Mobilytics-Gym: A simulation framework for analyzing urban mobility decision strategies

Chinmaya Samal[1], Abhishek Dubey[2], Lillian J. Ratliff[3]

Abstract—**The rise in deep learning models in recent years has led to various innovative solutions for intelligent transportation technologies. Use of personal and on-demand mobility services puts a strain on the existing road network in a city. To mitigate this problem, city planners need a simulation framework to evaluate the effect of any incentive policy in nudging commuters towards alternate modes of travel, such as bike and car-share options. In this paper, we leverage MATSim, an agent-based simulation framework, to integrate agent preference models that capture the altruistic behavior of an agent in addition to their disutility proportional to the travel time and cost. These models are learned in a data-driven approach and can be used to evaluate the sensitivity of an agent to system-level disutility and monetary incentives given, e.g., by the transportation authority. This framework provides a standardized environment to evaluate the effectiveness of any particular incentive policy of a city, in nudging its residents towards alternate modes of transportation. We show the effectiveness of the approach and provide analysis using a case study from the Metropolitan Nashville area.**

Index Terms—**Transportation; Urban Mobility; Simulation;**

I. Introduction

With rapid innovations in intelligent transportation technologies (ITS), the transportation landscape is changing at a faster pace today than at any point in history. On-demand mobility services such as ride-shares are becoming more commonplace and are often considered more convenient than public transit options. Use of personal and on-demand mobility services puts a strain on the existing road network in a city. This is evident from the United States Census Bureau's annual American Community Survey 2017 report [1], which states that 85.3% of all commuters commute to work by personal vehicles, either driving alone or carpooling.

Many cities in the United States have started revising their transportation demand management (TDM) programs [2] with a strong focus on investment in road networks, traffic sensors, and public transit services. However, these services often come with a cost that needs to be borne by the commuters. Planning and investing in these services requires a lot of manpower and time. Hence, TDM programs focus mostly on providing incentives to commuters for using alternate modes of travel (e.g., bike- and car-share options) [3], [4]; such efforts have been shown to alleviate some of the cost incurred by users for using certain preferred modes.

[1]Chinmaya Samal and [2]Abhishek Dubey are with the Department of Electrical Engineering and Computer Science and the Institute for Software Integrated Systems, Vanderbilt University, Nashville, TN 37240

[3]Lillian J. Ratliff is with the Department of Electrical and Computer Engineering at the University of Washington, Seattle, WA 98195

To develop an effective incentive policy, the city planners need to evaluate the effect of any incentive policy on key system metrics such as congestion, CO2 emissions, monetary cost and so on. To this end, they often utilize various simulation tools [5]–[8] to model traffic mechanics in road networks. A common process is to first gather commuter data from their database, which is then used to generate a time-varying demand distribution and mode distribution of commuters. These distributions are then used to generate user preference profiles and finally, these profiles are used by simulation tools to analyze the impact of commuters on network congestion. However, the process of generating user preference profiles needs some prior assumptions from the city planner to create more plausible trips for commuters, that will be then used in the simulation. There is a need for a simulation that can integrate data-driven models which learn commuter preferences from historical trip data of commuters and use those preferences in evaluating different incentive policies.

State of the art models [9]–[11] are used for modeling commuter preferences in simulation tools. However, these models often fail to capture complex multi-modal dynamics. The functions that govern basic mechanics of agent-based traffic simulation often use a one size fits all approach—i.e., models all agents or commuters in the system with some specific behavior model with a fixed set of features. As evidenced by recent studies [12], [13], this is often incongruent with real-world behavior of commuters which varies with time, network conditions, and other hidden features.

As more commuters adopt multi-modal mobility options, they often have a rich set of non-overlapping preferences which factor into their decision to choose a particular route. Many other facets of the commuter decision-making process are also not accounted for in state of the art simulation studies. For instance, how a commuter makes decisions on when to travel, which mode of travel to use, what routes to traverse are often not jointly considered. Furthermore, recent studies [14] show that commuters are getting more environmentally conscious and are sensitive to the costs imposed by other commuters in a city, in terms of local and global pollution, oil dependence, accidents, and noise. Hence, there is a critical modeling gap between simulation and data-driven models learned with a real-world dataset.

To bridge these gaps in a multi-modal environment, this paper focuses on how to offer a theoretically sound and practically useful integrated modeling framework. Towards this end, we leverage MATSim [5], an agent-based simulation framework, to incorporate various decision-making models

978-1-7281-1689-1/19/$31.00 ©2019 IEEE
DOI 10.1109/SMARTCOMP.2019.00064

and provide a standardized environment to evaluate the efficacy of these models in terms of the system-level impact of particular intervention mechanisms. In particular, we make the following key contributions:

1) Integration of agent choice models considering disutilities and altruism within a well-used agent based multi-modal transport simulator. These models enable us to evaluate the sensitivity of an agent to system-level disutility and the monetary incentives given by the transportation authority of the city.
2) We also show how to learn these agent behavior models from real-world data. This enables the simulation to be configured for specific cities.
3) Finally, we present a case study from the Metropolitan Nashville area. Based on models learned from the real-world data, our simulation shows that $\mathbf{23}\%$ of agents using cars changed their modes to bus and walk when provided higher incentives. For the same simulation, our results show that mean excess travel time of all agents decreased by **12 min** of mean excess travel time (MET), yet the system-level cost is approximately $\$\mathbf{20000}$ due to incentives.

The remainder of this paper is organized as follows. We overview prior related work in Section II. In Section III, we give a brief overview of the Mobilytics-Gym architecture including a discussion of each component and present a simulation workflow that uses this architecture to evaluate incentive policies. In Section IV, we describe the implementation of the simulation workflow specific to the case study for the Nashville metropolitan region. We conclude the article, suggest avenues of further investigation in Section V. Table I summarizes the symbols we used throughout this paper. The source code is available on Github[1].

II. State of the art Simulation Tools for Transportation Studies

State of the art simulation tools such as MATSim [5], AIMSUN [6], SUMO [7] which are used in transportation studies [8], are effective tools for analyzing the impact of commuter route choices on network congestion. MATSim [5], in particular, supports implementing large-scale agent-based transport simulations and is based on iterative dynamic traffic assignment. That is, every agent repeatedly optimizes its daily activity schedule while in competition for space-time slots with all other agents on the transportation infrastructure. While MATSim is capable of iteratively finding *best* routes for users, we are interested in creating a simulation framework that mirrors the real-world in the sense that given a system state by the simulator, the agents use learned models to choose their next actions and finally the effect is observed by all the agents. In this sense, we use MATSim as a one-shot simulator to test the viability and performance of different models that make up the system.

[1]https://github.com/scope-lab-vu/mobilytics-gym

TABLE I: List of symbols

Symbol	Description
V	Set of network vertices
E	Set of network edges
M	Set of modes allowed in graph G
M_e	Set of modes allowed on an edge $e \in E, M_e \subseteq M$
τ	Time of the day
G_τ	State of directed multi-modal graph (V, E, M) at time τ.
R	Directed path from source vertex $s \in V$ to destination vertex $d \in V$
A	Actuator function
T	Actual travel time disutility function on an edge $e \in E$
$\hat{T}$	Expected travel time disutility function on an edge $e \in E$
I	Incentive function set by the system
C	Cost disutility function for an edge e
S_τ	State of environment $(G_\tau, I, T, C, \hat{T})$ at time interval τ
F	Set of fixed or scheduled plans
ET	Expected Travel time disutility function for route R
EC	Expected cost function for route R
ESD	Expected Social disutility function for route R
TR	Function that uses uses current state along with current plan to trigger new routing decision
MC	Function that evaluates the probability of choosing a mode sequence in route R
RC	Function that evaluates route choices based on travel time, cost, social cost, mode choice
MET	Function that calculates mean excess travel time incurred due to route plan of all agents in simulation
IC	Function that calculates total cost incurred on the system, due to incentives

One of the key problems with state-of-the-art simulation tools is that the agents' plans are fixed during the simulation. This makes it difficult to evaluate the effect that incentive policies which depend on the state of the network (e.g., congestion level) can have on changing agent behavior. That is, if agents plans are fixed, then agents cannot react to the incentives or any policy that needs to change their decisions during the simulation. We can mitigate this problem by using an approach where we start with an initial route plan, observe the network congestion and then change route plans that have the potential to possibly minimize congestion. In practice, the approach to changing commuter behavior tends to be *adhoc*, differs from region to region, and often depends on prior assumptions made by local city planners. To capture these differences and enable simulation of a wide variety of approaches, the capability for simulated agents to change their route in simulation needs to be integrated into state-of-the-art simulators. Such a feature is essential for a standard framework that can be configured for any city.

For analyzing the commuter impact on congestion, commuter travel behavior models are widely used in simulation tools [9]–[11]. However, these models only capture the disutility of a user proportional to travel time and monetary cost. Studies from the literature on routing games show that some commuters are partially willing to suffer (take on additional cost) for the good of society [15], [16]. Models that capture

In the simulation tools previously discussed, several needed features are lacking. For instance, in the state-of-the-art, exact route plans are specified to the agents. In MATSim, agents can have different preferences, but the weights for each of the preferences need to be given explicitly before the simulation is started. However, in practice, a city planner does not have real-time routes of agents nor do they know the exact weights corresponding to the agents' preferences. Thus, data-driven models that can learn agent preferences automatically are also needed. Lastly, a traffic prediction model that can estimate future states is also needed in order to support a real-time router.

III. Mobilytics-Gym

In this section, we give a brief overview of the Mobilytics-Gym architecture. Then, we detail each component of the architecture. Finally, we present a simulation workflow that uses this architecture.

A. Architecture

Figure 1 shows an agent-environment interface, where the *Agent* is modeled as a decision-maker and everything external to the agent is part of the *Environment*. An agent receives the current state and makes a decision on what action to take, given the current state. Here action is an interaction with any entity in the transportation network such as choosing a road link, boarding a bus, and so on. Such actions from all the agents modify the environment state.

This is very similar to how the agent interacts with the environment in a *Markov Decision Process (MDP)* [17] where at each time step, the *Environment* is in some state, and the *Agent* may choose any action that is available in that state. The process responds at the next time step by randomly moving into a new state, and giving the *Agent* a corresponding reward. The key difference with our setup, however, is in how a commuter makes a decision when they plan their trip and the absence of reward function in our model. Note that we use the terms agent, commuter and user interchangeably. In this paper, we only focus on modeling and evaluating different learnable components that constitute the agent's sequential decision process, while leaving the discussion on designing proper reward functions to achieve some system objectives for future work.

An agent is defined by a request query (s, m, d, τ), where $s \in V$ is the source, $d \in V$ is the destination, $m \subseteq M$ are set of mode choices of agent and τ is the departure time or arrival time of the trip. Each agent has a *Plan Memory* component, which is responsible for storing the current route plan R for the query. The action taken by the agent is based on the route plan stored in *Plan Memory*. An agent can also change its next plans in *Plan Memory* based on the network state and its preferences. There are two types of agents in our framework:

1) **Fixed Plan Agent**: This agent has a pre-planned route and does not have any decision model. Hence, they always follow the route plan (from *Plan Memory*) and never change its route during a simulation. For example- Public transit vehicles, background traffic and so on. These agents play a role in affecting the plan of other agents in the system. We define $\mathbf{F} = \{\mathbf{R_1}, \mathbf{R_2}, \cdots, \mathbf{R_{|F|}}\}$ as a set of fixed routes, where $|F|$ denotes the total number of fixed plans in the system. Note, we are using the terms plans and routes interchangeably.
2) **Dynamic Plan Agent**: This agent differs from the *Fixed Plan Agent* in that, it can change it's routes dynamically in network based on network conditions and it's own preferences. The preference models used by this agent emulates an agent's decision-making process. Specifically, we try to capture the sequential decisions made by an agent while planning their trip. This process mostly involves following steps: (1) Agent decides if a router is needed to get a new trip plan, (2) Agent asks a router for new route considering its preferences, (3) Agent decides if the route given by the router should be followed. We enable these decisions by integrating *Trigger Model* (Section III-B7), *Router* (Section III-B8), *Mode Choice model* (Section III-B9), *Route Choice model* (Section III-B10). These preference models are learned from historical trips of an agent.

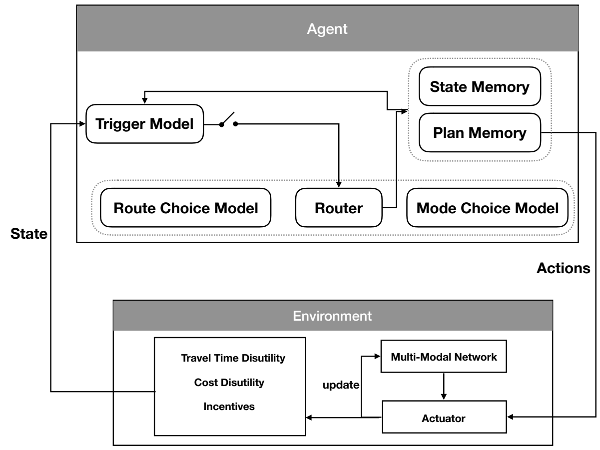

Fig. 1: Mobilytics-Gym architecture

B. Components

1) Multi-modal Network: The base component that constitutes *Environment* is a graph network which is used by agents to move from an origin to a destination. The graph is time-dependent since it changes with actions taken by all agents in the network. Let $\mathbf{G}_\tau = (\mathbf{V}, \mathbf{E}, \mathbf{M})_\tau$ be a time-dependent, multi-modal, directed graph at time τ, where V is the set of vertices, $E \subseteq V \times V$ the set of edges and M is the set of modes allowed at time τ. We say there is an edge from $u \in V$ to $v \in V$, if and only if $(u, v) \in E$. Note that we use the terms graph and network interchangeably. Each edge $e \in E$ has a set of edge labels M_e that denotes the different modes of transportation allowed on e, where $M_e \in M$.

Let R be a directed path from any source vertex $s \in V$ to a destination vertex $d \in V$. A directed path of length $|R|$

from s at time τ_s to d at time τ_d is a set of edges along with mode used to travel on each edge and time at which an edge is traversed (enter or exit), i.e

$$R = \{(e_1, m_1, \tau_1), (e_2, m_2, \tau_2), \cdots (e_{|R|}, m_{|R|}, \tau_{|R|})\} \subseteq E,$$

where $e_i \neq e_j, \forall i \neq j$, $m_i \subseteq M_{e_i}$ and $\tau_s < \tau_i < \tau_d$, $\forall i$.

2) Actuator: We need a function to evolve the state based on the actions taken by all the agents in the system. To this end, we formally define an actuator function as $\mathbf{A}(\mathbf{G_{\tau-1}}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_n})$, which uses the past network state $G_{\tau-1}$ and actions $\{a_1, a_2, \cdots, a_n\}$ taken by all agents in the network at time $\tau - 1$, to give new state of the network G_τ at time τ.

3) Travel Time Disutility: Since the graph G_τ is time-dependent, the travel time on an edge $e \in E$ also varies with time. The travel time on an edge also varies with the mode being used by the agent. Let $\mathbf{T}(\mathbf{e}, \mathbf{m}, \tau)$ be a function that that determines the actual travel time on an edge $e \in E$, for mode $m \subseteq M_e$, at time τ.

The function $\mathbf{T}$ is impacted by the actions taken by all agents since that changes the state of the graph and often can be modeled as a latency function [18]. It can be learned from historical states of the network $G_0, G_1, ..., G_{\tau-1}$ (from *Memory*) and the current state of the network G_τ, to get expected travel times for time $\tau+1, \tau+2, \cdots, \tau+f$, where f is the number of time intervals in future. To differentiate this from the actual travel time function, we denote the learned travel time function as $\hat{\mathbf{T}}$. The accuracy of estimates from $\hat{\mathbf{T}}$ affects the agent's plans and thus, the network congestion.

4) Incentives: We consider *Incentives* as any intervention by the system to achieve some desired results, such as a decrease in network congestion, a decrease in pollution and so on. Such incentives are often monetary in nature and have an immediate and direct effect on the user because these financial incentives decrease the cost of a route. However, non-monetary incentives such as paid holidays, free coupons and so on, are difficult to design. They do not affect the immediate cost of the route and instead gives a delayed reward to the user. In this paper, we have only considered a financial incentive. Let $\mathbf{I}(\mathbf{e}, \mathbf{m}, \tau)$ be the function that determines the actual value of incentive, where $e \in E$ is an edge in network G, $m \subseteq M$ is the mode used on edge e and τ is the time at which edge e is traversed. The function $\mathbf{I}$ is an intervention by the system and hence it does not change during the simulation. It is used to reduce the cost disutility mentioned later in this section.

5) Cost Disutility: There is the cost (monetary) associated with the graph G_τ, such as transit fare, toll costs, and so on. Let $\mathbf{C}(\mathbf{e}, \mathbf{m}, \tau)$ be a function that that determines the cost (monetary) incurred for traversing on an edge $e \in E$, for mode $m \subseteq M$, at time τ. The function $\mathbf{C}$ is static–i.e does not change during the simulation. It is also not learnable since this function is set by different agencies that provide some services in the transportation network.

6) State: State $\mathbf{S}_\tau$ of the Environment at time τ, is a set of environment variables at time τ, that is shared with all the agents in the system. Formally, it can be defined as the tuple $(\mathbf{G}_\tau, \mathbf{I}, \mathbf{T}, \mathbf{C}, \hat{\mathbf{T}})$, where G_τ is the state of the network G at time τ, I is the incentive function set by the system, T is the actual travel time function , C is the actual cost function and $\hat{T}$ is travel time estimator as discussed earlier in this section.

7) Trigger Model: This model is the primary entry point in the agent's decision-making process. The time at which an agent makes a new decision is crucial in route planning. If an agent re-evaluates it's planning at every intersection, then it is following the best possible plan at every turn. While, if an agent does not re-evaluate it's route plan during the trip, then it is not considering real-time network changes such as congestion, transit delay, etc. and hence, may not be following the best possible route to reach their destination.

Formally, we can realize this model as a function $\mathbf{TR}(\mathbf{S}_\tau, \mathbf{R}, \tau)$, where S_τ is the current state given by the environment and R is the current route plan followed by an agent and τ is the time at which the agent has to make a decision. Some agents change their original plans more frequently than others. The function $\mathbf{TR}$ can also be learned from historical trips of an agent.

8) Router: The goal of the *Router* is to get minimum cost route based on timed-query (s, d, m, τ) of an agent, where $s \in V$ is the source, $d \in V$ is the destination, $m \subseteq M$ are set of mode choices and τ is the departure time or arrival time of the trip. A time-dependent shortest path routing algorithm then computes a feasible minimum cost path R, from s to d when departing from s at time τ. It uses $\hat{T}$ to evaluate expected travel time in future states and cost function C to evaluate the cost of any potential path.

State of the art graph-based routing algorithms has been shown to be effective in real-time route planning in time-dependent networks [19]. In this paper, we have used a time-dependent variant of Dijkstra algorithm [20] called *ALT* [21], [22], which improves upon basic A* search algorithm [23] by introducing landmarks heuristics.

9) Mode Choice Model: The new route plan R_{new} returned by *Router* gives an agent information on next mode that should be followed by an agent. However, an agent might have some bias or preference over certain modes in a route. We define function $\mathbf{MC}(\mathbf{R}_{\text{curr}}, \mathbf{R}_{\text{new}}, \tau)$, which uses the route R_{curr} currently being followed by agent from *Plan memory*, new route R_{new} given by *Router*, and time τ, to give probability value of choosing the next mode that should be followed by an agent as per new route R_{new}. This function can be learned from historical route plans followed by an agent.

10) Route Choice Model: We discussed earlier that *Router* provides new route plan R_{new} to agent. We also discussed that the $\mathbf{MC}$ model gives the probability of choosing the mode in new route R_{new}. However, even if a route is present, it may not be feasible for the agent—that is, the feasibility of a route in the context of travel mode choice may depend on economic constraints, altruism level and characteristics of the agent. The attractiveness of travel alternatives is often expressed via a utility function [24]. The analogous concept of *disutility* has been used in transportation economics to evaluate the displacement of time associated with the choice of different transportation modes or routes [25]. Savings in travel time, by definition, reduce the disutility associated with the total time of displacement. Monetary cost, travel time, number of transfers, calories burned, safety, etc. are some common disutilities for a commuter [24], [25].

We have used the following factors to capture *disutility* experienced by an agent when it follows route R:

- **Expected Travel time:** Since the route R_{new} given by the *Router* is the route that should be followed by the agent in future time intervals, we need $\hat{\mathbf{T}}$ which gives the expected travel times in future for each edge e in the network. Let $\mathbf{ET}(\mathbf{R})$ be the function that evaluates the expected travel time for traversing route R and which is given by
$$ET(R) = \sum_{(e,m,\tau)\in R} \hat{T}(e,m,\tau). \tag{1}$$
- **Expected Cost:** The cost of a route R is not only determined by the cost function $\mathbf{C}$ (which is set by the agencies for using various services in the transportation network), but also by the incentive function $\mathbf{I}$, an intervention by the system to reduce the cost. Let $\mathbf{EC}(\mathbf{R})$ be the function that evaluates the expected cost for traversing route R, where
$$EC(R) = \sum_{(e,m,\tau)\in R} C(e,m,\tau) - I(e,m,\tau). \tag{2}$$
- **Expected Social Disutility:** Studies [15], [16] from Routing game literature, shows that some agents are partially willing to suffer (take on additional cost) for the good of society. This is also known as *altruism*. To capture the altruism level of an agent, we need to capture expected social disutility of the system during the time interval when the agent will follow its route plan R. Here, expected social disutility is the total expected travel time considering all the edges in the network graph. Let $\mathbf{ESD}(\mathbf{R})$ be the function that evaluates the expected Social Disutility for traversing route R, where
$$ESD(R) = \sum_{e\in E, m\in M, \tau\in R} \hat{T}(e,m,\tau). \tag{3}$$

We now define the discrete route choice model $\mathbf{RC}$ for an agent, as a function of following parameters: (1) $ET(R_{\text{curr}})$, (2) $EC(R_{\text{curr}})$, (3) $ESD(R_{\text{curr}})$, (4) $ET(R_{\text{new}})$, (5) $EC(R_{\text{new}})$, (6) $ESD(R_{\text{new}})$, (7) τ, where ET, EC, ESD are the functions defined earlier that evaluates a route R, R_{curr} is the current route present in *Plan memory*, R_{new} is the new route given by *Router*, τ is the time interval at which an agent is making a decision.

C. Simulation Workflow

We now discuss the typical simulation steps. The first step described later using the case study scenario is to learn the agent's route choice model, trigger model and the mode choice models from the transportation data collected by a city. Once these models have been learned (e.g., as a neural network), they are ready for simulation. Then, the following steps are performed.

1) Initialization: The first step is to initialize all the variables needed by the *Environment* and the agents in the system. Specifically the *Environment* is seeded with initial variables $\mathbf{G_0}, \mathbf{I}, \mathbf{C}, \mathbf{A}, \hat{\mathbf{T}}$ at time $\tau = 0$, as discussed in Section III-B. We do not need to initialize the actual travel time function $\mathbf{T}$ since it will be updated by the actuator function $\mathbf{A}$ as agents start taking actions in the network. Each agent i is seeded with initial route plan R_i given by its router based on its timed-query request (s_i, d_i, m_i, τ_i). Each agent is also provided with trained models $\mathbf{TR}, \mathbf{MC}, \mathbf{RC}$ as discussed in Section III-B.

2) Run Simulation with Mobilytics-Gym: After the environment variables are initialized and agents are seeded with their initial plans, the simulation runs in discrete time steps. In each time step τ, the environment gives state S_τ to all agents in system and gets actions $\{a_1, a_2, \cdots, a_n\}$ from all the agents, which is then used by actuator function A discussed in Section III-B6 to evolve the state of graph from G_τ to $G_{\tau+1}$. Then, the environment gives new state $S_{\tau+1}$ to all agents and the process continues until all the agents run out of their plans.

3) Evaluate System Impact: Finally, after the end of the simulation, we get the actual route R_i^* followed by each agent i in the simulation. We use these routes to calculate the following metrics for evaluating system impact:

1) **Mean Excess Travel time (MET):** We use this metric to calculate extra travel time incurred due to route plan of all agents in simulation, to give an indication on congestion level. The mean excess travel time can be calculated by $MET(\mathcal{R}^*) = \frac{1}{|\mathcal{R}^*|}\sum_{q\in Q}\sum_{(e,m,\tau)\in R_q^*} T(e,m,\tau) - \phi_f(e,m,\tau)$ where $\mathcal{R}^* = \{R_q^*\}_{q\in Q}$ is the set of routes followed by the agent population Q and $|\mathcal{R}^*|$ denotes the cardinality of $\mathcal{R}^*$. Here, $T(e,m,\tau)$ is the actual travel time experienced by an agent, as discussed earlier and $\phi_f(e,m,\tau)$ is the free-flow travel time function for a given edge e, mode m and time interval τ.
2) **Incentives cost (IC):** The total cost incurred on the system, due to incentives, can be calculated by $IC(\mathcal{R}^*) = \frac{1}{|\mathcal{R}^*|}\sum_{q\in Q}\sum_{(e,m,\tau)\in R_q^*} I(e,m,\tau)$ where $\mathcal{R}^* = \{R_q^*\}_{q\in Q}$ is the set of routes followed by the population Q and $|\mathcal{R}^*|$ denotes the cardinality of $\mathcal{R}^*$. Here, $I(e,m,\tau)$ is the incentive function set by the system, as discussed earlier.

IV. Case Study of Nashville

We now discuss the use of Mobilytics-Gym on a case study for the Nashville metropolitan region. First, we discuss the data used to learn the models.

A. Dataset

a) Trip Data: We use anonymized employee trip distribution data localized to traffic analysis zones (TAZ)[2]. Further,

[2]TAZ is a special area delineated by state and/or local transportation officials for tabulating traffic-related data, especially journey-to-work and place-of-work statistics.

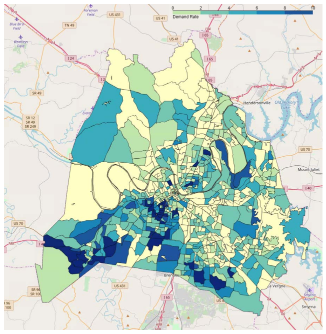

Fig. 2: Dataset over various TAZs. The colour scale increases with the density of population in a TAZ, and shows the number of trips to the University of Vanderbilt region.

we use the dataset from the United States Census [26] to model the relative traffic demand density from each TAZ back to the university area (see Fig. 2). Using these two datasets, we sample randomized daily morning trips to the university area.

b) Other Data: In addition to the trip data, we have access to static and real-time data from multiple data sources, such as the Nashville Metro Transportation Authority (MTA) for bus data, the Nashville Fire Department for accident data, the Dark Sky API[3] for weather data, and the HERE API [4] for traffic data, including speed and travel time data.

B. Configuration for Nashville

In each of the following subsections, we describe the components of Mobilytics-Gym for the Nashville simulation.

1) $\mathbf{G_0} = (\mathbf{V}, \mathbf{E}, \mathbf{M})_{\tau=\mathbf{0}}$*:* We use an OpenStreetMap (OSM) export of the Nashville metropolitan area [27] to create the road network. The OSM file contains information on allowed modes (M), length, geometry, and lane information, and other static properties of junctions (V) and links (E) in the road network. We use this to initialize the initial state of multi-modal network graph $G_0 = (V, E, M)_{\tau=0}$, as discussed in Section III-B1.

2) $\mathbf{A}(\mathbf{G_{\tau-1}}, \mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_n})$*:* We use the MATSim [5] simulator for our actuator function A. As noted in Section II, we use MATSim as a one-shot simulator to simulate the traffic dynamics and to test the viability and performance of different models that make up the system.

3) $\mathbf{C}(\mathbf{e}, \mathbf{m}, \tau)$*:* The disutility function C is defined in terms of transit fares presently in place in Nashville and given in the static general transit feed specification (GTFS) dataset [28]. On average, the transit fare is $\$1.5$. Since there is no cost incurred in walking, we kept the cost at $\$0$ for 'walk mode'. We calculate the cost of using a car based on the average cost of gas per gallon and average mileage per trip from following studies [29], [30].

[3]https://darksky.net/forecast/40.7127,-74.0059/us12/en

[4]https://www.here.com/en

TABLE II: Incentive profiles (in dollars) used in the case study. There are no incentives for using car. The incentive for walking is set on a per-mile basis while the incentive for using the bus is set on a per-ride basis.

Profile	Car $	Bus $	Walk $
IP-1	0	0	0
IP-2	0	0.25	1
IP-3	0	0.5	1.5
IP-4	0	1	2.5
IP-5	0	1.25	4
IP-6	0	1.5	6

TABLE III: Feature description for travel time prediction model.

Feature	Dim	Description
Hour of day, Day of week	2	Hour of the day and Day of week used to sample speed data
Length	1	Length of the street segment, collected from OpenStreetMap
Freespeed	1	Freespeed on the street segment, collected from HERE API
Number of lanes	1	Number of lanes on the street segment, collected from OpenStreetMap
TAZ	741	Binary indication of Traffic Analysis Zone (TAZ) corresponding to this feature vector.
Realtime speed	1	The true speed value collected from HERE API.

4) $\mathbf{I}(\mathbf{e}, \mathbf{m}, \tau)$*:* Incentive design is difficult in practice. Since the focus of this paper is in evaluating different agent models given an arbitrary incentive function, we leave the discussion on proper incentive design for future work. Giving incentives to commuters based on travel mode is a commonly used incentive mechanism by many cities in the United States [31]. Hence, we focus our experiments on this type of mechanism. Table II shows the incentive profiles used for the experiments in our paper. It starts with no incentives (for benchmark analysis) and increases for the bus and walk modes. We should note that the specific values used for the incentive profiles are chosen after conducting an extensive set of simulations.

5) $\hat{\mathbf{T}}(\mathbf{e}, \mathbf{m}, \tau)$*:* We use a data-driven approach to build the model $\hat{\mathbf{T}}$ for estimating travel times. We use historical traffic data collected via the HERE API from January 1 to January 31, 2018, for the Nashville metropolitan area. We build a feature set with the quantities described in Table III. All the features are numeric quantities.

The resulting feature space has 747 dimensions and the number of labeled data records is over 194 million, each of which represents a feature vector captured at a time point for a certain street segment. The training and testing data is min-max normalized before being used in the models.

6) Fixed Plan Agents: To create *Fixed Plan Agents*, we use GTFS data from Nashville MTA to create a set of fixed route plans $\mathbf{F}$ as discussed in Section III-A.

7) Dynamic Plan Agents: To create *Dynamic Plan Agents*, as discussed in Section III-A, we use real-world data from Nashville to specify origin-destination pairs, used modes, and origin zones based on the demand distribution data described

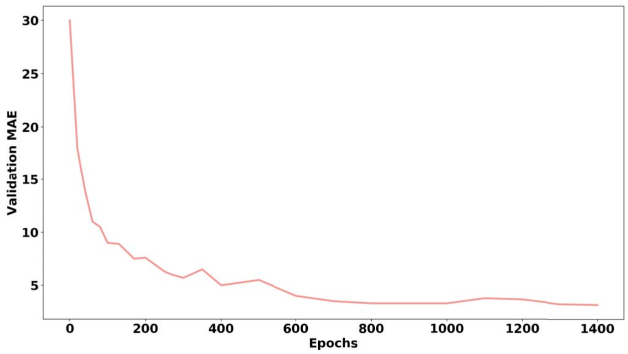

Fig. 3: MAE vs. Epoch curve during training of travel time prediction model.

earlier in this section. Note that trip start times are generated based on a uniform distribution in morning hours (i.e., 8AM-10AM) during this simulation. The total number of agents used for our case study is 4884.

8) Mode Choice Data: Since we do not have the mode choice data needed for training agent models, we generate this data in a simulated setting using MATSim. Specifically, we run a simulation where 20% of the agents prefer the bus and walk modes, 40% of the agents prefer the car mode, and 40% of the agents prefer the car or bus/walk mode selected uniformly at random. Since the agent population is already biased, the models will reflect this bias after being trained. The trip data collected from the agents in these simulations are then used by the model to train agent models.

9) Learning Travel Time Disutility Model ($\hat{T}$): We use a deep long short-term memory neural network (LSTM) [32], which is a recurrent neural network architecture that can capture the long-term temporal dependency for short-term travel speed prediction. Extensive tuning both in the configuration of hidden layers and the activation and optimization functions was done during training. The **Adam optimizer** [33] and **SGD** [34] are chosen as optimizers for the neural network. Early stopping criteria are employed to avoid overfitting. Fig. 3 shows the cost versus iteration curve during the training phase for this model.

Fig. 3 shows the change in validation Mean Absolute Error (MAE) with epoch steps during the training of the travel disutility model $\hat{\mathbf{T}}$. The test MAE was found to be $\mathbf{4.87}$. This is certainly not the best result compared to the start-of-the-art models for estimating traffic speed [35]. However, our focus in this paper is not on the model itself, but how the model is used in the simulation.

10) Learning Agent preference models: As discussed in Section III-B, the models used by *Dynamic Plan Agents* to make a decision needs to be learned. The agent model contains components that require training and in order to do so, agent historical trip data is needed. Different models need a different subset of overlapping features from the same trip data. We discuss the features needed by different agent models below.

TABLE IV: Hyper-parameter tuning of agent models.

Model	Hidden Layers	Activation Function	Optimizer
Trigger model	100,70,40, 20,10	tanh	SGD
Mode choice model	100,60,40,10	relu	Adam
Route choice model	200,170,100, 50,20,10	relu	Adam

1) **Trigger model** ($\mathbf{TR}$): For this model, we build a feature set containing following features: (a) One-hot encoded vector of *Source TAZ* and *Destination TAZ* in which source and destination vertices of the trip lies; (b) *Hour of day* and *Day of week* of the trip departure time; (c) *Mean Congestion* of outgoing links of source vertex; and (d) *Current mode* of the agent at source vertex. We use a deep feed-forward neural network (DNN) [36] for a binary classification that indicates whether a routing decision should be triggered or not given some inputs. For binary classification, the activation function used in the output layer is a sigmoid function [37].
2) **Mode choice model** ($\mathbf{MC}$): For this model, we build a feature set containing the following features: (a) One-hot encoded vector of *Source TAZ* and *Destination TAZ* in which source and destination vertices of the trip lies; (b) *Hour of day* and *Day of week* of the trip departure time; and, (c) *Current mode* of the agent at source vertex. We use a DNN [36], for a multiclass logistic regression to learn this model. For multi-class logistic regression, we used a softmax function for activation in the output layer.
3) **Route choice model** ($\mathbf{RC}$): For this model, we build a feature set containing following features of the current route the agent is following R_{curr} and the new route R_{new} evaluated by router: (a) One-hot encoded vector of *Source TAZ* and *Destination TAZ* in which source and destination vertices of the trip lies; (b) *Hour of day* and *Day of week* of the trip departure time; (c) Expected Travel time $ET(R_{\text{curr}}), ET(R_{\text{new}})$; (d) Expected cost $EC(R_{\text{curr}}), EC(R_{\text{new}})$; and, (e) Expected social disutility $ESD(R_{\text{curr}}), ESD(R_{\text{new}})$ for current and new route respectively. We use a DNN with a softmax activation function in the output layer to predict probabilities for routes R_{curr} and R_{new}.

Table IV shows the parameters used for each agent model after hyper-parameter tuning. Fig. 4 shows the loss vs epoch curve during the training phase for the $\mathbf{TR}$, $\mathbf{MC}$, and $\mathbf{RC}$ models. The loss was found to be $\mathbf{0.51}, \mathbf{0.46}, \mathbf{0.43}$ for $\mathbf{TR}$, $\mathbf{MC}$, and $\mathbf{RC}$ models respectively.

C. Evaluating different incentive policies

Fig. 5 shows MET and IC for all the agents in the simulation. The plot demonstrates that as the incentives provided for the bus and walk modes increases, the MET value decreases. This decrease in travel time is due to the fact that agents are incentivized to use the bus and walk more. At the same time, the IC increases steeply. For a modest decrease of 12min

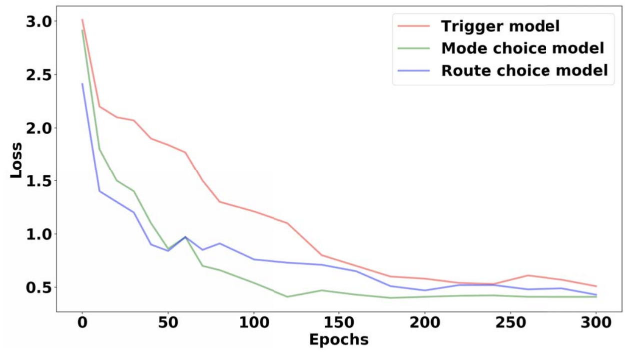

Fig. 4: Loss vs. Epoch curve during training of agent models.

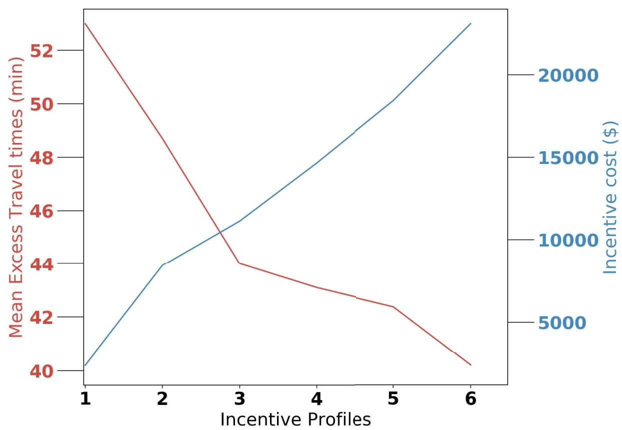

Fig. 5: System-level impact measured in terms of MET and incentives cost (IC) for different incentive profiles.

of MET, the IC value increases by approximately $20000. The higher amount of IC is due to following reasons: (1) incentives were designed in a *adhoc* manner without emphasis on the maximum budget set by the system or on appropriate incentive values for bus and walk modes, (2) Incentive values for walking are relatively steep compared to the bus. This is due to the fact that most Vanderbilt University employees do not live close to a bus stop, so higher incentives were needed to show any meaningful change.

Fig. 6 shows the mode distribution for different incentive profiles from Table II. The results imply that as more incentives are given to agents, the number of agents using transit increases, while at the same time number of agents using their personal car decreases. It should be noted that in this plot, agents are using multiple modes to reach their destination. The large increase in transit ridership is due to the fact that agents increasingly use the bus only in some legs of their entire trip. The result shows that only 23% agents using cars changed their modes when provided high incentives to walk and bus.

V. Conclusion

We presented a simulation framework that integrates decision-making models and provides a standardized environment to evaluate the efficacy of these models and incentive schemes in terms of their system-level impact. In support of the latter contribution, we included a case study from the Nashville metropolitan. Regarding future work, we aim to extend our current approach by: (a) including other modes of transportation such as car share, scooters, and rental vehicles; (b) using this simulation tool together with a more formal optimization framework for designing incentives; and, (c) designing models that can learn with causal inference—i.e., the models not only learn to predict accurately, but also understand the cause and effect of any prediction.

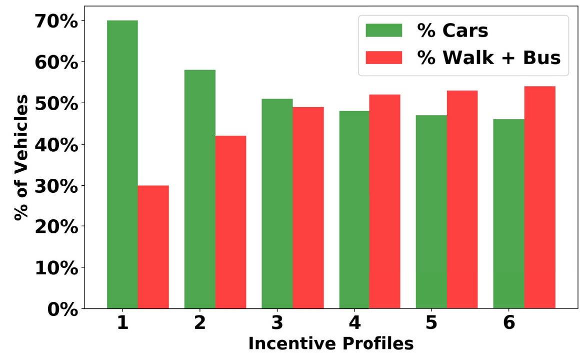

Fig. 6: Mode distribution plot for different incentive profiles.

Acknowledgment

This work is sponsored in part by the National Science Foundation under the award number CNS-1646912 (UW), CNS-1647015 (Vanderbilt), CNS-1528799 (Vanderbilt) and in part by the Vanderbilt Initiative in Smart City Operations and Research, a trans-institutional initiative funded by the Vanderbilt University.

References

[1] United States Census Bureau, "New American Community Survey Statistics for Income, Poverty and Health Insurance Available for States and Local Areas," 2018, [Online; accessed 8-March-2019]. [Online]. Available: {https://www.census.gov/newsroom/press-releases/2018/acs-1year.html}

[2] Tennessee Department of Transportation, "Transportation demand management program," https://www.tn.gov/tdot/article/transportation-demand-management-tdm, 2017.

[3] Ride Amigos, "What is Transportation Demand Management?" 2019, [Online; accessed 12-March-2019]. [Online]. Available: {https://rideamigos.com/transportation-demand-management-tdm/}

[4] United States Department of Transportation Federal Highway Administration, "Organizing and planning for operations," 2019, [Online; accessed 12-March-2019]. [Online]. Available: {https://ops.fhwa.dot.gov/plan4ops/trans_demand.htm}

[5] A. Horni, K. Nagel, and K. W. Axhausen, *The multi-agent transport simulation MATSim*. Ubiquity Press London, 2016.

[6] J. Barceló and J. Casas, "Dynamic network simulation with AIMSUN," in *Simulation Approaches in Transportation Analysis*. Springer, 2005, pp. 57–98.

[7] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (Simulation of Urban MObility)-an open-source traffic simulation," in *Proceedings of the 4th Middle East Symposium on Simulation and Modelling*, 2002, pp. 183–187.

[8] M. Balmer, K. Nagel, and B. Raney, "Large-scale multi-agent simulations for transportation applications," in *Journal of Intelligent Transportation Systems*, vol. 8, no. 4, 2004, pp. 205–221.

[9] L. Zhang and D. Levinson, "Agent-based approach to travel demand modeling: Exploratory analysis," *Transportation Research Record*, no. 1898, pp. 28–36, 2004.
[10] X. Chen, M. Yin, M. Song, L. Zhang, and M. Li, "Social Welfare Maximization of Multimodal Transportation: Theory, Metamodel, and Application to Tianjin Ecocity, China," *Transportation Research Record*, vol. 2451, no. 1, pp. 36–49, 2014.
[11] D. Levinson, "Identifying winners and losers in transportation," *Transportation Research Record*, no. 1812, pp. 179–185, 2002.
[12] Y. Li, L. Yang, H. Shen, and Z. Wu, "Modeling intra-destination travel behavior of tourists through spatio-temporal analysis," *Journal of destination marketing & management*, vol. 11, pp. 260–269, 2019.
[13] E. Martin, S. Shaheen, and I. Zohdy, "Understanding travel behavior: Research scan," 2016.
[14] K. Zhang and S. Batterman, "Air pollution and health risks due to vehicle traffic," *Science of the Total Environment*, vol. 450, pp. 307–316, 2013.
[15] P.-A. Chen and D. Kempe, "Altruism, selfishness, and spite in traffic routing," in *Proceedings of the 9th ACM conference on Electronic Commerce*, 2008, pp. 140–149.
[16] M. Hoefer and A. Skopalik, "Altruism in atomic congestion games," in *European Symposium on Algorithms*. Springer, 2009, pp. 179–189.
[17] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT Press, 2018.
[18] T. A. Manual, "Bureau of public roads," *US Department of Commerce*, 1964.
[19] T. Pajor, "Multi-modal route planning," *Universität Karlsruhe*, 2009.
[20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[21] A. V. Goldberg and C. Harrelson, "Computing the shortest path: A search meets graph theory," in *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2005, pp. 156–165.
[22] A. V. Goldberg and R. F. F. Werneck, "Computing point-to-point shortest paths from external memory," in *Proceedings of the Seventh Workshop on Algorithm Engineering and Experiments and the Second Workshop on Analytic Algorithmics and Combinatorics (ALENEX/ANALCO)*, 2005, pp. 26–40.
[23] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.
[24] A. L. Bazzan and F. Klügl, "Introduction to intelligent systems in traffic and transportation," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 7, no. 3, pp. 1–137, 2013.
[25] H. Raymundo and J. Reis, "Measuring passenger transport quality by disutilities," in *Proceedings of the 6th International Conference in Information, Systems, Logistics and Supply Chain Conference*, 2016.
[26] United States Census Bureau, "QuickFacts, Davidson County, Tennessee; UNITED STATES," 2016, accessed: 2017-12-02. [Online]. Available: {https://www.census.gov/quickfacts/fact/table/davidsoncountytennessee,US/PST045216}
[27] OpenStreetMap, "Openstreetmap," 2018. [Online]. Available: {https://www.openstreetmap.org}
[28] Google, "Gtfs static overview," 2018, [Online; accessed 8-March-2018]. [Online]. Available: {https://developers.google.com/transit/gtfs/}
[29] E. Blumenberg and G. Pierce, "Automobile ownership and travel by the poor: Evidence from the 2009 national household travel survey," *Transportation Research Record: Journal of the Transportation Research Board*, no. 2320, pp. 28–36, 2012.
[30] G. Barnes and P. Langworthy, "The per-mile costs of operating automobiles and trucks," 2003.
[31] P. C. Bueno, J. Gomez, J. R. Peters, and J. M. Vassallo, "Understanding the effects of transit benefits on employees travel behavior: Evidence from the new york-new jersey region," *Transportation Research Part A: Policy and Practice*, vol. 99, pp. 1–13, 2017.
[32] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with LSTM," in *Proceedings of the Ninth International Conference on Artificial Neural Networks*, 1999.
[33] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
[34] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*. Springer, 2010, pp. 177–186.
[35] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A comprehensive survey on graph neural networks," *arXiv preprint arXiv:1901.00596*, 2019.
[36] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT Press, 2016, vol. 1.
[37] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.