

# FastRLAP: A System for Learning High-Speed Driving via Deep RL and Autonomous Practicing

## Appendix

Anonymous CoRL 2023 Submission, Paper ID 10.

### A Hyperparameters

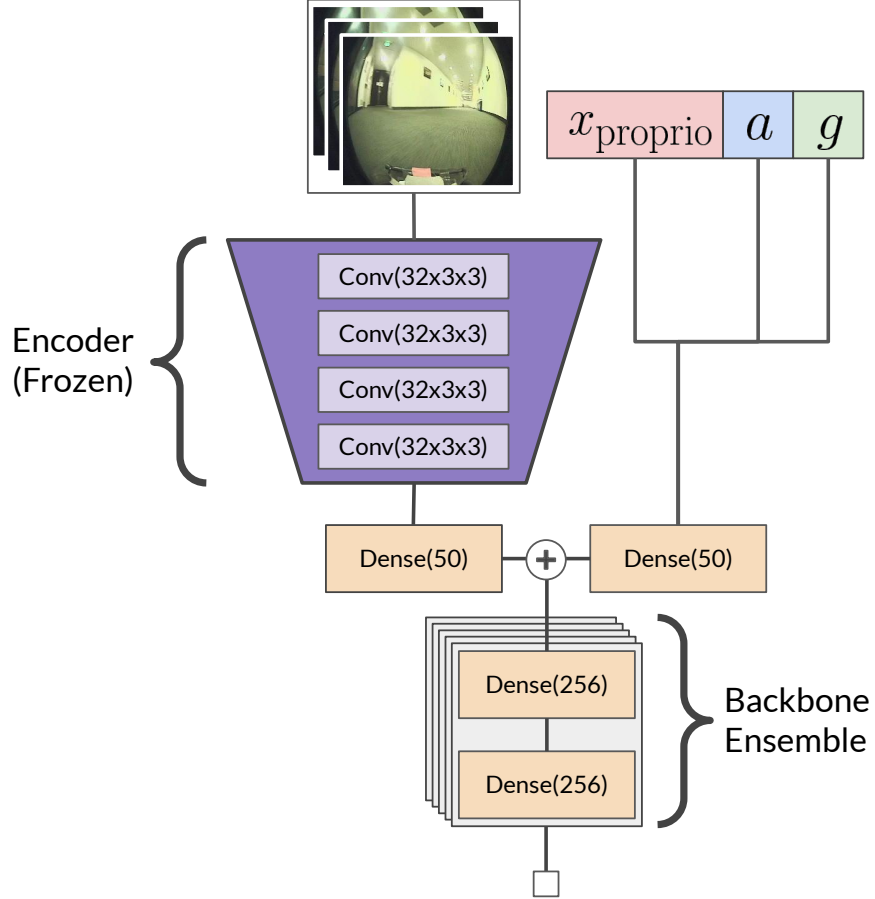
All our experiments use the *same* set of hyperparameters and there is no environment-specific tuning used in the results presented in the paper. See Tab. 1 for a list of hyperparameters and Fig. 1 for a detailed network architecture.

Category	Hyperparameter	Value
Actor/Critic	Actor learning rate	3e-4
	Critic learning rate	3e-4
	Temperature learning rate	3e-4
	Actor network architecture	2x256
	Critic network architecture	2x256
	Initial target entropy	-3
	Entropy decay rate	1e-5
	Critic ensemble size	10
MDP/System	Discount factor	0.99
	Time step	0.1s
	Velocity target range (m/s)	[0.5, 3.5]
	Servo target range (rad)	[-0.5, 0.5]
	$C_{\text{collide}}$ (real only)	$0.2s^2/m$
	$C_{\text{stuck}}$	-10
	Squashing range $\delta$	0.2
Encoder	Layer count	4
	Convolution size	3x3
	Stride	2
	Hidden channels	32
IQL	Expectile	0.7
	Value network structure	same as critic

**Table 1:** List of hyperparameters used throughout experiments

### B Additional Experimental Details

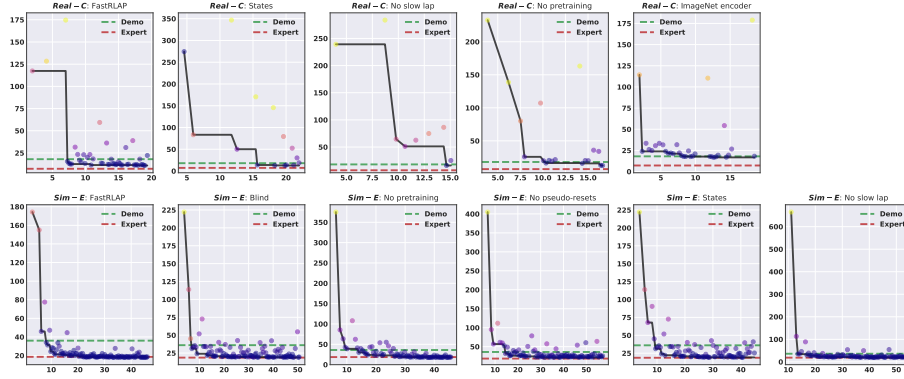
In addition to the metrics presented in the main paper, please see Tab. 2 for a list of additional metrics per experiment, and Fig. 2 for lap time progression charts for each experiment. FastRLAP



**Figure 1:** Network architecture for the critic. Actor (and, for IQL, value) architectures are identical, but with only one backbone rather than an ensemble. Proprioceptive information is concatenated with the action and the goal and fed through a dense layer, concatenated with the output of a convolutional encoder applied to a sequence of three camera images, and fed through a 2-layer MLP.

Course name	Method	Lap time (s)			median (5)			best	total	# collisions			mean (5)	median (5)	
		first	best	median						mean	median				
Indoor-A	FastRLAP	114.31	32.74	49.51	38.99			0	82	2.48	2		2.00		0
Indoor-B	FastRLAP	224.70	44.21	73.71	65.72			0	135	7.50	7		4.20		3
Indoor-C	States	274.42	12.70	50.13	18.88			0	190	12.67	4		3.40		2
Indoor-C	Ours	117.38	10.90	13.27	11.69			0	126	2.74	0		0.00		0
Indoor-C	No slow lap	239.21	16.01	64.51	62.62			0	206	22.89	12		8.80		11
Indoor-C	Human Oracle	54.40	7.21	10.08	8.79			0	7	1.00	0		0.00		0
Indoor-C	ImageNet encoder	49.75	19.66	30.20	21.12			0	168	6.46	1		0.00		0
Indoor-C	No pretraining	232.79	12.70	21.60	19.99			0	174	9.67	1		1.80		1
Outdoor-D	FastRLAP	196.01	17.80	23.20	19.50			0	77.0	1.0	0		0.00		0
Outdoor-E	FastRLAP	133.94	62.10	98.65	82.39			0	63.0	3.31	3.0		2.6		3.0
Sim-F	FastRLAP	925.08	104.19	107.00	107.00			0	157	4.76	0		0.00		0
Sim-G	Blind	222.17	18.89	21.70	19.50			0	127	1.18	0		1.20		0
Sim-G	States	222.17	19.10	23.69	26.20			0	113	1.30	0		1.00		1
Sim-G	FastRLAP	174.30	17.99	19.10	18.10			0	48	0.42	0		0.00		0
Sim-G	No slow lap	665.04	19.70	23.33	22.20			0	90	0.95	0		0.00		0
Sim-G	No pretraining	375.26	17.80	21.90	18.39			0	100	1.14	0		0.00		0
Sim-G	No pseudo-resets	405.02	21.70	26.31	25.10			0	156	1.64	0		0.20		0
Sim-G	Dino	322.98	31.19	35.74	32.80			0	174.0	9.66	1.0		1.8		1.0

**Table 2:** Detailed statistics of all runs



**Figure 2:** Detailed laptime progression charts for all baselines

outperforms baselines in *all* environments, as measured by any suitable metric — time-to-first lap, best lap time, mean/median lap times, and minimum number of average and best-case collisions.

## C Implementation Details

The overall system was implemented using ROS 1 Noetic Ninjemys, with the inference and training code using JAX. We transferred tensors between the components of our system (new data from the robot to the workstation, and parameters from the workstation to the robot) using ROS messages.

## D System Details

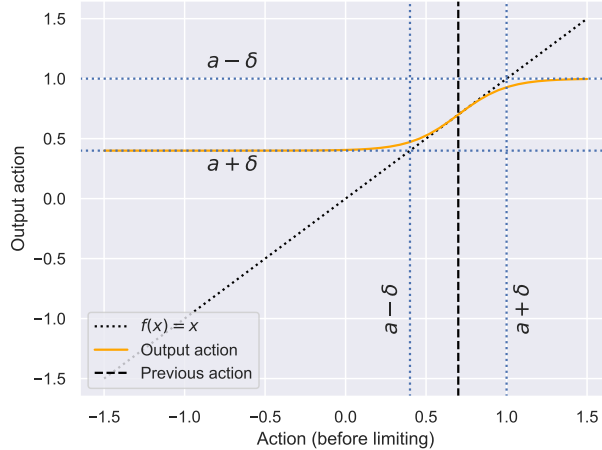
As mentioned in the paper, we squash the output of the actor dynamically to a range around the previous action  $a_{\text{prev}}$  to enforce continuity in output actions by constraining them to be no more than some positive constant  $\delta$  away from the previous action in each dimension. We choose our activation function so that the interval  $[a_{\text{prev}} - \delta, a_{\text{prev}} + \delta]$  is roughly mapped to itself. More precisely, we ensure that our activation  $f_{a_{\text{prev}}, \delta}(x)$  is accurate to first-order Taylor expansion around the previous action. This yields the following activation function applied to the output of the actor:

$$f_{a_{\text{prev}}, \delta}(x) = \tanh\left(\frac{(x - a_{\text{prev}})}{\delta}\right) \delta + a_{\text{prev}}$$

Figure 3 shows a graphical depiction of this activation function.

## E Qualitative Analysis

To examine whether the learned policy relies on visual cues or memorizes action sequences, we evaluate  $Q(s, a)$  for sample images by injecting a range of steering inputs 4. This reveals how the critic network assesses different actions based on visual information. We observe that the critic outputs align with visual observations, assigning lower values to actions steering towards obstacles and higher values to actions maintaining paths instead of tall grass. This correlation suggests that the learned policy makes decisions based on visual features rather than mere memorization, indicating genuine learning from the environment’s visual cues.



**Figure 3:** Our activation function (actor output only) ensures action continuity.

## F Environment Details

**Indoor-A** represents a large loop (70 meters in length) through the interior of a carpeted building with glass walls and many open corridors. The course is defined by a sequence of  $n_c = 4$  checkpoints spaced roughly 15-20 meters apart.

**Indoor-B** is a significantly larger course ( $\sim 120$  meters in length) with multiple obstacles, defined by  $n_c = 4$  checkpoints. The floor of this environment is smooth and has low friction, leading to over/understeer during cornering.

**Indoor-C** is a small but challenging indoor race course with two tight “hairpin” turns, taken at nearly the maximum steering angle and a tight “chicane” (a right-left sequence). This environment requires the robot to discover a fast “racing lines” to minimize steering and carry speed through the turns. We extensively compare FastRLAP to several baselines in this environment.

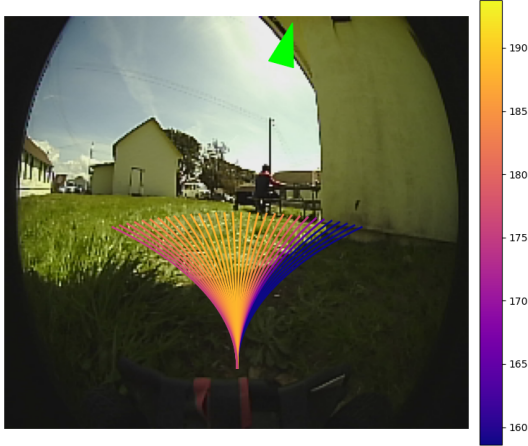
**Outdoor-D** is a medium-scale (60 meter loop) outdoor course around a building. In addition to straightforward obstacle avoidance with the building, a tree, and a nearby table, there are several patches of tall grass which tend to slow down the robot’s motion. A successful policy should avoid the tall grass to the maximum extent possible, staying near paths where the grass is shorter and keeping to the left of the tree when it passes to avoid the grass on the right.

**Outdoor-E** is a large-scale (120 meter loop) outdoor course between a dense grove of trees on one side and a tree and several fallen logs on the other side. The ground near the trees is covered in leaves, sticks, and other loose material, causing highly speed-dependent steering characteristics.

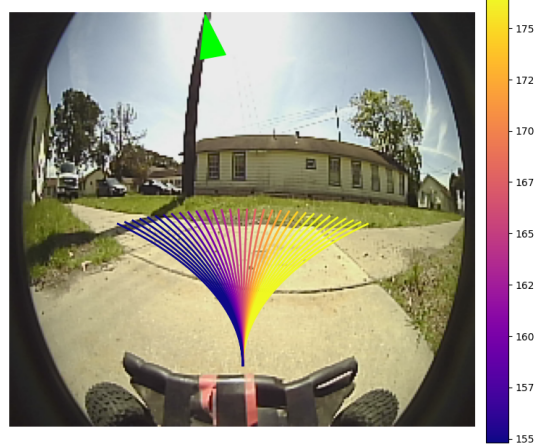
**Sim-F** is a large, complex environment containing a large pool of mud, which greatly limits the robot’s speed and should be avoided, and a narrow bridge that can be used to bypass the mud. FastRLAP successfully learns a high-performance policy in this environment, selecting the optimal path after only a few laps and achieving super-human lap times in under 10 laps. All baselines and ablations failed to solve this task.

**Sim-G** is a medium-scale track with sharp turns and chicanes, much like **Indoor-C**, making it a particularly interesting environment to study the emergence of *racing lines* and agile maneuvers. All baselines were able to solve this task within the allotted time, allowing direct comparison.

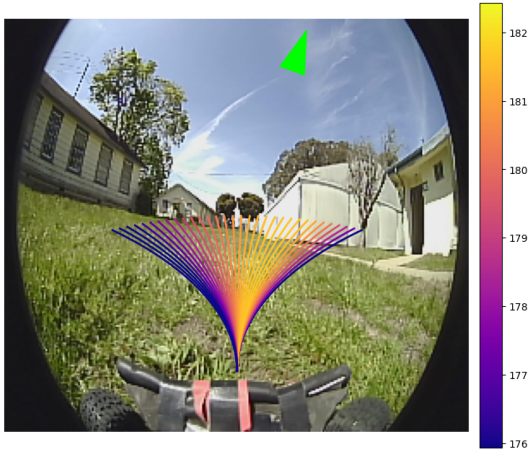
The Clearpath Jackal used for simulations differs from the real environments primarily in its action space, which (as a differential drive robot) allows turning in place. We limit the linear velocity actions of the robot to  $[-1, 2]$  and the angular velocity actions to  $[-1.0, 1.0]$ . Simulated position measurement is provided in lieu of the RealSense tracker for determining relative goal



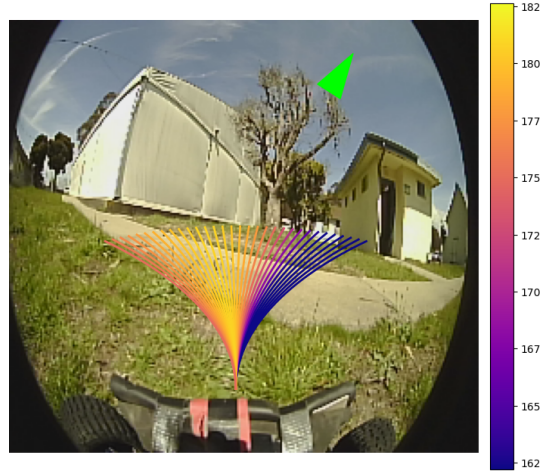
(a) The critic assigns very low value to actions that would cause contact with an obstacle.



(b) The critic suggests turning towards the next checkpoint before the current checkpoint is reached.

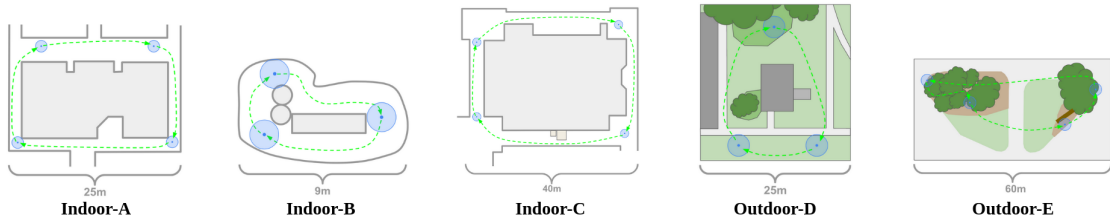


(c) In wide-open areas the critic reflects a beeline policy directly towards the next goal.



(d) The critic prefers turning left rather than the geometric shortest-path on the right to avoid tall grass.

**Figure 4: Qualitative critic evaluations** on sample images from **Outdoor-D** with steering actions represented as (approximate) paths overlaid. Hot (cold) colors represent high (low)-value actions. The green arrow points towards the next checkpoint.



**Figure 5:** Schematics for each test environment in the real-world. FastRLAP is able to learn to drive real-world courses of varying size in both indoor and outdoor settings. *Note that these schematics are not available to our system and are shown only for illustration.*

locations.

## G Code Release

Please see [sites.google.com/view/fastrlap](https://sites.google.com/view/fastrlap) for the training and robot-side inference code as well as modified simulation environments.