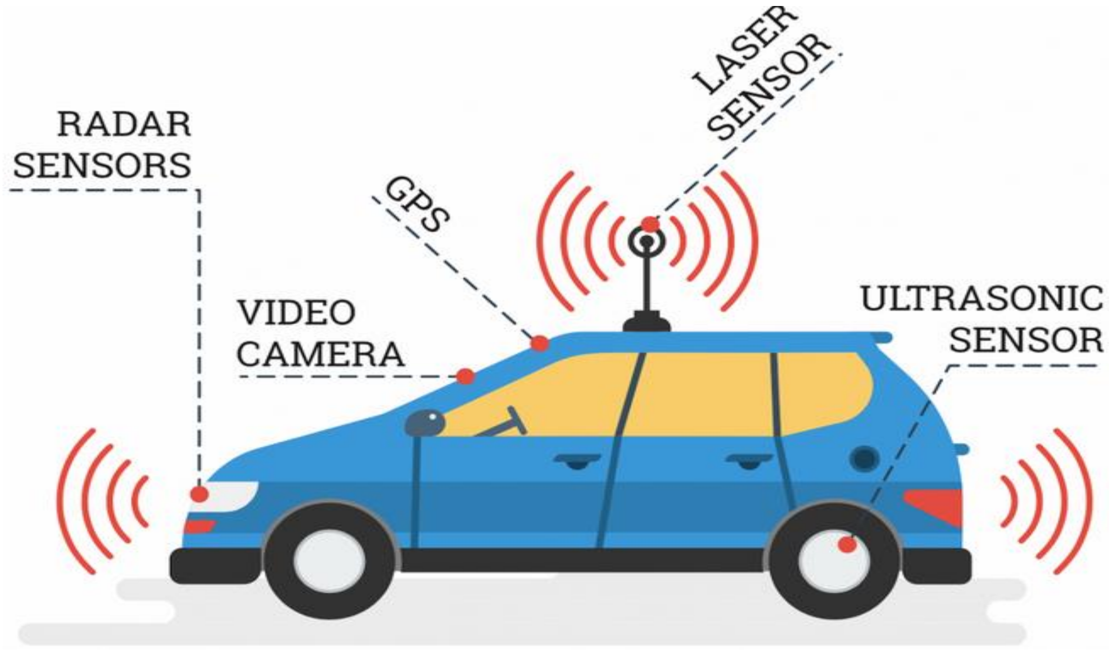
SECON 2021

A Reinforcement Learning based Decision-making System with Aggressive Driving Behavior Consideration for Autonomous Vehicles

Liuwang Kang and Haiying Shen
Department of Computer Science, University of Virginia

Background

- An autonomous vehicle (AV) measures its driving environment through sensors and cameras
- An AV makes real-time control decisions based on sensor measurements to ensure its driving safety

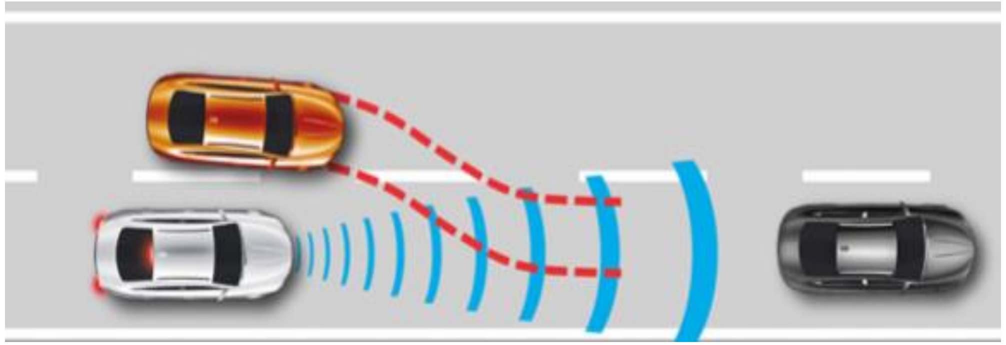


Sensors and cameras

Real-time control decision making

Background

- An AV and its surrounding vehicles will drive on the same road section to form a mixed driving environment
- One or more surrounding vehicles may conduct aggressive driving behaviors



Making control decisions with aggressive driving behavior consideration becomes necessary for AVs

Related Work

- Some methods [SECON'15, TMC'16, TR'19] try to detect aggressive driving behaviors by extracting their acceleration and orientation features

➢ Have large false detection alerts for situations where multiple surrounding vehicles conduct aggressive driving behaviors

- Some methods [TCDS'19, CDC'18, AR'17] try to make control decisions for a vehicle by considering driving behaviors of its surrounding vehicles

➢ Surrounding vehicles have time-varying driving behaviors and may conduct aggressive driving behaviors

Challenges

Propose a reinforcement learning based decision-making system (ReDS) to make control decisions for an AV by considering aggressive driving behaviors of surrounding vehicles

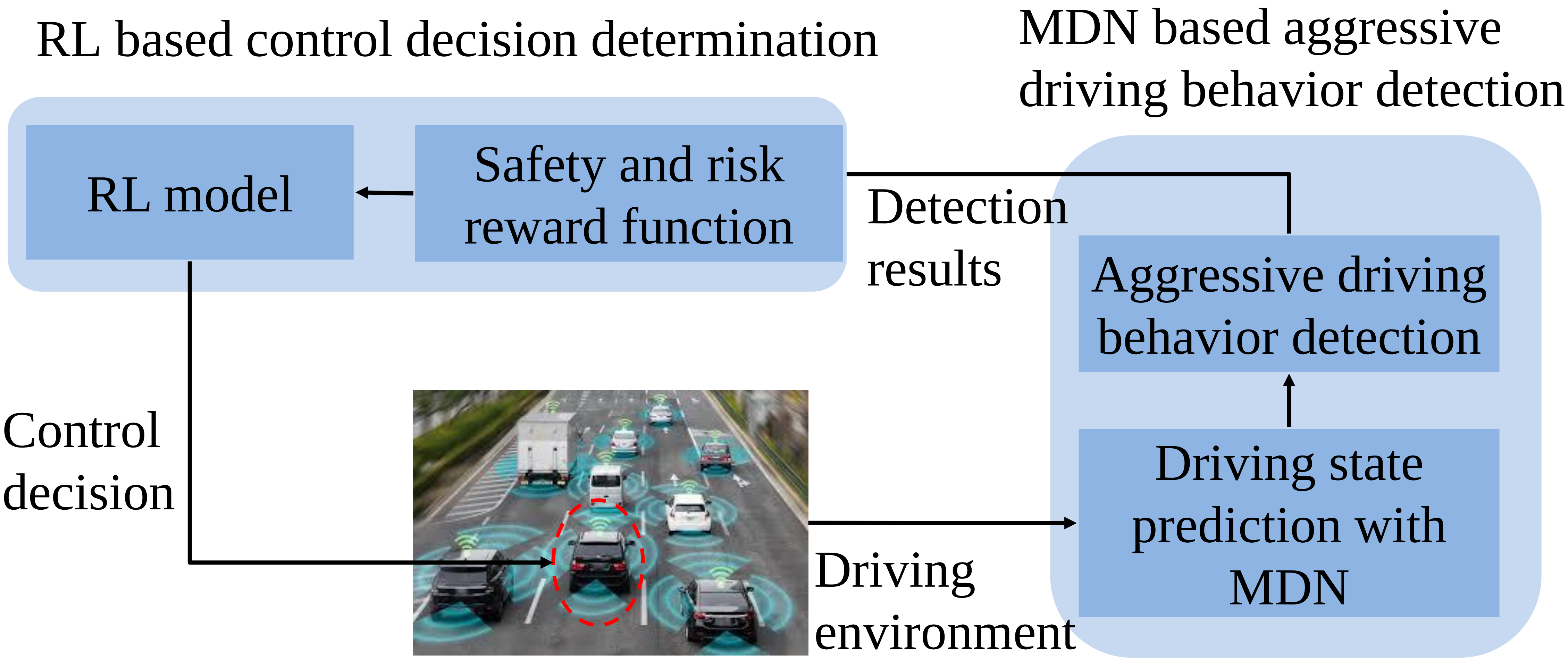
Challenge 1: How to detect multiple aggressive driving behaviors among surrounding vehicles accurately?

- Driving behaviors of multiple vehicles affect each other

Challenge 2: How to make control decisions with considering aggressive driving behaviors of surrounding vehicles?

- Driving behaviors are time-varying in one time period

Reinforcement Learning based Decision-making System



MDN: mixed density network

Challenge 1

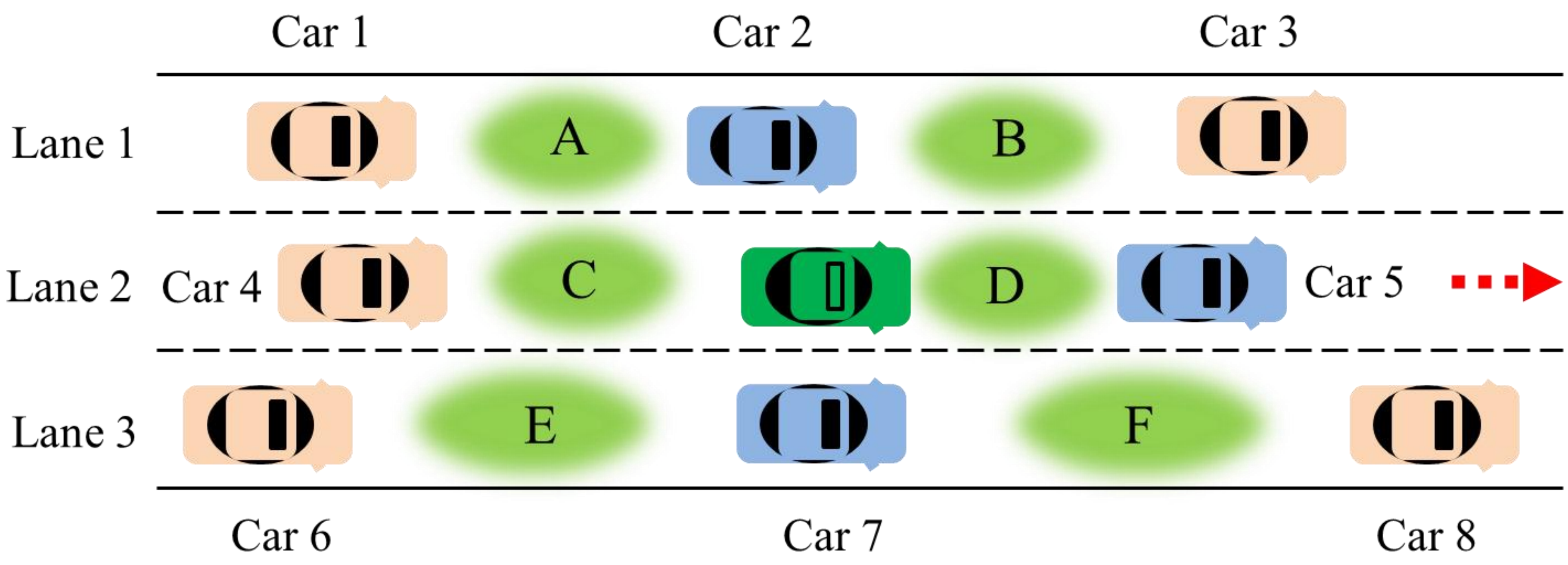
How to detect multiple aggressive driving behaviors among surrounding vehicles accurately?

MDN based Aggressive Driving Behavior Detection

Driving behavior prediction with MDN

Predict driving behavior of a vehicle based on its own vehicle states and its surrounding vehicle states

- Possible driving areas where a vehicle will drive



MDN based Aggressive Driving Behavior Detection

Driving behavior prediction with MDN

A combination of a mixture density model and a neural network to estimate the underlying distribution of the data

Inverse-to-Collision (ITC) quantifies the probability of existing a collision between two vehicles and is calculated as ${|\Delta v|}/{|\Delta d|}$

- Inputs of MDN $\boldsymbol{I}_t = [\boldsymbol{I}_t^G; \boldsymbol{I}_t^R; \boldsymbol{I}_t^O]$ in one time period Δ

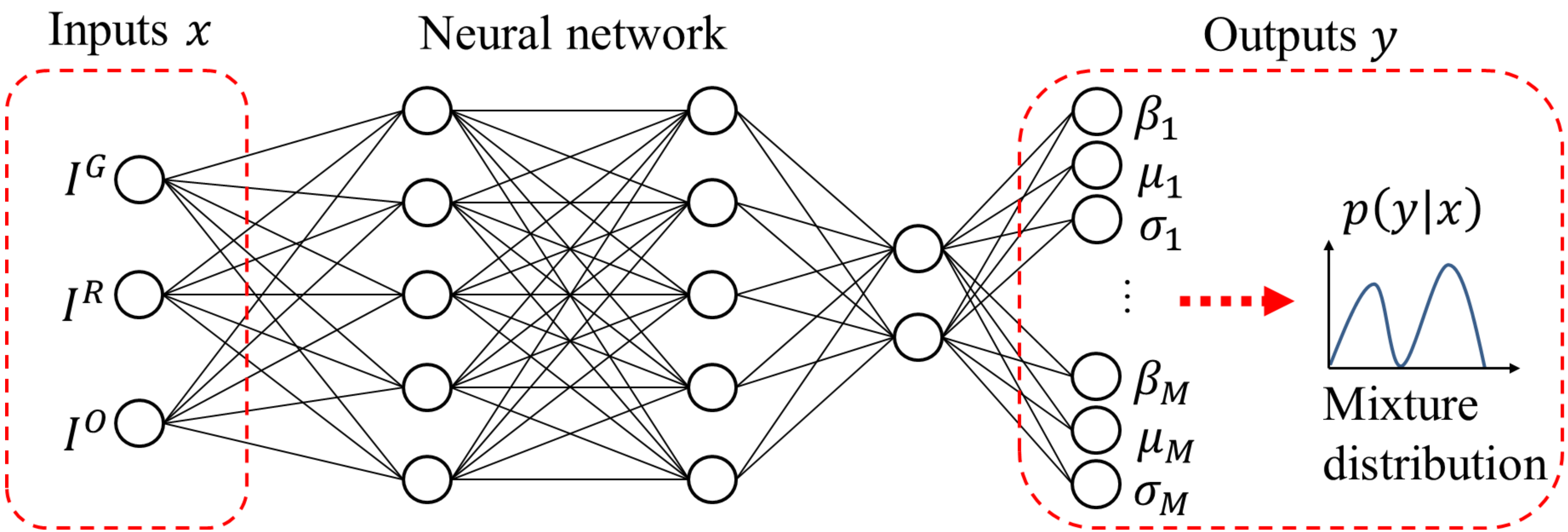
$I_{t-\Delta}$	$I_{t-\Delta+1}$	$I_{t-\Delta+2}$	$\cdots$	$\cdots$	I_t

- Predicted vehicle information $I_t^G = [v_t^G; d_t^G]$
- Reference vehicle information $I_t^R = [v_t^R; d_t^R; ITC_t^R]$
- Other vehicle information $I_t^O = [v_t^O; d_t^O; ITC_t^O]$

MDN based Aggressive Driving Behavior Detection

Driving behavior prediction with MDN

Output the driving area with the maximum value of probability $p(y|x)$ to indicate the driving behavior prediction result



MDN based Aggressive Driving Behavior Detection

Aggressive driving behavior detection

Step 1: Introduce a driving behavior error vector $\Delta \boldsymbol{d}_t$ to describe vehicle driving behavior differences from t-Γ to t

$\Delta \mathrm{d}_{t-\Gamma}$	$\Delta d_{t-\Gamma+1}$	$\Delta d_{t-\Gamma+2}$	$\cdots$	$\cdots$	Δd_t

$$\Delta \boldsymbol{d}_t = \boldsymbol{s}^t - \hat{\boldsymbol{s}}^t$$

$\boldsymbol{s}^t$– real latitudes and longitudes from t-Γ to t

$\hat{\boldsymbol{s}}^t$– predicted latitudes and longitudes from t-Γ to t

MDN based Aggressive Driving Behavior Detection

Aggressive driving behavior detection

Step 2: Use the boxplot method on $\Delta \boldsymbol{d}_t$ to extract its temporal consistency feature TC

- Calculate TC as $[\rho_1, \rho_2]$ with $\rho_1 = Q_3 - 1.5(Q_3 - Q_1)$ and $\rho_2 = Q_3 + 1.5(Q_3 - Q_1)$
- Introduce temporal consistency score TCS to indicate whether Δd_t satisfies the temporal consistency feature

$$TCS_t = \begin{cases} 1 \qquad\qquad , & if\ \Delta d_t \in TC \\ e^{-|\Delta d_t - 0.5(\rho_1 + \rho_2)| / \rho_2 - \rho_1}, & otherwise \end{cases}$$

MDN based Aggressive Driving Behavior Detection

Aggressive driving behavior detection

Step 3: Consider aggressive behaviors of multiple surrounding vehicles by analyzing both spatial and temporal consistency features

- Calculate TCS_t of all surrounding vehicles to form a temporal-spatial consistency vector $SC_t = [TCS_t^1; TCS_t^1; \cdots; TCS_t^8]$
- Calculate mean $\emptyset_t$ and absolute deviation δ_t of SC_t in time period Γ to form its Hampel filter boundaries

$$[\emptyset_t - \beta\delta_t, \emptyset_t + \beta\delta_t]$$

- Determine whether a surrounding vehicle has an aggressive driving behavior based on TCS_t and Hampel filter boundaries

Challenge 2

How to make control decisions with considering aggressive driving behaviors of surrounding vehicles

Reinforcement Learning based Control Decision Making

Driving safety and risk-based reward functions

Design a driving safety-based reward function and a driving risk-based reward function for reinforcement learning

- Driving safety-based reward function $r(a, t)$ helps the AV to drive on the road safely

$$r(s, a) = e^{-\mu_1 \left|\left|d_G - \boldsymbol{d}_R\right|\right| + \mu_1 |v_G - \boldsymbol{v}_R|}$$

- Driving risk-based reward function $\tilde{r}(a, t)$ helps the AV to drive with low driving risk

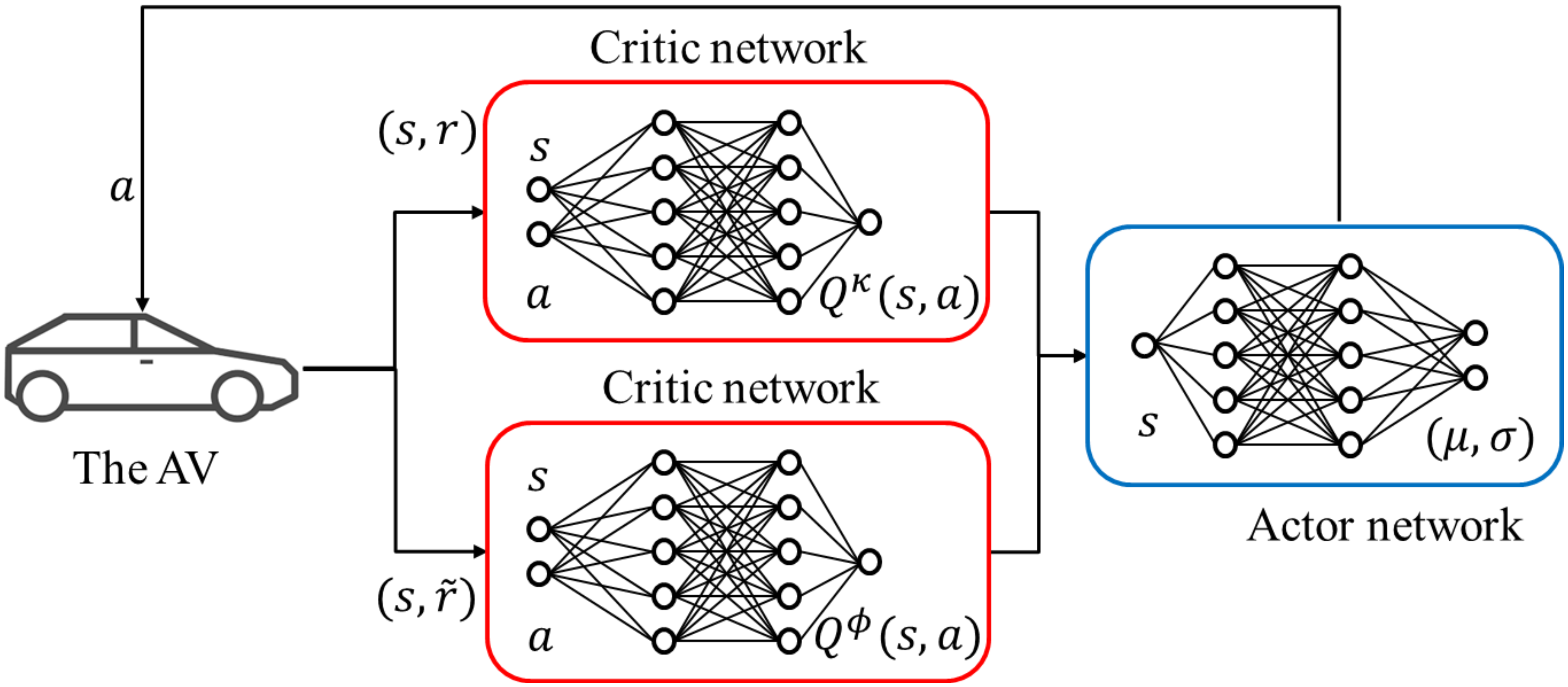
$$\tilde{r}(a, t) = e^{d(s,a)}$$

Where $d(s, a)$ denotes whether driving area is formed by surrounding vehicles with aggressive driving behaviors

Reinforcement Learning based Control Decision Making

Reinforcement learning for making control decisions

Uses an actor-critic architecture to approximate control policies and value functions to make control decisions for an AV



Performance Evaluation

Experiment settings

- Implement ReDS by running MATLAB on one laptop (Intel i5 CPU and 16 gigabyte memory)
- Contain 1,048,575 trajectory samples from total 3,366 vehicles driving in the traffic dataset

Comparison methods

- Markov decision process-based control decision system (MDP) [IV’17] and Reinforcement learning based control decision system (RLD) [ACC’18]

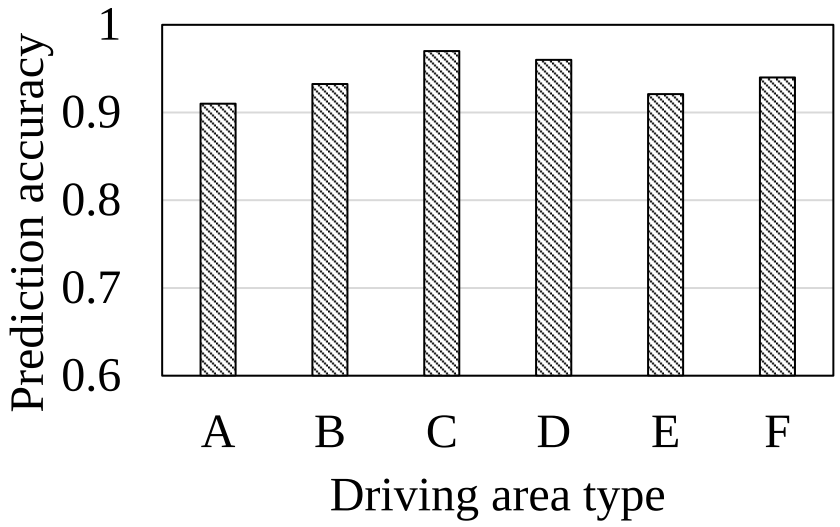
Evaluation metrics

- Aggressive driving behavior detection accuracy
- Optimal control decision success rate

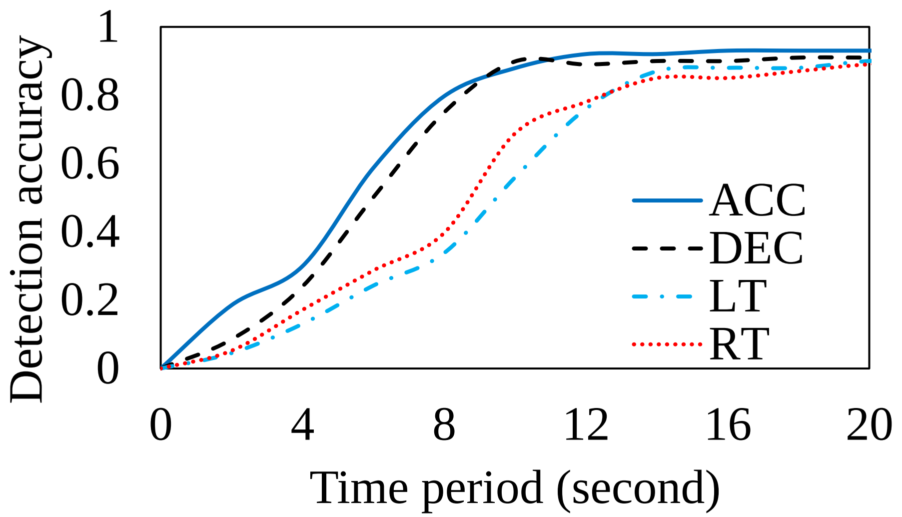
Performance Evaluation

Aggressive driving behavior detection accuracy of ReDS

- Driving behavior prediction accuracies of ReDS are more than 90% on different driving areas
- ReDS keeps almost constant aggressive driving behavior detection accuracies as the time period is larger than 14 seconds



Average driving behavior prediction accuracies

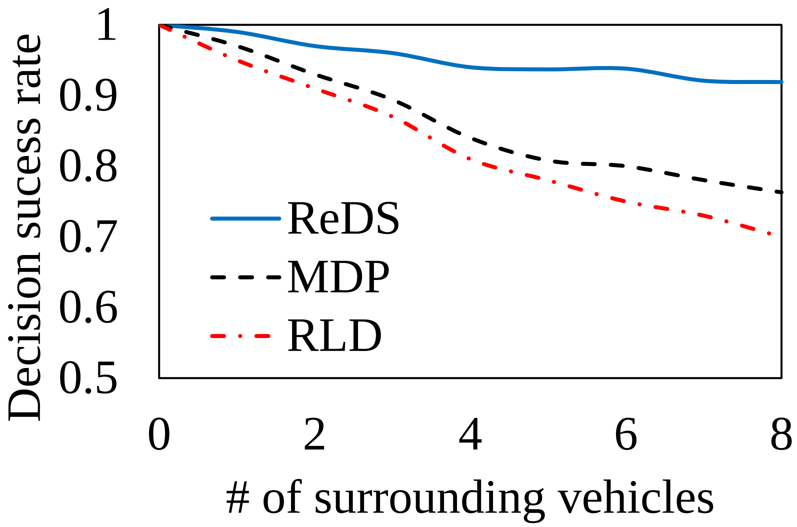


Aggressive driving behavior detection Accuracy comparisons

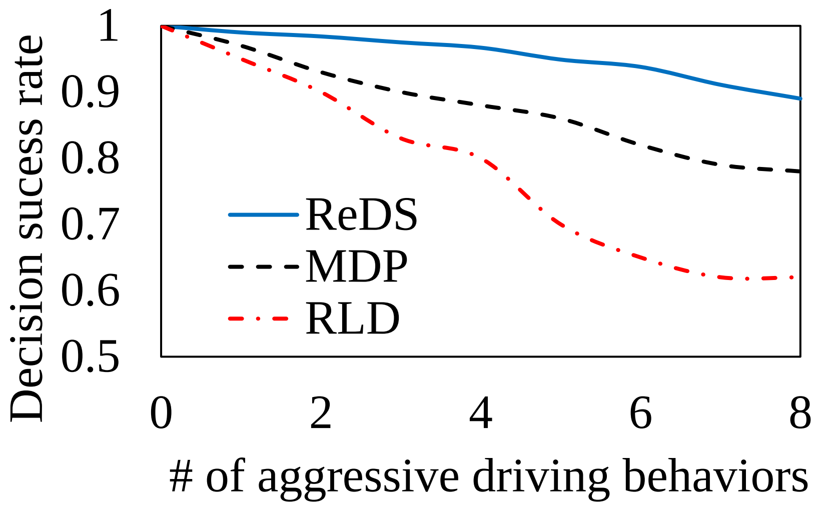
Performance Evaluation

Optimal control decision success rate of ReDS

- ReDS has higher optimal control decision success rates on different driving situations
- ReDS has more stable optimal control decision success rates as multiple aggressive driving behaviors exist



Success rates as the number of surrounding vehicles increases



Success rates as the number of aggressive driving behaviors increases

Summary

Propose ReDS to determine optimal control decisions for an AV with aggressive driving behavior consideration

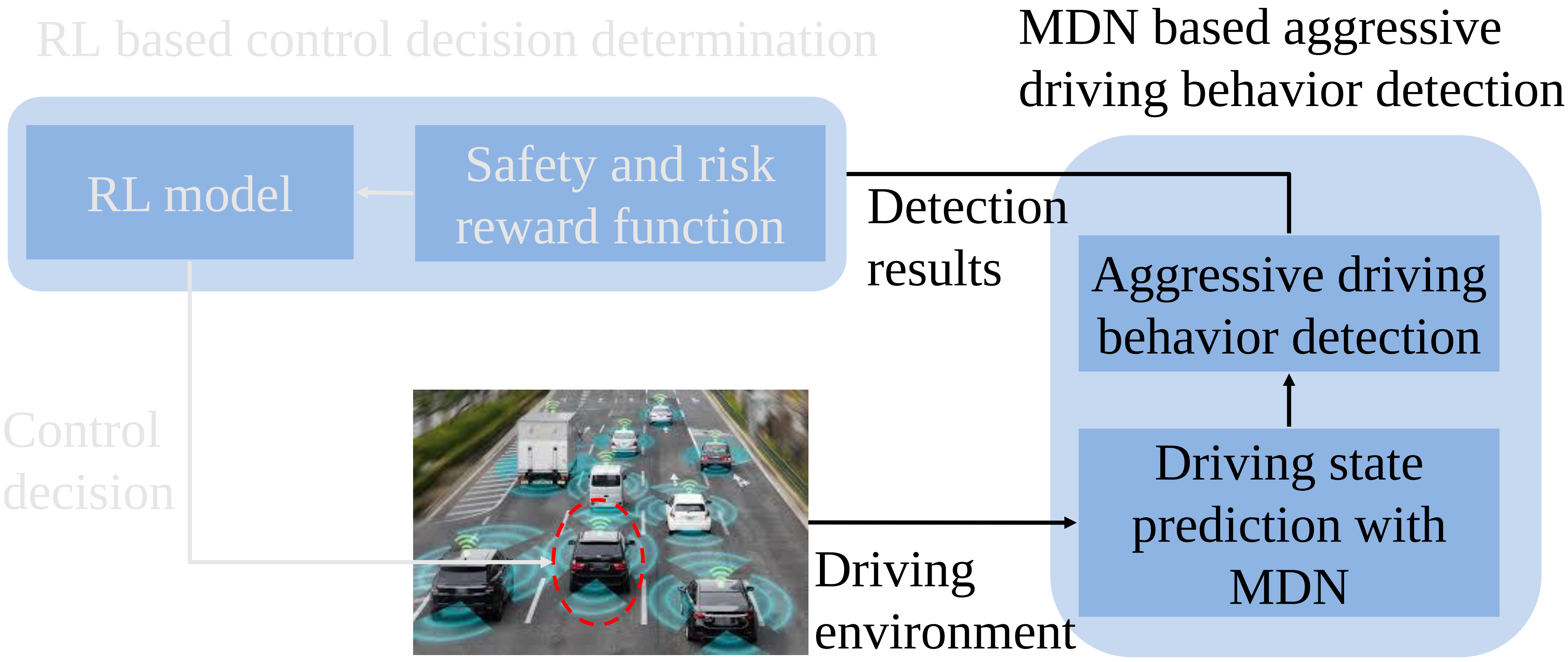
- Built a MDN based aggressive driving behavior detection method
- Developed driving safety and risk-based reward function to make control decisions using reinforcement learning
- Used real-world vehicle trajectory data to verify ReDS

Future work

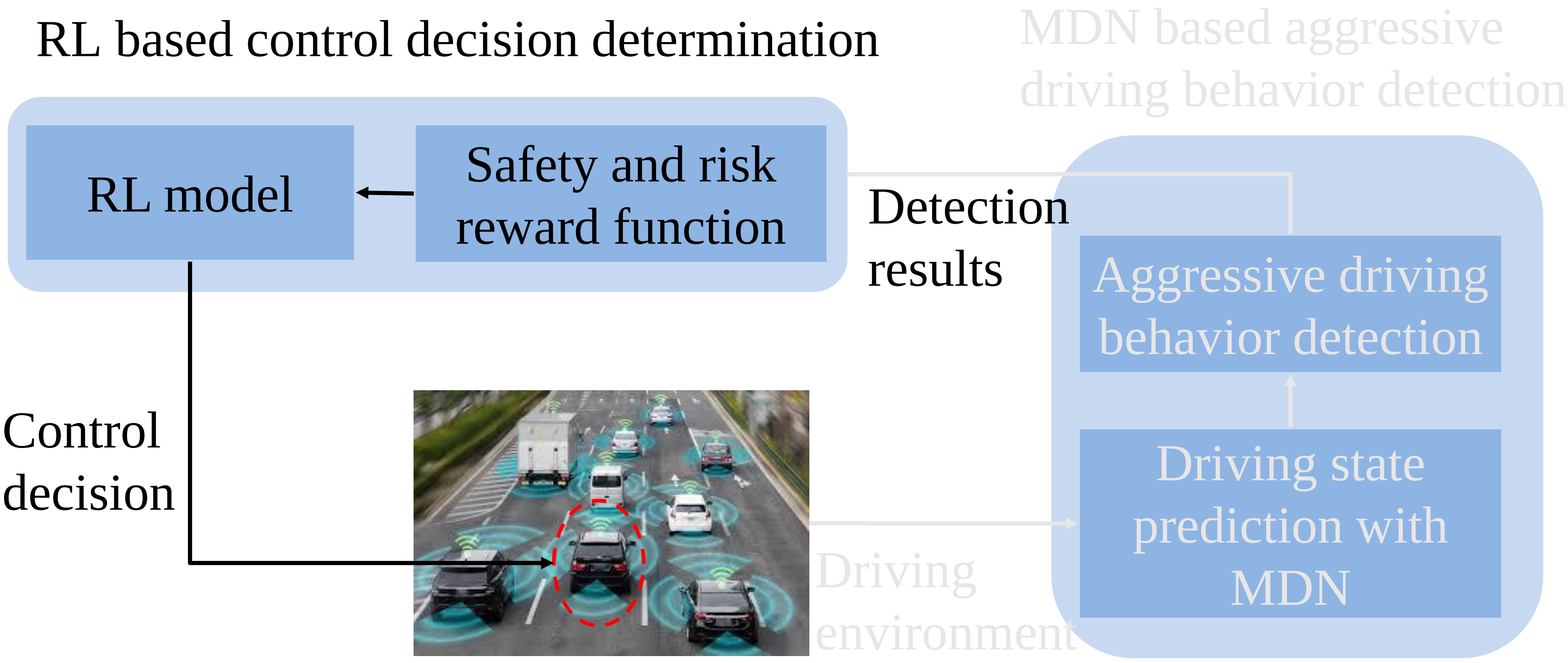
- Consider more driving factors (e.g., vehicle response time)

Thank you!

Reinforcement Learning based Decision-making System



Reinforcement Learning based Decision-making System



MDN based Aggressive Driving Behavior Detection

MDN working mechanism

A combination of a mixture density model and a neural network to estimate the underlying distribution of the data

- Probability density function $p(y|x)$ indicates the probability density of the data sample with kernel functions

$$p(y|x) = \sum_{m=1}^{M} \beta_m(x)\kappa_m(y|x)$$

x – a set of data samples
y – the output for a set of data samples
$\beta_m(x)$ – the mixing coefficient
$\kappa_m(y|x)$ – the Gaussian kernel function

Reinforcement Learning based Control Decision Making

Interaction between a target AV and surrounding vehicles

Focus on surrounding vehicles which are near to the AV and build a dynamic interaction system to update driving states of surrounding vehicles and the AV

- Step 1: Update driving state $a_G(t)$ of the AV after the AV conducts its control decision $a_G(t)$

$$s'_G(t) = f(s_G(t), \boldsymbol{s}_R(t), a_G(t))$$

- Step 2: Update driving states $\boldsymbol{s}_R(t)$ of surrounding vehicles after the surrounding conducts their control decision $\boldsymbol{a}_R(t)$

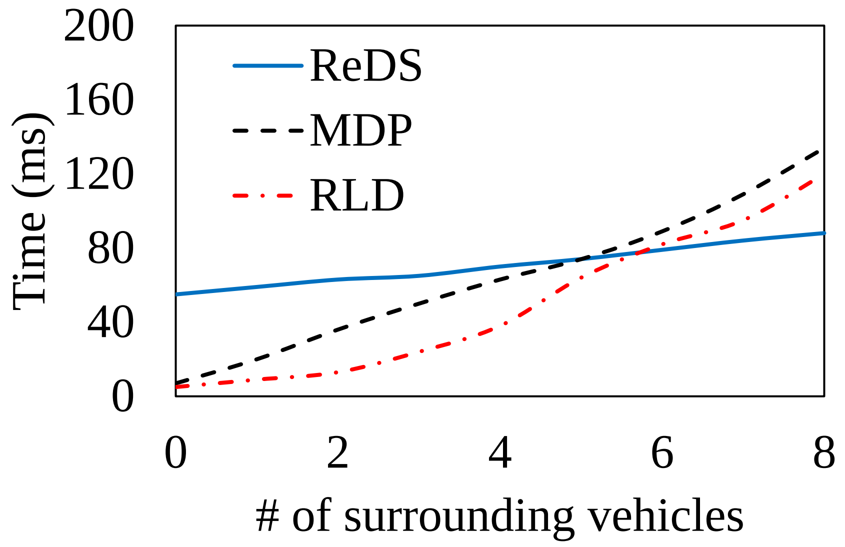
$$\boldsymbol{s}'_R(t) = f(s'_G(t), \boldsymbol{s}_R(t), \boldsymbol{a}_R(t))$$

Where dynamic interaction system f is obtained through the MDN based driving behavior model

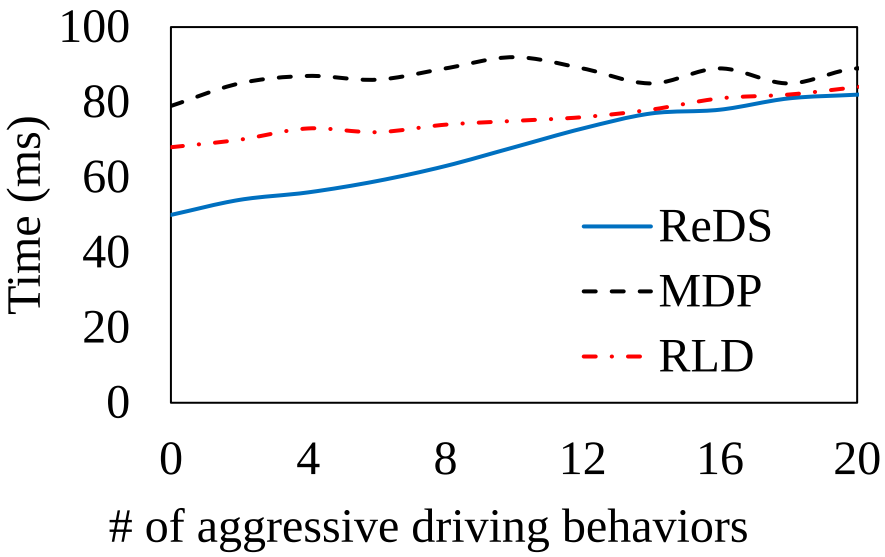
Performance Evaluation

Computation time of ReDS

- Computation time of ReDS is more stable as the increase of surrounding vehicle numbers
- ReDS needs more computation time as aggressive driving behavior numbers increases



Computation time as the number of surrounding vehicles increases



Computation time as the number of aggressive driving behaviors increases