Safe Reinforcement Learning on Autonomous Vehicles

David Isele, Alireza Nakhaei, and Kikuo Fujimura
Honda Research Institute USA
{disele, anakhaei, kfujimura}@honda-ri.com

***Abstract*— There have been numerous advances in reinforcement learning, but the typically unconstrained exploration of the learning process prevents the adoption of these methods in many safety critical applications. Recent work in safe reinforcement learning uses idealized models to achieve their guarantees, but these models do not easily accommodate the stochasticity or high-dimensionality of real world systems. We investigate how prediction provides a general and intuitive framework to constraint exploration, and show how it can be used to safely learn intersection handling behaviors on an autonomous vehicle.**

I. INTRODUCTION

With the increasing complexity of robotic systems, and the continued advances in machine learning, it can be tempting to apply reinforcement learning (RL) to challenging control problems. However the trial and error searches typical to RL methods are not appropriate to physical systems which act in the real world where failure cases result in real consequences.

To mitigate the safety concerns associated with training an RL agent, there have been various efforts at designing learning processes with safe exploration. As noted by Garcia and Fernandez [1], these approaches can be broadly classified into approaches that modify the objective function and approaches that constrain the search space.

Modifying the objective function mostly focuses on catastrophic rare events which do not necessarily have a large impact on the expected return over many trials. Proposed methods take into account the variance of return [2], the worst-outcome [3], [2], [4], and the probability of visiting error states [5]. Modified objective functions may be useful on robotic systems where a small number of failures are acceptable. However on safety critical systems, often a single failure is prohibited and learning must be confined to *always* satisfy the safety constraints.

For this reason methods that constrain the search space are often preferable. Because these approaches can completely forbid undesirable states, they are usually accompanied by formal guarantees, however satisfying the necessary conditions on physical systems can be quite difficult in practice. For example, strategies have assumed a known safe policy which can take over and return to safe operating conditions[6], a learning model that is restricted to tabular RL methods [7], [8], and states that can be deterministically perceived and mapped to logical expressions [9], [10].

While there are some approaches that have been implemented on physical robots such as the work of Gillula et al. which uses reachability to enforce strict safety guarantees [11], these approaches tend to be computationally expensive, preventing their application to high dimensional problems such as domains with multiple agents.

Fig. 1. An autonomous vehicle navigating an intersection. Prediction is used to shield the vehicle from making dangerous decisions, while allowing it to learn policies that are both efficient and not disruptive to other vehicles.

We investigate how prediction can be used to achieve a system that scales better to higher dimensions and is more suited to noisy measurements. Using prediction methods we show that we can safely constrain learning to optimize intersection behaviors on an autonomous vehicle where it must consider the behaviors of multiple other agents. While we believe prediction is a very general framework that lends itself to implementations on a variety of stochastic physical systems, we note that its safety constraints are weaker than other approaches in the literature: we assume other agents (traffic vehicles) follow a distribution and are not adversarial.

The specific application we investigate is making a turn at an unsigned intersection. This problem was recently explored as a non-safety constrained RL domain [12] where it was noted that the learned policy, which optimized efficiency, might be disruptive to traffic vehicles in practice. The primary concerns of these maneuvers are safety and efficiency, but balancing the two is a dynamic task. In dense traffic we may wish to seize an opportunity that leaves only several meters of a safety margin, as we might have to wait an unacceptable amount of time for the next opportunity. However in sparse traffic, adding an increased margin will only add a negligible delay and will likely be preferable to the passengers and other traffic vehicles. Figure 2 demonstrates the scenarios in dense and sparse traffic.

We demonstrate the use of prediction as a safety constraint by learning a policy that minimizes disruption to traffic (as measured by traffic braking) while avoiding collisions. Additionally, we learn a policy that maximizes distance to

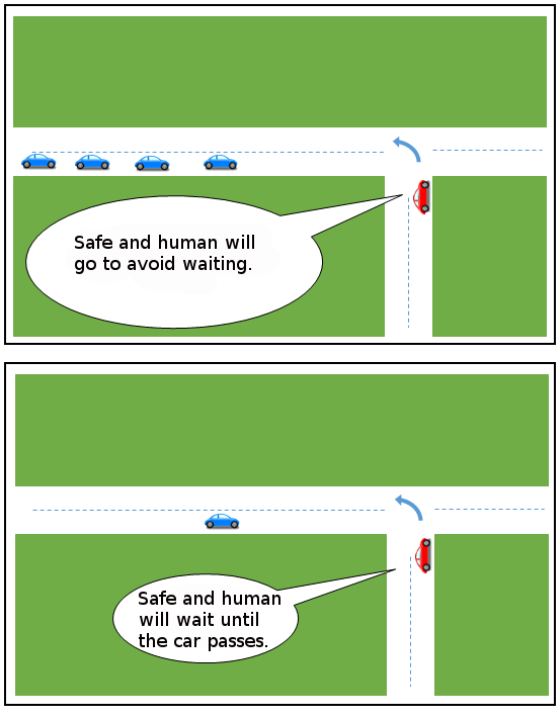

Fig. 2. In dense traffic, a human driver (red vehicle) might take the opening to minimize the wait that would result from the approaching heavy traffic. However, in sparse traffic, it would be more preferable to accept a small delay and let the car pass.

other vehicles, while still getting through the intersection in a fixed time window. We show that these two optimizations produce different behaviors and that both can be learned using RL with 0 collisions.

II. PROBLEM STATEMENT

In this document we use the subscript/superscript notation $variable_{time}^{agent,action}$. We define a safe set of policies $\mathbf{\Pi}^i$ as the set of policies π that generates a trajectory τ that with probability less than δ has agent i entering a danger state at any step in its execution[1].

To find a safe policy in a multi-agent setting, we formulate the problem as a stochastic game. In a stochastic game, at time t each agent i in state s_t takes an action a_t^i according to the policy π^i. All the agents then transition to the state s_{t+1} and receive a reward r_t^i. Stochastic games can be described as as tuple $\langle \mathcal{S}, \mathbf{A}, P, \mathbf{R} \rangle$, where $\mathcal{S}$ is the set of states, and $\mathbf{A} = \{\mathcal{A}^1, \ldots, \mathcal{A}^m\}$ is the joint action space consisting of the set of each agent's actions, where m is the number of agents. The reward functions $\mathbf{R} = \{\mathcal{R}^1, \ldots, \mathcal{R}^m\}$ describe the reward for each agent $\mathcal{S} \times \mathbf{A} \to \mathbf{R}$. The transition function $P : \mathcal{S} \times \mathbf{A} \times \mathcal{S} \to [0, 1]$ describes how the state evolves in response to all the agents' collective actions. Stochastic games are an extension to Markov Decision Processes (MDPs) that generalize to multiple agents, each of which has its own policy and reward function.

[1]Interesting corner cases were proposed for many existing definitions of safety by Moldovan and Abbeel [13]. Their proposed definition of safety in terms of ergodicity does not easily extend to a multi-agent setting.

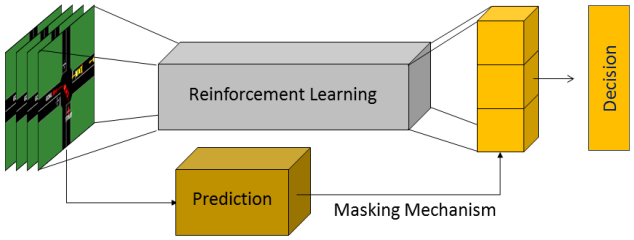

Fig. 3. Proposed pipeline.

Let x_t^i be the local state of a single agent. We will refer to the sequence of the local states, actions, and rewards for a single agent as a trajectory $\tau^i = \{(x_1^i, a_1^i, r_1^i), \ldots, (x_T^i, a_T^i, r_T^i)\}$ over a horizon T.

The goal is to learn an optimal ego-agent policy π^{*ego} where at every point in the learning process $\pi^{ego} \in \mathbf{\Pi}^{ego}$.

III. PROPOSED PIPELINE

To achieve safe learning we propose the following pipeline presented in Figure 3. In this pipeline, perception provides input to both an RL network and a prediction module. The prediction module masks undesired actions at each time step.

Given predictions, we can mask actions that result in unsafe behaviors. Masking unsafe behaviors in RL has also very recently been proposed in the RL community when states can be mapped to linear temporal logic [9]. This pipeline provides us with a mechanism to explore the safe subspace of an agent's possible behaviors during training and also guarantees that RL picks safe decisions during execution.

IV. PREDICTION

We propose using prediction models to mask unsafe actions from the agent, and then allow the agent to freely explore the safe state space using traditional RL techniques. Probabilistic predictions serve as an approximation to the computationally expensive task of identifying safe trajectories.

RL algorithms have been proposed for stochastic games without any restrictions on safety [14], [15]. However, in order to ensure that the agent never takes an unsafe action, we must check not only that a given action will not cause the agent to transition to an unsafe state in the next time step, but also that the action will not force the agent into an unsafe state at some point in the future. Note that this is closely related to the credit assignment problem, but the risk must be assigned prior to acting. One might imagine that ensuring the agent safely avoids all dangerous situations requires branching through all possible action combinations for a fixed time horizon T. Brute force implementations would result in an intractable runtime of $O\left(|\mathbf{A}|^T\right)$, where $|\mathbf{A}| = |\mathcal{A}^1| \times \cdots \times |\mathcal{A}^m|$. Indeed it has been shown that even for the more restricted case of MDPs, identifying the set

of safe policies is NP-Hard [13]. For this reason we look at efficient approximations for restricting our exploration space.

To reduce the complexity we assume the actions at each time step are components of a high-level action (known as options in the RL literature, and intentions in the autonomous driving literature). This has the effect of collapsing the branching factor of time associated with the exponential complexity. The cost of this approximation is that, for the fixed horizon, each agent is restricted in their ability to react and interact with other agents.

To accommodate the breadth of low-level action sequences that correspond to a single high-level action and also to allow for a bounded level of interaction, we make each high-level action a probability distribution over functions f. First we describe the trajectory of agent i in terms of high-level actions $p(\tau^i) \approx \prod_{j=1}^{|h^i|} p(h^i = j) p_{h^{i,j}}(x_1^i, \ldots, x_T^i)$. Here j indexes the high-level action h. Then we describe the functional local-state update as $x_{t+1} = f^{i,j}(x_t) + \epsilon_t$ where we model the noise as a Gaussian distribution $\epsilon_t = \mathcal{N}(0, \sigma_t)$. This means that the updated local state has a corresponding mean and variance.

Within the fixed time horizon, each agent takes a single high-level action. The variance acts as a bound that encompasses the variety of low-level actions that produce similar high-level actions. Additionally, we will use the variance to create safety bounds. These bounds allows for a bounded ability of each agent to react to other agents without violating our safety constraints. This can be thought of as selecting an open-loop high-level decision followed by subsequent bounded closed-loop low-level corrections. Note that by restricting an agent's ability to interact and limiting each agent to a restricted set of high-level actions, we are ignoring the existence of many pathological cases that may arise in an adversarial setting.

Given the assumption of high-level actions that follow a distribution, satisfying safety constraints can be computed in $O(|\mathbf{H}|T)$ where $|\mathcal{H}^i|$ is the number of high level actions available to agent i and $|\mathbf{H}| = |\mathcal{H}^1| \times \cdots \times |\mathcal{H}^m|$. This is still expensive for problems with a large number of actions or agents.

A further simplifying assumption arises when we assume an agent's action space is unimodal. This is the case when we assume the agent has a single action (e.g. a constant velocity assumption) or we make a hard prediction of the most probable action. This reduce the time complexity of a forward safety-checking prediction to $O(mT)$.

V. SAFETY GUARANTEES

One might suspect that our relaxations make it difficult to provide any safety guarantees. It does greatly limit the strength of the guarantees we can make, however we can still provide probabilistic guarantees on safety.

From Chebyshev's inequality we can state that the likelihood of an agent i taking action j leaving its safety margins $k\sigma^{i,j}$ is $p[|\tau^{i,j} - \mathbb{E}(\tau^{i,j})| \geq k\sigma^{i,j}] \leq \frac{1}{k^2}$. Where we define $|\tau^{i,j} - \mathbb{E}(\tau^{i,j})| \equiv \max_k |x_k^{i,j} - \mathbb{E}(x_k^{i,j})|$. Note that we use the weaker Chebyshev inequality for our bounds since according to the Fisher−Tippett−Gnedenko theorem the $\max$ operation results in a distribution that is not Gaussian.

Since we generally only care about one-sided error (e.g. if the traffic car is further away than predicted, we do not risk a collision) we can shrink the error by a factor of two. $p[\tau^{i,j} - \mathbb{E}(\tau^{i,j}) \geq k\sigma^{i,j}] \leq \frac{1}{2k^2}$.

In our model, we assume our safety margins create an envelope for an agent's expected trajectory. Collecting sufficient samples of independent trials, we can assume the predicted trajectory roughly models the reachable space of the agent. Now we probe the independence of trials. In expectation the agent follows the mean, but on each trial the deviations are likely not a purely random process, but are biased by a response to other agents.

In the autonomous driving literature it is assumed that agents behave with self-preservation [16]. We will assume that the measured distribution of the trajectory is the sum of two normally distributed random processes: the first associated with the agent's control and the second a random noise variable. The measured variance of a trajectory σ_M^2 is the sum of controlled σ_c^2 and noise σ_n^2 variances. We can express these relative to the measured standard deviation of the trajectory as $\alpha_c \sigma_M$ and $\alpha_n \sigma_M$ where $\alpha_c^2 + \alpha_n^2 = 1$ If we assume an agent controls away from the mean by $\kappa_c \alpha_c \sigma_M < k\sigma_M$ the probability that an agent leaves its safety margin is $p[\tau^{i,j} - \mathbb{E}(\tau^{i,j}) \geq \kappa_n \alpha_n \sigma_M] \leq \frac{1}{2\kappa_n^2}$, where $\kappa_n = \frac{k + \kappa_c \alpha_c}{\alpha_n}$. To put this in concrete terms for an autonomous driving scenario, if we assume a 5m measured standard deviation, 4m control standard deviation, 3m noise standard deviation, safety margin of $3\sigma_M$, and control action of $2\sigma_c$, the resulting safety margin is $7.6\sigma_n$. This analysis neglects any corrective controls of the ego agent. Applying the union bound and assuming a fixed κ_n for notational clarity, we can achieve our desired confidence δ by satisfying

$$\frac{m}{2\kappa_n^2} < \delta \ . \tag{1}$$

VI. APPLICATION TO AUTONOMOUS DRIVING

There are many works in the autonomous driving literature that look at risk-averse driving [17] and risk assessment [18], [17], [19]. Prediction is often used for safety in autonomous driving and accurate prediction models are a current topic of research in the autonomous driving community [16]. Simpler models are built upon kinematic motion models [20] with added uncertainty estimates to allow for errors in the measurements and assumptions [21]. These methods are limited in that the Gaussian probability models and kinematic transitions assume cars roughly follow a known trajectory. More sophisticated models allow for multiple maneuvers [22] which can be done by including road information (either heuristically or learned for particular intersections [23]) to allow for multiple possible maneuvers. More recent work in vehicle prediction is starting to consider the interactions between multiple vehicles [24], [25].

Related work has looked at learning policies for intersection handling [26], [27], [28], [29] however these approaches are restricted to simulation and do not investigate the issue

of preserving safety throughout the learning process under uncertainty. Using prediction as a safety constraint does not necessarily require additional learning. Accurate prediction models could be sufficient to create behaviors that enforce safety constraints. A robust vehicle prediction module could itself be used to safely navigate intersections:

1) Predict the movement of the ego car entering the intersection in conjunction with forward predictions of all other vehicles.
2) If a collision is predicted, wait.
3) Otherwise, go.

This however assumes a fixed behavior for the ego car - a single acceleration profile, a set time allowance to enter the intersection, and a set safety buffer to leave between cars.

Limiting the behavior of the autonomous vehicle to a one-size-fits-all motion is likely to lead to sub optimal behavior. Certain intersections may call for more aggressive accelerations to prevent excessive waiting. And the ability of other agents to interpret our actions can be just as important to safety as leaving sizable margins to allow for uncertainty. This work sets up a methodology by which we can explore the more nuanced aspects of decision making. As an example we consider learning a model that minimizes traffic disruptions.

VII. EXPERIMENTS

To demonstrate how prediction can be used as a safety constraint, we use deep Q-learning networks (DQNs) to learn policies that optimize aspects of intersection handling on autonomous vehicles. We consider two objectives. The first objective is to learn an adaptive stand off which seeks to increase the safety margin without compromising the ability to make the turn given a fixed time window. The second model looks at minimizing the disruption to other vehicles while navigating the intersection in the given time.

A. Prediction

We model traffic vehicles using a constant velocity assumption based our Kalman filter estimates of the detected vehicle. Each vehicle is modeled with a fixed 2m uncertainty in detection. An additional uncertainty per time step is accumulated forward in time following a quadratic curve which was fit to data collected from errors in the forward velocity assumption targeting a margin of six standard deviations. This allows the model to make allowances for some accelerations and braking of the traffic vehicles. The ego car has similar forward predictions of its behavior based on the target trajectory and three potential acceleration profiles. The prediction errors are smaller for the ego car, since the intentions are known in advance. At each time step, going forward in time until the ego car has completed the intersection maneuver, the predicted position of the ego car is compared against the predicted position of all traffic cars. If an overlap of the regions is detected, the action is marked as *unsafe*. Actions that are marked as *safe* are passed on to the network as permissible actions. If there are no permissible actions, or the network chooses to wait, the system waits at the intersection. Otherwise the vehicle moves forward with the selected acceleration until it reaches its target speed.

B. Simulation

Experiments were run using the Sumo simulator [30], which is an open source traffic simulation package. To simulate traffic in Sumo, users have control over the types of vehicles, road paths, vehicle density, and departure times. Traffic cars follow the Intelligent Driver Model (IDM) [31] to control their motion. In Sumo, randomness is simulated through driver imperfection models (based on the Krauss stochastic driving model [32]). The simulator runs based on a predefined time interval which controls the length of every step. For our experiments we use 0.2 second time step.

Each lane has a 30 mile per hour (13.4 m/s) max speed. The car begins from a stopped position. The maximum number of steps per trial is capped at 100 steps, which is equivalent to 20 seconds, starting from the first time prediction says a safe action is possible. This guarantees that a safe action is always possible in the allotted time. We use a 0.1 probability that a vehicle will be emitted per second to set the traffic density for our experiments.

The DQN architecture is modeled after the network presented in [12]. The simulator is designed to see cars 100m in either direction. This was selected to correspond to 25mph intersections. If we assume that it takes roughly 5 seconds to enter an intersection from a stopped position (measured from human demonstrations), we would like to detect traffic at a minimum of 55m from the intersection. This minimum increases to 75m when we allow for traffic vehicles that travel slightly above the speed limit. The remaining 25m provide an added buffer to allow for accurate detection and tracking. The IBEO sensors we use on the real vehicle are specified at 200m max range. The representation bins the traffic car positions into 26 bins per lane. Each lane is depicted as a separate row. Each spatial pixel, if occupied, contains the normalized real valued heading angles, velocity, and binary indicator.

The network has four outputs corresponding to *wait* and *go* commands where *go* can select from three different accelerations (0.5, 1.0, and 1.5 m/s^2). The network is optimized using the RMSProp algorithm [33].

Each network was trained on $20,000$ simulations. When learning a network that seeks to minimize braking. The per trial reward is $+1$ for successfully navigating the intersection with a -0.1 penalty applied for every time step a traffic vehicle was braking.

When learning a behavior that seeks to maximize the safety margin, the per trial reward is

$$r = \begin{cases} 0.1(d-10), & \text{if success} \\ z, & \text{if timeout} \end{cases}$$

Where d is the minimum distance the ego car gets to a traffic vehicle during the trial. d can be a maximum of 50m and the minimum observed distance during training is 4m. We conduct experiments with different z values $z = \{-1, -5, -10\}$ to study the affect on timeouts.

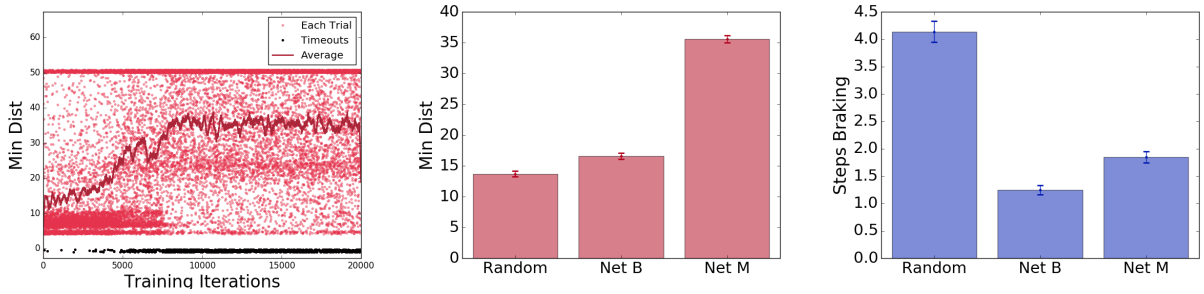

Fig. 4. **Left:** Minimum distance to traffic vehicles throughout the training process. **Center:** Comparison of the minimum distance to traffic vehicles using different policies. Net B is trained to minimize braking, net M is trained to maximize the minimum distance. **Right:** The amount of time steps traffic cars spend braking. We assume the ego vehicle is responsible for all traffic braking.

To evaluate the learned models we use two metrics: the average number of time steps per trial a traffic vehicle is braking, and the minimum distance between the ego car and the closest traffic car per trial. Statistics are collected over 1000 trials. Since both objectives could be improved by increasing the safety margin, we also conduct an experiment where we increase the safety margin of a rule-based only agent to ensure the learned policy gives us an improvement.

C. Real vehicle

We train in simulation and verify the learned policy on an autonomous vehicle. We collected data from an autonomous vehicle in Mountain View, California, at an unsigned T-junction, where the objective is to make a left turn. A point cloud, obtained from six IBEO Lidar sensors, is first pre-processed to remove points that reside outside the road boundaries. A clustering of the Lidar points with hand-tuned geometric thresholds is combined with the output from three Delphi radars to create the estimates for vehicle detection. Each vehicle is tracked by a separate particle filter. Figure 1 depicts an intersection where our algorithm was evaluated. We use the network trained to maximize the safety margin.

VIII. RESULTS

Using prediction as a constraint, we trained a DQN to minimize distance to other traffic vehicles. There were no recorded collisions during the entire training process. The left plot in Figure 4 shows how the minimum distance to traffic vehicles changes throughout the learning process when $z = -1$. Timeouts are shown as having a distance of -1 and are colored black. The minimum distance of each individual trial is plotted as a point. The moving average using a sliding window of 200 trials is shown in dark red. We see the concentration of points with a distance of less than 10m disappears at around 7000 training iterations, and there is an increased density in the region of 20m. Training does increase the number of timeouts. The extent of this can be changed by adjusting the penalty for timeouts, see Figure 5. Note that larger penalties produce larger gradients which can have an adverse affect on the learning process so there is a limit to how large the penalty can be set.

The number of trials that have a minimum distance of 50m or more also increases. This is more clear in Figure 5 where we show a histogram of the distances before and after training. Figure 6 shows that naively increasing the safety margin with a rule-based strategy using prediction gives much worse performance compared to the learned networks.

The network we trained to minimize braking should leave a large margin when moving in front of a car, however it may come up very close behind a car. We see that this is the case in center plot of Figure 4. As expected the network trained to maximize the minimum distance (Net M) greatly increases the average minimum distance. The network trained to minimize braking does achieve a larger average distance than a random policy acting in the safety constrained prediction framework, but the difference is much less pronounced. The network that was trained to maximize distance should also reduce braking. This result can be seen in right plot of Figure 4. Again we see the network specifically designed to minimize braking is better at satisfying the objective. The fact that these two objectives produce different behaviors makes sense. If we think in terms of the gap between two cars, it would be optimal to be in the middle of the gap if maximizing distance and the front of the gap if minimizing braking of other vehicles.

Qualitatively, we observe that the network trained to maximize the safety margin will often add short delays after prediction determines the situation is safe if traffic is sparse. In cases where traffic is more dense, the network is more likely to move the moment an opportunity presents itself.

IX. CONCLUSION

In this work we present a framework for safe RL using predictions to mask unsafe actions. We apply this methodology to an autonomous driving domain to learn policies that improve the performance of unsigned intersection handling. Specifically we look at 1) minimizing disruption to other vehicles and 2) maximizing safety margins while still navigating the intersection in a fixed time window.

While the safety guarantees we can make using prediction are not as strong as other approaches proposed in the literature, the framework is more general and likely more applicable to many real world applications. Since we are masking actions, some of which we know to be safe in order to provide safety margins when dealing with uncertainty,

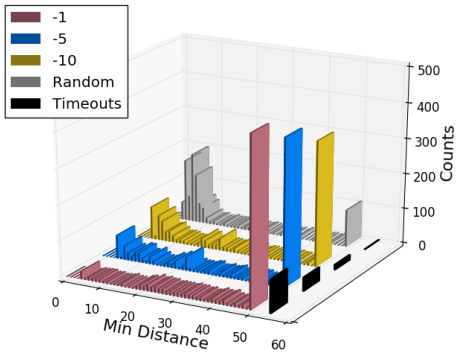

Fig. 5. Comparison of the effect of the timeout penalty on the network trained to maximize the minimum distance. Timeouts are shown in black.

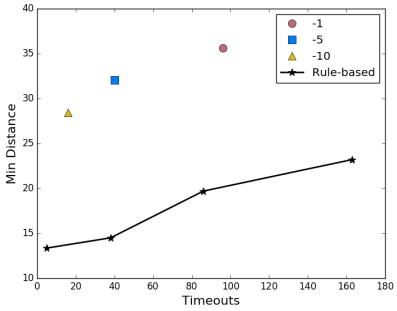

Fig. 6. Comparison between the networks trained with different timeout penalties, and a rule-based method with a fixed safety margin. The safety margin for the rule-based method is varied across trials.

the final policies are possibly suboptimal. This suggests open problems both related to developing more sophisticated prediction modules and a more careful characterization of the regret associated with them.

REFERENCES

[1] J. Garcıa and F. Fernández, "A comprehensive survey on safe reinforcement learning," *Journal of Machine Learning Research*, vol. 16, no. 1, pp. 1437–1480, 2015.

[2] R. A. Howard and J. E. Matheson, "Risk-sensitive markov decision processes," *Management science*, vol. 18, no. 7, pp. 356–369, 1972.

[3] M. Heger, "Consideration of risk in reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 105–111.

[4] Z. C. Lipton, J. Gao, L. Li, J. Chen, and L. Deng, "Combating reinforcement learning's sisyphean curse with intrinsic fear," *arXiv preprint arXiv:1611.01211*, 2016.

[5] P. Geibel and F. Wysotzki, "Risk-sensitive reinforcement learning applied to control under constraints." *J. Artif. Intell. Res.(JAIR)*, vol. 24, pp. 81–108, 2005.

[6] A. Hans, D. Schneegaß, A. M. Schäfer, and S. Udluft, "Safe exploration for reinforcement learning." in *ESANN*, 2008, pp. 143–148.

[7] M. Wen, R. Ehlers, and U. Topcu, "Correct-by-synthesis reinforcement learning with temporal logic constraints," in *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*. IEEE, 2015, pp. 4983–4990.

[8] M. Wen and U. Topcu, "Probably approximately correct learning in stochastic games with temporal logic specifications." in *IJCAI*, 2016, pp. 3630–3636.

[9] M. Alshiekh, R. Bloem, R. Ehlers, B. Könighofer, S. Niekum, and U. Topcu, "Safe reinforcement learning via shielding," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[10] S. Shalev-Shwartz, S. Shammah, and A. Shashua, "Safe, multi-agent, reinforcement learning for autonomous driving," *arXiv preprint arXiv:1610.03295*, 2016.

[11] J. H. Gillula and C. J. Tomlin, "Reducing conservativeness in safety guarantees by learning disturbances online: iterated guaranteed safe online learning," *Robotics: Science and Systems VIII*, p. 81, 2013.

[12] D. Isele, R. Rahimi, A. Cosgun, K. Subramanian, and K. Fujimura, "Navigating occluded intersections with autonomous vehicles using deep reinforcement learning," *IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[13] T. M. Moldovan and P. Abbeel, "Safe exploration in markov decision processes," *arXiv preprint arXiv:1205.4810*, 2012.

[14] M. L. Littman, "Markov games as a framework for multi-agent reinforcement learning," in *Machine Learning Proceedings 1994*. Elsevier, 1994, pp. 157–163.

[15] J. Hu and M. P. Wellman, "Nash q-learning for general-sum stochastic games," *Journal of machine learning research*, vol. 4, no. Nov, pp. 1039–1069, 2003.

[16] S. Lefèvre, D. Vasquez, and C. Laugier, "A survey on motion prediction and risk assessment for intelligent vehicles," *Robomech Journal*, vol. 1, no. 1, p. 1, 2014.

[17] F. Damerow and J. Eggert, "Risk-aversive behavior planning under multiple situations with uncertainty," in *Intelligent Transportation Systems (ITSC), 2015 IEEE 18th International Conference on*. IEEE, 2015, pp. 656–663.

[18] S. Lefèvre, D. Vasquez, C. Laugier, and J. Ibañez-Guzmán, "Intention-aware risk estimation: Field results," in *Advanced Robotics and its Social Impacts (ARSO), 2015 IEEE International Workshop on*. IEEE, 2015, pp. 1–8.

[19] S. Brechtel, T. Gindele, and R. Dillmann, "Probabilistic mdp-behavior planning for cars," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*. IEEE, 2011, pp. 1537–1542.

[20] R. Schubert, E. Richter, and G. Wanielik, "Comparison and evaluation of advanced motion models for vehicle tracking," in *Information Fusion*. IEEE, 2008, pp. 1–6.

[21] A. Carvalho, Y. Gao, S. Lefevre, and F. Borrelli, "Stochastic predictive control of autonomous vehicles in uncertain environments," in *12th International Symposium on Advanced Vehicle Control*, 2014.

[22] G. Aoude, J. Joseph, N. Roy, and J. How, "Mobile agent trajectory prediction using bayesian nonparametric reachability trees," *Proc. of AIAA Infotech@ Aerospace*, pp. 1587–1593, 2011.

[23] T. Streubel and K. H. Hoffmann, "Prediction of driver intended path at intersections," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 134–139.

[24] G. Agamennoni, J. I. Nieto, and E. M. Nebot, "Estimation of multivehicle dynamics by considering contextual information," *IEEE Transactions on Robotics*, vol. 28, no. 4, pp. 855–870, 2012.

[25] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *Intelligent Vehicles Symposium (IV), 2017 IEEE*. IEEE, 2017, pp. 204–211.

[26] W. Song, G. Xiong, and H. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[27] T. Gindele, S. Brechtel, and R. Dillmann, "Learning context sensitive behavior models from observations for predicting traffic situations," in *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*. IEEE, 2013, pp. 1764–1771.

[28] D. Isele and A. Cosgun, "Selective experience replay for lifelong learning," *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[29] M. Bouton, A. Cosgun, and M. J. Kochenderfer, "Belief state planning for navigating urban intersections," *IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[30] D. Krajzewicz, J. Erdmann, M. Behrisch, and L. Bieker, "Recent development and applications of SUMO–simulation of urban mobility," *International Journal on Advances in Systems and Measurements (IARIA)*, vol. 5, no. 3–4, 2012.

[31] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Physical Review E*, vol. 62, no. 2, p. 1805, 2000.

[32] S. Krauss, "Microscopic modeling of traffic flow: Investigation of collision free vehicle dynamics," Ph.D. dissertation, Deutsches Zentrum fuer Luft-und Raumfahrt, 1998.

[33] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop, coursera: Neural networks for machine learning," *University of Toronto, Tech. Rep*, 2012.