arXiv:2608.08035v1 [cs.MA] 8 Aug 2026

Decentralized Multi-Agent Urban Traffic Management via Spatio-Temporal Mobility Profile Planning

Lorenzo Mario Amorosa[1] (Member, IEEE), Lorenzo Farina[1] (Student Member, IEEE), Vittorio Todisco[1] (Member, IEEE), and Alessandro Bazzi[1] (Senior Member, IEEE)

[1]Department of Electrical, Electronic and Information Engineering (DEI), "Guglielmo Marconi", University of Bologna & WiLab - National Wireless Communications Laboratory of CNIT, Bologna, Italy.

Corresponding author: Lorenzo Mario Amorosa (email: lorenzomario.amorosa@unibo.it).

ABSTRACT As modern cities face increasingly severe traffic congestion, connected and autonomous vehicles (CAVs) have emerged as a crucial enabling technology for next-generation intelligent traffic management. However, fully realizing this potential is hindered by the limitations of current paradigms. Existing approaches typically optimize localized interactions rather than system-wide efficiency, incur severe communication overhead, or lack the deterministic guarantees required for safe kinematic execution. Furthermore, current multi-agent adaptations are frequently restricted to small predefined scenarios, failing to scale across large and complex urban networks. To bridge this gap, this paper introduces VeloCity, a decentralized multi-agent spatio-temporal mobility profile planning framework designed for CAVs operating in arbitrary urban areas. To minimize vehicles' travel times, VeloCity distributes mobility profile optimization directly to individual CAVs. Vehicles query a localized traffic coordinator for a reservation table, independently compute their fastest conflict-free mobility profile, and reserve their requested space-time slots back with the coordinator. By natively adapting to any arbitrary road topology, the framework manages highly irregular urban areas without requiring scenario-specific tuning, all while guaranteeing collision-free and physically executable vehicle trajectories. Extensive simulations across four large-scale real-world urban maps (Tokyo, Manhattan, Rome, and Bologna) demonstrate the framework's scalability. Compared to established state-of-the-art models, VeloCity yields drastically lower travel times, tightly bounds delay variance, and successfully prevents congestion gridlocks even under extremely high vehicular densities.

INDEX TERMS Connected and Autonomous Vehicles (CAVs), Multi-Agent Planning (MAP), Decentralized Traffic Management, Autonomous Intersection Management (AIM), Edge Computing, Urban Mobility.

I. INTRODUCTION

As urban traffic complexity grows, traditional traffic management paradigms are facing unprecedented scalability challenges [1]. To overcome these limitations, connected and autonomous vehicles (CAVs) have emerged as a crucial enabling technology for next-generation intelligent transportation. Conventional traffic control systems, such as fixed or adaptive traffic signals, often fail to exploit the fine-grained coordination and communication capabilities enabled by emerging vehicle-to-infrastructure (V2I) networks [2]. To mitigate these inefficiencies, advanced car-following models driven by both local and collective perception have been proposed to significantly improve traditional traffic flow management [3]. However, because they predominantly optimize for localized interactions rather than system-wide efficiency, their scalability remains constrained. Consequently, as urban centers grow more complex, there is a clear need for flexible and decentralized traffic planning frameworks capable of managing high-density vehicle coordination across arbitrary road topologies without relying on rigid scenario-specific control infrastructure. Multi-agent planning (MAP) offers a promising formulation for this challenge, treating each vehicle as an intelligent agent tasked with finding collision-free paths from a start to a target location [4]. However, existing traffic-oriented MAP adaptations often constrain themselves to small and predefined scenarios, such

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

as single intersections or simple merging lanes, failing to generalize to arbitrary map areas featuring diverse topological elements like roundabouts and multi-leg intersections [5, 6].

The practical deployment of advanced CAV coordination relies intrinsically on accurate environment perception and the reliable execution of planned mobility profiles over vehicle-to-everything (V2X) networks. Recently, standardization organizations such as the European telecommunications standards institute (ETSI) and the society of automotive engineering (SAE) are actively establishing the necessary foundations to make this feasible. Building upon well-established cooperative awareness frameworks validated through both simulations [7] and field testing [8], recent standardization efforts have expanded toward enabling complex maneuver coordination. Specifically, SAE has formalized the Maneuver Sharing and Coordinating Service in its J3186 and J3216 standards [9, 10]. Similarly, ETSI is detailing analogous specifications for maneuver coordination [11, 12]. These specifications dictate the precise messages and communication protocols required for vehicles to safely negotiate maneuvers in shared spaces, such as intersections. Importantly, these standards specify *how* vehicles exchange information for maneuver coordination, but they do not prescribe specific conflict-resolution or traffic-management algorithms. This decoupling creates a critical gap that must be filled by application developers, directly motivating the need for robust and decentralized planners capable of supplying the optimized maneuver strategies that these protocols will carry.

To bridge this gap, this paper introduces *VeloCity*, a decentralized multi-agent spatio-temporal mobility profile planning framework for CAVs operating in arbitrary urban areas. In our framework, each map area is equipped with a local controller that maintains a spatio-temporal reservation table for its respective road segments. Crucially, instead of the controller solving a joint optimization problem for all agents, mobility profile optimization is distributed across the vehicles. Upon approaching a managed area, a vehicle queries the controller for the current reservation table, locally computes its fastest conflict-free mobility profile, and subsequently reserves its requested spatio-temporal slots back at the controller. This decentralized MAP approach aims at accommodating arbitrary road geometry, including complex intersections and roundabouts, without requiring scenario-specific tuning. Specifically, the main contributions of this paper are summarized as follows:

- We propose *VeloCity*, a decentralized MAP framework designed for urban traffic management. By decomposing the mobility optimization problem into sequential spatio-temporal planning tasks to be executed at each CAV, the framework guarantees collision-free and physically executable vehicle trajectories. Its flexible space-time reservation architecture intrinsically supports arbitrary and heterogeneous road topologies without requiring scenario-specific tuning.
- We conduct an extensive performance evaluation utilizing the Simulation of Urban MObility (SUMO) simulator [13] across four diverse, large-scale urban environments imported directly from real-world OpenStreetMap[1] data (Tokyo, Manhattan, Rome, and Bologna). Through extensive experiments, we demonstrate that VeloCity significantly outperforms established baselines. The proposed approach yields drastically lower travel times, tightly bounded delay variance, and robustness against congestion gridlocks even under very high vehicular densities.
- To ensure reproducibility and support the broader research community, we release the complete open-source implementation of VeloCity. The repository, which includes the core algorithms, the four large-scale urban scenarios, and all simulation scripts, is publicly available at https://github.com/V2Xgithub/VeloCity.

To the best of our knowledge, VeloCity represents the first V2X-based framework to successfully scale decentralized and deterministic mobility profile optimization to arbitrary city-wide road networks.

The remainder of this paper is organized as follows. Section II reviews the relevant literature. Section III defines the system model and formulates the multi-agent mobility profile optimization problem. Section IV details the proposed VeloCity methodology. Section V presents the experimental setup and discusses the numerical results. Finally, Section VI summarizes the concluding remarks and outlines potential directions for future work.

II. RELATED WORK

The management of traffic flows in urban road networks has long been a critical challenge for transportation engineering. For decades, the primary mechanism for resolving vehicle conflicts at junctions has been the traditional traffic light, with researchers focusing heavily on phase optimization. While early traffic signal control systems utilized offline static plans or heuristic adaptive strategies like SCOOT and SCATS [14, 15], modern approaches have embraced predictive control, such as model predictive control (MPC) [16, 17], schedule-driven methods [18, 19], and max-pressure control [20, 21]. More recently, a hybrid architecture exploiting both MPC and reinforcement learning (RL) [22] and multi-agent reinforcement learning (MARL) frameworks have also been proposed to manage local networks [23, 24]. However, regardless of the underlying mathematical formulation, these solutions remain fundamentally constrained by the traffic-light paradigm and therefore cannot fully exploit the coordination capabilities enabled by advances in vehicle automation and V2X connectivity.

[1] See: https://www.openstreetmap.org/

Recently, CAVs encouraged a paradigm shift toward autonomous intersection management (AIM), where replacing traffic lights with V2X-based maneuver coordination and dynamic space-time reservations can eliminate stop-and-go delays and drastically increase intersection capacity [25]. Numerous AIM protocols have emerged [26, 27]. Early decentralized architectures functioned as virtual traffic lights, managing crossing orders via sequential vehicle-to-vehicle (V2V) priority passing [28], clustering [29], directed collision graphs [30], or RL [31]. While successful at conflict sequencing, these methods do not control kinematic profiles to prevent stop-and-go maneuvers. To fully exploit AIM, subsequent research combined scheduling with precise trajectory optimization. For instance, centralized MPC and independent energy-minimization controllers were developed to calculate stop-free acceleration profiles based on space-time reservations [4, 32, 33]. A specific subset of this literature bridges these models with real-world communication constraints, such as formulating protocols for packet loss and 4G/5G latency [2, 34, 35]. Yet, while these single-intersection frameworks successfully optimize local vehicle trajectories, they do not address how such solutions scale to larger arbitrary road networks.

Translating these approaches to large-scale urban grids introduces severe computational and communication bottlenecks. To date, efforts to extend AIM typically link a limited number of adjacent managers rather than addressing city-scale grids. For instance, expanding reservation architectures via time-based A* searches can suffer from information staleness, as vehicles route themselves using instantaneously outdated congestion data [36]. To address this, dynamic market-inspired auctions [37] and consensus protocols [38] have been proposed; however, both introduce severe communication overhead when scaled to large road scenarios. To explicitly consider time-evolution and precise kinematics, [39] reformulated routing as a time-expanded network flow problem, though its reliance on integer linear programming (ILP) necessitates a fully centralized and computationally heavy architecture. Recently, alternative solutions have turned to MARL by modeling road segments or vehicles as intelligent agents [40, 41]. Beyond reintroducing significant communication overhead, these MARL policies remain fundamentally reactive and rely on opaque neural networks, precluding the exact determinism and formal kinematic safety guarantees required for autonomous traffic coordination.

Consequently, a significant gap remains for a truly scalable AIM architecture capable of handling high-density arbitrary urban networks. The literature currently lacks an adaptive framework that resolves coordination through purely decentralized and physically executable optimizations, which is a gap that VeloCity aims to fill.

III. SYSTEM MODEL AND PROBLEM FORMULATION

We consider an urban area managed by a traffic coordinator, including a set of CAVs[2] denoted by $\mathcal{V} = \{1, 2, \ldots, N\}$. Each vehicle $i \in \mathcal{V}$ enters the area at a specific requested time $t_{\text{in},i}$ with a predetermined[3] trajectory $\mathcal{P}_i$ and initial speed $v_{\text{in},i}$. The objective of each CAV is to optimize its mobility profile, i.e., its speed and acceleration over time, to drive and reach the intended destination minimizing its travel time. A CAV's trajectory $\mathcal{P}_i$ is represented by a one-dimensional path of total physical length D_i, where the placement of vehicle i at time step t is defined by $s_i(t) \in [0, D_i]$. This 1D path maps directly to the two-dimensional Euclidean road plane[4] via a network geometry function $\gamma_i : [0, D_i] \to \mathbb{R}^2$, which defines the spatial coordinates of the vehicle's reference center. To account for physical dimensions, vehicle i is characterized by a two-dimensional bounding area $\mathcal{F}_i(s_i(t)) \subset \mathbb{R}^2$. This footprint represents the physical space occupied by the vehicle on the map at time step t, constructed by sweeping its length L_i and width W_i around the reference center $\gamma_i(s_i(t))$. Importantly, each $\mathcal{P}_i$ includes routing, comprising specific vehicle's lane assignments and intersection crossings; however, it does not specify the vehicle's mobility profile, leaving its speed and acceleration over time to be determined.

A. VEHICLE'S MOBILITY PROFILE

Each vehicle $i \in \mathcal{V}$ must determine its mobility profile over time, governed by its speed $v_i(t)$ and acceleration $a_i(t)$. To ensure physically executable and safe trajectories, these control variables are strictly bounded by both vehicle-inherent kinematic capabilities and area-specific speed regulations. First, the acceleration is limited by the vehicle's mechanical capabilities:

$$a_{\min,i} \leq a_i(t) \leq a_{\max,i} \tag{1}$$

where $a_{\min,i}$ and $a_{\max,i}$ represent the maximum deceleration and acceleration constants for vehicle i, respectively. Second, the maximum permissible speed depends not only on the vehicle's top performance but also on the localized speed limits of the specific road segment or lane it currently occupies. Let $V_{\max}(\mathbf{p}) : \mathbb{R}^2 \to \mathbb{R}^+$ be a function mapping a 2D coordinate to the local legal speed limit. The speed profile is thus constrained by:

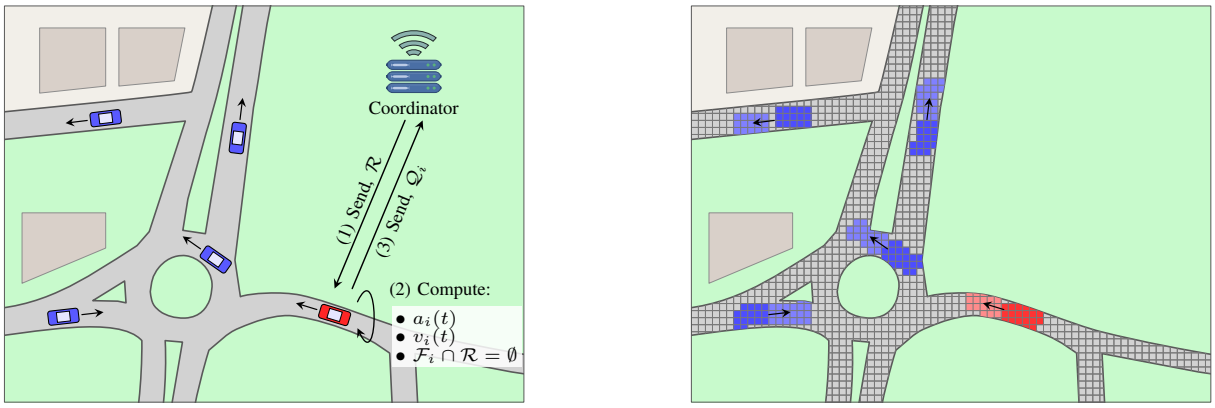
$$0 \leq v_i(t) \leq \min\left(V_{\max,i},\, V_{\max}\big(\gamma_i(s_i(t))\big)\right) \tag{2}$$

where $V_{\max,i}$ is the maximum structural speed of vehicle i. By evaluating $V_{\max}$ at the vehicle's instantaneous 2D

[2]We use the terms CAV and agent interchangeably.

[3]VeloCity focuses on mobility profile optimization given a fixed path; however, it remains fully compatible with proactive dynamic routing strategies. In our simulations, vehicle routing is managed autonomously by SUMO.

[4]For multi-level infrastructures, such as bridges or overpasses, vertically overlapping road segments are mapped to logically distinct two-dimensional planes to prevent false spatial conflicts.



(a) (b)

FIGURE 1: (a) Considered urban area, where the red vehicle i approaches the coordinated area: (1) the coordinator provides the reservation table $\mathcal{R}$ to i, (2) i locally computes its feasible mobility profile, and (3) i transmits its reservation request $\mathcal{Q}_i$ to the coordinator. (b) Discrete spatial representation of the reservation table $\mathcal{R}(t)$. The road network is mapped to a spatial grid, where solid colored cells represent the vehicles' physical footprints $\mathcal{F}_i(s_i(t))$ and lighter cells indicate their next intended spatial reservations for a $t' > t$.

coordinate $\gamma_i(s_i(t))$, this formulation dynamically enforces area-specific speed constraints, such as speed reductions at sharp turns, roundabouts, or intersection approaches.

B. TRAFFIC COORDINATOR

The coordination of CAVs within the managed urban area is supervised by a localized traffic infrastructure coordinator. This coordinator does not solve a joint centralized problem for determining CAVs profiles. Instead, it serves as a reference point to maintain a spatio-temporal reservation table, denoted by $\mathcal{R}$, while CAVs optimize their profiles in a decentralized way. The reservation table $\mathcal{R}(t)$ is discretized in time with a resolution Δt and maps the two-dimensional road network into a spatial grid with a resolution Δs. At any discrete time step t, $\mathcal{R}(t) \subset \mathbb{R}^2$ represents the set of all spatial coordinates reserved by authorized agents. When a vehicle i approaches the managed area, it queries the coordinator to retrieve the current state of $\mathcal{R}$. Given $\mathcal{R}(t)$, vehicle i locally computes a feasible mobility profile by ensuring its intended footprint sequence $\mathcal{Q}_i = \{\mathcal{F}_i(s_i(t)),\ \forall t \geq t_{\text{in},i}\}$ does not conflict with existing reservations:

$$\mathcal{F}_i(s_i(t)) \cap \mathcal{R}(t) = \emptyset, \quad \forall t \geq t_{\text{in},i}\,. \tag{3}$$

Upon computing this optimal and feasible profile, the vehicle submits its footprint sequence to the coordinator. The coordinator then commits the sequence and updates the reservation table by appending the vehicle's footprint: $\mathcal{R}(t) \leftarrow \mathcal{R}(t) \cup \mathcal{Q}_i$, where $\mathcal{Q}_i$ corresponds to a reservation request, comprising the set of spatial cells occupied by vehicle i over time. A depiction of the coordinated reservation is presented in Fig. 1.

C. PROBLEM FORMULATION

The primary objective of the traffic management framework is to optimize the overall traffic efficiency across the managed urban area while guaranteeing collision-free navigation. We frame this task as a multi-agent optimization problem aimed at minimizing the total travel time of all vehicles traversing the network. To formalize this objective, let $t_{\text{out},i}$ denote the arrival time step at which vehicle $i \in \mathcal{V}$ successfully reaches the end of its intended path $\mathcal{P}_i$, satisfying $s_i(t_{\text{out},i}) \geq D_i$. Given trajectory $\mathcal{P}_i$ and initial speed $v_{\text{in},i}$ for all vehicles $i \in \mathcal{V}$, the optimization problem can be formulated as a mixed-integer non-linear program (MINLP), where the decision variables are $\mathbf{a} = \{a_i(t)\}_{\forall i \in \mathcal{V}}$:

$$\min_{\mathbf{a}} \sum_{i \in \mathcal{V}} (t_{\text{out},i} - t_{\text{in},i}) \tag{4}$$

subject to the (1), (2), and (3) constraints for all $i \in \mathcal{V}$ and $t \geq t_{\text{in},i}$. In this formulation, (4) minimizes the aggregate system-wide delay, representing the total time elapsed between each vehicle's requested entry time $t_{\text{in},i}$ and its final destination arrival $t_{\text{out},i}$. The optimization problem defined by (4) is non-convex and highly complex, hence computing a global joint solution simultaneously would not be practical. This computational challenge motivates the need for a decentralized solution, which we propose in Section IV.

IV. THE VELOCITY FRAMEWORK

To resolve the global coordination problem formulated in Section III.C, we propose VeloCity: a framework that decomposes the multi-agent optimization problem in (4) into a sequence of distributed spatio-temporal planning tasks. The core principle in VeloCity is to let each vehicle independently compute its own feasible mobility profile while coordinating

Algorithm 1 Coordinator Loop

Require: Reservation table $\mathcal{R} \leftarrow \emptyset$

for each vehicle i entering the coordination area at $t_{\text{in},i}$, with speed $v_{\text{in},i}$, trajectory $\mathcal{P}_i$, and dynamics bound $a_{\min,i}, a_{\max,i}, V_{\max,i}$ **do**
  Retrieve current reservation table $\mathcal{R}$
  $\mathcal{Q}_i \leftarrow \textsc{LocalPlan}(\mathcal{R}, t_{\text{in},i}, v_{\text{in},i}, \mathcal{P}_i, a_{\min,i}, a_{\max,i}, V_{\max,i})$
  $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{Q}_i$ ▷ Commit trajectory
end for

Algorithm 2 LocalPlan

Require: Reservation table $\mathcal{R}$, entry time $t_{\text{in},i}$, speed $v_{\text{in},i}$, trajectory $\mathcal{P}_i$, and dynamics bounds $a_{\min,i}, a_{\max,i}, V_{\max,i}$

Ensure: Reservation request $\mathcal{Q}_i$

$\mathbf{x}_i^0 \leftarrow (s_i^0, v_i^0, t_{\text{in},i})$
$g(\mathbf{x}_i^0) \leftarrow 0$
$f(\mathbf{x}_i^0) \leftarrow h(\mathbf{x}_i^0)$ ▷ Eq. (18)
Open $\leftarrow \{\mathbf{x}_i^0\}$ ▷ priority queue ordered by $f(\cdot)$
Closed $\leftarrow \emptyset$
while Open $\neq \emptyset$ **do**
  $\mathbf{x}_i(t) \leftarrow \arg\min_{\mathbf{x} \in \text{Open}} f(\mathbf{x})$
  remove $\mathbf{x}_i(t)$ from Open, add to Closed
  if $s_i(t) \geq D_i$ **then** ▷ goal test
    $\Pi_i \leftarrow$ backtrack path to $\mathbf{x}_i^0$
    Convert Π_i into reservation request $\mathcal{Q}_i = \{(\mathcal{F}_i^k, t_k)\}_{k=0}^{M_i}$
    return $\mathcal{Q}_i$
  end if
  for each admissible $a_i(t) \in [a_{\min,i}, a_{\max,i}]$ **do**
    $v_i(t+\Delta t) \leftarrow v_i(t) + a_i(t)\Delta t$ ▷ Eq. (11)
    if $0 \leq v_i(t+\Delta t) \leq \min(V_{\max,i}, V_{\max}(\gamma_i(s_i(t))))$ **then**
      $s_i(t+\Delta t) \leftarrow s_i(t) + v_i(t+\Delta t)\Delta t$ ▷ Eq. (12)
      $\mathbf{x}_i(t+\Delta t) \leftarrow (s_i(t+\Delta t), v_i(t+\Delta t), t+\Delta t)$
      if $\mathcal{F}_i(\mathbf{x}_i(t+\Delta t)) \cap \mathcal{R}(t+\Delta t) = \emptyset$ **then** ▷ Eq. (14)
        if $\mathbf{x}_i(t+\Delta t) \notin$ Closed **then**
          $g' \leftarrow g(\mathbf{x}_i(t)) + \Delta t$
          if $\mathbf{x}_i(t+\Delta t) \notin$ Open **or** $g' < g(\mathbf{x}_i(t+\Delta t))$ **then**
            $g(\mathbf{x}_i(t+\Delta t)) \leftarrow g'$
            $f(\mathbf{x}_i(t+\Delta t)) \leftarrow g' + h(\mathbf{x}_i(t+\Delta t))$
            parent$(\mathbf{x}_i(t+\Delta t)) \leftarrow \mathbf{x}_i(t)$ ▷ to compute Π_i
            insert/update $\mathbf{x}_i(t+\Delta t)$ in Open
          end if
        end if
      end if
    end if
  end for
end while
return INFEASIBLE

only through a shared spatio-temporal reservation table $\mathcal{R}$ maintained by the traffic coordinator. The remainder of the section detail the algorithms, which are presented in Algorithm 1 and Algorithm 2.

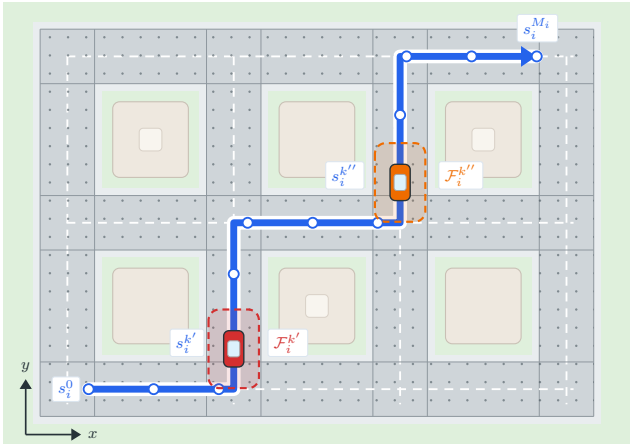


FIGURE 2: Discrete spatio-temporal representation for vehicle i's path. The reference positions $s_i^{k'}$ and $s_i^{k''}$ illustrate two discrete positions along the path, with corresponding vehicle footprints $\mathcal{F}_i^{k'}$ and $\mathcal{F}_i^{k''}$. Each footprint is associated with its planned time $t_{k'}$ or $t_{k''}$ and forms part of the reservation request $\mathcal{Q}_i = \left\{(\mathcal{F}_i^k, t_k)\right\}_{k=0}^{M_i}$.

A. SPATIO-TEMPORAL DISCRETE MODEL

VeloCity solves the optimization problem in the mixed-integer domain. Therefore, it relies on a discrete space-time representation of the urban area. Given the predefined path $\mathcal{P}_i$, each vehicle trajectory is discretized with spatial resolution Δs. Accordingly, the coordinate $s_i(t)$ is mapped into a sequence of M_i spatial cells:

$$\mathcal{S}_i = \left\{s_i^0, s_i^1, \ldots, s_i^{M_i}\right\}, \tag{5}$$

where consecutive elements satisfy $s_i^{k+1} - s_i^k = \Delta s$. Each spatial cell is associated with a two-dimensional position through the trajectory mapping $\gamma_i(\cdot)$. To preserve physical feasibility, each vehicle is represented as a sequence of occupied space-time regions. Let $\mathcal{F}_i^k$ denote the set of spatial cells occupied by vehicle i when its reference position is s_i^k. The corresponding reservation request is therefore:

$$\mathcal{Q}_i = \left\{(\mathcal{F}_i^k, t_k)\right\}_{k=0}^{M_i}. \tag{6}$$

This representation allows collision checking to be converted into a discrete space-time conflict test:

$$\mathcal{F}_i^k \cap \mathcal{R}(t_k) = \emptyset, \quad k = 0, \ldots, M_i, \tag{7}$$

where $\mathcal{R}(t)$ is the current reservation table maintained by the coordinator. A representation of a vehicle trajectory is provided in Fig. 2.

B. DECENTRALIZED MOBILITY OPTIMIZATION

Instead of solving the joint optimization problem in (4), VeloCity decomposes the problem into individual vehicle optimizations. When vehicle i enters the coordination area,

it retrieves the current reservation table $\mathcal{R}$ and locally solves:

$$\begin{aligned} \min_{\mathbf{a}_i} \quad & t_{\text{out},i} - t_{\text{in},i} \\ \text{s.t.} \quad & a_{\min,i} \le a_i(t) \le a_{\max,i}\,, \\ & 0 \le v_i(t) \le \min(V_{\max,i}, V_{\max}(\gamma_i(s_i(t))))\,, \\ & \mathcal{F}_i(t) \cap \mathcal{R}(t) = \emptyset\,, \\ & \forall t \ge t_{\text{in},i}\,. \end{aligned} \tag{8}$$

The local problem preserves the constraints of the original MINLP while removing the coupling among all vehicles. Consequently, each CAV only needs to consider previously committed trajectories rather than the complete fleet state. The local optimization in (8) is solved through a best-first search operating on the constructed state space. The search graph is defined as a labeled directed graph:

$$\mathcal{G}_i = (\mathcal{X}_i, \mathcal{E}_i), \tag{9}$$

where each node $\mathbf{x}_i(t) \in \mathcal{X}_i$ represents a feasible state. The state of vehicle i is defined as:

$$\mathbf{x}_i(t) = (s_i(t), v_i(t), t)\,, \tag{10}$$

where the discrete speed evolution follows:

$$v_i(t + \Delta t) = v_i(t) + a_i(t)\Delta t, \tag{11}$$

and the position evolves according to:

$$s_i(t + \Delta t) = s_i(t) + v_i(t + \Delta t)\Delta t. \tag{12}$$

Each edge is a directed triple:

$$(\mathbf{x}_i(t), a_i(t), \mathbf{x}_i(t + \Delta t)) \in \mathcal{E}_i, \tag{13}$$

which represents the transition from the current state $\mathbf{x}_i(t)$ to the successor state $\mathbf{x}_i(t + \Delta t)$ driven by the specific admissible acceleration command $a_i(t)$. The set of admissible successors is further constrained by the reservation table. A transition is valid only if:

$$\mathcal{F}_i(\mathbf{x}_i(t + \Delta t)) \cap \mathcal{R}(t + \Delta t) = \emptyset. \tag{14}$$

This condition prevents conflicts with previously committed vehicles. The optimal solution is obtained by solving a shortest-path problem on $\mathcal{G}_i$. Since the objective is to minimize the traversal time, the transition cost is defined as the elapsed time:

$$c(\mathbf{x}_i(t), \mathbf{x}_i(t + \Delta t)) = \Delta t\,. \tag{15}$$

Consequently, the accumulated cost of a path corresponds exactly to the arrival time:

$$g(\mathbf{x}_i(t)) = \sum_{\tau = t_{\text{in},i}}^{t} c(\mathbf{x}_i(\tau), \mathbf{x}_i(\tau + \Delta t)) = (t - t_{\text{in},i})\Delta t\,. \tag{16}$$

To determine the shortest-path, each candidate state $\mathbf{x}_i(t)$ is evaluated by:

$$f(\mathbf{x}_i(t)) = g(\mathbf{x}_i(t)) + h(\mathbf{x}_i(t)), \tag{17}$$

where the heuristic term estimates the remaining travel time:

$$h(\mathbf{x}_i(t)) = \frac{D_i - s_i}{V_{\max,i}}. \tag{18}$$

The heuristic in (18) is admissible because it assumes that the vehicle can immediately travel at its maximum allowed speed, thus providing a lower bound on the remaining traversal time. Once the goal condition

$$s_i(t) \ge D_i \tag{19}$$

is reached, the resulting sequence of states directly determines the vehicle mobility profile:

$$\Pi_i = \{(s_i(t), v_i(t), a_i(t))\}_{t = t_{\text{in},i}}^{t_{\text{out},i}}\,. \tag{20}$$

The computed profile is then converted into a sequence of physical space-time reservations $\mathcal{Q}_i$, which are submitted to the coordinator. Upon reception, the coordinator performs:

$$\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{Q}_i. \tag{21}$$

Therefore, future vehicles automatically avoid already committed trajectories. The coordinator only performs lightweight consistency checking, while the computational effort remains distributed among the CAVs.

C. COMPUTATIONAL COMPLEXITY ANALYSIS

By decomposing the MINLP in (4) into independent local searches, the computational complexity of the framework grows approximately linearly with the number of vehicles, as each CAV solves only one sequential best-first search. For a given vehicle i, the size of the local search graph $\mathcal{G}_i = (\mathcal{X}_i, \mathcal{E}_i)$ introduced in Section IV.B is governed by three primary parameters:

- $M_i = D_i/\Delta s$, the total number of discrete spatial cells along the predefined path $\mathcal{P}_i$;
- $V_i = V_{\max,i}/\Delta v$, the number of distinct speed levels induced by the discrete dynamics in (11);
- $b = |\{a_{\min,i}, \ldots, a_{\max,i}\}|$, the branching factor, representing the cardinality of the admissible discrete acceleration set.

Because vehicles cannot travel backward ($v_i(t) \ge 0$), and because the objective function strictly minimizes the arrival time, the algorithm is constrained to move forward through the spatial domain. Consequently, the maximum number of states the algorithm can expand is bounded by the combinations of spatial positions and speed levels:

$$|\mathcal{X}_i| \;\le\; O(M_i\, V_i). \tag{22}$$

1) Per-Vehicle Search Complexity

To determine the optimal mobility profile Π_i, Algorithm 2 employs a best-first search utilizing a binary-heap open list and a hash-indexed closed list. Considering the reservation table $\mathcal{R}$ stored as a hash set indexed by discretized space-time cells, the worst-case time complexity for a single vehicle is:

$$O(b\,|\mathcal{X}_i| \log |\mathcal{X}_i|) = O(b\, M_i V_i \log(M_i V_i))\,, \tag{23}$$

with an auxiliary memory footprint bounded by $O(|\mathcal{X}_i|) = O(M_i V_i)$. This efficiency arises because the admissible

heuristic defined in (18) guarantees that the search expands each reachable state in $\mathcal{X}_i$ at most once, resulting in at most $M_i V_i$ expansions. During each expansion, the algorithm evaluates up to b successors. For each successor, it executes a constant-time $O(1)$ kinematic propagation and a $O(1)$ conflict test against $\mathcal{R}$ via (14). Consequently, the dominant operation per state is the insertion or key-update within the priority queue, scaling at $O(\log(M_i V_i))$ for a binary-heap open list. Following the search, the reservation commit step ($\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{Q}_i$) inserts the sequence of space-time footprints into the global hash set. Because the path length is fundamentally limited by the number of spatial cells, this step requires at most $O(M_i)$, which is dominated by (23).

2) System-Level Complexity

At the system level, Algorithm 1 processes incoming traffic sequentially. Each CAV solves its local problem against the reservations committed by its predecessors. For a total fleet of N vehicles, the theoretical overall system running time scales as:

$$O\left(\sum_{i=1}^{N} b\, M_i V_i \log(M_i V_i)\right) = O\big(N\, b\, \bar{M}\bar{V} \log(\bar{M}\bar{V})\big)\,, \tag{24}$$

where $\bar{M}$ and $\bar{V}$ represent upper bounds across the heterogeneous fleet. While (24) demonstrates that the global complexity is strictly *linear* with respect to N, this theoretical worst-case bound severely overestimates the practical computational burden in realistic urban scenarios. Indeed, in these areas, vehicles do not accumulate indefinitely. As CAVs reach their destinations and exit the coordination area, their spatio-temporal footprints are left in the past. Because the reservation table intrinsically encodes time, a vehicle entering the network at $t_{\mathrm{in},i}$ only evaluates conflict checks against the reservations of currently active vehicles whose temporal horizons overlap with its own. Consequently, the local planning process is completely independent of the historical fleet. This natural spatio-temporal decoupling ensures that the operational lookup time against $\mathcal{R}$ remains constant and bounded by the instantaneous traffic density, rather than growing monotonically with the cumulative vehicle count N.

D. KEY-PROPERTIES OF VELOCITY

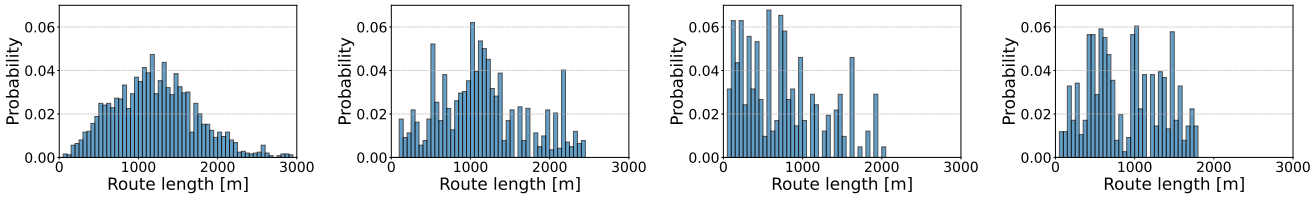
The architectural choices underlying the VeloCity framework yield several advantageous properties for urban traffic coordination:

1) The framework resolves the highly complex global coordination problem through *sequential* and *decentralized optimizations*. As each vehicle queries the coordinator, computes its local mobility profile, and commits its reservation, the system is immediately ready for the next incoming agent. This approach is highly realistic and computationally practical for real-time edge-computing deployment, avoiding costly joint CAV fleet optimization.
2) The reservation-based representation is *independent of any specific road topology*. Because collision avoidance is evaluated directly in physical space-time coordinates, the same mathematical formulation natively supports arbitrary urban layouts, including complex intersections, roundabouts, lane merges, and multi-leg junctions, without requiring custom conflict-resolution rules.
3) The explicit integration of vehicle kinematics into the search space guarantees that the generated *trajectories are not only collision-free but also physically executable*. Furthermore, the geometric footprint $\mathcal{F}_i$ offers additional flexibility; it can be enlarged to reserve extra spatial margins, effectively buffering against uncertainties related to low-level control tracking errors or physical actuation delays.
4) Driven by the objective to minimize arrival time without imposing a hard deadline, the framework gains flexibility. A vehicle can always find a feasible mobility profile by opting to enter the managed area, decelerate, and wait indefinitely until downstream congestion clears. Consequently, a planning fault is triggered *if and only if* the vehicle lacks sufficient space even to safely enter and immediately halt. *Unavoidable deadlocks are thus detected instantaneously at the boundaries*, preventing cascading gridlocks within the core of the infrastructure.
5) In severely congested scenarios where an immediate initial access request fails, the framework is easily extended to support relaxing the requested entry time $t_{\mathrm{in},i}$. *By allowing a vehicle to delay its entry, an optimal and collision-free crossing plan can always be identified*. This temporal relaxation is particularly suited for vehicles departing from parking spots or those waiting in structured access queues at the control area's perimeter.
6) To accommodate heterogeneous urban areas, *the coordinator can pre-initialize static or periodic reservations* within $\mathcal{R}$ prior to scheduling incoming CAVs. Let $\mathcal{R}_{\mathrm{prior}}(t)$ represent the spatio-temporal space pre-allocated to unmanaged traffic entities. This includes public transportation assets (e.g., scheduled tram or bus), deterministic pedestrian crossing windows, or any pre-configured users. Consequently, any newly entering CAV must find a mobility profile that is mutually compliant with both existing vehicular allocations and these background infrastructure constraints, satisfying:
$$\mathcal{F}_i(s_i(t)) \cap (\mathcal{R}_{\mathrm{veh}}(t) \cup \mathcal{R}_{\mathrm{prior}}(t)) = \emptyset\,, \tag{25}$$
$\forall t \geq t_{\mathrm{in},i}$, where $\mathcal{R}_{\mathrm{veh}}(t)$ denotes the union of footprints from previously committed CAVs.
7) Unlike distributed consensus algorithms that require continuous V2V state sharing, VeloCity *preserves privacy* and *minimizes network overhead*. Vehicles only



(a) Tokyo (b) Manhattan (c) Rome (d) Bologna

FIGURE 3: Experimental OpenStreetMap areas for the four evaluated urban scenarios.



(a) Tokyo (b) Manhattan (c) Rome (d) Bologna

FIGURE 4: Empirical probability mass function (PMF) of vehicle route lengths for the four evaluated urban scenarios.

TABLE 1: Spatial metrics of the evaluated urban networks

Scenario	Area	Road Length	Bounding Box (Lon × Lat)
Tokyo	3.07 km^2	93.68 km	$[139.66, 139.68] \times [35.68, 35.69]$
Manhattan	4.24 km^2	102.98 km	$[-73.99, -73.97] \times [40.75, 40.76]$
Rome	1.73 km^2	60.88 km	$[12.47, 12.48] \times [41.89, 41.90]$
Bologna	2.54 km^2	57.62 km	$[11.33, 11.35] \times [44.49, 44.49]$

exchange once their intended spatio-temporal footprints with the coordinator via V2I links, without exposing exact destinations, internal control parameters, or complete routing intentions to the broader fleet.

8) By processing reservation requests in the order they arrive at the coordinator, the framework enforces a fair sequential allocation of space-time slots. This guarantees deterministic access, ensuring that early vehicles are not starved by subsequent high-density traffic.

V. NUMERICAL RESULTS

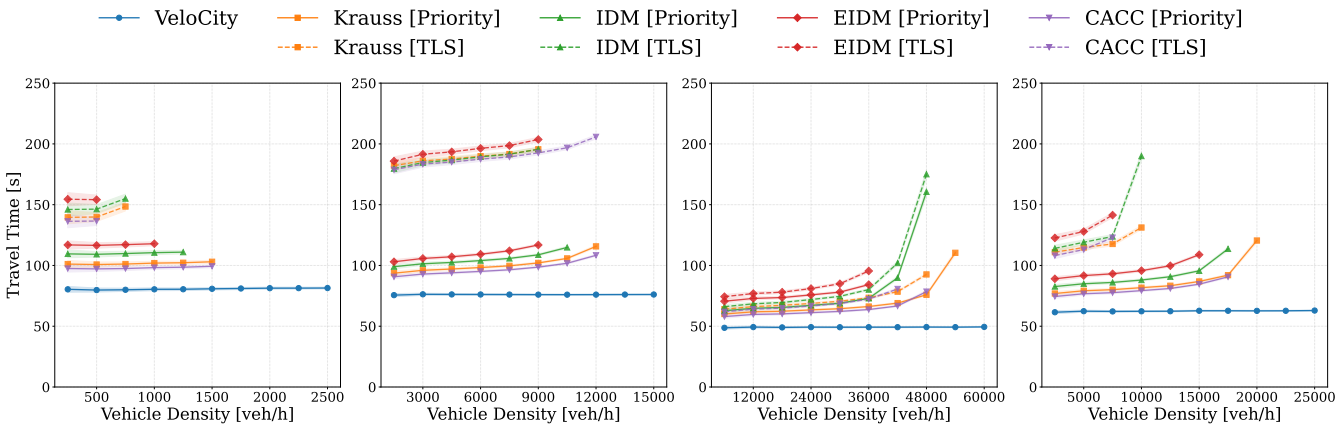
In this section, we extensively evaluate the performance of VeloCity across four large-scale real-world urban scenarios.

A. EXPERIMENTAL SETUP

To evaluate the performance and scalability of the proposed framework, we conduct extensive simulations across four distinct and highly heterogeneous urban areas: Tokyo, Manhattan, Rome, and Bologna. The geographic road networks for these scenarios are illustrated in Fig. 3. To ensure a high degree of realism in our mobility modeling, the network topology for each scenario is imported directly from OpenStreetMap, including lane definitions, explicit restrictions at intersections, legal speed limits, and detailed intersection layouts. The selected areas offer diverse topological characteristics, ranging from highly structured grid layouts to irregular historical streets. To quantify this diversity, the specific spatial metrics for each evaluated scenario are summarized in Table 1.

All numerical experiments simulate traffic operations over a duration of one hour in SUMO [13]. Realistic traffic demand is generated utilizing the `randomTrips.py` utility provided by the SUMO simulator. This tool automatically generates randomized but physically valid routes across the imported OpenStreetMap networks, ensuring that vehicles are spawned at valid distributed edges and assigned achievable destinations within the boundaries of the control area. To characterize the resulting traffic demand, Fig. 4 reports the empirical PMF of the vehicle route lengths for each of the four urban scenarios.

Within SUMO, the discrete space-time model operates with a temporal resolution of $\Delta t = 0.5\,\mathrm{s}$ and a spatial resolution of $\Delta s = 0.5\,\mathrm{m}$. To reflect typical urban driving behavior, all CAVs feature a maximum speed equal to $V_{\max,i} = 50\,\mathrm{km/h}$. Acceleration is bounded by a maximum capacity of $a_{\max,i} = 1\,\mathrm{m/s^2}$ and a maximum deceleration of $a_{\min,i} = -2\,\mathrm{m/s^2}$. All generated vehicles are 5 m long, 2 m



(a) Tokyo (b) Manhattan (c) Rome (d) Bologna

FIGURE 5: Vehicle travel time for the four evaluated urban scenarios.

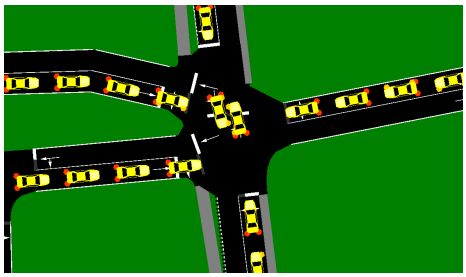


FIGURE 6: Example of a congestion event in the Tokyo scenario incurred by baselines.

wide. Furthermore, to ensure robust collision avoidance and to account for physical actuation uncertainties (as outlined in the third item in Section D), the geometric footprint $\mathcal{F}_i$ is defined to incorporate a strict safety margin. Consequently, an area of 2.5 m around each vehicle's boundaries is reserved during the spatio-temporal allocation process.

B. BENCHMARKS

To evaluate the performance of the proposed VeloCity framework, we compare it against a set of established car-following models natively implemented within the SUMO. Specifically, we consider the following models:

1) *Krauss:* The default car-following model in SUMO. It computes a safe speed based on the leading vehicle's speed and the available gap, incorporating a stochastic imperfection parameter to simulate realistic human driving behavior and reaction delays [42].
2) *Intelligent Driver Model (IDM):* A continuous-time car-following model that transitions between free-road acceleration and collision-avoidance deceleration. It is governed by a desired target speed and a safe time-headway parameter [43].
3) *Extended IDM (EIDM):* An enhancement of the classical IDM specifically calibrated for sub-second simulation resolutions. The EIDM introduces explicit reaction times, driver estimation errors, and jerk limitations to yield more accurate acceleration profiles in dense urban traffic [44].
4) *Cooperative Adaptive Cruise Control (CACC):* A deterministic model representing connected autonomous vehicles that leverage V2V communications. By sharing acceleration and speed data, CACC vehicles can safely maintain tight inter-vehicle headways and string stability [45].

The evaluation of these baseline models is split into two distinct operational settings across the simulated networks:

1) *Signalized Control:* Intersections within the managed urban areas are possibly governed by traffic light signals (TLS), according to their OpenStreetMap data. Vehicles operating under the baseline car-following models must react to cyclic red/green phases in addition to their leading vehicles.
2) *Autonomous Priority Control:* Traffic lights are completely disabled. Intersections operate based on right-of-way priority rules (e.g., yield and stop mechanics). Vehicles relying on the benchmark models must dynamically identify safe spatial gaps in the crossing traffic streams before proceeding.

Conversely, VeloCity inherently overrides both traditional traffic lights and static priority rules. Vehicles operating under the proposed framework navigate the identical unsignalized network layouts by relying entirely on the decentralized spatio-temporal algorithm introduced in Section IV. Overall, our experimental analysis compares the proposed methodology with eight alternative approaches.

C. TRAVEL TIME AND ENERGY CONSUMPTION ANALYSIS

To comprehensively evaluate the proposed framework, we assess both the travel time and energy consumption metrics

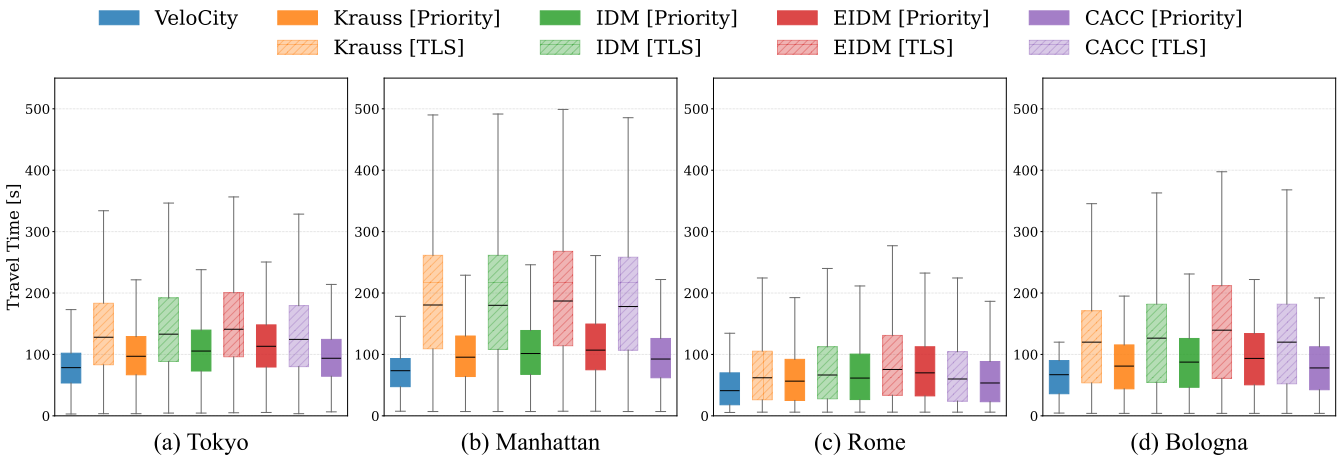


FIGURE 7: Vehicle travel time distribution for the four evaluated urban scenarios, comprising (a) 500 veh/h for Tokyo, (b) 9,000 veh/h for Manhattan, (c) 36,000 veh/h for Rome, and (d) 7,500 veh/h for Bologna.

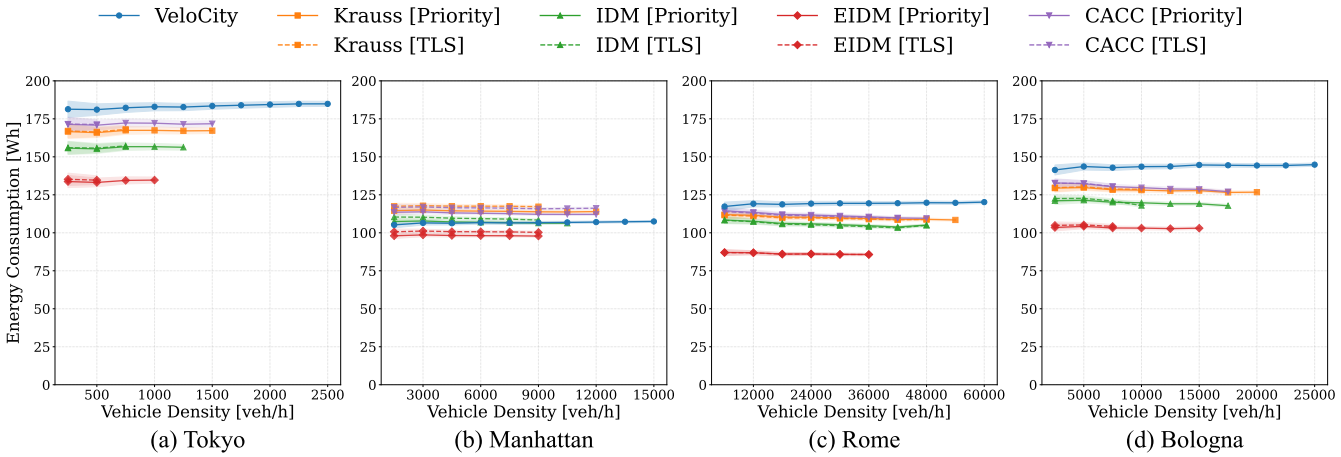
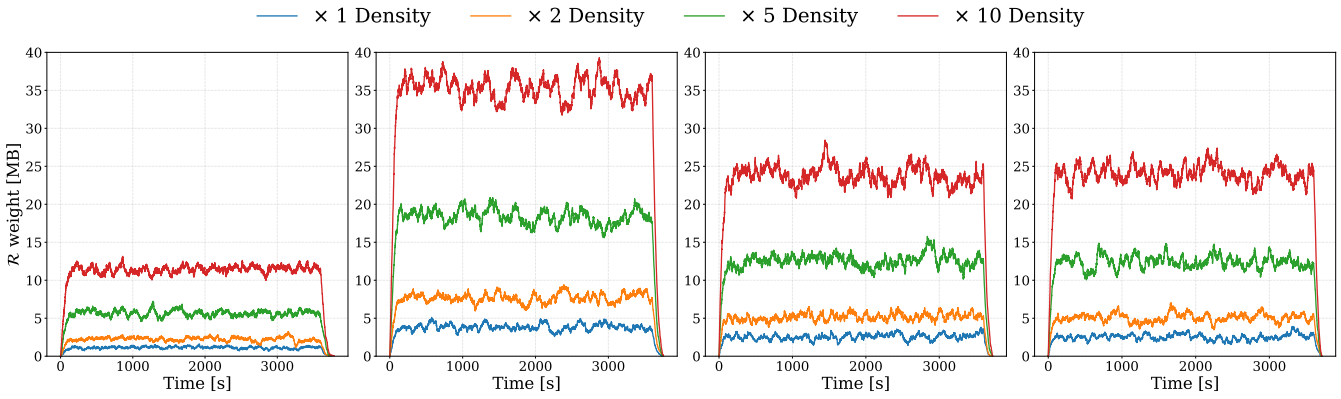


FIGURE 8: Vehicle energy consumption for the four evaluated urban scenarios.

across the CAV fleets. For the energy analysis, we assume a fleet composed entirely of fully electric vehicles. The experiments are conducted by varying the traffic demand, defined in terms of vehicles per hour. To ensure statistical robustness, each scenario-density combination is evaluated over five independent simulation runs with varying random seeds.

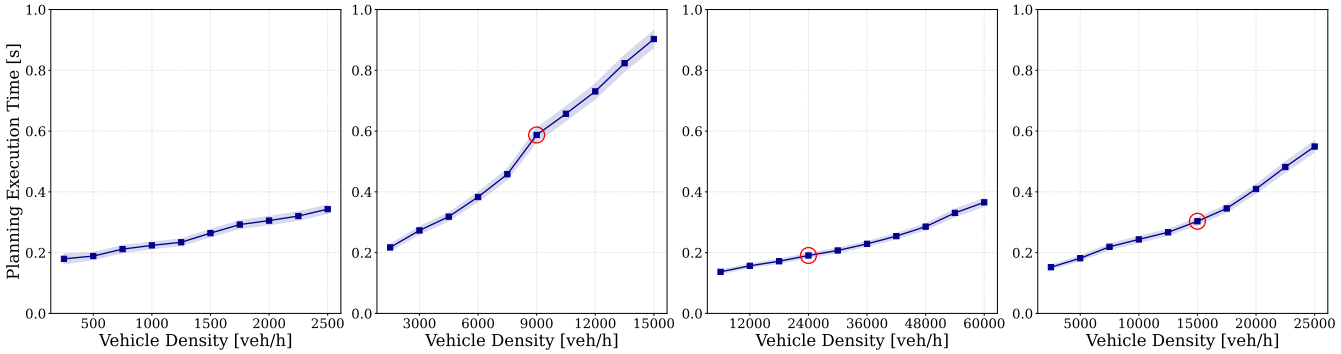
The aggregate travel time results are depicted in Fig. 5, with data reported in terms of average per-vehicle travel time and 95% confidence intervals. Across all four urban areas, VeloCity demonstrates superior efficiency and remarkable stability as density increases. Indeed, the average delay for vehicles managed by VeloCity remains practically constant across the entire tested density spectrum for each city. For example, in the Tokyo scenario, VeloCity maintains a steady average travel time of approximately 80 to 81 seconds even as the traffic demand scales drastically from 250 up to 2,500 veh/h. This flat trend is consistently replicated in Manhattan, Rome, and Bologna scenarios. In contrast, all baseline models exhibit a clear and rapid degradation in performance as traffic volume grows. The curves for these benchmarks escalate quickly and are truncated in the figures. This truncation occurs because, at sufficiently high traffic volumes, the baseline models can enter gridlock conditions in which mutually blocking vehicles prevent further progress through one or more intersections. The maximum traffic demand at which this occurs is highly scenario-dependent. Figure 6 illustrates a representative localized gridlock event. Such events arise because reactive car-following and gap-acceptance mechanisms can lead to queues that propagate across adjacent junctions, eventually creating mutually blocking traffic streams. VeloCity is inherently robust to this problem. By leveraging a sequential distributed planning architecture, each vehicle computes its spatio-temporal trajectory in advance, preemptively reserving clear paths through complex intersections.

Figure 7 illustrates the actual distribution of travel times for each scenario at the maximum density safely supported



(a) Tokyo (b) Manhattan (c) Rome (d) Bologna

FIGURE 9: Reservation table $\mathcal{R}$ weight over time for the four evaluated urban scenarios, where the base density is (a) 250 veh/h for Tokyo, (b) 1,500 veh/h for Manhattan, (c) 6,000 veh/h for Rome, and (d) 2,500 veh/h for Bologna.



(a) Tokyo (b) Manhattan (c) Rome (d) Bologna

FIGURE 10: Vehicle planning time for the four evaluated urban scenarios.

by all baselines. The box plots reveal that VeloCity not only reduces the mean travel time, but it also reduces variance. This confirms that VeloCity provides highly predictable and deterministic service guarantees. Finally, in Fig. 8, we evaluate the average per-vehicle energy consumption. Because VeloCity is designed to optimize travel times, vehicles traverse the network at higher average speeds. Therefore, the sustained higher cruising speeds can result in slightly higher overall energy consumption compared to the baselines.

D. COORDINATION AND PLANNING ANALYSIS

To ensure high computational efficiency and fast response times, the spatio-temporal reservation table $\mathcal{R}$ is implemented leveraging a hash map data structure, where each committed footprint $\mathcal{F}_i^k$ at time t_k is mapped to a unique hashed key. Essentially, a hash map functions as a highly optimized dictionary that links unique identifiers (i.e., the hash keys) directly to stored data. In this context, the spatial coordinates and timestamp of a footprint are mathematically transformed into a single key, enabling the system to instantly check a specific location's availability in constant time ($\mathcal{O}(1)$). In this way, rather than maintaining a dense grid (which would scale poorly with the area size and time horizon), the coordinator stores only the actively reserved discrete space-time cells. Fig. 9 illustrates the active memory footprint of the reservation table for VeloCity across the four evaluated scenarios, scaled at 1×, 2×, 5×, and 10× of their respective baseline densities. The hash map representation proves to be lightweight: for the vast majority of cases, the memory utilization remains in the order of a few megabytes. The maximum memory footprint is reached only under the most extreme tested conditions, specifically during the maximum density peaks in the Manhattan scenario.

Beyond memory efficiency, we evaluate the system's temporal scalability by analyzing the total planning time required for a vehicle to be fully scheduled. As depicted in Fig. 10, this total planning time encompasses three sequential steps: 1) the coordinator transmits the current reservation table $\mathcal{R}$ to the CAV entering the control area; 2) the CAV executes the local optimization outlined in Algorithm 1; and 3) the CAV transmits its computed reservation request $\mathcal{Q}_i$ back to the coordinator. For the communication overhead, we assume the deployment of standard 5G V2I networks, which impose a maximum nominal latency of 10 ms for both the uplink and

downlink transmissions [46]. The local computation (Step 2) is evaluated over a standard single-thread Intel Core i7 processor. Under these hardware and network assumptions, VeloCity successfully resolves the entire planning loop for each vehicle in just a few hundreds of milliseconds for most cases. For instance, in the Tokyo scenario, the mean planning time smoothly scales from roughly 180 ms at 250 veh/h to just 343 ms at 2,500 veh/h. However, our stress-test experiments deliberately subject the framework to extremely high vehicular demands to identify its theoretical operational limits. Specifically, we push the Manhattan scenario up to 15,000 veh/h, Rome to 60,000 veh/h, and Bologna to 25,000 veh/h. To prevent a backlog of unscheduled vehicles, the per-vehicle planning time t_{plan} must not exceed the average time between vehicle arrivals. Given a traffic density D (in vehicles per hour), this constraint mandates that $t_{\text{plan}} \leq (3.6 \times 10^6)/D$ (in ms). For example, a density of 1,000 veh/h yields a maximum planning window of 3,600 ms per vehicle. While VeloCity comfortably satisfies this threshold for most traffic densities, executing Algorithm 1 on a single thread eventually hits a computational bottleneck under extreme congestion. Notably, the adaption of dedicated hardware would significantly reduce planning times. Figure 10 highlights with red circles the specific scenario-density pairs where the computational time exceeds the inter-arrival threshold, indicating the onset of queue accumulation. It is important to emphasize that this limitation stems purely from the baseline hardware assumptions and the current single-thread search implementation. One way to overcome this computational bottleneck could be extending Algorithm 1 to a multi-threaded parallel search architecture, an enhancement previously proven highly effective for similar domains [47]. We leave the integration of parallelized local planning for future work.

VI. CONCLUSION

This paper presented *VeloCity*, a decentralized multi-agent spatio-temporal mobility profile planning framework designed to manage high-density automated traffic across arbitrary urban road topologies. VeloCity effectively bypasses the scalability bottlenecks of centralized traffic management systems by shifting from centralized mobility profile optimization to decentralized optimization directly solved at the individual CAVs. Extensive simulations conducted in SUMO across four large-scale real-world urban networks demonstrated the framework's superior efficiency. Compared to established baselines, VeloCity guarantees physically executable and collision-free mobility profiles while drastically reducing travel times, minimizing delay variance, and preventing cascading congestion even under high vehicular densities.

Despite these highly promising results, several avenues for future research remain. First, to further reduce the local planning time and eliminate computational queuing under ultra-high traffic demands, the underlying single-thread search algorithm can be extended into a multi-threaded parallel search architecture. Second, the reservation framework can be expanded to dynamically accommodate users with priorities (e.g., emergency vehicles). This would involve allowing priority entities to overwrite existing spatio-temporal allocations, requiring the development of a robust and efficient re-scheduling mechanism for affected CAVs. Third, future iterations can integrate cooperative mechanisms to safely handle mixed traffic scenarios, allowing non-cooperative or human-driven vehicles to coexist with the managed fleet. Finally, while the current objective function is tailored to strictly minimize system-wide travel time, incorporating energy consumption minimization into the local optimization problem will be crucial to explicitly balance the trade-off between crossing speed and overall energy efficiency.

REFERENCES

[1] J. Wu and X. Qu, "Intersection control with connected and automated vehicles: A review," *Journal of Intelligent and Connected Vehicles*, vol. 5, no. 3, pp. 260–269, 2022.

[2] L. Farina, L. M. Amorosa, M. Rapelli, B. M. Masini, C. Casetti, and A. Bazzi, "V2N-based algorithm and communication protocol for autonomous non-stop intersections," 2026. [Online]. Available: https://arxiv.org/abs/2603.05165

[3] T. T. Zhang, P. J. Jin, S. T. McQuade, A. Bayen, and B. Piccoli, "Car-following models: A multidisciplinary review," *IEEE Transactions on Intelligent Vehicles*, vol. 10, no. 1, pp. 92–116, 2025.

[4] K. Dresner and P. Stone, "A multiagent approach to autonomous intersection management," *Journal of artificial intelligence research*, vol. 31, pp. 591–656, 2008.

[5] A. Abbas-Turki, Y. Mualla, N. Gaud, D. Calvaresi, W. Du, A. Lombard, M. Dridi, and A. Koukam, "Autonomous intersection management: Optimal trajectories and efficient scheduling," *Sensors*, vol. 23, no. 3, p. 1509, 2023.

[6] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Optimal coordination of automated vehicles at intersections: Theory and experiments," *IEEE Transactions on Control Systems Technology*, vol. 27, no. 6, pp. 2510–2525, 2019.

[7] G. Avino, M. Malinverno, C. Casetti, C. Chiasserini, F. Malandrino, M. Rapelli, and G. Zennaro, "Support of safety services through vehicular communications: The intersection collision avoidance use case," in *2018 International Conference of Electrical and Electronic Technologies for Automotive*. IEEE, 2018, pp. 1–6.

[8] M. Rapelli, F. Raviglione, and C. Casetti, "OScar: An ETSI-compliant C-ITS stack for field-testing with embedded hardware devices," in *2024 22nd Mediterranean Communication and Computer Networking Conference (MedComNet)*. IEEE, 2024, pp. 1–4.

[9] SAE, "Application protocol and requirements for maneuver sharing and coordinating service," *SAE J3186*, 2023.

[10] ——, "Taxonomy and definitions for terms related to cooperative driving automation for on-road motor vehicles," *SAE J3216*, 2020.

[11] ETSI, "Intelligent Transport Systems (ITS); Vehicular Communications; Informative report for the Maneuver Coordination Service," *ETSI TR 103 578*, 2023.

[12] ——, "Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Maneuver Coordination Service," *ETSI TS 103 561*, Under development.

[13] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wiessner, "Microscopic traffic simulation using SUMO," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2575–2582.

[14] N. H. Gartner, J. D. Little, and H. Gabbay, "Optimization of traffic signal settings by mixed-integer linear programming: Part I: The network coordination problem," *Transportation Science*, vol. 9, no. 4, pp. 321–343, 1975.

[15] A. Stevanovic, C. Kergaye, and P. T. Martin, "Scoot and scats: A closer look into their operations," in *88th Annual Meeting of the Transportation Research Board. Washington DC*, vol. 500, no. 801, 2009, p. 18.

[16] K. Aboudolas, M. Papageorgiou, A. Kouvelas, and E. Kosmatopoulos, "A rolling-horizon quadratic-programming approach to the signal control problem in large-scale congested urban road networks," *Transportation Research Part C: Emerging Technologies*, vol. 18, no. 5, pp. 680–694, 2010.

[17] L. Wei, K. Ampountolas, A. Hirrle, and M. Wang, "Hierarchical predictive control of network traffic signals using link transmission model with queue dynamics," *IEEE Transactions on Intelligent Transportation Systems*, 2025.

[18] S. Smith, G. Barlow, X.-F. Xie, and Z. Rubinstein, "Smart urban signal networks: Initial application of the surtrac adaptive traffic signal control system," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 23, 2013, pp. 434–442.

[19] H.-C. Hu, A. Hawkes, and S. F. Smith, "Incorporating queue dynamics into schedule-driven traffic control," in *International Joint Conference on Artificial Intelligence*, 2021.

[20] P. Varaiya, "Max pressure control of a network of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 36, pp. 177–195, 2013.

[21] T. Xu, S. Barman, and M. W. Levin, "Smoothing-MP: A novel max-pressure signal control considering signal coordination to smooth traffic in urban networks," *Transportation Research Part C: Emerging Technologies*, vol. 166, p. 104760, 2024.

[22] W. Remmerswaal, D. Sun, A. Jamshidnejad, and B. De Schutter, "Combined MPC and reinforcement learning for traffic signal control in urban traffic networks," in *2022 26th International Conference on System Theory, Control and Computing (ICSTCC)*. IEEE, 2022, pp. 432–439.

[23] J. Ma and F. Wu, "Learning to coordinate traffic signals with adaptive network partition," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 1, pp. 263–274, 2023.

[24] K. Wang, Z. Shen, Z. Lei, X. Liu, and T. Zhang, "Towards multi-agent reinforcement learning based traffic signal control through spatio-temporal hypergraphs," *IEEE Transactions on Mobile Computing*, 2025.

[25] R. Tachet, P. Santi, S. Sobolevsky, L. I. Reyes-Castro, E. Frazzoli, D. Helbing, and C. Ratti, "Revisiting street intersections using slot-based systems," *PloS one*, vol. 11, no. 3, p. e0149607, 2016.

[26] D. Maksimovski, A. Festag, and C. Facchi, "A survey on decentralized cooperative maneuver coordination for connected and automated vehicles," in *Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems*. SciTePress, 2021, pp. 100–111.

[27] Q. Guo, L. Li, and X. J. Ban, "Urban traffic signal control with connected and automated vehicles: A survey," *Transportation research part C: emerging technologies*, vol. 101, pp. 313–334, 2019.

[28] A. Bazzi, A. Zanella, and B. M. Masini, "A distributed virtual traffic light algorithm exploiting short range V2V communications," *Ad Hoc Networks*, vol. 49, pp. 42–57, 2016.

[29] A. Jame, M. Rapelli, and C. Casetti, "Reducing pollutant emissions through virtual traffic lights," *Computer Communications*, vol. 188, pp. 167–177, 2022.

[30] Y.-T. Lin, H. Hsu, S.-C. Lin, C.-W. Lin, I. H.-R. Jiang, and C. Liu, "Graph-based modeling, scheduling, and verification for intersection management of intelligent vehicles," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 18, no. 5s, pp. 1–21, 2019.

[31] P. Karthikeyan, W.-L. Chen, and P.-A. Hsiung, "Autonomous intersection management by using reinforcement learning," *Algorithms*, vol. 15, no. 9, p. 326, 2022.

[32] Y. Zhang and C. G. Cassandras, "Decentralized optimal control of connected automated vehicles at signal-free intersections including comfort-constrained turns and safety guarantees," *Automatica*, vol. 109, p. 108563, 2019.

[33] R. Hult, M. Zanon, S. Gros, and P. Falcone, "Energy-optimal coordination of autonomous vehicles at intersections," in *2018 European control conference (ECC)*. IEEE, 2018, pp. 602–607.

[34] B. Zheng, C.-W. Lin, H. Liang, S. Shiraishi, W. Li, and Q. Zhu, "Delay-aware design, analysis and verification of intelligent intersection management," in *2017 IEEE International Conference on Smart Computing (SMARTCOMP)*. IEEE, 2017, pp. 1–8.

[35] Q. Lu, H. Jung, and K.-D. Kim, "Optimization-based approach for resilient connected and autonomous intersection crossing traffic control under V2X communication," *IEEE Transactions on Intelligent Vehicles*, vol. 7, no. 2, pp. 354–367, 2021.

[36] M. Hausknecht, T.-C. Au, and P. Stone, "Autonomous intersection management: Multi-intersection optimization," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 4581–4586.

[37] M. Vasirani and S. Ossowski, "A market-inspired approach for intersection management in urban road traffic networks," *Journal of artificial intelligence research*, vol. 43, pp. 621–659, 2012.

[38] C. Wuthishuwong and A. Traechtler, "Consensus-based local information coordination for the networked control of the autonomous intersection management," *Complex & Intelligent Systems*, vol. 3, no. 1, pp. 17–32, 2017.

[39] A. Okoso, B. Okumura, K. Otaki, and T. Nishi, "Network-flow-problem-based approach to multi-agent path finding for connected autonomous vehicles," in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2021, pp. 1946–1953.

[40] T. Tranos, C. Spatharis, K. Blekas, and A.-G. Stafylopatis, "Large-scale urban traffic management using zero-shot knowledge transfer in multi-agent reinforcement learning for intersection patterns," *Robotics*, vol. 13, no. 7, p. 109, 2024.

[41] S. Liu, M. Fan, W. Li, J. Du, and S. Li, "Large-scale mixed-traffic and intersection control using multi-agent reinforcement learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2025, pp. 13 777–13 784.

[42] S. Krauss, "Microscopic modeling of traffic flow: investigation of collision free vehicle dynamics," Apr 1998.

[43] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, pp. 1805–1824, Aug 2000. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.62.1805

[44] D. Salles, S. Kaufmann, and H.-C. Reuss, "Extending the intelligent driver model in sumo and verifying the drive off trajectories with aerial measurements," *SUMO Conference Proceedings*, vol. 1, p. 1–25, Jul. 2022. [Online]. Available: https://www.tib-op.org/ojs/index.php/scp/article/view/95

[45] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transportation Research Part C: Emerging Technologies*, vol. 48, pp. 285–300, 2014.

[46] K. Moto, M. Mikami, K. Serizawa, and H. Yoshino, "Field experimental evaluation on 5G V2N low latency communication for application to truck platooning," in *2019 IEEE 90th vehicular technology conference (VTC2019-fall)*. IEEE, 2019, pp. 1–5.

[47] S. Ahmadi, N. R. Sturtevant, A. Raith, D. Harabor, and M. Jalili, "Parallelizing multi-objective A* search," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 35, no. 1, 2025, pp. 131–139.



Lorenzo Mario Amorosa (Member, IEEE) received the B.S. degree in Computer Engineering and the M.S. degree in Artificial Intelligence from the University of Bologna, Italy, in 2019 and 2021, respectively. In 2025, he received the Ph.D. degree in Electronics, Telecommunications and Information Technologies Engineering from the University of Bologna. Currently he is with the Department of Electronic, Information and Electrical Engineering "Guglielmo Marconi" as a Postdoctoral Research Fellow at University of Bologna. He is Research Associate at the National Laboratory of Wireless Communications (WiLab) of CNIT (the National, Inter-University Consortium for Telecommunications). His main research interests include decentralized artificial intelligence, cooperative multi-agent systems, and deep learning for wireless communications.



Lorenzo Farina (Graduate Student Member, IEEE) received the B.S. degree in Electronic and Telecommunications Engineering and the M.S. degree in Telecommunications Engineering from the University of Bologna, Italy, in 2021 and 2024, respectively. He is currently pursuing the Ph.D. degree in Automotive Engineering for Intelligent Mobility at the University of Bologna. He is Research Associate at the National Laboratory of Wireless Communications (WiLab) of CNIT (the National, Inter-University Consortium for Telecommunications). His research interests include networks of connected and autonomous vehicles, with a particular focus on algorithms and protocols for cooperative maneuvering.



Vittorio Todisco (Member, IEEE) received the Ph.D. degree in Automotive Engineering for Intelligent Mobility from the University of Bologna, Italy, where he is currently a Research Fellow. His research focuses on wireless communications for connected and automated vehicles, including 5G/6G V2X, NR sidelink, distributed resource allocation, and interference management. He also closely follows ETSI standardization activities for intelligent transportation systems.



Alessandro Bazzi (Senior Member, IEEE) is an Associate Professor at the University of Bologna and co-founder of the Wireless Communications Laboratory of CNIT, WiLab. His research interests focus on networks of connected and autonomous vehicles, including radio resource management of wireless networks and large-scale automotive radar interference. On these topics, he coordinates projects and contributes to ETSI standardization.