

Dynamic Data Composition

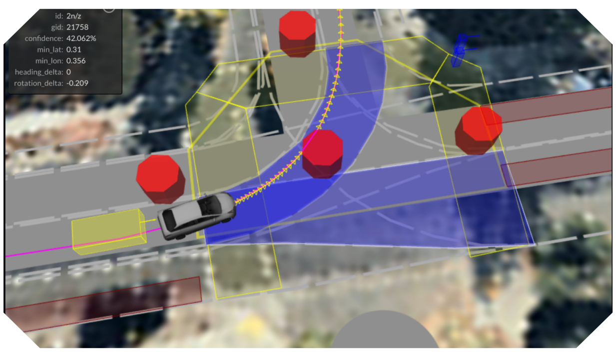
Data Dimension	Captured Information	Temporal Resolution	Information Density
Ego Vehicle	Position, Velocity, Orientation	Millisecond-level	High Precision Tracking
Surrounding Agents	Trajectories, Classification Probabilities	Frame-by-Frame	Complex Interaction Mapping
Environmental State	Traffic Signals, Road Conditions	Continuous Update	Contextual Awareness

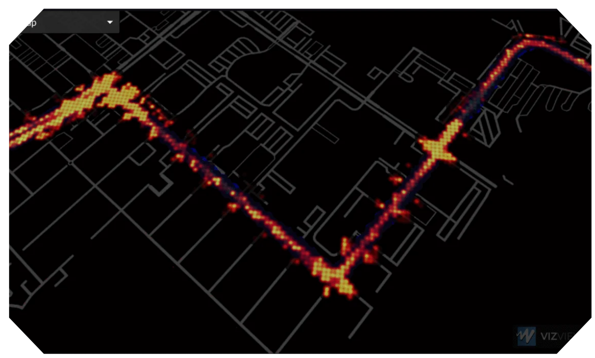
VizViewer: Transforming Raw Data into Insights

VizViewer is a **web-based interactive platform** for visualizing and analyzing large, multi-modal datasets – like those used in autonomous driving.

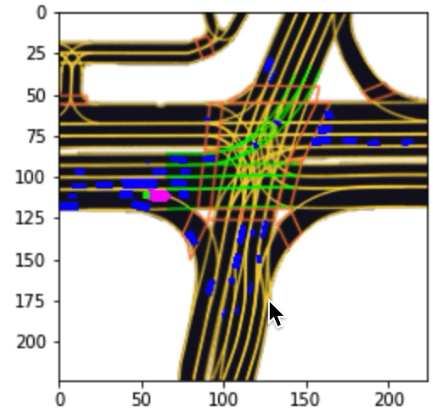
Key Features:

- Offers interactive, dynamic, and intelligent visualization capabilities
- Allows researchers to synchronize scene playback, inspect agent motion, query map elements, and evaluate prediction models
- combines data processing, communication, and visualization tools into one easy-to-use dashboard interface.
- Supports integration with Python and Jupyter Notebooks.
- Helps explore dense, complex data to gain insights, debug models, and build better training sets.
- Designed for collaborative and efficient data analysis workflows.





Semantic map visualization



The L5 Prediction Dataset Kit comes with a simple tool for visualizing the semantic map and scene data together. The tool can take a specific set of coordinates and dimensions to generate an image of the roads, lane lines, and other labeled elements. It can also render the traffic lanes' dynamic state by marking specific lanes a certain color if the lane's traffic is affected by the traffic light, etc.
When a traffic light is red, the lanes it controls are also marked red. These images can be merged to make a short movie clip of the scene, shown below.

As an **alternative**, VizViewer has an interactive 3D rendering toolkit that can render the semantic map with free form exploration along with a scene-specific view.
The map can be zoomed similarly to other online mapping tools and has support for satellite and vector map layers.

With VizViewer , the map can be navigated and examined for details that might be interesting for training our models. For example, if we are looking for samples related to left turns onto a multilane street, we can examine the map for street intersections that fit this case and then filter our samples by the coordinates of this region of interest.

VizViewer integrates with Phyton, allowing data to be aggregated and processed using Python code, then sending data to VizViewer for rendering via a Python API.
For example, VizViewer has data querying features that allow objects to be highlighted in the 3D view based on features of interest. A feature query can be defined in Python. Then, with an API call, the VizViewer dashboard will update, find, and select the features that satisfy these conditions.

The image shows the semantic map search results by highlighting roads with a decreasing minimum number of lanes criteria.

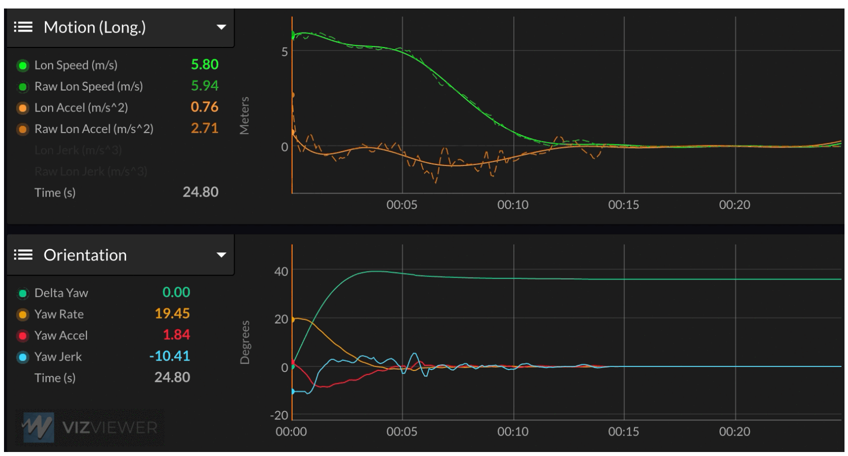
Feature Augmentation

In the context of machine learning, feature augmentation and data engineering are the processes of molding the data into a form that improves model convergence and accuracy. For example, models could converge faster if their feature values are rescaled into a smaller range.

A few helpful features can be derived from spatial data about the motion of the objects. These can be used to build a motion model for a given object type.
For example, because cars have wheels, they should move mostly forward and backward, but not freely from side to side. Therefore, a motion model that independently tracks orientation, longitudinal, and lateral motion would be desirable.
With a **motion model**, an object type's appropriate dynamics can be trained, simulated, and tested for validity.

Relationships between two objects

Another set of derived features can model the relationships between two objects. These features will help train the model to understand how to generate a planned path given the dynamics between objects (slow down if you are approaching an object) and the environment (slow down when approaching a turn or stop sign).



Above is a plot from VizViewer that shows some of these augmented features, such as velocity and acceleration.
The dotted lines represent the unfiltered values, while the solid lines represent the smoothed values derived from spline-based interpolation methods. The smoothing is applied via Python code to assist with the convergence of a trained model using these features.

Data Exploration & Visualization with VizViewer

Having described the data features in detail, exploring and visualizing these features is a helpful part of building effective training and validation sets for a Machine Learning model.

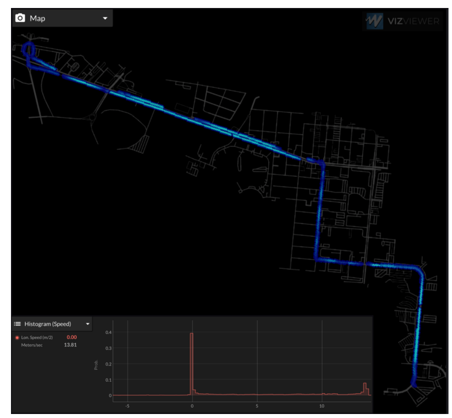
VizViewer enables users to interactively explore scenes on a clickable map.
By selecting a path, we can view detailed plots of the ego vehicle's motion along with interactions from surrounding agents such as vehicles and pedestrians.

To support in-depth analysis, VizViewer offers a **configurable dashboard** where multiple visualization panels (3D maps, time-series charts, and histograms) can be arranged according to the user's needs. This setup provides a holistic view of the scene, combining spatial and temporal data for better insights

Heat Map Analysis

Heatmaps in VizViewer are powerful tools for analyzing spatial patterns in driving behavior. They aggregate data over the map and use color to represent the magnitude of a specific feature, such as speed or agent density.

A **speed heatmap** reveals that brighter regions correspond to high-speed roads, while darker areas indicate slower side streets. This can indicate regions of the map with fast- moving traffic vs. slower, more regulated traffic.



VizViewer: Velocity heatmap and histogram. High speeds are brighter blue.

Similarly, heatmaps showing **agent observation continuity** highlight that longer tracking sequences are rare but valuable, as they provide more reliable training data for long-horizon prediction tasks.

An insightful pattern arises when overlaying heatmaps of **ego speed** and **agent density**. Intersections, where the ego vehicle typically slows down, often show a high concentration of agent observations- key scenarios for modelling complex traffic interactions.

By identifying these spatial trends, heatmaps help guide **feature engineering** and ensure the training dataset is balanced and diverse, avoiding bias toward common or simple driving situations.

Path Evaluation & Visualization with VizViewer

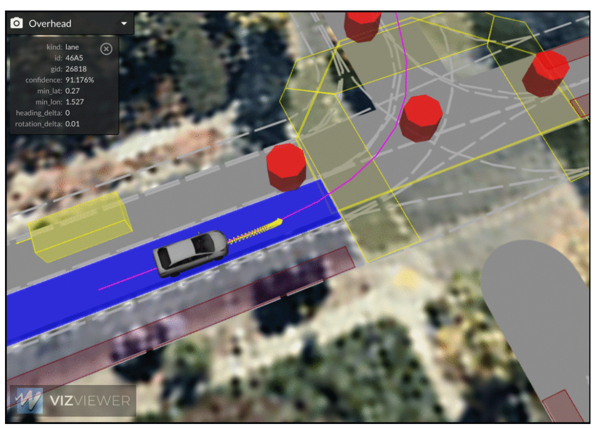
VizViewer is a powerful tool for visualizing and evaluating the performance of path prediction models. By integrating scene data with semantic maps, we can train models that predict the future paths of vehicles based on features like vehicle pose and motion.

The 3D interactive simulation in VizViewer allows for real-time visualization of vehicle routs and the surrounding environment, including agents, lane states, and traffic signals. This dynamic visualization is synchronized with data plots that show key features like speed, acceleration, and model confidence.

Lane-level path predictions are displayed with color-coded confidence values: the darker the blue, the higher the confidence in the prediction. This allows users to easily identify areas where the model is more certain about the vehicle's future path.

Additionally, users can **track how path predictions evolve over time** as the vehicle moves through different environments. Clicking on the predicted paths reveals **raw model outputs**, such as regression values and confidence scores, providing deeper insight into the model's decision- making process.

VizViewer's interactive UI enables easy debugging by detecting **deviations from predicted paths** or potential collisions with other objects. The smoothness of the path can also be assessed using speed and acceleration charts, which helps refine the model and ensure more accurate behaviour predictions.



the system correctly predicted a left turn before it occurred, indicating strong contextual understanding.

Conclusion

The Lyft L5 Prediction Dataset is more than a repository of driving scenes. It reflects our most advanced understanding of mobility, data, and urban complexity. Through tools like VizViewer and advanced data science workflows, we are not only building better AV systems- but we are also learning how to better understand the roads we share.