Trajectory Prediction in Autonomous Driving with a Lane Heading Auxiliary Loss

Ross Greer *, Nachiket Deo †, Mohan Trivedi ‡

Laboratory for Intelligent & Safe Vehicles (LISA)
University of California, San Diego

Email: *regreer@eng.ucsd.edu, †ndeo@eng.ucsd.edu, ‡mtrivedi@eng.ucsd.edu

***Abstract*—Predicting a vehicle's trajectory is an essential ability for autonomous vehicles navigating through complex urban traffic scenes. Bird's-eye-view roadmap information provides valuable information for making trajectory predictions, and while state-of-the-art models extract this information via image convolution, auxiliary loss functions can augment patterns inferred from deep learning by further encoding common knowledge of social and legal driving behaviors. Since human driving behavior is inherently multimodal, models which allow for multimodal output tend to outperform single-prediction models on standard metrics; the proposed loss function benefits such models, as all predicted modes must follow the same expected driving rules. Our contribution to trajectory prediction is twofold; we propose a new metric which addresses failure cases of the off-road rate metric by penalizing trajectories that contain driving behavior that opposes the ascribed heading (flow direction) of a driving lane, and we show this metric to be differentiable and therefore suitable as an auxiliary loss function. We then use this auxiliary loss to extend the the standard multiple trajectory prediction (MTP) model, achieving improved results on the nuScenes prediction benchmark by predicting trajectories which better conform to the lane-following rules of the road.**

***Index Terms*—machine learning, deep learning, convolutional neural networks, multimodal trajectory prediction, trajectory quality metrics, safe autonomous vehicles, real-world driving data**

I. Introduction

Predicting a vehicle's trajectory is a critical functionality of any system working to plan safe paths on modern roadways. The problem of optimal prediction has been addressed and framed through many different computational methods, data constraints, and evaluation metrics. In this research, we propose a novel metric to analyze predicted trajectories and an auxiliary loss function based on this metric which can improve the predictive capabilities of learning-based approaches to trajectory prediction.

Trajectories in urban driving adhere to a constraint that can be leveraged to improve prediction results: drivers follow legal convention ascribed to roadways. Namely, a driver in a northbound lane must, by social and legal convention, drive north, and a driver in a southbound lane must similarly drive south. By penalizing a machine learning system which predicts a trajectory in discord with known lane directions, the system can learn this driving convention during its optimization.

II. Related Research

A. Agent-Scene Modeling and Multimodal Trajectory Prediction

Agent-scene modeling and multimodal trajectory prediction are two techniques employed to forecast vehicle motion. Agent-scene modeling techniques make use of available contextual data, often bird's-eye-view maps of the static scene, to serve as input to machine learning models. Ridel et al. [1] propose a method to embed agent information in a 2D grid, directly combining with existing geographical information, then use convolutional neural networks to demonstrate the value of agent-scene temporal grids. Cui et al.'s [2] multimodal trajectory prediction (MTP) acknowledges the insufficiency of a single predicted trajectory to standard driving situations; at a two-way intersection, one can turn left or turn right, so a single predicted trajectory cannot account for both of the driver's possible intentions.

Cui et al. and derivative works use agent-scene modeling for their prediction by passing a rasterized map of ego vehicle context as input to a convolutional neural network (CNN), and augmenting the convolutional output with ego vehicle state information prior to fully-connected layers.

Other approaches to trajectory prediction involve focus on patterns in surround agents to inform ego vehicle motion. For example, Deo and Trivedi [3] use a reinforcement learning approach with the same input of past motion and scene structure, and Messaoud et al. [4] further integrate of surround vehicle behavior to a deep learning model via social tensors. Both of these novel methods, among any other learning-based model, have the potential for learning additional scene information from the proposed auxiliary loss.

B. Auxiliary Loss Functions

MTP and other learning-based approaches to trajectory prediction use a loss function to drive model optimization. Cui et al. use a two-part loss function which penalizes on the basis of classification (the $L2$-closest mode should have the highest likelihood) and regression (the $L2$-closest mode trajectory should be close to the target). Niedoba et al. [5] propose an additive auxiliary loss function to improve performance of deep learning models which penalizes a trajectory in proportion to the distance of trajectory points to the nearest on-road map point, reflecting prior knowledge that a predicted trajectory will rarely leave the road. Auxiliary loss functions do not replace classification and regression loss, but rather augment the learning step by imposing additional penalties.

C. Predicted Trajectory Evaluation Metrics

Common metrics used in evaluation of motion forecasting include:

- Minimum average displacement error over k most probable trajectories ($\mathrm{MinADE_k}$), defined as the average of pointwise $L2$ distances between the predicted trajectory and ground truth over the k most likely predictions, averaged over all samples.
- Minimum final displacement error over k most probable trajectories ($\mathrm{MinFDE_k}$), defined as the $L2$ distance between the final points of the prediction and ground truth over the k most likely predictions, averaged over all samples.
- Miss rate at 2 meters over k most probable trajectories ($\mathrm{Miss\ Rate_{k,2}}$), defined as the proportion of misses (trajectories in which maximum pointwise $L2$ distance between the prediction and ground truth is greater than 2 meters for the k most likely predictions) over all agents.
- Off-Road rate, the proportion of predicted trajectories that include points outside the drivable area of the map.

In this work, we (1) define a new metric, the Off-Yaw Rate, as a measurement of a trajectory's adherence to lane direction, (2) make a case for the metric as an additive auxiliary loss function, then (3) offer experimental results by extending the MTP model to include this loss. Similar to the aforementioned off-road auxiliary loss, this loss function may be combined with existing deep learning models which use agent-scene modeling to enhance their performance.

III. Off-Yaw Rate as a Metric

A. Shortcomings of the Off-Road Rate Metric

Fig. 1 illustrates an example trajectory predicted by the MTP model. There is an obvious improvement available to the prediction: the vehicle should stay on the drivable area. By applying an off-road loss term, the prediction can be updated to the trajectory shown in Fig. 2. While this does address the original intent as quantified by the Off-Road Rate metric (i.e. the car should stay on the road), it has arguably created an even less safe situation, as the car is now headed into oncoming traffic. To address this problem, we introduce the Off-Yaw Rate metric and the YawLoss function.

B. Off-Yaw Rate

By accepted legal and social convention, when driving in a lane, the vehicle must move in the direction of the lane heading as to not interfere with other traffic. The Off-Yaw Rate is a measure of a trajectory's ability to orient in the direction of the nearest lane.

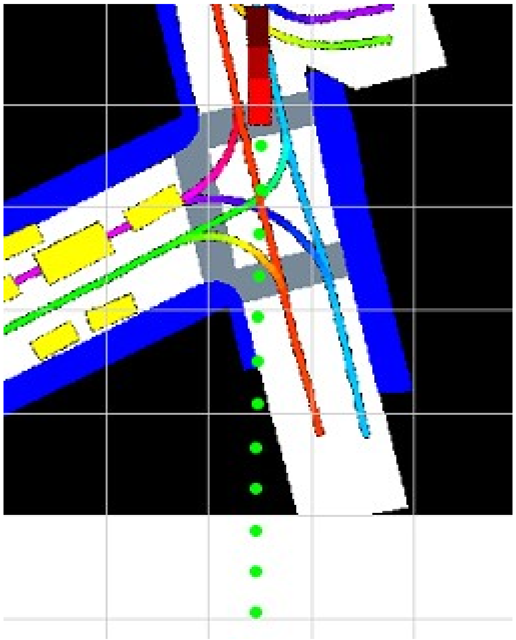


Fig. 1: A predicted trajectory (green, car in red) that leaves the drivable area.

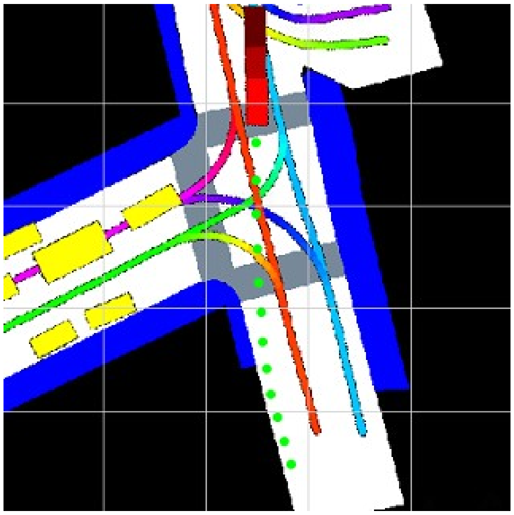


Fig. 2: The predicted trajectory has been improved according to the on-road metric, but is this actually a safer trajectory?

Define a vehicle's initial position on trajectory τ as $(x_0^\tau, y_0^\tau) = (0, 0)$, and its initial orientation in the local frame as $\theta = 0$ aligned with the standard y-axis. Given a trajectory of points $\tau = \{(x_0^\tau, y_0^\tau), (x_1^\tau, y_1^\tau), ..., (x_n^\tau, y_n^\tau)\}$, where points 1 through n correspond to predicted future points, we can estimate the vehicle heading relative to its initial orientation with the following procedure. First, we make use of the assumption that the trajectory sample rate relative to map scale is sufficiently high that we can accept a straight-line approximation between consecutive points. Let $(\hat{x}_i^\tau, \hat{y}_i^\tau)$ be the midpoint of two consecutive points $(x_i^\tau, y_i^\tau), (x_{i+1}^\tau, y_{i+1}^\tau)$, defined by the function:

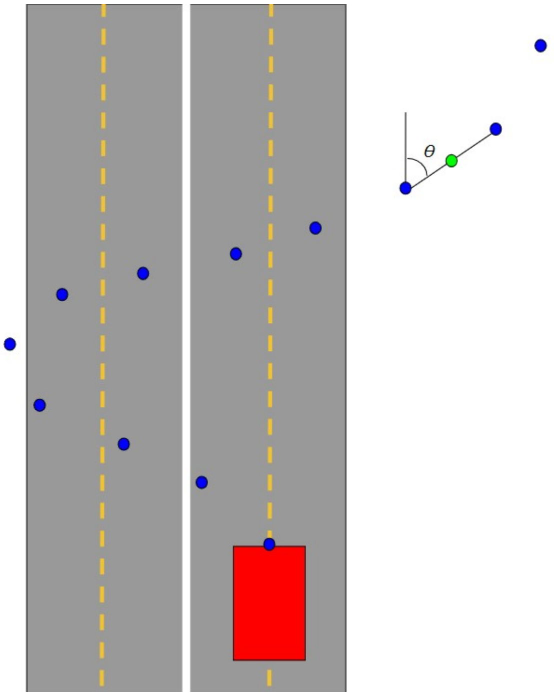


Fig. 3: The predicted trajectory of the ego vehicle (red) is shown in blue. The green circle represents a midpoint i between two points of the trajectory. The angle θ_i, in the local frame, is assigned to midpoint i.

$$(\hat{x},\hat{y})(x_1,y_1,x_2,y_2) = (\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2}) \tag{1}$$

The angle between the same two consecutive trajectory points surrounding $(\hat{x}_i^\tau, \hat{y}_i^\tau)$ is found using

$$\theta(x_1,y_1,x_2,y_2) = \arctan(\frac{x_2-x_1}{y_2-y_1}). \tag{2}$$

This angle $\theta(x_i^\tau, y_i^\tau, x_{i+1}^\tau, y_{i+1}^\tau)$ is then paired with the midpoint $(\hat{x}_i^\tau, \hat{y}_i^\tau)$. Fig. 3 illustrates this heading assignment method. In this manner, from a series of n estimated trajectory points, we create a series of n midpoints and associated headings relative to the initial orientation, which can be converted directly from the local frame to the global frame using the ego vehicle's rotation matrix. We refer to the i-th heading of a trajectory in the local frame as $\theta_{\tau,i}$, and the same heading in the global frame as $\theta_{\tau,i}^G$.

The angular difference between a trajectory midpoint heading in the global frame, θ and the heading of the nearest lane, $\theta_{NL}(x,y)$ can be calculated as follows:

$$\delta(x,y,\theta) = min(\theta - \theta_{NL}(x,y), \theta_{NL}(x,y) - \theta) \mod 360^\circ. \tag{3}$$

A successful measure of off-yaw driving should increase for any portion of the trajectory τ which deviates from the lane orientation. Further, greater angular differences should be assigned greater values than smaller angular differences. So, using each angular distance, the off-yaw measure of an n-point trajectory is:

$$Y(\tau) = \sum_{i=1}^{n} \delta(\hat{x}_i^\tau, \hat{y}_i^\tau, \theta_{\tau,i}^G). \tag{4}$$

Extending over all m predicted modes, we reach the per-sample average off-yaw expression:

$$Y = \sum_{\tau=1}^{m} Y(\tau) \tag{5}$$

C. Lane Change Approximations

There is a small margin of expected angular error, ϵ, which can account for minor adjustments to the vehicle heading in order to stay within the lane. In addition to lane-correcting error ϵ, a second exception to the assumption of driving in alignment with the nearest lane occurs when a driver changes lanes, during which their vehicle may orient at an angle no more than (and typically much less than) 90° to perform the lane change maneuver, with a 90° lane change occurring only when traffic is at a stop. Typical lane changes occur at angles relative to the flow of traffic and vehicle dynamics such as turning radius and velocity. Since a trajectory should is not off-yaw during a legal lane change, nor small-angle lane corrections, we therefore constrain the measure function to only penalize angular differences which exceed a threshold, α. The modified angular difference, $\hat{\delta}_i$, has the following formula:

$$\delta^\alpha(x,y,\theta) = \begin{cases} 0 & \delta(x,y,\theta) \leq \alpha \\ \delta(x,y,\theta) & \delta(x,y,\theta) > \alpha \end{cases} \tag{6}$$

For our experiments, we selected a threshold of 45°.

D. Off-Yaw in Intersections

When a vehicle passes through an intersection, the vehicle must cross over lanes which flow in discordant directions (look no further than the existence of stoplights as proof). For this reason, the measure should not count any deviation from the heading of the closest lane for midpoints which lie in an intersection. Thus, the measure is modified as the following function:

$$Y^\alpha(\tau) = \frac{1}{n} \sum_{i=1}^{n} I(x_i^\tau, y_i^\tau) \delta^\alpha(x_i^\tau, y_i^\tau), \tag{7}$$

where

$$I(x_i^\tau, y_i^\tau) = \begin{cases} 0 & (x_i^\tau, y_i^\tau) \text{ in intersection} \\ 1 & \text{otherwise} \end{cases} \tag{8}$$

.

Extending over all m predicted modes, we reach the modified per-sample off-yaw measure expression:

$$\bar{Y}^\alpha(T) = \sum_{\tau=1}^{m} Y^\alpha(\tau) \tag{9}$$

The Off-Yaw Rate for a set of samples and their predicted trajectory sets is the average fraction of trajectories which contain off-yaw events, defined in the following equation:

$$R_{\text{off-yaw}} = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{m}\sum_{\tau=1}^{m} I(Y^{\alpha}(\tau)) \tag{10}$$

where

$$I(Y) = \begin{cases} 0 & Y = 0 \\ 1 & Y > 0 \end{cases} \tag{11}$$

IV. YawLoss

A. Off-Yaw Metric as a Loss Function

In this section, we show that the Off-Yaw Metric in Eq. (9) is differentiable, and is therefore suitable as an auxiliary loss function which penalizes vehicle trajectories that move against the flow of traffic, which we name YawLoss.

We begin with (9) and differentiate with respect to network output set of trajectories $T = \{\tau_1, \tau_2, ..., \tau_m\}$. For brevity, we abbreviate $x_i^{\tau}, y_i^{\tau}, x_{i+1}^{\tau}, y_{i+1}^{\tau}$ as $\vec{x}_i^{\tau}$.

$$\nabla\bar{Y}^{\alpha}(T) = \frac{1}{m}\sum_{\tau=1}^{m}\nabla Y^{\alpha}(\tau) \tag{12}$$

$$= \frac{1}{mn}\sum_{\tau=1}^{m}\sum_{i=1}^{n}\nabla I(x_i^{\tau}, y_i^{\tau})\delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\vec{x}_i^{\tau})) \tag{13}$$

Since the sum of differentiable functions is differentiable, we continue our analysis with the sum term:

$$g(\vec{x}_i^{\tau}) = \nabla I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})\delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\vec{x}_i^{\tau})) \tag{14}$$

Computing the gradient, first for x_i^{τ}, we find:

$$\frac{\partial g}{\partial x_i^{\tau}} = \frac{\partial I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})}{\partial x_i^{\tau}}\delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\vec{x}_i^{\tau})) + I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})\frac{\partial\delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\vec{x}_i^{\tau}))}{\partial x_i^{\tau}} \tag{15}$$

Because the value of the function I in the expression

$$\frac{\partial I(\hat{x}_i^{\tau}, \hat{y}_i^{\tau})}{\partial x_i^{\tau}} \tag{16}$$

can only take on values of 0 or 1, the gradient function is simply 0 when the vehicle remains on-road or off-road, and the positive or negative reciprocal of the displacement of x_i^{τ} otherwise; in any case, defined for all input.

The function δ^{α} of

$$\frac{\partial\delta^{\alpha}(\hat{x}_i^{\tau}, \hat{y}_i^{\tau}, \theta(\vec{x}_i^{\tau}))}{\partial x_i^{\tau}} \tag{17}$$

will always give a value in the range [0, 360), so the rate change relative to any distance that the x_i^{τ} coordinate is displaced will be defined for all input. The same cases can be extended to the remaining three variables of differentiation (y_i, x_{i+1}, y_{i+1}), thus making the function $\bar{Y}^{\alpha}(T)$ differentiable and therefore a suitable loss function.

Ultimately, this auxiliary loss function encourages trajectories to stay near lanes whose headings they align with, and to adjust their own headings to more closely match that of the nearest lane.

V. Experimental Analysis and Evaluations

A. Dataset

We use the public nuScenes dataset [6] to train and evaluate our model. We use from each sample a rasterized image of the surrounding map, vehicle state information (velocity, acceleration, heading), and target trajectory. The data was collected over inner-city drives conducted in Boston and Singapore. We train and evaluate our model using the official benchmark split for the nuScenes prediction challenge; in total, we used 29889 instances in the train set, 7905 instances in the validation set, and 8397 instances in the test set.

B. Network Architecture

We use the MTP network defined in [2], consisting of map intput to a base CNN, whose output is flattened and concatenated with agent vehicle state information (velocity, acceleration, heading change rate), followed by two fully-connected layers. For our experiments, we use a network output of 15 modes with 12 predicted points per mode (representing 6 seconds of travel). We use a base CNN of ResNet-50 [7]. In accordance with the expected input to ResNet with ImageNet dataset pretraining, we normalize our rasterized map images in RGB space prior to training. We use the classification and regression loss functions as defined in [2], with an additive term for lane heading auxiliary loss.

C. Implementation Details

With earlier described rasterized map physical dimensions of 50 meters x 50 meters, using a scale of 0.1 meters per pixel, we assume the lane and trajectory to be approximately straight (i.e. of single uniform heading) on the pixel scale. Each scene map contains information on lane placement and heading, drivable area, and surrounding vehicles and pedestrians. Vehicle state is provided as a three-dimensional input. We use a batch size of 16 and Adam optimizer [8], implemented using PyTorch [9].

D. Reducing Network Training Time & Memory Requirements with Secondary Maps

Calculating this loss per-sample can be computationally expensive. For every predicted mode of each sample instance, it is required to find the $L2$-nearest lane point to each midpoint on the predicted trajectory, with predictions changing on every iteration.

This computational hurdle can be lowered through preprocessing; for each instance map, which in our case extends 10 meters behind the vehicle, 40 meters ahead, 25 meters left, and 25 meters right, we generate a secondary orientation map, covering a larger area to account for trajectories which leave the original map. This secondary map extends 20 meters behind the vehicle, 80 meters ahead, 50 meters left, and 50

TABLE I: Results of comparative analysis on nuScenes dataset, over a prediction horizon of 6-seconds

	MinADE$_1$	$MindADE_5$	$MinADE_{10}$	$MinFDE_1$	$MinFDE_5$	$MinFDE_{10}$	$MissRate_{5,2}$	$MissRate_{10,2}$	Off-Road Rate	Off-Yaw Rate
Constant Velocity, Yaw	4.61	4.61	4.61	11.21	11.21	11.21	0.91	0.91	0.14	–
Physics Oracle	3.69	3.69	3.69	9.06	9.06	9.06	0.88	0.88	0.12	–
MTP	4.59	2.44	1.57	10.75	5.37	3.16	0.70	0.55	0.11	0.13
MTP with Off Road Loss	4.51	2.16	1.60	10.44	4.73	3.23	0.72	0.58	0.13	0.14
MTP with Lane Heading Loss	4.16	2.23	1.57	9.65	4.85	3.14	0.69	0.56	0.10	0.13

meters right. On this map, each pixel location is assigned a value which equals the orientation of the nearest lane point.

Prior to training, these secondary maps are generated and saved for each data sample. Each grid location on the map represents a heading from the continuous range [0, 360) degrees in the global frame. To represent each grid location as a 64-bit floating point value can quickly become storage intensive for a large set of 500x500 maps. However, only a coarse precision of the angle is required for this problem; we would never consider a driver to be going the 'wrong way' if their heading was off by just a few degrees. For this reason, a representation with precision only to the scale of degrees is appropriate for this problem. With this in mind, we can create a data-efficient representation which encodes each heading as an 8-bit grayscale integer pixel value in the range [1, 255], with the value of 0 reserved for map locations corresponding to intersections. Headings are mapped from range [0, 360) degree values to [1, 255] grayscale values as follows:

$$\theta_{map} = 1 + \lfloor \frac{254}{360}\theta \rceil. \tag{18}$$

Using the above function, we assign to each point on the secondary map the mapped value of the heading of the $L2$-nearest lane, illustrated in Fig. 4. During training, when inversely mapping from grayscale integer to degrees, there is a loss of precision that occurs, since the 360 degrees are mapped to 254 values. In this sense, each 'bin' of the data representation actually represents a span of approximately $1.417°$, a reasonable precision for this task.

E. Baselines

Results are shown in comparison to the following baselines:

- Constant Velocity, Yaw: The predicted trajectory is a continuation of the vehicle's current velocity and heading.
- Physics Oracle: As introduced in [10], the proposed trajectory is selected as the best trajectory from four dynamics models:
  - constant velocity and yaw,
  - constant velocity and yaw rate,
  - constant acceleration and yaw, and
  - constant acceleration and yaw rate.

  Note that this method is not used to make predictions, but rather provides a reference benchmark to four simple physical models, to illustrate improvement from models which account for more complex maneuvers.
- MTP: The predicted trajectories are the output of the original MTP model, without auxiliary loss, as described above.

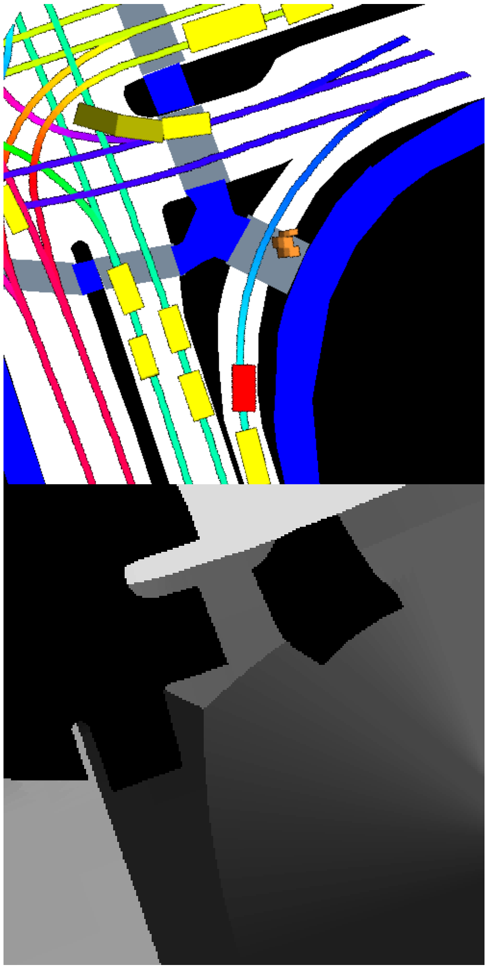
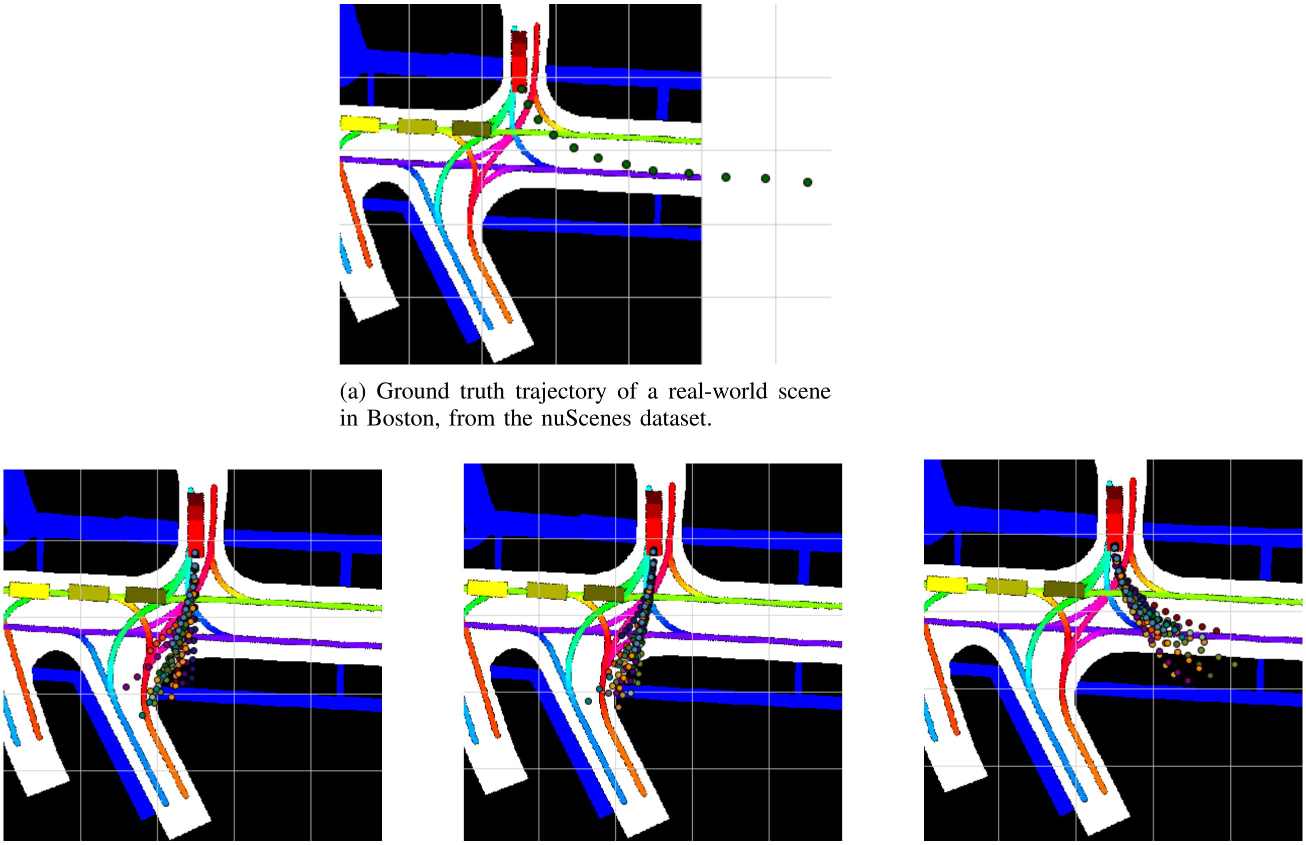


Fig. 4: Above: The rasterized bird's-eye-view RGB input map for a sample. Below: The secondary map for the same sample, where each pixel maps to the approximate heading of the nearest lane, or zero if in an intersection.

F. Metrics

Reported metrics include the following:

- Minimum average displacement error over k most probable trajectories (MinADE$_k$), for $k = 1, 5, 10$
- Minimum final displacement error over k most probable trajectories (MinFDE$_k$), for $k = 1, 5, 10$
- Miss rate at 2 meters over k most probable trajectories (Miss Rate$_{k,2}$), for $k = 5, 10$
- Off-road rate
- Off-yaw rate, the new metric as defined in this paper, measuring the proportion of predicted trajectories that include points which have positive angular difference



(a) Ground truth trajectory of a real-world scene in Boston, from the nuScenes dataset.

(b) Predicted trajectories by MTP with no auxiliary losses.

(c) Predicted trajectories by MTP with off-road loss.

(d) Predicted trajectories by MTP with YawLoss.

Fig. 5: This figure illustrates three different methods of trajectory prediction for a single instance from the nuScenes dataset. The ground truth trajectory is provided in (a). In (b), we see that MTP falsely predicts a straight pass through the intersection into the lane of oncoming traffic, with some deviation off the side of the road. In (c), after adding the off-road loss, we see that the off-road predictions are pushed closer to the drivable area (the closest available road segment), but this still places the vehicle into oncoming traffic. Only after applying the YawLoss (d) do the trajectories align with the proper lane heading.

(beyond a threshold) from the nearest lane yaw, averaged over all agents.

G. *Quantitative Results*

We compare our extension of the MTP model to the various baselines in Table I. Our model outperforms the baseline on 8 of the 10 reported metrics, while being second by just .001 m for minADEK for $k = 10$. In contrast, the model with off-road loss outperforms the baseline on just 5 of the 10 reported metrics, demonstrating the capability of model improvement via YawLoss.

A qualitative illustration comparing the effects of Off-Road Loss and YawLoss on an MTP base model is shown in Figure 5.

VI. Future Work

One apparent extension of this work would be the application of the lane heading auxiliary loss to existing deep learning models. State-of-the-art models which use social context, such as [11], have been shown to outperform MTP on the nuScenes dataset; their results may be further improved through the introduction of this auxiliary loss function, in tandem with other auxiliary losses such as off-road loss.

Additionally, the loss function itself can be further investigated and improved. Two possibilities include better selection of the angular difference threshold (either selection through statistical study of lane change maneuvers, or through varying the parameter during training based on agent dynamics), and development of a weighting scheme which penalizes deviations earlier in the trajectory as it may be harder to re-align later points in the trajectory when already off-course.

VII. Concluding Remarks

In this paper we presented an auxiliary loss function which may be used to augment the performance of existing models for vehicle trajectory prediction in urban environments. This lane heading loss function leverages the expectation that vehicles follow the direction ascribed to roadway lanes at all times, with exception for corrective maneuvers, lane changes, and intersection crossings. This loss function applies to all predicted modes, since no mode should predict driving opposite the lane direction. Experiments showed that extending the benchmark MTP model with the lane heading auxiliary loss outperforms the model's original classification and regression losses.

As stated by Daily et al. [12], "Self-driving and highly automated vehicles must navigate smoothly and avoid obstacles, while accurately understanding the highly complex semantic interpretation of scene and dynamic activities." While convolutional neural networks and other data-driven approaches may be effective at repeating known patterns, there is a lost element of explainability which is crucial towards public safety and adoption. By encoding familiar driving expectations through the introduced Off-Yaw Rate metric and YawLoss, we initiate a step towards autonomous vehicle computational models which can both learn and explain.

References

[1] D. Ridel, N. Deo, D. Wolf, and M. Trivedi, "Scene compliant trajectory forecast with agent-centric spatio-temporal grids," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2816–2823, 2020.

[2] H. Cui, V. Radosavljevic, F. Chou, T. Lin, T. Nguyen, T. Huang, J. Schneider, and N. Djuric, "Multimodal trajectory predictions for autonomous driving using deep convolutional networks," *International Conference on Robotics and Automation (ICRA)*, 2019.

[3] N. Deo and M. M. Trivedi, "Trajectory forecasts in unknown environments conditioned on grid-based plans," *arXiv preprint arXiv:2001.00735*, 2020.

[4] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation."

[5] M. Niedoba, H. Cui, K. Luo, D. Hegde, F.-C. Chou, and N. Djuric, "Improving movement prediction of traffic actors using off-road loss and bias mitigation," in *Workshop on'Machine Learning for Autonomous Driving'at Conference on Neural Information Processing Systems*, 2019.

[6] 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," *arXiv preprint arXiv:1903.11027*, 2019.

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: http://arxiv.org/abs/1412.6980

[9] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: a survey," *The Journal of Machine Learning Research*, vol. 18, no. 1, pp. 5595–5637, 2017.

[10] T. Phan-Minh, E. C. Grigore, F. A. Boulton, O. Beijbom, and E. M. Wolff, "Covernet: Multimodal behavior prediction using trajectory sets," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 074–14 083.

[11] K. Messaoud, N. Deo, M. M. Trivedi, and F. Nashashibi, "Trajectory prediction for autonomous driving based on multi-head attention with joint agent-map representation," 2020.

[12] M. Daily, S. Medasani, R. Behringer, and M. Trivedi, "Self-driving cars," *Computer*, vol. 50, no. 12, pp. 18–23, 2017.