

Beelines: Motion Prediction Metrics for Self-Driving Safety and Comfort

Skanda Shridhar, Yuhang Ma, Tara Stentz, Zhengdi Shen, Galen Clark Haynes, Neil Traft

Abstract— The commonly used metrics for motion prediction do not correlate well with a self-driving vehicle’s system-level performance. The most common metrics are *average displacement error (ADE)* and *final displacement error (FDE)*, which omit many features, making them poor self-driving performance indicators. Since high-fidelity simulations and track testing can be resource-intensive, the use of prediction metrics better correlated with full-system behavior allows for swifter iteration cycles. In this paper, we offer a conceptual framework for prediction evaluation highly specific to self-driving. We propose two complementary metrics that quantify the effects of motion prediction on safety (related to recall) and comfort (related to precision). Using a simulator, we demonstrate that our safety metric has a significantly better signal-to-noise ratio than displacement error in identifying unsafe events.

I. INTRODUCTION

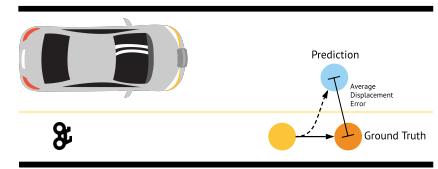
In self-driving systems, actor detection and motion prediction are essential. Most methods train machine-learned models to predict trajectories or occupancy maps. These models are usually trained on variants of the L2 or cross-entropy loss and evaluated with metrics such as *average displacement error (ADE)* and *final displacement error (FDE)*. However, consider the shortcoming illustrated by Fig. 1, which shows two simple scenarios with identical displacement error, but in which the error in Fig. 1b is more severe because the vehicle has not predicted the object’s future entry into its path.

As a result, evaluating predictions’ impact on on-road behavior usually entails full-system simulation and track testing. Both are expensive and require a fully-capable self-driving platform. Moreover, system factors beyond the actor motion prediction method (for instance, the particular planning algorithm) can influence results, making it difficult to isolate the predictions’ impact. A desirable motion prediction metric would anticipate its system-level impact while remaining agnostic of the broader system.

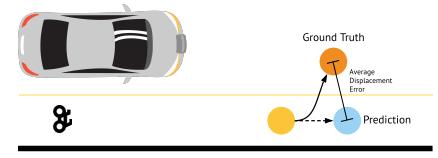
The *safety* impact of predictions is related to recall. When we forecast an actor’s future motion, we predict what parts of the world they will occupy. It follows then that *recall* is the percentage of the actor’s future occupied space that has been covered by the predictions made. Low recall suggests that a self-driving vehicle does not correctly anticipate the actor’s future occupancy regions, a safety concern.

Conversely, the *comfort* impact of predictions is related to precision. Precision is the fraction of the space predicted occupied that does end up as occupied. Low precision suggests that the system is over-predicting, and this will make the ego-vehicle brake unnecessarily. In extreme cases, it may be unable to move.

The authors are with Uber Advanced Technologies Group, PA, USA.



(a) The future prediction (blue) is in the vehicle’s path while the future ground-truth (orange) is not.



(b) The future ground-truth (orange) is in the vehicle’s path while the future prediction (blue) is not.

Fig. 1: Two examples of prediction errors, with identical displacement error. The bottom is more of a safety concern.

This work proposes a metrics framework that directly connects precision and recall to the self-driving specific concerns of ride comfort and safety. It addresses many flawed properties of metrics that evaluate predictions in isolation (such as ADE), including:

- Insensitivity to Object Shape and Orientation Error.
Solution: Since we deal with occupied space, we account for the position, orientation, shape, and uncertainty error.
- Reporting a Single Horizon or Aggregating Over All.
Solution: Since we evaluate predictions in relation to ego-vehicle spatiotemporal maneuvers, we consider future horizons holistically.
- Insensitivity to Ego-Vehicle (as in Fig. 1).
Solution: We grade safety based on the predictions’ ability to “protect” ground-truth “exposed” to the ego-vehicle, and comfort based on the extent to which predictions deprive the ego-vehicle of maneuvers.

While system-level evaluation (e.g., simulation) addresses the same concerns, it has shortcomings of its own, which we seek to overcome:

- Inability to Isolate Detection and Prediction Impact from Broader System.
Solution: Since we use fixed ego-vehicle maneuvers, we anticipate the system-level impact. Since these are system-agnostic, we isolate detection and prediction impact from the broader system.
- One Scenario Per Simulation.

Solution: We evaluate 1000s of counterfactual ego-vehicle behaviors and marginalize over these in a single pass.

- Difficulty Backpropagating Error Gradients.

Solution: Our approach lends itself to a GPU vectorized implementation through which we can backpropagate error gradients.

We show that our method is better than displacement error at marking prediction failures that lead to unsafe events in a simulator.

II. RELATED WORK

The most widely used metrics, ADE and FDE, were proposed in [1] and popularized by the TrajNet benchmark [2], [3]. Since it is unclear what the most important prediction horizon is, works such as [4], [5] report these metrics at multiple horizons. The nuScenes Prediction Challenge [6] and Argoverse Motion Forecasting Challenge [7] use the *minADE* variant, also referred to as “oracle error” [8], which takes the minimum error over the top k trajectories. Many multimodal prediction models [9], [8], [10], [11] use this, but it does not penalize false-positive modes, and we must consider it alongside ADE for a complete picture.

The Lyft Motion Prediction Competition [12] and other works [13], [14], [15] output a 2D probability distribution in the state space, and so can replace ADE with negative log-likelihood (NLL).

All these metrics have two drawbacks: 1) They only consider a single point on an object and do not account for orientation, shape, or relevance to the ego-vehicle, and 2) They require *ground-truth matching* to use in actual practice. Since we define them on a pair of points (prediction and the ground-truth), we must first match each predicted actor with a ground-truth label. If we cannot associate a prediction (false-positive), we must ignore it and likewise for ground-truth. To capture these omissions, we must rely on other metrics, and this lack of comprehensiveness is a drawback.

Methods like ours, which evaluate an occupancy representation instead of trajectories, can transcend this limitation [16], [17], [18]. Their evaluation is usually done with NLL or cross-entropy loss. Lacking awareness of the ego-vehicle, this suffers the same fundamental drawback highlighted in Fig. 1.

Jain et al. (2019) [16] also report *safety-sensitive* recall, which takes the drivable region into account and is therefore closely related to our work. While this gives some safety-related feedback, it does not incorporate comfort nor relevance to the ego-vehicle. Our framework addresses these factors in full.

III. METHOD OUTLINE

We evaluate actor predictions in relation to a set of fixed maneuvers the ego-vehicle may perform, the *ego-vehicle trajectories*. We frame the safety problem by asking: Given a distribution of ego-vehicle trajectories, how well do the predictions protect real-world objects? The comfort problem

is similarly posed: Given the same distribution of maneuvers, what fraction are needlessly blocked? To quantify the answers, we:

- Generate a set of fixed ego-vehicle trajectories to approximate the full set of dynamically feasible maneuvers. (Section VI)
- Given the prediction output, for each ego-vehicle trajectory,
 - Compute the likelihood that any exposed objects along that trajectory are not “blocked” by predictions. (Section IV)
 - Compute the likelihood that any reachable free space along that trajectory is “blocked” by predictions. (Section IV)
- Marginalize across all trajectories. (Section V)

IV. PROTECTION AND EXPOSURE

Let us first assume that we have a set of ego-vehicle trajectories available (we discuss generating this in more detail in Section VI). We can think of an ego-vehicle trajectory as a region of fixed area (the ego-vehicle’s body) moving through time. At a given time, we refer to the region as the ego-vehicle’s *footprint*. Figure 2 represents the ego-vehicle’s trajectory as a sequence of its footprints. It also represents ground-truth actors and predictions with their footprints in the same spatiotemporal volume.

When executing a fixed trajectory, if one region prevents the ego-vehicle from accessing another by requiring it to stop, the former *blocks* the latter. For instance, if queued behind a line of cars, the car immediately in front blocks all subsequent cars. In Fig. 2, actor G is blocked by actor A at the earlier time. Predictions have the effect of blocking real-world actors since they induce the ego-vehicle to stop.

A. Protection of Space

The *protection* of a space is the probability that a prediction blocks that space.

In the spatiotemporal grid setting of Fig. 2, let $P_{pred}(x)$ be the predicted probability that a cell x is occupied. The system either produces this directly or produces an output from which this can be derived. Given the ego-vehicle trajectory’s footprint at time t , denoted F_t , the predicted probability that at least one cell in the footprint is occupied is given by,

$$P_{pred}(F_t \text{ is Occupied}) = 1 - \prod_{x \in F_t} [1 - P_{pred}(x)] \quad (1)$$

At time H , the probability of F_H being unprotected is the joint probability of no predicted occupancy in any footprint up to H :

$$P(F_H \text{ is Unprotected}) = \prod_{t=t_1}^H [1 - P_{pred}(F_t \text{ is Occupied})] \quad (2)$$

where t_1 is the time of the first footprint of interest. Note that we have assumed independence across spatiotemporal cells.

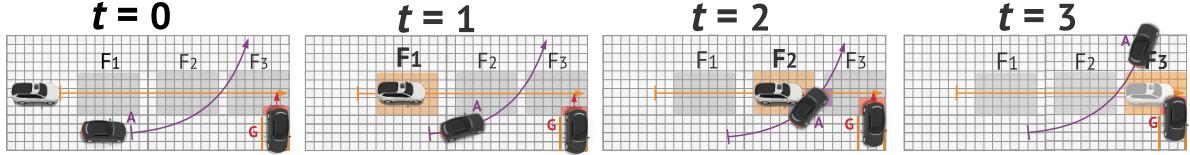


Fig. 2: At $t = 0$, a fixed ego-vehicle trajectory is shown as a sequence of its future footprints, F_1 , F_2 , and F_3 . At $t = 1$, the ego-vehicle occupies F_1 . At $t = 2$, it occupies F_2 and overlaps A 's space. It cannot reach F_3 and encounter G at $t = 3$ (hence shown faded) since it must re-plan to stop for A , so A blocks G along this specific ego-vehicle trajectory.

If A in Fig. 2 is a prediction, and G is a ground-truth object. Then,

$$\begin{aligned} P(F_3 \text{ is Unprotected}) &= [1 - P_{pred}(F_1 \text{ is Occupied})] \\ &\cdot [1 - P_{pred}(F_2 \text{ is Occupied})] \cdot [1 - P_{pred}(F_3 \text{ is Occupied})] \\ &= [1 - (1 - \prod_{x \in F_1} [1 - P_{pred}(x)])] \cdot [1 - (1 - \prod_{x \in F_2} [1 - P_{pred}(x)])] \\ &\cdot [1 - (1 - \prod_{x \in F_3} [1 - P_{pred}(x)])] \\ &= 1 \cdot 0 \cdot 1 = 0 \end{aligned}$$

which quantifies our intuition that A blocks G , and therefore G cannot be unprotected.

B. Exposure of Space

The *exposure* of a space (such as an ego-vehicle footprint) is the probability that no ground-truth blocks that space. Let $P_{gt}(x)$ be the ground-truth probability that a cell x is occupied. We obtain this, for instance, by converting available ground-truth object polygons into a spatiotemporal occupancy grid. Then, given a footprint F_t at time t of a specific ego-vehicle trajectory,

$$P_{gt}(F_t \text{ is Occupied}) = 1 - \prod_{x \in F_t} [1 - P_{gt}(x)] \quad (3)$$

At time H ,

$$P(F_H \text{ is Exposed}) = \prod_{t=t_1}^{H-1} [1 - P_{gt}(F_t \text{ is Occupied})] \quad (4)$$

where t_1 is the first footprint's time. Note the use of $H - 1$ rather than H as in Eq. 2, which prevents us from crediting a ground-truth region for blocking itself. Real-world objects overlapping high-exposure footprints are vulnerable, and predictions should prioritize blocking them.

V. QUANTIFYING SAFETY AND COMFORT IMPACT

A. Quantifying Safety Across Ego-Vehicle Trajectories

Assume B is the set of fixed ego-vehicle trajectories. Let T be the set of valid trajectory horizons $\{0, 1, \dots, T_{max}\}$. We define a risky event λ as occurring when the system does not protect exposed real-world occupancy. We quantify this risk by finding the fraction of the exposed, reachable space that is occupied but unprotected:

$$P(\lambda) = \frac{\sum_{b,t \in B \times T} P(F_t^b \text{ is Reached})d(F_t^b)}{\sum_{b,t \in B \times T} P(F_t^b \text{ is Reached})e(F_t^b)} \quad (5)$$

where,

$$d(F_t^b) = P(F_t^b \text{ is Unprotected}).$$

$$P_{gt}(F_t^b \text{ is Occupied}) \cdot P(F_t^b \text{ is Exposed})$$

$$e(F_t^b) = P(F_t^b \text{ is Exposed})$$

$P(F_t^b \text{ is Reached})$ is the probability of the ego-vehicle occupying F_t^b , its footprint at time t when executing trajectory b .

Regions where $e(F_t^b) = 0$ represent footprints inaccessible to the ego-vehicle because they are blocked by ground-truth, and these are excluded from the risk calculation.

If A in Fig. 2 is a prediction and G is ground-truth, and there is only one ego-vehicle trajectory, and the footprints at all times are equally likely, i.e. $P(F_t \text{ is Reached}) = 1/3$,

$$P(\lambda) = \frac{\frac{1}{3} \cdot 1 \cdot 0 \cdot 1 + \frac{1}{3} \cdot 0 \cdot 0 \cdot 1 + \frac{1}{3} \cdot 0 \cdot 1 \cdot 1}{\frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1} = 0$$

which captures the intuition of A protecting G . If however, A is ground-truth,

$$P(\lambda) = \frac{\frac{1}{3} \cdot 1 \cdot 0 \cdot 1 + \frac{1}{3} \cdot 1 \cdot 1 \cdot 1 + \frac{1}{3} \cdot 1 \cdot 1 \cdot 0}{\frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 0} = 0.5$$

This is because two equally likely footprints (F_1 , F_2) are “reachable” to the ego-vehicle and one of them F_2 involves an unsafe event, i.e. colliding with unprotected ground-truth A . Since F_3 is blocked by A it is excluded.

If interested in a particular actor, we may marginalize the numerator of Eq. 5 over the subset of ego-vehicle trajectory footprints that intercept the actor. We term this actor-specific quantity $P(\lambda_{actor})$.

The expression $e(F_t^b)$, when marginalized in Eq. 5, represents the probability apportioned to ego-vehicle states not blocked by ground-truth. If we modify it as follows,

$$e'(F_t^b) = P(F_t^b \text{ is Exposed})P(F_t^b \text{ is Unprotected}) \quad (6)$$

we also exclude states blocked by predictions. In addition to safety, this penalizes constricting the ego-vehicle's space. The next section discusses isolating comfort impact.

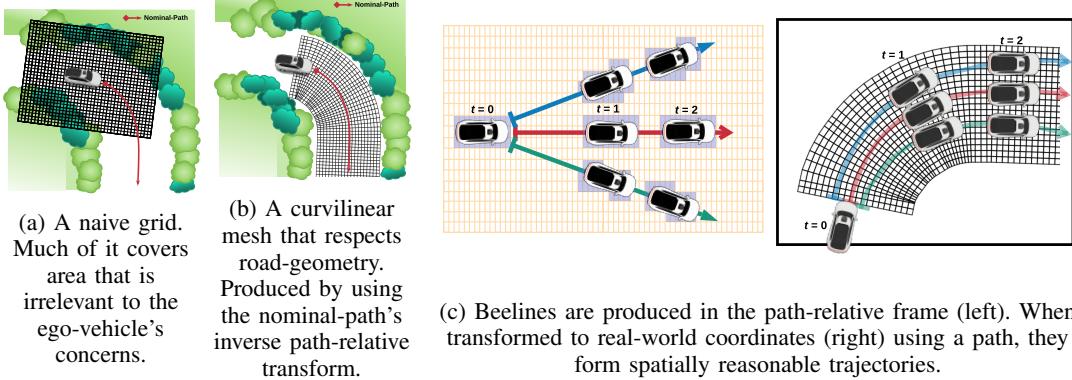


Fig. 3: Given a path, we can use the path-relative transformation to apply the same set of beelines to any scene.

B. Quantifying Comfort Across Ego-Vehicle Trajectories

A comfort violation ζ occurs when the predictions block free footprints. We capture this by finding the fraction of exposed, reachable space that is unoccupied but blocked by predictions:

$$P(\zeta) = \frac{\sum_{b,t \in B \times T} P(F_t^b \text{ is Reached})h(F_t^b)}{\sum_{b,t \in B \times T} P(F_t^b \text{ is Reached})g(F_t^b)} \quad (7)$$

where,

$$\begin{aligned} h(F_t^b) &= (1 - P(F_t^b \text{ is Unprotected})) \cdot \\ &\quad (1 - P_{gt}(F_t^b \text{ is Occupied})) \cdot \\ &\quad P(F_t^b \text{ is Exposed}) \\ g(F_t^b) &= (1 - P_{gt}(F_t^b \text{ is Occupied})) \cdot \\ &\quad P(F_t^b \text{ is Exposed}) \end{aligned}$$

Regions where $g(F_t^b) = 0$ represent footprints that are blocked by real-world obstacles and inaccessible to the ego-vehicle, and we skip the calculation for these regions. In Fig. 2 if A is a prediction and G is ground-truth,

$$P(\zeta) = \frac{\frac{1}{3} \cdot 0 \cdot 1 \cdot 1 + \frac{1}{3} \cdot 1 \cdot 1 \cdot 1 + \frac{1}{3} \cdot 1 \cdot 0 \cdot 1}{\frac{1}{3} \cdot 1 \cdot 1 + \frac{1}{3} \cdot 1 \cdot 1 + \frac{1}{3} \cdot 0 \cdot 1} = 0.5$$

This score results from the prediction arriving a timestep too early. Of the two available footprints F_1, F_2 (G occupies F_3), A blocks one, and therefore deprives the ego-vehicle of half its maneuverable space.

VI. IMPLEMENTATION OF METHOD

If available, a motion planner may be used to produce the fixed ego-vehicle trajectories. However, this may be an excessive computational burden. Therefore we present a simple, performant approximation of a motion planner that uses the path-relative transformation.

A. Path-Relative Space

A Cartesian grid will frequently result in the situation of Fig. 3a, where much of the grid is irrelevant. A solution that adapts to the intended travel path, as shown in Fig. 3b, is preferable. The path-relative transformation enables this.

To transform a point $p_{abs} = (x, y)$ from real-world coordinates into path-relative, $p_{pr} = (a, c)$, we first define the origin as the closest point on the path to the ego-vehicle's position. Next, we find p_{proj} , the closest point on the path to p_{abs} . The along-path distance from the origin to p_{proj} gives the along-track coordinate, a . The distance between p_{abs} and p_{proj} along the normal to the path at p_{proj} gives the cross-track coordinate, c .

We define the inverse transformation similarly; given (a, c) in path-relative coordinates, we advance a distance a along the nominal-path. From there, we shift c in the normal direction to get p_{abs} .

B. Beelines

In path-relative space, we produce trajectories spanning a range of initial heading angles with constant acceleration over their lifetime. We call them *beelines*. When transformed into real-world coordinates, these beelines are spatially appropriate for the scene, as in Fig. 3c. Once generated, we can apply the set of beelines to any scene, given a nominal-path.

C. Probability of an Ego-Vehicle Footprint

We define independent distributions $f_\theta(\theta)$ and $f_a(a)$ over the initial heading θ and acceleration a of the ego-vehicle in path-relative space. Given a starting velocity v_i and a footprint time t , since $r = v_i t + \frac{1}{2} a t^2$, we can define a density of possible footprint centers in polar coordinates, using $f_\theta(\theta)$ and applying a change of variables to $f_a(a)$,

$$f(r, \theta \mid v_i, t) = \frac{2}{t^2} f_a\left(\frac{2(r - v_i t)}{t^2}\right) f_\theta(\theta) \quad (8)$$

where we assume constant acceleration and heading. Transforming to Cartesian coordinates, and distributing the density uniformly over time up to a maximum horizon T_{max} ,

$$f(x, y, t \mid v_i) = \begin{cases} \frac{2}{T_{max} t^2} \frac{f_a\left(\frac{2(\sqrt{x^2+y^2}-v_i t)}{t^2}\right)}{\sqrt{x^2+y^2}} f_\theta(\tan^{-1}(y/x)), \\ t \in (0, T_{max}] \\ 0, \text{ otherwise} \end{cases} \quad (9)$$

Next, we determine the probability of a footprint as follows,

$$\begin{aligned} P(F_t^{x,y|v_i} \text{ is Reached}) &= \\ \int_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \int_{y-\frac{\Delta y}{2}}^{y+\frac{\Delta y}{2}} \int_{t-\frac{\Delta t}{2}}^{t+\frac{\Delta t}{2}} f(x, y, t | v_i) dx dy dt \\ &\approx f(x, y, t | v_i) \Delta x \Delta y \Delta t \quad (10) \end{aligned}$$

Where $\Delta x \Delta y$, and Δt are the granularities with which we discretize the continuous spatiotemporal space of footprints. Equation 10 may then be used in Eq. 5 and Eq. 7 to yield $P(\lambda)$ and $P(\zeta)$.

VII. EVALUATION OF SAFETY METRIC

We evaluate this approach by testing how well it identifies unsafe system-level outcomes when running an intentionally crippled prediction system and comparing this against an L2 error baseline. We used a cell size of $\Delta x = \Delta y = 0.5m$ and path-relative grid dimensions $30m \times 10m$. We used $\Delta t = 0.3s$, a triangular distribution maxing out at ± 15 degrees for $f_\theta(\theta)$, and a Gaussian truncated at $\pm 3m/s^2$ for $f_a(a)$. Additionally, we used the stricter $e'(F_t^b)$ from Eq. 6 in Eq. 5 (we did not observe significant difference in outcomes using $e'(F_t^b)$ rather than $e(F_t^b)$). We also set $t_1 = H - 2$ in Eq. 2.

We took 99 challenging real-world scenarios encountered during manual data collection. Across all of these, there were several “unpredictable” pedestrians or cyclists. Human labelers tagged 145 of these as actors of interest (AOIs) because they were likely to interact with the ego-vehicle. Examples included people playing football in the middle of a busy intersection, panhandling, etc.

We then deployed an intentionally crippled motion prediction system (in which we curtailed trajectory lengths) to test this metric’s efficacy at identifying unsafe scenarios. After integrating into a self-driving stack, we simulated all 99 scenarios (using a high-fidelity simulator in which simulated actors mimic their real-world counterparts’ behavior). Eighteen scenarios produced unsafe outcomes, involving 27 AOIs (“unsafe AOIs”). We captured each actor’s worst $P(\lambda_{actor})$ during its time in the spatial region of interest (from Fig. 3b). Next, we did the same with L2 error at 3s for actors in the spatial region of interest. We then separately ordered actors by descending $P(\lambda_{actor})$ and L2 error (larger values are worse for both).

1) *Global Signal-To-Noise Ratio*: We measure the signal-to-noise ratio by calculating the fraction of genuine violations in the top N given an ordering of instances. First, we ordered all instances by both metrics separately. Then we filtered to the “unsafe AOIs” flagged by the simulator and plotted histograms of their rankings (Fig. 5a). When ordered by descending $P(\lambda_{actor})$, the “unsafe AOIs” rank higher (farther left), suggesting superior SNR. Additionally, Fig. 5b, which plots SNR against N shows that the $P(\lambda_{actor})$ SNR is 40% for $N = 10$, 35% for $N = 20$ and so on, significantly above L2 error.

Next, we examined orderings of *all* AOI actors (not just “unsafe” ones). As can be seen in Fig. 6a, the $P(\lambda_{actor})$ is better at delineating AOIs than L2 error (since more

density is concentrated left). The $P(\lambda_{actor})$ SNR is also much greater; 60% for $N = 10$.

2) *Actor Of Interest Sensitivity*: We also considered how well the two metrics rank an AOI within a given scenario relative to other non-AOI actors encountered. As can be seen from Fig. 5c and Fig. 6c, the $P(\lambda_{actor})$ metric is consistently capable of outperforming the L2 error baseline at ranking the AOI highly within a scenario.

3) *Qualitative Instances*: Consider Fig. 4a which depicts a person stepping into the street. The artificially crippled predictions (blue) fail to predict this actor’s ground-truth future motion (orange). Among pedestrians and cyclists, $P(\lambda_{actor})$ ranks this AOI 9th; when ordered by L2 error at 3s, it ranks 120th. The L2 error between proceeding down the sidewalk and stepping into the street is too slight. However, the safety consequence is large.

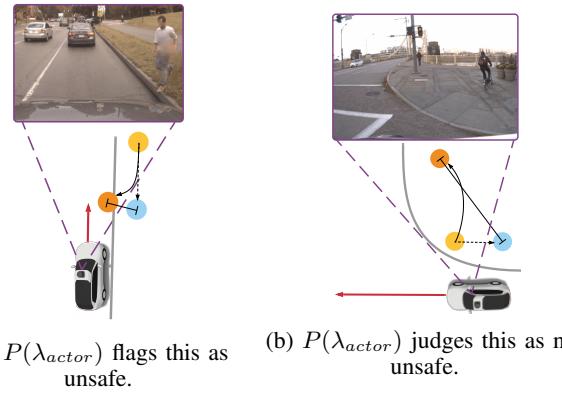


Fig. 4: Ground-truth at $t + 0$ s (yellow), $t + 3$ s (orange) and the predictions at $t + 3$ s (blue) for two real-world instances.

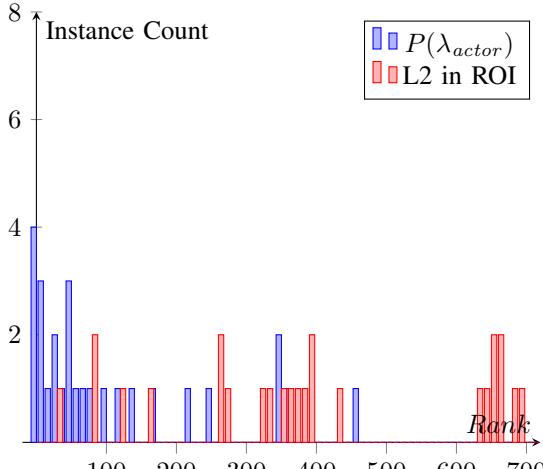
Figure 4b shows the opposite case. The prediction fails to anticipate the cyclist’s off-road trajectory, and a large L2 error results. Ordered by L2, this instance ranks 3rd amongst cyclists and pedestrians while $P(\lambda_{actor})$ ranks it 108th, which is more appropriate since there is little chance of a risky interaction with this actor.

VIII. CONCLUSIONS AND FUTURE WORK

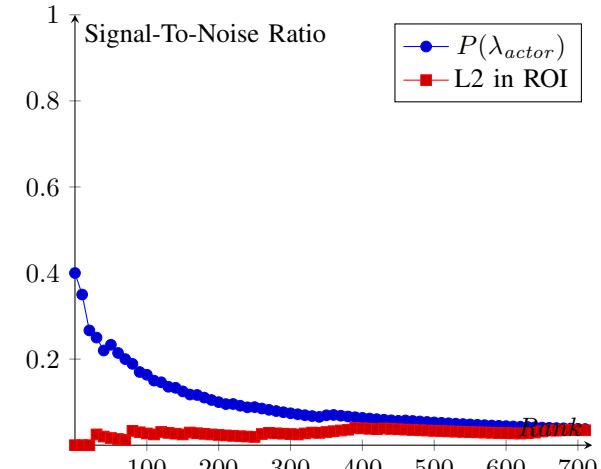
In this work, we demonstrated a method to directly translate the concepts of recall and precision into the self-driving specific metrics $P(\lambda)$ and $P(\zeta)$ reflecting safety and ride comfort. We proposed a general formulation that applies to *any* ego-vehicle trajectory distribution or spatiotemporal grid representation.

Next, we applied our formulation to a particular distribution of simple ego-vehicle trajectories (beelines), which we make agnostic to road-geometry using the path-relative transformation. We are thus able to use the same set of beelines for *any* potential scene, as long as a nominal path is available and the transformation is well defined.

Using a simulator, we demonstrated that our method shows a much better signal-to-noise ratio than displacement error at identifying risky events. This work evaluated the safety metric $P(\lambda)$; in future work, we plan to repeat the same for comfort $P(\zeta)$.



(a) Histogram of rankings for unsafe AOIs.

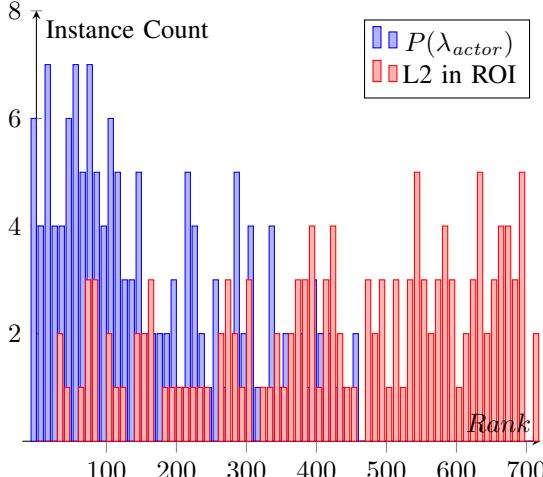


(b) Percentage of actors in each bin which are unsafe AOIs.

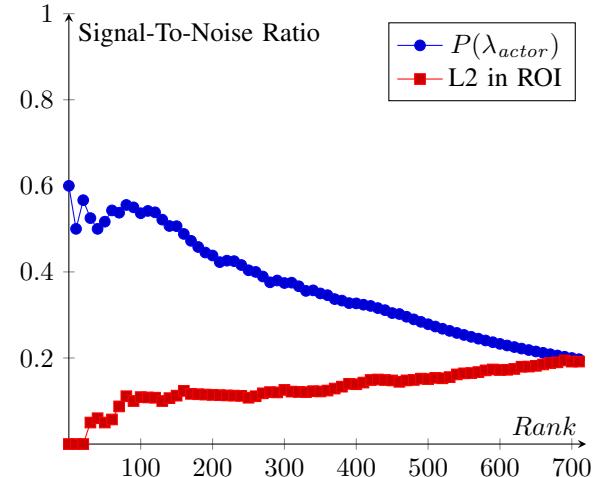
	P(λ_{actor})	L2 in ROI	P(λ_{actor})	L2 in ROI
Top 1	56% (15/27)	7% (2/27)	Top 5%	59% (16/27)
Top 2	78% (21/27)	30% (8/27)	Top 10%	63% (17/27)
Top 3	89% (24/27)	37% (10/27)	Top 25%	85% (23/27)
Top 4	96% (26/27)	56% (15/27)	Top 33%	89% (24/27)
Top 5	96% (26/27)	67% (18/27)	Top 50%	96% (26/27)

(c) Percentage of scenes where unsafe AOI ranked in Top K within scene (left) or Top Qth percentile within scene (right).

Fig. 5: Statistics for known “unsafe” AOIs



(a) Histogram of rankings for all AOIs.



(b) Percentage of actors in each bin which are AOIs.

	P(λ_{actor})	L2 in ROI	P(λ_{actor})	L2 in ROI
Top 1	41% (60/145)	9% (13/145)	Top 5%	43% (62/145)
Top 2	69% (100/145)	25% (36/145)	Top 10%	52% (76/145)
Top 3	81% (118/145)	35% (51/145)	Top 25%	75% (109/145)
Top 4	89% (129/145)	51% (74/145)	Top 33%	79% (114/145)
Top 5	91% (132/145)	59% (85/145)	Top 50%	91% (132/145)

(c) Percentage of scenes where AOI ranked in Top K actors within scene (left) or Top Qth percentile within scene (right).

Fig. 6: Statistics for all AOIs

REFERENCES

- [1] S. Pellegrini, A. Ess, K. Schindler, and L. Van Gool, "You'll never walk alone: Modeling social behavior for multi-target tracking," in *2009 IEEE 12th International Conference on Computer Vision*. IEEE, 2009, pp. 261–268.
- [2] A. Sadeghian, V. Kosaraju, A. Gupta, S. Savarese, and A. Alahi, "Trajnet: Towards a benchmark for human trajectory prediction," *arXiv preprint*, 2018.
- [3] S. Becker, R. Hug, W. Hubner, and M. Arens, "Red: A simple but effective baseline predictor for the trajnet benchmark," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 0–0.
- [4] N. Djuric, V. Radosavljevic, H. Cui, T. Nguyen, F.-C. Chou, T.-H. Lin, and J. Schneider, "Short-term motion prediction of traffic actors for autonomous driving using deep convolutional networks," *arXiv preprint arXiv:1808.05819*, 2018.
- [5] S. Becker, R. Hug, W. Hübner, and M. Arens, "An rnn-based imm filter surrogate," in *Scandinavian Conference on Image Analysis*. Springer, 2019, pp. 387–398.
- [6] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liang, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. (2020) nuscenes prediction task. [Online]. Available: <https://www.nuscenes.org/prediction>
- [7] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan *et al.* (2019) Argoverse motion forecasting competition. [Online]. Available: <https://evalai.cloudcv.org/web/challenges/challenge-page/454/evaluation>
- [8] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 336–345.
- [9] H. Cui, V. Radosavljevic, F.-C. Chou, T.-H. Lin, T. Nguyen, T.-K. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 2090–2096.
- [10] C. Rupprecht, I. Laina, R. DiPietro, M. Baust, F. Tombari, N. Navab, and G. D. Hager, "Learning in an uncertain world: Representing ambiguity through multiple hypotheses," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3591–3600.
- [11] T. van der Heiden, N. S. Nagaraja, C. Weiss, and E. Gavves, "Safecritic: Collision-aware trajectory prediction," *arXiv preprint arXiv:1910.06673*, 2019.
- [12] J. Houston, G. Zuidhof, L. Bergamini, Y. Ye, A. Jain, S. Omari, V. Iglovikov, and P. Ondruska. (2020) Lyft motion prediction for autonomous vehicles. [Online]. Available: <https://github.com/lyft/l5kit/blob/master/competition.md>
- [13] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2821–2830.
- [14] J. Schulz, C. Hubmann, J. Löchner, and D. Burschka, "Interaction-aware probabilistic behavior prediction in urban environments," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 3999–4006.
- [15] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.
- [16] A. Jain, S. Casas, R. Liao, Y. Xiong, S. Feng, S. Segal, and R. Urtasun, "Discrete residual flow for probabilistic pedestrian behavior prediction," *arXiv preprint arXiv:1910.08041*, 2019.
- [17] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.
- [18] W. Luo, B. Yang, and R. Urtasun, "Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 3569–3577.