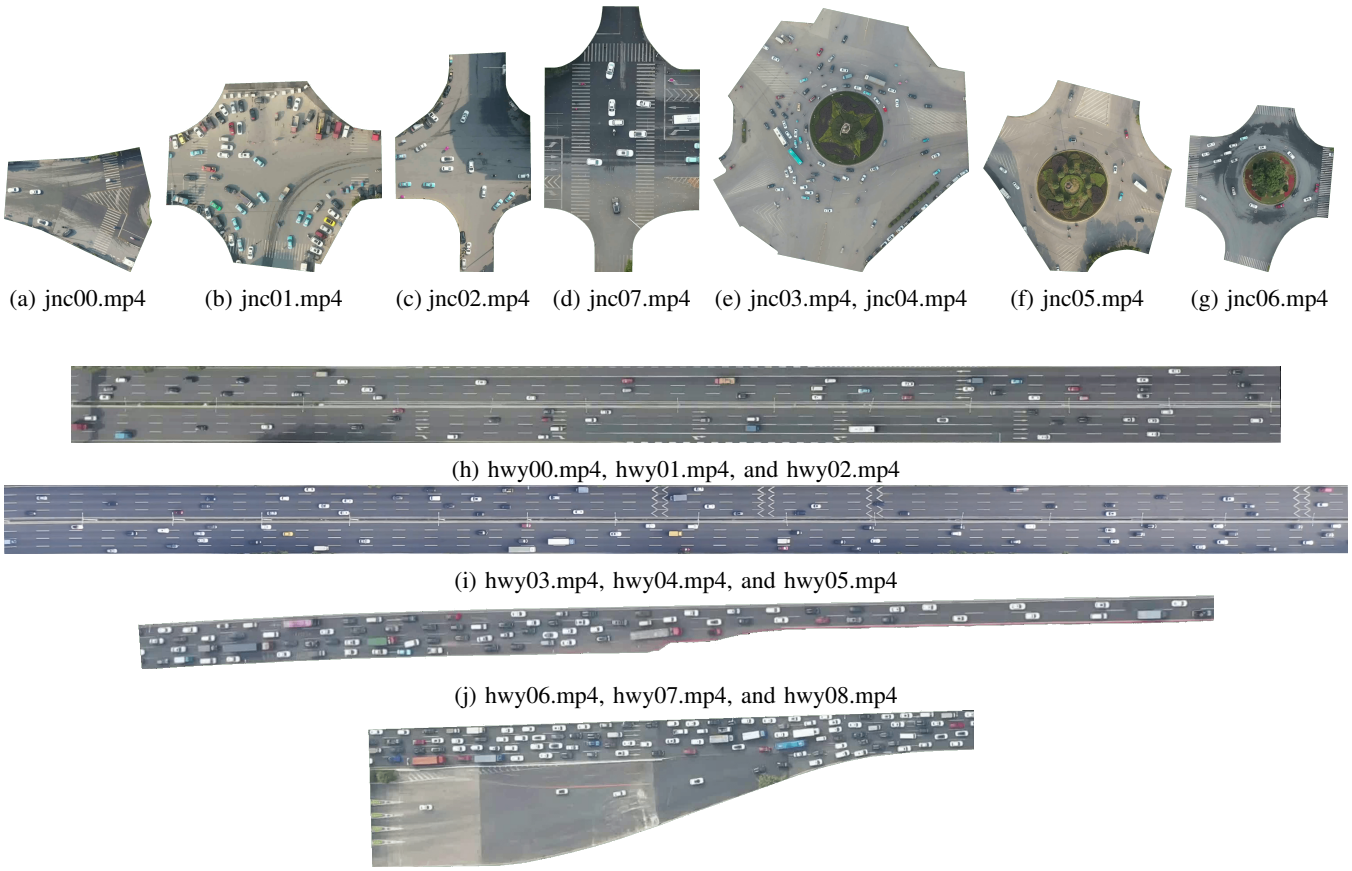
Decentralized Vehicle Coordination: The Berkeley DeepDrive Drone Dataset and Consensus-Based Models

Fangyu Wu[1,2], Dequan Wang[2], Minjune Hwang[2], Chenhui Hao[3], Jiawei Lu[2], Jiamu Zhang[2], Christopher Chou[2], Trevor Darrell[2], and Alexandre Bayen[2]



(a) jnc00.mp4 (b) jnc01.mp4 (c) jnc02.mp4 (d) jnc07.mp4 (e) jnc03.mp4, jnc04.mp4 (f) jnc05.mp4 (g) jnc06.mp4

(h) hwy00.mp4, hwy01.mp4, and hwy02.mp4

(i) hwy03.mp4, hwy04.mp4, and hwy05.mp4

(j) hwy06.mp4, hwy07.mp4, and hwy08.mp4

(k) hwy09.mp4, hwy10.mp4, and hwy11.mp4

Fig. 1: We propose a specialized dataset—Berkeley DeepDrive (B3D) dataset—and a modeling framework for studying decentralized vehicle coordination. The dataset, open sourced at **https://github.com/b3d-project/b3d**, captures decentralized vehicle coordination on understructured roads. The modeling framework provide a novel perspective to distributed motion planning over networks.

***Abstract*— A significant portion of roads, especially in densely populated developing countries, lacks explicitly defined right-of-way rules. These understructured roads pose substantial challenges for motion planning in autonomous vehicles, where efficient and safe navigation depends on understanding how human drivers coordinate to avoid collisions in decentralized settings. This coordination, often referred to as "social driving etiquette," remains underexplored due to a lack of empirical data and suitable modeling frameworks. In this paper, we introduce a novel dataset and modeling framework tailored to study motion planning in understructured environments. The dataset comprises 20 aerial videos of representative understructured road scenarios, an image dataset for training vehicle detection models, and a development kit for estimating vehicle trajectories. We show that a consensus-based modeling**

*This work was supported by Berkeley DeepDrive.

[1]Department of Electrical and Computer Engineering at Cornell University, [2]Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley, and [3]Department of Civil and Environmental Engineering at the University of California, Berkeley

approach can explain the emergence of priority orders observed in the dataset, supporting robust, real-time collision avoidance planning.

I. Introduction

Navigation in built environments—such as driving, cycling, and walking—is a crucial area of research in autonomous driving and human-robot interaction. The development of autonomous agents capable of operating in *structured* environments is a well-established field with roots extending back to the early days of control theory and robotics. In the context of transportation, researchers have extensively studied navigation in structured environments, such as free-flow highways and signalized urban streets. However, compared to navigation on structured roads, motion planning in *understructured* road environments—roads without explicitly defined right-of-way regulations—is much less explored due to the lack of empirical data and the complexity of the problem.

The first essential ingredient for understanding understructured navigation is empirical data. To this end, camera videos are particularly effective because *1*) they capture rich dynamics on roads at a relatively low cost, *2*) they allow for quantitative assessment through direct inspection, and *3*) they enables qualitative analysis through modern computer vision. Despite extensive past research, most existing video datasets focus solely on driving behaviors in structured environments. Behaviors in understructured road environments—such as crowded highways with frequent merges and unsignalized intersections—have rarely been surveyed. The scarcity of data for this problem undoubtedly hinders understanding of navigation in such environments.

To bridge this gap in empirical data, we propose the Berkeley DeepDrive Drone (B3D) dataset. This inertial-framed dataset records rich dynamics of driving behaviors in understructured road environments, including unsignalized intersections, unsignalized roundabouts, highways with collisions, highways with stop-and-go waves, and highways with merging bottlenecks, as shown in Figure 1. To the best of our knowledge, it is the first drone dataset to date that extensively covers understructured driving behaviors.

The other piece of the puzzle is a suitable modeling paradigm. The conventional control and planning architecture in an autonomous vehicle consists of four layers of abstraction, from top to bottom: *1*) routing, *2*) behavioral decision-making, *3*) motion planning, and *4*) vehicle control [14]. Particularly in the second behavioral layer, a prediction module is often employed to forecast the motion of surrounding vehicles, around which the motion planning layer then plans to avoid collisions.

This classical *predict-then-plan* paradigm, while effective for driving in structured environments, is inadequate in understructured road environments. For instance, at an unsignalized intersection, drivers negotiate the right-of-way dynamically: when two conflicting vehicles approach the intersection at similar times, the one perceived as more aggressive often "wins" the priority of passage. This negotiative process fundamentally deviates from the predict-then-plan paradigm.

Modeling such a negotiative process is fundamental to autonomous driving. Given the vast population of drivers in developing countries, where roads are often unsignalized, understanding how humans operate in these environments is crucial for enhancing the generalizability of autonomous driving technologies. This knowledge enables autonomous vehicles to mimic human drivers, facilitating navigation through understructured road environments. Furthermore, it may inspire innovative designs of decentralized motion planning algorithms, applicable not only to autonomous vehicles but also to other forms of mobile robots.

To this end, we propose a novel planning approach for the behavioral decision-making layer, replacing the traditional predict-then-plan approach with a *negotiate-then-commit* approach. Central to this new planning paradigm is a consensus model that implicitly arbitrates the priority of agents with conflicting paths. This consensus model is founded on the principle of least action, a universal law observed in nature.

In summary, we list the main contributions of our work below.

- We release the **B3D dataset**, which captures rich dynamics of driving behaviors on understructured roads.
- We propose a **consensus-based modeling framework** based on the principle of least action.
- We evaluate our framework through **data-driven validation and integrative simulation**.

II. Related Works

Existing *driving datasets* can be classified into either *body-framed* dataset or *inertial-framed* dataset. The body-framed datasets have cameras placed on the traffic participants, e.g., on top a survey vehicle, to observe the movements of the surrounding traffic. The inertial-framed datasets have cameras installed at some overhead positions above some roads of interest to observe all road occupants within its field of view.

Body-framed datasets are useful to study behaviors of the surrounding traffic participants with respect to the host. It is arguably the most commonly used method for autonomous driving perception research. Many well-known datasets fall within this category, such as the KITTI dataset [7], the Oxford RoboCar dataset [13], the Cityscapes dataset [5], the Waymo Open dataset [21], and the Waymo Open Motion dataset [6]. Nevertheless, the body-framed placement makes it difficult to study persistent traffic patterns within a fixed spatial range. Moreover, placing the sensors on a low-profile moving object also results in undesirable occlusion.

Inertial-framed datasets are better suited to observe traffic within a fixed spatial range. By construction, the inertial-framed placement makes it very easy to estimate the recorded vehicle and human movements with respect to the ground. Because the cameras are often placed on a vantage point, occlusion is also not as problematic. In this category, we also find many seminal datasets, including the NGSIM dataset [1],

the ARED dataset [26], the Stanford Drone dataset [17], the highD dataset [10], and the INTERACTION dataset [29].

Existing *motion planning* literature for autonomous driving can be broadly divided into three major categories of methods: *prediction-based* approaches, *dynamics-based* approaches, and *end-to-end* approaches. Prediction-based planning deals with dynamically changing environments by predicting future states of the environment and planning based on those static predictions. In contrast, dynamics-based planning maintains a model of the world and plans interactively based on that model. Lastly, unlike the above two methods, end-to-end planning replaces conventional modular motion planning stack with a unified neural network model.

Prediction-based approaches are suitable for planning in environments where future states can be predicted with high confidence. The predictability of such environments is characterized by the degree of confidence in forecasting the system's trajectory over a suitable time horizon. For navigation in reasonably predictable environments, robust methods have been developed for planning around agents whose motions have bounded uncertainty [8], follow known probability distributions with chance constraints [12], [25], or are amenable to be described within a distributionally robust framework [15], [19].

Dynamics-based approaches are particularly suitable for highly coupled environments where the future state depends on the ego agent's current actions. Within these approaches, consensus-based methods form a special class where all agents agree on a common navigation logic. Such models are widely applied in driver and pedestrian behavior modeling [9], [23], air traffic control [22], [4], and swarm robotics [16], [24]. In contrast, non-consensus-based methods allow the ego agent and environmental agents to operate on different models, such as [18].

Recently, end-to-end approaches have gained significant attention in the community, driven by advances in machine learning, upgrade in computing infrastructure, and the availability of large-scale driving datasets. For example, viable driving paths have been planned directly from LiDAR, GPS, IMU, and navigation map overlays [2], or from on-vehicle cameras and past vehicle states [28]. The pros and cons of these approaches are clear: while they offer significant potential and scalability with large datasets and computing power, it lacks theoretical tractability and safety guarantees. For a comprehensive overview of the literature in this area, please refer to [3] and the references therein.

III. Berkeley DeepDrive Drone Dataset

To study decentralized vehicle coordination in understructured environments, we introduce the Berkeley DeepDrive Drone (B3D) dataset, available at **https://github.com/b3d-project/b3d**. The dataset was recorded with a DJI Mavic 2 Pro quadcopter between December 11 and December 21 of 2019 in China. It consists of 20 post-processed aerial drone videos, 16002 annotated images, and a development kit for estimating vehicle trajectories from the videos. The total size of the dateset is about 86.3 GB. We briefly describe the components of the dataset below.

A. Aerial Videos

Among the 20 processed aerial videos, eight were recorded on top of junctions and 12 on top of highways. An overview of the types of the roads covered in the videos is illustrated in Figure 1. Scenarios recorded in the video can be classified into the following six categories: *1*) unsignalized intersections, *2*) unsignalized roundabouts, *3*) tailgating accidents, *4*) stop-and-go waves, *5*) roadwork-induced merging, and *6*) ramp-induced merging.

Unsignalized intersections can be found in videos jnc00.mp4, jnc01.mp4, jnc02.mp4, and jnc07.mp4. Videos jnc00.mp4 and jnc01.mp4 are two variants of three-way intersections, as shown in Figure 1a and Figure 1b, respectively. Videos jnc02.mp4 and jnc07.mp4 are two variants of four-way intersections, as shown in Figure 1c and Figure 1d, respectively.

Unsignalized roundabouts are captured in jnc03.mp4, jnc04.mp4, jnc05.mp4, and jnc06.mp4. Videos jnc03.mp4 and jnc04.mp4 are two recordings of a five-way roundabout, as shown in Figure 1e. Videos jnc05.mp4 and jnc06.mp4 are two variants of four-way roundabouts, as shown in Figure 1f and Figure 1g, respectively. Compared to video jnc03.mp4, video jnc04.mp4 has slightly more traffic.

The *tailgating accidents* consist of two collision events, first in hwy00.mp4 and then in hwy01.mp4. At 00:45 of hwy00.mp4, we observe the first accident near the left margin of the frame, as shown in Figure 3a. At 13:10 of hwy01.mp4, we find another traffic accident in the middle of the frame, as shown in Figure 3b. Video hwy02.mp4 captures the resulting congested traffic induced by the second incident. The the timestamps of the collision events are visualized in Figure 4.

(a) First collision in hwy00.mp4

(b) Second collision in hwy01.mp4

Fig. 3: Tailgating collisions in hwy00.mp4 and hwy01.mp4. The collided vehicles are circled in black. The first accident involves at least two vehicles, while the second incident involves four vehicles.

Stop-and-go waves are recorded in hwy04.mp4 and hwy05.mp4. The first stop-and-go wave forms between 02:30 and 05:07 of hwy04.mp4. The second stop-and-go wave emerges between 06:06 and 08:10 of hwy04.mp4. The third stop-and-go wave is observed between 10:26 and 12:25 of hwy04.mp4. The fourth wave happens between 00:00 and 01:33 of hwy05.mp4. The last visible wave occurs between 05:19 and 06:07 of hwy05.mp4. For comparison, we provide video hwy03.mp4 as a free-flow baseline. The formation and

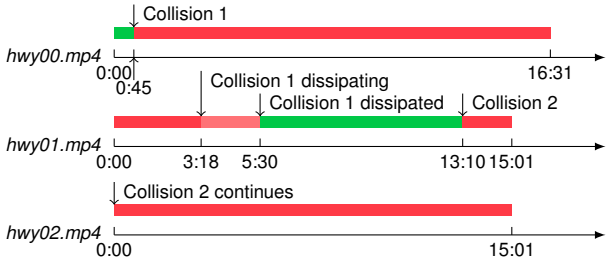


Fig. 4: Timeline of the *tailgating accidents*. Green indicates regular traffic. Red indicates congestion caused by a collision. Light red indicates the induced congestion starts dissipating.

dissipation events of the stop-and-go waves are visualized in Figure 5.

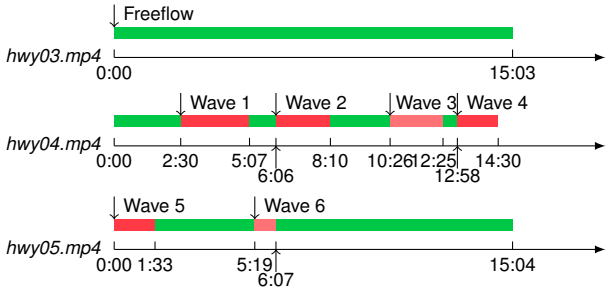


Fig. 5: Timeline of the *stop-and-go waves*. Green indicates regular traffic. Red indicates congestion caused by a strong stop-and-go wave. Light red indicates congestion caused by a weak stop-and-go wave.

Roadwork-induced merging is recorded in hwy06.mp4, hwy07.mp4, and hwy08.mp4. The topology of the scenario is a four-lane-to-two-lane bottleneck, as shown in Figure 1j. Persistent congestion is observed before the merge point, while free-flow traffic is formed after the merge point.

Ramp-induced merging is recorded in hwy09.mp4, hwy10.mp4, and hwy11.mp4. The topology of the ramp is shown in Figure 1k, where a three-lane on-ramp is merging into a four-lane congested highway. The traffic stays congested *before and after* the merge point.

B. Annotated Images

In addition to the videos, we build an image dataset, which can be used to train vehicle detection models for trajectory estimation. The dataset consists of 16002 annotated images, 80% of which is split for training, 10% of which for validation, and 10% of which for testing.

Using annotation tool CVAT [20], a total of 135303 axially aligned bounding boxes (AABB) are annotated for junction images and a total of 129939 AABBs created for highway images. Note that the annotations contain only one object class, that is, the *vehicle* class. We do not distinguish large vehicles, such as buses and trucks, from small vehicles, such as sedans and SUVs.

C. Development Kit

Besides, we also provide a development kit consisting of three example scripts: train .py, test .py, and mask.py. For reproducibility, we encapsulate the development environment in a Docker image. The image can be directly built from the provided Dockerfile in the dataset repository.

The script train .py is used to show how the annotated image data may be used to train a neural network model for vehicle detection. In this script, we use the object detection library Detectron2 [27] to train a RetinaNet model [11] for detecting locations of vehicles in an input image.

The script test .py applies the trained model, trained via the train .py script, to an input image. This script is intended to serve as an example for post-training evaluation and inference. For convenience, we provide an optional pre-trained model, which can be used directly for inference. This way, users can directly use a working detection model to estimate vehicle trajectories without having to go through the computationally expensive training step locally.

Finally, the script mask.py crops an image according to a pre-defined polygon mask. The script intends to help users to focus on the only relevant part of the scene, where relevance is defined by the user via CVAT. To crop a video, one only needs to define a polygonal mask for *one* frame of a video and then apply masking to *every* frame of the video.

IV. Consensus-Based Models

To describe the decentralized vehicle coordination observed in the B3D dataset, we propose a modeling framework based on the idea of consensus.

We consider a setup with N agents and M pairs of intersecting paths. Denote the road network by $G = (V, E)$, where V represents the set of vertices and E the set of edges. A path p is defined as a finite sequence of directed edges connecting a set of unique vertices.

Let $s \in \mathbb{R}_{\geq 0}$ represent the traveled distance along path p. Define the state of an agent as $\mathbf{x} \coloneqq [s \quad \dot{s}]^\top$. Assuming the agent begins its path at t_0 and completes it at t_f, a *traversal policy* $\pi : \mathbb{R} \times \mathbb{R}^2 \to \mathbb{R}^2$ is a function defined as $\dot{\mathbf{x}}(t) = \pi(t, \mathbf{x}(t))$ over $t \in [t_0, t_f]$. The trajectory of an agent is defined as the evolution of an agent's state over time, i.e., $X = \{\mathbf{x}(t) \mid \forall t \in [t_0, t_f]\}$.

Two traversal policies $\pi_1 : [t_0, t_f] \to \mathbb{R}^2$ and $\pi_2 : [t_0, t_f] \to \mathbb{R}^2$ are considered collision-free if the resulting trajectories $\mathbf{x}_1(t)$ and $\mathbf{x}_2(t)$ remain sufficiently separated according to some distance function $d : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}_{\geq 0}$ for all $t \in [t_0, t_f]$. Specifically, we require that $d(\mathbf{x}_1(t), \mathbf{x}_2(t)) \geq \epsilon$ for a certain separation margin $\epsilon > 0$ and for all t.

With the preceding definition, we are now equipped to present the two-agent collision avoidance problem as follows:

Problem 1: (Two-Agent Collision Avoidance Problem) Consider two agents of paths p_1 and p_2 intersecting at either one vertice or a sub-path. Given the initial conditions $(t_0, \mathbf{x}_{0|1})$ and $(t_0, \mathbf{x}_{0|2})$, find a pair of traversal policies, π_1 and π_2, such that they establish a priority order at the conflicting location with finite arrival times $t_{f|1}$ and $t_{f|2}$.

The above two-agent problem is an special case of the more general N-agent problem, as formally stated below:

Problem 2: (N-Agent Collision Avoidance Problem) Consider N agents of paths $P = \{p_1, p_2, \ldots, p_N\}$ intersecting at either one vertice or a sub-path. Given the initial conditions $(t_0, \mathbf{x}_{0|i}), \forall i = 1, \ldots, N$, find some traversal policies $\pi_1, \pi_2, \ldots, \pi_N$, such that they solve the corresponding two-agent collision avoidance problem for each unique pair of paths in P.

A. Collision Avoidance Consensus

A consensus model, informally, refers to a control policy that, when deploy to individual agents, steers each agent towards consensus on certain critical quantities despite initial disagreements. For Problem 1 and Problem 2, the critical quantity of interest here is the priority order at all conflicting locations. In following paragraphs, we present a general framework on consensus-based conflict resolution.

Without loss of generality, we start by first considering Problem 1. Label the two agents as Agent 1 and Agent 2. Define a binary indicator $r_i \in \{0, 1\}$ at the conflicting location for agent $i \in \{1, 2\}$ such that $r_i = 1$ when Agent i has the priority and $r_i = 0$ otherwise. For the two-agent problem, a priority order is established if and only if $r_1 + r_2 = 1$.

Consequently, we assume a *consensus model* for collision avoidance $\sigma : \mathbb{R} \times \mathbb{R}^2 \times \mathbb{R}^2 \to \{0, 1\}$ to take the following form:

$$r = \sigma(t, \mathbf{x}(t), \bar{\mathbf{x}}(t)), \tag{1}$$

where t is the time of evaluation and $\mathbf{x}(t), \bar{\mathbf{x}}(t)$ the states of the ego agent and the competing agent, respectively. Clearly, a valid consensus model satisfies $\sigma(t, \mathbf{x}_1(t), \mathbf{x}_2(t)) + \sigma(t, \mathbf{x}_2(t), \mathbf{x}_1(t)) = 1$, where $\mathbf{x}_1(t), \mathbf{x}_2(t)$ are the states of Agent 1 and Agent 2.

Now consider the general N-agent version in Problem 2. By similar argument, we must have $\sigma(t, \mathbf{x}_p(t), \mathbf{x}_q(t)) + \sigma(t, \mathbf{x}_q(t), \mathbf{x}_p(t)) = 1$, where $\mathbf{x}_p(t), \mathbf{x}_q(t)$ are the states of any unique pair of Agent p and Agent q selected from the N agents.

Altternatively, we can define $r_i \in \{0, 1, \ldots, N-1\}$ at the conflicting location for Agent $i \in \{1, \ldots, N\}$ such that the set $R = \{r_i : \forall i = 1, \ldots, N\}$ is a permutation of the ordered set $\{0, 1, \ldots, N-1\}$. The priority in which agent i clears the conflicting location is denoted by r_i, i.e., if $r_i > r_j$, then Agent i clears the conflicting location *earlier* than Agent j.

Next, we propose a specific form of consensus model based on the principle of least action.

B. Least-Action Consensus

To develop a least-action consensus model, one must first quantify the cost of an action associated with each conflict resolution scheme. To this end, a common approach is to measure the cost of an action through a cost functional J on the trajectories resulting from a given priority order.

For example, we can define a L^2-norm induced cost functional. Given a priority order $R = \{r_i : \forall i = 1, \ldots, N\}$ and initial conditions $I = \{(t_0, \mathbf{x}_{0|i}) : \forall i = 1, \ldots, N\}$ for the N agents respectively, we can find a set of valid traversal policies $\Pi_{R,I} = \{\pi_i : \forall i = 1, \ldots, N\}$. Let $T^*_{R,I} = \{[t_i^-, t_i^+] : i = 1, \ldots, N\}$ be the set of time intervals, during which the respective agents interact with their neighboring agents. With R, I, T^*, one can therefore define the cost functional J as follows:

$$J(R, I, \Pi_{R,I}, T^*_{R,I}) = \sum_{i=1}^{N} \left(\int_{t_i^-}^{t_i^+} \pi_i^2(t, \mathbf{x}_i(t))\, dt \right)^{\frac{1}{2}}, \tag{2}$$

for $[t_i^-, t_i^+] \in T^*_{R,I}, \pi_i \in \Pi_{R,I}$ for all $i = 1, \ldots, N$.

In Problem 1, if one can associate a cost functional with each valid choice of traversal policies π_1, π_2 and initial conditions $(t_0, \mathbf{x}_{0|1})$ and $(t_0, \mathbf{x}_{0|2})$, then a least-action consensus model favors the choice associated with the least cost. With the cost functional J, this leads to solving the following optimization problem:

$$\begin{aligned} R^* = \arg\min_{R} \quad & J(R, I, \Pi_{R,I}, T^*_{R,I}) \\ \text{s.t.} \quad & r_1 + r_2 = 1, \\ & r_1, r_2 \in \{0, 1\}. \end{aligned} \tag{3}$$

Let $\sigma(t_0, \mathbf{x}_{0|1}, \mathbf{x}_{0|2}) \coloneqq r_1^*$ and $\sigma(t_0, \mathbf{x}_{0|2}, \mathbf{x}_{0|1}) \coloneqq r_2^*$. Clearly, if the optimization (3) admits an unique solution, the resulting consensus model ensures zero collision.

Generalizing to Problem 2, we find the total number of possible orderings extends to $N!$. Using the same cost functional, the consensus model enumerates the costs for each possible orderings and select the one with the least amount of costs. , as defined below:

$$\begin{aligned} R^* = \arg\min_{R} \quad & J(R, I, \Pi_{R,I}, T^*_{R,I}) \\ \text{s.t.} \quad & R \in S(\{0, 1, \ldots, N-1\}), \end{aligned} \tag{4}$$

where $S(\{0, 1, \ldots, N-1\})$ is the set of all permuations of $\{0, 1, \ldots, N-1\}$. Let $\sigma(t_0, \mathbf{x}_{0|i}, \mathbf{x}_{0|j}) \coloneqq \mathbb{1}(r_i > r_j)$ for all $i = 1, \ldots, N$. As expected, optimization (3) is a special case of optimization (3), where $N = 2$. Like before, if the optimization (4) admits an unique solution, the resulting consensus leads to zero collision.

Remark 1: The time interval $[t_i^-, t_i^+]$ marks the begining and the end of Agent i's interaction with other conflicting agents. When setting the time interval, one can choose t_i^- as a few seconds before entering the conflicting location and t_i^+ as the time of exiting the conflicting location.

Remark 2: Note the disparity in costs among candidate choices can serve as a measure of confidence. The more pronounced the difference in cost between the two options, the more robust the consensus will be. Moreover, a slack variable ξ may be incorporated to model the internal defensiveness or aggressiveness of the agents. For instance, consider two priority plans with respective costs J_1 and J_2. Once can introduce a slack variable ξ such that the plan with cost J_1 is deemed favorable if and only if $J_1 < J_2 + \xi$, and vice versa. A negative ξ represents a defensive agent, while a positive ξ indicates an aggressive agent.

Remark 3: In the events where optimization (3) or (4) does not admit an unique solution, it becomes necessary

to implement additional logic for tie-breaking. In pactice, this may entail that all conflicting vehicles simultaneously coming to a stop, thereafter proceeding based on established social norms or sign language cues.

V. Numerical Simulation

In this section, we evaluate our least-action consensus model by conducting tests using *1*) an sequence of two-agent validation scenarios with ground truth provided by the B3D dataset and *2*) an integrative N-agent simulation scenario.

A. Two-Agent Validation

The two-agent scenarios are extracted from a specific unsignalized intersection as recorded in video "jnc07.mp4" shown in Figure 1d. The primary objective is to determine whether our least-action consensus model recovers the order of proceeding observed in the video. We manually selected six relatively isolated two-agent interactions from this video to apply the least-action consensus model.

Results from the two-agent validation tests are presented in Table I. As indicated in the table, the model successfully recovers naturalistic conflict resolution under all initial conditions.

TABLE I: Results of the Two-Agent Validation

Entry Speeds (m/s)*	Distances to Conflict (m)*	True Priority Order	Modelled Priority Order
(0.01, 3.57)	(5.52, 21.25)	Westbound First	Westbound First
(0.01, 5.32)	(5.37, 20.04)	Westbound First	Westbound First
(0.02, 3.24)	(5.87, 21.14)	Westbound First	Westbound First
(0.16, 2.65)	(5.82, 21.71)	Westbound First	Westbound First
(0.04, 1.79)	(7.33, 21.97)	Westbound First	Westbound First
(0.41, 3.18)	(4.35, 21.19)	Southbound First	Southbound First

* Data on entry speeds and distance to conflict are arranged as (southbound, westbound)

B. N-Agent Simulation

To evaluate the integrative performance of the model, we constructed a N-agent cross-intersection simulation involving southbound and westbound traffic flows as shown in Figure 6. The goal of this evaluation is to check if our least-action model can be deployed in an integrative simulation without collision. Each arm of the cross-intersection is symmetrically configured to be 20 meters in length. The simulation is designed to run for a total duration of approximately 300 seconds, during which inbound traffic is randomly generated based on a Poisson distribution with an average arrival rate of 900 vehicles per hour. The initial position of each vehicle is set at the respective intersection entrances, and the initial velocity is sampled from a normal distribution with a mean of 3 m/s and a standard deviation of 1 m/s, truncated between 0 and 4 m/s.

The results of the N-agent simulation tests are depicted in Figure 7. The figure features two timelines: the top timeline visualizes the entry times of incoming vehicles, while the bottom timeline displays their crossing times. Red data points represent vehicles approaching from the southbound direction, and blue data points from the westbound. Although vehicles enter the intersection randomly, as per the simulation setup, they cross the conflict point in a distinct batch pattern. Each direction alternates in crossing the conflict point in batches of queues. This qualitative behavior aligns with observations from the B3D dataset.

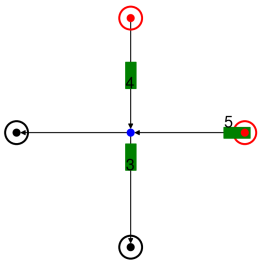


Fig. 6: The N-agent simulation scenario. Red circles indicate the origins of traffic while black circles denote the respective destinations. Green rectangles, labeled with numbers, represent the vehicles as they pass through the intersection.

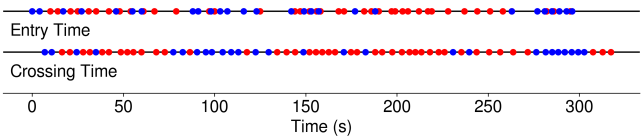


Fig. 7: Results of the N-agent simulation. Each red point represents a southbound vehicle, while each blue point represents a westbound vehicle.

VI. Discussion

The preliminary results from the numerical simulations suggest that the consensus-based approach is a viable framework for decentralized conflict resolution in understructured road environments. Despite these promising findings, several critical aspects of this research remain unexplored. We identify and discuss two particularly salient open questions: theoretical convergence analysis and further empirical validation. Detailed discussions of these issues are presented below.

First and foremost, an interesting theoretical challenge is to show the convergence of consensus in the presence of disturbances. Although we have empirically shown that this consensus approach to decentralized conflict resolution accurately describes how human drivers negotiate right-of-way in real-time, it has yet to be proven under which conditions the method will converge to a consistent priority order in a distributed setting.

Beyond theoretical analysis, it is crucial to evaluate the model's performance through more complex empirical testing. For instance, rather than limiting tests to two-agent conflict resolution, the model could be evaluated against N-agent interactions involving diverse conflict types and demand profiles. Additionally, conducting tests of the consensus model, in a Turing test-like manner, is also interesting. A successful model should operate in an understructured environment in such a way that it is indistinguishable from human drivers by a human observer.

REFERENCES

[1] United States Federal Highway Administration. Next generation simulation (ngsim) vehicle trajectories and supporting data, 2016.

[2] Luca Caltagirone, Mauro Bellone, Lennart Svensson, and Mattias Wahde. Lidar-based driving path generation using fully convolutional neural networks. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–6. IEEE, 2017.

[3] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] Hong-Cheol Choi and Inseok Hwang. Data-driven trajectory-based consensus approach to traffic management for manned and unmanned aviation. In *AIAA SCITECH 2024 Forum*, page 0537, 2024.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[6] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles Qi, Yin Zhou, Zoey Yang, Aurelien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[7] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[8] Andrew Gray, Yiqi Gao, J Karl Hedrick, and Francesco Borrelli. Robust predictive control for semi-autonomous vehicles with an uncertain driver model. In *2013 IEEE intelligent vehicles symposium (IV)*, pages 208–213. IEEE, 2013.

[9] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[10] Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In *International Conference on Intelligent Transportation Systems*, pages 2118–2125, 2018.

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[12] Brandon Luders, Mangal Kothari, and Jonathan How. Chance constrained rrt for probabilistic robustness to environmental uncertainty. In *AIAA guidance, navigation, and control conference*, page 8160, 2010.

[13] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

[14] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on intelligent vehicles*, 1(1):33–55, 2016.

[15] Venkatraman Renganathan, Iman Shames, and Tyler H Summers. Towards integrated perception and motion planning with distributionally robust risk constraints. *IFAC-PapersOnLine*, 53(2):15530–15536, 2020.

[16] Craig W Reynolds. Flocks, herds and schools: A distributed behavioral model. In *Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, pages 25–34, 1987.

[17] A Robicquet, A Sadeghian, A Alahi, and S Savarese. Learning social etiquette: Human trajectory prediction in crowded scenes. In *European Conference on Computer Vision*, volume 2, 2020.

[18] Dorsa Sadigh, Shankar Sastry, Sanjit A Seshia, and Anca D Dragan. Planning for autonomous cars that leverage effects on human actions. In *Robotics: Science and systems*, volume 2, pages 1–9. Ann Arbor, MI, USA, 2016.

[19] Sleiman Safaoui and Tyler H Summers. Distributionally robust cvar-based safety filtering for motion planning in uncertain environments. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 103–109. IEEE, 2024.

[20] Boris Sekachev, Nikita Manovich, Maxim Zhiltsov, Andrey Zhavoronkov, Dmitry Kalinin, Ben Hoff, TOsmanov, Dmitry Kruchinin, Artyom Zankevich, DmitriySidnev, Maksim Markelov, Johannes222, Mathis Chenuet, a andre, telenachos, Aleksandr Melnikov, Jijoong Kim, Liron Ilouz, Nikita Glazov, Priya4607, Rush Tehrani, Seungwon Jeong, Vladimir Skubriev, Sebastian Yonekura, vugia truong, zliang7, lizhming, and Tritin Truong. opencv/cvat: v1.1.0, 8 2020.

[21] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Sheng Zhao, Shuyang Cheng, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[22] Claire Tomlin, George J Pappas, and Shankar Sastry. Conflict resolution for air traffic management: A study in multiagent hybrid systems. *IEEE Transactions on automatic control*, 43(4):509–521, 1998.

[23] Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805, 2000.

[24] Bambang Riyanto Trilaksono. Distributed consensus control of robot swarm with obstacle and collision avoidance. In *2015 2nd International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE)*, pages 2–2. IEEE, 2015.

[25] Michael P Vitus and Claire J Tomlin. A probabilistic approach to planning and control in autonomous urban driving. In *52nd IEEE Conference on Decision and Control*, pages 2459–2464. IEEE, 2013.

[26] Fangyu Wu, Raphael E Stern, Shumo Cui, Maria Laura Delle Monache, Rahul Bhadani, Matt Bunting, Miles Churchill, Nathaniel Hamilton, Benedetto Piccoli, Benjamin Seibold, Jonathan Sprinkle, and Daniel Work. Tracking vehicle trajectories and fuel rates in phantom traffic jams: Methodology and data. *Transportation Research Part C: Emerging Technologies*, 99:82–109, 2019.

[27] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2, 2019.

[28] Huazhe Xu, Yang Gao, Fisher Yu, and Trevor Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2174–2182, 2017.

[29] Wei Zhan, Liting Sun, Di Wang, Haojie Shi, Aubrey Clausse, Maximilian Naumann, Julius Kummerle, Hendrik Konigshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps. *arXiv:1910.03088 [cs, eess]*, September 2019.