Learning Realistic Traffic Agents in Closed-loop

Chris Zhang, James Tu, Lunjun Zhang, Kelvin Wong, Simon Suo, Raquel Urtasun

Traffic Simulation for Self-driving

Motivation: Developing self-driving in simulation is safer and more scalable than driving purely in the real world.

Goal: Learn models of how humans drive in order to use them as actor models in simulation.

Task: Given environmental information (e.g. high definition map, current actor positions and velocity), control how each actor should behave subsequently.

Challenges and Existing Work

Realistic actor models must:

1. Capture nuances of **human driving**
2. **Avoid infractions** like collisions or driving off-road

Existing approaches have shortcomings which can result in a **trade-off** between the two.

Imitation Learning:

- ✓ Leverages offline data for realism
- ✗ No explicit knowledge of infractions

Reinforcement Learning:

- ✓ Explicit reward signal
- ✗ Manual reward design lacks realism

Learning Objective

We model the problem with an MDP $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, P, \gamma)$

A trajectory $\tau_{0:T} = (\boldsymbol{s}_0, \boldsymbol{a}_0, \ldots, \boldsymbol{s}_{T-1}, \boldsymbol{a}_{T-1}, \boldsymbol{s}_T)$ is a state action sequence for all agents in the scene.

We aim to recover the expert distribution while satisfying an infraction-based constraint:

$$\arg\min_{\pi} \; D_{\mathrm{KL}}\left(P^{\pi}(\tau) \,\|\, P^{E}(\tau)\right) \quad \text{s.t.} \quad \mathbb{E}_{P^{\pi}}\left[R(\tau)\right] \geq 0$$

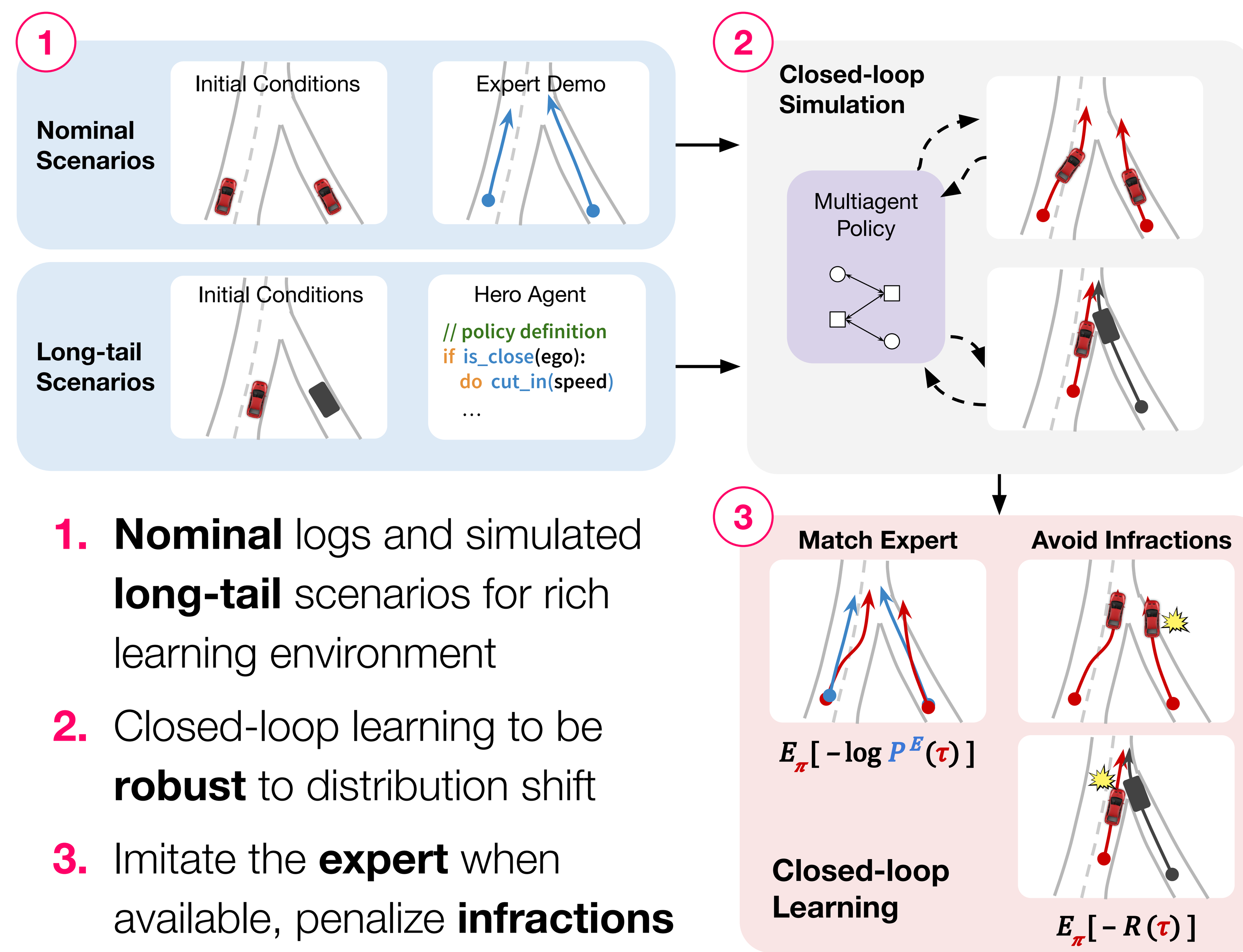
$$R^{(i)}(\boldsymbol{s}, a^{(i)}) = \begin{cases} -1 & \text{if infraction} \\ 0 & \text{otherwise,} \end{cases}$$

Taking the Lagrangian decomposes the objective into a combination of imitation and reinforcement learning:

$$\mathcal{L} = \mathbb{E}_{P^{\pi}}\Big[\underbrace{-\log P^{E}(\tau)}_{\text{IL}} - \lambda \underbrace{R(\tau)}_{\text{RL}}\Big] - H(\pi)$$

Reinforcing Traffic Rules (RTR)

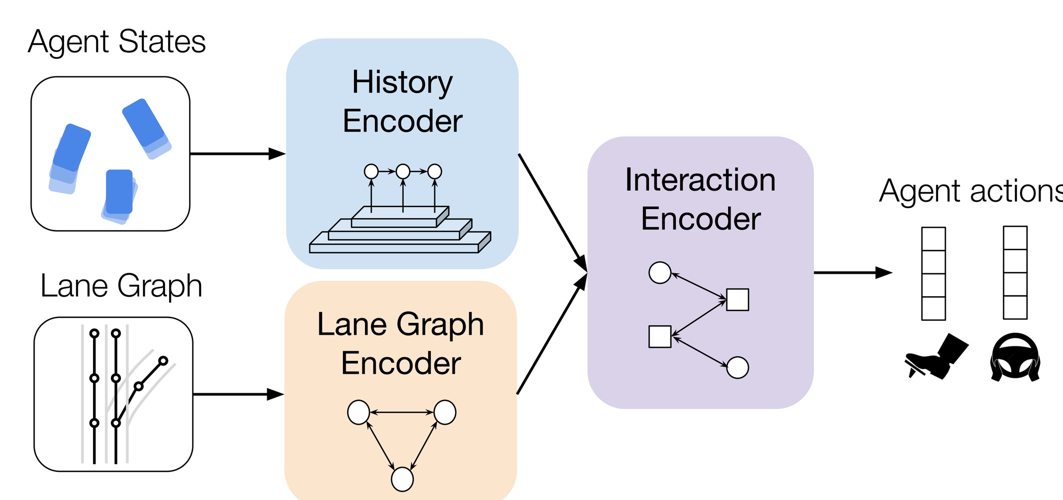
We **combine RL and IL** to learn robust policies in closed-loop.



1. **Nominal** logs and simulated **long-tail** scenarios for rich learning environment
2. Closed-loop learning to be **robust** to distribution shift
3. Imitate the **expert** when available, penalize **infractions**

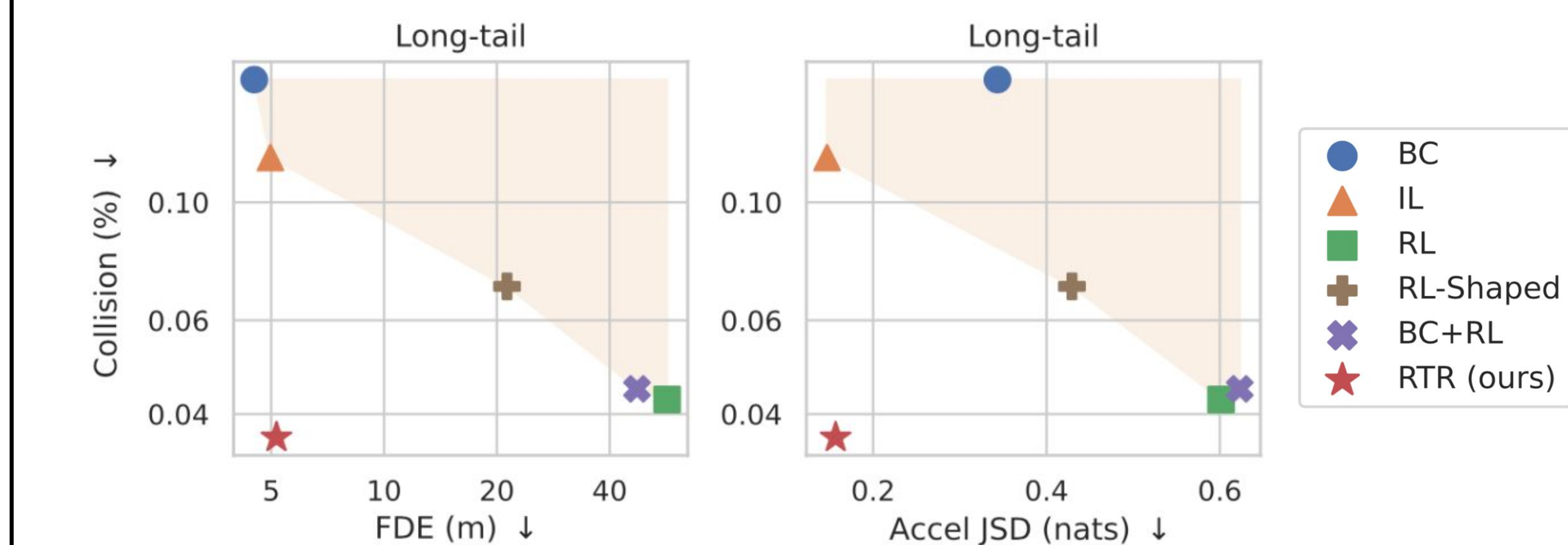
Architecture:

We use an efficient **multi-agent** architecture to extract features and jointly predict all agent actions.



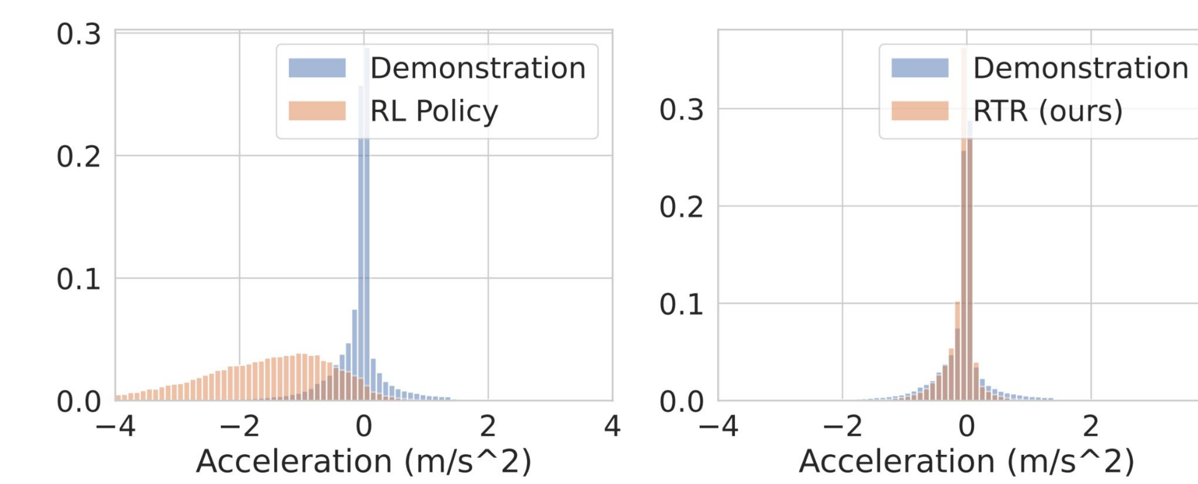
Value network design is the same as policy network but regresses value targets instead.

Realism and Infraction Avoidance



RTR achieves the **best tradeoff**, outperforming the Pareto frontier of baselines which vary between IL, RL and IL + RL

RTR learns to **avoid infractions** while still capturing **human**-like driving.



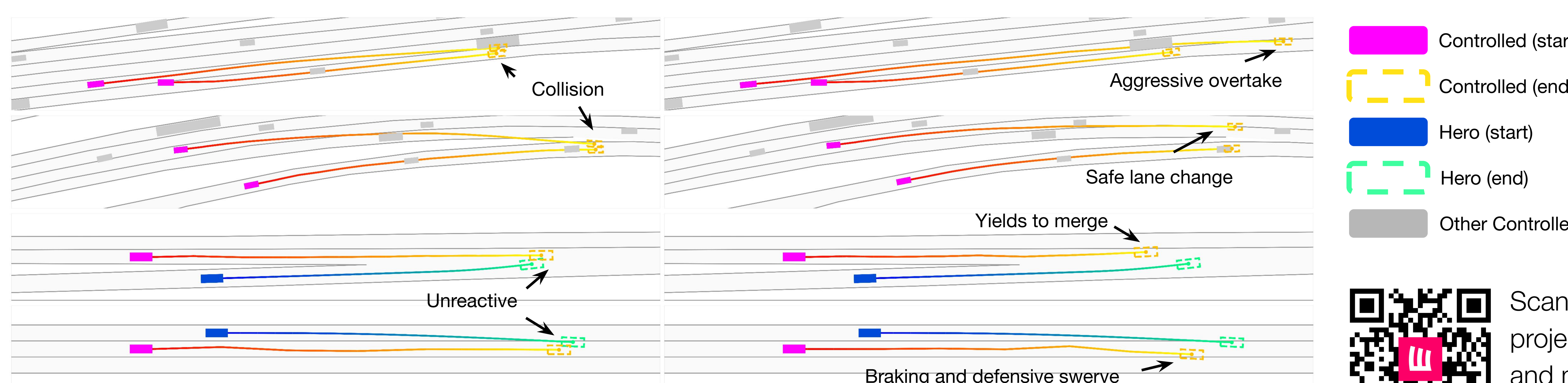
Downstream evaluation

We train a prediction model on **actor-simulated** data and evaluate them on **real data**.

RTR simulations have **lower domain gap** vs. baselines

Method	FDE (m)	CTE (m)
BC	2.44 ± 0.05	0.90 ± 0.04
IL	1.75 ± 0.06	0.28 ± 0.01
RL	15.42 ± 1.21	0.32 ± 0.02
RL-Shp	6.66 ± 0.26	0.33 ± 0.01
BC+RL	9.06 ± 0.50	0.42 ± 0.03
RTR	**1.58 ± 0.05**	**0.27 ± 0.03**

Qualitative Results



Baseline Imitation Learning **RTR (ours)**

Scan for project page and more results!