Predicting Lane Change Decision Making with Compact Support

Hua Huang[1] and Adrian Barbu[2]

***Abstract*— In the foreseeable future, autonomous vehicles will have to drive alongside human drivers. In the absence of vehicle-to-vehicle communication, they will have to be able to predict the other road users' intentions. Moreover, they will also need to behave like a typical human driver so that other road users can infer their actions. It is critical to be able to learn a human driver's mental model and integrate it into the Planning & Control algorithm. In this paper, we present a robust method to predict lane changes as cooperative or adversarial. For that, we first introduce a method to extract and annotate lane changes as cooperative and adversarial based on the entire lane change trajectory. We then propose to train a specially designed neural network to predict the lane change label before the lane change has occurred and quantify the prediction uncertainty. The model will make lane change decisions following human drivers' driving habits and preferences, i.e., it will only change lanes when the surrounding traffic is considered to be appropriate for the majority of human drivers. It will also recognize unseen novel samples and output low prediction confidence correspondingly to alert the driver to take control in such cases. We published the lane change dataset and codes at `https://github.com/huanghua1668/lc_csnn`.**

I. INTRODUCTION

One of the biggest obstacles in deploying autonomous vehicles is the necessity for the autonomous vehicles to interact with human drivers, especially in scenarios that need cooperation, e.g., lane changes, unprotected left turns, roundabouts, unsignaled intersections, etc. In particular, lane changes are considered to be one of the most challenging maneuvers even for human drivers. Around 18% of all accidents happen during the execution of a lane change, and most of them are rear-end collision [1]. As can be seen in Fig. 1, ego has to consider both the relative position and relative velocity of the surrounding 3 vehicles. In particular, ego should understand the intentions of V_0, which could be either:

- Respect the cut-in request. If necessary, it will decelerate to create enough space.
- Ignore the request and might even accelerate to close the window to deter ego from merging.
- Keep its speed and wait for the ego's next action, i.e., a wait-and-see mode.

Without vehicle-to-vehicle communication, ego has to be able to recognise other road users' intentions. If ego does not recognize the environment and recklessly changes the lane, the lag vehicle in the target lane might have to do a harsh brake or be forced to change lane. In the worst scenario, a collision might happen with the lag vehicle. Equally importantly, ego has to behave like a vehicle driven by a typical human driver so that other road users can anticipate its actions.

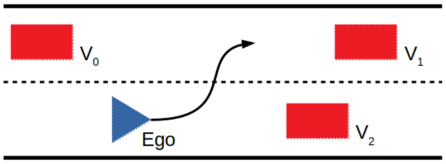

Fig. 1. Lane change illustration. V_2 is the immediate leading vehicle in the old lane, V_1 is the leading vehicle in the target lane, V_0 is the lag vehicle in the target lane, and Ego is the autonomous vehicle carrying out the lane change.

To accelerate the integration of autonomous vehicles with human driven vehicles, the human driver's mental model must be learned and incorporated in the Planning & Control algorithm. Encouraged by the successful applications of deep learning in object classification, language translation, game playing and many other tasks [2], recently there has been a surge of interest in applying deep learning to predict the lane change decisions [3, 4, 5, 6].

Even though deep learning is a very powerful tool, it can be fooled easily by adversarial samples [7, 8]. In many cases, neural networks are overconfident in their predictions [9]. The ReLU based neural networks have been proved to produce almost always high confidence predictions far away from the training data [10]. If one wants to apply neural networks to safety-critical domains like autonomous driving, the model has to be able to assess its prediction confidence and detect OOD samples [11], i.e., know when it doesn't know.

The contributions of this paper are:

- We propose an annotation method that can extract human drivers' preferences and habits in lane changes.
- We investigate reliable neural networks for lane change decision making. We demonstrated that the proposed networks can assess their prediction uncertainty and detect when the scenario is out-of-distribution (OOD) to alert a human operator. The obtained models also achieved similar test accuracy for in-distribution samples compared with normal neural networks and greatly outperform them for OOD samples detection.

II. RELATED WORK

A. *Predicting Lane Changes with Deep Learning*

Xie et al. [3] employed deep belief networks to model the lane change decision making and long-short-term-memory (LSTM) networks to model lane change implementation.

[1] Hua Huang is with Department of Mathematics, Florida State University, Tallahassee, FL 32306 USA. `hhuang@math.fsu.edu`

[2] Adrian Barbu is with the Department of Statistics, Florida State University, Tallahassee, FL 32306 USA. `abarbu@stat.fsu.edu`

Zhang et al. [4] proposed to use LSTM to model both the lane change and lane following behaviors. Attention mechanisms have also been introduced to improve prediction accuracy [6]. Jeong et al. [5] trained an end-to-end deep convolutional neural network from images directly to classify whether it's safe to initialize the lane change. Yan et al. [12] built a neural-network based payoff model to describe the interactions with other road users.

Deep learning has also been applied to reinforcement learning to learn both the lane change decision making and implementation [13, 14, 15, 16, 17]. The major disadvantage of reinforcement learning is that a simulation bed has to be built, in which behaviors of agents should be as close as possible to human drivers' behaviors. Compared with learning a good driving policy, characterizing high-fidelity agent driving behaviors is equally difficult. Another challenge is that the rewards need to be hand-crafted to be able to train a smooth and natural policy.

B. OOD Detection

There are primarily four types of OOD detection techniques. Deep ensembles [18] have been proven to work well in high dimensional space as individual networks tend to disagree on OOD samples and eventually lead to a higher prediction entropy. The second approach is to modify the training process by incorporating OOD samples and minimize a hybrid loss function to penalize the high confidence prediction on OOD samples [10, 19, 20, 21]. The major disadvantage is that the space of OOD samples will be too large to cover. The model trained on one set of OOD samples might not be able to detect another unseen set of OOD samples. The third type is modifying the score function. Temperature scaling has been introduced into the softmax score to enlarge the difference between in-distribution and OOD samples [19]. Energy scores [22] have also been found to better separate the in-distribution samples from OOD samples compared with softmax scores. Lastly, compact support networks have also been introduced through variations of Radial-Basis-Function networks [23, 24].

The Compact Support Neural Network (CSNN) [25] smoothly interpolate between a ReLU-type network and a traditional RBF network through a shape hyperparameter. It has the same-level accuracy in predicting in-distribution samples compared with normal neural network. It will have zero output for samples outside the support, i.e., OOD samples. In this paper, CSNN will be adopted to predict human drivers' lane change decision makings and detect unseen OOD samples.

In this paper we will work with the NGSIM dataset [26], which is described in Section V-B. It contains trajectory data from the I-80 and the US-101 highways.

To capture the likely actions of human drivers in lane change and the likely reactions of the lag vehicle in the target lane, cooperative and adversarial lane changes need to be defined and extracted from this data. Since human drivers' intentions cannot be observed directly, there are various methods in the literature to label cooperative/adversarial lane change behaviors. The positive samples (cooperative) are relatively easy to characterize, however, the negative (adversarial) samples are much more difficult to define. The fact that a human driver does not begin the lane change could be caused by many factors, e.g., competing behavior of the lag vehicle in target lane, or the driver does not have an intention to carry out the lane change and prefers car-following for now. The implicit reasoning can not be observed. For these reasons, different papers adopt different ways to label the negative samples:

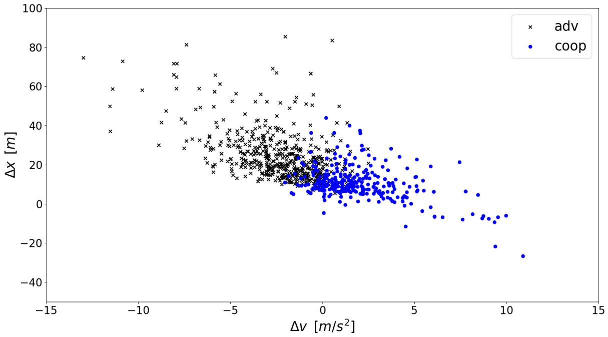

Fig. 2. Lane changes for the I-80 dataset set with labels based on the change in Δx, as proposed by [12].

- Negative labels for the lane change preparation stage. Lane changes usually experience a preparation process for seeking a suitable acceptable gap or adjusting the velocity before lane-changing execution. Thus, Scheel et al. [1] labeled the lane-changing preparation stage as a number of lane-changing execution rejecting events.
- Negative labels for observations before and after the ego starts the lateral move [27]. Car following maneuvers are also counted as negative samples. Under these conditions less than 0.1% samples are positive.
- Negative labels for decreases in relative longitudinal distance after lane change. The cooperative/adversarial strategy is labeled based on whether the relative distance between ego and lag vehicle $\Delta x = x_{ego} - x_0$ in the target lane decreases or not from time $t = -3\ s$ to $t = 0\ s$, in which $t = 0\ s$ is the time the ego crosses the lane divider. If the relative distance increases, it is labeled as cooperative [12].

The extracted negative samples in both the first and second methods do not always fall in the adversarial category. For the third approach, the extracted lane changes from the I-80 dataset are labeled and plotted in Fig. 2. The plot shows Δx vs Δv at time $t = -3\ s$ for each trajectory. As one could see from Fig. 2, when the ego is slower than the lag vehicle in the target lane, i.e., $\Delta v = v_{ego} - v_0 < 0$, Δx will almost always decrease and the sample will be labeled as negative. This labeling will lead to overly conservative lane change decisions and cannot be used in moderate or heavy traffic.

III. LANE CHANGE EXTRACTION AND ANNOTATION

Since motorcycle and truck change lanes differently from cars, only lane changes carried out by cars are included in this research. Lane changes from/to the rightmost lane are also excluded as the rightmost lane is for ramp merging/diverging, in which vehicles have to finish the merging/diverging before the merge/diverge point, so drivers tend to behave differently than a typical lane change.

Two types of lane changes are defined and extracted in this research. Successful lane change is defined as merging in front of the lag vehicle in the target lane V_0. The positions and velocities of ego and surrounding vehicles are extracted in $t \in [t_0,\ 5s]$, in which $t_0 < 0$ is defined as the time when the ego starts to have a lateral velocity $|v_y| > 0.213\, m/s^2$ [1] and without oscillation thereafter. The time $t = 0s$ corresponds to when the ego crosses the lane divider. The other type is aborting the current open window and merging after the lag vehicle V_0 in the target lane. When it is deemed too aggressive or even dangerous to carry out lane change immediately, ego will prefer to wait for the next available window. In this scenario, we first find successful lane changes, and further require that at $t = -8s$, ego is in front of V_1.

For successful lane changes, instead of Δx, the deceleration of the lag vehicle in the target lane is inspected. If ego's lane change causes no forced harsh brake (a harsh brake is defined as a deceleration smaller than the comfortable deceleration [28] $a_{comfortable} = -3\ m/s^2$ as recommended by the Institute of Transportation Engineers) for vehicle V_0, it will be labeled as a cooperative lane change. Since the acceleration is calculated by the second order derivative of the position and therefore can be noisy, we require the total duration of deceleration $a_t < -3\, m/s^2$ in $t \in [t_0,\ 5s]$ for V_0 be less then $1s$ for cooperative samples, otherwise, it will be labeled as adversarial. Window abortion is also labeled as adversarial.

Overall, 1,558 lane changes are extracted from the I-80 dataset and 1,290 lane changes from the US-101 dataset. The statistics of the extracted lane change are summarized in Table I. The extracted lane changes from the I-80 dataset are plotted in Fig. 3, in which the samples are scattered in the $\Delta x - \Delta v$ 2d space. The most significant observation is that there is class overlap, i.e., data uncertainty. For one particular relative velocity, timid or polite drivers will choose to give up the current lane change window, while aggressive or impatient drivers will perform the lane change. Understandably, models should output low confidence in the ambiguous region, meanwhile, they are also required to have low confidence in OOD samples.

TABLE I

STATISTICS OF I-80 AND US-101 DATASETS

	I-80	US-101
Location	Emeryville	Los Angeles
Time	4pm-4:15pm, 5pm-5:30pm	7:50am-8:35am
Samples	1,558	1,290
Merge in front coop	1,095 (70.28%)	1,116 (86.51%)
Merge in front adv	150 (9.63%)	32 (2.48%)
Merge after	313 (20.09%)	142 (11.01%)

IV. PROPOSED METHOD

The neuron with compact support [25] is defined as

$$\begin{aligned} f(\mathbf{x}) &= \max(R^2 - ||\mathbf{x} - \mu||^2, 0) \\ &= \max(\alpha(R^2 - \mathbf{x}^T\mathbf{x} - \mu^T\mu) + 2\mu^T\mathbf{x}, 0) \end{aligned} \tag{1}$$

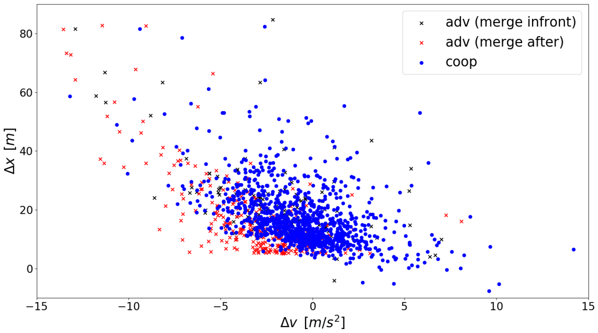

Fig. 3. Lane changes extracted and labeled with the proposed approach for the I-80 dataset.

When the shape parameter $\alpha = 0$, it will be a standard ReLU neuron. When $\alpha > 0$, it will be a Compact Support Neuron. In practice, CSNN is trained by starting from a regular neuron network ($\alpha = 0$) and then gradually increasing α to 1. It can be shown that the neuron only has support within a sphere of radius

$$R_\alpha^2 = R^2 + ||\mu||^2(\frac{1}{\alpha^2} - 1) \tag{2}$$

and center

$$c = \frac{\mu}{\alpha} \tag{3}$$

R and μ are learnable parameters. To further constrain the support, radius penalties are added to the loss function. Experiments show that the infinity norm works best. For this binary classification task, the overall loss is binary cross entropy loss and radius penalty.

$$\ell = \sum_i y_i \log(p_i) + (1 - y_i)\log(1 - p_i) + \lambda ||R||_\infty \tag{4}$$

where λ is the radius penalty coefficient. It's worth noting that the ability to measure the distance from a testing sample to training dataset is a necessary condition to get a high-quality estimation of distribution uncertainty [24]. The neuron output in the CSNN is determined by the distance of the inputs to the neuron's parameter vector, hence satisfies the necessary condition.

V. EXPERIMENTS

A. Synthetic dataset

To show the effectiveness of the CSNN in detecting OOD samples, CSNN models with α of 0 and 1 are trained on the moons dataset. The moons dataset contains two interleaving half circles corrupted by noise, one for each class, as illustrated in Figure 4. We generated 1,500 samples using the scikit-learn library [29]. Another 10,000 samples are generated on a uniform grid spanning $[-2.5,\ 3.5] \times [-3,\ 2]$. The samples are normalized to have 0 mean and standard deviation $1/\sqrt{d}$, in which d is the feature dimension, i.e., 2 here. A two layer CSNN model with 256 compact support neurons in the 1st layer is implemented. The output layer is a fully connected layer. The radius penalty coefficient is $\lambda = 0.64$.

The samples and the prediction confidence are plotted in Fig. 4. With $\alpha = 0$, i.e., a ReLU-type neuron, the model will

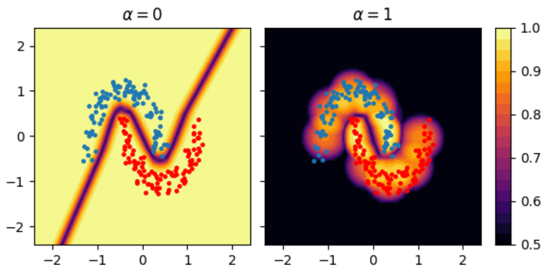

Fig. 4. Confidence map and data points for the moons dataset. Left: regular neural network, right: CSNN.

generalize the prediction far from the training dataset and output high confidence for OOD samples. With $\alpha = 1$, the confidence is 0.5 away from the two circles, while around the two circles, the confidence is near 1. Just as designed, CSNN will shrink its support to a neighbouring domain surrounding the training samples and will output a low confidence for samples far from the training dataset. The confidence is also low for samples between the two moons, i.e., in addition to distribution uncertainty, model successfully identifies the class overlap and data uncertainties [30].

B. NGSIM dataset

We will use the FHWA's Next Generation Simulation (NGSIM) dataset [26] for real data experiments. The NGSIM dataset has been widely used to investigate human driving behaviors. The dataset contains videos of the northbound traffic on I-80 and southbound traffic on US-101. The detailed location and time are given in Table I. The study site is approximately 500m long for I-80 and 640m for US-101. The vehicle positions were recorded every 0.1s. The dataset contains 11,779 vehicle trajectories, out of which 11,328 trajectories are carried out by cars.

1) Classifiers: Multi-Layer Perceptron (MLP) networks are trained on the NGSIM dataset will be used as baseline models for in-distribution prediction performance. CSNN models will then be trained and the test accuracy will be compared with the MLP results. The Area under the ROC curve (AUROC) score for classifying between in-distribution and OOD samples will be used to gauge the OOD detection performance. The in-distribution prediction accuracy and OOD detection performance will also be compared with results obtained from recently proposed OOD detection algorithms.

2) Features: An instance at time t is represented by the following features

$$\mathbf{x} = [v_{ego},\ \Delta v_0,\ \Delta x_0,\ \Delta y_0,\ \Delta v_1,\ \Delta x_1,\ \Delta y_1,\ \Delta v_2,\ \Delta x_2,\ \Delta y_2] \quad (5)$$

containing the relative velocity $\Delta v_i = v_{ego} - v_i$, relative longitudinal position $\Delta x_i = x_{ego} - x_i$ and relative lateral position $\Delta y_i = y_{ego} - y_i$ for $i \in \{0,\ 1,\ 2\}$. The features are extracted every 0.1 s from $t_0 - 0.5$ to t_0 and averaged to get the final features.

To facilitate the downstream OOD evaluation task, a backward feature selection is carried out using MLP models with two hidden layers with 64 neurons. Using just 4 features out of the 10 original features (5),

$$\mathbf{x} = [\Delta v_0,\ \Delta x_0,\ \Delta v_1,\ \Delta x_1] \quad (6)$$

we found the best average test accuracy over 10 independent runs, dropping from 0.876 for the 10 features to 0.871. It is reasonable to conclude only these 4 features are strongly related to this prediction task, hence hereafter, this 4D feature space will be used.

3) OOD sample generation: OOD samples are generated through uniform sampling.

First, all the in-distribution samples, i.e., samples from both datasets (I-80 and US-101), are normalized to have 0 mean and 1 std in each dimension.

Then for each in-distribution sample $\mathbf{x}_i$, the minimum distance d_i to other in-distribution samples is computed. Then we find a distance threshold τ as the 99 percentile of the d_i values, thus 99% of the d_i will be less than τ.

Then we generate 160,000 samples through uniform sampling in the hyper-rectangle

$$\begin{aligned}\mathscr{R} = &[1.5 \min_i \Delta v_0^i, 1.5 \max_i \Delta v_0^i] \times [-r_0, 0.5 r_0] \\ &\times [1.5 \min_i \Delta v_1^i, 1.5 \max_i \Delta v_1^i] \times [-0.5 r_0, r_0]\end{aligned} \quad (7)$$

where $r_0 = 100m$ is the detection range.

The generated samples are then transformed using the mean and std of the in-distribution samples and the minimum distance to any in-distribution samples is calculated. The generated samples with a distance $d_i < \tau$ are discarded. This way, 147,496 generated samples are kept as OOD samples. In comparison, there are 2,848 in-distribution samples.

4) Architectures: For the baseline, MLP models with 2 hidden layers of 64 hidden neurons are trained. To have a fair comparison, the CSNN models have the same number of layers and neurons except that the neurons in the last hidden layer are compact support neurons. There is also a batch normalization layer without learnable parameters after the first hidden layer.

5) Training: Samples from the I-80 and US-101 datasets are combined as the in-distribution samples, in which 75% samples are used for training and the remaining samples for test. Adam optimization with learning rate 0.0001 is employed. 10 independent runs are carried out for each algorithm with different random initializations and shuffles of the training data. The shape parameter α is increased linearly from 0 to 1 as the epoch number increases to 1000, i.e., we begin with normal type ReLU neurons and gradually shrink the support. The parameter R is initialized to 1 and is learnable. The radius penalty coefficient is set by grid search over the range $\lambda \in [0,\ 2]$ and the best test accuracy is obtained at $\lambda = 0.1$.

6) Methods compared: The most similar OOD detection approach to ours is the DUQ [23] algorithm. DUQ computes a feature vector through MLP and then calculate the distance between the feature vector with class centroids. The class centroid is updated with an exponential moving average of the feature vectors belonging to that class. When the distance between the feature vector and any class centroid is large, it is considered to be a OOD sample. To have a fair comparison, a MLP with 1 hidden layer of 64 units is used to extract the feature vector and the centroid is of size 64. The length scale

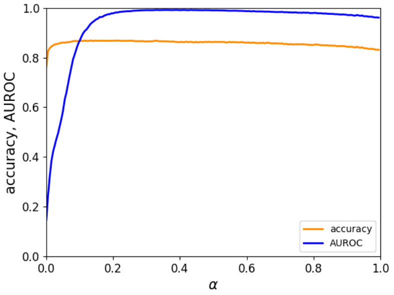

Fig. 5. Test accuracy and AUROC

σ and gradient penalty coefficient λ are set by grid search over the space $\sigma \in (0, 1.0]$ and $\lambda \in [0, 1.0]$.

Another method we compared with is the deep ensemble. Models with the same architecture but different random initializations have been experimentally demonstrated to tend to disagree on OOD samples, hence leading to a higher entropy in prediction. In deep ensemble, the average prediction is defined as

$$\hat{p}(y|x) = \frac{1}{N}\sum_{i=1}^{N} p_{\theta_i}(y|x) \tag{8}$$

in which N is number of models and i_{th} model is parameterized by θ_i. The entropy of the prediction is defined as

$$H(\hat{p}(y|x)) = -\sum_{i=0}^{C} \hat{p}(y_i|x) log \hat{p}(y_i|x) \tag{9}$$

An ensemble was constructed from 10 independent nets and each net is a MLP network with two hidden layers of 64 units.

7) Results: The results are presented in Table II. For the baseline, the best test accuracy obtained by averaging 10 independent runs is 0.871 ± 0.002. For the CSNN algorithm, the average test accuracy and AUROC over 10 independent runs are plotted in Fig. 5 as functions of α.

TABLE II

TEST ACCURACY AND AUROC IN OOD DETECTION

	Test accuracy	AUROC
MLP	0.871 (.002)	0.156 (.012)
CSNN	0.868 (.002)	0.991 (.001)
DUQ	0.868 (.002)	0.971 (.005)
Deep ensemble	0.873 (.001)	0.189 (.005)

As we can see in Fig. 5, the test accuracy increases as α increases. The AUROC also increases as α increases and the support becomes more compact. The best AUROC 0.991 ± 0.001 is obtained at $\alpha = 0.33$, where test accuracy is 0.868 ± 0.002 . Compared with the baseline results 0.871 ± 0.002 obtained with normal neuron-based network, there is only a 0.3% decrease, i.e., the in-distribution prediction performance is comparable to a typical neuron-based network. When the α keeps increasing, both the test accuracy and AUROC decrease gently.

The best average test accuracy for DUQ 0.868 ± 0.002 over 10 independent runs is obtained at $\sigma = 0.4$ and $\lambda = 0.3$, where the AUROC in detecting OOD is 0.971 ± 0.005. As shown in Table II, CSNN obtained the same in-distribution prediction accuracy compared with DUQ but beat DUQ in detecting OOD samples for this task. CSNN has more flexibility and fewer assumptions as each neuron has its own support, while in DUQ, each class is assumed to have its own centroid, hence stronger assumptions and less flexibility.

The best average test accuracy of the deep ensemble is 0.873 ± 0.001, where the AUROC is 0.189 ± 0.005. The deep ensemble completely failed in this low-dimension OOD prediction task. Van Amersfoort et al. [23] also reported that deep ensembles do not work in low dimensional applications for the OOD detection task.

To reveal the reasons, the entropy of average prediction in another low-dimension ODD detection task, i.e. the moons dataset, is plotted in Fig. 6. In contrast to the high-dimension applications like image classification, in the low-dimensional scenario, the nets tend to only disagree with each other on the class overlap and near the decision boundary, i.e., data uncertainty, instead of the distributional uncertainty.

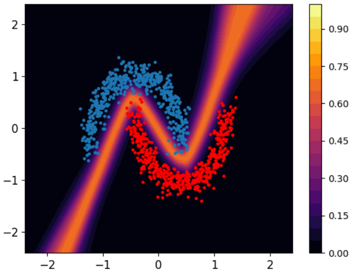

Fig. 6. Entropy of average prediction

The generalization capability of CSNN algorithm is investigated by training on dataset I-80 and testing on dataset US-101 and vice-versa. We first trained MLP networks and set the test accuracy as the baseline. CSNN algorithms are then trained and the test accuracy and AUROC are given in Table III. When CSNN is trained on dataset I-80 and tested on dataset US-101, the test accuracy decreases negligibly and the AUROC in detecting OOD samples is still high. There is a noticeable downgrade in OOD detection when CSNN is trained on dataset US-101 and tested on dataset I-80, i.e., some samples from dataset I-80 are recognized as OOD samples. We can conclude that there is a more diverse behavior in dataset I-80 than in dataset US-101.

TABLE III

TEST ACCURACY AND AUROC IN OOD DETECTION

	Train I-80, test US-101		Train US-101, test I-80	
	Test accuracy	AUROC	Test accuracy	AUROC
MLP	0.925 (.003)	0.158 (.020)	0.8 (.004)	0.102 (.007)
CSNN	0.923 (.003)	0.993 (.002)	0.797 (.003)	0.974 (.003)
DUQ	0.923 (.001)	0.982 (.005)	0.809 (.003)	0.972 (.007)
Deep ensemble	0.924 (.001)	0.177 (.009)	0.803 (.002)	0.145 (.006)

VI. CONCLUSIONS AND FUTURE WORK

In this paper we proposed a method for predicting human drivers' lane change decisions using Compact Support Neural Networks. We first extracted lane changes from a naturalistic driving dataset and labeled them based on the reactions of the lag vehicle in the target lane and the window preferences in lane change. We then trained CSNN models to predict the lane change behaviors and experimentally demonstrated that the trained models have comparable in-distribution prediction accuracy compared with normal neuron-based networks. The model achieved an AUROC of 0.991 in detecting OOD samples. We also compared the in-distribution prediction accuracy and OOD detection performance with recently developed OOD methods. The trained model can be integrated to the planning & control module of an autonomous vehicle and the vehicle will mimic human driving behavior, i.e., only carry out lane change when most human drivers consider it is appropriate. The model can also separate the unseen novel samples from the training dataset and alleviate over-generalization.

In the future, we will try to separate the distribution uncertainty from the data uncertainty. Currently the model will output low confidence when the sample is far from training dataset or when there is class overlap. Uncertainty from class overlap is arguably less risky compared with distribution uncertainty as human drivers will do both of them, while for distribution uncertainty, we simply do not know what might happen.

References

[1] O. Scheel et al. "Situation assessment for planning lane changes: Combining recurrent models and prediction". In: *ICRA*. 2018, pp. 2082–2088.

[2] Y. LeCun, Y. Bengio, and G. Hinton. "Deep learning". In: *Nature* 521.7553 (2015), pp. 436–444.

[3] D.-F. Xie et al. "A data-driven lane-changing model based on deep learning". In: *Transp. res. C: emerging technologies* 106 (2019), pp. 41–60.

[4] X. Zhang et al. "Simultaneous modeling of car-following and lane-changing behaviors using deep learning". In: *Transpp. res. C: emerging technologies* 104 (2019), pp. 287–304.

[5] S.-G. Jeong et al. "End-to-end learning of image based lane-change decision". In: *IV*. 2017, pp. 1602–1607.

[6] O. Scheel et al. "Attention-based lane change prediction". In: *ICRA*. 2019, pp. 8655–8661.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. "Explaining and harnessing adversarial examples". In: *arXiv preprint arXiv:1412.6572* (2014).

[8] A. Nguyen, J. Yosinski, and J. Clune. "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images". In: *CVPR*. 2015, pp. 427–436.

[9] C. Guo et al. "On calibration of modern neural networks". In: *ICML*. 2017, pp. 1321–1330.

[10] M. Hein, M. Andriushchenko, and J. Bitterwolf. "Why ReLU networks yield high-confidence predictions far away from the training data and how to mitigate the problem". In: *CVPR*. 2019, pp. 41–50.

[11] D. Amodei et al. "Concrete problems in AI safety". In: *arXiv preprint arXiv:1606.06565* (2016).

[12] Z. Yan, J. Wang, and Y. Zhang. "A game-theoretical approach to driving decision making in highway scenarios". In: *IV*. 2018, pp. 1221–1226.

[13] C.-J. Hoel, K. Wolff, and L. Laine. "Automated speed and lane change decision making using deep reinforcement learning". In: *ITSC*. 2018, pp. 2148–2155.

[14] Y. Chen et al. "Attention-based Hierarchical Deep Reinforcement Learning for Lane Change Behaviors in Autonomous Driving". In: *IROS*. 2019.

[15] A. Alizadeh et al. "Automated Lane Change Decision Making using Deep Reinforcement Learning in Dynamic and Uncertain Highway Environment". In: *ITSC*. 2019, pp. 1399–1404.

[16] T. Shi et al. "Driving decision and control for automated lane change behavior based on deep reinforcement learning". In: *ITSC*. 2019, pp. 2895–2900.

[17] B. Mirchevska et al. "High-level decision making for safe and reasonable autonomous lane changing using reinforcement learning". In: *ITSC*. 2018.

[18] B. Lakshminarayanan, A. Pritzel, and C. Blundell. "Simple and scalable predictive uncertainty estimation using deep ensembles". In: *NeurIPS*. 2017.

[19] S. Liang, Y. Li, and R. Srikant. "Enhancing the reliability of out-of-distribution image detection in neural networks". In: *ICLR*. 2018.

[20] K. Lee et al. "Training Confidence-Calibrated Classifiers for Detecting Out-of-Distribution Samples". In: *ICLR*. 2018.

[21] J. Ren et al. "Likelihood ratios for out-of-distribution detection". In: *NeurIPS*. 2019, pp. 14680–14691.

[22] W. Liu et al. "Energy-based Out-of-distribution Detection". In: *NeurIPS* 33 (2020).

[23] J. Van Amersfoort et al. "Uncertainty estimation using a single deep deterministic neural network". In: *ICML*. 2020, pp. 9690–9700.

[24] J. Liu et al. "Simple and principled uncertainty estimation with deterministic deep learning via distance awareness". In: *NeurIPS* 33 (2020).

[25] A. Barbu and H. Mou. "The Compact Support Neural Network". In: *arXiv preprint arXiv:2104.00269* (2021).

[26] V. G. Kovvali, V. Alexiadis, and L. Zhang PE. "Video-based vehicle trajectory data collection". In: *Transp. Res. Board Annual Meeting*. 2007.

[27] E. Balal, R. L. Cheu, and T. Sarkodie-Gyan. "A binary decision model for discretionary lane changing move based on fuzzy inference system". In: *Transp. Res. C: Emerging Technologies* 67 (2016), pp. 47–61.

[28] B. Wolshon and A. Pande. *Traffic engineering handbook*. John Wiley & Sons, 2016.

[29] F. Pedregosa et al. "Scikit-learn: Machine learning in Python". In: *JMLR* 12 (2011), pp. 2825–2830.

[30] A. Malinin and M. Gales. "Predictive uncertainty estimation via prior networks". In: *NeurIPS*. 2018.