



HAL
open science

Multi-Rules Reachability Analysis for Road Agents Using Graph-Based Maps and Real-Time Kinematics

Monica Fossati, Ezio Malis, Philippe Martinet

► To cite this version:

Monica Fossati, Ezio Malis, Philippe Martinet. Multi-Rules Reachability Analysis for Road Agents Using Graph-Based Maps and Real-Time Kinematics. IV 2025 - IEEE Intelligent Vehicles Symposium, Jun 2025, Cluj - Napoca, Romania. pp.119-124, <10.1109/IV64158.2025.11097553>. <hal-05042259>

HAL Id: hal-05042259

<https://hal.science/hal-05042259v1>

Submitted on 22 Apr 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Multi-Rules Reachability Analysis for Road Agents Using Graph-Based Maps and Real-Time Kinematics

Monica Fossati, Ezio Malis and Philippe Martinet

Abstract—Automated vehicles perform well in simple environments with clear rules, but urban traffic presents significant challenges due to the unpredictable behavior of road users, sometimes beyond traffic rules. Achieving full autonomy in such settings requires a systematic approach to modeling the possible actions of all agents. This paper presents a multi-rules reachability analysis framework that integrates graph-based maps with real-time perception data to dynamically characterize the surrounding space. By leveraging the semantic richness and modularity of Lanelet2 maps, our method provides a structured representation that enhances situational awareness. This allows for the extraction of navigation-relevant information, with the goal of supporting safer and more efficient decision-making in complex urban environments.

I. INTRODUCTION

Recent advances in autonomous vehicle (AV) research have led to more efficient systems in specific contexts, but achieving full autonomy remains a complex challenge [1]. While automated vehicles perform well in controlled environments like highways, urban settings are more challenging due to the dynamic and unpredictable behavior of diverse road users, sometimes outside strict traffic rules. These complexities can disrupt traffic flow and reduce efficiency, as automated vehicles typically follow rigid rules and conservative behaviors [2]. To address these issues, we aim to develop truly autonomous systems capable of context-aware decision-making, moving beyond purely rule-based approaches. To make informed decisions, an autonomous vehicle requires clear situation awareness, which involves identifying relevant environmental elements that influence its behavior [3]. This includes determining the agent’s reachable set on the map, which reflects its action capabilities. By incorporating map data and the state of other agents, the reachable set helps identify potential interactions, enhancing the vehicle’s ability to anticipate and respond to its surroundings. Our work focuses on improving the identification of the reachable set, integrating information from both the map and perception. The adopted approach provides a multi-level view of the problem, considering not only legal reachability, defined by traffic regulations, but also physical reachability (Figure 1). This approach allows for a deeper understanding of the situation, and aims for improving the decision-making process.

Several previous works have significantly contributed to environment modeling for autonomous vehicles. Early approaches, such as [4], [5], primarily focused on graph-based

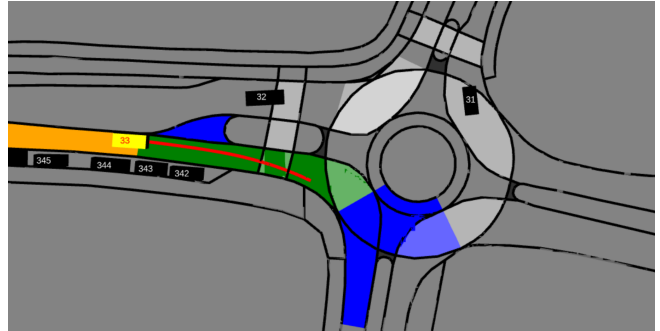


Fig. 1: Example of legal (green) and physical (blue) reachable sets

representations of road networks and the relationships between map elements. However, many of these functionalities, including the representation of relationships between map elements, are now fully integrated into the Lanelet2 standard [6], which provides an efficient and flexible framework for high-definition (HD) maps. Lanelet2 also introduces additional capabilities, such as pathfinding and the calculation of reachable sets based on the traffic rules [7]. Despite these advancements, Lanelet2 remains an offline tool, lacking the ability to incorporate dynamic, real-time updates from sensor data, as highlighted by Dempster et al. [8]. Koschi et al. [9]- [10] performed a reachability analysis in which they used the lanelet concept to define the road as a highly structured environment, focusing primarily on the geometry of the reachable space under such constraints. Their approach relies on traffic rules, and when a rule is violated, the corresponding constraints are ignored. In the case of violated accessibility rules, no road constraints are applied to the participant, leading to excessive uncertainty in predicting its future behavior. Subsequent to their work, the Lanelet2 framework was extended with new tools for creating routing graphs, enabling the possibility to have different rule sets and cost models [7].

This work introduces key advancements in enhancing the situational awareness of autonomous vehicles:

- **Improved Reachable Set Calculation** integrating Lanelet2 maps with road agents’ actual states and kinematic limits. We provide a more accurate reachability estimation at the lanelet level.
- **Physical Rule Implementation** leveraging the modularity of the Lanelet2 standard to estimate multiple reachable sets capable of describing and anticipating scenarios involving relaxed or violated traffic rules.

The proposed approach provides a real-time reachability analysis based on the current kinematics of road agents. By integrating both traffic and physical rules, it enhances the

understanding of human dynamics and supports proactive strategies, improving urban navigation.

II. THEORETICAL BACKGROUND

A. Lanelet2 Maps

Lanelet2 maps are a powerful tool for describing the scenery and the relationships between its elements, called primitives.

a) **Primitives:** Lanelet2 maps are built upon a set of fundamental *primitives*, which include:

- **Linestrings:** Polylines representing real-world elements like curbs and markings, or virtual boundaries defining lanelets without a physical counterpart. These serve as the foundational building blocks for the map structure.
- **Lanelets:** Drivable lane segments with one-way directional movement, defined by left and right linestrings. Lanelets are the primary navigable components in Lanelet2 maps.
- **Areas:** Closed regions composed of linestrings, designed for undirected movement. Areas complement lanelets by representing navigable spaces that do not adhere to strict lane-based movement.

Each primitive can have associated *tags* that specify its characteristics. Tags include semantic information, speed limits, and types of participants (e.g., cars, bicycles, or pedestrians) permitted to traverse the primitive. There is no fixed standard or limit on the tags that can be associated with a primitive, allowing for flexible and detailed descriptions of map elements. The Lanelet2 library also provides sophisticated tools for pathfinding and routing, which leverage the key components listed below.

b) **Traffic Rules:** Traffic rules define access policies, right-of-way and speed limits. These rules specify permissible behaviors for different participants (e.g., cars, bicycles, or pedestrians), meaning they are inherently *subject-dependent*. They specify whether a participant can access a lanelet or area and whether it can transition between them by checking boundary passability. Traffic rules are based on the **tags** associated with primitives, enabling a detailed and flexible definition of road usage policies.

c) **Routing Graph:** Lanelet2 library generates a *routing graph* based on the map primitives and the given traffic rules.

- **Vertices:** Lanelets and Areas form the graph's nodes.
- **Edges:** Edges represent valid transitions between primitives, such as "following" or "lane change", which are determined by geometric relationships and traffic rules.

d) **Routing Costs:** To evaluate various possible paths according to traffic rules, it is essential to define a cost function that describes the "cost" of transitioning from one lanelet or area to another. This approach allows for the identification of not only feasible paths but also the optimal ones based on specific criteria, such as minimizing time or distance.

This library allows estimates that do not account for the participant's real-time state beyond the lanelet they are currently in.

B. Reachability

a) **Reachable Set of a System:** The *reachable set* of a system is the set of all states that the system can reach from its initial state $x_0 = x(0)$ and a control input set U based on the system's dynamics. LaValle [11, Eq. 14.4] formally defined it as:

$$\mathcal{R}(x_0, U) = \{x_1 \in X \mid \exists \bar{u} \in U \text{ and } \exists t \in [0, \infty): x(t) = x_1\},$$

- X is the system's state set,
- U is the control input set,
- $x(t)$ is the state trajectory.

Mathematical methods exist to compute the exact *reachable set*, relying on reachability matrices derived from the system's dynamics [12]. A major drawback of these methods is their high computational cost when maintaining mathematical accuracy. To make them suitable for real-time applications such as autonomous driving, where rapid response is critical, certain approximations must be introduced [13], which inevitably compromise their precision.

b) **Reachable Set in Lanelet2:** Lanelet2 maps leverage the structured nature of driving environments, which are governed by well-defined traffic rules. These maps provide a discretization of the environment into primitives, as presented in the previous subsection II-A. Through this discretization, the complex problem of reachability is transformed into the more manageable task of determining the *reachable set* of primitives. The *reachable set*, defined in [7], represents the collection of lanelets or areas that can be accessed starting from a specific initial lanelet l_0 , given a maximum allowable cost and the connections defined by the routing graph. To determine it, we compute the path (or sequence if it includes areas) [7] from l_0 to each lanelet l in the routing graph. Formally, the reachable set can be defined as:

$$\mathcal{R}_{\text{lanelet}} = \{l \mid l \in \mathcal{L}, \text{path}(l_0, l) \in \mathcal{P}, \text{cost}(\text{path}(l_0, l)) \leq C\},$$

- l is a lanelet in the map,
- \mathcal{L} is the set of all lanelets,
- \mathcal{P} is the set of all paths,
- l_0 is the initial lanelet,
- C is the maximum allowable cost (e.g., distance, time, fuel, etc.) based on the vehicle's rules and constraints.

C. Situation Awareness

To enhance the situation awareness of an autonomous vehicle, it is crucial to first define what constitutes a *situation*. As described by Ulbrich et al. in [3], a situation includes all *relevant* conditions and factors influencing behavior at a given moment. It is derived from the *scene*, which represents a snapshot of the environment, including the scenery, dynamic elements, and the relationships between all actors and their self-representations. The situation is formed by selecting and augmenting relevant information from the scene based on specific goals and values, making it inherently *subjective* as it reflects the agent's perspective. Given these definitions, the *reachable set* plays a crucial role in defining the situation because it encompasses the action capabilities

of both the agent and other dynamic elements. This helps in identifying the relevant components of the scene. Static elements are relevant if they fall within or affect the agent's reachable set, such as obstacles that hinder movement or road signs that establish rules to be applied within the agent's reachable set. Dynamic elements, with their own reachable sets, highlight potential interactions by indicating how other agents might influence the vehicle's trajectory and decisions.

III. PROPOSED APPROACH

A. Cost calculation

The first contribution of our work is to improve the computation of the reachable set by considering the agent's current state and its kinematic constraints in the cost function. Suppose the agent is on lanelet l_0 and aims to reach the following lanelet l_1 . The Lanelet2 library estimates the cost in terms of distance to reach l_1 from l_0 as $\frac{L_0+L_1}{2}$, where L_0 and L_1 are the lengths of l_0 and l_1 , respectively. However, this approximation ignores the agent's actual position on l_0 and assumes that l_1 is reached only at its center, which can be particularly inaccurate for long lanelets. To improve this, we incorporate the agent's longitudinal position x_0 on l_0 into the cost calculation. The actual travel distance to l_1 is then given by $L_0 - x_0$, reflecting the precise distance the agent must cover. For lanelets beyond l_1 , such as when evaluating the distance from l_1 to l_2 , we simply consider the full length of the centerline of l_1 , since the agent, upon reaching l_1 , must traverse it entirely to proceed to l_2 .

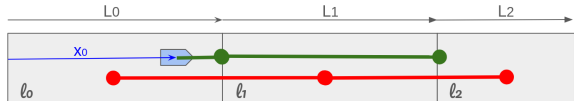


Fig. 2: Distance to reach following lanelets according to Lanelet2 library (red) and according to our method (green)

Regarding the cost of a lane change, from the current lanelet to an adjacent one, we consider the participant's distance from the centerline of the adjacent lanelet. For successive lane changes, a constant cost is applied.

Traditionally, the reachability problem is defined over a time interval $[0, T]$. The Lanelet2 library provides an estimate of the reachable set over a given time interval, assuming the agent always travels at the maximum allowed velocity. We enhance the precision of the maximum travel distance by incorporating the agent's acceleration limits and its actual kinematics. In its motion we identify two phases: the first involves the agent accelerating from its current speed v_{current} to the maximum allowed velocity v_{max} , and the second phase involves the agent maintaining this maximum velocity for the remaining time horizon. The maximum reachable distance d_{max} is computed as follows:

$$d_{\text{max}} = \begin{cases} v_0 \cdot T + \frac{1}{2} a_{\text{max}} \cdot T^2 & \text{if } T \leq t_{\text{acc}} \\ v_{\text{max}} \cdot (T - t_{\text{acc}}) + \frac{(v_{\text{max}}^2 - v_0^2)}{2a_{\text{max}}} & \text{if } T > t_{\text{acc}} \end{cases}$$

- v_0 is the agent's current velocity, used as the initial velocity in the prediction,

- v_{max} is the maximum allowed velocity,
- a_{max} is the maximum acceleration,
- T is the time horizon,
- $t_{\text{acc}} = \frac{v_{\text{max}} - v_0}{a_{\text{max}}}$ is the time required for the agent to reach the maximum velocity.

B. Physical rules

Our second contribution extends the reachability concept by integrating physical reachability alongside legal reachability, providing greater adaptability and enabling behaviors beyond strict rule-following. The Lanelet2 [7] library provides traffic rules based on local regulations to guide vehicles navigating the map. However, strict adherence to these rules can cause inefficiencies or even deadlocks in constrained environments. Human drivers adapt to situational contexts, occasionally relaxing rules to maintain traffic flow without compromising safety. For example, in Figure 3, a red and a

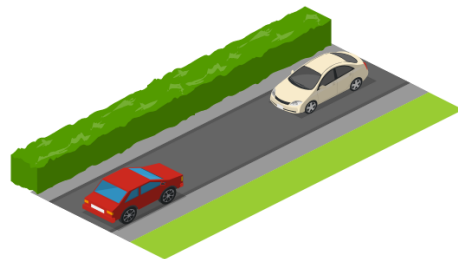


Fig. 3: Deadlock scenario resolved by relaxing traffic rules.

white car face a deadlock on a narrow street. While a hedge is non-navigable, the grassy area is physically accessible. Relaxing traffic rules lets the red car resolve the deadlock safely. Physical constraints are essential for understanding the reachability of not only the ego vehicle but also other road users.

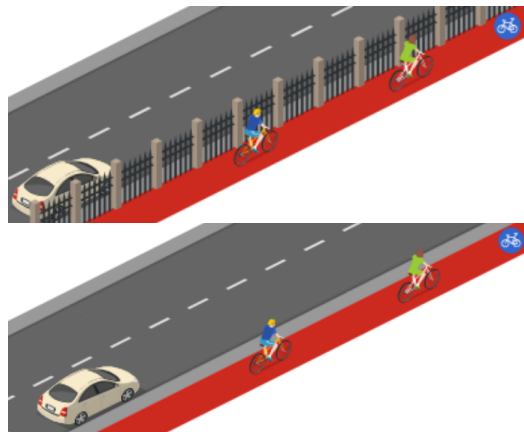


Fig. 4: Two different situations with same legally reachable space but different physical reachable space

Consider a scenario where cyclists ride in a bike lane adjacent to a vehicle's lane (Figure 4). According to Ulbrich's definition of a situation [3], cyclists' roles depend on the physical setup. Without a barrier, they may veer into the vehicle's lane to overtake or avoid obstacles, creating an overlap in their reachable spaces and potential interactions. In contrast, a fenced-off bike lane prevents such overlap, eliminating any influence on the vehicle's behavior. If only

legally reachable spaces are considered, both cases appear identical: cyclists are restricted to the bike lane, incorrectly ruling out interactions in the first scenario. Distinguishing between legal and physical reachability is therefore essential for accurate interaction assessment and autonomous vehicle behavior. A multi-perspective approach, balancing strict legal constraints with broader physical possibilities, allows autonomous vehicles to adapt dynamically, predict rule-breaking behavior, and exclude inaccessible areas.

Physical rules can be defined based on the semantic tags of lanelets, areas, and linestrings, reflecting the physical capabilities of the road participants. These rules specify:

- **Boundary passability:** Some boundaries (e.g., *high curbstone*, *wall*) are impassable for *vehicles*, while *pedestrians* can pass any kind of boundary present on the maps except *wall*.
- **Surface passability:** Certain surfaces (e.g., *sidewalk*, *stair*) are inaccessible for *vehicles*, while others (e.g., *traffic island*) are passable under relaxed traffic rules. For example, a *walkway* is accessible to *pedestrians* and *bicycles*, while *stairs* are only passable by pedestrians. Additionally, areas like *buildings* and *vegetation* are generally considered inaccessible to all participants.
- **Speed limits:** Speed limits are set based on the type of road participant and can be adjusted depending on the road type. In our testing environment, *pedestrians* have a speed limit of 11 km/h, *bicycles* 40 km/h, and *vehicles* 90 km/h on roads and 30 km/h on off-road surfaces.

The semantic tags of the map, which are the basis for these rules, can be provided as prior knowledge, coming from predefined map data. However, they can also be dynamically updated or refined based on real-time perception data. Visual recognition methods [14] can be used to identify navigable areas or obstacles and update the map accordingly. Using semantic tags allows for the definition of multiple sets of rules, enabling a layered interpretation of the problem. By classifying the reachable space according to the rules in place, it provides flexibility and allows the selection of the most appropriate set of rules based on the current situation.

IV. EXPERIMENTS

A. Dataset and baseline

To validate our method, we selected the roundD dataset [15], an open-source collection of real-world traffic data recorded at three roundabouts in Germany. The dataset captures a highly interactive driving environment, where participants do not always follow traffic rules, such as vehicles stopping on traffic islands, pedestrians on the roadway, and bicycles on pedestrian sidewalks. Its open-source nature promotes reproducibility and comparisons with other methods. We compared our method with the baseline provided by the Lanelet2 framework [6], which utilizes a routing graph based on German traffic rules to estimate reachable sets in terms of distance and travel time. From this point onwards, we refer to this as the baseline method. Regarding the works from the Technical University of Munich (TUM) [9], [10],

they were not included in our comparison as they focus on the geometric aspects of reachable space rather than accessibility rules. Nonetheless, their geometric approach could be combined with ours to further refine the reachable set of lanelets and areas by geometrically identifying which portions are truly reachable.

B. Metrics

To demonstrate that including the current state and kinematic limits of the participant improves the precision and reliability of the *reachable set*, we evaluate its performance using the following metrics.

a) **Precision:** quantifies the proportion of correctly identified lanelets or areas (*TP*) within the *reachable set* relative to all identified areas:

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP})$$

where FP represents primitives incorrectly identified as reachable.

b) **Recall:** measures the proportion of actual reachable primitives (*TP*) correctly predicted within the *reachable set*:

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN})$$

with FN denoting primitives that were reached but not predicted. High recall ensures comprehensive coverage of potential trajectories.

c) **Trajectory Containment:** evaluates the percentage of the agent’s actual trajectory, within the predicted time interval (3 seconds), that lies inside the *reachable set*. Lower values indicate overly strict constraints that fail to capture feasible actions.

C. Experimental setup

Our experiments were conducted using the ROS 2 [16] framework. For visualizing the results, we utilized its built-in visualization tool, RViz 2, which facilitated a comprehensive qualitative assessment of the system’s performance.

TABLE I: Maximum accelerations by participant type

Participant Type	Maximum Acceleration (m/s ²)
Car	10
Truck	11
Bus	8
Van	7
Motorcycle	5
Bicycle	2.5
Pedestrian	1.7

We chose acceleration limits that are reasonable based on the observed behaviors in the dataset. These kinematic constraints are incorporated into our method to refine the reachable set. Table I lists the maximum acceleration values for each participant type, which serve as key parameters in the model. Legal speed limits are taken directly from the *speedLimit* tag, while the physical speed limits are defined in the physical rules definition described in III-B. The goal is to estimate the reachable set within the next 3 seconds for each participant in the scene, outperforming the baseline method.

D. Qualitative Results

We highlight two examples¹ where our method shows a significant improvement over the baseline, demonstrating how it better adapts to real-world dynamics.

Example 1: In Figure 5, we show lanelet reachability predictions for a pedestrian (participant ID 2, frame 25409, recording 00). The baseline method (Figure 5a) fails to identify reachable lanelets, assuming the participant is always at the center of the lanelet. If the lanelet is too long compared to the distance the participant can cover within the considered time interval, the next lanelet is never deemed reachable. Our method (Figure 5b), by considering the participant’s actual position, correctly identifies the next reachable lanelet, offering more accurate predictions.

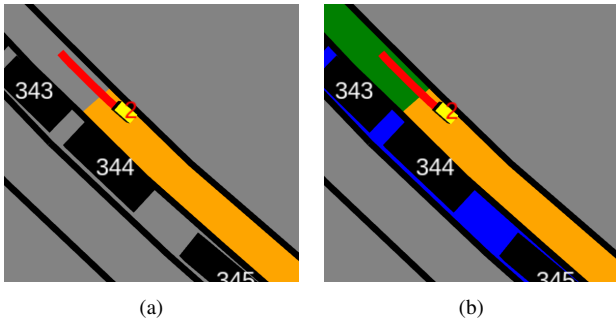


Fig. 5: Example 1: Starting from the matching lanelet (orange), the baseline (a) misses reachable lanelets, while our method (b) correctly identifies reachable lanelets (blue: physical, green: legal), with the pedestrian’s future trajectory (red) inside the reachable set.

Example 2: In Figure 6 a cyclist is located in an inner part of the roundabout that is not considered legally accessible. As a result, the baseline approach (Figure 6a) fails to identify any reachable lanelets, as it is unable to find a legal path from that illegal starting point. Our method (Figure 6b) also does not find any legally reachable lanelets, but it successfully identifies the physically reachable set, which contains the cyclist’s future trajectory. This highlights the advantage of incorporating physical constraints even when legal ones are not satisfied.

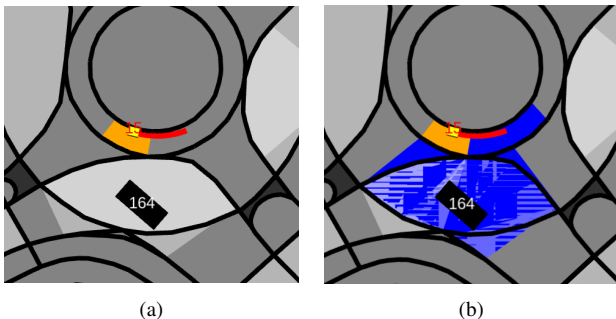


Fig. 6: Example 2: The baseline (a) fails to predict a reachable set considering only traffic rule constraints, whereas our method (b) successfully identifies the physically reachable set (blue) that contains the future trajectory (red) of the cyclist.

¹<https://youtu.be/e7bLZ1Mdtco>

E. Quantitative Results

Our method was tested on all 24 recordings from the round dataset [15], featuring a total of 13,740 road users, to extensively evaluate its performance across a variety of real-world scenarios. We expect that incorporating the vehicle’s position within the lanelet will enhance recall by allowing a more accurate assessment of whether a lanelet is reachable. Additionally, applying kinematic limits will improve precision by restricting the reachable set, thereby excluding unfeasible trajectories. In fact, by considering the same traffic rules and changing only the cost calculation, we can observe from Figure 7 that our method outperforms the baseline in both metrics across all recordings.

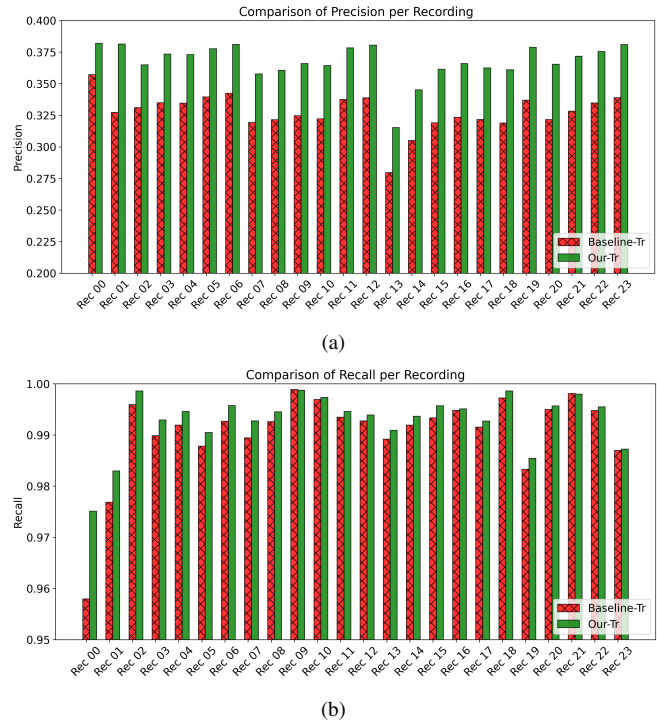


Fig. 7: Precision (a) and Recall (b) trends across recordings in the round dataset considering traffic rules for both baseline (red) and our method (green)

The recall improvement is more evident in the first two recordings, which use maps with longer lanelets. However, despite this improvement, we observe that the estimated reachable set does not fully cover the actual reached set. This discrepancy occurs because, in some cases, the traffic rules used to construct the routing graph were not strictly followed by the agents. These instances emphasize the importance of complementing traffic rules with physical reachability constraints to better account for deviations from expected behavior and ensure more accurate modeling. By considering the physical reachability rules, we observe that the actual trajectory of the participants is contained within the estimated reachable set. In some cases, particularly in recording 13, we see a clear improvement, with a 13% increase in coverage (Figure 8), due to a higher percentage of rule violations by road users in this particular scene. This result underscores the importance of using a multi-level description of the

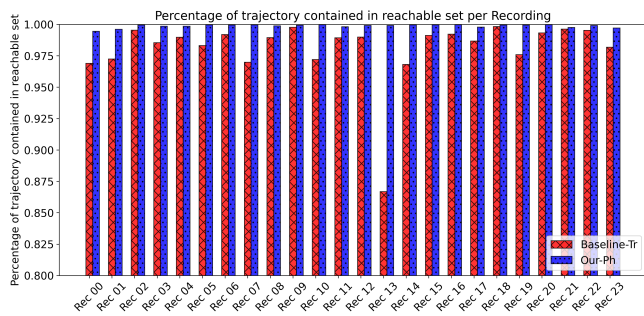


Fig. 8: Percentage of future trajectory included in the reachable set predicted by the baseline method under traffic rules (red) and our method under physical rules (blue)

space surrounding the agent to ensure all possible actions are considered.

TABLE II: Mean and Standard Deviation for Each Metric

Method	Precision	Recall	Traj. Contained
Baseline-Tr	0.327 ± 0.015	0.991 ± 0.008	0.981 ± 0.026
Our-Tr	0.368 ± 0.014	0.993 ± 0.005	0.983 ± 0.025
Our-Ph	0.200 ± 0.008	0.999 ± 0.004	0.999 ± 0.001

The results, summarized in Table II, demonstrate that the legally reachable set obtained using our proposed method outperforms the corresponding set generated by the baseline method across all evaluation metrics. When computing the reachable set based on physical rules, we observe a decrease in precision, as this set is generally larger than the legally reachable set. Nevertheless, it provides 99.9% coverage, ensuring with high confidence that the trajectory lies within its bounds, nearly reaching theoretical completeness. The slight discrepancy can be attributed to the approximation of a constant cost for lane changes and to the current level of map discretization, which may not fully capture certain connections. While we used the maps provided by the dataset for these experiments, we believe that a more careful consideration of the map design, particularly in modeling critical connections and refining the discretization in specific areas, could further enhance the accuracy of our approach. Experiments were conducted on a workstation with an Intel Xeon W-2223 CPU (4 cores, 8 threads, 3.6 GHz). Our method initializes each routing graph in 1.27 s (vs. 0.3 s for the baseline), and since this occurs only once per map, it has minimal impact on runtime. During execution, the computation of each reachable set requires 0.755 ms versus 0.639 ms for the baseline, and since these calculations can be parallelized, the runtime remains efficient even with multiple agents. This demonstrates that our approach improves prediction quality while maintaining real-time suitability.

V. CONCLUSIONS

In this work, we have proposed an innovative approach to accurately describe the space around road users at the lanelet level, using multiple reachability criteria and integrating real-time data on the agent’s kinematics and its constraints. The results of our experiments on real-world data confirm the effectiveness of our method in describing the navigation of road agents. Thanks to the modularity and semantic

richness offered by the Lanelet2 library and its map format, our approach provides a more flexible and context-aware representation of the environment. Future developments will focus on the automatic identification of scenarios where traffic rules are more likely to be relaxed, in order to predict which parts of the reachable set are most likely to be effectively reached. This will allow us to explore how human drivers dynamically relax certain rules, adapting to specific traffic situations. Additionally, we plan to extend our method to support more informed, safe, and effective decision-making in driving scenarios, particularly those where strictly conservative behavior is insufficient.

ACKNOWLEDGMENT

This research is part of the ANNAPOLIS project (<https://project.inria.fr/annapolis/>), funded by the French National Research Agency (ANR-21-CE22-0014).

REFERENCES

- [1] H. Y. Makahleh, E. J. S. Ferranti, and D. Dissanayake, “Assessing the role of autonomous vehicles in urban areas: A systematic review of literature,” *Future Transportation*, vol. 4, no. 2, pp. 321–348, 2024.
- [2] E. Ziraldo, M. E. Govers, and M. Oliver, “Enhancing autonomous vehicle decision-making at intersections in mixed-autonomy traffic: A comparative study using an explainable classifier,” *Sensors*, vol. 24, no. 12, 2024.
- [3] S. Ulbrich, T. Menzel, A. Reschka, F. Schuldt, and M. Maurer, “Defining and substantiating the terms scene, situation, and scenario for automated driving,” in *IEEE ITSC*, 2015, pp. 982–988.
- [4] J. Knaup and K. Homeier, “Roadgraph - graph based environmental modelling and function independent situation analysis for driver assistance systems,” in *IEEE ITSC*, 2010, pp. 428–432.
- [5] S. Ulbrich, T. Nothdurft, M. Maurer, and P. Hecker, “Graph-based context representation, environment modeling and information aggregation for automated driving,” in *IEEE IV*, 2014, pp. 541–547.
- [6] F. Poggenhans, J.-H. Pauls, J. Janosovits, S. Orf, M. Naumann, F. Kuhnt, and M. Mayr, “Lanelet2: A high-definition map framework for the future of automated driving,” in *IEEE ITSC*, 2018, pp. 1672–1679.
- [7] F. Poggenhans and J. Janosovits, “Pathfinding and routing for automated driving in the lanelet2 map framework,” in *IEEE ITSC*, 2020, pp. 1–7.
- [8] R. Dempster, M. Al-Sharman, Y. Jain, J. Li, D. Rayside, and W. Melek, “Drg: A dynamic relation graph for unified prior-online environment modeling in urban autonomous driving,” in *IEEE ICRA*, 2022, pp. 8054–8060.
- [9] M. Koschi and M. Althoff, “Spot: A tool for set-based prediction of traffic participants,” *IEEE IV*, pp. 1686–1693, 2017.
- [10] M. Koschi and M. Althoff, “Set-based prediction of traffic participants considering occlusions and traffic rules,” in *IEEE T-IV*, vol. 6, no. 2, 2021, pp. 249–265.
- [11] S. M. LaValle, *Planning Algorithms*. Cambridge, U.K.: Cambridge University Press, 2006, available at <http://planning.cs.uiuc.edu/>.
- [12] E. Yip and R. Sinovec, “Solvability, controllability, and observability of continuous descriptor systems,” *IEEE Trans. Autom. Control*, vol. 26, no. 3, pp. 702–707, 1981.
- [13] I. Hwang, D. Stipanovic, and C. Tomlin, “Polytopic approximations of reachable sets applied to linear dynamic games and a class of nonlinear systems,” *Systems and Control: Foundations and Applications*, pp. 3–19, 01 2005.
- [14] W. Li, M. Liao, and W. Zou, “A progressive segmentation network for navigable areas with semantic-spatial information flow,” *Expert Systems with Applications*, vol. 261, p. 125465, 2025.
- [15] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, “The round dataset: A drone dataset of road user trajectories at roundabouts in germany,” in *IEEE ITSC*, 2020, pp. 1–6.
- [16] S. Macenski, T. Foote, B. Gerkey, C. Lalancette, and W. Woodall, “Robot operating system 2: Design, architecture, and uses in the wild,” *Science Robotics*, vol. 7, no. 66, p. eabm6074, 2022.