ICRA2026 Late Breaking Results Poster presented at
2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

Beyond Reactive Adaptation: Long-Horizon Memory for Autonomous Racing via State Space Models

Grzegorz Czechmanowski, Jan Wegrzynowski, Piotr Kicki, Krzysztof Walas

***Abstract*—Autonomous racing pushes vehicles to their physical limits, requiring control policies that can rapidly adapt to localized changes in track conditions, such as varying surface friction. Current Reinforcement Learning (RL) approaches rely either on ground-truth system identification, which is impractical in the real world, or short-horizon reactive adaptations (e.g., Rapid Motor Adaptation (RMA)) that cannot remember spatial disturbances across multiple laps. In this extended abstract, we propose a novel RL architecture based on Mamba, a structured State Space Model (SSM), for autonomous racing. By fusing vehicle state with Fourier features of vehicle position on the racetrack, our Mamba-based policy builds a long-horizon episodic memory. This allows the policy not only to adapt to unknown friction online but also to map and memorize slippery zones for future laps. Evaluated in a simulated F1Tenth environment, our approach demonstrates continuous lap-to-lap improvement, approaching the performance of an "oracle" policy trained on exact ground-truth friction, whereas standard Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN) baselines plateau at inferior performance levels.**

I. Introduction

Pushing autonomous vehicles to their dynamic limits requires precise understanding of tire-road interactions. In autonomous racing, slight miscalculations in friction estimation can lead to catastrophic failure. Standard RL pipelines typically address sim-to-real gap through Domain Randomization (DR) [1]. However, purely reactive DR policies are inherently conservative, sacrificing optimal lap times to maintain stability across all possible randomized parameters [2].

Recent state-of-the-art approaches attempt to close this gap via explicit adaptation. Methods relying on exact system identification, from on-policy data, often require ground-truth data from external motion capture systems (e.g., OptiTrack) [3], severely limiting deployment in unstructured environments. Alternatively, RMA [4] utilizes a history buffer of recent states to infer latent environmental dynamics online. While effective for immediate, reactive compensation, RMA's fixed, short-horizon context window prevents episodic learning. For instance, if a vehicle encounters a slippery section at Turn 1, a short-horizon policy will adapt during the slide but will "forget" this condition by the time it reaches Turn 1 on the next lap.

To achieve superhuman racing performance, policies must transition from reactive adaptation to proactive exploitation. We propose replacing traditional MLPs and RNNs with Mamba [5], [6], a structured SSM. Mamba offers unbounded context lengths without the vanishing gradient problems of RNNs or the computational bottleneck of Transformers. In this work, we demonstrate that a Mamba-based RL policy, provided with car postion encodings, successfully memorizes localized friction patches online, allowing it to progressively decrease lap times in an F1Tenth racing environment.

All authors are with the Institute of Robotics and Machine Intelligence, Poznan University of Technology, Poznan, Poland, and with IDEAS Research Institute, Warsaw, Poland. This work was supported by an internal PUT grant 0214/SBAD/0256 and DNMK: 0214/SBAD/0257.

II. Methodology

A. Environment and Problem Formulation

We formulate the racing problem as a Markov Decision Process (MDP) in a custom vehicle dynamics simulator. The observation space $o_t \in \mathbb{R}^{21}$ encompasses vehicle kinematics (longitudinal/lateral velocities, yaw rate, wheel speeds), its position relative to track centerline (closest point on the centerline, heading difference, signed distance to centerline/edge), and control inputs. Crucially, we provide the agent with its spatial location via Fourier features of its normalized progress along the track centerline (e.g., $\sin(s), \cos(s)$). This spatial encoding acts as an indexing mechanism, allowing the policy to anchor learned dynamics to specific track regions. The continuous action space $a_t \in [-1, 1]^2$ commands target steering and wheel speed derivative.

To effectively train the racing policy, we designed the reward function to quantify the vehicle's progress along the track while severely penalizing boundary violations. Although the ultimate goal is achieving faster laps, optimizing directly for this sparse metric hinders learning. Therefore, progress along the centerline is used as an effective proxy that provides denser feedback, a formulation commonly employed in RL racing [3]. Specifically, we define the reward as $r_t = s_t - s_{t-1}$ and $r_t = -1$ if a track boundary is exceeded, where s_t denotes the progress along the centerline expressed in Frenet coordinates [7].

B. Mamba-based Policy Architecture

Standard MLPs lack memory and treat observations as isolated snapshots; consequently, they cannot deduce underlying track conditions from past experience. RNNs (e.g., Long Short-Term Memorys (LSTMs)) maintain a hidden state, but their memory horizons struggle with the thousands of timesteps required to complete an F1Tenth lap.

We integrate a Mamba block as the core of both actor and critic networks. Mamba's input-dependent state transitions allow it to selectively compress relevant information (e.g.,

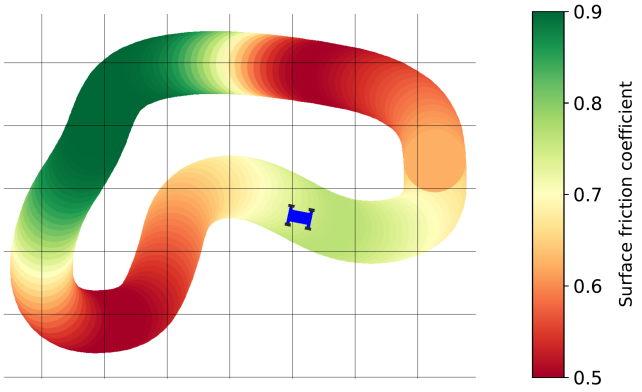

Fig. 1. Visualization of the simulated F1Tenth track used for evaluation. The color gradient illustrates the continuously varying surface friction spline, ranging from high grip (green, $\mu = 0.9$) to slippery patches (red, $\mu = 0.5$). The grid lines represent a 1-meter spacing for scale, and the blue F1Tenth car is included for size reference.

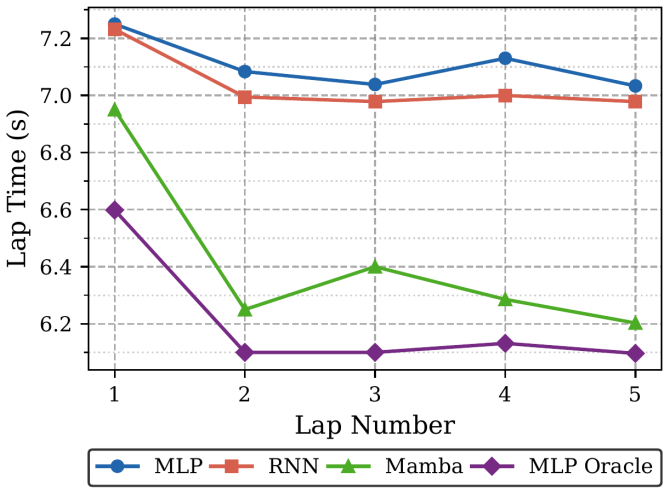

Fig. 2. Lap-to-lap temporal evolution. Mamba successfully utilizes its hidden state to "memorize" the track's friction profile, proactively adjusting control to reduce lap times on consecutive laps.

"loss of grip at $s = 0.2$") into a persistent hidden state and propagate it across the entire episode. The policies are trained using Proximal Policy Optimization (PPO). During training, track friction is heavily randomized to force the policy to map the environment online. Before each training episode, the simulator generates a friction spline using 2 to 5 random control points, with friction values ranging from high grip ($\mu = 1.0$) to slippery ($\mu = 0.5$). Since friction is not observed, the agent must physically "feel" and memorize it. Consequently, the Mamba policy acts as an *in-context meta-learner*, rather than merely memorizing a static driving line, it learns *how to learn* the underlying track characteristics dynamically, mapping unseen friction coefficients online without requiring weight updates.

III. Experimental Setup and Results

We evaluate our method on a simulated F1Tenth track featuring a continuously varying surface friction profile (see Fig. 1). We compare four architectures: 1) MLP, 2) RNN, 3) Mamba, and 4) an Oracle MLP, which is trained exclusively on the exact test track friction (empirically approximating the theoretical performance ceiling). All sequential models (RNN, Mamba) process the exact same observation as MLP, relying entirely on their internal hidden states to retain history.

As shown in Fig. 2, the MLP and RNN policies fail to fully exploit the vehicle's limits. The MLP, lacking memory, drives conservatively to avoid crashing in unknown slippery zones. The RNN fails to assign credit over the long time-horizon of a full lap, resulting in suboptimal state representations. In contrast, the Mamba policy successfully utilizes its memory, achieving peak lap times only 0.1 seconds slower than the ground-truth Oracle.

Fig. 2 highlights our core contribution. Because Mamba fuses the track positional encodings with the localized slip dynamics it experiences on Lap 1, it builds an internal friction map. By operating as a meta-learner, it anticipates conditions on subsequent laps, resulting in a continuous downward trend in lap times that closely matches the oracle. In contrast, the baselines lack this long-horizon spatial memory and exhibit flat, purely reactive lap-to-lap performance.

IV. Conclusion and Future Work

We introduced a Mamba-based RL architecture for autonomous racing that bridges the gap between short-horizon reactive adaptation and long-horizon episodic memory. By tracking vehicle state and its position on track, the SSM effectively meta-learns and memorizes localized track conditions without requiring explicit state estimation. Our results show Mamba progressively improves lap times online, matching oracle baselines. For future work, we intend to deploy this policy zero-shot to a physical F1Tenth vehicle, demonstrating real-world sim-to-real robustness against dynamically changing track surfaces.

References

[1] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain Randomization for Transferring Deep Neural Networks from Sim to Real," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 23–30.

[2] G. Czechmanowski, J. Wegrzynowski, P. Kicki, and K. Walas, "On learning racing policies with reinforcement learning," in *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2025, pp. 6395–6402.

[3] E. Kaufmann, L. Bauersfeld, A. Loquercio, M. Müller, V. Koltun, and D. Scaramuzza, "Champion-level drone racing using deep reinforcement learning," vol. 620, no. 7976, pp. 982–987.

[4] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "RMA: Rapid motor adaptation for legged robots," in *Robotics: Science and Systems*, 2021.

[5] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," *arXiv preprint arXiv:2312.00752*, 2023.

[6] T. Dao and A. Gu, "Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality," in *International Conference on Machine Learning (ICML)*, 2024.

[7] J. F. Frenet, "Sur les courbes à double courbure," *Journal de Mathématiques Pures et Appliquées*, 1852.