The Learn-to-Race Autonomous Racing Virtual Challenge

Jonathan Francis[*, †] JMF1@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA
Human-Machine Collaboration, Bosch Research Pittsburgh, 2555 Smallman St., Pittsburgh, PA, USA

Bingqing Chen[*] BINGQINC@ANDREW.CMU.EDU
College of Engineering, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA

Siddha Ganju[*] SGANJU@NVIDIA.COM
NVIDIA, 2788 San Tomas Expy, Santa Clara, CA, USA

Sidharth Kathpal[*] SKATHPAL@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA

Jyotish Poonganam[*] JYOTISH@AICROWD.COM
AICrowd, EPFL Innovation Park, Lausanne, CH

Ayush Shivani[*] ASHIVANI@AICROWD.COM
AICrowd, EPFL Innovation Park, Lausanne, CH

Sahika Genc
Amazon, 550 Terry Ave N, Seattle, WA

Ivan Zhukov ZHUKOV@ARRIVAL.COM
Autonomous Driving & Advanced Driver Assistance Systems, Arrival Ltd., London, UK W14 8TS

Max Kumskoy KUMSKOY@ARRIVAL.COM
Autonomous Driving & Advanced Driver Assistance Systems, Arrival Ltd., London, UK W14 8TS

Anirudh Koul AKOUL@ALUMNI.CMU.EDU
Pinterest, 651 Brannan St, San Francisco, CA, USA

Jean Oh JEANOH@CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA

Eric Nyberg EHN@CS.CMU.EDU
School of Computer Science, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA, USA

∗. Indicates equal contribution from these authors.
†. Corresponding author — Jonathan Francis, jmf1@alumni.cmu.edu

Abstract

We present the results of our new autonomous racing virtual challenge, predicated on the high-fidelity `Learn-to-Race` (L2R) simulation framework, which seeks to encourage interdisciplinary research in autonomous driving and to help advance state-of-the-art on a practical benchmark. The main goal of the challenge is to evaluate the joint safety, performance, and generalization capabilities of perception and control algorithms, in autonomous racing. Analogous to racing being used to test cutting-edge vehicle technology, we envision autonomous racing to serve as a particularly challenging proving ground for safe learning algorithms as: (i) vehicles are required to drive at their physical limits, with barely any margin for safety, where any infraction could lead to catastrophic failure; (ii) autonomous agents are required to make sub-second decisions, in fast-changing environments; and (iii) visual perception pipelines must remain robust to distribution shifts, novel road features, and other obstacles, in order to facilitate cross-domain safety and performance. In this paper, we describe the new L2R Task 2.0 benchmark, with new metrics and baseline approaches. We also provide an overview of deployment, evaluation, and rankings for the inaugural instance of the L2R Autonomous Racing Virtual Challenge (supported by Carnegie Mellon University, Arrival Ltd., AICrowd, Amazon Web Services, and Honda Research), which officially used the new `L2R` Task 2.0 benchmark and received over 20,100 views, 437 active participants, 46 teams, and 733 model submissions—from 88 unique institutions, in 28 different countries. Finally, we release leaderboard results from the challenge and provide description of the two top-ranking approaches in cross-domain model transfer, across multiple sensor configurations and simulated races.

1. Introduction

Autonomous driving has been gaining traction in the automotive and trucking industries, with the potential to reduce driving accidents and increase roadway efficiency. Recent, notable failures in vehicle perception pipelines have exposed severe *limitations* in models' abilities to make quick, safe, and generalisable decisions, without compromising performance-oriented objectives. Several real-world autonomous racing challenges (e.g., RoboRace (2015); Indy (2019)) were introduced with the intention of deploying algorithms at the extreme of *Formula*-style track racing, but, without high-fidelity racing simulation and model training environments, large-scale experimentation in this space could prove particularly costly and dangerous.

Analogous to traditional real-world racing being used as the proving ground for high-performance automotive technology, we also envision autonomous racing to serve as such for autonomous technology. Vehicles are required to drive at their physical limits, with barely any margin for safety, where any infraction could lead to catastrophic failure; autonomous agents are required to make sub-second decisions, in fast-changing environments; and visual perception pipelines must remain robust to distribution shifts, novel road features, and other obstacles, in order to facilitate cross-domain safety and performance. In Herman et al. (2021), we released an open-source, high-fidelity simulation environment for high-speed racing, `Learn-to-Race` (`L2R`), with the hope to democratize autonomous racing research.[1] In order to encourage use of `L2R` and to facilitate standardized benchmarking, we launch the `Learn-to-Race` autonomous racing virtual challenge, with multi-institutional support from Carnegie Mellon University, Arrival Ltd., AICrowd, Amazon Web Services, and Honda Research. In its inaugural instance, the 5-month international virtual AI challenge

1. See: `https://learn-to-race.org`

enjoyed strong participation, with 20,100+ views, 437 individual participants, 46 teams, and 733 submissions—from 88 unique institutions, in 28 different countries.

While the `L2R` framework was intended to facilitate a wide range of research topics, e.g., learning from demonstration, reinforcement learning (RL), model predictive control (MPC), sim-to-real transfer, and multimodal representation learning, we focus specifically on safe learning in the first instance of the `Learn-to-Race` challenge. As autonomous technology advances, it is of paramount importance for autonomous vehicles to adheres to safety specifications, whether in urban driving or high-speed racing. Racing demands each vehicle to drive at its physical limits with little margin for safety, when any infraction could lead to catastrophic failures. Given this inherent tension, autonomous racing is a particularly challenging task for safe learning algorithms.

Thus, the objective of the `Learn-to-Race` challenge is to push the boundary of autonomous technology, with a focus on realizing safety-awareness in autonomous driving. In the challenge, participants develop RL agents to drive as fast as possible, while adhering to the safety constraints. Furthermore, participants are required to use high-dimensional visual data as inputs, in contrast to low-dimensional features as in Florian et al. (2020); Wurman et al. (2022). Finally, we also test the agents' ability to adapt to a new environment, with a fixed time budget for safe exploration. Through this challenge, we pose the following fundamental research questions:

- How can an autonomous agent push performance towards its physical limits, while adhering to safety specifications?
- How does an autonomous agent learn salient representations from high-dimensional sensory inputs that are generalisable and robust?
- How should an autonomous agent explore safely and adapt to unseen scenarios?
- How can we inject domain knowledge (e.g., model-based priors, common sense, logical rules, safety specifications, skill primitives, expert demonstrations) such that the autonomous agent is safer and more sample-efficient?
- What models and architectures are realistic, to facilitate simulation-to-real transfer of models—to be integrated within autonomous racing vehicle software stacks?

In this paper, we release the `Learn-to-Race` Task 2.0 benchmark, with new metrics and baselines. We provide an overview of the inaugural `Learn-to-Race` Autonomous Racing Virtual Challenge, including description of deployment, evaluation, and ranking procedures. Finally, we present the leaderboard results as well as descriptions of two top-ranking approaches, detailing strategies for cross-domain transfer, single- versus multi-camera configuration, and hybrid modeling.

The remainder of this manuscript is organized as follows: we provide high-level desiderata for autonomous racing vehicles in Section 2. In Section 3, we summarize related work, existing simulation frameworks, and previous challenges in autonomous racing. In Sections 4 and 5, we briefly describe the `Learn-to-Race` (`L2R`) framework and release the official `L2R` Task 2.0 Benchmark. In Section 6, we summarize the inaugural `L2R` autonomous racing virtual challenge and, in Section 6.3, we provide the results of the challenge, in the form of leaderboard scores and top-ranking approach descriptions. We wrap up this manuscript with insights from the challenge and descriptions of future directions, in Section 7, and with conclusions in Section 8.

2. Desiderata

In autonomous racing, vehicles must trade-off multiple (possibly-conflicting) objectives, in order to pursue safe, robust, and generalisable behaviour. Below, we outline some desired characteristics, as goals to achieve in this domain.

1. *Capable of traveling at high speeds*: as the main proxy for performance capability in autonomous racing, agents should maximise track speed, in order to complete laps quickly.
2. *Does not engage in safety infractions*: in the pursuit of real-world deployment of autonomous racing algorithms, where system failures could have catastrophic ramifications, agents are not permitted to crash. Agents must also negotiate obstacles, turns, and other agents on the road, e.g., with an explicit model of safe behaviour.
3. *Adapts to unseen contexts*: the agent must be able to adapt and generalise to unseen contexts, such as new racetracks, new visual conditions (e.g., glare, shadows), and new dynamical regimes (e.g., effects of new weather conditions, different vehicles), that are not seen during training or initial periods of task-execution.
4. *Adheres to task structure*: regardless of whether the environment context is seen or unseen, the agent must appropriately adhere to task structure—e.g., staying on the driveable area, steering away from road alert stripes, and driving the correct way on the track.
5. *Manifests algorithmic efficiency*: the agent should employ smooth and energy/resource-efficient behaviour when possible, e.g., trading off cornering speed with boundary-relative position on the track.
6. *Algorithmic transferability*: agents must perform well in both simulation and the real-world.

3. Related Work

In this section, we review prior work on autonomous racing, and existing simulators and similar competitions, which motivate and inform the design of `Learn-to-Race` challenge.

Autonomous racing. One approach for autonomous racing is via MPC (Liniger et al., 2015; Rosolia et al., 2017; Kabzan et al., 2019), which solves a planning problem iteratively over a receding time horizon with a model of the system dynamics. Aside from the challenges in modeling the complex dynamics, a significant drawback of such approach is the dependence on extensive sensor installation for localization and state estimation (Cai et al., 2021). Another approach is to use a modular pipeline (Kabzan et al., 2019; Strobel et al., 2020; Francis et al., 2022; Tatiya et al., 2022), starting from perception on raw sensory inputs, to localization and object-detection, and finally to planning and control. While this approach is most commonly used in practice, disadvantages of the approach include over-complexity and error propagation (Yurtsever et al., 2020; Francis et al., 2022). Recently, there is a lot of interest in using RL-based approaches for autonomous racing. In Florian et al. (2020); Chisari et al. (2021); Wurman et al. (2022), RL agents were trained using low-dimensional features as inputs. In Chen et al. (2015); Drews et al. (2017), intermediate features were extracted from perception pipelines to determine control actions. In Cai et al. (2021); Weiss and Behl (2020), RL agents were trained end-to-end on visual inputs by imitating expert demonstration;

in Cai et al. (2021), a data-driven model of the environment was further utilized to train the agent by unrolling future trajectories.

While there is a large body of literature on end-to-end RL on perception data for urban driving (Codevilla et al., 2018; Ohn-Bar et al., 2020; Codevilla et al., 2019; Chen et al., 2020; Zhang and Ohn-Bar, 2021; Prakash et al., 2021; Zhang et al., 2021; Zhang and Ohn-Bar, 2021), there are significantly fewer works on the same topic for high-speed racing. We hypothesize this may partly be attributed to the lack of open-source, high-fidelity simulation environments for racing, in comparison to the ubiquity of CARLA simulator Dosovitskiy et al. (2017a) for urban driving research. It is our hope that `Learn-to-Race` will democratize and facilitate autonomous racing research.

Autonomous racing simulators and competitions. `CarRacing-v0`, an OpenAI-gym environment (Brockman et al., 2016), is a simple racing environment, which provides a bird-eye view of 96×96 pixels. TORCS (TORCS, 1997) is an open-source simulator, which enabled research on perception-based autonomous racing, such as Chen et al. (2015); Drews et al. (2017), and is also used by the Simulated Car Racing Championship (Loiacono et al., 2013). However, the vision quality of TORCS is low by contemporary standards and in comparison to the CARLA simulator for urban driving. Gran Turismo, a video game by Sony, has been used in for autonomous racing research in Florian et al. (2020); Wurman et al. (2022), wherein the researchers reported super-human performance from RL agents trained on a handful of selected features, e.g., vehicle states and way-points for upcoming track segments. However, Gran Turismo is not open-source, only runs on specialised gaming hardware, and provides no support for interfacing with the real-world environment. These factors significantly limit the ability for external researchers to benchmark the environment and flexibly use it for real-world deployment. DeepRacer (Balaji et al., 2020) is a open-source platform that support both simulation and real-world deployment of 1/18$^{\text{th}}$ scale cars, which facilities research on end-to-end RL on raw visual inputs and sim-to-real transfer. There is also a recurring competition, DeepRacer League, based on the platform. F1TENTH (O'Kelly et al., 2020) is a similar framework based on 1/10$^{\text{th}}$ scale cars and has its competition series. In comparison to the aforementioned environments, `Learn-to-Race` is a open-source simulation environment based on full-scale race cars with complex dynamics, and high-fidelity visual rendering.

Aside from DeepRacer and F1TENTH, which use reduced-scale cars, there are also a number of real-world challenges on full-scale race cars, such as RoboRace (RoboRace, 2015), Indy Autonomous Challenge (Indy, 2019), and AMZ Driverless. Due to the significant cost for incurring safety infractions, the participants generally adopt a modular pipeline, instead of end-to-end RL. In comparison, a simulation environment allows for low-cost experimentation of novel algorithms. We envision the `Learn-to-Race` framework, which closely follow the format of RoboRace, will enable sim-to-real transfer to RoboRace in the near future.

4. `Learn-to-Race` Framework

In our prior work (Herman et al., 2021), we released `Learn-to-Race` (`L2R`): a python- and pytorch-based open-source, OpenAI Gym-compliant (Brockman et al., 2016) framework which leverages a high-fidelity racing simulator or real-world vehicle interface (e.g., implemented in the Robot Operating System; ROS Quigley et al. (2009)), for training and deployment of machine learning algorithms for continuous control contexts. We utilise the Arrival Ltd. autonomous racing simulator,

Figure 1: `Learn-to-Race` interfaces with a racing simulator, which features numerous real-world racetracks such as the Thruxton Circuit (*top-left*) and Las Vegas Motor Speedway (*top-right*).

which not only captures complex vehicle dynamics and renders photo-realistic views, but also plays a key role in bringing autonomous racing technology to real life—through official contribution to the Roborace Challenge series, the world's first extreme competition of teams developing autonomous racing technologies. Figure 2 illustrates the `L2R` system components, and we refer interested readers to Herman et al. (2021); Chen et al. (2021) for further details about the framework.

Learn-to-Race environment. The `L2R` environment consists of four core component classes: CONTROLLER, ENVIRONMENT, AGENT, and TRACKER. The Controller initialises the backend (i.e., the simulator runtime or the ROS-based vehicle software stack) and initiates software communication interfaces with the backend's sensors, peripherals, and operation state-control mechanisms; after this initialisation step, the Controller can be used to perform 'get' and 'set' operations on the backend's configuration of sensor and vehicle parameters. The Environment class facilitates a training and deployment paradigm for machine learning models, in the form of popular OpenAI gyms, complete with the characterisation of the continuous control task in autonomous racing as a Markov decision process (MDP)—with an action space, an observation space, and reward/cost functions. The Environment's `make`, `step`, and `reset` methods, typical of MDP-inspired software control loops, are indeed implemented as adapters through the Controller's requisite interfaces to the backend. The Agent code implements a control policy, which maps *observations* (received from the Environment, via the Controller's sensor interfaces with the backend) to *actions* (sent to the Environment, for issuing vehicle control commands to the backend, via the Controller interfaces); prototypical Agent code is provided by the `L2R` framework, as templates for implementing the `select_action` method, which is typical of approaches that operate within MDP-like training and inference settings. The Tracker module maintains a registry of the Agent's state, measures its progress towards task completion, records past success/failure conditions, and calculates the official task metrics.

Autonomous racing simulator backend. `L2R` supports the use of an autonomous driving simulator, as a backend API service, for developing policy algorithms for vehicle control. Currently, `L2R` uses the Arrival simulator (Herman et al., 2021), which is a powerful tool for the development and testing of autonomous vehicles. It is based on Unreal Engine 4 and includes such features as: (i) a vehicle prototyping framework; (ii) full software-in-the-loop (SIL) simulation, to model all vehicle control devices; (iii) controller area network (CAN) bus interface; (iv) camera, inertial

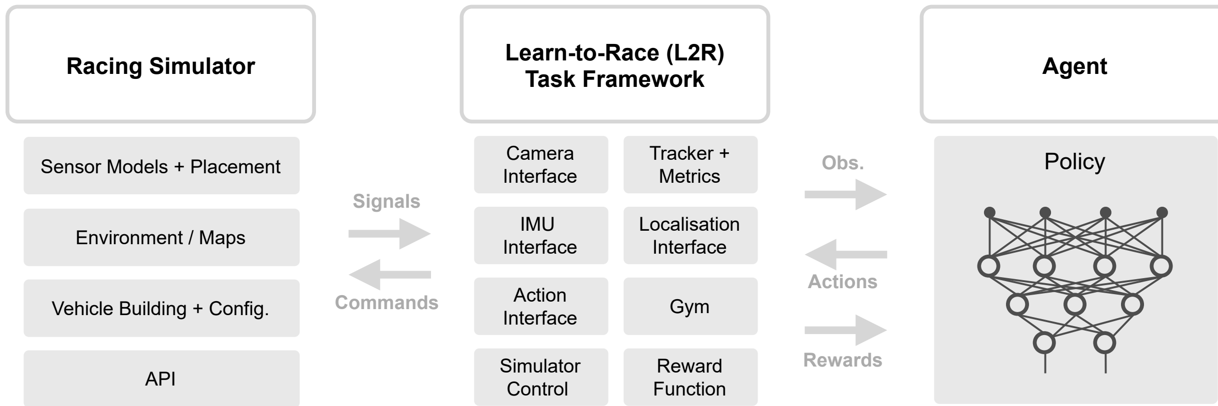

Figure 2: `Learn-to-Race` Framwork

measurement unit (IMU), light detection and ranging (LiDAR), ultrasonic, and radar sensor models; (v) semantic segmentation; (vi) sensor placement and configuration facilities; (vii) V2V/V2I interface subsystem; (vii) dynamic racing scenario creation; (viii) race track generation from scanned datasets; (ix) support for full integration with the CARLA simulator (Dosovitskiy et al., 2017b); and (x) an application programming interface (API), which is automatically generated based on C++ code analysis. The `Learn-to-Race` framework maintains a connection with a simulator, via the interfaces established by the Controller, on initialisation and throughout task execution.

5. `Learn-to-Race` Task 2.0 Benchmark

Alongside the `Learn-to-Race` framework, we defined Task 1.0 in Herman et al. (2021) as a benchmark for evaluating progress towards autonomous racing technology. In Task 2.0, we seek to assess agents on the basis of their joint safety, performance, and generalisability to unseen contexts. We provide an overview of new metrics and evaluation procedures, introduced in `L2R` Task 2.0.

5.1 Primary Task Metrics

We follow Herman et al. (2021) in utilising our previous driving quality metrics, for the new benchmark: Episode Duration (ED), Average Adjusted Track Speed (AATS), Average Displacement Error (ADE), Trajectory Admissibility (TrA), Trajectory Efficiency (TrE), and Movement Smoothness (MS). Relative to these original Task 1.0 metrics, we also highlight the three important metrics for the Task 2.0 benchmark, introduced in this paper. In particular, we introduce success rate (as a proxy for safety), we retain average adjusted track speed (as a proxy for performance), and we introduce the total number of safety infractions.

Success Rate. Success rate (SR) is evaluated upon completion of a lap on a race track. Each race track is partitioned into a fixed number of segments, and the success rate is calculated as the number of successfully completed segments over the total number of segments (Eqn. 1). In `Learn-to-Race`, the agent is considered to have failed and the current episode terminates whenever the agent incurs a safety infraction. If the agent fails at a certain segment, it will respawn, stationary at the beginning of the next segment. If the agent successfully completes a segment, it will continue on to the next segment carrying over the current speed. Success rate serves as a proxy for safety, and its relationship with the number of safety infractions is made explicit in Eqn. 2. Thus, a higher success rate is better. Success rate is a newly-introduced metric, which intends to improve upon Episode Completion Percentage (ECP) originally defined in Herman et al. (2021). An episode here refers to one lap. ECP measures the percentage of a lap the agent successfully completes when spawned once at the starting line. In our experience, ECP has large variance due to the length of an episode / lap. For reference, a single lap in `Track01:Thruxton` is 3.8km, whereas CARLA, the *de facto* environment for urban driving research, has in *total* 4.3km of drivable roads in their original benchmark (Codevilla et al., 2019). Formally, we define success rate as follows:

$$\text{Success Rate} = \frac{\text{\# Completed Segments}}{\text{Total Number of Segments}} \times 100\% \tag{1}$$

$$\text{Success Rate} + \frac{\text{\# Safety Infractions}}{\text{Total Number of Segments}} = 100\% \tag{2}$$

Average Adjusted Track Speed. Average speed is defined as the total distance traveled over time (Eqn. 3), which serves as a proxy for performance. As this is *Formula*-style racing, higher is better; this is the same as Average Adjusted Track Speed (AATS) in Task 1.0 (Herman et al., 2021).

$$\text{Average Speed} = \frac{\text{Total Distance Traveled}}{\text{Total Time}} \tag{3}$$

Total Number of Safety Infractions. The total number of safety infractions (NSI) is accumulated during the 1-hour ‘practice’ period in Stage 2 of the competition. The agent is considered to have incurred a safety infraction if 2 wheels of the vehicle leave the drivable area, the vehicle collides with an object, or does not make sufficient progress (e.g. get stuck). In Learn-to-Race, the episode terminates upon a safety infraction. A smaller number of safety infractions is better, i.e. the agent is safer. This is a newly-introduced metric to take into consideration that an autonomous agent should remain safe throughout its interaction with the environment (Ray et al., 2019).

5.2 Task 2.0 Evaluation Procedure: Cross-domain Model Transfer

Agent assessment is conducted through leaderboard competition, with two distinct stages: *training* and *evaluation*. During the training phase, agents have access to the Thruxton Circuit racetrack (`Track01:Thruxton`) for model training and the Trac Môn Anglesey National Circuit racetrack (`Track02:Anglesey`) for model validation, as well as access to all camera sensors and configurations. During Evaluation, agents will be deployed on the unseen track, North Road at Las Vegas Motor Speedway (`Track03:Vegas`), and required to maximise the SR and AATS metrics, subject to NSI being as close to zero infractions as possible. The Evaluation stage, itself, is sub-divided into two sub-phases: pre-evaluation and final evaluation. During pre-evaluation, an agent is given a 1-hour ‘practice period’, in which access to all camera sensor types and the execution of model weight updates (i.e., for learning-based approaches) are permitted; in the final evaluation sub-phase, models weights are frozen for inference and access to only RGB cameras is permitted.

Results of model performance may be provided on the basis of three task execution paradigms—In-domain performance, Cross-domain validation, and Cross-domain test/deployment—against all the metrics in Section 5.1.

In-domain performance: {`Track01:Thruxton`}→{`Track01:Thruxton`} transfer task, where the goal is to maximises an agent's performance on a single track.

Cross-domain validation: {`Track01:Thruxton`}→{`Track02:Anglesey`} transfer task, where the goal is to maximises an agent's cross-domain performance on a validation track.

Cross-domain deployment (leaderboard submission): {`Track01:Thruxton`}→{`Track03:Vegas`} transfer task, where the goal is to maximises an agent's cross-domain performance on a test track. This option is only available via leaderboard submission, where evaluation on `Track03:Vegas` is done automatically. Leaderboard and `L2R` Challenge information is provided in Section 6.

6. `Learn-to-Race` Autonomous Racing Virtual Challenge

The first `Learn-to-Race` Autonomous Racing Virtual Challenge (L2R-ARVC) leaderboard is offered for free as a service to the research community, thanks to the generosity of our sponsors and collaborators. The objective of this instance of the `Learn-to-Race` challenge is to push the boundary of autonomous technology, with a focus on jointly maximising safety, performance, and generalisability in autonomous driving. We specifically highlight the `L2R` Task 2.0 benchmark (Section 5), as we see high-fidelity autonomous racing simulation as a particularly challenging proving ground for autonomous systems–as they are required to adapt to new environments, while making fast decisions, where any safety infraction could have devastating ramifications.

6.1 Challenge Overview

This competition follows from the procedure outlined in the `L2R` Task 2.0 benchmark description (Section 5.2) and consists of two stages: in Stage 1, participants develop their agents on the Thruxton Circuit track (`Track01:Thruxton`) to drive as fast as possible, while adhering to the safety constraints; in Stage 2, we test the agents' ability to adapt to a new environment, with a fixed time budget for safe exploration. Participants will be evaluated on an unseen track (`Track03:Vegas`), with the opportunity to 'practice' with unfrozen model weights for a 1-hour period prior to final evaluation. During the practice period, the number of safety infractions will be accumulated as one of the evaluation metrics, under the consideration that an autonomous agent should remain safe throughout its interaction with the environment (Ray et al., 2019). After the 'practice' period, agents are evaluated on the unseen track, against the Task 2.0 benchmark metrics (Section 5.1).

Participants are required to use high-dimensional visual data as inputs. `Learn-to-Race` provides access to customizable, multimodal sensory inputs. One can access RGB images from any specified location, semantic segmentation, and vehicle states (e.g. pose, velocity). During local development, the participants may use any of these inputs; during final evaluation, the agents will ONLY have access to speed and RGB images from cameras placed on the front, right, and left of the vehicle.

Figure 3: `Learn-to-Race` Autonomous Racing Virtual Challenge overview.

6.2 Challenge Infrastructure

We have solicited sponsorship from companies such as Amazon Web Services, which enables participants to partake in the competition regardless of the personal compute they have available. The evaluation phase is also supported separately by AWS with 15,000 USD. The competition is hosted on AICrowd, a popular website that enables data science experts and enthusiasts to collaboratively solve real-world problems, through challenges. AICrowd has previously hosted competitions for NeurIPS, OpenAI, Spotify, Uber, Stanford University and UNICEF. Carnegie Mellon University has also generously funded in cash and kind with students helping support the competition.

Teams are provided with a time budget (currently 300 hours or approximately 20 training runs to convergence) to train their submissions on AWS, and upload for evaluation to AICrowd. Each submission will be evaluated in AWS using a g4dnxlarge/T4 instance. This gives users access to a dedicated node with a modern GPU (Nvidia T4 or better) and CPU. Teams are provided a number of submissions (currently 20 submissions) for a given month. Both budgets are automatically refilled every month. The organisers of the `Learn-to-Race` Autonomous Racing Virtual Challenge leaderboard reserve the right to assign an additional budget to a team. The organisers also reserve the right to modify the default values of the monthly time budget and/or the number of submissions.

6.3 Challenge Results

Official evaluation of Challenge submission is performed in accordance with the L2R Task 2.0 benchmark metrics (Section 5.1) and evaluation procedures (Section 5.2). We further describe Challenge-specific logistics and results.

Challenge stages. In the Stage 1 (training) phase, participants develop their agents locally, using `Track01:Thruxton` and optionally `Track02:Anglesey`, then submit their trained models for evaluation on `Track01:Thruxton`. The submissions will first be ranked on the SR metric, then submissions with the same SR will be ranked on the basis of AATS. The ten (10) top-ranking teams will be permitted to progress to Stage 2. In Stage 2, the top-ranking participants from Stage 1 will develop cross-domain transfer methods, using `Track01:Thruxton` and/or `Track02:Anglesey`; participants will then submit their models for evaluation on the AICrowd server, on the unseen `Track03:Vegas`. Here, agents are provided with a 1-hour practice period, where access to all camera sensor types and the execution of model weight updates (i.e., for learning-based approaches) are permitted; afterwards, the weights of the approach (at the end of the 1-hour practice phase) are frozen for inference and the the approach is restricted to the use of only RGB camera sensors. Following Stage 2, submissions from the participating teams will first be ranked on success rate, and then submissions with the same SR will be ranked on a weighted sum of the NSI and AATS, based on Eqn. 4, where the subscript i denotes metrics from each participating team:

$$\frac{\text{AATS}_i}{\max_i(\text{AATS}_i)} + \max\left(100\% - \frac{\text{NSI}_i}{\text{median}_i(\text{NSI}_i)}, -100\%\right) \tag{4}$$

Multiple leaderboards in each stage. In order to maintain fair comparison, approaches are split across multiple leaderboards, based on how much multimodal/multiview information is used as input: the results from approaches that use single RGB cameras during Stage 2 evaluation are recorded on a different leaderboard form those approaches that use more than one RGB cameras (e.g., RGB-left, RGB-front, and RGB-right) during Stage 2 evaluation. We also provide approach descriptions for the top-ranking method in each category, below.

Leaderboard rankings. Leaderboard rankings are provided in tables 1 and 2; for conciseness, we show SR, AATS, and NSI for each participant/team, across multiple Challenge stages and camera configurations. We also provide approach descriptions of top-ranking participants, below:

- *Top team on single-camera leaderboard (lachlan_mares)*: During stage 1 of the competition multiple approaches to solving the problem were evaluated; investigations conducted included: multiple reinforcement learning algorithms (in both discrete and continuous space), track localisation using monte-carlo, Bayesian and machine learning methods and classical vehicle controllers. However, given the constraints of Stage 2, in terms of cross-domain generalisation requirements and given the reduced set of permitted camera sensors, a combination of these methods with the best generalisation capabilities was chosen. The Stage 2 solution used two neural networks to feed information to separate steering and acceleration controllers. The first such model was a semantic segmentation model that required low inference time and high accuracy; the architecture chosen was a custom implementation consisting of an EfficientNet-V2-Small encoder, paired with a Feature Pyramid Network (FPN) decoder. The segmentation model was trained using 384x512 augmented greyscale images,

Table 1: Results from the `Learn-to-Race` Autonomous Racing Virtual Challenge Leaderboard, for the Stage 1 "`Track01:Thruxton`→`Track01:Thruxton`" in-domain task paradigm.

`Single-camera`

Participant	Success Rate (SR; %) (↑)	Speed (AATS; KPH) (↑)	Infractions (NSI) (↓)	# Entries
saleh9292	0.500	117.875	6.000	4
White-Wolf	0.700	53.115	1.000	1
SS	0.700	59.953	2.000	2
shan_osphere	0.700	60.968	4.000	1
number9473	0.700	64.448	4.000	2
kire	0.800	42.943	2.000	7
NotSoLate	0.900	32.485	2.000	18
jiangwen_su	0.900	57.615	1.000	20
agnprz	0.900	69.045	2.000	2
AnimeshSinha1309	0.900	69.045	2.000	2
kobe_bb	0.900	78.910	2.000	4
boliu0	1.000	36.140	0.000	4
avrl	1.000	63.080	0.000	1
denis9	1.000	72.000	0.000	16
any_name	1.000	80.760	0.000	11
ling_thoth	1.000	93.940	0.000	6
TCS_Autoscape	1.000	95.960	0.000	60
matthew_howe	1.000	102.010	0.000	12
UniTeam	1.000	105.350	0.000	27
xLab_UPenn	1.000	115.660	0.000	2
lachlan_mares	1.000	126.350	0.000	15
Downforce615	1.000	137.940	0.000	39
Werner_Duvaud	1.000	152.090	0.000	15

`Multi-camera`

Participant	Success Rate (SR; %) (↑)	Speed (AATS; KPH) (↑)	Infractions (NSI) (↓)	# Entries
Downforce615	1.000	137.230	0.000	1

rather than RGB, to reduce the domain gap between the Thruxton, AngleseyNational, and VegasNorthRoad tracks. The classifier was a fully-connected network, which took a latent vector from the segmentation model and classified the track into one of three zone types. The zone types permitted the use of multiple parameter settings for the steering and acceleration controllers. The steering and velocity controllers both rely on predictions from the segmentation model. An estimate of the track centerline and track curvature was calculated using the boundaries of the predicted road surface: the zone classifier allowed relaxed parameters in high-speed regimes, so that the vehicle could exceed speeds permitted by only relying on the camera's field of view.

- *Top team on multi-camera leaderboard (matthew_howe)*: We provide a baseline that can, in a single lap, safely adapt to an unseen race-track and achieve competitive lap times. Our solution consists of a perception module which informs a model predictive controller. The perception module is responsible for processing observations in the form of a camera feed into a representation of the local limits of the road. These road limits and additional safety constraints are used by the model predictive controller to provide steering and acceleration inputs. *Perception*—Using ground truth segmentation masks provided by the simulation on the Thruxton circuit, a Feature Pyramid Network (FPN) [Lin et al.2017] with an EfficentNet [Tan and Le2019] backbone was trained to classify whether a given image pixel belongs to the

Table 2: Results from the Learn-to-Race Autonomous Racing Virtual Challenge Leaderboard, for the Stage 2 "Track01:Thruxton→Track03:Vegas" cross-domain task paradigm.

Single-camera				
Participant	Success Rate (SR; %) (↑)	Speed (AATS; KPH) (↑)	Infractions (NSI) (↓)	# Entries
xLab_UPenn	0.000	31.098	10.333	32
TCS_Autoscape	0.100	4.485	4.333	38
denis9	0.667	64.889	3.667	24
any_name	1.000	30.44	0.000	6
Werner_Duvaud	1.000	45.253	0.000	40
UniTeam	1.000	73.187	0.000	45
matthew_howe	1.000	85.22	0.000	33
lachlan_mares	1.000	92.527	0.000	34

Multi-camera				
Participant	Success Rate (SR; %) (↑)	Speed (AATS; KPH) (↑)	Infractions (NSI) (↓)	# Entries
UniTeam	0.667	62.094	1.000	8
any_name	1.000	51.373	0.000	3
lachlan_mares	1.000	80.723	0.000	6
matthew_howe	1.000	84.227	0.000	1

drive-able portion of the race track or not. Generalisation across racetracks was achieved via data augmentation. Specifically a combination of horizontal flipping and brightness changes were found to enable acceptable inference performance on the Anglesey racetrack. The segmentation mask boundaries are projected onto the ground plane using camera intrinsic and extrinsic calibrations to yield local track boundaries. To smooth the often jagged mask boundaries, this module uses cubic polynomial fitting to produce the final track limits used downstream. *Control*—A model predictive controller [Mats Steinweg2020] takes observed track limits and plans a path that both stays on the track and achieve high speeds. Given a set of physical constraints, a spatial bicycle model is used to predict achievable values for yaw and velocity. The controller plans a series of future states that are within track limits, gets to the last point on the trajectory as quickly as possible, do no exceed maximum lateral acceleration and be able to slow down to a minimum cornering speed by the end of the observed trajectory. Both the lateral acceleration limit and cornering speed were tuned manually by observing vehicle behaviour on the training circuits. A lateral acceleration limit of 2.0m/s2 was found to rarely exceed grip limits and a minimum cornering speed of 10m/s ensured that excess velocity was not carried into corners. The lateral acceleration limit serves to reduce cornering speed; leading to less opportunity for loss of control. Enforcing cornering speeds prevent the vehicle entering turns too dangerously, such as diving deep and exceeding track limits. Combining these hand-crafted constraints led to more reliable and consistent cornering behaviour.

7. Insights + Future Directions

Towards safe reinforcement learning. Racing demands each vehicle to drive at its physical limits, when any safety infraction could lead to catastrophic failure. We encourage the study safe reinforcement learning (RL) for autonomous racing, where autonomous agents must: identify and avoid unsafe scenarios under the complex vehicle dynamics, and make sub-second decisions in a fast-changing environment. This thrust focuses on the extension, development, and benchmarking of new safe learning algorithms for autonomous racing.

Towards fast and generalisable visual feature-extraction. Autonomous systems must carefully characterise their environment, in order to identify unsafe states and avoid obstacles. In the context of autonomous racing, agents are endowed with visual perception of their environment, from which they must extract meaningful features (e.g., distances to road boundaries, other agents, and obstacles), while remaining invariant to irrelevant features (e.g., shadows, glare, sky, vegetation). Whereas we would like to use a large-capacity visual feature extractor for this purpose, we are limited in the autonomous racing tasks by constraints on inference speed. Moreover, even though we want to maximise performance on the tracks that the agent directly interacts with, we do not want to overfit; the visual encoder must generalise to unseen contexts. This thrust focuses on the further development of new visual encoding strategies for autonomous racing.

Distributed training and optimisation. The optimisation of a reinforcement learning agent depends on the comprehensiveness of its exploration, during interaction with its environment. Prior art pursued distributed reinforcement learning architectures, to maximise environment interactions of some performance oriented policy. In the context of safe reinforcement learning for autonomous racing (Chen et al., 2021), we have *two* coupled policies to optimise (i.e., a safety policy and a performance policy), yielding multiple options and hierarchies for distributed training. This development thrust focuses on implementing and experimenting with different distributed reinforcement learning training paradigms, e.g.: multiple model instances + single environment, single model instance + multiple environments, centralised versus decentralised replay memory, centralised versus decentralised safety backup policies/critics/experts, heterogeneous types of policies (on-policy, off-policy) and safety critics (cost-limit regressor, rules-based expert, learnable safety critic), etc.

Towards effective transfer imitation learning. Recent work in urban driving settings couple imitation learning (IL) objectives with driving policies that have knowledge of the underlying action prior distribution. We hypothesise that this distribution-awareness would provide agents with robustness to noise artifacts in the training data, would provide a window into agents' intentions for more interpretable predictions, would yield improved unseen generalisation, and would help bypass common issues in imitation- and transfer learning such as causal confusion and negative transfer. An interesting direction for future work is in the selection of more informative priors, e.g., those that incorporate logical rules for appropriate multi-agent interaction on the racetrack. Another important direction for future work is in characterising the causal relationships between observations, actions, and rewards in a scene. This causal structure can be represented *explicitly*, by way of causal knowledge graphs (enabling explainability and counterfactual reasoning), or it can be represented *implicitly*, by way of learning identifiable latent representations. Regardless of explicit or implicit representation, the goal would be to eliminate extraneous connections/edges (disturbances, confounders) in the underlying causal structure, thereby reducing causal confusion when transferring an observation model from a source domain to a target domain.

Planned extensions of the `L2R` platform:

- *Towards simulation-to-real transfer.* The `Learn-to-Race` framework will interface with the real-world vehicle software stacks used in the Roborace Challenge and Indy Autonomous

Challenge competitions, via the Robot Operating System (ROS). Our goal is to support flexible transfer of algorithms—from simulation in L2R to real-world autonomous racing vehicles.

- *Future framework extensions.* We believe that a primary direction for future tasks must be towards higher simulation complexity and realism. The Learn-to-Race framework will feature new evaluation procedures and test-cases that match upcoming Roborace competition scenarios, such as virtual static/dynamics obstacles in Roborace Metaverse and multi-agent settings, across many new tracks; we will introduce new tasks and challenges, accordingly.
- *Future outreach and modes of engagement with L2R.* The Learn-to-Race team wishes to continue outreach with the scientific community, through further publications, challenges, and collaborations. We also organise and sponsor workshops at notable scientific publication venues, such as the *Workshop on Artificial Intelligence for Autonomous Driving* (AI4AD; co-located with the *International Joint Conference on Artificial Intelligence*, IJCAI 2022, in Vienna) and the *Workshop on Safe Learning for Autonomous Driving* (SL4AD; co-located with the *International Conference on Machine Learning*, ICML 2022, in Baltimore).

8. Conclusions

In this paper, we summarised the results of the inaugural Learn-to-Race Autonomous Racing Virtual Challenge, which encouraged interdisciplinary research and development towards safe, performant, generalisable autonomous systems. We introduce the L2R Task 2.0 benchmark, with new metrics and task definitions. Then we provide an overview of the deployment, evaluation, and rankings, which used the new L2R Task 2.0 benchmark and received over 20,100 views, 437 participants, 46 teams, and 733 model submissions—from 88 unique institutions, in 28 countries.

Acknowledgments

There are several entities that the authors would like to thank — principally, the *Masters in Computational Data Science* (MCDS) program, in the School of Computer Science, at Carnegie Mellon University: MCDS provided student development support, via capstone projects that were proposed and led by JF, SK, SGa, AK, and EN. We thank AICrowd and Amazon AWS for extensive challenge sponsorship and support, and we thank the Challenge participants themselves for their engagement and feedback! In particular, we thank the *lachlan_mares* and *matthew_howe* participants for contributing their approach descriptions. This work was supported, in part, by a doctoral research fellowship from Bosch Research. The views expressed in this article do not necessarily represent those of the aforementioned entities. JF is the corresponding author, reachable at *jmf1@alumni.cmu.edu*.

References

Balaji, B., Mallya, S., Genc, S., Gupta, S., Dirac, L., Khare, V., Roy, G., Sun, T., Tao, Y., Townsend, B., et al. (2020). Deepracer: Autonomous racing platform for experimentation with sim2real reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2746–2754. IEEE.

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., and Zaremba, W. (2016). Openai gym.

Cai, P., Wang, H., Huang, H., Liu, Y., and Liu, M. (2021). Vision-based autonomous car racing using deep imitative reinforcement learning. *IEEE Robotics and Automation Letters*, 6(4):7262–7269.

Chen, B., Francis, J., Oh, J., Nyberg, E., and Herbert, S. L. (2021). Safe autonomous racing via approximate reachability on ego-vision.

Chen, C., Seff, A., Kornhauser, A., and Xiao, J. (2015). Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE international conference on computer vision*, pages 2722–2730.

Chen, D., Zhou, B., Koltun, V., and Krähenbühl, P. (2020). Learning by cheating. In *Conference on Robot Learning*, pages 66–75. PMLR.

Chisari, E., Liniger, A., Rupenyan, A., Van Gool, L., and Lygeros, J. (2021). Learning from simulation, racing in reality. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8046–8052. IEEE.

Codevilla, F., Müller, M., López, A., Koltun, V., and Dosovitskiy, A. (2018). End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4693–4700. IEEE.

Codevilla, F., Santana, E., López, A. M., and Gaidon, A. (2019). Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9329–9338.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017a). Carla: An open urban driving simulator. In *Conference on robot learning*, pages 1–16. PMLR.

Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., and Koltun, V. (2017b). CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, pages 1–16.

Drews, P., Williams, G., Goldfain, B., Theodorou, E. A., and Rehg, J. M. (2017). Aggressive deep driving: Combining convolutional neural networks and model predictive control. In *Conference on Robot Learning*, pages 133–142. PMLR.

Florian, F., Yunlong, S., Kaufmann, E., Scaramuzza, D., and Duerr, P. (2020). Super-human performance in gran turismo sport using deep reinforcement learning.

Francis, J., Kitamura, N., Labelle, F., Lu, X., Navarro, I., and Oh, J. (2022). Core challenges in embodied vision-language planning. *Journal of Artificial Intelligence Research*, 73(1).

Herman, J., Francis, J., Ganju, S., Chen, B., Koul, A., Gupta, A., Skabelkin, A., Zhukov, I., Kumskoy, M., and Nyberg, E. (2021). Learn-to-race: A multimodal control environment for autonomous racing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9793–9802.

Indy (2019). Indy autonomous challenge. https://www.indyautonomouschallenge.com/. Last accessed: 2021-01-30.

Kabzan, J., Hewing, L., Liniger, A., and Zeilinger, M. N. (2019). Learning-based model predictive control for autonomous racing. *IEEE Robotics and Automation Letters*, 4(4):3363–3370.

Liniger, A., Domahidi, A., and Morari, M. (2015). Optimization-based autonomous racing of 1: 43 scale rc cars. *Optimal Control Applications and Methods*, 36(5):628–647.

Loiacono, D., Cardamone, L., and Lanzi, P. L. (2013). Simulated car racing championship: Competition software manual. *arXiv preprint arXiv:1304.1672*, 73(1).

Ohn-Bar, E., Prakash, A., Behl, A., Chitta, K., and Geiger, A. (2020). Learning situational driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11296–11305.

O'Kelly, M., Zheng, H., Karthik, D., and Mangharam, R. (2020). F1tenth: An open-source evaluation environment for continuous control and reinforcement learning. *Proceedings of Machine Learning Research*, 123.

Prakash, A., Chitta, K., and Geiger, A. (2021). Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087.

Quigley, M., Conley, K., Gerkey, B., Faust, J., Foote, T., Leibs, J., Wheeler, R., Ng, A. Y., et al. (2009). Ros: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan.

Ray, A., Achiam, J., and Amodei, D. (2019). Benchmarking safe exploration in deep reinforcement learning. *arXiv preprint arXiv:1910.01708*, 7.

RoboRace (2015). Roborace. https://roborace.com/. Last accessed: 2021-01-30.

Rosolia, U., Carvalho, A., and Borrelli, F. (2017). Autonomous racing using learning model predictive control. In *2017 American Control Conference (ACC)*, pages 5115–5120. IEEE.

Strobel, K., Zhu, S., Chang, R., and Koppula, S. (2020). Accurate, low-latency visual perception for autonomous racing: Challenges, mechanisms, and practical solutions. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1969–1975. IEEE.

Tatiya, G., Francis, J., Navarro, I., Kitamura, N., Nyberg, E., Sinapov, J., and Oh, J. (2022). Knowledge-driven scene priors for semantic audio-visual embodied navigation.

TORCS (1997). Torcs, the open racing car simulator. http://torcs.sourceforge.net/index.php?name=Sections&op=viewarticle&artid=19. Last accessed: 2021-01-30.

Weiss, T. and Behl, M. (2020). Deepracing: a framework for autonomous racing. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1163–1168. IEEE.

Wurman, P. R., Barrett, S., Kawamoto, K., MacGlashan, J., Subramanian, K., Walsh, T. J., Capobianco, R., Devlic, A., Eckert, F., Fuchs, F., et al. (2022). Outracing champion gran turismo drivers with deep reinforcement learning. *Nature*, 602(7896):223–228.

Yurtsever, E., Lambert, J., Carballo, A., and Takeda, K. (2020). A survey of autonomous driving: Common practices and emerging technologies. *IEEE access*, 8:58443–58469.

Zhang, J. and Ohn-Bar, E. (2021). Learning by watching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12711–12721.

Zhang, Z., Liniger, A., Dai, D., Yu, F., and Van Gool, L. (2021). End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15222–15232.