HyLEAR: Hybrid Deep Reinforcement Learning and Planning for Safe and Comfortable Automated Driving

Dikshant Gupta[1] and Matthias Klusch[2]

Abstract— **We present a novel hybrid learning method, named HyLEAR, for solving the collision-free navigation problem for self-driving cars in POMDPs. HyLEAR leverages interposed learning to embed knowledge of a hybrid planner into a deep reinforcement learner to faster determine safe and comfortable driving policies of the car. In particular, the hybrid planner combines pedestrian path prediction and risk-aware path planning with driving-behavior rule-based reasoning such that the determined safe trajectories also take into account, whenever possible, the ride comfort and a given set of driving-behavior rules. Our experimental performance analysis over the CARLA-CTS benchmark of critical traffic scenarios revealed that HyLEAR can significantly outperform the selected baselines in terms of safety and ride comfort.**

I. INTRODUCTION

The basic problem of collision-free navigation (CFN) of a self-driving car is to navigate on a driveable path to a given goal in minimal time and collisions with objects such as other cars or pedestrians in a partially observable traffic environment. This problem can be modeled as a partially observable Markov decision process (POMDP) to be solved by the car online and subject to the given car and pedestrian model. An additional challenge is to compute driving policies online that are not only safe but, whenever possible, also passenger comfortable and driving-rule compliant.

Current CFN methods for self-driving cars leverage either a deep reinforcement learner (DRL) [31], [16], [7], or an approximate POMDP planner (APPL) [21], [1], or a hybrid combination of thereof [23]. While a hybrid method of DRL-assisted online planning such as HyLEAP in [23] can outperform its individual components for collision-free navigation in terms of safety, it may suffer from long training and online planning time. However, it is not known yet, whether some hybrid method with opposite type of architecture, that is hybrid planning-assisted deep reinforcement learning for the same problem can be comparatively more efficient and safe. The interposed learning framework in [29] was shown to enable faster training of vanilla deep reinforcement learning but has not been applied to hybrid deep reinforcement learning for CFN yet. On the other hand, many works and user studies investigated the effect of various road and load disturbance factors including smoothness and passenger-perceived risk of driving actions [24], [15], [17] on ride comfort [27], [3], [8]. However, the potential of hybrid CFN methods taking ride comfort into account without compromising safety in critical scenarios remains unclear. Besides, self-driving cars are expected to navigate whenever possible in compliance with a given set of default driving-behavior or traffic rules, such as not to drive on sidewalks, or to keep the lane. In some critical situations, however, safe driving may require the violation of certain rules as experienced human drivers might do without making it a habit and still being able to explain their decision for the exceptional trajectory. In [5], a hierarchical rule-book is used for explainable (production rule) reasoning to select rule-compliant safe trajectories but without integration of hybrid learning and ride comfort.

To this end, we developed HyLEAR, a novel hybrid planning-assisted deep reinforcement learning method for collision-free navigation of self-driving cars in POMDPs. This hybrid method determines driving policies that are not only safe but also take into account, whenever possible, the ride comfort and a given set of driving-behavior rules. In particular, HyLEAR leverages interposed learning with a hybrid planner that combines pedestrian path prediction and risk-aware path planning with driving-behavior rule-based reasoning for this purpose.

For our experimental comparative performance evaluation of HyLEAR with selected baselines, we created the publicly available CARLA-CTS benchmark [12] that consists of critical traffic scenarios largely based on the GIDAS accident study [2] for the driving simulator CARLA, and sources of HyLEAR and selected baseline methods.

II. PROBLEM DESCRIPTION

As mentioned above, the collision-free navigation problem of a self-driving car is, in short, to minimize the time to a given goal subject to constraints with specific focus on avoiding near misses or even hits of pedestrians. In this section, we adopt from [23] the description of the same considered problem as discrete-time POMDP (cf. Sect. 2.2), as well as the underlying models of car and pedestrian (cf. Sect. 2.1). We then outline the set of traffic scenarios in our created virtual benchmark CARLA-CTS (cf. Sect. 2.3).

A. Car and pedestrian model

We assume the car to perceive its environment with a 360° surround view. Pedestrians in a perceived traffic scene are observable within a maximum viewing distance of 50 meters, if no other obstacle occludes them. Following [1], the car only knows the exact positions of observable pedestrians, and use the kinematic bicycle model in [18] to approximately

[1] Dikshant Gupta is with the Computer Science Department, Saarland University, 66123 Saarbruecken, Germany

[2] Matthias Klusch is with the German Research Center for Artificial Intelligence (DFKI), 66123 Saarbruecken, Germany matthias.klusch@dfki.de

model the car driving on the road. Accordingly, the *car state* at time t is defined as $(p_t^c, p_{goal}^c, v_t^c, \theta_t^c)$ with current position $p_t^c \in \mathbb{R}^2$, goal position $p_{goal}^c \in \mathbb{R}^2$, velocity $v_t^c \in \mathbb{R}^2$ and orientation $\theta_t^c \in [0, 2\pi)$ of the car. For goal-directed navigation of the car in continuous space of POMDPs, the anytime weighted hybrid A* k-path planner ensures that the generated paths are driveable for the car according to its kinematics, though paths are not guaranteed to be optimal and complete. The planning of safe car paths requires a cost-map. Our basic cost-map maps environmental information of the actual scene to a discretized grid map from a bird's eye view with obstacle costs of car states defined as maximum of 1, 50, and 100, if the car is on the road, partly on the sidewalk, with any part colliding with an obstacle, respectively, infinite else. The so-called car intention $\mathcal{I} \in \mathbb{R}^{400\times 400\times 3}$ is a small RGB segmentation image (400x400 snipet) taken from the standard cost-map with included past and the planned path of the car in CARLA as done, for example, in [10], [23]. Please note that, in contrast to HyLEAP in [23], our method HyLEAR uses a risk-aware anytime-weighted hybrid 3-path A* planner with not only the basic cost-map but two extensions of it to generate three alternative safe paths (cf. Section III).

The *pedestrian state* at time t is defined as (p_t^{ped}, p_{goal}^{ped}, v_t^{ped}, θ_t^{ped}) with current position $p_t^{ped} \in \mathbb{R}^2$, goal position $p_{goal}^{ped} \in \mathbb{R}^2$, velocity $v_t^{ped} \in \mathbb{R}^2$ and orientation $\theta_t^{ped} \in [0, 2\pi)$ of the pedestrian. In our experiments, pedestrians only move into the direction of the goal in a straight line. Accident areas for collisions between car and pedestrian are defined via rectangles including the car with safety margins (1.5m front, 0.5m back, 0.5m side) for the near-miss area, which are added to the hit or crash area, both associated with respectively negative rewards for the car (cf. Sect. II-B).

B. Collision-free navigation problem

We model the considered collision-free navigation problem as a POMDP $(S, A, T, R, \gamma, Z, O)$ just as in [23]:

S: Set $\{[c, ped_1, ..., ped_{|\mathcal{P}|}] \mid c \in \mathbb{R}^2 \times \mathbb{R}^2 \times [0, 2\pi),\ ped_i \in \mathbb{R}^2 \times \mathbb{R}^2 \times \mathbb{R}^2\ \times [0, 2\pi), 1 \le i \le |\mathcal{P}|\}$ of states $s_t \in S$ at time t with car state c and pedestrian states $ped_i = [ped_i^o, ped_i^h] \in \mathcal{P}$, $1 \le i \le |\mathcal{P}|$, where ped_i^o denotes the observable position of pedestrian i

A: Set $\{(\alpha, acc)\}$ of car driving control actions $a \in A$ with (a) steering angle $\alpha \in \mathbb{Z}, |\alpha| \le \alpha_{max} \wedge \alpha \bmod 25 = 0$, $\alpha_{max} = 50^o$ in both directions, and (b) step-wise speed action (acceleration) $acc \in SpeedActions$ where $SpeedActions$ is the set $\{Accelerate, Maintain, Decelerate\}$ with about $+5, 0, -5km/h$, respectively.

T: Transition probability $T(s_t, a_t, s_{t+1}) = p(s_{t+1}|s_t, a_t) \in [0, 1]$ of transitioning from state $s_t \in S$ into state $s_{t+1} \in S$ when executing action $a_t \in A$ at time t. State changes fully defined by car movement kinematics, pedestrian model.

Z: Set $\{[c, ped_1^o, ..., ped_{|\mathcal{P}|}^o]\}$ of observations $o_t \in Z$ with car state and observable part of perceived pedestrians in the scene.

O: Observation probability $O(s_{t+1}, a_t, o_{t+1}) = p(o_{t+1}|s_{t+1}, a_t) \in [0, 1]$ of observing $o_{t+1} \in Z$ when transitioning into s_{t+1} after executing a_t, and $O(s_{t+1}, a_t, o_{t+1}) = 1$, if $o_{t+1} = [c, ped'^o_1, ..., ped'^o_{|\mathcal{P}'|}]$ for pedestrians $\mathcal{P}' \subseteq \mathcal{P}$ currently perceived by car, else 0; states are not deducable from single observations.

R: Immediate reward $R(s_t, a_t) \in \mathbb{R}$ for executing action a_t in state s_t is:
$r_{goal} = +1000$, if goal position reached; r_{near_miss} -500, if pedestrian in near-miss area of car (rectangular area around car; includes smaller hit area) and $|v_t^c| > 0$; $r_{hit} = -1000$, if pedestrian in hit area and $|v_t^c| > 0$; $r_{obst} = -max\{obstCosts\}$, $obstCost$ defined in Sect. II-A; $r_{acc} = -0.1$, if ac-/deceleration; $r_{steer} = -1$, if steering: $|\alpha_t| > 0$; $r_{notgoal} = -0.1$ else.

γ: Discount factor in $[0, 1]$, we set $\gamma = 0.98$ similar to [1].

The reward function R encourages safe, fast, and smooth driving. Its component rewards r_{goal} and $r_{notgoal}$ encourage the car to reach the goal, and r_{near_miss} to keep a safety distance to perceived pedestrians, while r_{hit}, r_{obst} penalize crashing into a pedestrian or other obstacles. Small penalties r_{acc} and r_{steer} intend to avoid unnecessary acceleration and steering which decrease the smoothness of driving.

C. Benchmark CARLA-CTS

For comparative experimental evaluation of CFN methods of self-driving cars in critical traffic scenarios (cf. Sect. IV), we created the virtual CARLA-CTS benchmark[12] as an extension of OpenDS-CTS in [23] for the driving simulator CARLA[1]. This benchmark contains about thirty-thousand scenes of twelve scenarios (cf. Fig. 1) mainly based on the GIDAS study of road accidents with pedestrians in Germany [2] and simulated in CARLA on a test drive of about 100 meters. In Fig. 1, the blue boxes with solid arrow denote the ego-car movements, the red boxes with solid arrow denote static or dynamic obstacles such as parking or incoming cars, and the dotted arrows denote the pedestrian movement. The first nine scenarios in CARLA-CTS are based on GIDAS without incoming car and the same as in OpenDS-CTS [23], while CARLA-CTS also includes three additional scenarios with incoming car in different street environments.

III. Hybrid Solution HyLEAR

A. Overview

HyLEAR is a hybrid planning-assisted deep reinforcement learner that solves the above mentioned collision-free navigation problem but also addresses the challenge of ride comfort and driving behavior-rule compliance. In particular, it consists of a soft actor-critic deep reinforcement learner based on [13], named NavSAC, that is assisted by a hybrid planner which leverages functional modules for (a) risk-sensitive planning of three alternative safe and short paths, (b) driving behavior rule-based reasoning for the selection

[1] CARLA: https://carla.org/

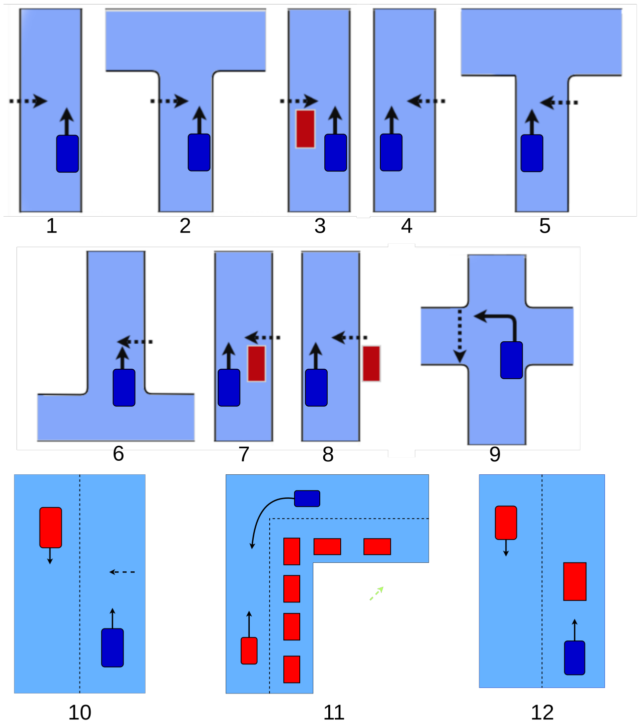

Fig. 1. CARLA-CTS traffic scenarios.

of one path with minimal risk and rule violations, and only during interposed learning of NavSAC, (c) online POMDP planning of an optimal speed action for the next step on this path. HyLEAR's training and testing architecture differs in that the latter does not rely on the computationally expensive online POMDP planner (cf. Figs. 3 and 4).

At each time step of driving scene simulation during training and testing of HyLEAR in CARLA, the risk-aware path planner uses an anytime weighted hybrid A* k-path planner and a multi-pedestrian path predictor M2P3 [22] to determine three alternative safe and short paths together with their estimated human passenger-perceived risk values as in [17]. While the first path is planned with the basic cost-map, the planning of two alternative paths relies on an extended cost-map with lower cost of driving on free sidewalks, and a further extended cost-map with pedestrian positions predicted with M2P3 as obstacles. The human passenger-perceived risk values for each of these alternative paths are computed based on the driver risk fields for the paths as in [17] together with the respective cost-map for path planning.

The rule-based reasoner of the hybrid planner performs hierarchical rule reasoning on a given rule-book as in [5] of initially four priority-ordered default driving-behavior rules (no driving on sidewalk, minimize risk, minimize lane changes, take shortest path) to select the safe path with minimal human-perceived risk and rule violations. The top-priority rule (avoid driving on sidewalks) in our current but easily extendable rule-book can be violated in emergency cases when no alternative path with acceptable risk is available that does not lead through the restricted area. If multiple paths satisfy a rule equally, the next rule for path selection with lower priority is applied. Eventually, for the next full car control action, the steering angle is extracted from this selected path and the optimal speed action is planned by the approximate POMDP planner IS-DESPOT [21], [1] during the interposed learning of NavSAC only and by the trained learner NavSAC during testing of HyLEAR.

B. Training

1) HyLEAR DRL network NavSAC: The off-policy soft actor-critic deep reinforcement learning network NavSAC (cf. Fig. 2) takes as input the car intention (cf. Sect. 2.1) including the selected best path determined by the hybrid planner, as well as the latest reward, current speed and previous action of the car in order to learn to output the best next speed action acc_t for this path. For this purpose, it processes the image with a convolution neural network with three convolution layers of filter size 8x8, 4x4, 3x3, strides of 4x4, 2x2, 1x1, and 32, 64, 64 number of filters, respectively. The flattened convolution layer output (flatten layer of 135424 units) is concatenated with the reward $\mathcal{R}_t \in \mathbb{R}$, speed $v_t^c \in \mathbb{R}^2$ and previous action $a_{t-1} \in \mathbb{R}^{|A|}$ (concatenation layer) and fed into fully connected layer consisting of 512 units. The output of the latter is then fed into three-way parallel fully connected layers that provide as an output the estimated state value $V^\psi \in \mathbb{R}$, Q-value $Q^\theta \in \mathbb{R}$, and estimated optimal speed action policy $\pi^\phi \in \mathbb{R}^{|SpeedActions|}$.

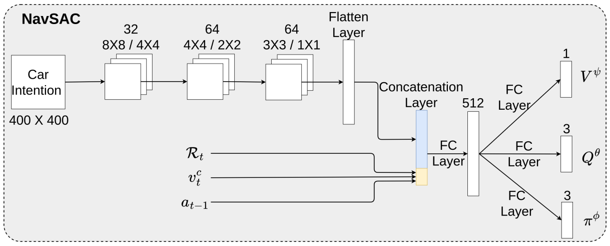

Fig. 2. NavSAC architecture.

For its off-policy soft actor-critic learning, NavSAC uses the loss functions [13]
$J_V(\psi) = \mathrm{E}_{o_t \sim D}\left[\frac{1}{2}(V^\psi(o_t) - \mathbb{E}_{a_t \sim \pi^\phi}\left[\mathrm{Q}^\theta(o_t, a_t) - \log \pi^\phi(a_t|o_t))\right])^2\right]$;
$J_Q(\theta) = \mathbb{E}_{(o_t, a_t) \sim D}\left[\frac{1}{2}(Q^\theta(o_t, a_t) - \hat{Q}(o_t, a_t))^2\right]$;
$J_\pi(\phi) = \mathbb{E}_{o_t \sim D}\left[\log \pi^\phi(a_t|o_t) - Q^\theta(o_t, a_t)\right]$, where $\hat{Q}(o_t, a_t) = R(o_t, a_t) + \gamma \mathbb{E}_{o_{t+1} \sim p}\left[\mathrm{V}^{\bar{\psi}}(o_{t+1})\right]$; observations o_t and actions a_t are sampled from memory buffer D, and ψ, θ and ϕ are the NavSAC network weights to be trained during its interposed learning with the hybrid planner of HyLEAR.

2) Interposed learning with hybrid planner: The training architecture of HyLEAR (cf. Fig. 3, Algorithm 1) follows the general interposed learning framework introduced in [29] for its soft actor-critic DRL network NavSAC assisted by the hybrid planner. At each time step t, a car control action a_t (steering angle α_t and speed action acc_t) is generated by the hybrid planner and then executed in CARLA according to given categorical probability distribution of speed actions that are produced by the hybrid planner with IS-DESPOT.

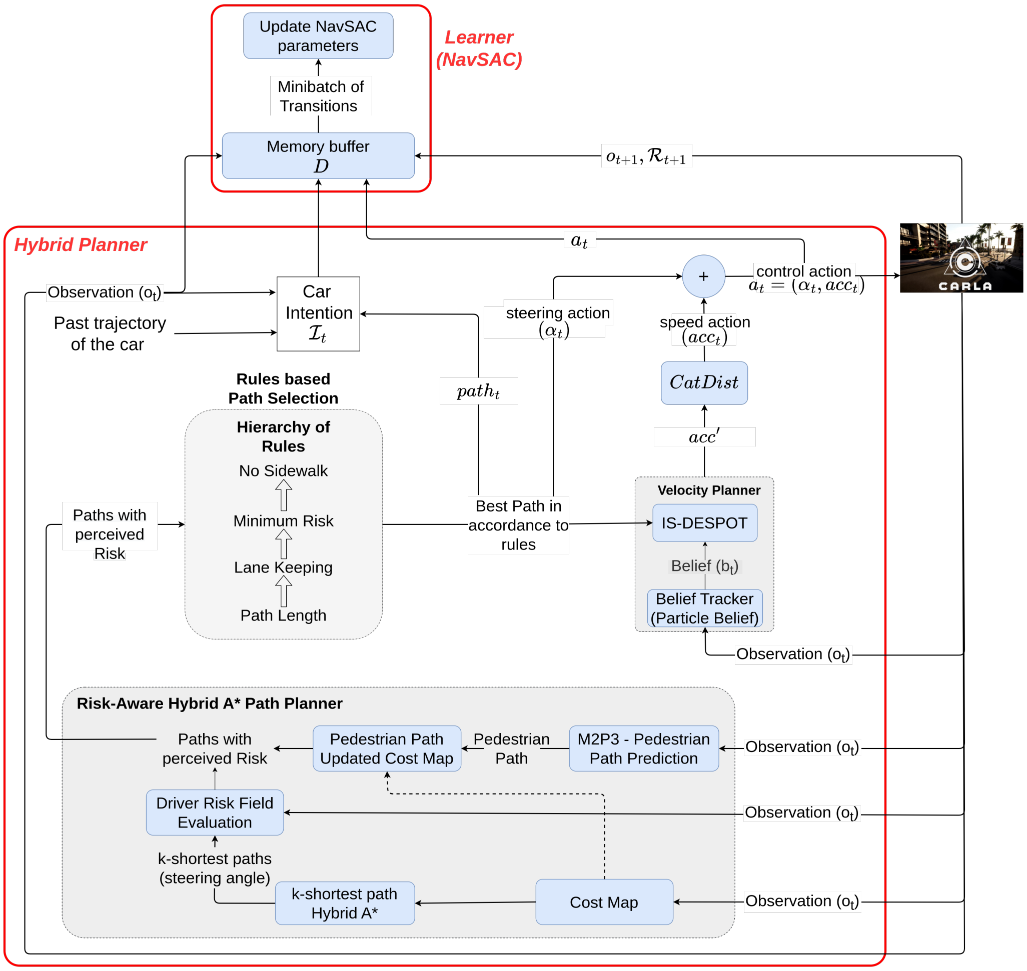
Update NavSAC parameters
Learner (NavSAC)
Minibatch of Transitions
Memory buffer D
$o_{t+1}, \mathcal{R}_{t+1}$
Hybrid Planner
a_t
CARLA
Observation (o_t)
Car Intention $\mathcal{I}_t$
+
control action $a_t = (\alpha_t, acc_t)$
Past trajectory of the car
steering action (α_t)
speed action (acc_t)
$CatDist$
Rules based Path Selection
$path_t$
acc'
Hierarchy of Rules
No Sidewalk
Minimum Risk
Lane Keeping
Path Length
Paths with perceived Risk
Best Path in accordance to rules
Velocity Planner
IS-DESPOT
Belief (b_t)
Belief Tracker (Particle Belief)
Observation (o_t)
Risk-Aware Hybrid A* Path Planner
Paths with perceived Risk
Pedestrian Path Updated Cost Map
Pedestrian Path
M2P3 - Pedestrian Path Prediction
Observation (o_t)
Driver Risk Field Evaluation
Observation (o_t)
k-shortest paths (steering angle)
k-shortest path Hybrid A*
Cost Map
Observation (o_t)

Fig. 3. HyLEAR training architecture.

More concrete, the risk-aware path planning module of the hybrid planner generates $k = 3$ short and safe paths based on the basic and two extended cost-maps.

For each of these paths, the human passenger-received risk is estimated by means of computing the respective driver risk field (DRF) as in [17]. The DRF represents the perceived belief of the passenger over the planned path of the car, and yields a DRF value for any grid location in the underlying cost-map. Eventually, the passenger-perceived risk of the planned path is estimated as respective sum product of DRF and cost-map values. Out of these paths associated with their risk values, the best driving rule-compliant path $path_t$ with minimum risk is selected by the hierarchical rule-based reasoning module of the hybrid planner, and the steering angle α_t gets obtained from it. Only in emergency cases, when none of the planned paths with acceptable passenger-perceived risk value below a given risk threshold does avoid driving on a sidewalk, this rule is violated and the one with minimum risk is selected. In case of multiple paths with acceptable risk, they are further filtered through all rules for final selection of the respectively best rule-compliant path with minimal risk.

For this selected path, the velocity planner (IS-DESPOT) then determines the speed action acc', which is used to initialize a categorical distribution $CatDist$ for the speed action set (Accelerate, Maintain, Decelerate; cf. Sect. 2.2). For example, if $acc' = Maintain$ then the distribution $CatDist$ is initialized with the probability vector $[\frac{1-P}{2}, P, \frac{1-P}{2}]$. The speed action acc_t is randomly sampled from the $CatDist$, and the full car control action $a_t = (\alpha_t, acc_t)$ gets executed in the driving simulator CARLA. After execution, the car intention and the respective POMDP belief state transition tuple, $(\mathcal{I}_t, o_t, a_t, \mathcal{R}_{t+1}, o_{t+1})$ is added to the memory buffer D of NavSAC.

For its off-policy learning of an approximate optimal speed action policy for given car intention in observed environment, NavSAC then randomly samples a set (mini-batch) of state transitions from this hybrid planning influenced memory buffer D, takes the car intention, and trains its network parameters using the above mentioned soft actor-critic loss functions. During this interposed learning, the action policy making knowledge of the hybrid planner is embedded into

Algorithm 1: HyLEAR Training

```
Number of scenes: $N \in \mathbb{R}_+$
Number of time steps per scene simulation: $T_{max} \in \mathbb{R}_+$
Transition buffer: $D \leftarrow \phi$
Regularization parameters: $\lambda_V, \lambda_Q, \lambda_\pi, \tau$
3-dim speed action probability vector $cpd$ for categorical speed
  action probability distribution $CatDist$,
speed action selection probability: $P = 0.8$
input  : Randomly initialized NavSAC network weights $\psi, \theta, \phi$
output: Trained NavSAC network weights $\psi, \theta, \phi$
Initialize $\overline{\psi} \leftarrow \psi$
for $scene \leftarrow$ 1 to $N$ do
    ego-car goal position for $scene$ : $p_{goal}^{scene}$
    $t \leftarrow 1$
    while $t \leq T_{max}$ do
        $\{path\}_t^k \leftarrow$ RiskAwarePathPlanner($o_t, p_{goal}^{scene}$)
        $path_t \leftarrow$ RuleBasedPathSelection($\{path\}_t^k$)
        $\alpha_t \leftarrow$ GetSteeringAngle($path_t$)
        $acc' \leftarrow$ VelocityPlanner($o_t, path_t$)
        $acc_t \sim_{rd} CatDist(cpd, P)$
        $a_t \leftarrow (\alpha_t, acc_t)$
        $o_{t+1} \leftarrow$ CARLA_execution($a_t$)
        $\mathcal{R}_{t+1} \leftarrow$ Reward($s_{t+1}, a_t$)
        $D \leftarrow D \cup \{(o_t, a_t, \mathcal{R}_{t+1}, o_{t+1})\}$
        if $t \ \% \ 4 = 0$ then
            Sample minibatch of transitions
              $\{(o_j, a_j, \mathcal{R}_{j+1}, o_{j+1})\}$ from $D$
            UpdateParameters($\{(o_j, a_j, \mathcal{R}_{j+1}, o_{j+1})\}$)
        end
        $t \leftarrow t + 1$
    end
end

function UpdateParameters($\{(o_j, a_j, \mathcal{R}_{j+1}, o_{j+1})\}$):
    input  : Minibatch of transitions $\{(o_j, a_j, \mathcal{R}_{j+1}, o_{j+1})\}$
             from $D$
    output: Updated network weights $\psi, \theta, \phi, \overline{\psi}$
    $\psi \leftarrow \psi - \lambda_V \hat{\nabla}_\psi J_V(\psi)$
    $\theta \leftarrow \theta - \lambda_Q \hat{\nabla}_\theta J_Q(\theta)$
    $\phi \leftarrow \phi - \lambda_\pi \hat{\nabla}_\phi J_\pi(\phi)$
    $\overline{\psi} \leftarrow \tau\psi + (1 - \tau)\overline{\psi}$ // $J_V, J_Q, J_\pi$
    return $\psi, \theta, \phi, \overline{\psi}$
```

the off-policy learner NavSAC of HyLEAR.[2] In particular, the hybrid planner guided exploration of the state space by NavSAC may allow for faster training. It allows to avoid cold start and unnecessary explorations caused by sampling transitions from D that may lead to a crash or near-miss of pedestrians. Such accidents can be anticipated and avoided by IS-DESPOT of the hybrid planner during its n-step look-ahead planning of optimal speed actions for next steering action on the given path.

C. Testing

As mentioned above, the testing architecture of HyLEAR (cf. Fig. 4) is the same as for training except that it does not include the computationally expensive planner IS-DESPOT to determine an optimal speed actions for given situation and path. This capability is now embedded in and performed much faster by the trained learner NavSAC.

During testing, the hybrid planner of HyLEAR generates the shortest safe path with minimal human-perceived risk and rule violations as input for the trained NavSAC such that the extracted steering angle together with the optimal speed action determined now by the trained NavSAC instead of the planner IS-DESPOT is then executed as a full control action by the car in the driving simulator CARLA.

IV. EVALUATION

A. Experimental Setting

For our experimental comparative performance evaluation of HyLEAR with selected baselines, we created the CARLA-CTS benchmark which consists of twelve parameterized scenarios with about thirty thousand synthesized scenes in total, simulated in the driving simulator CARLA 0.9.12. Most of the traffic scenarios are taken from the GIDAS accident study [2] where the car is confronted with street crossing pedestrian, possibly occluded by some parking car, an incoming car, and different street intersections (cf. Sect. 2.2, Fig. 1). The scenes per scenario are generated with varying speed and crossing distance of pedestrians from the car.

The selected baselines are the individual collision-free navigation action planning and learning methods IS-DESPOTp and NavSACp, as well as the socially-aware DRL method A2C-CADRLp [9] In addition, we take the deep reinforcement learning-assisted online POMDP planner HyLEAP [23] as a baseline for hybrid AI methods for collision-free navigation that differs fundamentally from the HyLEAR approach of hybrid planning-assisted deep reinforcement learning. Besides, all baselines are guided by a hybrid A* path planner as in [23], which differs from the risk-aware anytime hybrid A* k-path planner of HyLEAR. All methods were trained on the first nine scenarios and tested on all twelve scenarios of the CARLA-CTS benchmark following a 18:17:65% ratio of train, validation and test set.

The performance of each method is measured in terms of (a) the overall safety index (SI) defined as total number of scenarios in which the method is below given percentages of crashes (5) and near-misses (10); (b) the crash and near-miss rates (%), and time to goal (TTG) in seconds; (c) the ride comfort defined as being inversely proportional to the equally weighted sum of jerks and passenger-perceived risk of planned trajectory of car; with risk normalized to [0,1] and the risk threshold for rule-based selection of paths with acceptable risk has been set to 0.1 [17]; and (d) the training time in days and execution time in milliseconds.

HyLEAR is implemented in Python and PyTorch framework; training and testing was done on the DL supercomputer NVIDIA DGX-1 at DFKI Kaiserslautern.

B. Results

The overall results of our experiments, averaged across all scenarios, are shown in Table I. For each measure, we first average the scores across all scenes of a scenario and then average across all scenarios such that each scenario is weighted equally.

[2] The action selection probability value $P = 0.8$ was experimentally found to offer a good trade-off between exploitation of the hybrid planner action policy and exploration by the deep reinforcement learner NavSAC.

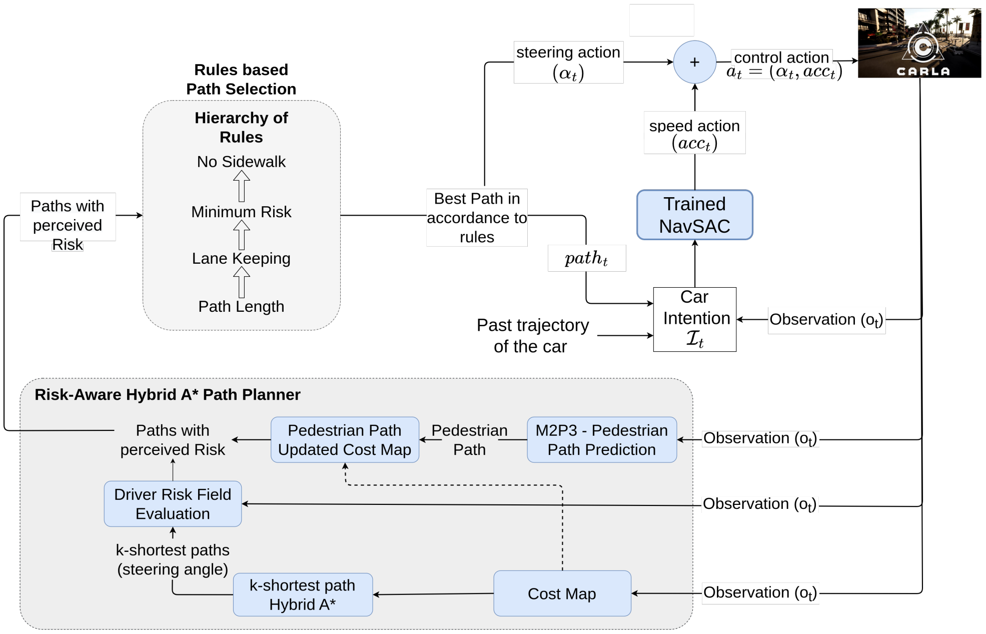
Rules based Path Selection
Hierarchy of Rules
No Sidewalk
Minimum Risk
Lane Keeping
Path Length
Paths with perceived Risk
Best Path in accordance to rules
steering action (α_t)
+
control action $a_t = (\alpha_t, acc_t)$
CARLA
speed action (acc_t)
Trained NavSAC
$path_t$
Car Intention $\mathcal{I}_t$
Past trajectory of the car
Observation (o_t)
Risk-Aware Hybrid A* Path Planner
Paths with perceived Risk
Pedestrian Path Updated Cost Map
Pedestrian Path
M2P3 - Pedestrian Path Prediction
Observation (o_t)
Driver Risk Field Evaluation
Observation (o_t)
k-shortest paths (steering angle)
k-shortest path Hybrid A*
Cost Map
Observation (o_t)

Fig. 4. HyLEAR testing architecture.

Method	Safety	Crash	Near-Miss	TTG	Comfort	Training	Execution
NavSACp	1	21.44	9.24	17.57	0.262	10	60.29
IS-DESPOTp	1	21.01	6.05	16.20	0.689	N/A	259.56
A2C-CADRLp	0	25.14	11.47	**14.26**	1.010	4	**58.57**
HyLEAP	4	**19.26**	**7.41**	16.16	0.803	5	215.80
HyLEAR	**5**	19.88	8.49	15.86	**1.064**	**3**	71.50

TABLE I

Overview of experimental results for HyLEAR and baselines on CARLA-CTS benchmark

In general, HyLEAR provided a safer and more comfortable ride than the other selected baselines, following comparatively the fastest training and with a relatively acceptable time to goal and execution time. In some cases, taking the safe and more comfortable but not shortest route by HyLEAR may come at the expense of minimal time to goal compared to the fastest method A2C-CADRL. The latter, however, performed worse on safety due to always accelerate policy to reach the goal faster, which resulted in second best comfort due to zero jerks in the calculation.

On average, the n-step look-ahead action planning with IS-DESPOT-p was as safe as the learner NavSACp and with more ride comfort, in particular in scenarios with temporarily occluded pedestrians but required extremely more execution time due to online planning than both DRL methods. Like HyLEAP, the IS-DESPOTp does not take risk-aware path planning into account but their n-step look-ahead planning supported some reduction of jerks, yielding lower average comfort than HyLEAR.

The interposed learning allowed HyLEAR to learn the optimal speed action for given situation and safe path faster than the other DRL methods and 20% faster compared to HyLEAP. Moreover, due to its hybrid planning HyLEAR performed best in terms of safety and ride comfort, only driving on safe paths through free sidewalks if there are no alternatives with acceptable human-perceived risk.

While both hybrid methods, HyLEAP and HyLEAR, are by far more safe than the tested individual planning and DRL methods, the hybrid planning-assisted learning of HyLEAR outperformed the DRL-assisted online planning of HyLEAP in safety, comfort, time to goal, training and execution time.

In scenarios 1, 2, 4, 5, where the pedestrian is visible at all times and is crossing the road from the opposite lane, HyLEAR has the lowest crash rate as it can even predict the future position of the pedestrian. This allows HyLEAR to either brake in time or consider navigating through the sidewalk to avoid crash when braking in time is not possible, for example, in scenes with high pedestrian speed and small crossing distance. While both HyLEAP and IS-DESPOTp construct a belief tree for observation including the pedestrian position, the data-driven pedestrian path predictor M2P3 in HyLEAR provides better estimates. NavSacp and A2C-CADRLp are not supported by any such pedestrian path predictor, hence perform comparitively poorly in these scenarios.

In scenario 9, where the ego-vehicle is approaching an inter-

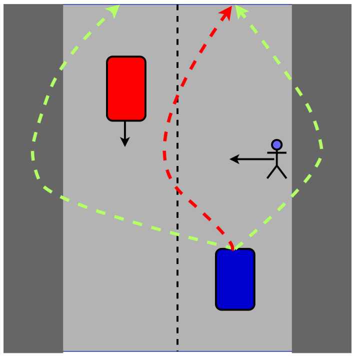

Fig. 5. Critical traffic scenario 10 with incoming car (in red) on the left lane and pedestrian crossing from right to left, together with three alternative safe and short paths (dotted lines) determined by risk-aware path planner of HyLEAR car (in blue) at some time step. The red path is technically safe, rule-compliant and shortest but perceived by the passenger of the self-driving car as of unacceptable risk; the rule-violating path via the sidewalk in opposite to walking direction of pedestrian is selected.

section and a pedestrian is crossing at the intersection, the online POMDP planner IS-DESPOT-p with its n-step look-ahead planning comparatively performed best together with HyLEAR. The alternative pure learning methods, aiming to reach the goal faster, mostly do not sufficiently reduce the car speed when approaching the intersection, which often led to collisions with buildings along the street at the intersection when turning.

In scenario 10, the ego-vehicle is driving on the right lane, an incoming car drives on the left lane, and a pedestrian is crossing the street straight from right to left. All scenes of this scenario are parameterized such that braking in time (before hitting the pedestrian) is not possible for the self-driving car with minimum speed set to 20km/h. In general, due to their limited path planning capabilities compared to HyLEAR, the baselines A2C-CADRLp, NavSACp, IS-DESPOTp and HyLEAP did not navigate via the sidewalk in this emergency case, which resulted in relatively high crash rates for each of them. HyLEAR was able to better generalise for this previously unseen scenario on the basis of scenarios 7 and 8. Through utilizing its risk-aware path planning of alternative safe trajectories including those via sidewalks (cf. Fig. 5) and driving-behaviour rule-based reasoning, it achieved a significantly reduced crash rate and, at the same time, acceptable passenger comfort compared to the baselines.

In scenario 11, the left lane is blocked by parked cars such that only the right lane is available for navigation in both directions such as in a narrow one-way street. All tested methods decided to navigate via the free left sidewalk to avoid collision with incoming car and parked cars. However, unlike HyLEAR, the neural models learned to make the exception a habit and continued to drive partly on this restricted area until they reached the goal position fast but with low comfort (path risk value based on the basic cost-map, cf. Sect. 3.2) and less rule-compliant. HyLEAR was able to navigate back onto the street to make its trajectory more rule-compliant and comfortable as trained during its interposed learning with the hybrid planner.

In scenario 12, where a parked car is blocking the right lane and there is an incoming car on the left lane, the selected baseline methods navigated through the gap between the parked and the incoming car, though mostly resulting in a collision with the latter. This is mainly due to their lack of an integrated multi-object path predictor, unlike in HyLEAR.

V. RELATED WORK

As mentioned above, there are various DRL and online POMDP planning methods for CFN in POMDPs. For example, the hybrid learning method RLfD [20] combines RL and imitation learning through the combining of samples from expert demonstrations with DRL exploration in order to improve data efficiency. GA3C-CADRL [9] and PPO [26] are end-to-end on-policy actor-critic, respectively, policy-based CFN methods, while SA-CADRL [6] also codifies some social norms of driving behavior in the reward function. While the methods of intention-aware online POMDP planning for CFN in [1], [21] account for pedestrians and vehicles in the traffic environment, the CFN planning in [30] for multi-lane intersection scenarios does not do so for pedestrians. However, none of the above methods were evaluated for safety and comfort in synthesized car-pedestrian accident scenarios based on real-world accident studies like GIDAS. The hybrid DRL-assisted online POMDP planning method HyLEAP [23] may be safer than purely DRL or POMDP planning methods for CFN but it suffers from extreme training times and no real-time POMDP planning. To the best of our knowledge, there is no planning-assisted DRL method for CFN available yet.

In our context, rule-based reasoning in DRL may be required to guarantee the safety of DRL policy at all times and to ensure DRL policy is rule compliant. There are various approaches to constrained (deep) reinforcement learning with rules. For example, in [25] rule-based reasoning is combined with DRL by modifying the reward function with traffic rules, while in [11] this is achieved by behaviour-based reasoning with specifications in linear time logic on finite traces. The rule-interposed learning framework presented in [29] embeds high level rules into DRL by sampling rule-compliant actions to train the learner. While these methods follow rules when possible, applied to the CFN problem they cannot guarantee safety and rule compliance at all times. [5] introduced a modular hierarchy of driving behavior rules (*rulebook*) such that a set of safe navigation actions can be determined on the go. However, there is no method that combines it with rule-interposed learning for guaranteed rule compliance.

According to [14] comfort can be broadly defined 'as a state of well-being, ease and physical and psychological harmony between a person and the environment'. As to passenger comfort in autonomous cars [8], there is no common agreement on an objective measurement of it yet, even not in general [27], [3], but on comfort related factors of road and load disturbance. The latter class is concerned with smoothness and apparent safety or perceived risk of driving, which we take to measure passenger or ride comfort as inversely proportional to the equally weighted sum of jerks and passenger-perceived risk of the planned car path. To the best of our knowledge, there is no CFN method for self-driving car that takes ride comfort into account without compromising safety.

VI. Conclusion

We presented HyLEAR, the first hybrid planning-assisted deep reinforcement learning method for collision-free driving policies for self-driving cars that also take into account, whenever possible, the ride comfort and a given set of driving-behavior rules. The experimental results over the CARLA-CTS benchmark revealed that HyLEAR can outperform the selected baselines in safety and ride comfort with faster training and acceptable execution time. Ongoing work on improving HyLEAR is concerned with, for example, the adding of predicted paths of other cars in simulated scenes, the addressing of car perception with (sensor) noise, the testing of additional recent deep reinforcement learners, and parallel k-path planning of the hybrid planner.

Acknowledgement

This work has been funded by the German Ministry for Education and Research (BMB+F) in the project MOMENTUM.

References

[1] Bai, H. et al. (2015): Intention-aware online POMDP planning for autonomous driving in a crowd. Proc. IEEE Intern. Conference on robotics and Automation (ICRA). IEEE.

[2] Bartels, B. & Liers, H. (2014): Bewegungsverhalten von Fussgaengern im Strassenverkehr, Teil 2. FAT-Schriftenreihe, Nr. 268.

[3] Bellem, H. et al. (2018): Comfort in automated driving: An analysis of preferences for different automated driving styles and their dependence on personality traits. *Transp. research part F: traffic psychology & behaviour*, 55

[4] Bougie, N.; Cheng, L.K. & Ichise, R. (2018): Combining deep reinforcement learning with prior knowledge and reasoning. *Applied Computing Review*, 18(2), ACM.

[5] Censi, A. et al. (2019): Liability, ethics, and culture-aware behavior specification using rulebooks. Proc. International Conference on Robotics and Automation (ICRA). IEEE.

[6] Chen, Y.F., Everett, M., Liu, M., & How, J.P. (2017): Socially aware motion planning with deep reinforcement learning. Proc. IEEE Intern. Conf. on Intelligent Robots and Systems (IROS).

[7] Chen, J.; Yuan, B. & Tomizuka, M. (2019): Model-free deep reinforcement learning for urban autonomous driving. Proc. IEEE intelligent Transportation Systems Conference (ITSC). IEEE.

[8] Elbanhawi, M., Simic, M., & Jazar, R. (2015): In the passenger seat: investigating ride comfort measures in autonomous cars. *IEEE Intelligent transportation systems magazine*, 7(3).

[9] Everett, M.; Chen, Y.F. & How, J.P. (2021): Collision avoidance in pedestrian-rich environments with deep reinforcement learning. *IEEE Access*, 9:10357–10377. IEEE.

[10] Gao, W. et al. (2017): Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. Proc. International Conference on Robot Learning.

[11] Giuseppe, D. et al.(2019): Foundations for Restraining Bolts: Reinforcement Learning with LTLf/LDLf Restraining Specifications. Proc. International Conference on Automated Planning and Scheduling (ICAPS).

[12] Gupta, D. (2022): CARLA-CTS: Synthetic benchmark of critical and non-critical traffic scenarios in driving simulator CARLA; source code of HyLEAR and baseline methods included. Available at https://github.com/dikshant2210/Carla-CTS02.

[13] Haarnoja, T. et al. (2018): Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. Proc. International Conference on Machine Learning (ICML).

[14] Hartwich, F., Beggiato, M., & Krems, J.F. (2018): Driving comfort, enjoyment and acceptance of automated driving–effects of drivers' age and driving style familiarity. *Ergonomics*, 61(8).

[15] Kamran, D. et al. (2020): Risk-aware high-level decisions for automated driving at occluded intersections with reinforcement learning. Proc. IEEE Intelligent Vehicles Symposium (IV). IEEE.

[16] Kiran, B.R. et al. (2021): Deep reinforcement learning for autonomous driving: A survey. *IEEE Transactions on Intelligent Transportation Systems*.

[17] Kolekar, S.; de Winter,J. & Abbink, D. (2020): Human-like driving behaviour emerges from a risk-based driver model. *Nature Communications*, 11(1).

[18] Kong, J. et al. (2015): Kinematic and dynamic vehicle models for autonomous driving control design. Proc. IEEE Intelligent Vehicles Symposium (IV). IEEE.

[19] Liang, X. et al. (2018): Cirl: Controllable imitative reinforcement learning for vision-based self-driving. Proc. European Conference on Computer Vision (ECCV).

[20] Liu, H. et al. (2022): Improved deep reinforcement learning with expert demonstrations for urban autonomous driving. Proc. IEEE Intelligent Vehicles Symposium (IV). IEEE.

[21] Luo, Y. et al. (2019): Importance sampling for online planning under uncertainty. *Robotics Research*, 38(2-3).

[22] Poibrenski, A. et al. (2020): Multimodal multi-pedestrian path prediction for autonomous cars. *Applied Computing Review*, 20(4), ACM.

[23] Pusse, F. & Klusch, M. (2019): Hybrid online POMDP planning and deep reinforcement learning for safer self-driving cars. Proc. IEEE Intelligent Vehicles Symposium (IV). IEEE.

[24] Skoglund, C. (2021): Risk-aware autonomous driving using POMDPs and responsibility-sensitive safety. Master Thesis, KTH Royal Institute of Technology, Division of Decision and Control Systems, Stockholm (SE)

[25] Talamini, J. et al.(2020): On the impact of the rules on autonomous drive learning. *Applied Sciences*

[26] Tang, Y. (2019): Towards learning multi-agent negotiations via self-play. Proc. IEEE Intern. Conf. on Computer Vision CVF.

[27] Woolridge, E. & Chan-Pensley, J. (2020): Measuring user's comfort in autonomous vehicles. In: Human Drive Project (https://humandrive.co.uk/)

[28] Yang, F. et al. (2018): Peorl: Integrating symbolic planning and hierarchical reinforcement learning for robust decision-making. arXiv preprint arXiv:1804.07779

[29] Zhang, H. et al. (2019): Faster and safer training by embedding high-level knowledge into deep reinforcement learning. arXiv preprint arXiv:1910.09986

[30] Zhou, B. et al. (2018): Joint multi-policy behavior estimation and receding-horizon trajectory planning for automated urban driving. Proc. IEEE Intern. Conf. on Robotics and Automation (ICRA).

[31] Zhu, Z., & Zhao, H. (2021): A survey of deep RL and IL for autonomous driving policy learning. *IEEE Transactions on Intelligent Transportation Systems*.