Appendix A. Proof of Proposition 3

Consider a policy parameter θ^* such that $\Phi(\mathbf{x}, \theta^*) \geq \max_{\theta \in \Theta} \Phi(\mathbf{x}, \theta) - \lambda$. For any $\theta^i \in \Theta^i$,

$$\begin{aligned} V^i(\mathbf{x}, \theta^{*,i}, \theta^{*,-i}) - V^i(\mathbf{x}, \theta^i, \theta^{*,-i}) &\geq \Phi(\mathbf{x}, \theta^{*,i}, \theta^{*,-i}) - \Phi(\mathbf{x}, \theta^i, \theta^{*,-i}) - \alpha \\ &\geq -\lambda - \alpha, \end{aligned}$$

where first inequality is due to (2) and the second inequality is because θ^* maximizes Φ. The proof follows using the definition of Nash equilibrium (Definition 1).

Appendix B. Description of Single-agent Racing Line

Race drivers follow a racing line for specific maneuvers. This line can be used as a reference path by the motion planner to assign time-optimal trajectories while avoiding collision. The racing line is minimum-time or minimum-curvature. They are similar, but the minimum-curvature path additionally allows the highest cornering speeds given the maximum legitimate lateral acceleration Heilmeier et al. (2020).

There are many proposed solutions to finding the optimal racing line, including nonlinear optimization Rosolia and Borrelli (2020); Heilmeier et al. (2020), genetic algorithm-based search Vesel (2015) and Bayesian optimization Jain and Morari (2020). However, for our work, we calculate the minimum-curvature optimal line, which is close to the optimal racing line as proposed by Heilmeier et al. (2020). The race track is represented by a sequence of tuples (x_i, y_i, w_i), $i \in \{0, ..., N-1\}$, where (x_i, y_i) denotes the coordinate of the center location and w_i denotes the lane width at the i-th point. The output racing line consists of a tuple of seven variables: coordinates x and y, longitudinal displacement s, longitudinal velocity v_x, acceleration a_x, heading angle ψ, and curvature κ. It is obtained by minimizing the following cost:

$$\begin{aligned} &\min_{\eta_1 \dots \eta_N} \quad \sum_{n=0}^{N-1} \kappa_i^2(n) \\ &\text{s.t.} \quad \eta_i \in \left[-\frac{w_i}{2} + \frac{w_{veh}}{2}, \frac{w_i}{2} - \frac{w_{veh}}{2} \right] \end{aligned} \tag{4}$$

where the vehicle width is w_{veh}, and η_i is the lateral displacement with respect to the reference center line.

To create a velocity profile, we need to consider the vehicle's constraints on both longitudinal and lateral acceleration Heilmeier et al. (2020). Our approach involves generating a library of velocity profiles, each tailored to specific lateral acceleration limits determined by the friction coefficients for the front (μ_f) and rear (μ_r) tires, as well as the vehicle's mass (m) and the gravitational constant (g). In particular, we produce a set of velocity profiles covering a range of maximum lateral forces corresponding to the friction μ_{eff} within the interval $[\mu_{min}, \mu_{max}]$. This library allows us to retrieve a velocity profile that matches a given value of μ. Interpolation is necessary when we encounter a friction value that falls within the valid range but is not explicitly present in the library.

An example of a racing line calculated for the racetrack used in our numerical study in Section 4 is shown in Figure 4.

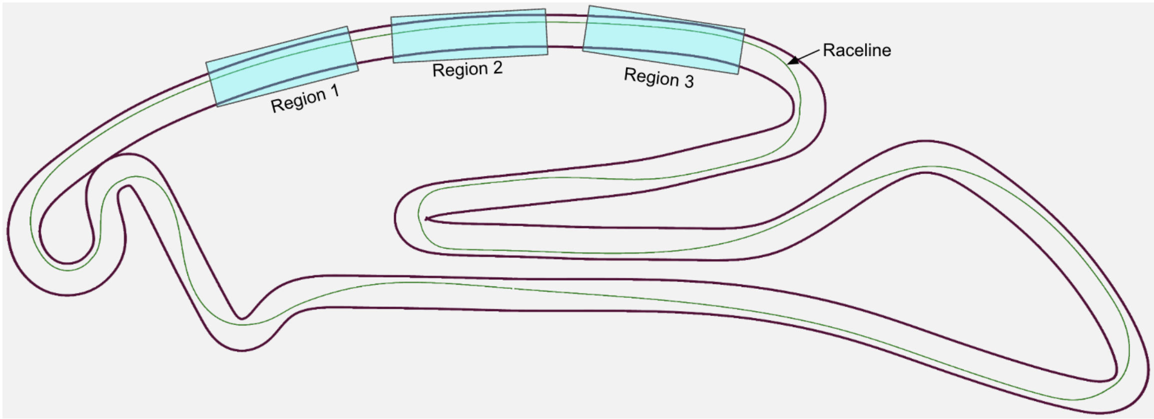

Figure 4: Track and the starting position regions

Appendix C. Dynamic Bicycle Model

For any car i, we denote its mass by m^i, its moment of inertia in the vertical direction about the center of mass by I_z^i, the distance between the center of mass (COM) and its front wheel by l_f^i, and the distance from the COM to the rear wheel l_r^i. Also, κ_t^i denotes the inverse of radius of curvature of the track at $p_{x,t}^i$. Using these notations, the dynamics of car i is defined below:

$$\begin{bmatrix} p^i_{x,t+1} \\ p^i_{y,t+1} \\ \phi^i_{t+1} \\ \tilde{v}^i_{x,t+1} \\ \tilde{v}^i_{y,t+1} \\ \omega^i_{t+1} \end{bmatrix} = \begin{bmatrix} p^i_{x,t} \\ p^i_{y,t} \\ \phi^i_t \\ v^i_{x,t} \\ v^i_{y,t} \\ \omega^i_t \end{bmatrix} + \Delta t \begin{bmatrix} v^i_{x,t} \\ v^i_{y,t} \\ \omega^i_t - \frac{\kappa^i_t}{(1-\kappa^i_t p^i_{y,t})}(\tilde{v}^i_{x,t}\cos(\phi^i_t) - \tilde{v}^i_{y,t}\sin(\phi^i_t)) \\ \frac{1}{m^i}(F^i_{r,x,t} - F^i_{f,y,t}\sin(\delta^i_t) + m^i \tilde{v}^i_{y,t}\omega^i_t) \\ \frac{1}{m^i}(F^i_{r,y,t} + F^i_{f,y,t}\cos(\delta^i_t) - m^i \tilde{v}^i_{x,t}\omega^i_t) \\ \frac{1}{I^i_z}(F^i_{f,y,t} l^i_f \cos(\delta^i_t) - F^i_{r,y,t} l^i_r) \end{bmatrix}, \tag{5}$$

where (i) $v^i_{x,t} = \frac{1}{(1-\kappa^i_t p^i_{y,t})}(\tilde{v}^i_{x,t}\cos(\phi^i_t) - \tilde{v}^i_{y,t}\sin(\phi^i_t))$, $v^i_{y,t} = \tilde{v}^i_{x,t}\sin(\phi^i_t) + \tilde{v}^i_{y,t}\cos(\phi^i_t)$ are the velocities in frenet frame; (ii) $\tilde{v}^i_{x,t}, \tilde{v}^i_{y,t}$ are velocities in body frame; (iii) $F^i_{r,x,t} = (C_1 - C_2\tilde{v}^i_{x,t})d^i_t - C_3 - C_4(\tilde{v}^i_{x,t})^2$ is the longitudinal force on the rear tire at time t. Here, C_1 and C_2 are parameters that govern the longitudinal force generated on the car in response to the throttle command, while C_3 and C_4 are parameters that account for the friction and drag forces acting on the car; (iv) $F^i_{f,y,t} = D_f \sin(C_f \tan^{-1}(B_f \alpha^i_{f,t}))$ is the lateral force on the front tire depending on the slipping angle $\alpha^i_{f,t}$, which is given by $\alpha^i_{f,t} = \delta^i_t - \tan^{-1}\left(\frac{\omega^i_t l_f + \tilde{v}^i_{y,t}}{\tilde{v}^i_{x,t}}\right)$. Here B_f, C_f, D_f are the parameters of Pacejka tire model; and (v) $F^i_{r,y,t} = D_r \sin(C_r \tan^{-1}(B_r \alpha^i_{r,t}))$ is the lateral force on the rear tire depending on the slipping angle $\alpha^i_{r,t}$, which is given by $\alpha^i_{r,t} = \tan^{-1}\left(\frac{\omega^i_t l_r - \tilde{v}^i_{y,t}}{\tilde{v}^i_{x,t}}\right)$. Here B_r, C_r, D_r are the parameters of Pacejka tire model.

Appendix D. Hyperparameters

D.1. Network architecture

We use a simple feed-forward deep neural network with ReLU activation except for the last layer to represent the value function and the potential function. The network for value function consists of 3 hidden layers with $(128, 128, 64)$ hidden features on each layer. The network for potential function consists of 3 hidden layers with $(384, 384, 192)$ hidden features on each layer.

D.2. Training

We use a learning rate of 0.0001 and train for 50000 epochs (both value functions and potential function). Each race consists of 500 time steps with $dt = 0.1s$, hence $50s$ race.

Appendix E. Self-play RL training

We use standard PPO training parameters as available in stable_baselines3 with batch size 1024, number of epochs 5, learning rate 0.0005, $\gamma = 0.99$ and 8 environments in parallel. The observation used is the same as the joint state input used for our work for fair comparison. The reward design used is also the same as the utility used in our work. We train for $100K$ time-steps for each iteration of self-play RL where we switch agents for training for total of 99 times i.e. 33 cycles of training for 3 agents

Appendix F. Iterated Best Response (IBR) hyperparameters

We use $N = 6$ iterations for iterated best response with the same utility as the one used in our work and with horizon length of $2s$ with 20 time-steps of length $dt = 0.1s$. The solve time with the following parameters is $0.1s$ which is comparable to the compute time required by our algorithm

Appendix G. Table of Notations

Notation	Description
$\mathbf{x}^i$	State vector of car i, including position, velocity, and angular velocity.
p_x^i, p_y^i	Longitudinal and lateral positions of car i in the global frame.
ϕ^i	Orientation of car i in the global frame.
v_x^i, v_y^i	Longitudinal and lateral velocities of car i in the body frame.
ω^i	Angular velocity of car i in the global frame.
$\mathbf{u}^i = (d^i, \delta^i)$	Control input for car i, where d^i is the throttle and δ^i is the steering angle.
d^i	Throttle input of car i.
δ^i	Steering angle of car i.
$d_{\min}, d_{\max}$	Minimum and maximum throttle limits for car i.
$\delta_{\min}, \delta_{\max}$	Minimum and maximum steering angle limits for car i.
m^i	Mass of car i.
I_z^i	Moment of inertia of car i in the vertical direction about the center of mass.
l_f^i	Distance from the center of mass to the front wheel of car i.
l_r^i	Distance from the center of mass to the rear wheel of car i.

Notation	Description
$F^i_{r,x,t}$	Force applied to the rear wheel of car i in the longitudinal direction at time t.
$F^i_{r,y,t}$	Force applied to the rear wheel of car i in the lateral direction at time t.
$F^i_{f,y,t}$	Force applied to the front wheel of car i in the lateral direction at time t.
Δt	Time step for the simulation.
γ	Discount factor in the utility function.
$\mathbf{x}_t$	State of the system at time t.
$\mathbf{u}^i_t$	Control input of car i at time t.
ω^i_t	Angular velocity of car i at time t.
θ^i	Parameter of car i's policy.
Θ^i	Set of possible parameters for car i's policy.
Π^i	Set of possible strategies for car i.
Π	Set of joint strategies for all players (cars).
ϵ	The tolerance used in the definition of ϵ-Nash equilibrium.
$V^i(\mathbf{x}, \theta^i, \theta^{-i})$	The expected long-run utility of car i given the state $\mathbf{x}$ and strategy profile θ.
θ^*	The ϵ-Nash equilibrium strategy profile.