# Mode Collapse Happens: Evaluating Critical Interactions in Joint Trajectory Prediction Models

MSc Thesis

by

M.D. Hugenholtz

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Thursday July 25, 2024 at 9:30.


| | | |
|---|---|---|
| Student number: | 4649516 | |
| Project duration: | January 8, 2024 – July 25, 2024 | |
| Thesis committee: | Dr. Jens Kober, | TU Delft, supervisor |
| | Dr. Victor L. Knoop, | TU Delft |
| | Dr. Chris Pek, | TU Delft |
| | M.Sc. Anna Mészáros, | TU Delft, daily supervisor |
| | Dr. Zlatan Ajanović, | RWTH Aachen, daily supervisor |


An electronic version of this thesis is available at http://repository.tudelft.nl/.

TU Delft

# Acknowledgment

Firstly, I would like to thank Dr. Jens Kober for being my academic supervisor during this thesis, and providing valuable feedback to my research.

Secondly, I am extremely grateful to M.Sc. Anna Mészáros and Dr. Zlatan Ajanović, who were my daily supervisors during this project. Our discussions during the weekly meetings were always a source of inspiration and motivation, thanks to their expertise and feedback.

Third, I would like to thank Dr. Victor L. Knoop and Dr. Chris Pek, for taking part in the thesis committee that assesses the quality of my research.

Finally, I am also thankful to Ian van Vorst for proofreading my thesis and helping me push the quality of my work forwards.

*M.D. Hugenholtz*
*Delft, July 2024*

# Mode Collapse Happens: Evaluating Critical Interactions in Joint Trajectory Prediction Models

Maarten Hugenholtz[1]

***Abstract*—Autonomous vehicles rely on prediction modules, in order to plan collision-free trajectories. Vehicle trajectory prediction models are multimodal, to account for the multiple route options and the inherent uncertainty in human behavior. The state-of-the-art prediction models are deep-learning models, which are susceptible to mode collapse, a phenomenon in which the model fails to output the full distribution of modes and only predicts the most likely one. Mode collapsing poses safety concerns for autonomous driving, as missed predictions could result in collisions. Most works have focused on addressing this issue by generating diverse predictions that cover various route options at the environmental level. However, there are no metrics for mode-collapse. Furthermore, little attention has been given to generating diversity in the interaction modes among agent trajectories. Additionally, the traditional distance-based metrics are heavily dependent on datasets and do not evaluate interactions between agents. To this end, we propose a novel evaluation framework that assesses the interaction modes of joint trajectory predictions, focusing only on the safety-critical interactions in a dataset. We introduce a metric for mode-collapse and time-based metrics for mode correctness and coverage, shedding light on the temporal dimension of the predictions. We test four multi-agent trajectory prediction models on the widely used nuScenes dataset and conclude that mode collapse happens. While the rate of correctly predicted interaction modes increases closer to the interaction event, there are still cases where the models are unable to predict the interaction mode even right before the interaction happens. With the introduction of our novel framework, researchers can now benchmark their models' performance in predicting critical interactions. This provides new insights and perspectives, helping the holistic evaluation and interpretation of a model's performance. Additionally, our work offers a new developmental direction for prediction models, aiming for greater consistency and accuracy in predicting agent interactions, thereby advancing the safety of autonomous driving systems. Our evaluation framework is available online at:** `https://github.com/MaartenHugenholtz/InteractionEval`

## I. INTRODUCTION

Autonomous vehicles (AVs) have the potential to revolutionize personal transportation, motivated by improved driving comfort, energy efficiency and road safety [1]. Part of the autonomous driving challenge involves the planning of safe, comfortable and efficient trajectories. To achieve this, modular planning systems rely on a prediction module that predicts the motion of surrounding vehicles [2]. Because human behavior is naturally uncertain and multimodal, it is unrealistic to predict a single trajectory for each agent, based on the limited clues that can be extracted from the scene, without knowing the agent's intent. Therefore, multimodal trajectory prediction (MTP) was introduced by [3], in which multiple trajectories are predicted for each agent, to cover all possible modalities.

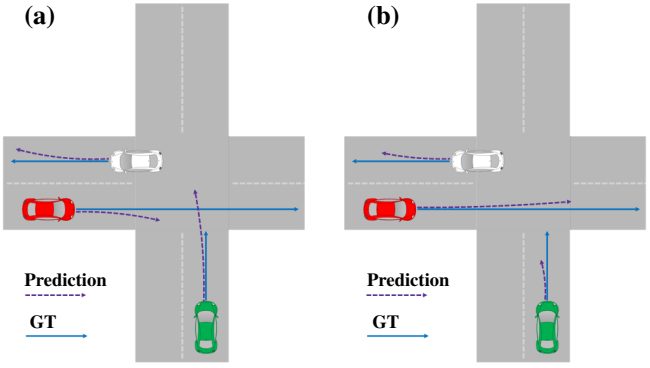



Fig. 1: We consider an exemplary intersection scenario, with two interacting vehicles, and one non-interacting vehicle. Both model (a) and (b) have similar mean final displacement errors, while only model (b) correctly predicts the interaction between the green and red vehicle.

A common problem is the models' susceptibility to mode collapse. This machine learning phenomenon occurs when the model fails to learn the true distribution of modes and only outputs the most likely mode, or two distant modes collapse into a single, infeasible mode [4]. In a safety-critical application like autonomous driving, it is crucial that such failures are avoided, as incomplete or inaccurate predictions, that are used in a downstream planner, could result in collisions. Several works have addressed the mode collapse issue by using goal-conditioned prediction and a diverse set of goals [5]–[7] or by using training objectives that allow for diverse predictions [8], [9] or promote distributions with high entropy [10]. These works mitigate mode collapse on the environmental level by generating diverse predictions that cover various route options, but little attention has been given to guaranteeing diversity in the interaction modes among agent trajectories. Furthermore, there are currently no metrics to evaluate mode collapse.

Vehicle trajectory prediction (VTP) models are evaluated in open-loop, and their performance is primarily evaluated with distance-based metrics that assess the models' accuracy. While these metrics are an obvious choice and easy to compute, they are heavily constrained on datasets, making it impossible to compare models from different datasets, complicating interpretation. Furthermore, none of the existing evaluation frameworks explicitly evaluate the model's ability to correctly

[1]Maarten Hugenholtz is with Faculty of Mechanical Engineering, Delft University of Technology, 2628 CD Delft, The Netherlands `M.D.Hugenholtz@student.tudelft.nl`

predict the interaction between agents, which we argue is the most safety-critical aspect of driving. In Figure 1, we demonstrate that traditional distance-based metrics fail to evaluate interactions between agents effectively and that averaging distance errors complicates the interpretation of results.

The aim of this study is to evaluate mode collapse on the interaction level in VTP models in an insightful and more data-independent way. More specifically, we want to research when mode collapse occurs and get insight into the temporal dimension of the predictions. Towards this end, we introduce a novel evaluation framework to benchmark a model's interaction prediction performance. Our contributions are fourfold: First, we evaluate the interaction modes of joint trajectory predictions by introducing an explicit metric for mode collapse and utilizing metrics for mode correctness and coverage. This is a safety-critical aspect that has previously been neglected in VTP evaluation. Second, we introduce time-based variants of these metrics, shedding light on the temporal evolution and consistency of the predictions. Third, we only consider the relevant parts of path-crossing interactions, thereby making the evaluation less dependent on datasets, and improving interpretability of the metrics. Finally, we benchmark two state-of-the-art trajectory prediction models, along with two other baseline models, on the nuScenes dataset and evaluate them using our novel metrics. Our results show that the models suffer from mode collapse and, in some cases, fail to correctly predict the interaction mode between agents, even just before the interaction happens.

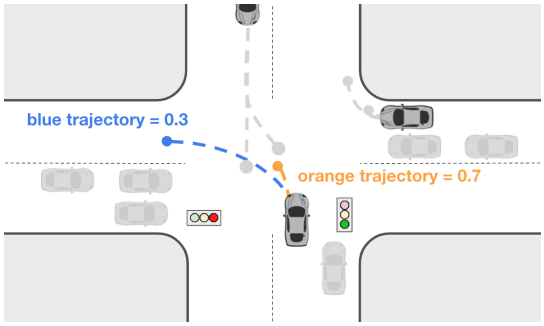
The rest of this paper is organized as follows: In Section II we give a brief literature overview on multimodal trajectory prediction models and the performance metrics used in popular benchmarks. In Section III we present our methodology and formulate our novel metrics. Section IV describes the models that we tested and in Section V their performance on the nuScenes dataset is discussed, with both qualitative and quantitative results. Finally, Section VI concludes this work, and we discuss limitations as well as exciting directions for future research in this area.

## II. RELATED WORKS

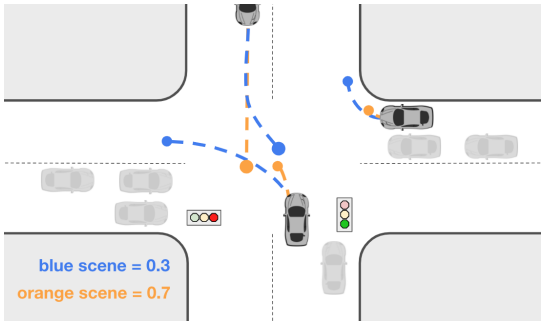
In Section II-A we discuss how multimodal trajectory prediction models mitigate mode collapse, what mode representations have been used, and the difference between marginal and joint prediction. Section II-B discusses the current trajectory prediction evaluation frameworks, and how they fail to effectively evaluate interactions.

### A. Multimodal trajectory prediction models

Multimodal trajectory prediction models employ various techniques to mitigate mode collapse. A common remedy is to first predict diverse modes and condition the prediction upon these modes, to guarantee diverse predictions. A mode is an abstraction of a trajectory referring to a high-level behavior, and can be represented on the environment level (goal lanes or points) [5], [11], vehicle level (lane change, accelerating, braking) [12], [13] or interaction level (yielding, going) [14]–[16]. Using such a mode as an intermediate representation to condition the prediction upon, improves interpretability and helps mitigate mode collapse. However, since no unified definition of a mode exists, there are also no metrics to quantify the discrete mode prediction performance of the models.




(a) Marginal prediction




(b) Joint prediction

Fig. 2: Illustration from [19], demonstrating the difference between marginal prediction (a) and joint prediction (b).

Multimodal trajectory prediction models can be categorized into node-centric and scene-centric models, which perform the prediction per-agent and jointly for the whole scene, respectively. Figure 2 demonstrates the difference. Generally, scene-centric models better capture the interactions among agents, have higher scene consistency and are more compatible with downstream planners [17]. On the other hand, node-centric models are easier to train and better cover the agents' motion [18]. In order to evaluate the interaction mode of a trajectory pair, joint trajectory predictions are required. Therefore, we will solely focus on the evaluation of scene-centric vehicle trajectory prediction models.

Categorical Traffic Transformer (CTT) [7] is an example of such a model. It uses an interpretable set of Scene Modes (SM) to supervise the latent mode distribution. Uniquely, these modes consist of two types: agent2lane (a2l) modes and agent2agent (a2a) modes, thereby capturing both the route and interaction intention of agents. However, because the number of modes scales exponentially with the number of agents in the scene (both in lane options and agent interaction), it is infeasible to predict all modes in scenes with many agents.

### B. Performance metrics

Vehicle trajectory prediction models are evaluated in open loop, using various metrics that assess the accuracy, probability, diversity, and admissibility of the predicted trajectories. Distance-based metrics like the minimum average displacement error (minADE), minimum final displacement error (minFDE) and miss rate (MR) have been the primary accuracy metrics used to compare multimodal trajectory prediction models. However, the performance of these metrics is heavily dependent on the used dataset, making comparison between different datasets impossible and complicating interpretation.

Another aspect that has been neglected in the evaluation is the interactions among agents. Recent works [7], [8], [20] have turned to scene-centric models, to better capture interactions between agents by simultaneously rolling-out their future trajectories. The Waymo Open Motion Dataset (WOMD) [21] prediction benchmark introduced joint metrics for the minADE, minFDE and MR. Their definitions are similar to their marginal variants, except that the minimum error of $K$ predictions is taken over the whole scene instead of agent-wise. This means that we cannot mix-and-match the best prediction for each agent over different scene samples, which means the prediction task is inherently more challenging but also gives a more realistic idea of the performance. While these joint metrics implicitly evaluate agent interactions, the lack of an explicit metric makes interpretation challenging, as demonstrated in Figure 1.

In CTT [7] a2l and a2a modes are defined and used to condition the prediction task upon the scene mode. Additionally, they introduce corresponding mode metrics: the mode correct rate and mode cover rate. The mode correct rate is the percentage of most likely (ML) predictions that match the ground truth (GT) mode (a2a, a2l or both). The mode coverage rate is the rate at which one of the $K$ predicted trajectories matches the GT mode. They compare their performance on these metrics to AgentFormer (AF) [20] on the nuScenes and WOMD datasets. While this is a promising step towards formalizing modes and improving intention prediction (lane and interaction modes), their metrics lack interpretability and are still heavily dependent on the dataset. The latter is demonstrated by the fact that for AF there is almost a 50% performance difference in the a2a cover rate between nuScenes and the WOMD. In this work, we will extend their mode metrics for a2a interactions and use them to quantify a model's interaction prediction performance in a more insightful and data-independent manner.

## III. METHODOLOGY

We argue that current evaluation frameworks lack interpretability because they are constrained to datasets, which vary in size, density, number of agents, etc. Therefore, these frameworks are not able to capture the model's critical interaction prediction performance, because *all* interactions are considered for *all* time steps. We propose to only evaluate the safety-critical interactions, and give a formal definition in Section III-A. To characterize the interactions, we use a two-class free-end homotopy concept (Section III-B). Furthermore, we only evaluate the predictions until the point where the interaction class becomes inevitable. To find this point, we simulate feasible future roll-outs for the interacting agents (Section III-C). Finally, in Section III-D we present our novel metrics for evaluating mode collapse on the interaction level. Additionally, we introduce time-based metrics to get insight into the temporal evolution of the predictions.

### A. Safety-critical interactions

A unified definition for inter-vehicle interactions was defined by [22]:

> *"A situation where the behavior of at least two road users can be interpreted as being influenced by the possibility that they are both intending to occupy the same region of space at the same time in the near future."*

This possibility is very low for a lot of the theoretical number of interactions, as the traffic flows are constrained by infrastructure and traffic rules. These interaction pairs are not interesting because the vehicles are on different lanes or are already in the same lane, e.g., in car-following scenarios. The interesting and safety-critical interactions are those where agents initially occupy different lanes but intend to occupy the same region of space or lane in the near future. Exemplary scenarios are merging and crossings at unsignalized intersections. To identify these interaction pairs, we will first formally define path-sharing and then outline our criteria for safety-critical interactions.

*Path-sharing definition.* We define the trajectories of two agents as $\tau_1$ and $\tau_2$:

$$\tau_1 = [(x_1, y_1), \dots, (x_n, y_n)]_1$$

$$\tau_2 = [(x_1, y_1), \dots, (x_n, y_n)]_2$$

where $(x_i, y_i)$ is an agent's position at time step $t_i$, where $i = 1, 2, \dots, n$ and is defined for the maximum interval at which both agents are present in the scene, i.e., recorded in the data.

To determine whether agents are on the same path, we compute the pairwise distance from each point of $\tau_1$ to all other points of $\tau_2$. Thus, the position difference matrix $\Delta P$ is calculated as:

$$\Delta P = \begin{bmatrix} (x_1, y_1) \\ \vdots \\ (x_n, y_n) \end{bmatrix}_1 - \begin{bmatrix} (x_1, y_1) & \dots & (x_n, y_n) \end{bmatrix}_2$$

The resulting matrix is of shape $(n \times n \times 2)$ and the entries $\Delta p_{ij}$ denote the position difference $(\Delta x, \Delta y)_{ij}$ between the agents. The entries $d_{ij}$ of the distance matrix $D$ are calculated by taking the Euclidean norm of the position differences $\Delta p_{ij}$:

$$d_{ij} = ||\Delta p_{ij}||_2$$

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots \\ \vdots & \ddots & \\ d_{n1} & & d_{nn} \end{bmatrix}$$

To determine if a point along the path was or will be occupied by the other agent, we take the minimum of $D$ over the columns and rows, respectively. A path point is on the commonly shared path if the distance is smaller than a threshold, $d < d_{\text{onpath}}$, which is set to 1.5 m. This is a reasonable threshold, considering that two narrow cars would be in collision if the distance between their path centerlines is less than 1.5 m. Thus, we calculate the boolean path-sharing vectors of the agent-pair as:

$$\text{onpath}_1 = \min_{\text{axis=1}} D < d_{\text{onpath}}$$
$$\text{onpath}_2 = \min_{\text{axis=0}} D < d_{\text{onpath}}$$

In Figure 3 we show exemplary scenarios of interacting and non-interacting agent-pairs, where the True values of the onpath vectors are visualized with bigger markers. To make sure the real minimum distance is calculated, the position vectors are interpolated to increase the resolution for the distance calculation.

*Interaction criteria.* We define an interaction pair to be safety-critical if the trajectories are not path-sharing at first, but are path-sharing at a later stage. We define the time at which an agent starts to be on the commonly shared path as:

$$t_{\text{path-sharing}} = \min\{t \mid \text{onpath} = 1\}$$

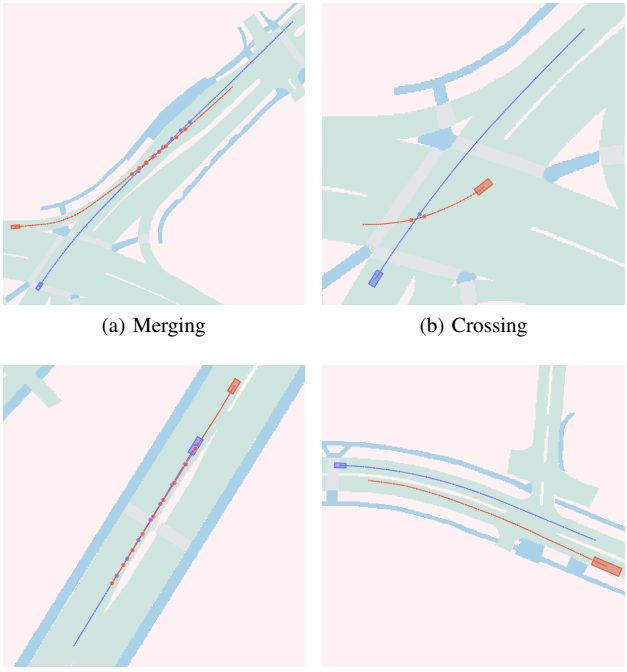
Thus, we are looking for the interactions where:

$$t_{\text{path-sharing},1} > t_1$$
$$t_{\text{path-sharing},2} > t_1$$

If the sequence is long, two trajectories can be path-sharing at the end, even if the cars are very far apart. Therefore, we impose an additional time-based constraint on the interaction: the time difference between the instances at which the vehicles begin to occupy the shared region should be no more than a prediction horizon, which is 6 seconds in the case of the nuScenes benchmark:

$$\Delta t_{\text{path-sharing}} = |t_{\text{path-sharing},2} - t_{\text{path-sharing},1}| \leq 6\,s$$

An interaction is defined to be safety-critical, if the trajectory pair satisfies all three conditions. With this definition, we can separate the safety-critical interesting interactions, like merging and crossing, from basic car-following and traffic light scenarios. This reduces the dependency on the dataset, as we only evaluate similar, and safety-critical, interactions. In the traditional trajectory prediction evaluation, all cars and scenes are considered, which complicates interpretation, because the distance errors are averaged, making it unclear what kind of scenarios were evaluated and how the model performed in critical cases. Thus, by applying our methodology, the metrics become more interpretable and insightful. Statistics on the interactions in the nuScenes dataset will be discussed in the results Section V-A.



(a) Merging (b) Crossing

(c) Car-following (d) Passing vehicles

Fig. 3: Exemplary traffic scenarios of safety critical interaction agent-pairs (a, b) and non-interacting agent-pairs (c, d) from the nuScenes dataset. The time steps where the agents are on the commonly shared path are visualized with big markers (True values of the onpath vectors).

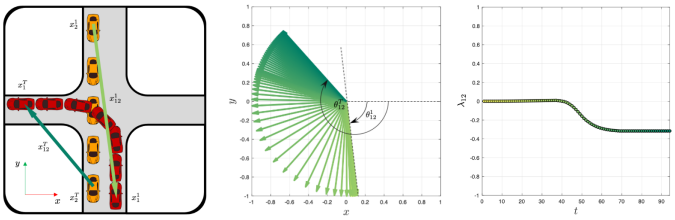
### B. Homotopy classes and convergence

To categorize the interaction between agent-pairs, we use homotopy classes. A group of trajectories with common start- and endpoints belong to the same homotopy class if they can be continuously deformed into each other without intersecting any obstacle [23]. We will follow [24] and build upon their concept of free-end homotopy, which has more flexible classes because the end-point of the trajectories does not have to be shared. The agents' interaction class is determined by the winding angle, which is the integrated angular difference between the agent-pair. Let $\tau$ be the trajectory of the ego and $\tau^0$ be the trajectory of an obstacle, with the sequence of waypoints discretized as $\{(x_i, y_i)\}_{i=1}^N$ and $\{(x_i^{\text{o}}, y_i^{\text{o}})\}_{i=1}^N$. The angular distance $\Delta\theta$ is computed as:

$$\Delta\theta\left(\tau, \tau^{\mathbf{o}}\right) := \sum_{i=1}^{N-1} \arctan\frac{y_{i+1} - y_{i+1}^{\text{o}}}{x_{i+1} - x_{i+1}^{\text{o}}} - \arctan\frac{y_i - y_i^{\text{o}}}{x_i - x_i^{\text{o}}}$$

The angle describes the relative rotation of the agents with respect to each other, as illustrated by Figure 4. The angle is used to categorize the agent-to-agent (a2a) interaction into three modes: $[S, CW, CCW]$ (static, clockwise, counterclockwise):

$$\text{h} := \begin{cases} CW, & \Delta\theta\left(\tau, \tau^{\mathbf{o}}\right) < -\hat{\theta} \\ S, & -\hat{\theta} \leq \Delta\theta\left(\tau, \tau^{\mathbf{o}}\right) < \hat{\theta} \\ CCW, & \Delta\theta\left(\tau, \tau^{\mathbf{o}}\right) > \hat{\theta} \end{cases}$$



(a) Top view of agents' trajectories. (b) Unit winding vector trajectory. (c) Winding number convergence.

Fig. 4: Visualization of the winding number (angular distance) calculation and convergence of two agents traversing an intersection, from [25]. The darkness of the colors increases with time, showing the temporal dimension of the calculation.

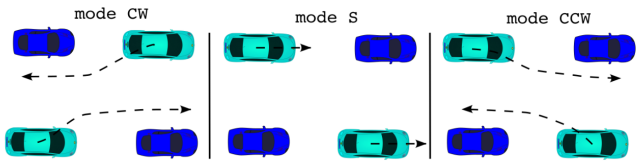



Fig. 5: Illustration from [24] of the three homotopy classes: CW, S, CCW.

Where $\hat{\theta}$ is a fixed threshold, large enough to differentiate between the classes. In Figure 5 the three classes are visualized.

In contrast to [7], [24], we set the threshold $\hat{\theta}$ to zero, effectively eliminating the static class. A fixed non-zero threshold can lead to ambiguities, as the angular distance $\Delta\theta$ not only depends on the speed and intention of the agents, but also on the road topology and the used prediction horizon. By eliminating the static class, we always have a distinct interaction class for a trajectory pair. This is especially important for the predictions, as they might not be close to the ground truth, but still contain the model's implicit homotopy class prediction, i.e., the intuition for how the agents will interact (CW or CCW rotation with respect to each other).

Furthermore, as described in Section III-A, we will only evaluate the safety-critical path-crossing interactions. Therefore, most static interactions like car-following and distant agent-pairs will be already filtered out, making the static class redundant anyway.

Besides filtering which interactions to evaluate, we also want to filter the temporal aspect of the interactions, i.e., once an interaction has happened, there is no point in further evaluating it. Figure 4 depicts the calculation process of the winding number (angular distance) over time of two agents traversing an intersection. From this figure, it becomes clear that the angular distance is only significant if the agents are close. Furthermore, once either of the agents has entered the common ground (the middle of the intersection in this case), the homotopy class of the interaction becomes inevitable and the winding number converges afterward. Geometrically, this happens once $\mathrm{onpath} = 1$ for either of the agents. It would however be too conservative to define this as the inevitable homotopy state, as vehicles cannot instantly accelerate and decelerate. Thus, to find the true instance at which the homotopy class becomes inevitable, dynamic simulations are needed, which will be discussed in the next subsection.

### C. Simulating future roll-outs

In this subsection, we describe our methodology for simulating feasible future trajectories for agent-pairs. Our goal is to find the set of feasible homotopy classes and the true inevitable homotopy state. These are important concepts for the novel metrics we propose in Section III-D. For each time step in the scene, and all agent-pairs that will cross paths in the near future, we want to find the set of feasible homotopy classes. To find this set, we simply accelerate one agent and decelerate the other, and vice versa. Thus, the set of future roll-outs for agent-pair $(i, j)$ at time step $t$ is: $y_{\mathrm{roll-outs,t}} = [(\tau_{\mathrm{i,decel}}, \tau_{\mathrm{j,accel}}), (\tau_{\mathrm{i,accel}}, \tau_{\mathrm{j,decel}})]$. We keep the ground truth paths of the agents, and only the velocity profile of the agents is altered (either accelerated or decelerated), whilst keeping both longitudinal and lateral accelerations within realistic limits for comfortable driving. The absolute longitudinal acceleration limit is set to $|a_{\mathrm{lon}}| = 1.47\,\mathrm{m/s^2}$ and the lateral to $|a_{\mathrm{lat}}| = 1.18\,\mathrm{m/s^2}$, which is based on the numbers in [26]. For the accelerations, we also set the maximum velocity equal to the maximum velocity of the scene, thereby implicitly respecting any speed limits or traffic that influences the maximum velocity in the scene.

Finally, we check whether a roll-out pair is feasible by using a binary collision detection function, denoted by $\mathrm{IsCollision}(\tau_i, \tau_j)$. To take into account the vehicle dimensions and headings, we take inspiration from [27], and fit three disks with radii $r_i = \frac{1}{2}\mathrm{width}$ of to each vehicle: at the vehicle center and at both bumpers. A collision is detected by computing the minimum distances between all disks of both vehicles, for all time steps. The vehicles are in collision if the minimum distance is smaller than the sum of the disk radii fitted to the vehicles: $d_{\min} < r_i + r_j$. Whilst there can still be hypothetical cases where a collision is missed, this approach works in most practical cases and is computationally efficient.

Now, we can define the set of feasible roll-outs as:

$$y_{\mathrm{feasible,t}} = \{y \in y_{\mathrm{roll-outs,t}} \mid \neg\,\mathrm{IsCollision}(\tau_i, \tau_j)\}$$

And consequently, the unique set of feasible homotopy classes is:

$$h_{\mathrm{feasible,t}} = \{h(y) \mid y \in y_{\mathrm{feasible,t}}\}$$

Where $h \in \{CW, CCW\}$. Thus, we define the inevitable homotopy collapse state, as the point in time at which only one unique homotopy class is feasible (non-colliding):

$$t_{\mathrm{h,collapse}} = \min\{\ t \ \big|\ \ |h_{\mathrm{feasible,t}}| = 1\ \}$$

### D. Interaction prediction metrics

Vehicle trajectory prediction models are multimodal, meaning they predict a set of $K$ trajectories, with corresponding probabilities. Since we want to evaluate the interaction between trajectories of agent-pairs, we require joint multi-agent predictions. Let us denote the predicted modalities of

agent-pair $(i, j)$ as $y_{\text{pred,i,j}} = [(\tau_\text{i}, \tau_\text{j})_1, \dots, (\tau_\text{i}, \tau_\text{j})_K]$, with the predictions ordered in decreasing likelihood, so $(\tau_\text{i}, \tau_\text{j})_1$ corresponds to the most likely (ML) prediction. The set of homotopy classes of the model's predictions is:

$$h_{\text{pred}} = \{h(y_\text{k}) \mid y_\text{k} \in y_{\text{pred}}\}$$

Furthermore, we denote $h_{\text{ml}}$ the homotopy class of the ML prediction, and $h_{\text{gt}}$ the ground truth homotopy class. To evaluate the model's ability to correctly predict the interaction mode, we follow [7] in defining mode correctness and coverage. The a2a mode is correct if the ML prediction's mode corresponds to the ground truth mode, i.e., $h_{\text{ml}} = h_{\text{gt}}$. The a2a mode is covered if one of the $K$ predictions covers the ground truth mode, i.e., $h_{\text{gt}} \in h_{\text{pred}}$.

In contrast to the default setting in VTP evaluation, we will not evaluate these metrics for the whole scene. Instead, we only consider the safety-critical interactions, and only till the inevitable homotopy state. Thus, we evaluate till the last point at which both classes are still feasible, i.e.:

$$t_{\text{h,final}} = \max\{\ t \ \big|\ \ |h_{\text{feasible,t}}| = 2\ \}$$

The evaluation starts once the homotopy class starts to converge towards the inevitable homotopy state, but at most a whole prediction horizon $T_\text{p}$ before then:

$$t_{\text{h,start}} = \min\{\ \ t \ \big|\ \ h_{\text{gt}}(t) = h_{\text{gt}}(t_{\text{h,final}})\ \}$$
$$\text{for} \quad t \in [t_{\text{h,final}} - T_\text{p}, t_{\text{h,final}}]$$

Thus the evaluation interval is $[t_{\text{h,start}}, t_{\text{h,final}}]$. Note that the duration of this interval varies, and in many cases is shorter than the 6-second prediction horizon, because the interval for which both agents are recorded in the data is shorter or the ground truth homotopy class starts to converge later. To get insight into the temporal evolution of the interaction prediction, we propose a time-based metric: the time-to-correct-mode-prediction ($\Delta T_{\text{correct}}$), which is the time the model needs to recognize the intention of the cars before the interaction has happened, i.e., before the inevitable homotopy state is reached.

$$t_{\text{incorrect}} = \max\{t \mid h_{\text{gt,t}} \neq h_{\text{ml,t}})\}$$
$$\Delta T_{\text{correct}} = t_{\text{h,final}} - t_{\text{incorrect}}$$

Similarly, we compute the time-to-covered-mode-prediction ($\Delta T_{\text{covered}}$). The difference is that we consider all $K$ predictions of the model, instead of just the most likely one.

$$t_{\text{uncovered}} = \max\{t \mid h_{\text{gt,t}} \notin h_{\text{pred,t}}\}$$
$$\Delta T_{\text{covered}} = t_{\text{h,final}} - t_{\text{uncovered}}$$

If the predictions are correct or covered from the beginning of the prediction interval, we cannot calculate the respective times, because we cannot make any assumptions about the model's predictions before then. In these cases, we consider the predictions a discrete correct class rather than a time. Therefore, we report two metrics, aggregated over all interactions: the percentage of predictions that are correct or covered from the beginning of the interaction interval ($@T_{\text{pred}}$) and the mean times for the predictions that are not.

*Mode collapse.* We define a2a mode collapse as an interaction mode being feasible, but not predicted by any of the model's predictions, i.e., $h_{\text{feas}} \not\subseteq h_{\text{pred}}$. So, mode collapse does not necessarily consider the ground truth, but the feasibility of hypothetical future roll-outs. Finally, we define the mode collapse rate as the percentage of time steps in the relevant interval $t \in [t_{\text{h,start}}, t_{\text{h,final}}]$, where mode collapse occurs. It is worth noting that in many cases (i.e., scenes with many agents) it is impossible for the model to cover all feasible modes with a finite number of joint predictions, due to the cardinality of the mode space growing exponentially with the number of agents.

*Temporal consistency predictions.* In [17] the temporal consistency of the predictions was found to be an important factor for the planner's performance in closed-loop simulation. In order to plan a safe path, the model's predictions should stay somewhat consistent throughout the scene, i.e., small variations in motions in a consecutive time step should not constitute a new mode. Therefore, we propose to evaluate the consistency of the ML prediction's interaction mode. The prediction consistency is a hit-or-miss metric, evaluated for each pair within the aforementioned relevant time horizon. The predictions for an agent-pair are said to be consistent if the model's ML mode prediction changes at most one time. So, given the mode predictions of consecutive time steps are $[CW, CCW, CCW]$, the predictions are said to be consistent, as it is acceptable for the model to correct itself. On the other hand, consecutive mode predictions of $[CCW, CW, CCW]$ are considered to be inconsistent.

*Implementation example.* Let us look at an example from AgentFormer's (AF) [20] predictions on one of the validation scenes of the nuScenes dataset. In this scene, only the relevant interacting agent-pairs are considered. At each frame, we simulate future roll-outs and check their feasibility with the collision checker. Furthermore, we calculate the homotopy classes of the ground truth, the predictions and the roll-outs. In Figure 6, we visualize this process for a single frame. Table I shows an overview of the interaction modes of predictions and roll-outs for all relevant frames of this interaction-pair.

For this specific interaction, the inevitable homotopy state is at frame 16, as only the $CW$ mode is still feasible, and the $CCW$ mode would yield a collision. We wish to evaluate the mode predictions a whole prediction horizon $T_\text{p}$ before then. However, in many cases (such as this example), this is not possible, simply because the interval for which both agents are recorded in the dataset is not long enough. Thus, we will evaluate the mode predictions from the first point at which there are predictions for both agents, until the inevitable homotopy state. In this case: from frame 5 until frame 15. Since nuScenes is recorded at 2Hz, we find that it takes the model $\Delta T_{\text{correct}} = 1.5\text{s}$ to correctly predict the interaction class. For $\Delta T_{\text{covered}}$ we see that the predictions cover the ground truth class from the start of the interaction interval. Since there are no prior time steps available, we consider such cases a correct/covered class, rather than a time. Furthermore, from the table, it becomes clear that the predictions are

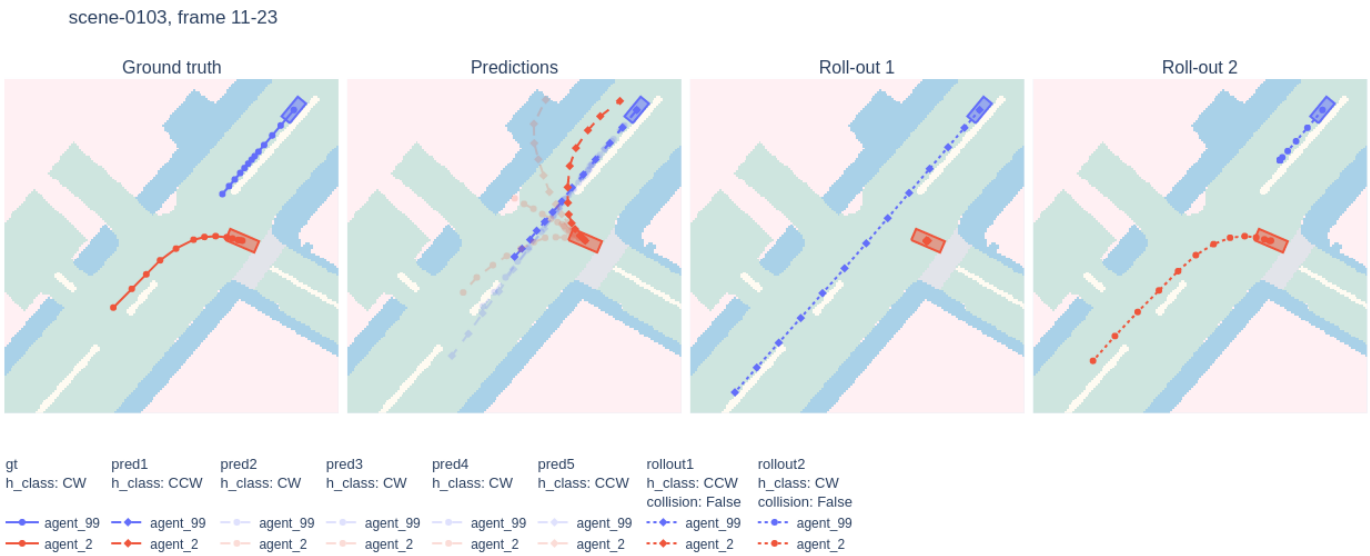



Fig. 6: Visualization of the interaction mode evaluation for AgentFormer on agent-pair (99,2) at frame 11 in scene-0103 of the nuScenes dataset. For the predictions, only the ML prediction is shown with full opacity. The corresponding homotopy classes (h_class) are shown in the legend, and displayed in the plot using $\circ$ and $\diamond$ markers for the CW and CCW class, respectively. Additionally, the collision boolean is shown in the legend for both roll-outs. For this specific frame, both interaction modes are still feasible. The mode is not predicted correctly (ML prediction), but it is covered by one of the other predictions.

TABLE I: Example mode metrics evolution for AF's predictions on agent-pair (99,2) in scene-0103 of the nuScenes dataset.

| frame | GT mode | ML mode | all K modes | feasible modes | mode correct | mode covered | mode collapse |
|---|---|---|---|---|---|---|---|
| 5 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 6 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 7 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 8 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 9 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 10 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 11 | CW | CCW | CCW CW | CCW, CW | | ✓ | |
| 12 | CW | CCW | CCW CW | CCW, CW | | ✓ | |
| 13 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 14 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 15 | CW | CW | CW | CCW, CW | ✓ | ✓ | ✓ |
| 16 | CW | CW | CW | CW | ✓ | ✓ | |

inconsistent because the ML prediction's mode changes more than once. Finally, in 9 out of the 11 frames not all feasible modes were predicted, so the mode collapse rate for this scene is $81.8\%$.

## IV. TRAJECTORY PREDICTION MODELS

We test our novel evaluation methodology on the nuScenes dataset [28] and report results for four models: AgentFormer (Section IV-A), Categorical Traffic Transformer (Section IV-B), an oracle model (Section IV-C) and a constant velocity model (Section IV-D). In the following subsections, we briefly discuss the characteristics and implementation of these models.

### A. AgentFormer

AgentFormer (AF) [20] is a multi-agent trajectory prediction model. They utilize a transformer-based architecture, that simultaneously models the social and temporal dimension of agents. Their prediction framework jointly models the agents' intentions, to predict diverse and socially-aware future trajectories. They test their model on the ETH/UCY and nuScenes datasets and publish their code including pre-trained models [29]. We will utilize their pre-trained model for nuScenes, and use the version which outputs $K = 5$ multi-agent trajectories.

### B. Categorical Traffic Transformer

Categorical Traffic Transformer (CTT) [7] is a multi-agent trajectory prediction model, with an interpretable latent space consisting of agent-to-agent and agent-to-lane modes. CTT generates diverse behaviors by conditioning the trajectory prediction on different modes. The authors published their code including pre-trained weights for the nuScenes dataset [30]. Unfortunately, we did not succeed in reproducing the numbers reported in their paper and uncovered various issues,

making direct comparison with the other models difficult. Firstly, their pre-trained model is trained for a prediction horizon of 3 seconds, whereas AF is trained for 6 seconds, as dictated by the nuScenes benchmark. To match the varying prediction horizons, the 6-second predictions from AF are cut to 3 seconds. Secondly, all predicted modes and trajectories are identical, making the model effectively unimodal. Finally, whereas AF predicts for all vehicles in the scenes, CTT predicts only for the road users within a certain attention radius of the ego-vehicle, but it does include pedestrians whereas AF does not. We use AF's data preprocessing backbone and match CTT's predictions to the corresponding agents. However, due to the aforementioned attention radius used in CTT, many predictions are missing for certain agents. In these cases, the current ground truth position is kept static and used as a prediction instead. Due to these issues, we are not able to report the real performance of CTT on interaction prediction. However, we still report the metrics and compare them to the other models, to set a baseline and show that our methodology generalizes to other models.

### C. Oracle model

The cardinality of the space of interaction modes grows exponentially with the number of agents in the scene. Because trajectory prediction models usually predict a fixed set of $K$ modes, covering all feasible modes becomes infeasible in scenes with many agents. To test this limitation, we propose a multimodal oracle model. The oracle's goal is to predict a set of $K$ multimodal trajectories that cover all feasible modes of the interacting agents. The oracle will be given access to the agents' ground truth paths, so it knows which agents will be interacting, i.e., crossing the same path, in the near future. However, the trajectories are unknown, i.e., it does not know the velocity profiles along the path, so the interaction class is still to be determined by the model. The oracle's goal is to cover all feasible interaction modes between the path-crossing agent-pairs. Analogously to the methodology described in Section III-C, we keep the agents' ground truth paths and simulate future roll-outs with a constant velocity, deceleration, or acceleration profile. Firstly, all agents are initialized with their constant velocity profile. Next, we calculate all combinations of constant velocity, acceleration, and deceleration profiles between the interacting agents and reject the combinations with collisions. Finally, we must assign each joint prediction a likelihood. We argue that the likelihood of a joint scene prediction is proportional to the overall utility in the scene, where the average speed of a roll-out combination can be used as a utility measure. Therefore, to get a finite set of $K$ joint predictions, we calculate the average speed of the roll-outs and output the top-$K$ trajectory combinations with the highest average speed.

### D. Constant velocity model

The constant velocity (CV) model is a simplistic unimodal model that assumes the vehicle will remain in its current heading and velocity [31]. Because it produces a single mode, it inherently suffers from mode collapse. However, it is an interesting baseline for comparison, because it tells us in how many scenarios we can correctly assess the vehicle pair's interaction class by simply extrapolating their current trajectories.

## V. RESULTS

Our aim is to evaluate the interaction mode prediction performance of VTP models in an insightful and data-independent way. More specifically, we want to research when mode collapse happens and get insight into the temporal dimension of the predictions. First, we employ our methodology for finding path-crossing safety-critical interactions on the widely used nuScenes traffic dataset, and report interaction statistics in Section V-A. Next, we test four baseline models (described in Section IV) and evaluate their performance using our novel evaluation framework in Section V-B. We show that mode collapse happens and shed light on the temporal evolution of the predictions. Finally, we compare qualitative results in Section V-C and compare our metrics to the traditional distance-based metrics in Section V-D.

### A. Interaction statistics nuScenes

We analyzed interactions across the entire train and validation splits of the nuScenes dataset, and applied our methodology to identify safety-critical interactions. In total, we identified 18,299 theoretical interactions across the entire dataset. The theoretical upper limit per scene is calculated as $N(N-1)/2$, considering the symmetry of interactions and the absence of self-pairs. However, in reality, only 16,756 theoretical interaction pairs exist, as not all agents are recorded for the full scene duration. After applying our first two interaction criteria, i.e., the agents are not path-sharing at first but are later on, only 730 interaction pairs are left. We characterize the closeness of these interactions in both distance and time in a density heatmap, see Figure 7. From the figure it becomes clear that the majority of interactions are close, i.e., the time difference is smaller than 5 seconds and the real-time closest distance is smaller than 20m. However, there is also a substantial part of path-sharing interactions, where there is a big time difference between the agents starting to occupy the same path or the distance between them is quite large. Since we are interested in safety-critical interactions, the time difference between the agents should be relatively small. Therefore, we apply our third interaction criterion, i.e., $\Delta t_{\mathrm{path-sharing}} \leq 6\,s$, after which only 351 interaction pairs are left in the full train and validation split. That means only $2.1\%$ of the possible interactions are considered safety-critical.

For testing the models on nuScenes, we evaluate them only on the validation split, which contains just 41 safety-critical interaction pairs. After identifying which interactions to evaluate, we now determine when to evaluate them. Employing our methodology for determining the inevitable homotopy state, we analyze the duration of the interaction interval $[t_{\mathrm{h,start}}, t_{\mathrm{h,final}}]$ before the homotopy class collapses. In Figure 8, we present a histogram showing the distribution

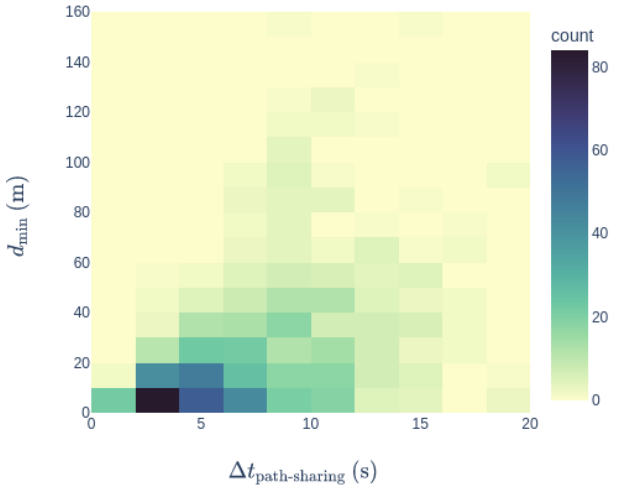



Fig. 7: Density heatmap of the path-sharing interactions in the full train-validation split of nuScenes. The interactions are characterized in closeness, with the real-time closest distance on the y-axis and the time difference between the agents occupying the shared-path on the x-axis.

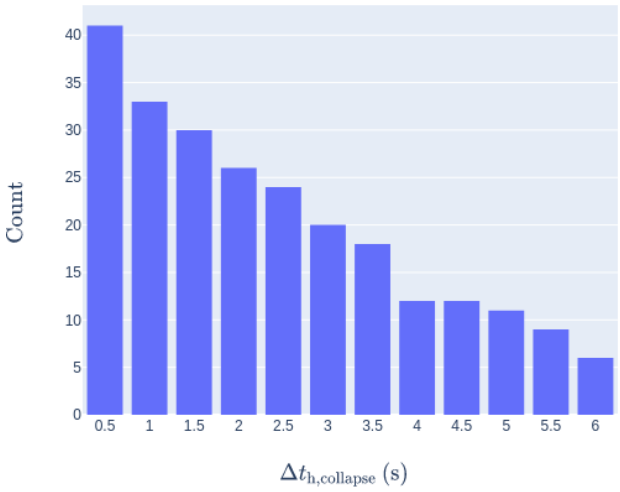



Fig. 8: Histogram of data samples before the inevitable homotopy state. The samples are prediction frames of safety-critical interaction pairs in the nuScenes validation split.

of samples over their time to the inevitable homotopy state, $\Delta t_{\text{h,collapse}}$. Naturally, this histogram shows a decreasing trend, as the interval during which both agents are recorded in the dataset is relatively short for many interactions. In total, we have just 41 usable interaction pairs, however, for the majority there are just a few samples available before the homotopy class becomes inevitable. There are just 6 pairs for which we can evaluate the predictions a full 6-second prediction horizon before $t_{\text{h,collapse}}$. Next, we will evaluate the models' mode prediction performance on these interaction pairs.

### *B. Model intention prediction performance*

Predicting the driver's intentions 6 seconds before the interaction happens is far less important than predicting them 1 second before it happens. On the other hand, correctly predicting the intentions 1 second before the interaction happens, is also a lot easier, as the drivers in the scene likely have already implicitly communicated who takes priority and crosses first, resulting in increased margins and speed differences. To shed light on the temporal evolution of a model's mode prediction performance, we analyze the mode correct, covered and collapse rates against the time to inevitable homotopy state $\Delta t_{\text{h,collapse}}$, see Figure 9. Indeed, we see that, as the interaction comes closer (smaller $\Delta t_{\text{h,collapse}}$), all models are naturally better able to correctly predict the interaction class. That the alternative roll-outs are less likely to happen, is also reflected by the higher mode collapse rate of AF for samples closer to the inevitable homotopy state. Failing to cover a feasible mode when it is unlikely is not problematic. However, in some cases, the models are not even able to correctly predict the interaction class right before the homotopy class becomes inevitable, indicating that mode collapse also occurs in critical situations.

In Table II the mode correctness, coverage and collapse rates of all models are summarized, as well as the time-based metrics and consistency. First, we will compare the intention prediction performance of AF, the oracle and the CV model on a prediction horizon of 6 seconds, and focus on the time-based metrics. AF correctly predicts the interaction mode at the beginning of the prediction horizon ($@T_{\text{pred}}$) in 56% of the cases, and if not, it takes up to 1.9 seconds on average to correctly predict the correct mode. Interestingly, the CV model outperforms the other models and in 78% of the cases it can correctly predict the interaction mode from the beginning, by simply extrapolating the vehicle's current trajectory. This shows that in the majority of the cases, the interaction class is a natural evolution of the vehicle's current heading and speed.

Naturally, in the covered category, the multimodal models perform better, as all predictions are considered. AF manages to directly cover ground truth mode in 80% of the cases, whereas the oracle model achieves a perfect score. The oracle model inherently tries to cover all feasible modes of the interacting agents, but since the models can only predict $K = 5$ futures, it cannot completely mitigate mode collapse in scenes with many agents. The oracle scores a mode collapse rate of 19%, versus the 70% of AF. The CV model inherently suffers from 100% mode collapse due to its unimodal predictions.

Finally, we compare the models, including CTT, on a 3-second prediction horizon, with the results reported in the bottom half of Table II. As explained earlier, CTT's predictions are unimodal and sometimes even missing, resulting in the model's disastrous performance. Only for 40% of the interaction-pairs the mode is predicted correctly right away, and in 53% of the cases, the mode is not predicted at all, resulting in a mean $\Delta T_{\text{correct}}$ of 0.1 seconds. Because of the model's unimodal predictions, it inherently suffers from

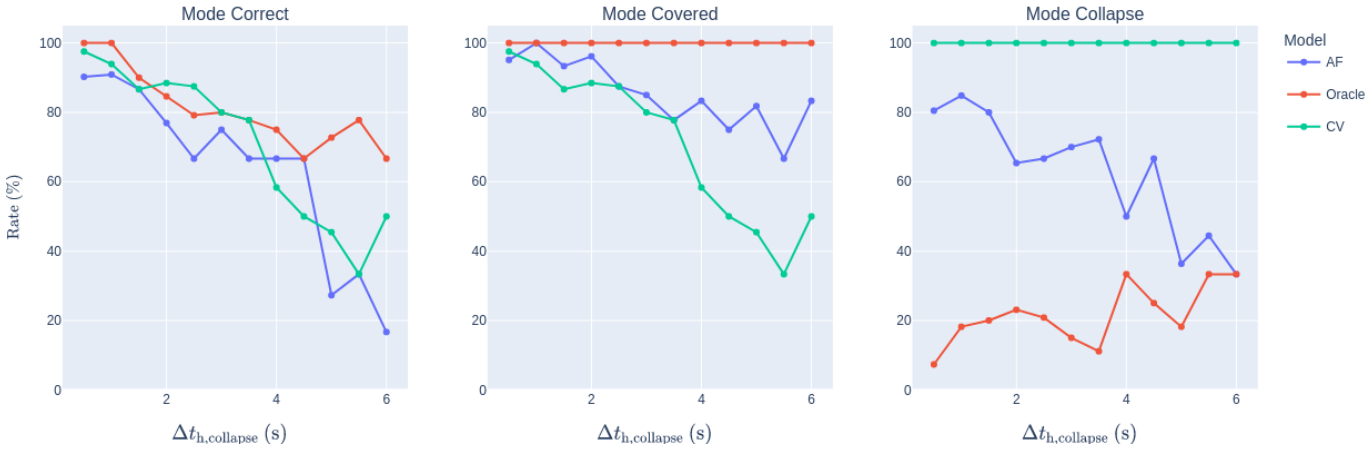



Fig. 9: Relative mode prediction performance plotted against the time to inevitable homotopy state. From left to right, we consider the correct, covered and collapsed modes. We evaluate AF, the oracle, and CV model on a prediction horizon of 6 seconds.

TABLE II: Interaction mode prediction metrics for AF, CTT, the oracle and the CV model. The rates are evaluated over all interaction-pair samples, whereas the time-based metrics and consistency are calculated per interaction-pair sequence and later averaged. We compare the mean time-to-correct/covered mode prediction, as well as the percentage of predictions at 0 seconds and the percentage of predictions that are correct from the beginning of the prediction interval ($@T_{\text{pred}}$). The best metrics in each category are printed **bold** and the second-best *italic*.

| Method | $T_{\text{pred}}$ (s) | Mode correct rate ↑ (%) | Mode covered rate ↑ (%) | Mode collapse rate ↓ (%) | $\Delta T_{\text{correct}}$ / $\Delta T_{\text{covered}}$ | | | Prediction Consistency ↑ (%) |
|---|---|---|---|---|---|---|---|---|
| | | | | | mean ↑ ($s$) | @0$s$ ↓ (%) | $@T_{\text{pred}}$ ↑ (%) | |
| AF | 6 | 74.0 | *89.3* | *69.8* | 1.9 / 1.8 | 9.8 / 4.9 | 56.1 / *80.5* | 92.7 |
| Oracle | | **86.0** | **100.0** | **18.6** | **2.4** / - | **0.0** / **0.0** | *73.2* / **100.0** | *97.6* |
| CV model | | *80.6* | 80.6 | 100.0 | *2.3* / *2.3* | *2.4* / *2.4* | **78.0** / 78.0 | **100.0** |
| AF | 3 | 83.4 | *92.9* | *76.9* | *1.0* / 0.8 | 12.2 / *4.9* | *70.7* / *87.8* | *95.1* |
| CTT* | | 49.3 | 49.3 | 100.0 | 0.1 / 0.1 | 53.3 / 53.3 | 40.0 / 40.0 | **100.0** |
| Oracle | | *86.4* | **100.0** | **13.0** | **1.6** / - | **2.4** / **0.0** | *70.7* / **100.0** | **100.0** |
| CV model | | **87.0** | 87.0 | 100.0 | 0.9 / *0.9* | *7.3* / 7.3 | **80.5** / 80.5 | **100.0** |

* Note that we were not able to reproduce the numbers reported in CTT's paper, and that some predictions are missing due to the issues discussed in Section IV-B.

mode collapse for all scenarios. As we could not reproduce CTT's results, this is not representative of its real performance. However, by testing our methodology on multiple models, we show that it generalizes to other models.

Comparing the other models on the 6-second prediction horizon, we see that the results are slightly different because a shorter prediction horizon changes the number of samples and some predictions may fall in a different homotopy class for the shorter horizon. However, the relative performance differences between the models remain unchanged. Although we cannot compare results from different prediction horizons directly, we demonstrated that our methodology is not limited to a single prediction horizon.

In terms of prediction consistency, all models score high: only in some cases the interaction mode changes inconsistently. The CV model and CTT score 100%, which is more trivial, as they only output a single mode, so inconsistent mode predictions are less likely.

### C. Qualitative results

We compare the qualitative results of AF and the CV model. These are the most interesting models to compare, since we use the ground truth paths for the Oracle and were not able to reproduce the results for CTT, resulting in unimodal and missing predictions. As the visualizations take up a lot of space, we analyze them in the appendix and only present the conclusions here.

The homotopy class is determined based on the sign of the angular distance between the vehicles, which is influenced by two factors: the relative speed difference between the agents and their paths. Since we only evaluate interaction-pairs that by definition have a commonly shared path, we see that the speed difference between the vehicles is the most important factor for the evolution of the homotopy class. This is especially true for frames closer to the interaction event, which is also reflected by the superior performance of the CV model.

We also encountered cases where there is a 100% mode

collapse for a feasible interaction mode, i.e., it is not covered by any of the predictions. In the analyzed cases, however, this mode collapse does not seem problematic as the class might be feasible, but also very unlikely, based on the speed differences of the cars. This uncovers a weakness of our mode collapse definition: we only evaluate the feasibility and not the likelihood of the roll-outs, which is important to realize when interpreting the results.

We also find that the interaction metrics do not assess the severity of the consequences of incorrect predictions. In some cases, the interaction mode is not predicted at all before we reach the inevitable homotopy state, while the vehicles are still quite far apart. This is because the time before the inevitable homotopy state does not reflect how close the agents are, but only how close one of the agents is to entering the shared future path.

Furthermore, we see that sometimes the predictions of AF are not admissible, e.g., not kinematically feasible, going off-road, colliding or going into the wrong traffic direction. This shows the importance of holistic VTP evaluation: besides measuring the accuracy of predictions, we should evaluate their diversity and admissibility.

### *D. Distance-based metrics results*

In Table III we compare the models on the traditional distance-based metrics. We report the average and final displacement errors (ADE/FDE) for the most likely (ML) predictions, as well as the joint lower-bound metrics calculated for $K = 5$ modes. In contrast to our novel interaction metrics, these are calculated over all scenes and time steps of the nuScenes validation split.

TABLE III: Distance-based metrics for AF, CTT, the oracle and the CV model, for 3-second and 6-second prediction horizons. The best metrics in each category are printed **bold** and the second-best *italic*.

| Method | $T_{\text{pred}}$ (s) | ML ADE (m) ↓ | ML FDE (m) ↓ | Joint minADE (m) ↓ | Joint minFDE (m) ↓ |
|---|---|---|---|---|---|
| AF | 6 | 3.88 | *9.10* | **2.86** | **6.48** |
| Oracle | | *3.84* | 9.12 | *3.56* | *8.41* |
| CV model | | **3.64** | **9.04** | 3.64 | 9.04 |
| AF | 3 | 1.48 | 3.00 | **1.11** | **2.17** |
| CTT* | | 5.93 | 10.59 | 5.93 | 10.59 |
| Oracle | | *1.45* | *2.85* | 1.36 | *2.63* |
| CV model | | **1.22** | **2.68** | *1.22* | 2.68 |

* Note that we were not able to reproduce the numbers reported in CTT's paper, and that some predictions are missing due to the issues discussed in Section IV-B.

Comparing the ML ADE metrics to the ML interaction metrics, we see that the relative performance order remains similar, with the oracle and the CV model performing the best. Interestingly, we see that on the joint metrics, AF performs best, which contradicts with our findings from the interaction metrics. This is partially caused by the fact that the oracle was designed specifically to cover modes of path-crossing vehicles, and not to get the lowest minimum distance errors. But it also shows that in some cases, the joint lower-bound distance-based metrics are not able to capture the model's ability to cover interaction modes amongst agent trajectories.

## VI. CONCLUSION AND DISCUSSION

We introduced a novel evaluation framework to benchmark a model's interaction prediction performance. Our framework simulates alternative interaction modes, and we use this to define a metric for mode collapse on the interaction level. We also use metrics for mode correctness and coverage, and propose time-based variants, that provide insight into the temporal evolution of mode predictions. Uniquely, our method does not evaluate all scenes and frames of a dataset, but only the relevant frames for closely interacting agent-pairs. This reduces the dataset dependency and makes our metrics more insightful and interpretable. We tested four models on the nuScenes dataset and showed that mode collapse happens. Interestingly, a simple constant velocity model outperformed the other models in correctly predicting the interaction mode, showing that in many cases the interaction mode is dictated by the vehicles' current heading and speed. While AgentFormer (AF) manages to produce diverse predictions for each agent, it did not cover all feasible interaction modes between the interacting agents, averaging a mode collapse rate of 70% for the safety-critical interaction pairs. The oracle model, designed to cover all feasible interaction modes, had a mode collapse rate of 20%. Thus, completely alleviating mode collapse (i.e., covering all feasible interaction modes) is not possible with a finite number of $K = 5$ joint predictions due to the exponentially growing cardinality of the mode space. Although the oracle was superior in covering the interaction modes, it was outperformed by AF on the joint distance-based metrics, indicating that these metrics do not necessarily capture the model's performance in predicting interaction modes. Finally, we analyzed the temporal evolution of the predictions, and found that both the mode correct and collapse rate increase as the inevitable homotopy state comes closer. In the majority of the scenarios, these collapsed interaction modes do not seem problematic, as they are not likely to happen. However, in a few cases, the models are not able to correctly predict the real interaction class right before it happens. These incorrectly predicted driver intentions could pose safety concerns for autonomous driving.

While we demonstrate that mode collapse occurs, our metrics do not evaluate the severity of consequences, nor the likelihood, of the collapsed modes. In our framework, we simulate feasible futures for interacting agents at every time step, but the model inputs remain the ground truth history of the agents as we replay the scene. Assessing the safety implications of collapsed modes requires a closed-loop simulation setup, in which the predictions are used in a downstream planner. Estimating the likelihood of a collapsed mode could involve comparing the scenario to a distribution learned from traffic data. However, rare but feasible scenarios might be underrepresented and deemed unlikely. Alternatively, planning-like costs could be used to evaluate the safety, com-

fort, and utility of future roll-outs as a measure of probability. Extending our framework to assess the associated risks of false predictions presents an exciting opportunity for future research in this area.

In our framework, we only perform roll-outs and collision checks for pairs of interacting agents. However, in reality, the scenes can be more complex, with multiple agents interacting and influencing each other. While this is a conceptual limitation of our method, the feasibility of our simulations remains valid, as the feasibility is primarily determined by the yielding vehicle's ability to brake before entering the common path, which is not affected by other vehicles.

By applying our methodology to identify safety-critical interactions, we aimed to make the evaluation less dependent on the dataset while focusing on the most crucial aspect of driving: the interactions. In the nuScenes dataset, we found that only 2% of the theoretical interactions are considered safety-critical according to our criteria. This finding highlights the need for more interactive datasets and the importance of metrics that are less constrained on the scenarios in a dataset. However, it also reveals a limitation of our approach: we evaluate only real interactions, not hypothetical ones. We opted for this simplistic approach to ensure that the interactions we assess are realistic. Furthermore, simulating all hypothetical interactions would be extremely complex and computationally demanding.

Finally, we analyzed the temporal evolution of the interactions between the critical agent-pairs. For the majority of the pairs, however, there were only a few samples available prior to the interaction, limiting the interpretability of our time-based metrics. This limitation arises because nuScenes is recorded from an on-road viewpoint, constraining the annotations to the range of the ego vehicle. To address this issue, future research could apply our methodology to traffic datasets recorded from a top-down perspective, such as those captured by drones monitoring traffic at intersections [32]–[34].

Our novel interaction metrics provide new ways to measure the intention prediction of models in safety-critical interactions. These metrics only take into account the relevant interactions, thereby reducing the dependency on datasets and improving interpretability. Furthermore, our time-based metrics shed light on the temporal evolution of predictions, an aspect that was previously neglected in VTP evaluation. Our new evaluation methodology thus offers new insights and perspectives, helping the holistic evaluation and interpretation of a model's performance. Finally, our evaluation methodology can aid the development of VTP models towards more accurate and consistent interaction predictions. Future work should focus on alleviating the aforementioned weaknesses and further generalizing our framework to other datasets and models to establish a benchmark for prediction models.

## REFERENCES


[1] K. Othman, "Exploring the implications of autonomous vehicles: a comprehensive review," *Innovative Infrastructure Solutions*, vol. 7, no. 2, p. 165, 2022. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8885781/

[2] S. Hagedorn, M. Hallgarten, M. Stoll, and A. Condurache, "Rethinking Integration of Prediction and Planning in Deep Learning-Based Automated Driving Systems: A Review," Aug. 2023, arXiv:2308.05731 [cs]. [Online]. Available: http://arxiv.org/abs/2308.05731

[3] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially Acceptable Trajectories with Generative Adversarial Networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, Jun. 2018, pp. 2255–2264. [Online]. Available: https://ieeexplore.ieee.org/document/8578338/

[4] J. Amirian, J.-B. Hayet, and J. Pettre, "Social Ways: Learning Multi-Modal Distributions of Pedestrian Trajectories With GANs," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. Long Beach, CA, USA: IEEE, Jun. 2019, pp. 2964–2972. [Online]. Available: https://ieeexplore.ieee.org/document/9025550/

[5] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, C. Li, and D. Anguelov, "TNT: Target-driven Trajectory Prediction," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 895–904, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v155/zhao21b.html

[6] J. Gu, C. Sun, and H. Zhao, "DenseTNT: End-to-end Trajectory Prediction from Dense Goal Sets," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 15 283–15 292. [Online]. Available: https://ieeexplore.ieee.org/document/9710037/

[7] Y. Chen, S. Tonkens, and M. Pavone, "Categorical Traffic Transformer: Interpretable and Diverse Behavior Prediction with Tokenized Latent," Nov. 2023, arXiv:2311.18307 [cs]. [Online]. Available: http://arxiv.org/abs/2311.18307

[8] Y. Chen, B. Ivanovic, and M. Pavone, "ScePT: Scene-consistent, Policy-based Trajectory Predictions for Planning," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 17 082–17 091. [Online]. Available: https://ieeexplore.ieee.org/document/9880415/

[9] Y. Yuan and K. Kitani, "DLow: Diversifying Latent Flows for Diverse Human Motion Prediction," in *Computer Vision – ECCV 2020*, A. Vedaldi, H. Bischof, T. Brox, and J.-M. Frahm, Eds. Cham: Springer International Publishing, 2020, pp. 346–364.

[10] N. Deo and M. M. Trivedi, "Trajectory Forecasts in Unknown Environments Conditioned on Grid-Based Plans," Apr. 2021, arXiv:2001.00735 [cs]. [Online]. Available: http://arxiv.org/abs/2001.00735

[11] J. Wang, T. Ye, Z. Gu, and J. Chen, "LTP: Lane-based Trajectory Prediction for Autonomous Driving," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. New Orleans, LA, USA: IEEE, Jun. 2022, pp. 17 113–17 121. [Online]. Available: https://ieeexplore.ieee.org/document/9878679/

[12] A. Benterki, M. Boukhnifer, V. Judalet, and C. Maaoui, "Artificial Intelligence for Vehicle Behavior Anticipation: Hybrid Approach Based on Maneuver Classification and Trajectory Prediction," *IEEE Access*, vol. 8, pp. 56 992–57 002, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9043491/

[13] H. Berkemeyer, R. Franceschini, T. Tran, L. Che, and G. Pipa, "Feasible and Adaptive Multimodal Trajectory Prediction with Semantic Maneuver Fusion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, May 2021, pp. 8530–8536, iSSN: 2577-087X. [Online]. Available: https://ieeexplore.ieee.org/document/9561380

[14] S. Kumar, Y. Gu, J. Hoang, G. C. Haynes, and M. Marchetti-Bowick, "Interaction-Based Trajectory Prediction Over a Hybrid Traffic Graph," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2021, pp. 5530–5535, iSSN: 2153-0866. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9636143

[15] D. Lee, Y. Gu, J. Hoang, and M. Marchetti-Bowick, "Joint Interaction and Trajectory Prediction for Autonomous Driving using Graph Neural Networks," Dec. 2019, arXiv:1912.07882 [cs, stat]. [Online]. Available: http://arxiv.org/abs/1912.07882

[16] Y. Hu, W. Zhan, L. Sun, and M. Tomizuka, "Multi-modal Probabilistic Prediction of Interactive Behavior via an Interpretable Model," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2019, pp. 557–563, iSSN: 2642-7214. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/8813796

[17] Y. Chen, P. Karkus, B. Ivanovic, X. Weng, and M. Pavone, "Tree-structured Policy Planning with Learned Behavior Models," in *2023 IEEE International Conference on Robotics and Automation*

*(ICRA)*, May 2023, pp. 7902–7908. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10161419

[18] W. Luo, C. Park, A. Cornman, B. Sapp, and D. Anguelov, "JFP: Joint Future Prediction with Interactive Multi-Agent Modeling for Autonomous Driving," in *Proceedings of The 6th Conference on Robot Learning*. PMLR, Mar. 2023, pp. 1457–1467, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v205/luo23a.html

[19] J. Ngiam, V. Vasudevan, B. Caine, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, D. J. Weiss, B. Sapp, Z. Chen, and J. Shlens, "Scene Transformer: A unified architecture for predicting future trajectories of multiple agents," Oct. 2021. [Online]. Available: https://openreview.net/forum?id=Wm3EA5OlHsG

[20] Y. Yuan, X. Weng, Y. Ou, and K. Kitani, "AgentFormer: Agent-Aware Transformers for Socio-Temporal Multi-Agent Forecasting," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9793–9803. [Online]. Available: https://ieeexplore.ieee.org/document/9710708/

[21] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large Scale Interactive Motion Forecasting for Autonomous Driving : The Waymo Open Motion Dataset," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 9690–9699, iSSN: 2380-7504. [Online]. Available: https://ieeexplore.ieee.org/document/9709630

[22] G. Markkula, R. Madigan, D. Nathanael, E. Portouli, Y. M. Lee, A. Dietrich, J. Billington, A. Schieben, and N. Merat, "Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, Nov. 2020, publisher: Taylor & Francis _eprint: https://doi.org/10.1080/1463922X.2020.1736686. [Online]. Available: https://doi.org/10.1080/1463922X.2020.1736686

[23] S. Bhattacharya, M. Likhachev, and V. Kumar, "Topological constraints in search-based robot path planning," *Autonomous Robots*, vol. 3, no. 33, pp. 273–290, 2012. [Online]. Available: https://www.infona.pl//resource/bwmeta1.element.springer-4001607d-3d84-31db-bac2-8bbfc2c08bf4

[24] Y. Chen, S. Veer, P. Karkus, and M. Pavone, "Interactive Joint Planning for Autonomous Vehicles," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 987–994, Feb. 2024, conference Name: IEEE Robotics and Automation Letters. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/10316575

[25] J. Roh, C. Mavrogiannis, R. Madan, D. Fox, and S. Srinivasa, "Multimodal Trajectory Prediction via Topological Invariance for Navigation at Uncontrolled Intersections," in *Proceedings of the 2020 Conference on Robot Learning*. PMLR, Oct. 2021, pp. 2216–2227, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v155/roh21a.html

[26] K. N. de Winkel, T. Irmak, R. Happee, and B. Shyrokau, "Standards for passenger comfort in automated vehicles: Acceleration and jerk," *Applied Ergonomics*, vol. 106, p. 103881, Jan. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0003687022002046

[27] J. Ziegler and C. Stiller, "Fast collision checking for intelligent vehicle motion planning," in *2010 IEEE Intelligent Vehicles Symposium*. La Jolla, CA, USA: IEEE, Jun. 2010, pp. 518–522. [Online]. Available: http://ieeexplore.ieee.org/document/5547976/

[28] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A Multimodal Dataset for Autonomous Driving," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 11 618–11 628, iSSN: 2575-7075. [Online]. Available: https://ieeexplore.ieee.org/document/9156412

[29] Y. Yuan, "Khrylx/AgentFormer," May 2024, original-date: 2021-03-24T16:40:46Z. [Online]. Available: https://github.com/Khrylx/AgentFormer

[30] "NVlabs/diffstack at CTT release." [Online]. Available: https://github.com/NVlabs/diffstack/tree/CTT_release

[31] P. Karle, M. Geisslinger, J. Betz, and M. Lienkamp, "Scenario Understanding and Motion Prediction for Autonomous Vehicles - Review and Comparison," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 10, pp. 16 962–16 982, 2022.

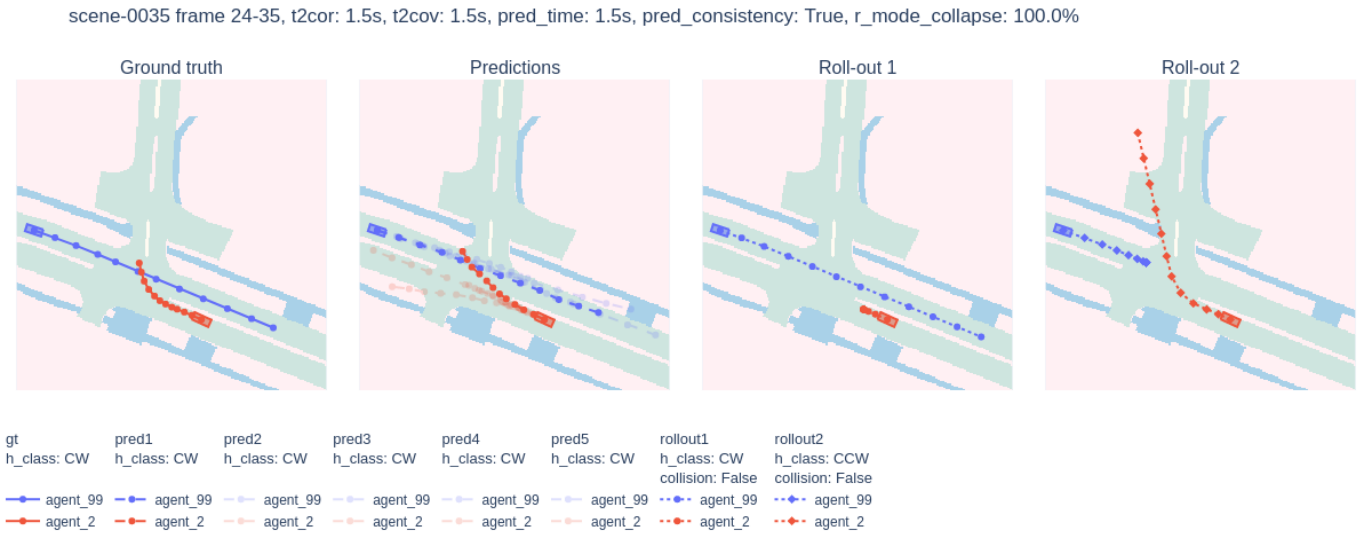
[32] R. Krajewski, T. Moers, J. Bock, L. Vater, and L. Eckstein, "The rounD Dataset: A Drone Dataset of Road User Trajectories at Roundabouts in Germany," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2020, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9294728

[33] J. Bock, R. Krajewski, T. Moers, S. Runde, L. Vater, and L. Eckstein, "The inD Dataset: A Drone Dataset of Naturalistic Road User Trajectories at German Intersections," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, Oct. 2020, pp. 1929–1934, iSSN: 2642-7214. [Online]. Available: https://ieeexplore.ieee.org/abstract/document/9304839

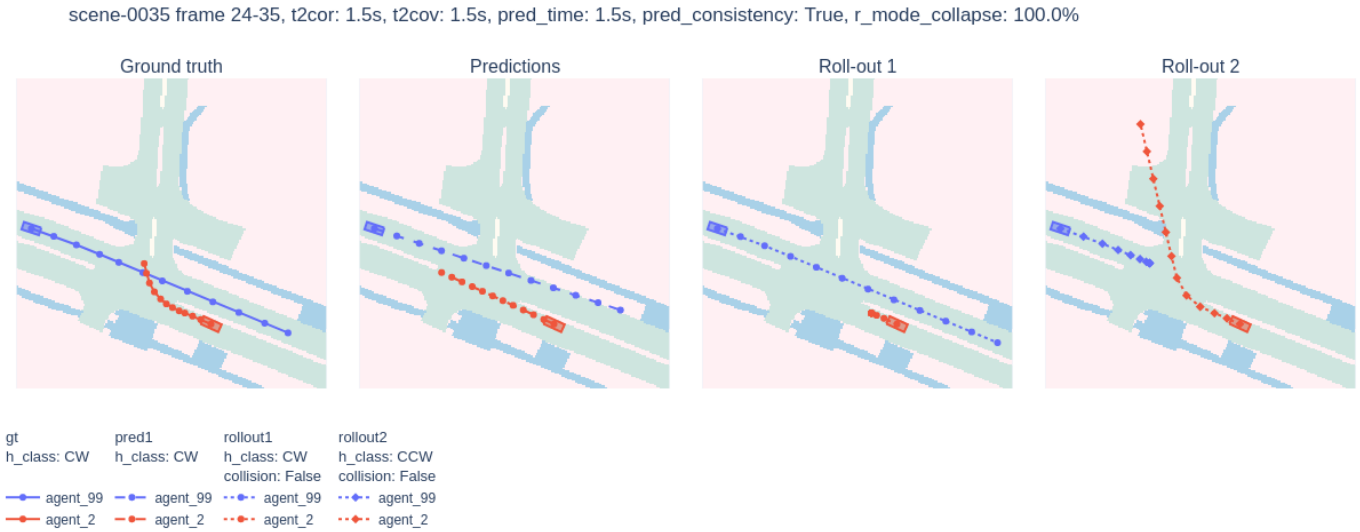
[34] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kummerle, H. Konigshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERnational, Adversarial and Cooperative moTION Dataset in Interactive Driving Scenarios with Semantic Maps," Sep. 2019, arXiv:1910.03088 [cs, eess]. [Online]. Available: http://arxiv.org/abs/1910.03088

## APPENDIX

We analyze qualitative results of AgentFormer (AF) and the constant velocity (CV) model for a number of scenes, see Figure 10, Figure 11, Figure 12, Figure 13, Figure 14, Figure 15, Figure 16, Figure 17, Figure 18. The top of the figures show the predictions of AF, and the bottom those of the CV model. The visualizations are made one frame before the inevitable homotopy state. The time-based interaction metrics, which are calculated over the preceding frames, are shown in the title of the plots. The total duration of the evaluated interaction frames is denoted by `pred_time`. We only show the interacting agent-pairs, and the ego-agent is denoted by `agent_99`. The legend shows the interaction classes for all trajectories, as well as the collision states of the roll-outs. The interaction class is also visualized in the plots by using $\circ$ and $\diamond$ markers for the CW and CCW class, respectively. We analyze the results for each scenario in the figure's caption.

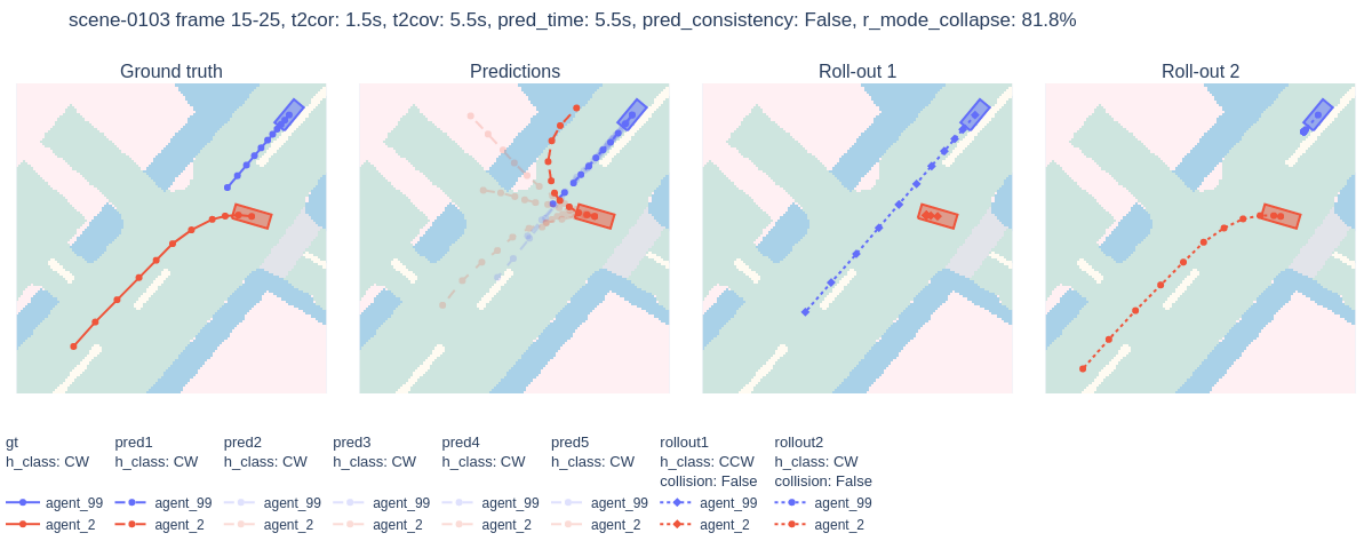



(a) AgentFormer




(b) CV Model

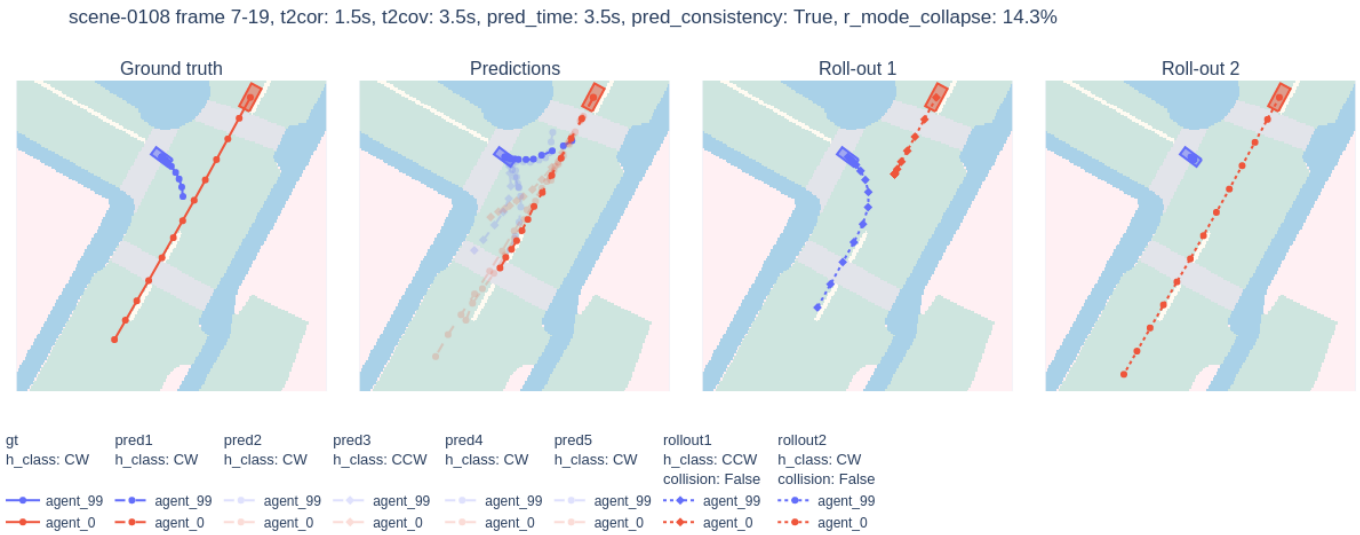
Fig. 10: Interaction prediction performance comparison on nuScenes, scene-0035, agent-pair (99,2). Both models correctly predict the mode at the beginning of the interaction interval (1.5s in this case). AF even correctly predicts the route intention of agent 2. However, AF does suffer 100% mode collapse, as roll-out 2 was feasible, but not predicted in any of the preceding frames. That being said, the CCW class seems highly unlikely, based on the observed speed differences of the vehicles (visualized as the distance between the markers).



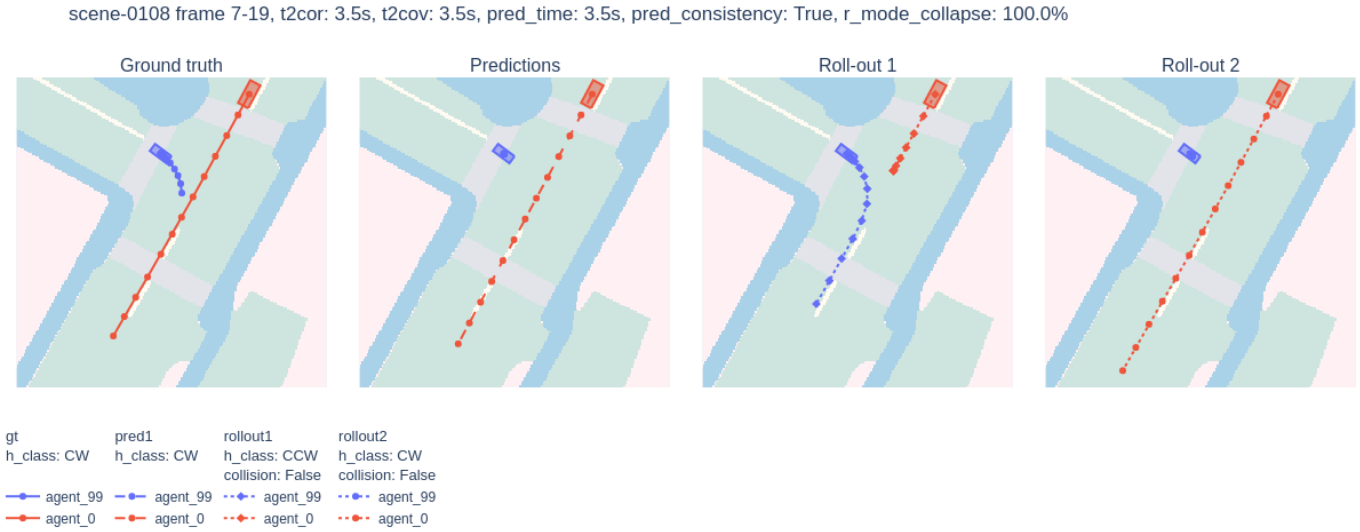

(a) AgentFormer


(b) CV Model

Fig. 11: Interaction prediction performance comparison on nuScenes, scene-0103, agent-pair (99,2). AF outperforms the CV model here, and correctly assesses the interaction mode at 1.5s before it happens instead of 1s. At the start of this scenario, agent 99 was travelling at a higher speed, whereas agent 2 was static, so at the start the CCW class was actually more likely. This explains why the CV model only correctly predicts the interaction once the speeds of the vehicles have changed accordingly. We also note that the ML prediction is really poor in terms of route intention: even though the vehicle is already heading towards making a left turn, the ML prediction is a right turn in the wrong traffic direction. However, as this prediction does encapsulate a CW prediction, the interaction class is correct. This showcases the need for holistic evaluation, i.e., evaluating predictions with a range of metrics assessing accuracy, diversity, admissibility, etc.



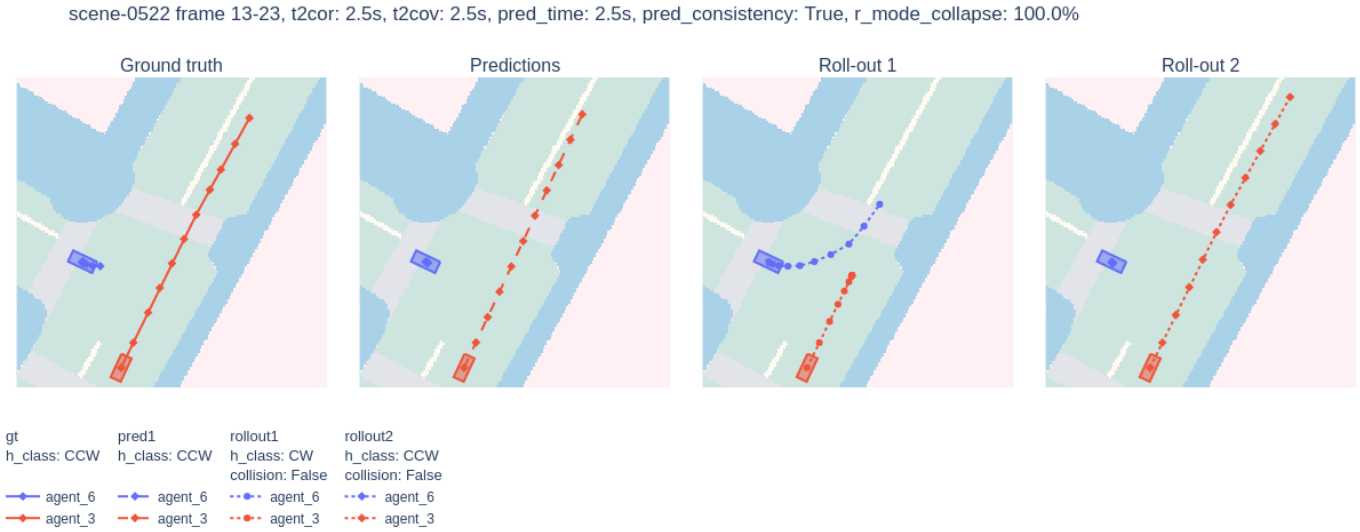

(a) AgentFormer




(b) CV Model

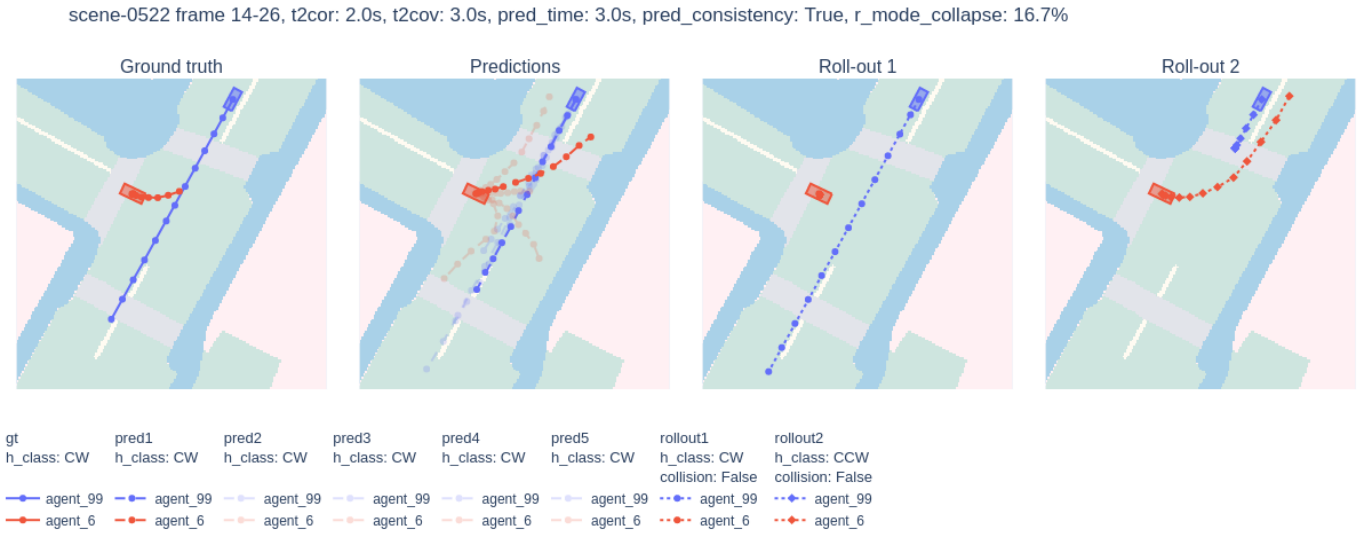
Fig. 12: Interaction prediction performance comparison on nuScenes, scene-0108, agent-pair (99,0). In this example, the CV model outperforms AF, by simply extrapolating the current paths. As agent 99 is static, this yields the correct class: CW. AF covers this class from the beginning of the prediction time, but only correctly predicts the class 1.5s before the interaction happens.


(a) AgentFormer

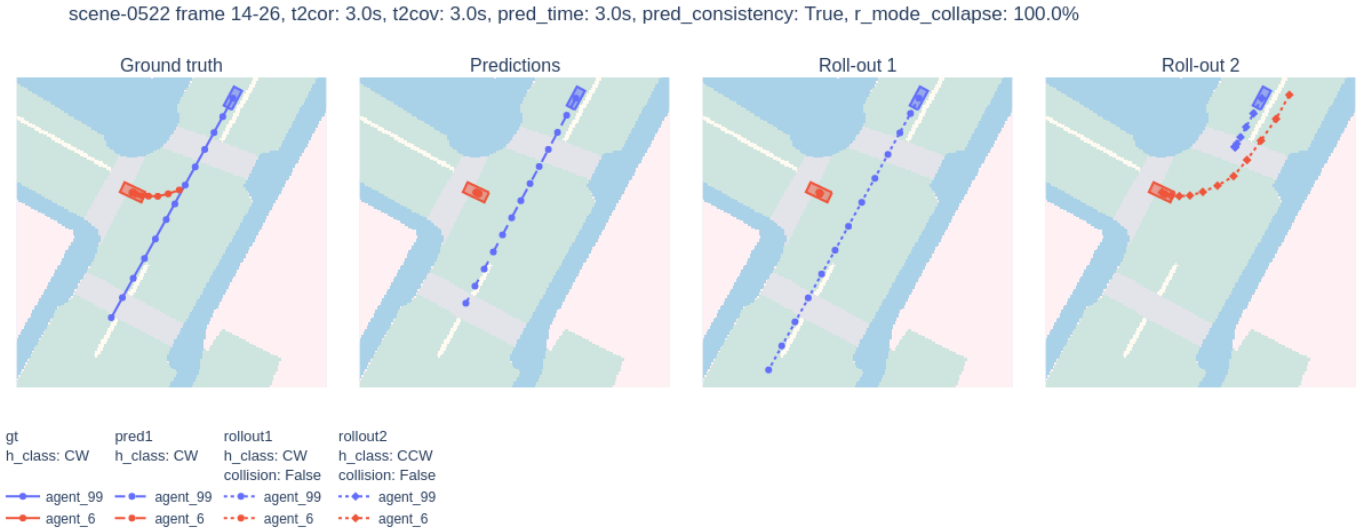



(b) CV Model

Fig. 13: Interaction prediction performance comparison on nuScenes, scene-0522, agent-pair (6,3). Again, we see that the CV model outperforms AF from the beginning of the horizon. AF tends to predict more dynamic trajectories where the vehicles move a lot. Even here, at the visualized frame, we see that the margins between trajectories of the ML predictions are very small.

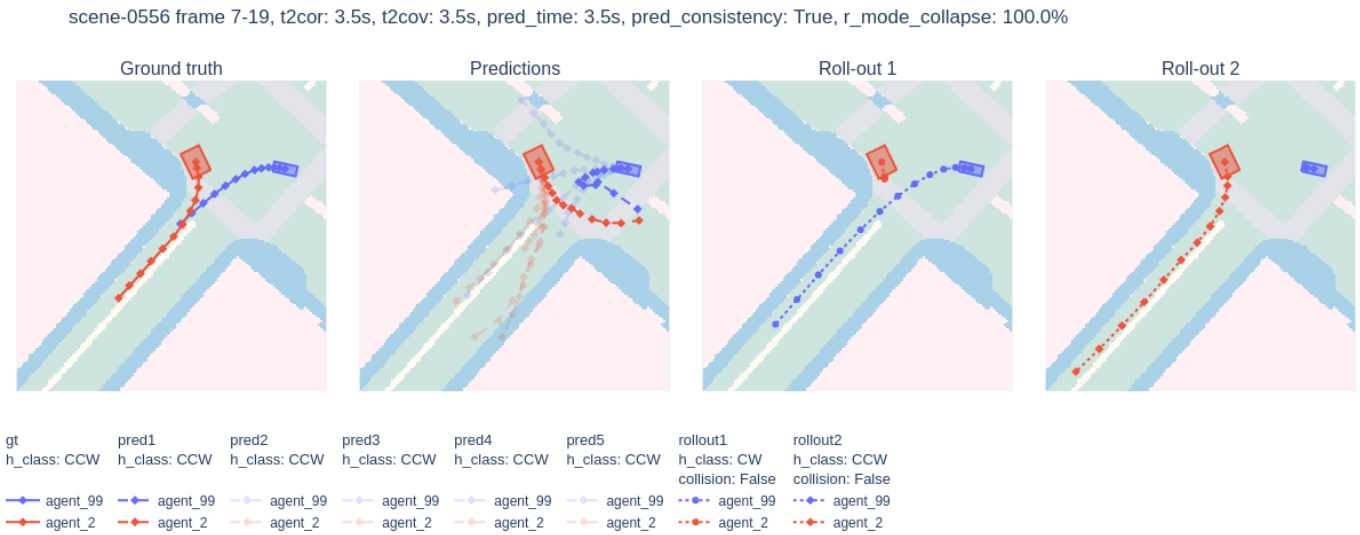



(a) AgentFormer

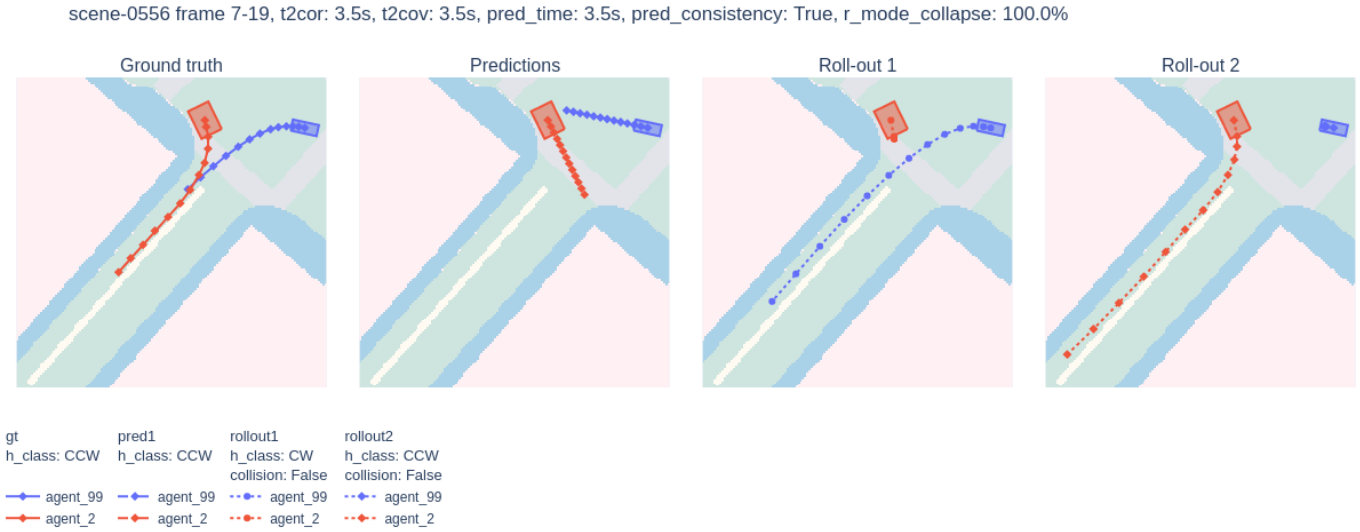



(b) CV Model

Fig. 14: Interaction prediction performance comparison on nuScenes, scene-0522, agent-pair (99,6). We look at the same scene as in the previous figure, only this time the interaction between agent 6 and agent 99. Again, AF overshoots the trajectory of agent 6 (blue vehicle), and the CV model is able to correctly predict the interaction class 0.5s earlier. However, AF does cover both interaction classes in most of the preceding frames, resulting in the low mode collapse ratio.

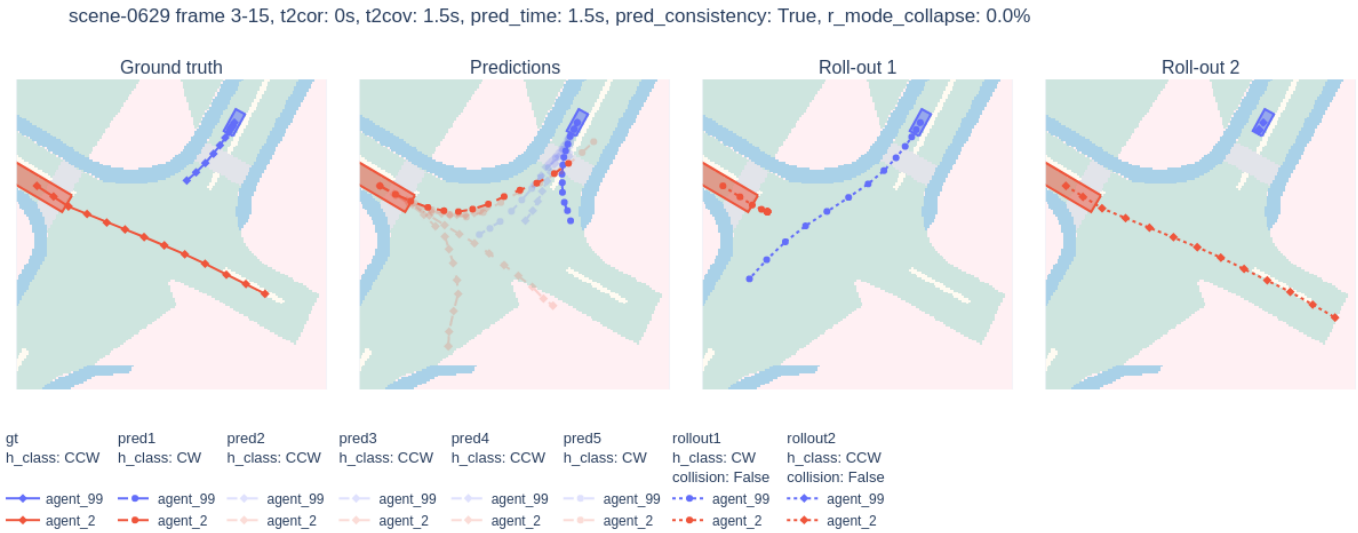



(a) AgentFormer

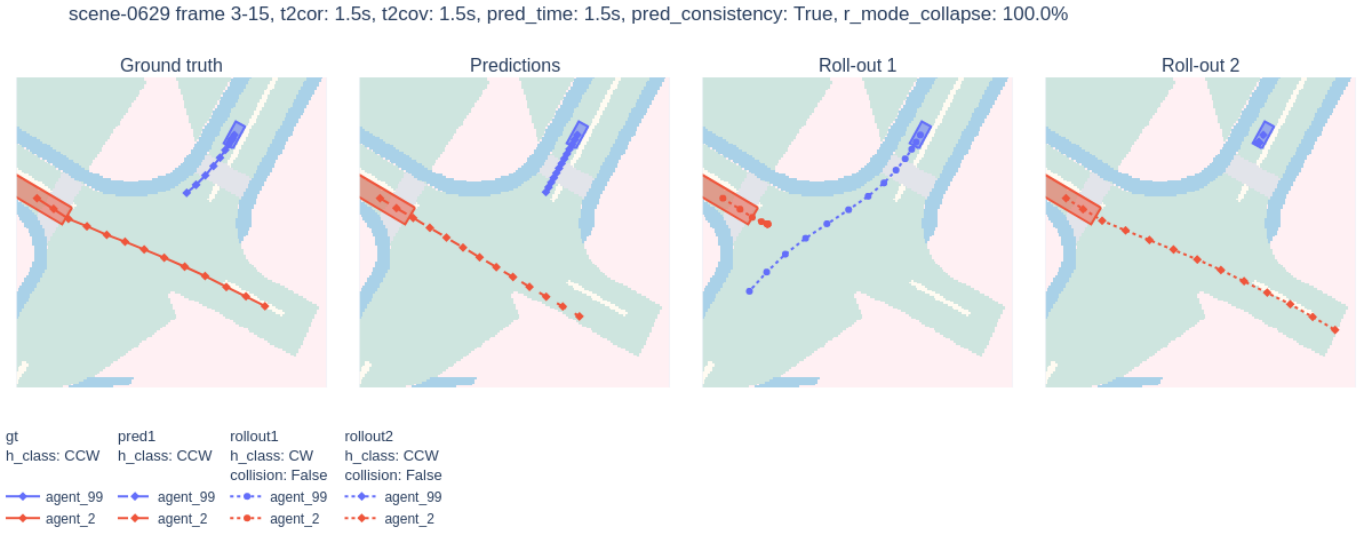



(b) CV Model

Fig. 15: Interaction prediction performance comparison on nuScenes, scene-0556, agent-pair (99,2). Both models perfectly predict and cover the ground truth mode. While feasible, the CW (roll-out 1) mode is not predicted at all. We also note, that AF's ML prediction is clearly agent-aware, as it seems to avoid a collision between the agent-pair. However, the turning radius of agent 99's trajectory (blue vehicle), is clearly not dynamically feasible.

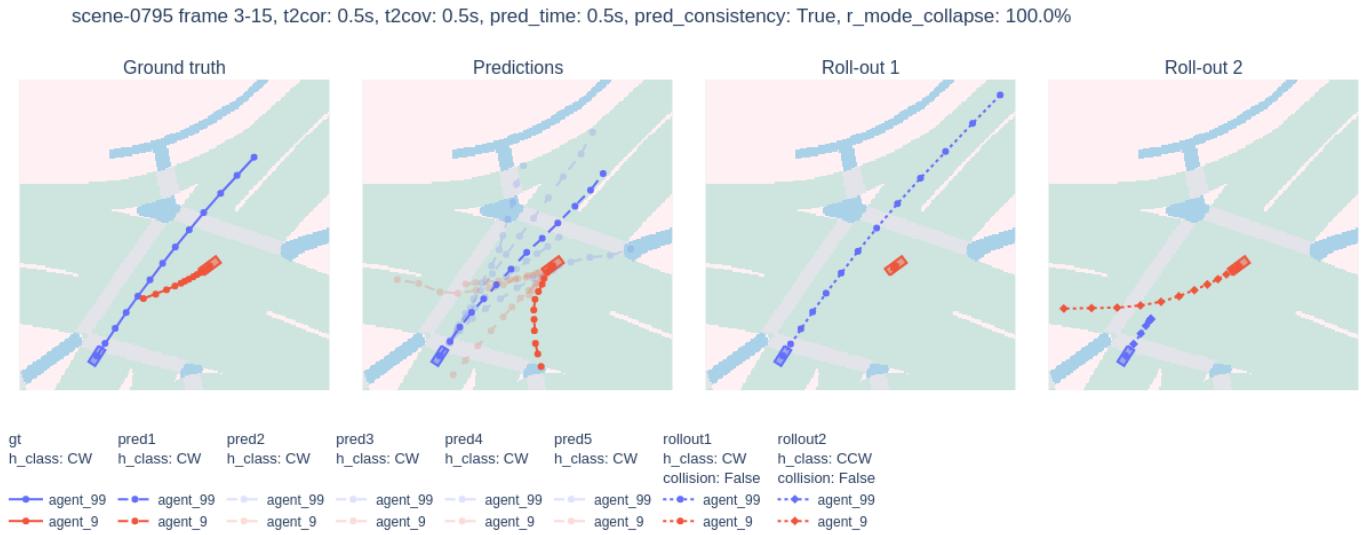



(a) AgentFormer

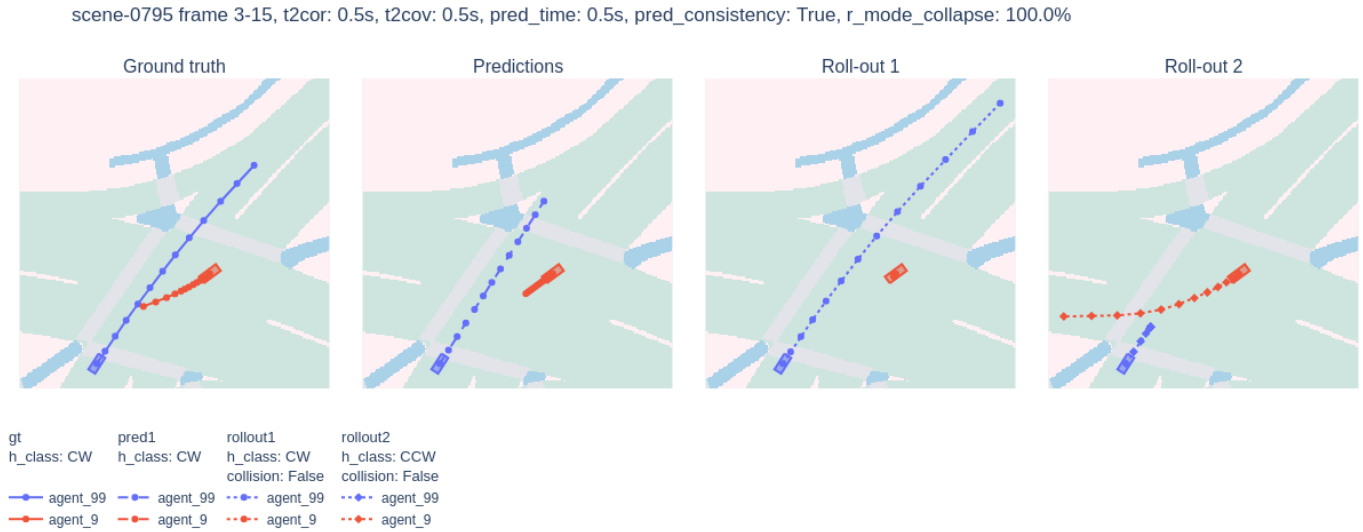



(b) CV Model

Fig. 16: Interaction prediction performance comparison on nuScenes, scene-0629, agent-pair (99,2). The CV model correctly predicts the interaction class from the beginning. Interestingly, AF fails to do so, not even right before the interaction class becomes inevitable. AF does predict higher speeds for agent 2 than for agent 99, however, because of the predicted route intentions, this still results in a CW class. This prediction still implies that agent 99 will go first, as can be seen from the prediction's subplot. However, this example also shows that it is not possible to assess the consequences of incorrect mode predictions; even though the time-to-correct-mode-prediction is 0s, the vehicles are relatively far apart in this scenario.

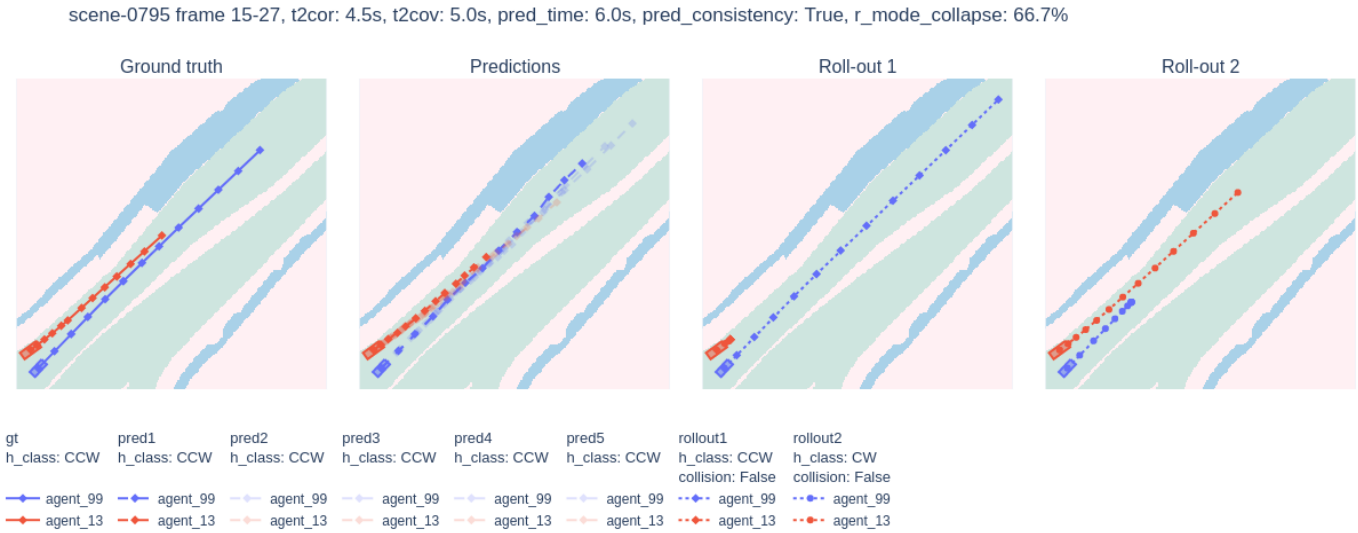



(a) AgentFormer

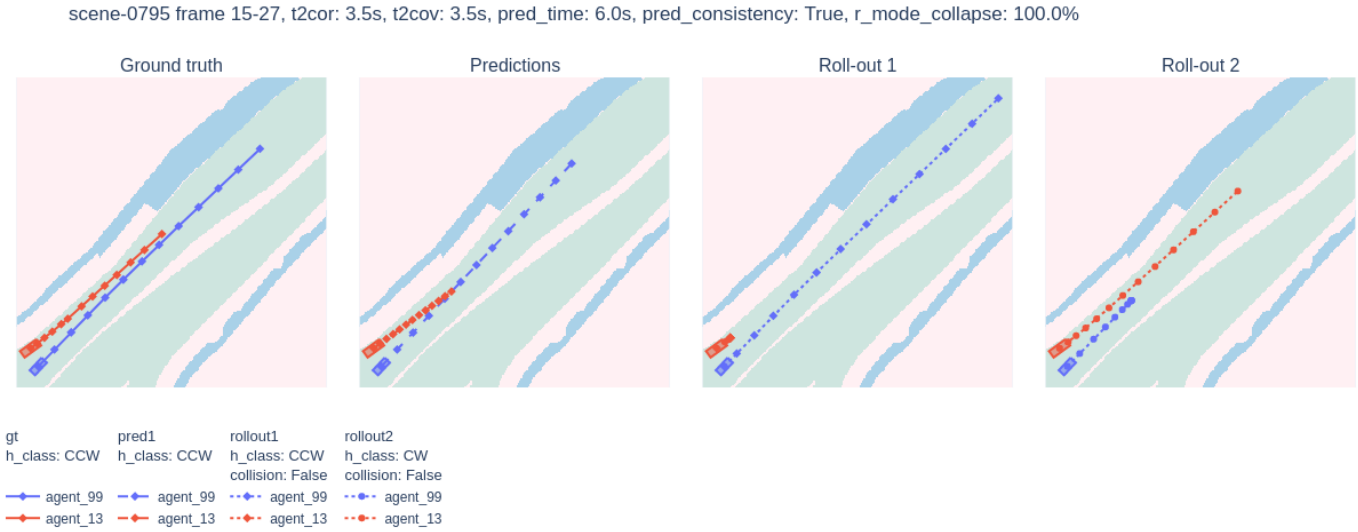



(b) CV Model

Fig. 17: Interaction prediction performance comparison on nuScenes, scene-0795, agent-pair (99,9). Both models correctly assess the interaction class from the beginning of the prediction horizon, which is just a single frame in this example (agent 9 is not annotated earlier in the data). While feasible, roll-out 2 seems highly unlikely in this scenario, given the speed differences between the vehicles and the tight collision margins for roll-out 2. Therefore, it is not surprising nor a problem, that the CCW class is not covered in this case.




(a) AgentFormer




(b) CV Model

Fig. 18: Interaction prediction performance comparison on nuScenes, scene-0795, agent-pair (99,13). In this merging scenario, AF outperforms the CV model and already covers the ground truth class 5s before it happens, and correctly predicts it 4.5 seconds before, versus the CV model's 3.5s. This shows that in some scenarios, when the inevitable homotopy state is relatively far away, the CV model cannot predict the correct interaction class as quick as AF.