A Implementation Speedups

As mentioned in Sec. 4, the pseudocode is presented so that the fundamentals of GTNS and isNashEquilibrium are as clear as possible. In practice, we obtain significant efficiency improvements by leveraging several implementation techniques.

The most consequential (inspired by works such as [4,6,27]) is *lazy validation* of collision and equilibrium checks: every neighbour of x visited in line 7 is pushed to the `OPEN` list without immediate validation; only when a neighbour is popped from `OPEN` (i.e., it currently has the lowest cost) do we verify that it is in fact a valid node. This defers the most expensive computations and spends resources only on promising candidates.

The implementation also caches edge-collision results (Boolean flags indicating whether an edge $e \in E$ is in collision) to allow fast lookup before invoking an expensive check, precomputes heuristics, and performs early feasibility checks for the query. Additional optimizations include careful memory management and the use of pointers to stable storage. Finally, we note that isNashEquilibrium is trivially parallelizable, as the per-robot A* runs with respect to fixed trajectories of all other robots are independent.

B Additional Experiments & Figures

This appendix includes supplementary figures and experiments that exceed the space limitations of the main manuscript. Specifically, we examine the effects of NE mismatch and its resulting adverse outcomes. Additionally, we present an experiment that isolates the influence of λ_{prox} in a semi-collaborative setting. Finally, a summary of the notation used in the paper is provided.

B.1 Effect of NE mismatch

Our approach allows for computing various NE solutions to the same problem. The experimental results indicate that different equilibria can result in vastly distinct performance of the multi-robot system. So far, we have assumed that all the agents execute the same equilibrium, which may not be guaranteed in practice. To motivate further study in coordinating the equilibrium between agents, we evaluate the consequences of equilibrium mismatch.

In the majority of tested cases, when agents executed in an open-loop fashion trajectories belonging to different NEs, collisions between robots were encountered. This underscores the need for a coordinated, team-wide equilibrium in behavior-aware motion planning. To address the concern that open-loop per-agent NEs are "obviously brittle," we also evaluated an MPC variant: at every step each agent re-solved *its own* NE instance, where $J = J_i$, i.e., the global cost strictly optimizes the solution of the specific robot executing its trajectory from this NE. At every step, the robot resolved the problem towards the goal using

this objective while observing the updated locations of the other robots, executed its step, and repeated the process. Such an approach is hoped to be more resilient as it could provide more opportunities for the robots to avoid colliding with each other. Unfortunately, even under this closed-loop regime, the lack of shared equilibrium selection caused systematic issues and in almost all scenarios (e.g., those in Figs. 4 and 1) a collision. In rare "best-case" scenarios, when interaction geometry was simple, (e.g., on the track in Fig. 5 with large start distance) mismatched individual NEs yielded similar solutions to those returned by the global planner.

We believe that such a NE mismatch could explain the cause of motor accidents between human drivers where a driver "fails to judge other driver's path or speed", which contributes to at least %13 of all accidents in the US, according to a 2013 report [28], and %11 of fatal accidents in the UK, based on a 2022 report [7]. This suggests that algorithms such as GTNS could be used by transportation engineers to assess the impact, in terms of safety, of introducing a new road feature (e.g., an intersection or a traffic island).

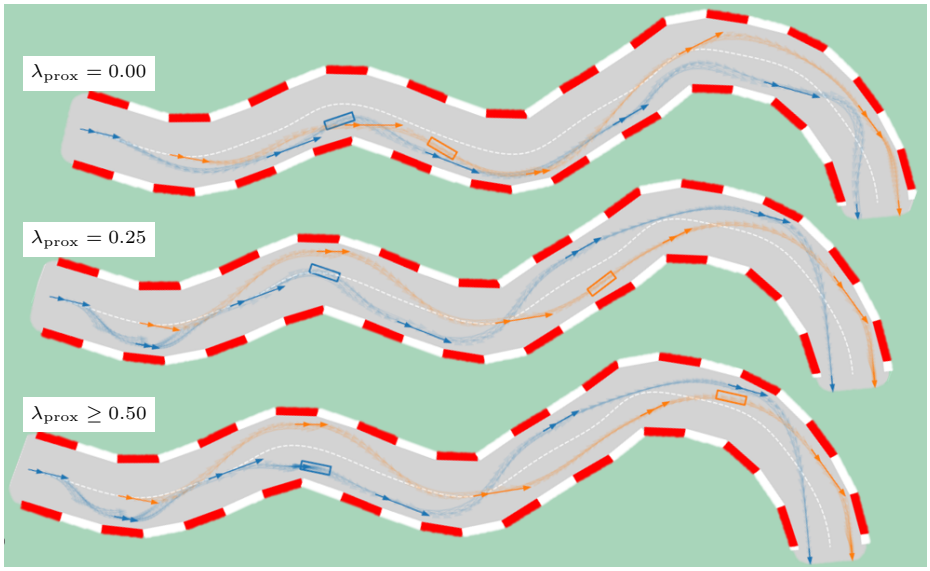
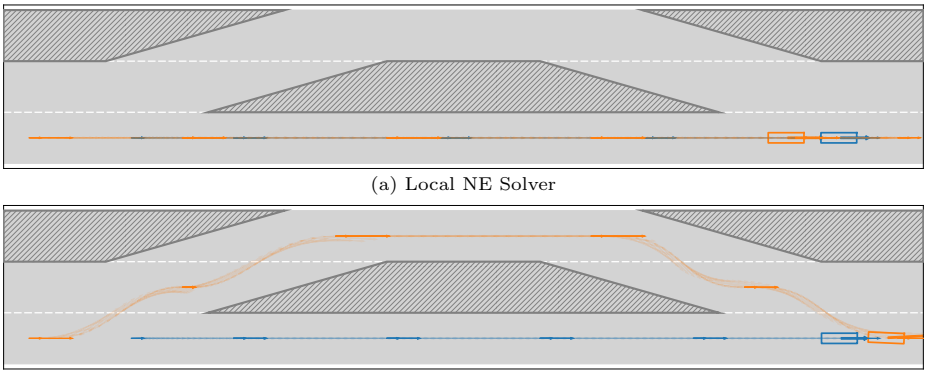


Fig. 5: Following distance. Semi-collaborative setting, such as two cars on the same team, with social welfare cost $J = \sum_i J^i$ and $\lambda_{\text{prox}} \in [0, 1]$. The cars are attempting to cross the finish line, with the lowest possible summed cost. The addition of a proximity penalty to the cost, affects the following distance that Robot 1 (blue) maintains from Robot 2 (orange): A small λ_{prox} value yields close drafting, and larger values increase the time-gap as Robot 1 deliberately backs off from Robot 2, visible at the first corner.



(a) Local NE Solver

(b) Global NE Solver

Fig. 6: Local vs. global Nash equilibrium (NE) in a narrow-overtake scenario, where Robot 1 (blue) is attempting to overtake Robot 2 (orange). (a) A local, initialization-dependent solver converges to a non-overtake NE within the homotopy class implied by its seed. (b) Our global NE method searches across homotopy classes and discovers here the overtaking NE.

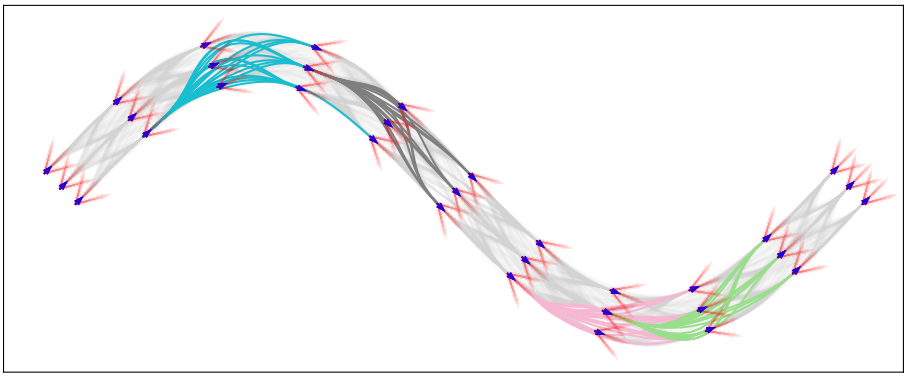


Fig. 7: Kinodynamic *track graph* built from racetrack centerline with lateral offsets. At each longitudinal sample we define a wayline (normal to the centerline), set the heading from the centerline tangent, and discretize (v, δ). From each wayline ℓ we attempt connections from all nodes on ℓ to all nodes on any wayline ℓ' with $|\ell' - \ell| \leq k$. Each discrete state $(x, y, \theta, v, \delta)$ is drawn with two arrows anchored at (x, y): a blue heading arrow pointing in direction θ (fixed short length), and a red steering arrow pointing in direction $\theta + \delta$ with length $\propto v$. Edge styling as in grid-graph.

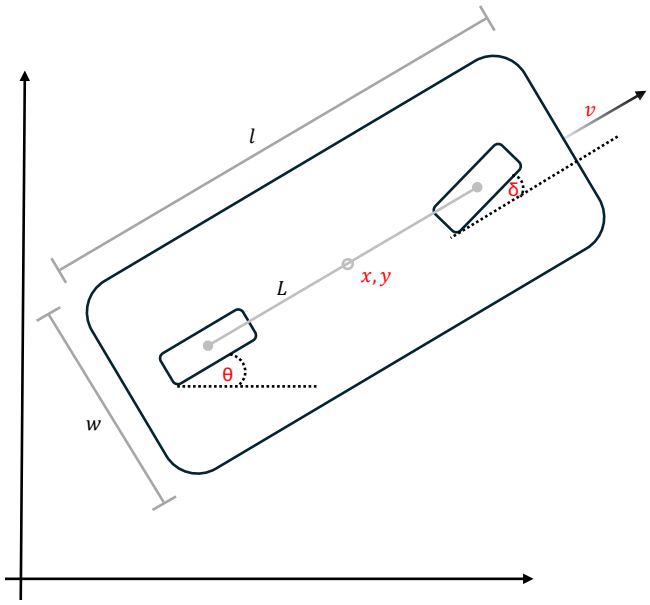


Fig. 8: Second-order bicycle model for car dynamics. Car dimensions l, w, L in black; State variables $(x, y, \theta, v, \delta)$ in red. In our experiments, we used a scale-model car of length $l = 0.70\,\text{m}$, width $w = 0.20\,\text{m}$, and wheelbase $L = 0.50\,\text{m}$.

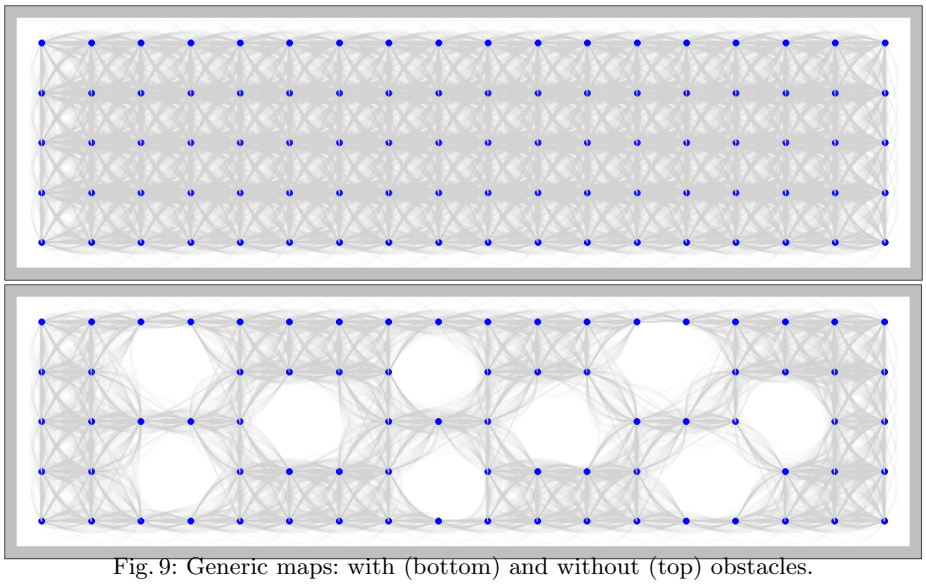


Fig. 9: Generic maps: with (bottom) and without (top) obstacles.

Table 2: Notation for continuous and graph-based multi-robot planning.

SR Symbol	MR Symbol	Meaning
	$[m]$	Robot index set $\{1, ..., m\}$
d_i, D_i		Dim. of $\mathcal{X}^i \subseteq \mathbb{R}^{d_i}$, $\mathcal{U}^i \subseteq \mathbb{R}^{D_i}$
$\mathcal{X}^i$	$\mathcal{X} = \times_{i=1}^m \mathcal{X}^i$	Robot i (joint) state space
$\mathcal{U}^i$	$\mathcal{U} = \times_{i=1}^m \mathcal{U}^i$	Robot i (joint) control space
$\mathcal{X}_f^i \subset \mathcal{X}^i$	$\mathcal{X}_f = \times_{i=1}^m \mathcal{X}_f^i \subset \mathcal{X}$	Collision–free state space
$\mathcal{U}_T^i$	$\mathcal{U}_T = \times_{i=1}^m \mathcal{U}_T^i$	Admissible controls $u^i : [0, T] \mapsto \mathcal{U}^i$ $\big(u : [0, T] \mapsto \mathcal{U}\big)$
$x^i(t)$	$x(t)=(x^1(t), ..., x^m(t))$	Robot i (joint) state at time $t \in [0, T]$
$u^i(t)$	$u(t)=(u^1(t), ..., u^m(t))$	Robot i (joint) control at time $t \in [0, T]$
x_0^i	x_0	Robot i (joint) initial state
$\mathcal{X}_{\text{goal}}^i$	$\mathcal{X}_{\text{goal}} = \times_{i=1}^m \mathcal{X}_{\text{goal}}^i$	Robot i (joint) goal region
$\pi^i_{x_0^i, u^i}$	$\pi_{x_0,u}=(\pi^1_{x_0^1,u^1}, ..., \pi^m_{x_0^m,u^m})$	Robot i (joint) continuous trajectory $\pi^i : [0, T] \mapsto \mathcal{X}^i$ $\big(\pi : [0, T] \mapsto \mathcal{X}\big)$
Π^i	$\Pi = \times_{i=1}^m \Pi^i$	All robot i (joint) feasible continuous trajectories
$c^i(\pi^i(t), \pi^{-i}(t))$		Stage cost of robot i given its and others' trajectories
	$c(\pi(t))$	Joint stage cost over joint trajectory
$J^i(\pi^i, \pi^{-i})$		Cost of robot i given its and others' trajectories
	$J(\pi)$	Global cost over joint trajectory
$f^i(x^i, u^i)$	$f(x, u)$	Robot i (joint) dynamics: $\dot{x}^i(t) = f^i(x^i(t), u^i(t))$ $\big(\dot{x}(t) = f(x(t), u(t))\big)$
Δt		Fixed duration of each edge
$G^i = (V^i, E^i)$	$G = (V, E)$	Robot i (joint, tensor-product) graph
V^i	$V = \times_{i=1}^m V^i$	Robot i (joint) sampled states $x^i \in \mathcal{X}^i$ incl. x_0^i, $\mathcal{X}_{\text{goal}}^i$ $\big(x \in \mathcal{X}$ incl. x_0, $\mathcal{X}_{\text{goal}}\big)$
E^i	$E = \times_{i=1}^m E^i$	Robot i (joint) edges $e^i = (x^i, y^i)$ $\big(e = (x, y)\big)$ with control $u_{e^i}^i \in \mathcal{U}_{\Delta t}^i$ $\big(u_e \in \mathcal{U}_{\Delta t}\big)$
$\pi^i_{e^i}$	$\pi_e=(\pi^1_{e^1}, ..., \pi^m_{e^m})$	Robot i (joint) trajectory on edge e^i $\big(e\big)$
$\pi_n^i=(x^i_{e_1^i}, ..., x^i_{e_n^i})$	$\pi_n=(x_{e_1}, ..., x_{e_n})$	Robot i (joint) n-step trajectory on G^i $\big(G\big)$
$\pi^i_{x^i}$	$\pi_x=(\pi^1_{x^1}, ..., \pi^m_{x^m})$	Robot i (joint) trajectory to state x^i $\big(x\big)$
$\pi^i_{x^i \to y^i}$	$\pi_{x \to y}$	Robot i (joint) local trajectory from state x^i to y^i $\big(x$ to $y\big)$
Π_n^i	$\Pi_n = \times_{i=1}^m \Pi_n^i$	Robot i (joint) feasible n-step trajectories
$P^i(x^i)$	$P(x)$	Predecessor map of node x^i $\big(x\big)$
	ε	Approximation factor for ε-equilibrium