Robot Navigation in Interactive Environments with Structured Behavior Models

Shuijing Liu

Preliminary Exam Presentation

11/28/2022

HUMAN-CENTERED AUTONOMY LAB

ILLINOIS

Robots that serve humans

Isolation between robots and humans

Video credits:
https://www.amazon.science/latest-news/robin-deals-with-a-world-where-things-are-changing-all-around-it
https://www.youtube.com/watch?v=fn3KWM1kuAw
https://www.youtube.com/watch?v=KhDEEN4gcpI

Motivation

- Goal: Enable autonomous navigation in real human environments

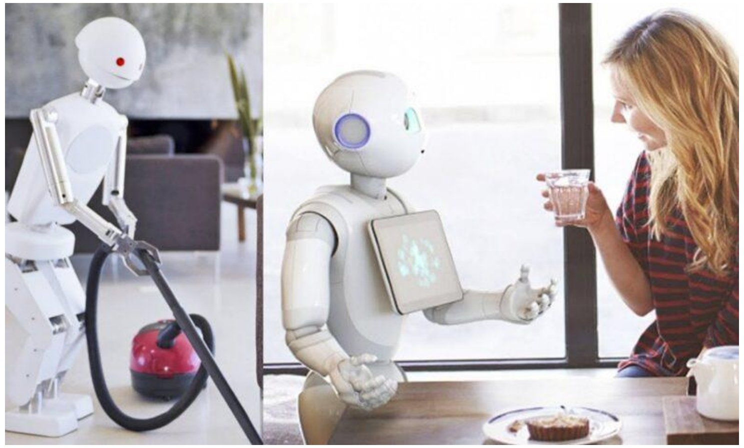


Photo credits: https://www.youtube.com/watch?v=MhqtVoVIuRs; https://www.urdesignmag.com/technology/2019/03/01/fedex-sameday-bot; https://incubees.com/get-ready-for-in-house-robots

Motivation

- Goal: Enable autonomous navigation in real human environments
- Challenge: It is difficult to infer the way that agents influence each other, making the interactive environments harder to navigate
- Steps to take:
  - Understand the interactive behaviors of agents in the environment
  - Learn safe but not overly conservative navigation strategies

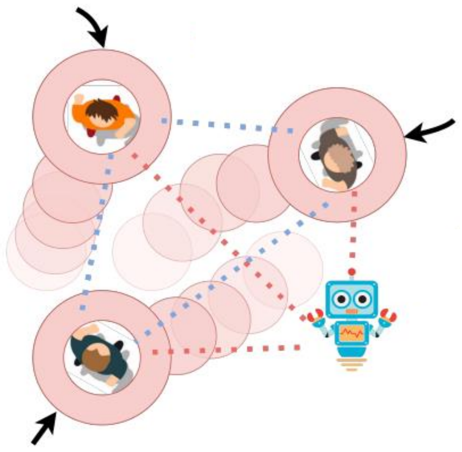
Photo credits: https://ideas.4brad.com/uber-robocar-hits-and-kills-pedestrian-Arizona, https://www.youtube.com/watch?v=nDmUyxjdCO8, https://www.realsimple.com/shopping/amazon-post-prime-day-irobot-roomba-692-robot-vacuum-deal-2022

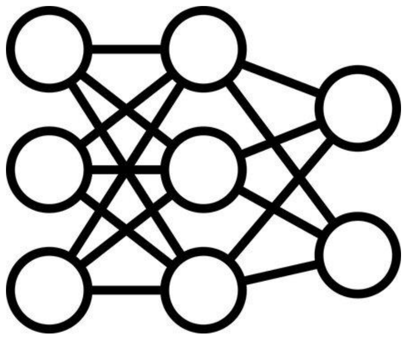
Overview of our approach

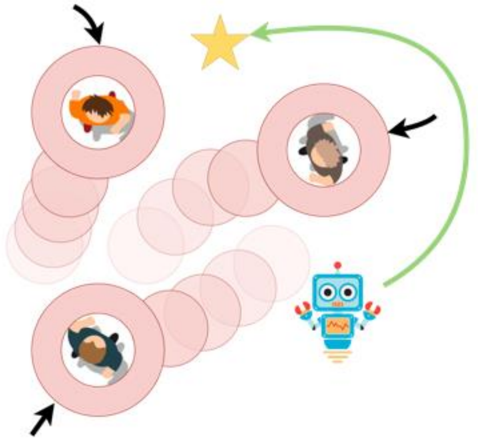
Uncover the structures of interactive scenes

Design a structured network to train the robot

Improve robot navigation in interactive scenes





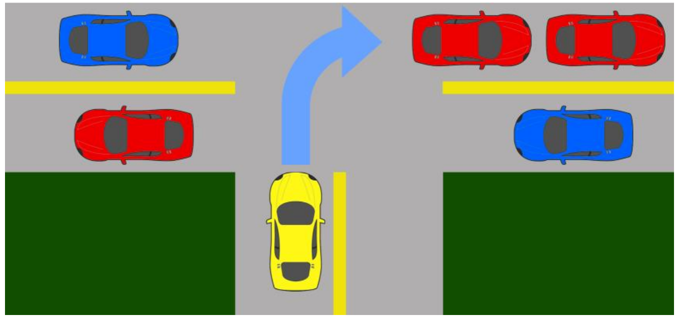


Key insights

By uncovering the structures beneath the interactive behaviors of agents, we can improve the robot navigation in interactive environments.

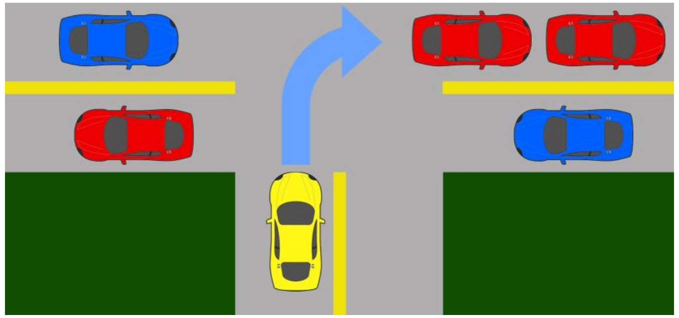
Contributions

- Driver internal state inference for navigation in a structured environment [ICRA `22]
  - Unsupervised driving style representation learning
  - RL framework for ego car navigation



- Intention-aware graph interaction model for unstructured crowd navigation [ICRA `21, ICRA `23 under review]
  - Spatio-temporal graphs to model crowd interactions
  - RL navigation framework combined with prediction
  - General crowd navigation for large-scale deployment [In progress]

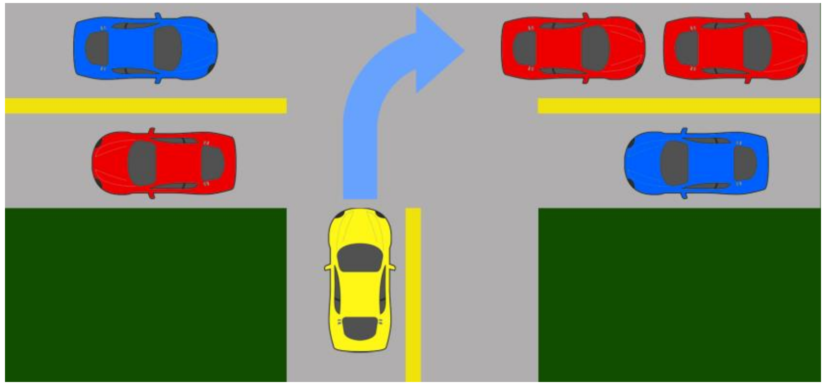
- Driver internal state inference for navigation in a structured environment [ICRA `22]



- Graphical model of interactive agents for crowd navigation [ICRA `21, ICRA `23 under review, In progress `22]

Introduction

- Task: The yellow ego car must merge into the upper lane of the T-intersection while other drivers with different driving styles are present
- Challenge for trait inference: Trait labels are hard to obtain ⇒ supervised learning is not ideal [Ma et al. 2021]
- Contributions:
  - Unsupervised driver trait representation learning with variational autoencoder
  - Navigation policy through an uncontrolled T-intersection with the learned trait representation



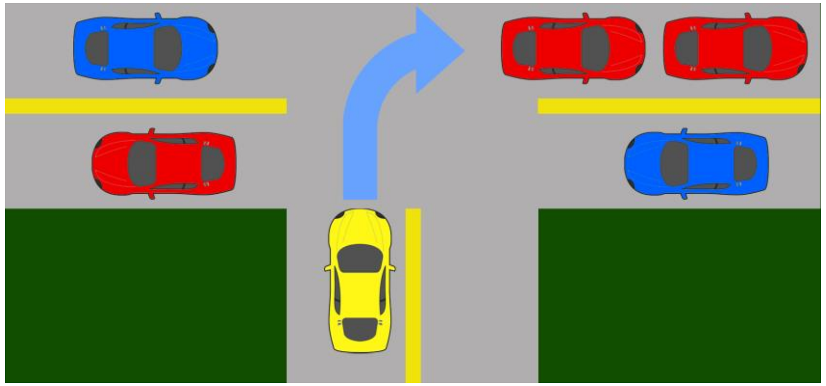
The simulated T-intersection environment in left-handed traffic.

Step 1: Trait representation learning

Simulation setting

The observed drivers are aggressive or conservative

- Aggressive drivers: higher desired velocity, smaller desired front gap, will not yield
- Conservative drivers: lower desired velocity, larger desired front gap, will yield



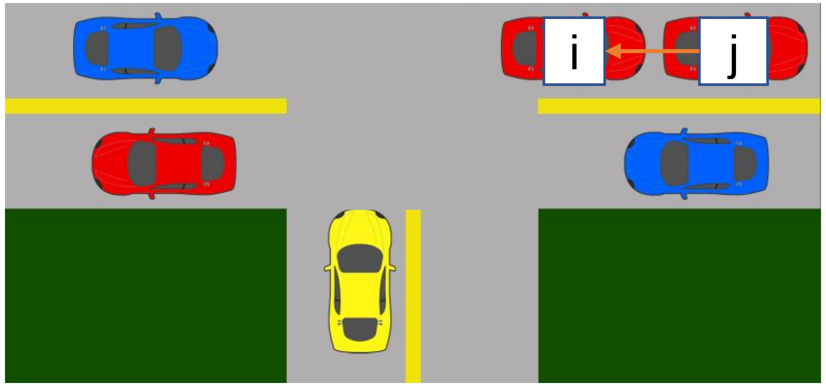
The simulated T-intersection environment in left-handed traffic.

Key observations

- The behavior of each car i is only affected by its front car j

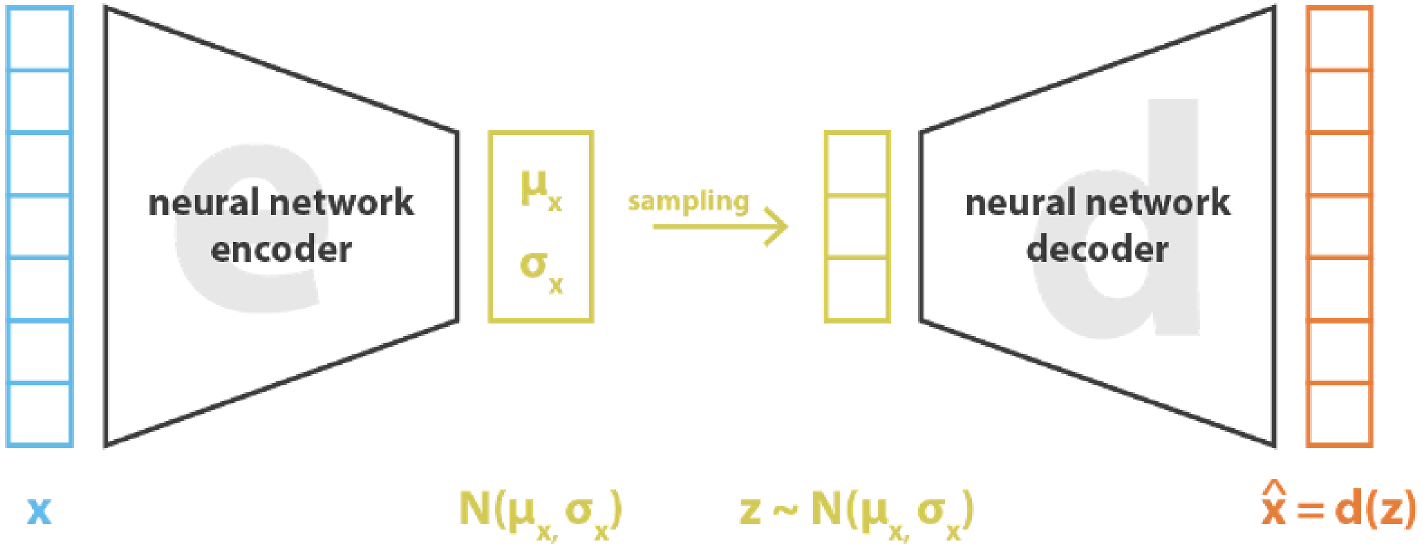
  ⇒ The state of car i at time t is $x_t = [x_t^i, x_t^j]$
- For a driver, the trait is a persistent and long-term property

  ⇒ Infer traits from a trajectory $\boldsymbol{x} = [x_1, \ldots x_L]$ instead of an instantaneous state



The simulated T-intersection environment in left-handed traffic.

Background: Variational Autoencoders (VAE)

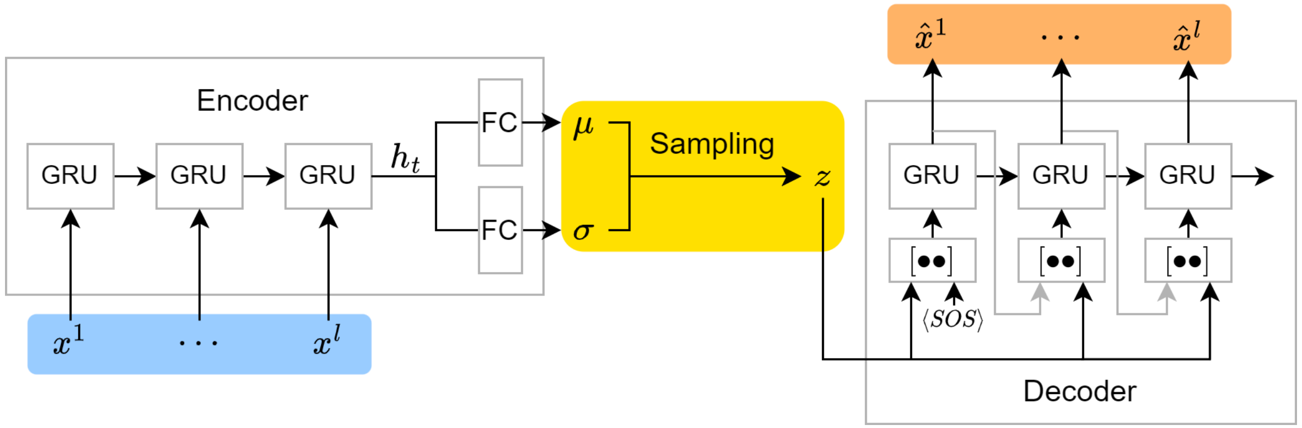


$$\text{loss} = \| x - \hat{x} \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,] = \| x - d(z) \|^2 + \text{KL}[\, N(\mu_x, \sigma_x), N(0, I) \,]$$

https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73

Trait representation learning

- Dataset: a set of trajectories of simulated trajectories $\{\boldsymbol{x}\}_{i=1}^{N}$
- Network: VAE with a gated recurrent unit (GRU) encoder and an GRU decoder

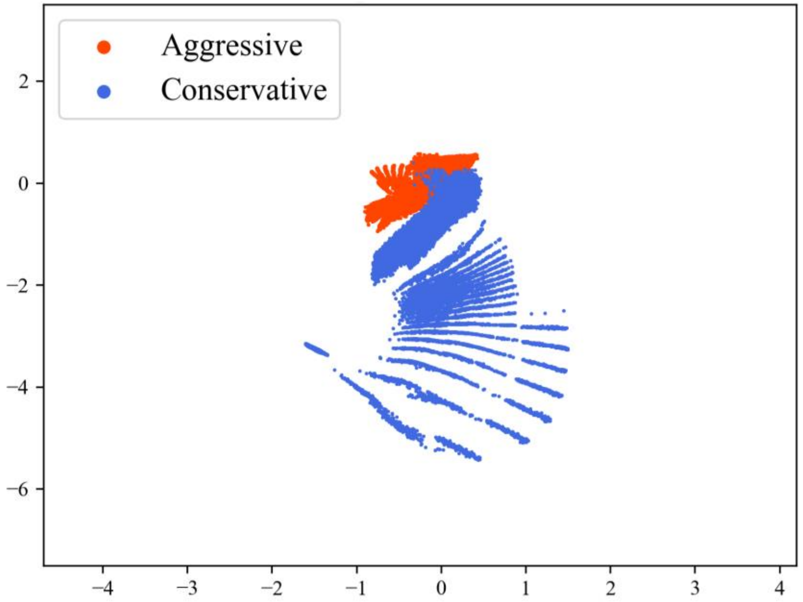


- Training: optimizing the evidence lower bound (ELBO)

$$\mathcal{L} = \underbrace{\beta D_{KL}(\mathcal{N}(\mu, \sigma) || \mathcal{N}(0, I))}_{\text{Regularization loss}} + \underbrace{||\boldsymbol{x} - \widehat{\boldsymbol{x}}||_2}_{\text{Reconstruction loss}}$$

Trait representation results

Qualitative: latent representations of unseen trajectories



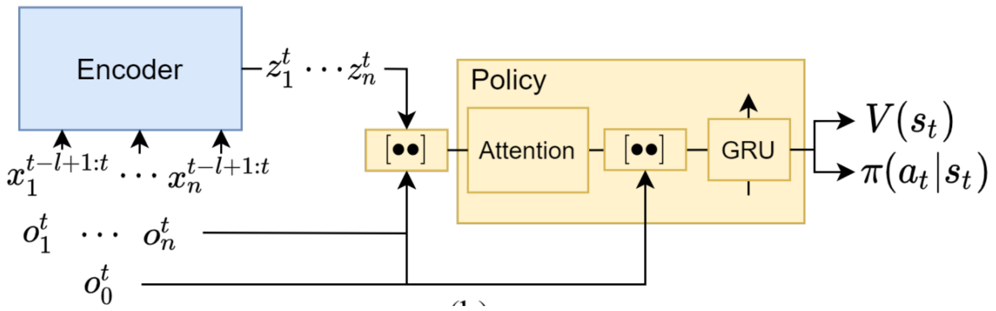
Quantitative: classification accuracy with a linear SVM

Method	Accuracy
Ours	**98.08%**
Morton *et al.*	60.22%

Step 2: Navigation policy learning

Trait-aware navigation

- POMDP formulation
  - Observable states: positions of the ego driver o_0 and other drivers $o_1, \dots, o_n$
  - Hidden states: traits of other drivers from the encoder: $z_1, \dots, z_n$
  - Actions: desired longitudinal velocity of the ego car
- Policy network: GRU with attention on each human driver
- Training: freeze the encoder, train the policy network with model-free reinforcement learning (RL)



The policy network

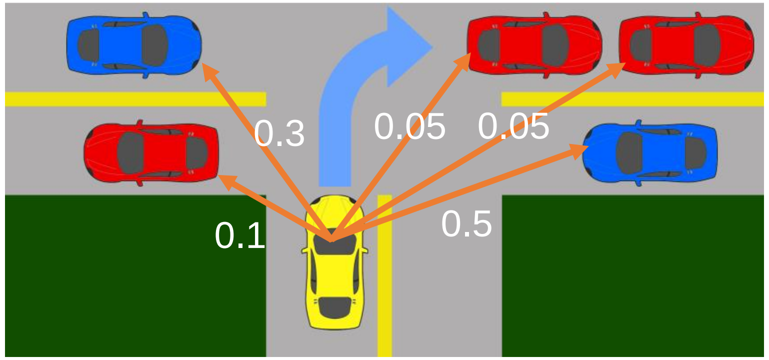
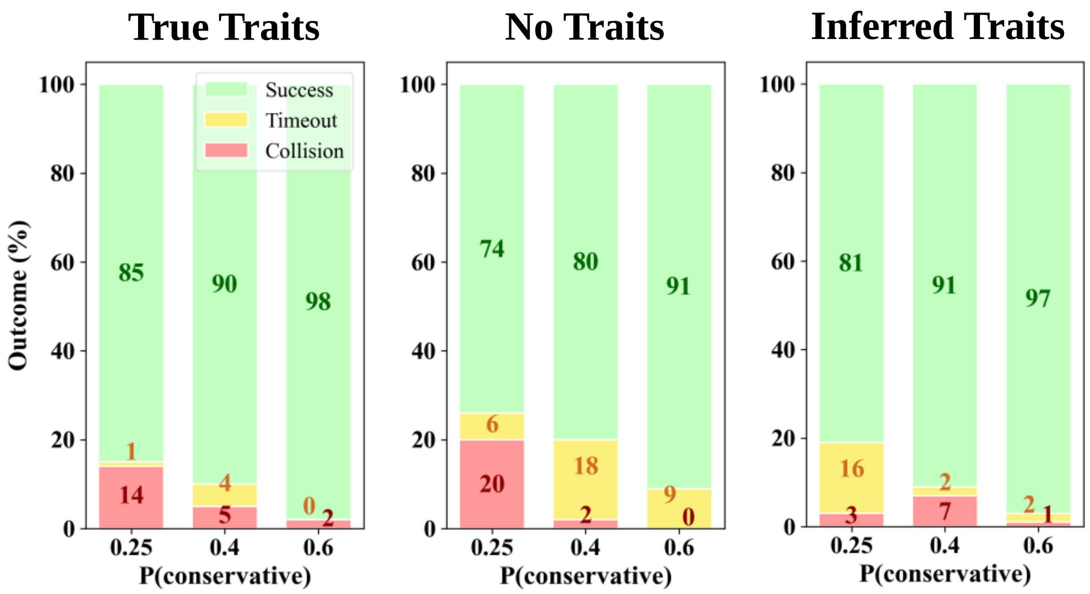
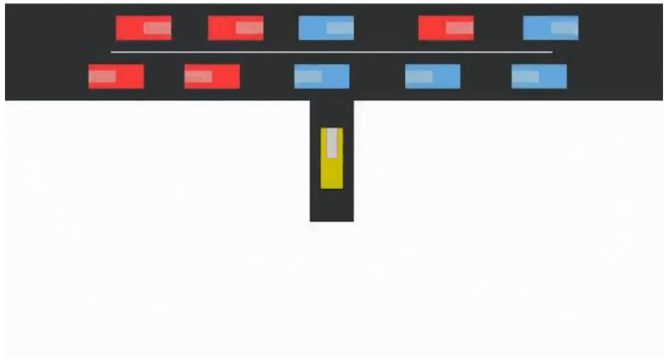


Illustration of attention scores

Navigation results

- The performance of our method (inferred traits) is
  - Close, if not equal, to the oracle policy with true trait labels
  - Much better than the baseline policy with no trait inference
- The ego car has learned to
  - Stop and wait for aggressive cars
  - Intercept in front of the first conservative car it observes



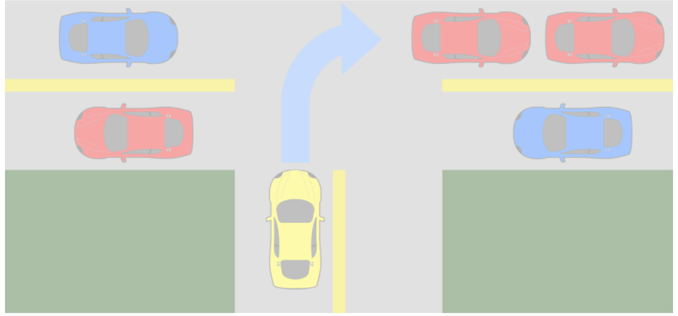


An example episode

Takeaways

- Without access to any labels, we propose an unsupervised approach to learn a representation of driver internal states from interactive trajectories
- With the inferred trait, we learn an adaptive navigation policy with RL, which can be potentially applied to realistic uncontrolled intersections
- Limitations and future work:
  - The environment is relatively structured and the interactions are simple
    $\Rightarrow$ navigation in unstructured interactive environments
  - Assumed that each agent has a distinctive trait
    $\Rightarrow$ structured model for general multi-agent interactions

- Driver internal state inference for navigation in a structured environment [ICRA `22]



- Graphical model of interactive agents for crowd navigation [ICRA `21, ICRA `23 under review, In progress `22]

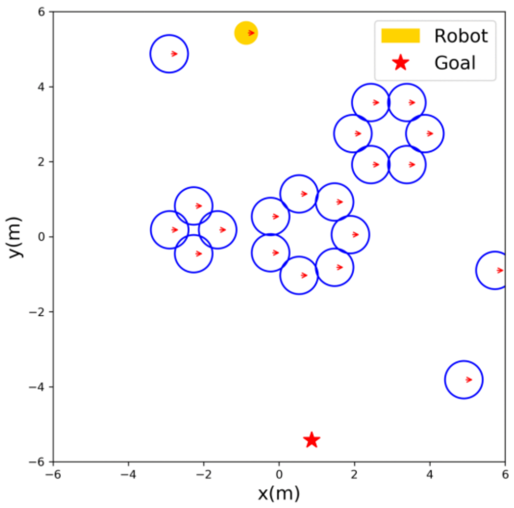
Introduction

- Goal: Enable robots to navigate in unstructured interactive environments.
- Task: The robot must navigate to a goal position without colliding with or intruding into the intended path of pedestrians.

Real-world crowd navigation

Limitations of previous works

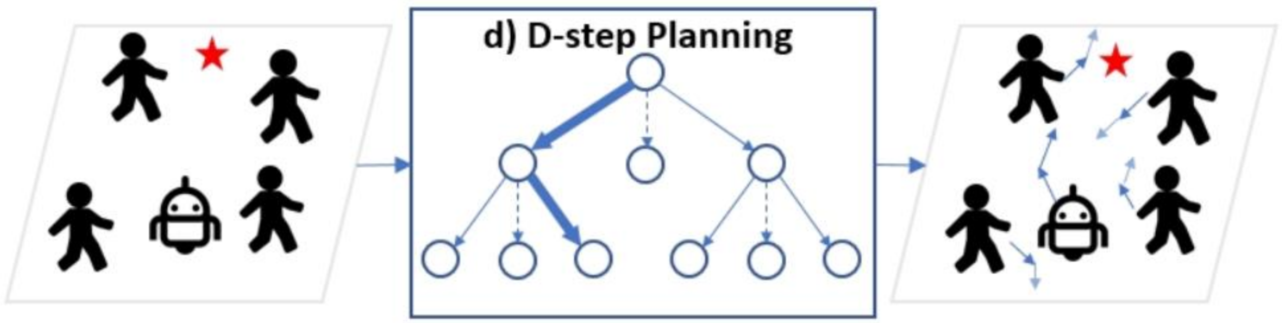
- Fail to consider people's intentions and different types of interactions, resulting in shortsighted robot behaviors [Van Den Berg et al. 2011, Chen et al. 2019]



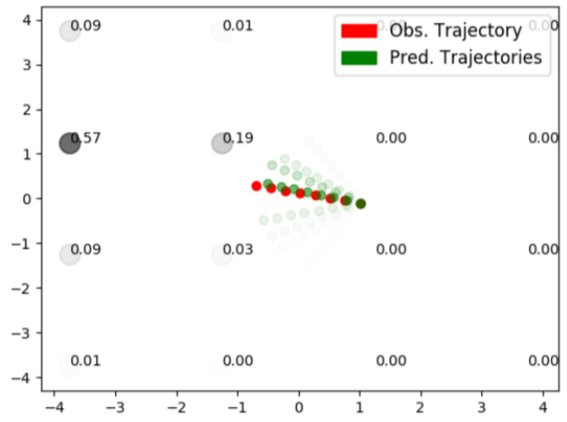
An example of freezing robot

Limitations of previous works

- Fail to consider people's intentions and different types of interactions, resulting in shortsighted robot behaviors [Van Den Berg et al. 2011, Chen et al. 2019]
- Prediction based methods does not scale well
  - Discrete robot action space [Chen et al. 2020]
  - A small set of human intentions [Katyal et al. 2020]



Discrete robot action space



A small, discrete set of human intentions [Katyal et al. 2020]

Contributions

- Model-free RL navigation pipeline that incorporates predicted trajectories of pedestrians
- Novel network architecture with attention mechanism to capture the spatial and temporal interactions in the unstructured crowds
- An open-source simulation benchmark, good results in simulation and real world

Real-world crowd navigation

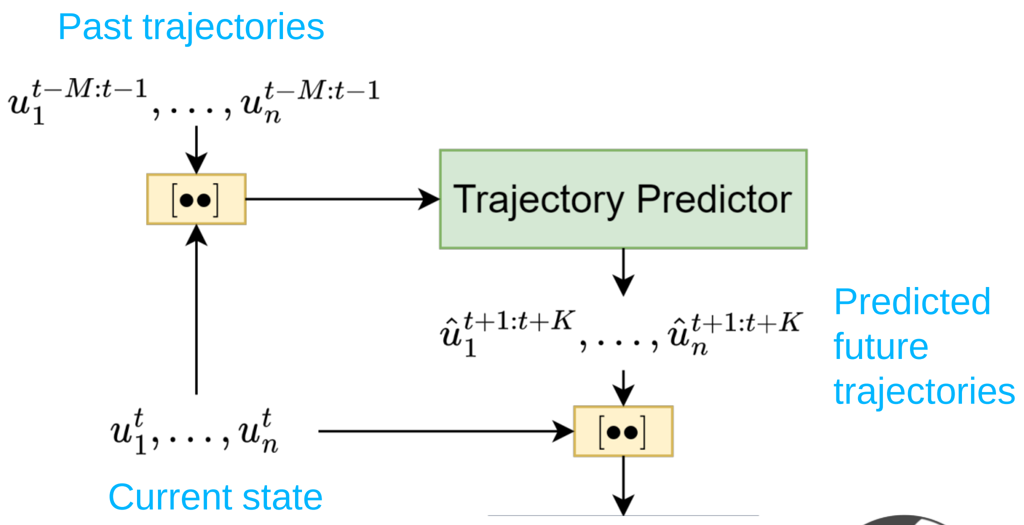
Intention aware RL framework

Intention-aware RL

- Given past trajectories, use any trajectory predictor to predict future trajectories of each pedestrian u_i

$$\hat{\mathbf{u}}_i^{t+1:t+K} = Predictor(\mathbf{u}_i^{t-M:t}), \quad i \in \{1, ..., n\}$$

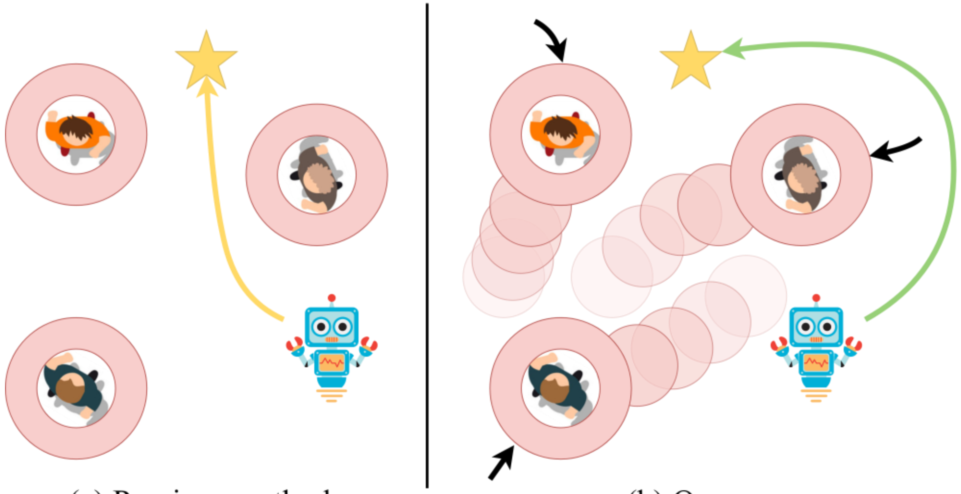
- In our MDP, a state includes
  - Robot state $\boldsymbol{w}^t$
  - Human current and future states $\boldsymbol{u}_1^t, \widehat{\boldsymbol{u}}_1^{t+1:t+K}, \dots, \boldsymbol{u}_n^t, \widehat{\boldsymbol{u}}_n^{t+1:t+K}$
- Action $a_t = [v_x, v_y]$ of the robot
- Assume the state transition probability $\mathcal{P}(\cdot \mid s_t, a_t)$ is unknown



Intention-aware RL

Reward function:

- Award(+): if the robot gets closer or arrives at the goal
- Penalty(-): if the robot moves away from the goal, or collides with the current or future human positions



(a) Previous methods (b) Ours

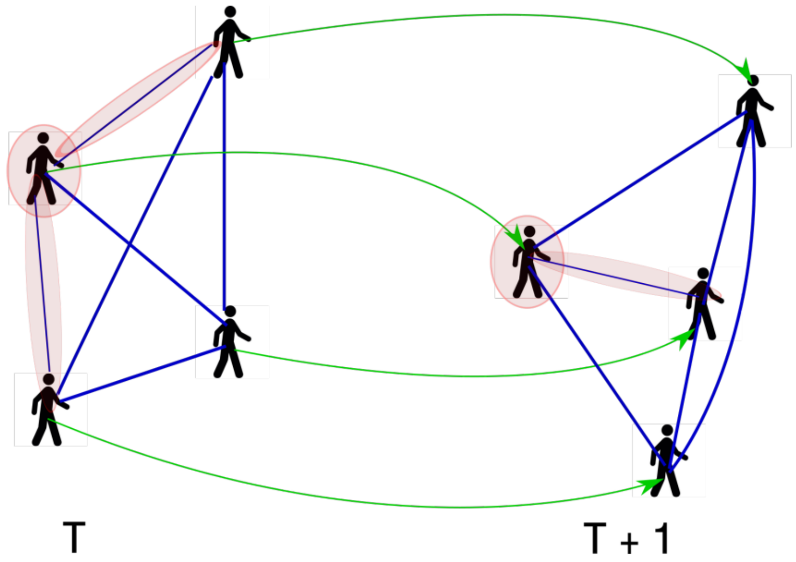
Graph network architecture

Motivation: Spatio-temporal (st) graph

- In our MDP, a state includes robot state and the states of all observed humans
- Question: Besides a simple concatenation, is there a better way to combine the agents’ states with a more principled approach?
- Our answer: st-graph

Background: Spatio-temporal (st) graph

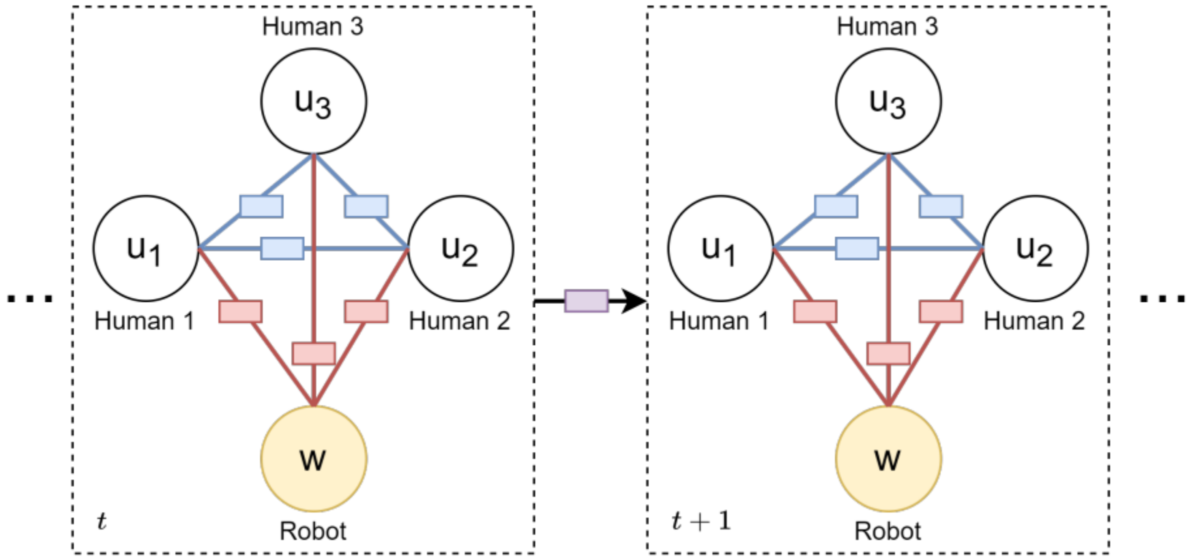
- Enables spatial and temporal reasoning for problems with inherent structures [Jain et al. 2016]
- Possible to add attention to model different importance of each edge [Vemula et al. 2018]



Spatio-temporal interaction graph

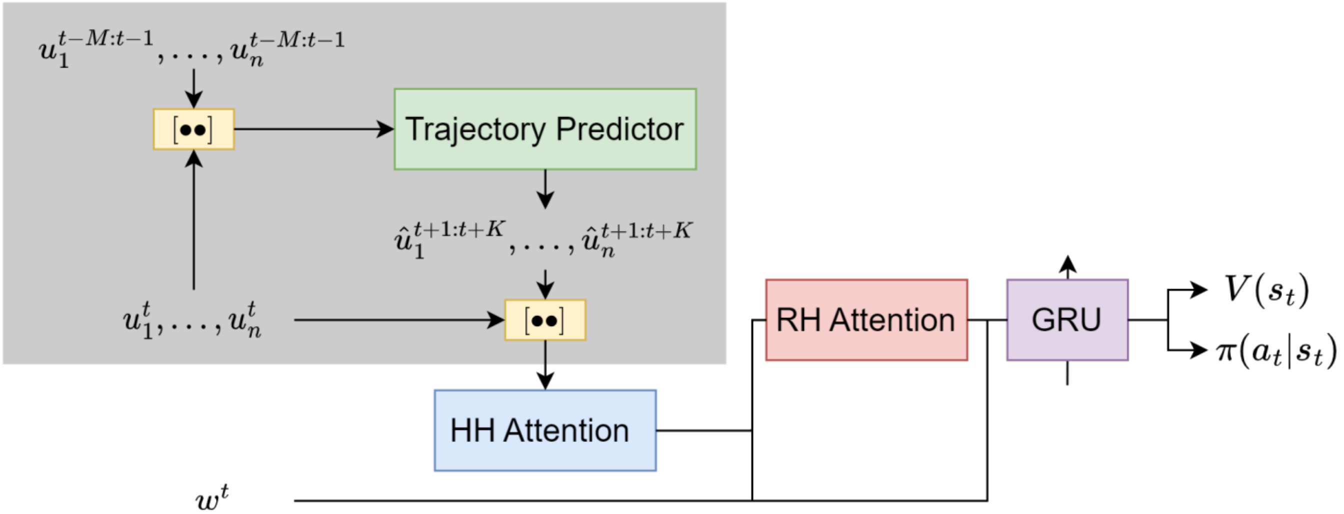
Formulate the crowd navigation scenario as an st-graph:

- robot-human (RH) spatial edges
- human-human (HH) spatial edges
- temporal edge



Robot policy network architecture

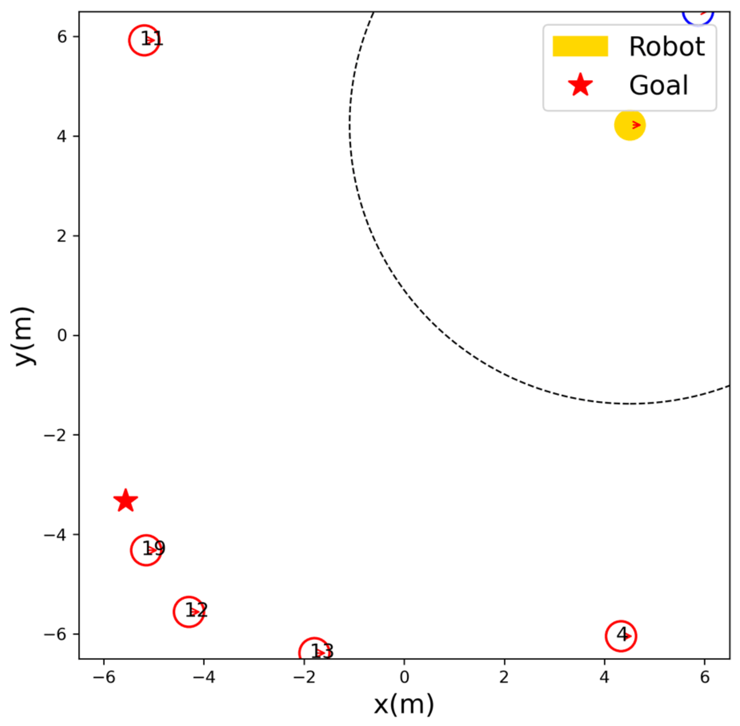
- Use separate attention networks to represent RH and HH interactions
  - HH attention: weights the features of each human w.r.t. other humans
  - RH attention: weights the features of each human again w.r.t. the robot
- Use GRU to represent the temporal function
- Train the policy network (non-shaded part) with RL



Simulator

OpenAI gym environment:

- Empty circles: A variable number of humans are controlled by reaction-based policies such as ORCA
- Randomized starting and goal positions for robot and humans
- Solid yellow circle: Robot with a limited field of view
    - Blue humans: detectable
    - Red humans: undetectable



Code link: https://github.com/Shuijing725/CrowdNav_DSRNN

Simulation experiments

- Baselines and variants
  - Previous crowd navigation works:
    - Reaction-based: ORCA [Van Den Berg et al. 2011], Social force (SF) [Helbing et al. 1995],
    - RL: DS-RNN [Liu and Chang et al. 2021]
  - Choice of trajectory predictor: constant velocity model (const vel), Gumbel Social Transformer (GST) [Huang et al. 2022]
  - Ablations: No prediction (no pred), no HH attention (no HH attn)
- Evaluation metrics
  - Navigation metrics: success rate, navigation time, path length
  - Social metrics: portion of intrusions, social distance during intrusions

Results: Effectiveness of interaction models

- Methods that models both RH and HH interactions > methods with only RH interactions

Navigation metrics Social metrics

Method	SR↑	NT↓	PL↓	ITR↓	SD↑
ORCA	78.0	15.87	18.53	26.04	0.36
SF	34.0	19.95	**17.75**	21.35	0.35
DS-RNN	67.0	20.06	25.42	13.31	0.37
Ours (No pred, HH attn)	82.0	19.15	22.82	14.87	0.37
Ours (GST, no HH attn)	82.0	**14.21**	19.35	7.22	0.40
Ours (Const vel, HH attn)	**94.0**	18.26	23.98	4.49	**0.43**
Ours (GST, HH attn)	**94.0**	17.64	22.51	**3.06**	**0.43**
Ours (Oracle, HH attn)	94.0	15.38	21.23	2.97	0.45

Results: Effectiveness of predictions

- Methods with predictions > methods without predictions

Navigation metrics (SR, NT, PL) Social metrics (ITR, SD)

Method	SR↑	NT↓	PL↓	ITR↓	SD↑
ORCA	78.0	15.87	18.53	26.04	0.36
SF	34.0	19.95	**17.75**	21.35	0.35
DS-RNN	67.0	20.06	25.42	13.31	0.37
Ours (No pred, HH attn)	82.0	19.15	22.82	14.87	0.37
Ours (GST, no HH attn)	82.0	**14.21**	19.35	7.22	0.40
Ours (Const vel, HH attn)	**94.0**	18.26	23.98	4.49	**0.43**
Ours (GST, HH attn)	**94.0**	17.64	22.51	**3.06**	**0.43**
Ours (Oracle, HH attn)	94.0	15.38	21.23	2.97	0.45

No prediction: ORCA, SF, DS-RNN, Ours (No pred, HH attn)

With prediction: Ours (GST, no HH attn), Ours (Const vel, HH attn), Ours (GST, HH attn), Ours (Oracle, HH attn)

Qualitative results

Simulation experiments

Non-randomized human scenario

- Red star: robot goal
- Yellow circle: robot
- Blue circles: detectable humans
- Red circles: undetectable humans
- Set of gray circles: true future human positions
- Set of orange circles: predicted human positions

Real world experiments

- Human detection with 2D LiDAR [Jia et al. 2020]
- Robot localization with a tracking camera
- All sensors are on-board
- 1-4 humans in a 5m x 5m indoor space
- **15 successes out of 18 trials (success rate ≈ 83.3%)**
- Failure cases: robot collides with walls

Takeaways

- Proposed a method to incorporate trajectory prediction of other agents into RL framework for interactive navigation

  ⇒ Intention-aware and proactive robot

- Proposed spatio-temporal graphs to model the heterogeneous interactions in unstructured crowd navigation scenario

  ⇒ A novel network architecture that learns a better policy

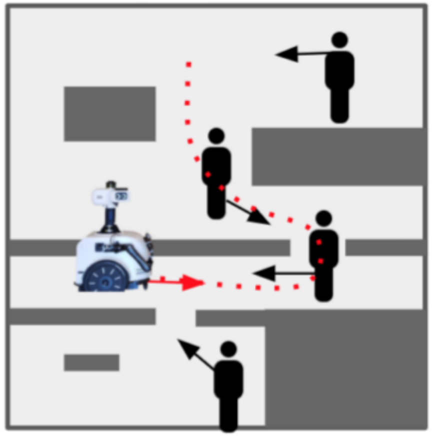
- Limitations:
  - Cannot handle the environmental constraints and static obstacles in a scalable way
  - Training RL from scratch is computationally expensive

Remaining work

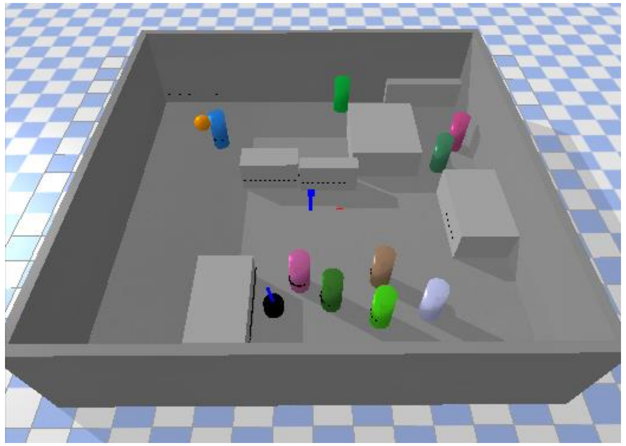
Combine RL method with existing approaches to obtain a crowd navigation policy that can be deployed in real environments

Navigation in constrained environments

- Previous path planners work well for mostly static environment, but have limitations in crowded environments [Fox et al. 1997]
- Our method: Include the static obstacles and constraints into the st-interaction graph
  - Base goal: Assume the environment is mapped, use the positions of objects as input
  - Stretch goal: Use raw sensor signals as inputs for unmapped navigation



Illustrations of concept
[Pérez D'Arpino et al. 2021]



PyBullet simulator of crowd navigation with constraints

Improve training efficiency of RL

- Learning an RL policy from scratch might not be affordable

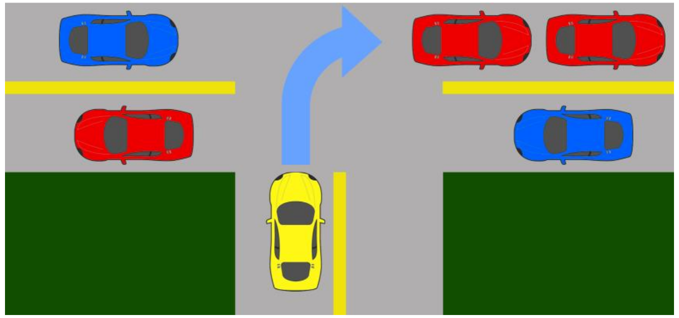
Training cost of different proposed networks

Task	Input	# of PPO training steps
T-intersection	Car positions	10×10^6
Pedestrian crowd	Human positions	20×10^6
	Raw LiDAR scans	30×10^6

- Take advantage of existing navigation methods to improve the data efficiency of RL
  - Base goal: Imitation learning as a warmup for RL [Chen et al. 2019]
  - Stretch goal: Residual policy learning [Johannink et al. 2019]

Closing remarks

- Interactive scenarios usually have underlaying structures
  - A suitable choice of interaction model leads to better robot navigation
- We developed methods and tools for navigation with
  - Prediction of individual agent characteristics
  - Graph model of collective agent behaviors
- Proposed next step:
  - Improve the training efficiency and generalization of our model for large-scale deployment



References

- S. Liu, P. Chang, H. Chen, N. Chakraborty, and K. Driggs-Campbell, "Learning to Navigate Intersections with Unsupervised Driver Trait Inference." In IEEE International Conference on Robotics and Automation (ICRA), 2022, pp. 3576-3582.
- S. Liu, P. Chang, W. Liang, N. Chakraborty, and K. Driggs-Campbell, “Decentralized structural-rnn for robot crowd navigation with deep reinforcement learning,” in IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 3517–3524.
- S. Liu, P. Chang, Z. Huang, N. Chakraborty, K. Hong, W. Liang, J. Geng, D. L. McPherson, and K. Driggs-Campbell, “Intention Aware Robot Crowd Navigation with Attention-Based Interaction Graph, ” arXiv preprint arXiv:2203.01821, 2022.
- X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, “Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships,” in IEEE International Conference on Robotics and Automation (ICRA), 2021.
- J. Morton and M. J. Kochenderfer, “Simultaneous policy learning and latent state inference for imitating driver behavior,” in IEEE International Conference on Intelligent Transportation Systems (ITSC), 2017, pp. 1–6.
- J. Van Den Berg, S. J. Guy, M. Lin, and D. Manocha, “Reciprocal n-body collision avoidance,” in Robotics research. Springer, 2011, pp. 3–19.
- C. Chen, Y. Liu, S. Kreiss, and A. Alahi, “Crowd-robot interaction: Crowd-aware robot navigation with attention-based deep reinforcement learning,” in International Conference on Robotics and Automation (ICRA), 2019, pp. 6015–6022.
- C. Chen, S. Hu, P. Nikdel, G. Mori, and M. Savva, “Relational graph learning for crowd navigation,” in IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020.

References

- K. D. Katyal, G. D. Hager, and C.-M. Huang, “Intent-aware pedestrian prediction for adaptive crowd navigation,” in IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 3277–3283.
- A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, “Structural-rnn: Deep learning on spatio-temporal graphs,” in IEEE conference on computer vision and pattern recognition (CVPR), 2016, pp. 5308–5317.
- A. Vemula, K. Muelling, and J. Oh, “Social attention: Modeling attention in human crowds,” in IEEE international Conference on Robotics and Automation (ICRA), 2018, pp. 1–7.
- D. Helbing and P. Molnar, “Social force model for pedestrian dynamics,” Physical review E, vol. 51, no. 5, p. 4282, 1995.
- Z. Huang, R. Li, K. Shin, and K. Driggs-Campbell, “Learning sparse interaction graphs of partially detected pedestrians for trajectory prediction,” IEEE Robotics and Automation Letters, vol. 7, no. 2, pp. 1198–1205, 2022.
- D. Fox, W. Burgard, and S. Thrun, “The dynamic window approach to collision avoidance,” IEEE Robotics and Automation Magazine, vol. 4, no. 1, pp. 23–33, 1997.
- C. Pérez-D’Arpino, C. Liu, P. Goebel, R. Martın-Martın, and S. Savarese, “Robot navigation in constrained pedestrian environments using reinforcement learning,” in IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 1140–1146.
- T. Johannink, S. Bahl, A. Nair, et al., “Residual reinforcement learning for robot control,” in IEEE International Conference on Robotics and Automation (ICRA), 2019, pp. 6023–6029.

Appendix A

Driver trait inference for AV navigation

Simulation settings

- Other drivers are controlled by IDM
  - Aggressive: front gap [0.3, 0.5]m, desired speed 2.4m/s
  - Conservative: front gap [0.5, 0.7]m, desired speed 3m/s
- Ego car has a fixed path, the desired speed of longitudinal PD controller is provided by RL network
- Reward function:

$$r(s,a) = \begin{cases} 2.5, & \text{if } s \in S_{goal} \\ -2, & \text{if } s \in S_{fail} \\ r_{speed}(s) + r_{step}, & \text{otherwise.} \end{cases}$$

where $r_{speed} = 0.05 \times \left|\left|v_{ego}\right|\right|_2$ is a small reward on the speed of the ego car, and $r_{step} = -0.0013$ is a constant penalty that encourages the ego car to finish as soon as possible

Trait representation learning – Data collection

- How: run the simulator without the ego car, record the trajectories of all IDM vehicles
- State definition:
  - At each timestep t, the observable state of each car is $x^t = (\Delta p_x, \Delta p_{x,f})$, where Δp_x is the horizontal offset from its starting position, and $\Delta p_{x,f}$ is the horizontal displacement of the car from its front car
  - Removed the features that are not useful for trait inference:
    - The lateral states
    - The direction information
- Size of dataset: 69600 trajectories, train/test split = 2:1

Appendix B

Robot crowd navigation with spatio-temporal graph

Reward function

Reward function:

$$r(s_t, a_t) = \begin{cases} 10, & \text{if } s_t \in S_{goal} \\ r_c, \text{ =-20} & \text{if } s_t \in S_{fail} \\ r_{pot}(s_t) + r_{pred}(s_t), & \text{otherwise.} \end{cases}$$

- Prediction reward r_{pred} discourages the robot from intruding into the predicted human positions: ($\mathbb{1}_i^{t+k}$ indicates whether the robot collides with human i at time $t+k$)

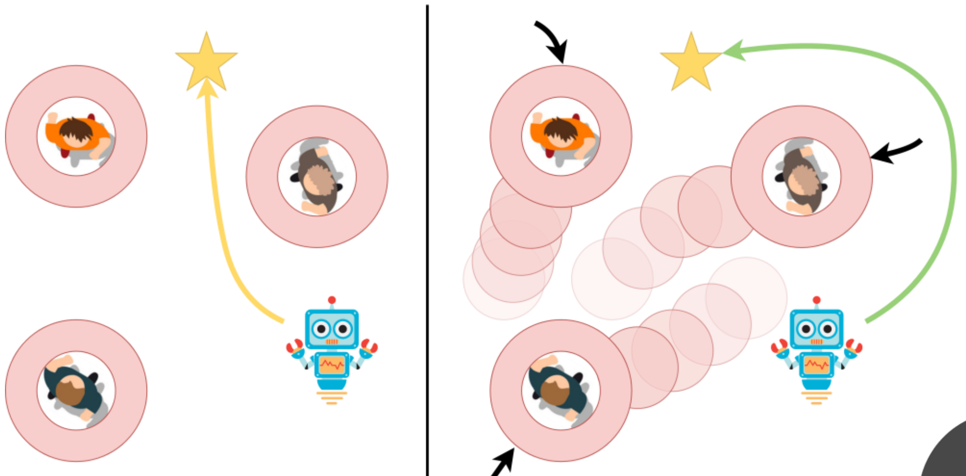
$$r^i_{pred}(s_t) = \min_{k=1,\ldots,K} \left(\mathbb{1}_i^{t+k} \frac{r_c}{2^k} \right)$$

$$r_{pred}(s_t) = \min_{i=1,\ldots,n} r^i_{pred}(s_t)$$

- Potential reward r_{pot} guides the robot to approach the goal

$$r_{pot} = 2(-d^t_{goal} + d^{t-1}_{goal})$$

where d^t_{goal} is the L2 distance between the robot and its goal



(a) Previous methods (b) Ours

RH and HH attention networks

- Scaled dot-product attention [Vaswani et al. 2017]:

$$\mathrm{Attn}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V$$

- HH attention:

$$Q_{HH}^t = [\mathbf{u}_1^{t:t+K}, ..., \mathbf{u}_n^{t:t+K}]^\top W_{HH}^Q$$
$$K_{HH}^t = [\mathbf{u}_1^{t:t+K}, ..., \mathbf{u}_n^{t:t+K}]^\top W_{HH}^K$$
$$V_{HH}^t = [\mathbf{u}_1^{t:t+K}, ..., \mathbf{u}_n^{t:t+K}]^\top W_{HH}^V$$

- RH attention: if v_{HH}^t is the output embeddings from HH attention

$$Q_{RH}^t = v_{HH}^t W_{RH}^Q,\ K_{RH}^t = \mathbf{w}^t W_{RH}^K,\ V_{RH}^t = v_{HH}^t W_{RH}^V$$