AuE-8930
Computing and Simulation for Autonomy

Capstone Project

A Scalable and Parallelizable Multi-Agent Reinforcement Learning Framework for Cooperative and Competitive Autonomous Vehicles

Chinmay Samak
PhD Candidate, CU-ICAR
csamak@clemson.edu

Tanmay Samak
PhD Candidate, CU-ICAR
tsamak@clemson.edu

Project Management

Scalable and Parallelizable Multi-Agent Reinforcement Learning for AVs

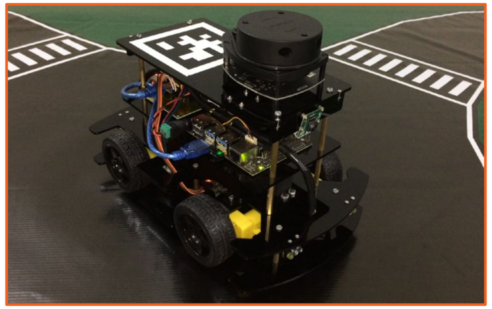
Planned Project Objectives

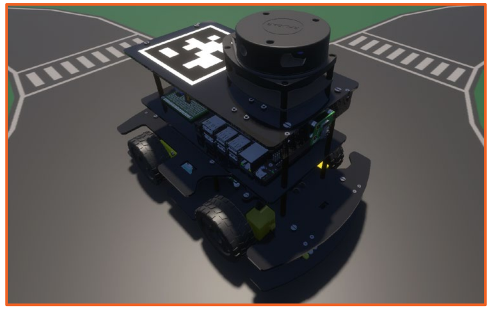
- Set up high-fidelity 3D simulation platform based on real-world vehicle/environment representations (real2sim transfer) [Tanmay]
- Set up modular and scalable simulation framework using object oriented programming (OOP) [Chinmay]
- Set up intelligent agent/environment parallelization framework for accelerating RL training [Tanmay]
- Formulate multi-agent reinforcement learning (MARL) problems for: [Tanmay]
  - Cooperative MARL - collaborative intersection traversal
  - Competitive MARL - head-to-head autonomous racing
- Implement the formulated deep reinforcement learning (DRL) pipeline and conduct parallelized training [Chinmay]
- Deploy and analyze the trained policies and procedures to comment on the aspects of “computing and simulation for autonomy” [Chinmay]

Note: *The name in square bracket indicates primary responsibility and NOT necessarily contribution. Both members worked together and contributed equally to this project.*

AutoDRIVE

F1 TENTH





Planned Project Timeline

AuE-8930 Capstone Project (Fall 2023)

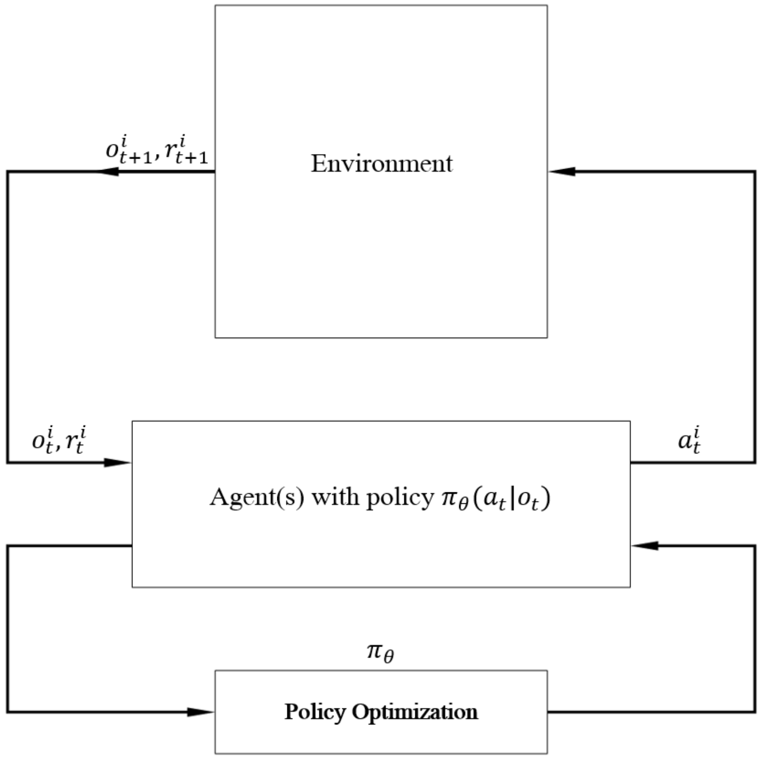
Task	10/17	10/24	10/31	11/07	11/14	11/21	11/28	12/05
Announcements								
Project Proposal								
Phase 1								
Phase 2								
Phase 3								
Project Presentation								
Project Report								

Research Motivation

Scalable and Parallelizable Multi-Agent Reinforcement Learning for AVs

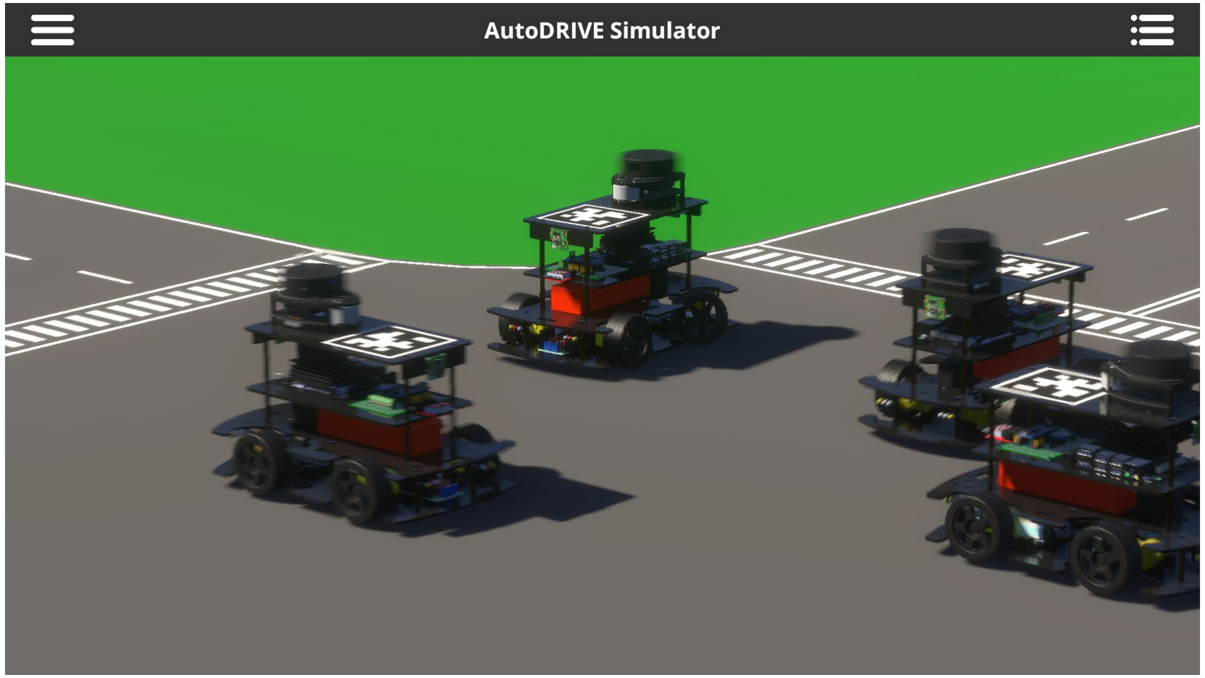
Multi-Agent Reinforcement Learning

- Reinforcement learning
  - Learning through experience
- Multi-agent reinforcement learning
  - Complex and dynamic interactions
  - Cooperative and competitive scenarios
- Autonomous vehicles
  - **Cooperation:** autonomous driving
  - **Competition:** autonomous racing
  - **Coopetition:** cooperation and competition



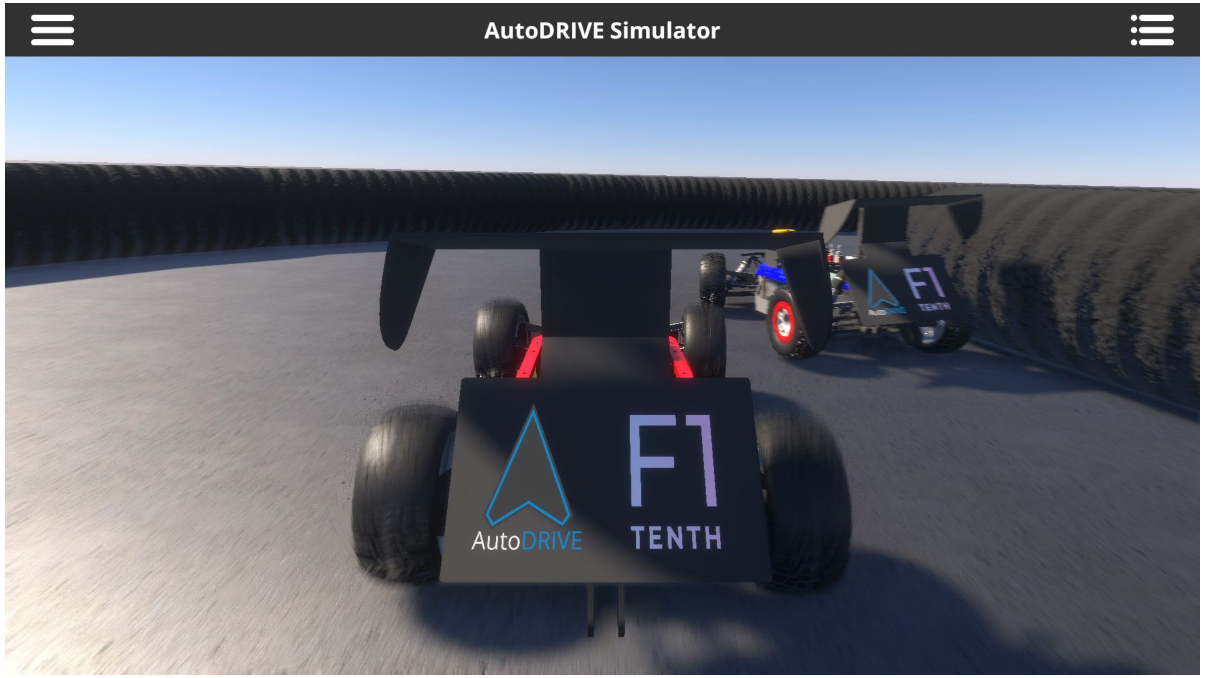
Multi-Agent Reinforcement Learning

- Reinforcement learning
  - Learning through experience
- Multi-agent reinforcement learning
  - Complex and dynamic interactions
  - Cooperative and competitive scenarios
- Autonomous vehicles
  - **Cooperation:** autonomous driving
  - **Competition:** autonomous racing
  - **Coopetition:** cooperation and competition



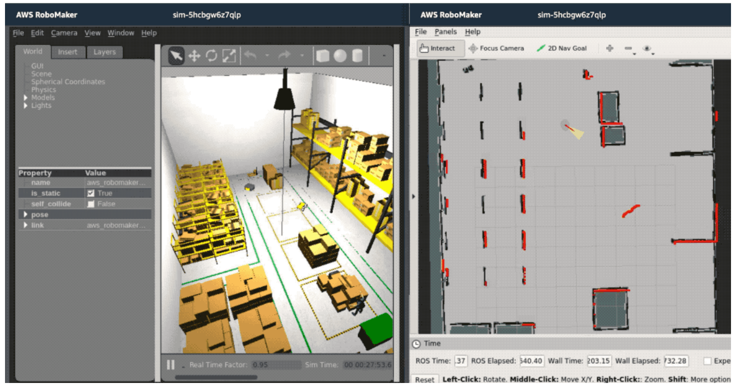
Multi-Agent Reinforcement Learning

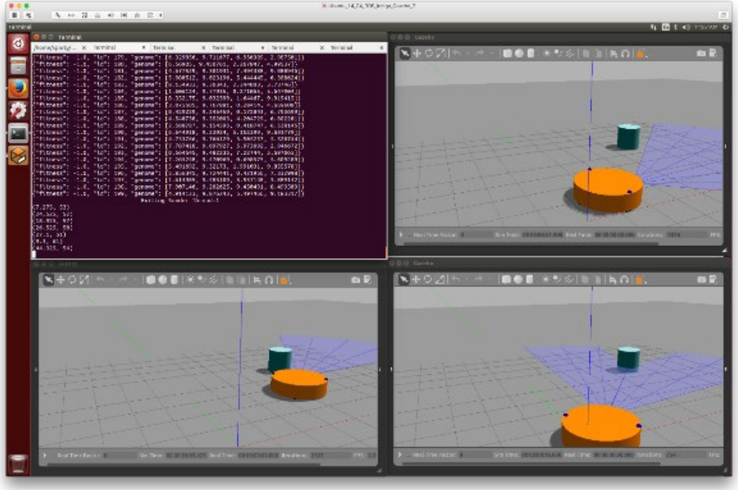
- Reinforcement learning
  - Learning through experience
- Multi-agent reinforcement learning
  - Complex and dynamic interactions
  - Cooperative and competitive scenarios
- Autonomous vehicles
  - **Cooperation:** autonomous driving
  - **Competition:** autonomous racing
  - **Coopetition:** cooperation and competition

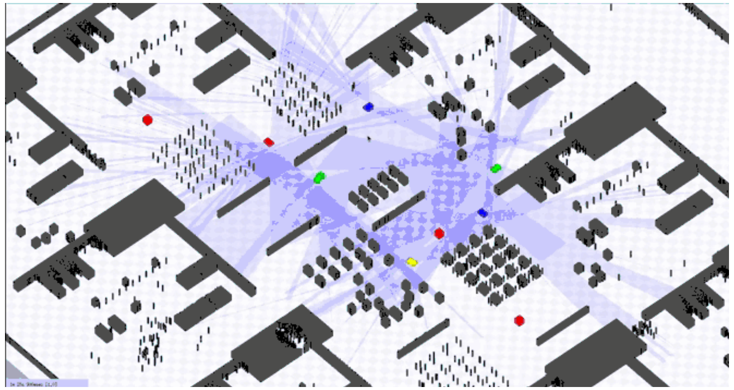


Simulation Parallelization

- Existing approaches
  - Multiple simulation instances
  - Multiple robots in single simulation instance
- Challenges
  - Unnecessary computational overhead
  - Exteroceptive perception modalities
- Ideal approaches
  - Parallelize only what is required
  - Environment parallelization
  - Agent parallelization







Literature Survey

Scalable and Parallelizable Multi-Agent Reinforcement Learning for AVs

Comparative Analysis of Hardware Platforms

Platform/Ecosystem				Cost *	Sensing Modalities							Computational Resources		Actuation Mechanism		Dedicated Simulator	V2X Support			API Support				
	Scale	Open Hardware	Open Software		*Throttle*	*Steering*	*Wheel Encoders*	*GPS/IPS*	*IMU*	*LIDAR*	*Camera*	*High-Level*	*Low-Level*	*Ackermann Steered*	*Differential-Drive/ Skid-Steered*		Multi-Agent Support	*V2V*	*V2I*	*C++*	*Python*	*ROS*	*MATLAB/Simulink*	*Webapp*
AutoDRIVE	1:14	✓	✓	$450	✓	✓	✓	✓	✓	✓	✓	Jetson Nano	Arduino Nano	✓	★	✓	✓	✓	✓	✓	✓	✓	★	✓
MIT Racecar	1:10	★	✓	$2600	✗	✗	✗	✗	✓	✓	✓	Jetson TX2	VESC	✓	✗	Gazebo	★	★	✗	✗	✗	✓	✗	✗
AutoRally	1:5	★	✓	$23,300	✗	✗	✓	✓	✓	✓	✓	Custom	Teensy LC/Arduino Micro	✓	✗	Gazebo	★	★	✗	✗	✗	✓	✗	✗
F1TENTH	1:10	★	✓	$3260	✗	✗	✗	✗	✗	✓	✗	Jetson TX2	VESC 6MkV	✓	✗	RViz/Gazebo	✓	✓	✗	✗	✗	✓	✗	✗
DSV	1:10	★	✓	$1000	✗	✗	✓	✗	✓	✓	✓	ODROID-XU4	Arduino (Mega + Uno)	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
MuSHR	1:10	★	✓	$930	✗	✗	✗	✗	✗	✗	✓	Jetson Nano	Turnigy SK8-ESC	✓	✗	RViz	✓	✓	✗	✗	✗	✓	✗	✗
HyphaROS RaceCar	1:10	★	✓	$600	✗	✗	✗	✗	✓	✓	✗	ODROID-XU4	RC ESC TBLE-02S	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
Donkey Car	1:16	★	✓	$370	✗	✗	✗	✗	✗	✗	✓	Raspberry Pi	ESC	✓	✗	Gym	✗	✗	✗	✗	✓	✗	✗	✗
BARC	1:10	★	✓	$1030	✗	✗	✓	✗	✓	✗	✓	ODROID-XU4	Arduino Nano	✓	✗	✗	✗	✗	✗	✗	✗	✓	✗	✗
OCRA	1:43	★	✓	$960	✗	✗	✗	✗	✓	✗	✗	None	ARM Cortex M4 μC	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗
QCar	1:10	✗	✗	$20,000	✗	✗	✓	✗	✓	✓	✓	Jetson TX2	Proprietary	✓	✗	Simulink	✓	✓	✗	★	★	★	✓	✗
AWS DeepRacer	1:18	✗	✗	$400	✗	✗	✗	✗	✓	★	✓	Proprietary	Proprietary	✓	✗	Gym	✗	✗	✗	✗	✗	✗	✗	✓
Duckietown	N/A	✓	✓	$450	✗	✗	★	✗	★	✗	✓	Raspberry Pi/Jetson Nano	None	✗	✓	Gym	✓	✗	★	✗	✗	✓	✗	✗
TurtleBot3	N/A	✓	✓	$590	✗	✗	✓	✗	✓	✓	★	Raspberry Pi	OpenCR	✗	✓	Gazebo	★	★	✗	✗	✗	✓	✗	✗
Pheeno	N/A	✓	✓	$350	✗	✗	✓	✗	✓	✗	✓	Raspberry Pi	Arduino Pro Mini	✗	✓	✗	✓	✓	✗	✗	✓	★	✗	✗

✓ indicates complete fulfillment; ★ indicates conditional, unsupported or partial fulfillment; and ✗ indicates non-fulfillment. * All cost values are ceiled to the nearest $10.

T. Samak, C. Samak, S. Kandhasamy, V. Krovi, and M. Xie, "AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education," Robotics, vol. 12, no. 3, p. 77, May 2023, doi: https://doi.org/10.3390/robotics12030077

Comparative Analysis of Simulation Platforms

Simulator	Year	Open Source	Realistic Perception	Customized Scenario	Back-end	Map Source		API Support		
						Real World	Human Design	Python	C++	ROS
TORCE [178]	2000	✓	✓	×	None	×	✓	×	✓	×
Webots [179]	2004	✓	✓	✓	ODE	✓	✓	✓	✓	✓
CarRacing [180]	2016	✓	×	×	None	×	✓	✓	×	×
CARLA [142]	2017	✓	✓	✓	UE4	×	✓	✓	✓	✓
SimMobilityST [181]	2017	✓	×	✓	None	×	✓	✓	×	×
GTA-V [156]	2017	×	✓	✓	RAGE	×	×	×	×	×
highway-env [182]	2018	✓	×	✓	None	×	✓	✓	×	×
Deepdrive [183]	2018	✓	✓	✓	UE4	×	✓	✓	✓	×
esmini [184]	2018	✓	✓	✓	Unity	×	✓	✓	✓	×
AutonoViSim [185]	2018	×	✓	✓	PhysX	×	✓	×	×	×
AirSim [186]	2018	✓	✓	✓	UE4	×	✓	✓	✓	✓
SUMO [187]	2018	✓	×	✓	None	✓	✓	✓	✓	×
Apollo [188]	2018	✓	×	✓	Unity	×	✓	✓	✓	×
Sim4CV [189]	2018	✓	✓	✓	UE4	×	✓	✓	✓	×
SUMMIT [72]	2020	✓	✓	×	UE4	✓	✓	✓	×	✓
MultiCarRacing [190]	2020	✓	×	×	None	×	✓	✓	×	×
SMARTS [80]	2020	✓	×	✓	None	×	✓	✓	×	×
LGSVL [191]	2020	✓	✓	✓	Unity	✓	✓	✓	×	✓
CausalCity [77]	2021	✓	✓	✓	UE4	×	✓	✓	×	×
MetaDrive [74]	2021	✓	✓	✓	Panda3D	✓	✓	✓	×	×
L2R [192]	2021	✓	✓	✓	UE4	✓	✓	✓	×	×
AutoDRIVE [193]	2021	✓	✓	✓	Unity	×	✓	✓	✓	✓

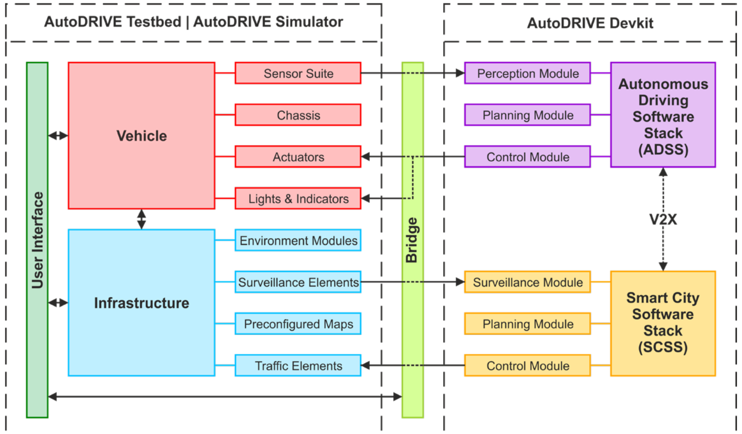
W. Ding, C. Xu, M. Arief, H. Lin, B. Li and D. Zhao, “A Survey on Safety-Critical Driving Scenario Generation—A Methodological Perspective,” in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 7, pp. 6971-6988, July 2023, doi: 10.1109/TITS.2023.3259322

AutoDRIVE Ecosystem

Source: https://autodrive-ecosystem.github.io

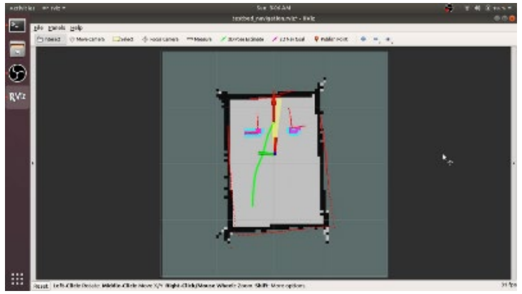
- Develop autonomy algorithms using flexible APIs
- Simulate for initial prototyping and corner-case analysis
- Deploy on mechatronic testbed for real-world validation
- Small, mid and full-scale vehicles and infrastructure
- Autonomous driving + smart-city applications
- Affordable cost and completely open-source

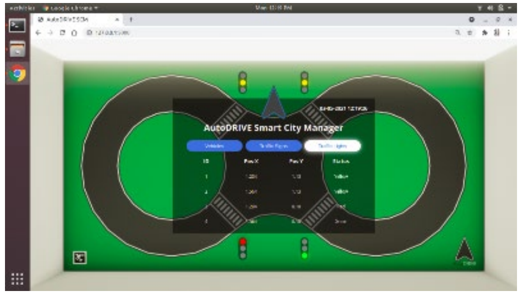
- Resources:
  - Website
  - Paper
  - GitHub
  - YouTube



Vehicle	Mobile Base	Battery	Computer(s)	Sensor(s)	Actuator(s)	Developer	Release	Cost
Nigel	Open-Source	5200 mAh	Jetson Nano or Jetson Orin Nano	RPLIDAR A-1, Pi-Cameras, Intel RealSense D435i, 9-Axis IMU, 6-Axis IPS, Encoders, Microphone, Steering Feedback, Throttle Feedback	DC Motors, Steering Servo(s)	CU-ICAR + NTU + SRMIST	2021	$450

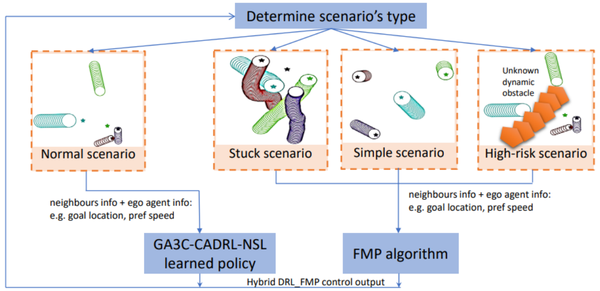
Simulation Quality	Physics Engine	Graphics Rendering	Vehicle Dynamics Support	Sensor Support	API Support	Developer	Cost	Open Source	Applications
3D	PhysX	Unity HDRP	Full car model for lateral, longitudinal, vertical and RPY dynamics with tire-terrain interaction	2D/3D LIDAR, Camera, GNSS, IPS, IMU, Encoders Steering Feedback, Throttle Feedback, State Variables	ROS, ROS 2, Python, C++, MATLAB, Simulink, Webapp	CU-ICAR, NTU, SRMIST	Free	Yes	Exploration, education and research





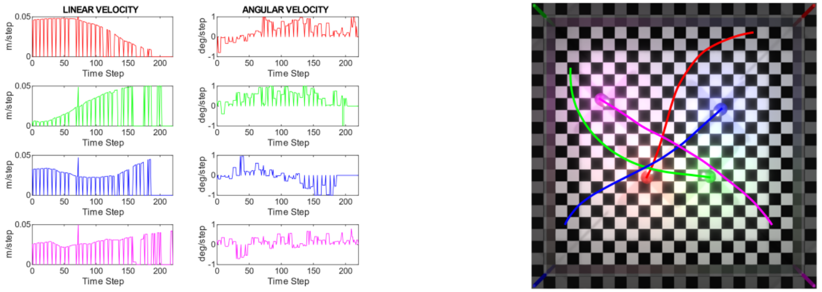
State-of-the-Art: Cooperative DRL

Article/Author	Methodology	Tools/Frameworks	Observations	Actions	Application	Summary	Year
Semnani et al. [3]	FMP+DRL	Python Simulation	$\boldsymbol{s}_t^o: \{\boldsymbol{p}, \boldsymbol{v}, r\}$ $\boldsymbol{s}_t^h: \{\boldsymbol{p}_g, v_{pref}, \psi, \phi\}$	$a_t: \{v_t, \psi_t, \phi_t\}$ for differential drive robots (2D/3D)	Motion planning	Hybrid algorithm of deep reinforcement learning (RL) and force-based motion planning (FMP) to solve distributed motion planning problem in dense and dynamic environments.	2020
Long et al. [4]	DRL	Stage Simulator	$\boldsymbol{o}^t: \{\boldsymbol{o}_z^t, \boldsymbol{o}_g^t, \boldsymbol{o}_v^t\}$	$a_t: \{v_t, \omega_t\}$ for differential drive robots	Collision avoidance	End-to-end decentralized sensor-level collision avoidance policy for multi-robot systems.	2018
Aradi et al. [5]	DRL	N/A	N/A	N/A	Motion planning	Survey on hierarchical motion planning using DRL.	2020
Wang et al. [6]	DRL	Custom	$\boldsymbol{s}_i^j: \{f_{i-3}^j, f_{i-2}^j, f_{i-1}^j, f_i^j\}$	$a_t: \{v_t, \omega_t\}$ for differential drive robots	Motion planning	End-to-end method to train directly from each robot-centered, relative perspective generated image, and each robot's reward as the input.	2020
Zhou et al. [7]	DRL	Python Simulation	$\boldsymbol{s}_i^j: \{x_i^j, y_i^j, \phi_i^j\}$	$a_t: \phi_t \in [-60°: 10°: 60°]$ for USVs	Formation and collision avoidance	DRL for USV formation path planning with specific focus on a reliable obstacle avoidance in constrained maritime environments.	2020
Sivanathan et al. [8]	DRL	MARL Simulator	$\boldsymbol{o}_t^i: \{\boldsymbol{p}_t^i, \boldsymbol{g}_t^i, \tilde{\boldsymbol{p}}_t^i\}$	$a_t: \{v_t, \omega_t\}$ for differential drive robots	Motion planning	Decentralized motion planning framework for addressing the task of multi-robot navigation using deep reinforcement learning.	2020



DRL-FMP hybrid control framework [3]

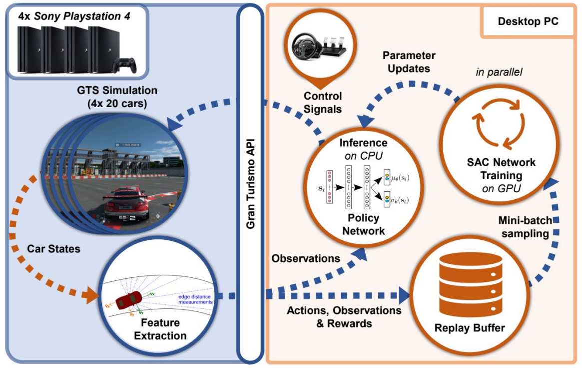
Collision avoidance using DRL [4]



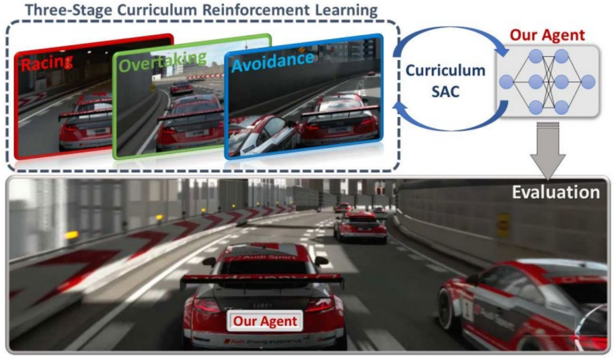
Decentralized motion planning using DRL [8]

State-of-the-Art: Competitive DRL

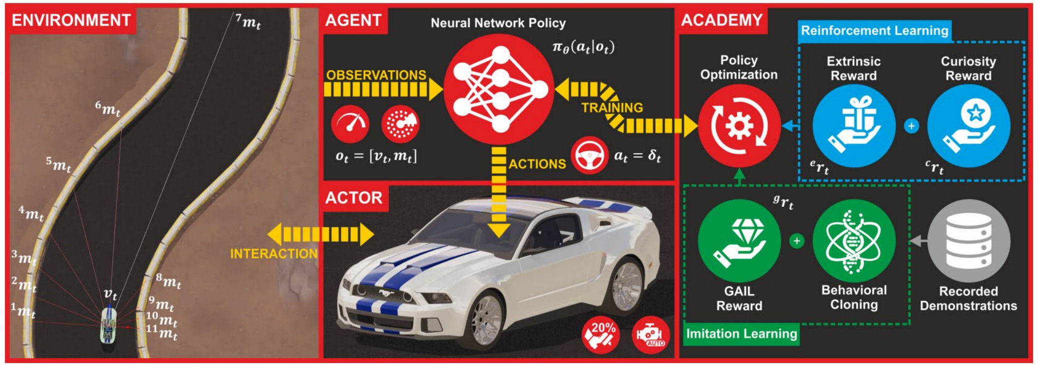
Article/Author	Methodology	Tools/Frameworks	Observations	Actions	Application	Summary	Year
Fuchs et al. [9]	DRL	Gran Turismo Sport	$\boldsymbol{s_t^o}: \{\boldsymbol{p}, \boldsymbol{v}, r\}$ $\boldsymbol{s_t^h}: \{\boldsymbol{p}_g, v_{pref}, \psi, \phi\}$	$a_t: \{\delta_t, \omega_t\}$ for full-scale Ackermann steered vehicle	Autonomous racing	Learning-based system for autonomous car racing by leveraging a course-progress proxy reward and deep reinforcement learning.	2020
Song et al. [10]	DRL	Gran Turismo Sport	$\boldsymbol{o_t}: \{v_t, \dot{v}_t, \boldsymbol{d_t}, \delta_{t-1}, f_t, f_c, \boldsymbol{c_L}\}$	$a_t: \{\delta_t, \omega_t\}$ for full-scale Ackermann steered vehicle	Autonomous overtaking	Curriculum-learning-based method to tackle the autonomous overtaking problem using DRL.	2021
Samak et al. [11]	Hybrid IL+RL	AutoRACE Simulator	$\boldsymbol{o_t}: \{\boldsymbol{m_t}, v_t\}$	$a_t: \{\delta_t\}$ for full-scale Ackermann steered vehicle	Autonomous racing	Hybrid imitation-reinforcement learning architecture to train a rigorous end-to-end control strategy for autonomous vehicles aimed at minimizing lap times in a time attack racing event.	2020
Betz et al. [12]	Mixed	N/A	N/A	N/A	Autonomous racing	Survey on autonomous vehicle racing.	2022



DRL-FMP hybrid control framework [9]



Autonomous overtaking using DRL [10]



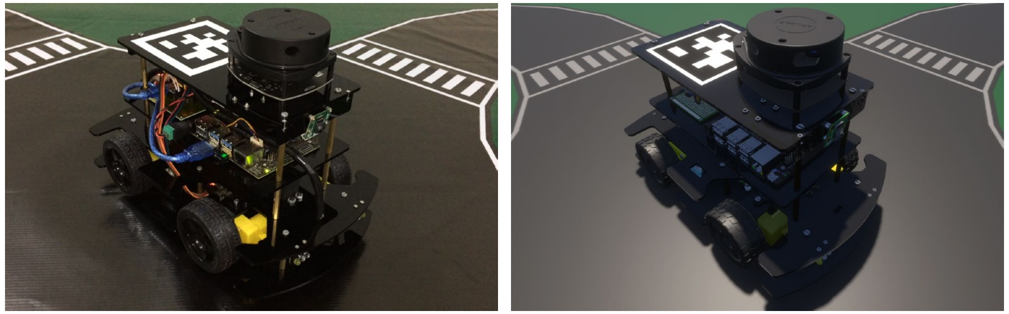
Hybrid IL+RL architecture for autonomous racing [11]

Digital Twin Creation

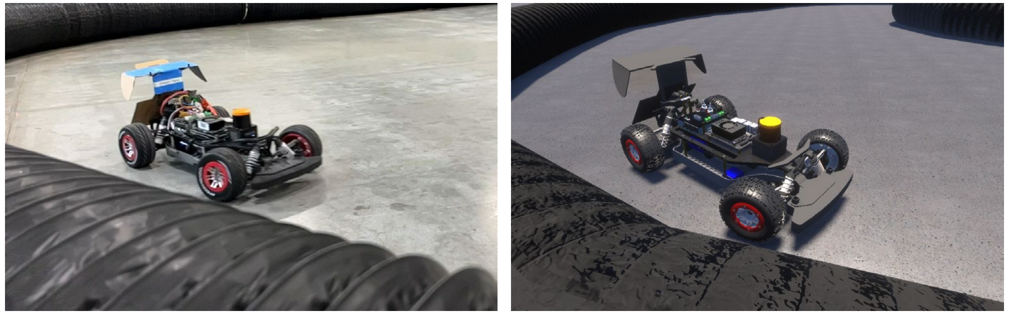
Physically and Graphically Accurate Digital Twins for Sim2Real-Worthy RL Training

Digital Twin Capabilities of AutoDRIVE Ecosystem

SMALL-SCALE



Nigel (Native Vehicle)



F1TENTH (1/10th Scale Racecar)

MID-SCALE

Husky (On/Off-Road Skid-Steer Robot)

Hunter SE (On/Off-Road Ackermann Steered Vehicle)

FULL-SCALE

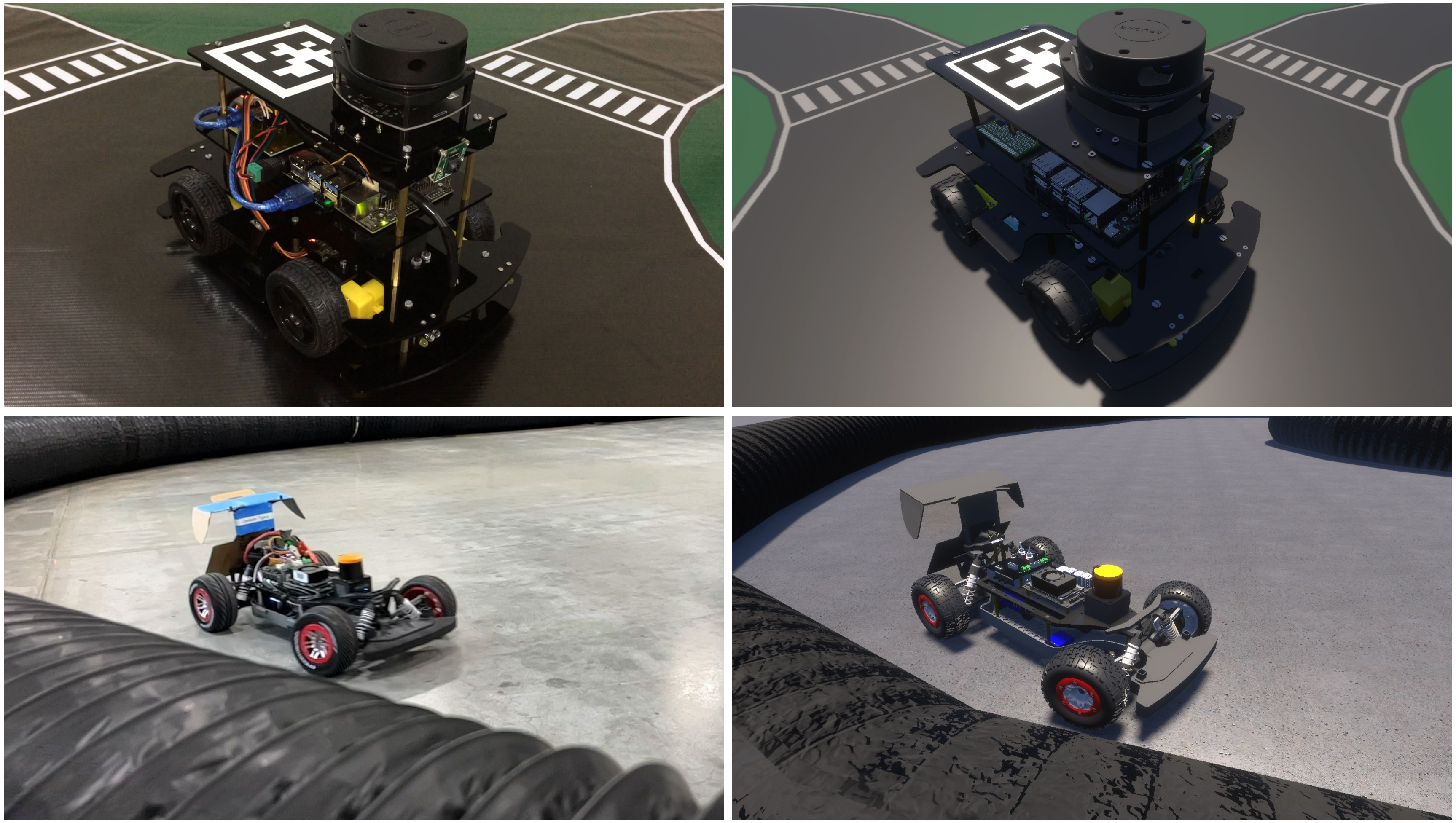
OpenCAV (On-Road Commercial Vehicle)

RZR (Recreational Off-Highway Vehicle)

Source: AutoDRIVE Ecosystem

Digital Twins of Nigel and F1TENTH

- Nigel
  - 1:14 scale
  - Autonomous driving
- F1TENTH
  - 1:10 scale
  - Autonomous racing
- Digital twinning
  - Dynamics interface
  - Perception interface



Digital Twins: Dynamics Interface

- Rigid-body dynamics
- Suspension dynamics
- Tire dynamics
- Actuator dynamics

$$M = \sum {}^iM \qquad X_{COM} = \frac{\sum {}^iM * {}^iX}{\sum {}^iM}$$

$${}^iM * {}^i\ddot{Z} + {}^iB * ({}^i\dot{Z} - {}^i\dot{z}) + {}^iK * ({}^iZ - {}^iz)$$

$${}^im * {}^i\ddot{z} + {}^iB * ({}^i\dot{z} - {}^i\dot{Z}) + {}^iK * ({}^iz - {}^iZ)$$

$$\begin{cases} {}^iF_{t_x} = F({}^iS_x) \\ {}^iF_{t_y} = F({}^iS_y) \end{cases} \qquad {}^iS_x = \frac{{}^ir * {}^i\omega - v_x}{v_x} \qquad {}^iS_y = \tan(\alpha) = \frac{v_y}{|v_x|}$$

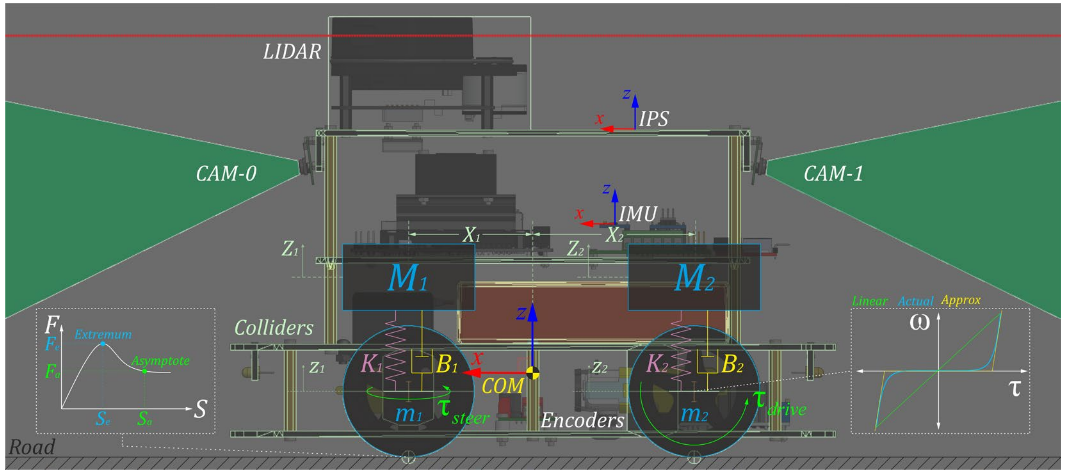
$$F(S) = \begin{cases} f_0(S); & S_0 \leq S < S_e \\ f_1(S); & S_e \leq S < S_a \end{cases}$$

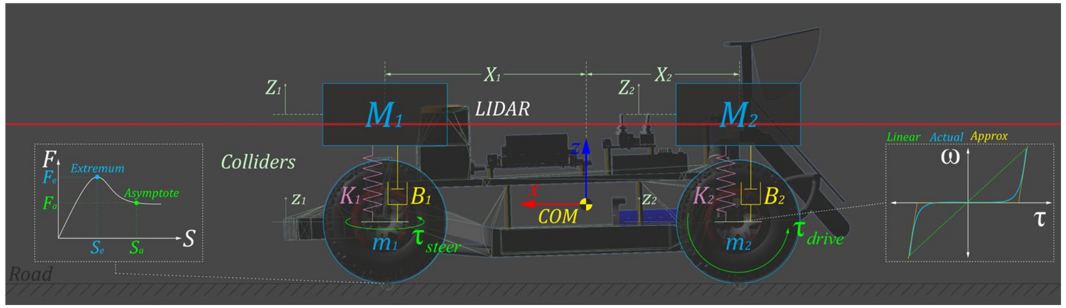
$$f_k(S) = a_k * S^3 + b_k * S^2 + c_k * S + d_k$$

$${}^i\tau_{drive} = {}^iI_w * {}^i\dot{\omega}_w \qquad \tau_{steer} = I_{steer} * \dot{\omega}_{steer}$$

$${}^iI_w = \frac{1}{2} * {}^im_w * {}^ir_w{}^2 \qquad \begin{cases} \delta_l = \tan^{-1}\left(\frac{2*l*\tan(\delta)}{2*l+w*\tan(\delta)}\right) \\ \delta_r = \tan^{-1}\left(\frac{2*l*\tan(\delta)}{2*l-w*\tan(\delta)}\right) \end{cases}$$

$${}^i\tau_{idle} = {}^i\tau_{brake}$$





Digital Twins: Perception Interface

- Throttle sensor
- Steering sensor
- Indoor positioning system
- Inertial measurement unit
- Incremental encoders
- LIDAR
- Cameras

$$\tau_f^t = \tau_u^{t-1} \qquad \delta_f^t = \delta_u^{t-1}$$

$$^{w}\mathbf{T}_v = \left[\begin{array}{c|c} \mathbf{R}_{3\times 3} & \mathbf{t}_{3\times 1} \\ \hline \mathbf{0}_{1\times 3} & 1 \end{array}\right] \in SE(3)$$

$$\{x, y, z\} \quad \{a_x, a_y, a_z\} \quad \{\omega_x, \omega_y, \omega_z\}$$

$$\{\phi_x, \theta_y, \psi_z\} \quad \{q_0, q_1, q_2, q_3\}$$

$$\texttt{raycast}\{^{w}\mathbf{T}_l,\ \vec{\mathbf{R}},\ r_{max}\}$$

$$\theta \in [\theta_{min} : \theta_{res} : \theta_{max}]$$

$$^{w}\mathbf{T}_l = {}^{w}\mathbf{T}_v * {}^{v}\mathbf{T}_l \in SE(3)$$

$$\vec{\mathbf{R}} = [r_{max} * sin(\theta) \ \ r_{min} * cos(\theta) \ \ 0]^T$$

$$\texttt{ranges[i]} = \begin{cases} \texttt{hit.dist} & \text{if } \texttt{ray[i].hit} \text{ and } \texttt{hit.dist} \geq r_{min} \\ \infty & \text{otherwise} \end{cases}$$

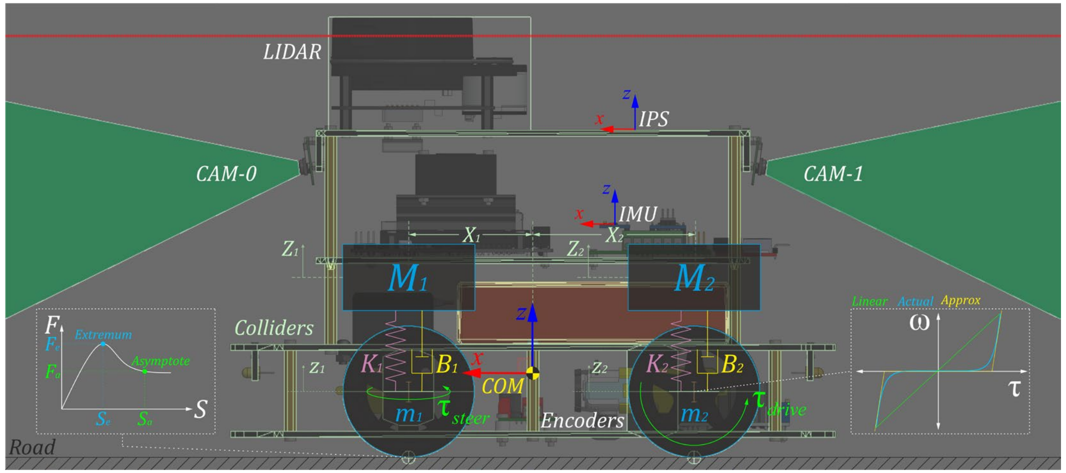
$$\texttt{hit.dist} = \sqrt{(x_{hit} - x_{ray})^2 + (y_{hit} - y_{ray})^2 + (z_{hit} - z_{ray})^2}$$

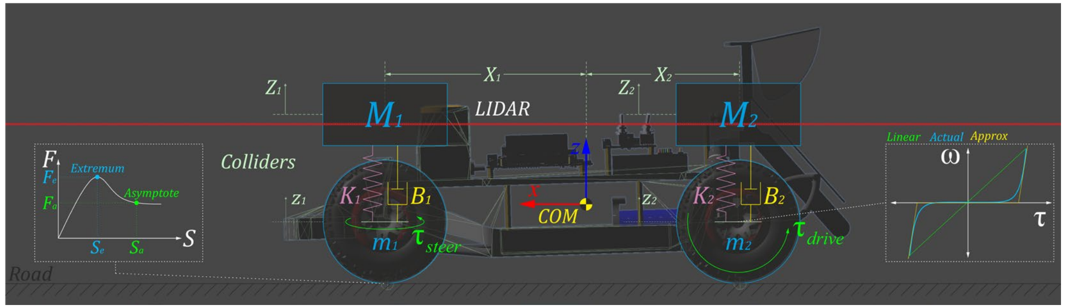
$$\mathbf{V} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & t_0 \\ r_{10} & r_{11} & r_{12} & t_1 \\ r_{20} & r_{21} & r_{22} & t_2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{P} = \begin{bmatrix} \frac{2*N}{R-L} & 0 & \frac{R+L}{R-L} & 0 \\ 0 & \frac{2*N}{T-B} & \frac{T+B}{T-B} & 0 \\ 0 & 0 & -\frac{F+N}{F-N} & -\frac{2*F*N}{F-N} \\ 0 & 0 & -1 & 0 \end{bmatrix}$$

$$f = \frac{2*N}{R-L},\ a = \frac{s_y}{s_x},\ \text{and}\ \frac{f}{a} = \frac{2*N}{T-B}$$

$$\mathbf{W} = [x_w \ \ y_w \ \ z_w \ \ w_w]^T$$

$$\mathbf{C} = [x_c \ \ y_c \ \ z_c \ \ w_c]^T \qquad \mathbf{C} = \mathbf{P} * \mathbf{V} * \mathbf{W}$$





Cooperative Multi-Agent RL

Collaborative Intersection Traversal

Problem Formulation

- POMDP
  - Limited state-sharing (V2V)
- Observation space
  - Relative goal position
  - Relative peer poses
  - Absolute peer velocities
- Action space
  - Discrete throttle
  - Discrete steering
- Reward formulation
  - Goal
  - Collision
  - Lanes

$$o_t^i = \left[g^i, \tilde{p}^i, \tilde{\psi}^i, \tilde{v}^i\right]_t \in \mathbb{R}^{2+4(N-1)}$$

$$g_t^i = \left[g_x^i - p_x^i, g_y^i - p_y^i\right] t \in \mathbb{R}^2$$

$$\tilde{p}t^i = \left[p_x^j - p_x^i, p_y^j - p_y^i\right] t \in \mathbb{R}^{2(N-1)}$$

$$\tilde{\psi}t^i = \psi_t^j - \psi_t^i \in \mathbb{R}^{N-1}$$

$$\tilde{v}t^i = vt^j \in \mathbb{R}^{N-1}$$

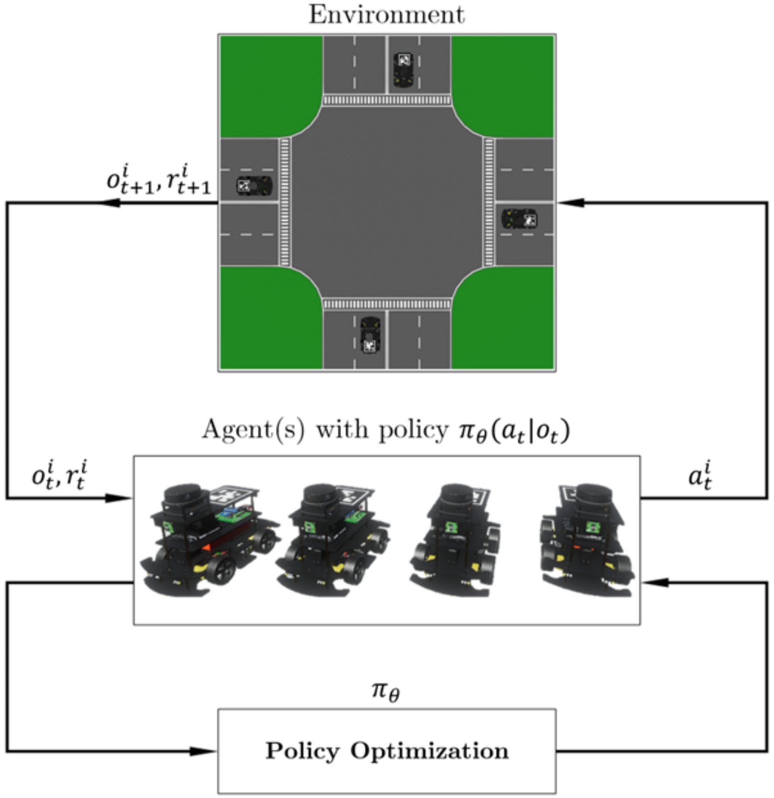
$$a_t^i = [\tau_t^i, \delta_t^i] \in \mathbb{R}^2$$

$$\tau_t \in \{0.5, 1.0\}$$

$$\delta_t \in \{-1, 0, 1\}$$

$$r_t^i = \begin{cases} r_{goal}; & \text{if safe traversal} \\ -k_p * \left\|g_t^i\right\|_2; & \text{if traffic violation} \\ k_r * (0.001 + \left\|g_t^i\right\|_2)^{-1}; & \text{otherwise} \end{cases}$$

$$r_{goal} = +1 \qquad k_r = 0.01 \qquad k_p = 0.425$$

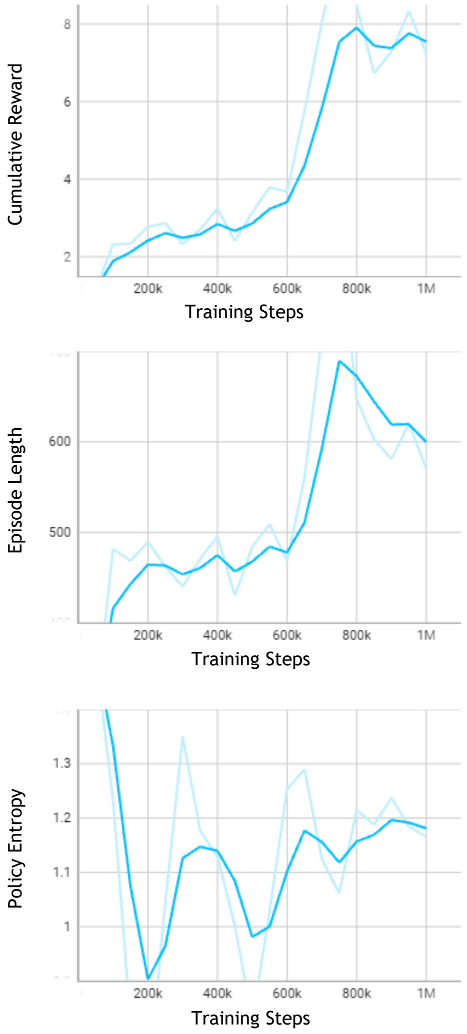


MARL Training

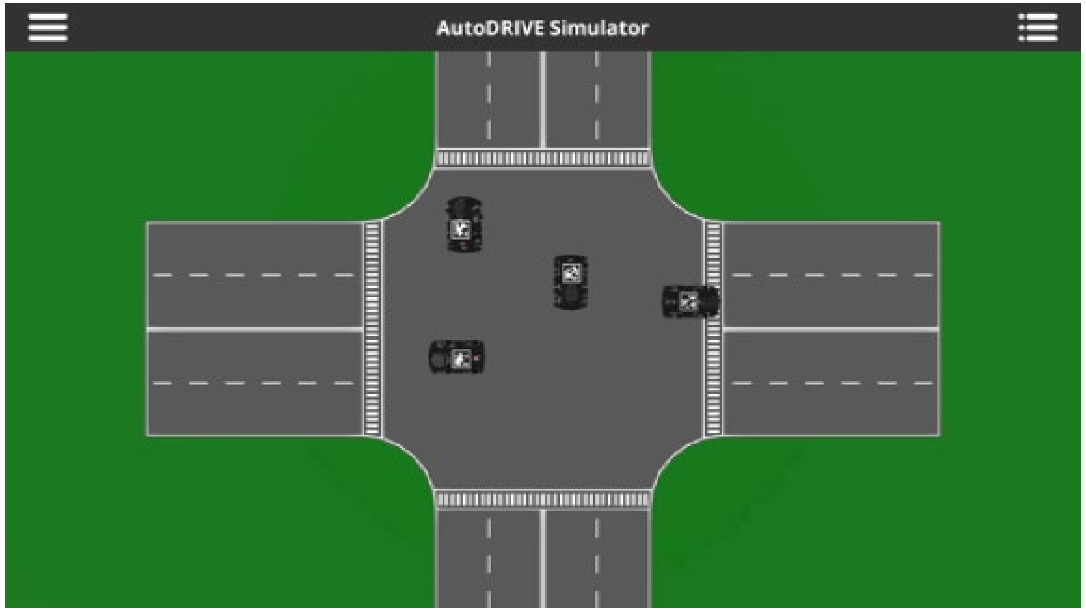
- Training hardware
  - **CPU:** 12th Gen Intel Core i9-12900H 2.50 GHz
  - **RAM:** 32.0 GB (31.7 GB usable)
  - **GPU:** NVIDIA GeForce RTX 3080 Ti
- Non-parallelized training time
  - 3.543 hours
- Scenarios
  - Single-agent learning
  - Multi-agent learning
- Training analysis
  - Cumulative reward
  - Episode length
  - Policy entropy

Single-Agent Scenario

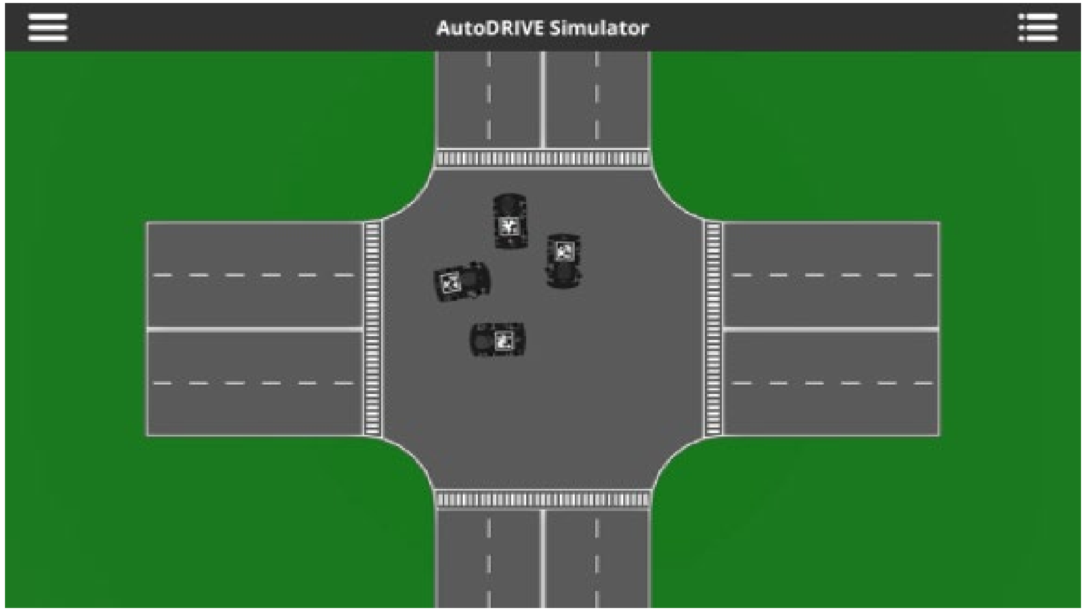
Multi-Agent Scenario



MARL Deployment



Single-Agent Learning Scenario



Multi-Agent Learning Scenario

Competitive Multi-Agent RL

Head-to-Head Autonomous Racing

Problem Formulation

- POMDP
  - No state-sharing
- Observation space
  - Own velocity
  - Sparse LIDAR measurements
- Action space
  - Discrete throttle
  - Discrete steering
- Reward formulation
  - Collision
  - Checkpoints (19 virtual ckpts)
  - Lap completion (1 virtual ckpt)
  - Best time
  - Velocity

$$o_t^i = [v_t^i, m_t^i] \in \mathbb{R}^{28}$$

$$\begin{aligned} v_t^i &\in \mathbb{R}^1 \\ m_t^i &= [{}^1m_t^i, {}^2m_t^i, \cdots, {}^{27}m_t^i] \in \mathbb{R}^{27} \end{aligned}$$

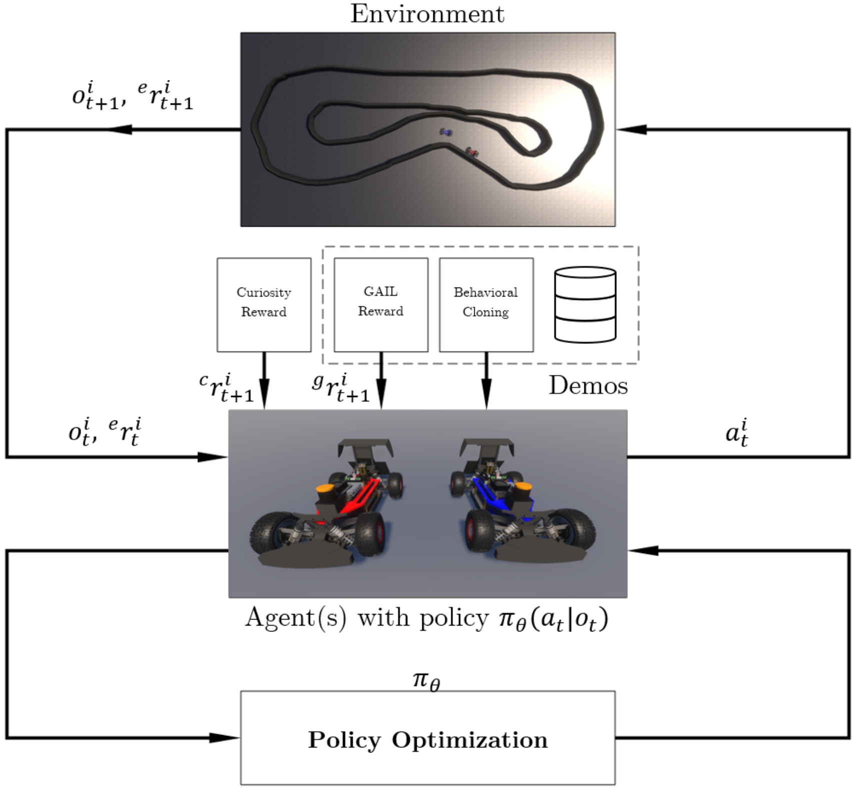
$$a_t^i = [\tau_t^i, \delta_t^i] \in \mathbb{R}^2$$

$$\begin{aligned} \tau_t^i &\in \{0.1, 0.5, 1.0\} \\ \delta_t^i &\in \{-1, 0, 1\} \end{aligned}$$

$${}^er_t^i = \begin{cases} r_{collision} & \text{if collision occurs} \\ r_{checkpoint} & \text{if checkpoint is passed} \\ r_{lap} & \text{if completed lap} \\ r_{best\ lap} & \text{if new best lap time is achieved} \\ 0.01 * v_t^i & \text{otherwise} \end{cases}$$

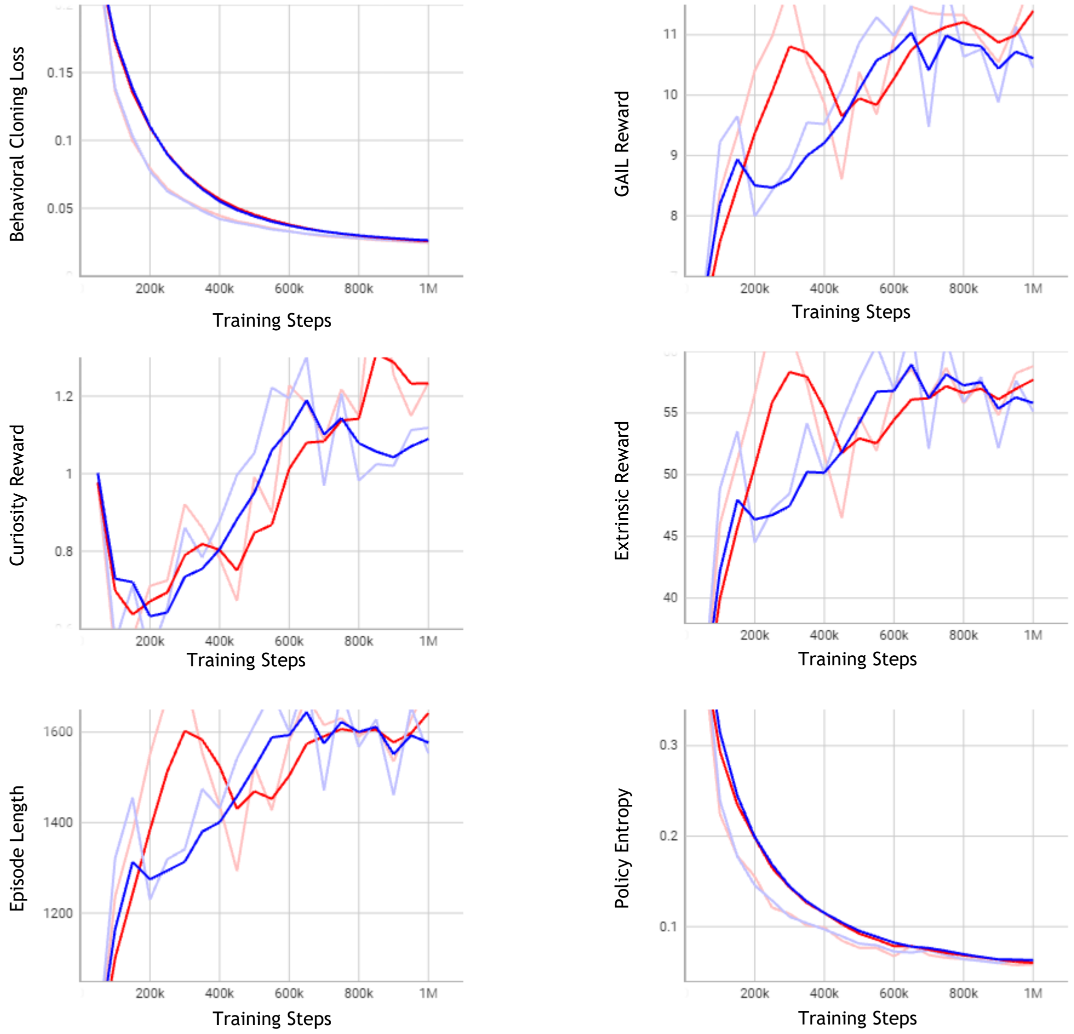
$$r_{collision} = -1 \qquad r_{checkpoint} = +0.01$$

$$r_{lap} = +0.1 \qquad r_{best\ lap} = +0.7$$

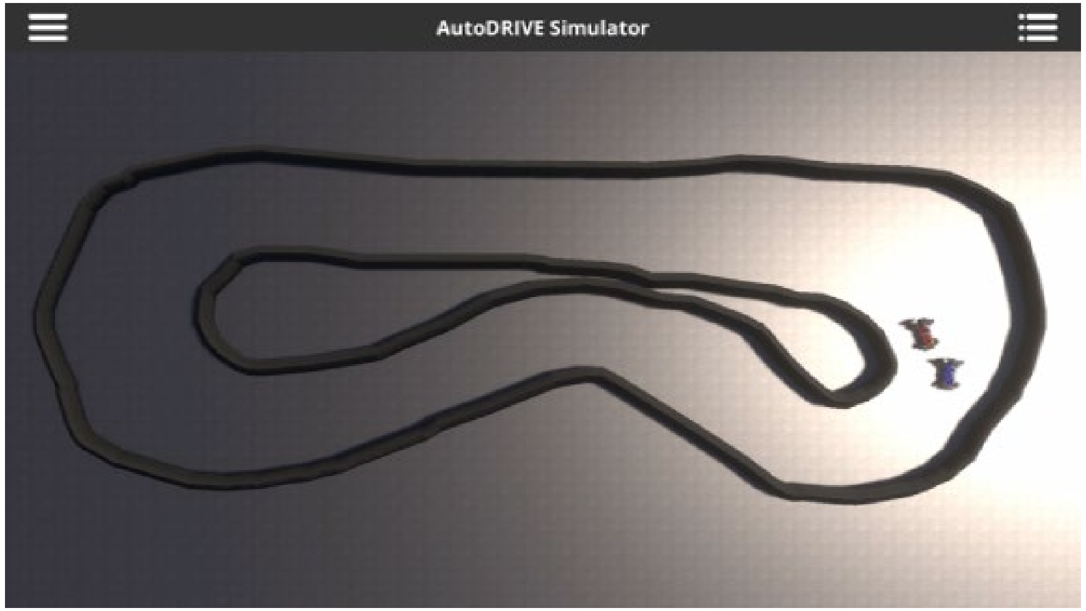


MARL Training

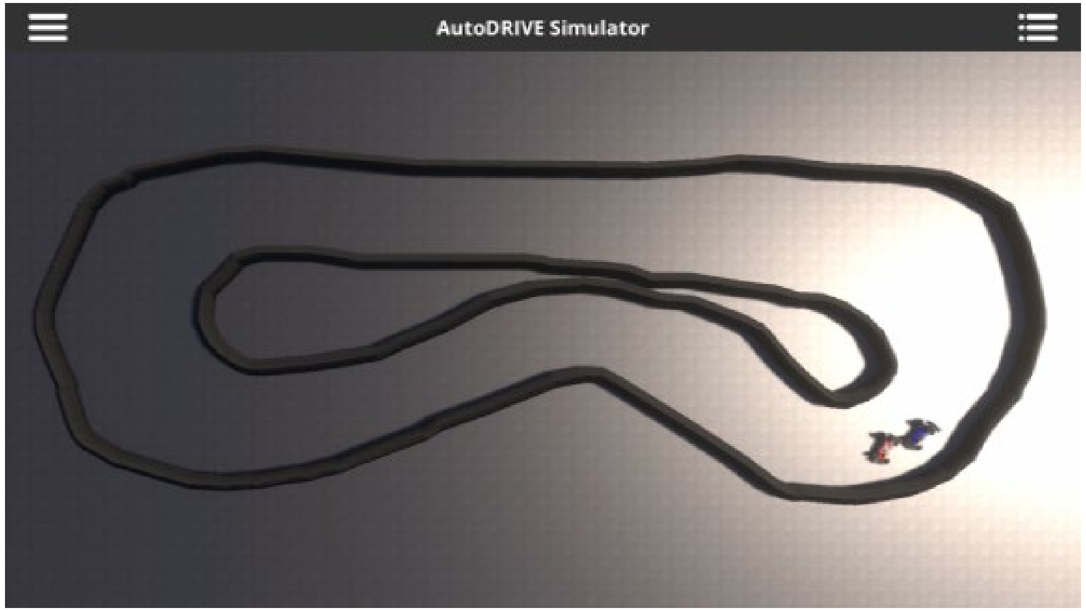
- Training hardware
  - **CPU:** 12th Gen Intel Core i9-12900H 2.50 GHz
  - **RAM:** 32.0 GB (31.7 GB usable)
  - **GPU:** NVIDIA GeForce RTX 3080 Ti
- Non-parallelized training time
  - 4.211 hours (SOTA: >72 hours with better compute [9])
- Training analysis
  - BC Loss
  - GAIL reward
  - Curiosity reward
  - Extrinsic reward
  - Episode length
  - Policy entropy



MARL Deployment



Block-Block-Overtake



Let-Pass-and-Overtake

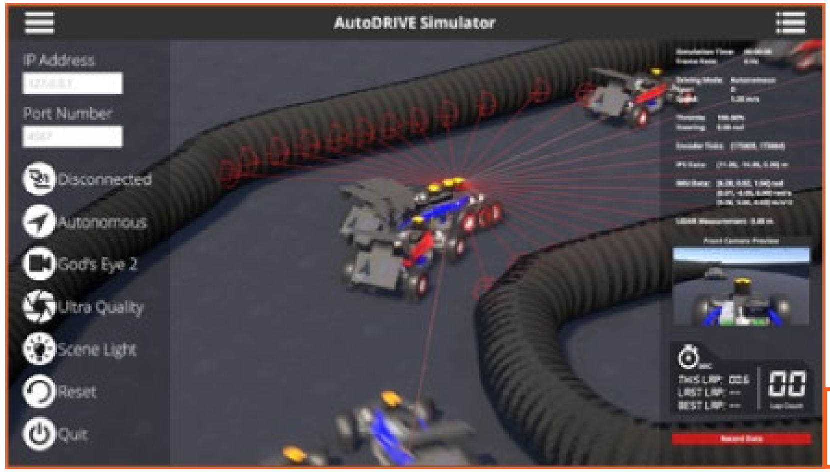
Scalable Simulation Parallelization

Scalable and Parallelizable Multi-Agent Reinforcement Learning for AVs

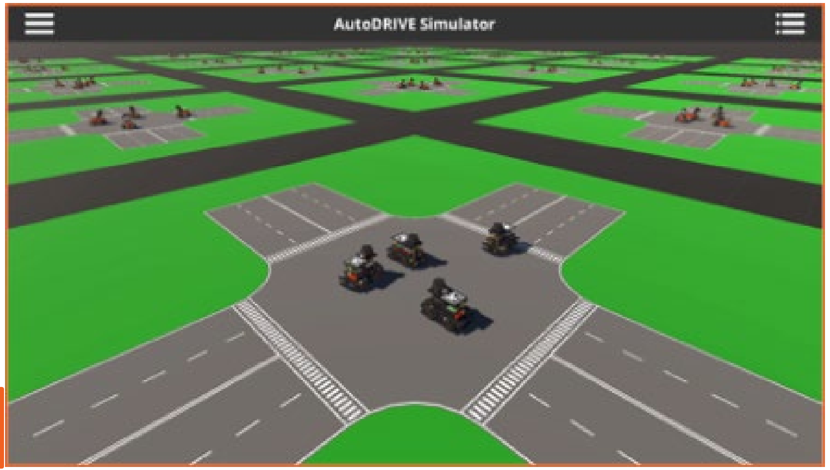
Simulation Parallelization

- Scalable parallelization architecture
    - Object oriented programming (OOP)
    - CPU multi-threading
    - GPU instancing
    - Isolation in different “layers” of simulation
    - Environment on “default” layer
    - Each family of multi-agent system on same layer
- Parallel perception, collision and interaction
    - Interoceptive sensors have little to no issue in parallelizing
    - **Camera:** rendering culling mask only for specific layers
    - **LIDAR:** raycasting returns hit only for specific layers
    - Interactions and collision-checks between specific layers

Note: The term “family” here denotes the problem formulation for multi-agent RL. In our case, this translates to 4 agents for the intersection traversal scenario and 2 agents for the autonomous racing scenario.



n-Agents



n-Environments

Parallel MARL

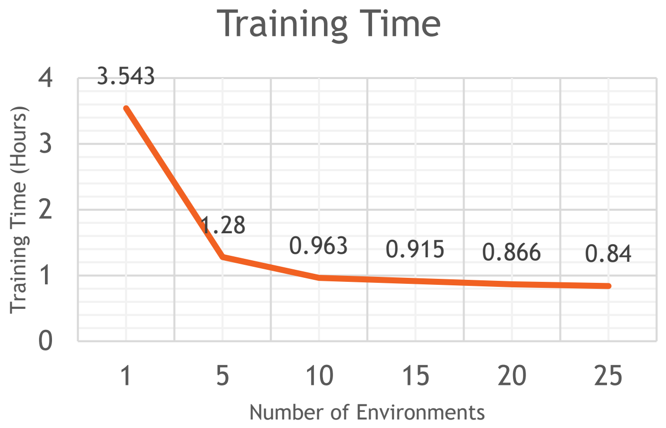
- Cooperative MARL parallelization
  - {1, 5, 10, 15, 20, 25} environments
  - {1x4, 5x4, 10x4, 15x4, 20x4, 25x4} agents
- Competitive MARL parallelization
  - 1 environment
  - {1x2, 2x2, 4x2, 6x2, 8x2, 10x2} agents
- Computational analysis
  - Training time (hours)
  - Reduction in training time (%)

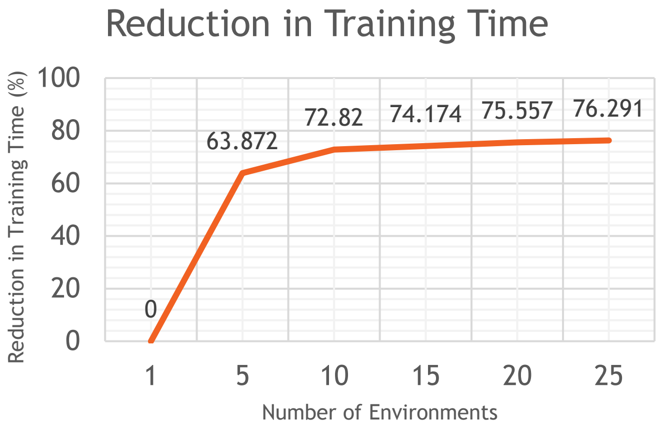
Notation:

$$\{i \times j\}$$

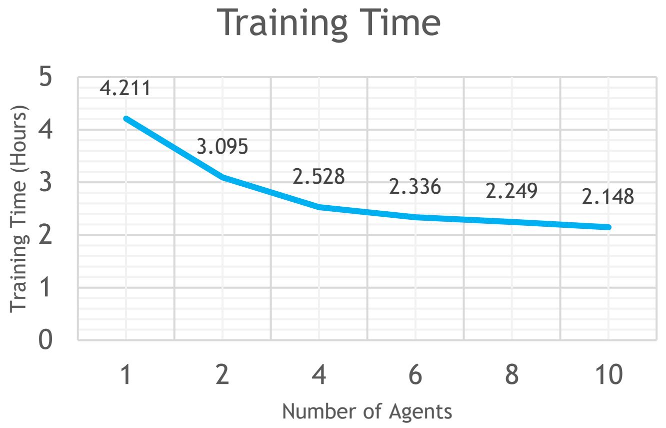
i = Parallel agents/environments
j = Number of agents per family

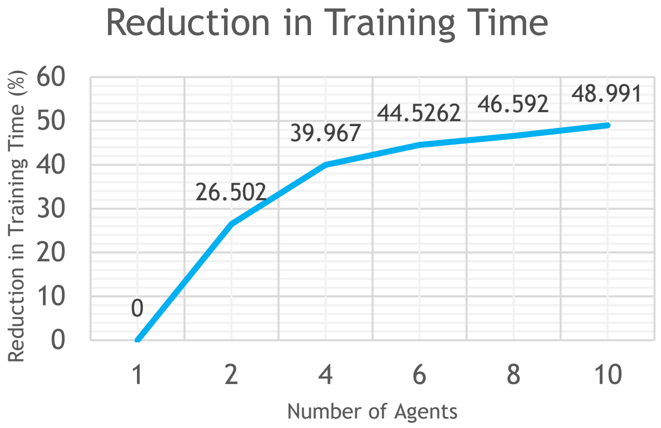
Cooperative MARL





Competitive MARL





Conclusion

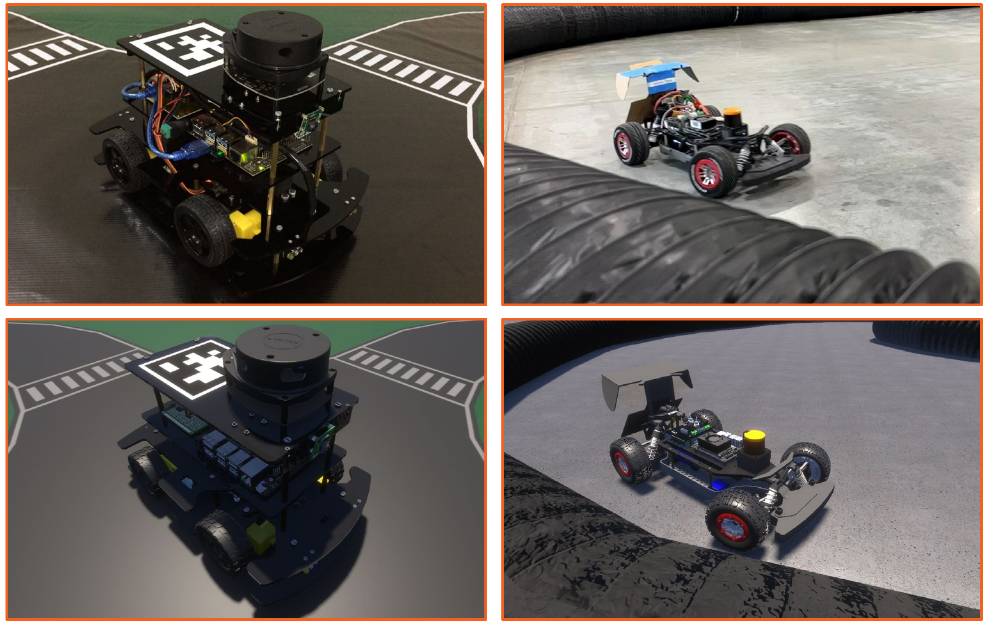
Scalable and Parallelizable Multi-Agent Reinforcement Learning for AVs

Summary

- Digital twin creation
  - Geometric measurements & modeling
  - Static measurements & calibration
  - Simulation models (dynamics + graphics)
  - Physics-based sensor simulation
- Scalable multi-agent reinforcement learning
  - Vanilla RL (1 agent problem in dynamic environment)
  - Vanilla RL (4 agent cooperative problem)
  - Demonstration guided RL (2 agent competitive problem)
- Modular and parallelizable simulation
  - Environment parallelization
  - Agent parallelization
  - Computational analysis

AutoDRIVE

F1 TENTH



Future Plans

- Analyze RL training across different types of computing platforms
  - Laptops, desktops, workstations, Palmetto HPC (different configs.)
- Integrated digital twin framework for sim2real transfer of MARL
  - 1 physical vehicle
  - Virtual peers/competitors in loop with the physical vehicle
- Physics informed RL
  - State transition models, physics-informed rewards, safety guarentees, etc.
- Extend this approach to different vehicles and architectures

AutoDRIVE

Thank You!

…open to questions and suggestions

References

1. T. Samak, C. Samak, S. Kandhasamy, V. Krovi, and M. Xie, “AutoDRIVE: A Comprehensive, Flexible and Integrated Digital Twin Ecosystem for Autonomous Driving Research & Education,” Robotics, vol. 12, no. 3, p. 77, May 2023, doi: 10.3390/robotics12030077
2. W. Ding, C. Xu, M. Arief, H. Lin, B. Li and D. Zhao, “A Survey on Safety-Critical Driving Scenario Generation—A Methodological Perspective,” in IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 7, pp. 6971-6988, July 2023, doi: 10.1109/TITS.2023.3259322
3. S. H. Semnani, H. Liu, M. Everett, A. de Ruiter, and J. P. How, “Multiagent Motion Planning for Dense and Dynamic Environments via Deep Reinforcement Learning,” 2020.
4. P. Long, T. Fan, X. Liao, W. Liu, H. Zhang, and J. Pan, “Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning,” 2018
5. S. Aradi, “Survey of Deep Reinforcement Learning for Motion Planning of Autonomous Vehicles,” 2020.
6. D. Wang, H. Deng, and Z. Pan, “MRCDRL: Multi-Robot Coordination with Deep Reinforcement Learning,” Neurocomputing, 2020.
7. X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, “Learn to navigate: cooperative path planning for unmanned surface vehicles using deep reinforcement learning,” IEEE Access, vol. 7, pp. 165 262–165 278, 2019.

References

8. K. Sivanathan, B. K. Vinayagam, T. Samak, and C. Samak, "Decentralized Motion Planning for Multi-Robot Navigation using Deep Reinforcement Learning," in 2020 3rd International Conference on Intelligent Sustainable Systems (ICISS), 2020, pp. 709–716, doi: 10.1109/ICISS49785.2020.9316033
9. F. Fuchs, Y. Song, E. Kaufmann, D. Scaramuzza, and P. Durr, "Super-Human Performance in Gran Turismo Sport Using Deep Reinforcement Learning," IEEE Robotics and Automation Letters, vol. 6, no. 3, pp. 4257–4264, 2021.
10. Y. Song, H. Lin, E. Kaufmann, P. Durr, and D. Scaramuzza, "Autonomous Overtaking in Gran Turismo Sport Using Curriculum Reinforcement Learning," 2021.
11. C. V. Samak, T. V. Samak, and S. Kandhasamy, "Autonomous Racing using a Hybrid Imitation-Reinforcement Learning Architecture," 2021. [Online]. Available: https://arxiv.org/abs/2110.05437
12. J. Betz, H. Zheng, A. Liniger, U. Rosolia, P. Karle, M. Behl, V. Krovi, and R. Mangharam, "Autonomous Vehicles on the Edge: A Survey on Autonomous Vehicle Racing," IEEE Open Journal of Intelligent Transportation Systems, vol. 3, pp. 458–488, 2022.