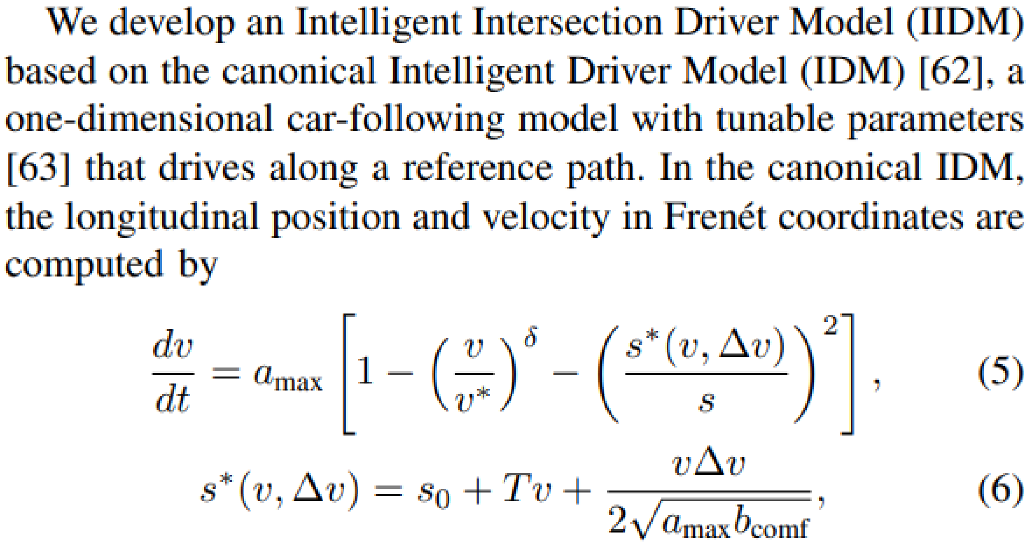
[TRO24] Interactive Autonomous Navigation with Internal State Inference and Interactivity Estimation

1. Link: https://arxiv.org/pdf/2311.16091
2. Authors: Honda (USA) + Stanford Intelligent Systems Laboratory (SISL)
3. [jintian]comments/critisism
    1. A sophisticatedly designed solution for addressing the 'driving through junction' problem
    2. The authors proposed the ISI module, which demonstrated a significant impact on these metrics. They claimed that the module performs robustly even when $\mathcal{P}(aggresive)$ changes during testing. Can I interpret this as an inaccuracy in the module's estimation?
    3. Is it feasible to implement this work in the real world? (I'm doubtful, since we cannot get the GT of intention)
    4. The difference between the setting of conservative and aggresive is not that significant.
4. hypothesis:
    1. modeling human internal states explicitly improves the decision making performance
    2. the inferred internal states can serve as explainable indicators
5. previous methods:
    1. use attention mechanism to "weight the importance of obs. to ego" may make mistakes
    2. DRL methods for sequencial decision-making is underexplored, especially in satefy-critical applications
    3. (1) brings similiar problem when it's used for decision-making
6. related works
    1. Interactive Decision Making: 2 ways for doing so
        1. agents are co-learning: use game theoritic formulation and multi-agent DRL (nash equilibria, stakelberge game....)
        2. other agents is a part of environment --> single agent contrl
    2. counterfactual learning: to answer those "what if..." problems
7. proposed methods:
    1. model:
        1. estimate the difference between the predicted trajectory distributions of the agent under the situations with and without the existence of the ego vehicle as a quantitative indicator (feature)
        2. use (1) as “interactivity score" to measure the impact of ego car to obs.
        3. (2) is a counterfactual reasoning problem, so they trained a model by using a dataset which removes ego car
        4. use spatial-temporal GNN to model relational reasoning
        5. add auxiliary features to indicate how the model infers
    2. simulator:
        1. four-way partially controlled intersectionenvironment that simulates challenging traffic scenarioswith interactive vehicles and crossing pedestrians
8. Details
    1. INTERSECTION DRIVING SIMULATION

1. Intelligent Intersection Driver Model (IIDM) a model for simulating the motion of obs

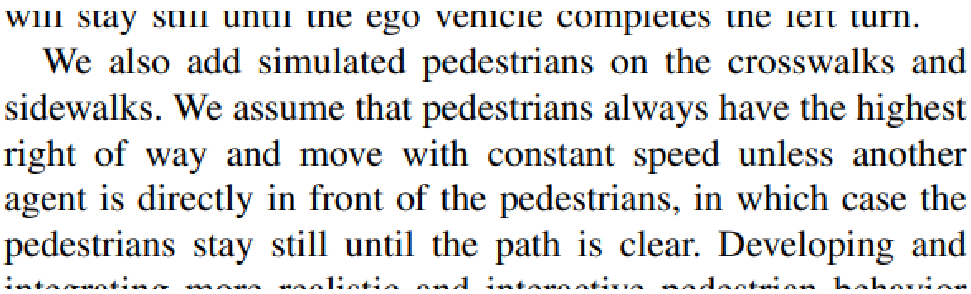


We develop an Intelligent Intersection Driver Model (IIDM) based on the canonical Intelligent Driver Model (IDM) [62], a one-dimensional car-following model with tunable parameters [63] that drives along a reference path. In the canonical IDM, the longitudinal position and velocity in Frenét coordinates are computed by

$$\frac{dv}{dt} = a_{\max}\left[1-\left(\frac{v}{v^*}\right)^{\delta}-\left(\frac{s^*(v,\Delta v)}{s}\right)^{2}\right], \tag{5}$$

$$s^*(v,\Delta v) = s_0 + Tv + \frac{v\Delta v}{2\sqrt{a_{\max}b_{\text{comf}}}}, \tag{6}$$

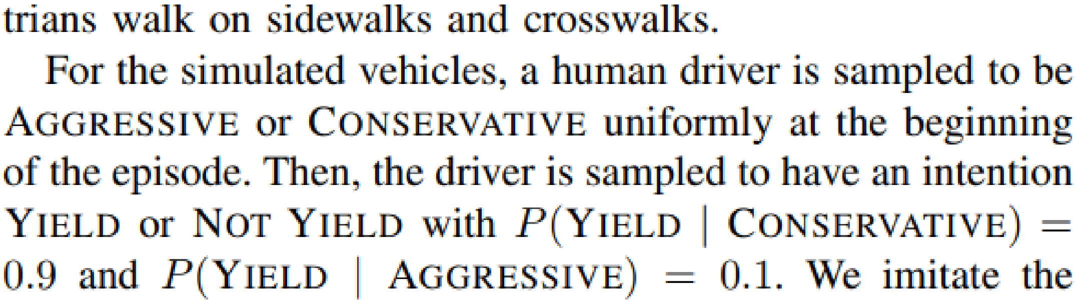
2. the peds are simulated as



will stay still until the ego vehicle completes the left turn.

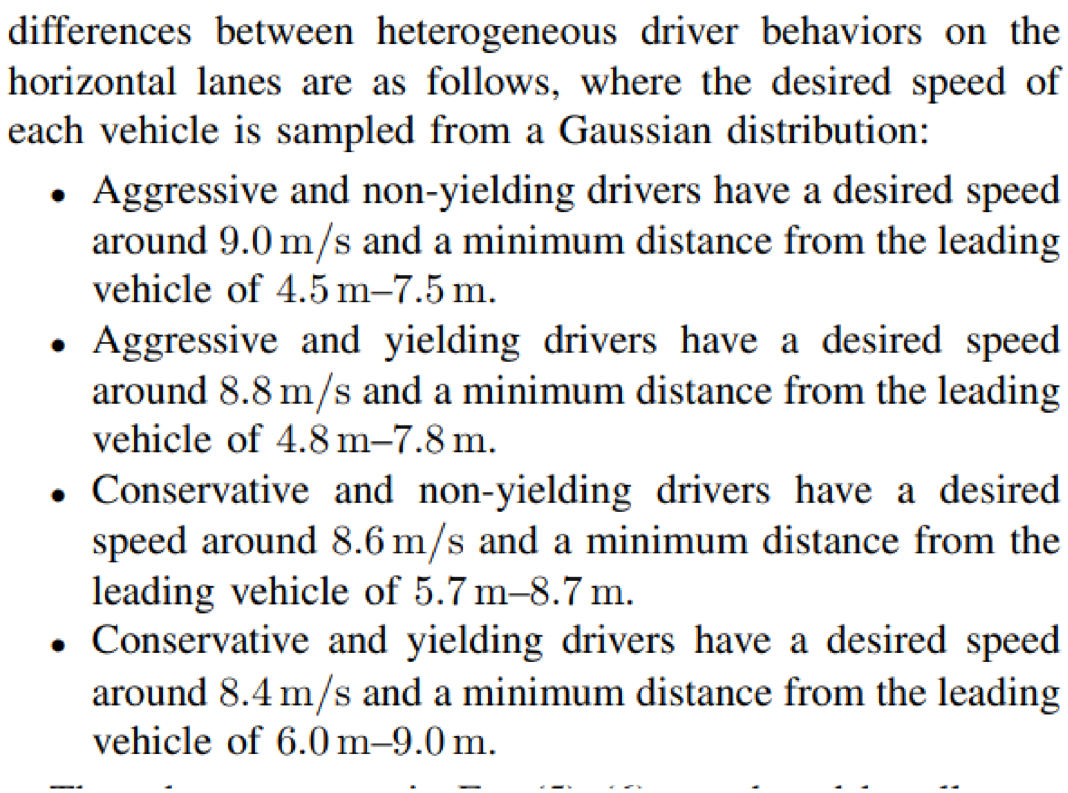
We also add simulated pedestrians on the crosswalks and sidewalks. We assume that pedestrians always have the highest right of way and move with constant speed unless another agent is directly in front of the pedestrians, in which case the pedestrians stay still until the path is clear. Developing and integrating more realistic and interactive pedestrian behavior

3. ego car init. in random, obs. cars on lane, peds on crosswalk/sidewalk



trians walk on sidewalks and crosswalks.

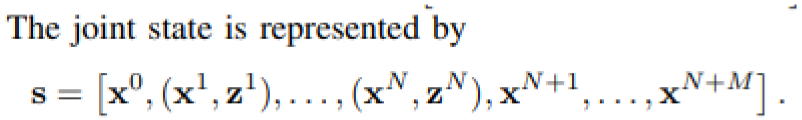
For the simulated vehicles, a human driver is sampled to be Aggressive or Conservative uniformly at the beginning of the episode. Then, the driver is sampled to have an intention Yield or Not Yield with $P(\text{Yield} \mid \text{Conservative}) = 0.9$ and $P(\text{Yield} \mid \text{Aggressive}) = 0.1$. We imitate the



differences between heterogeneous driver behaviors on the horizontal lanes are as follows, where the desired speed of each vehicle is sampled from a Gaussian distribution:

- Aggressive and non-yielding drivers have a desired speed around $9.0\,\mathrm{m/s}$ and a minimum distance from the leading vehicle of $4.5\,\mathrm{m}$–$7.5\,\mathrm{m}$.
- Aggressive and yielding drivers have a desired speed around $8.8\,\mathrm{m/s}$ and a minimum distance from the leading vehicle of $4.8\,\mathrm{m}$–$7.8\,\mathrm{m}$.
- Conservative and non-yielding drivers have a desired speed around $8.6\,\mathrm{m/s}$ and a minimum distance from the leading vehicle of $5.7\,\mathrm{m}$–$8.7\,\mathrm{m}$.
- Conservative and yielding drivers have a desired speed around $8.4\,\mathrm{m/s}$ and a minimum distance from the leading vehicle of $6.0\,\mathrm{m}$–$9.0\,\mathrm{m}$.

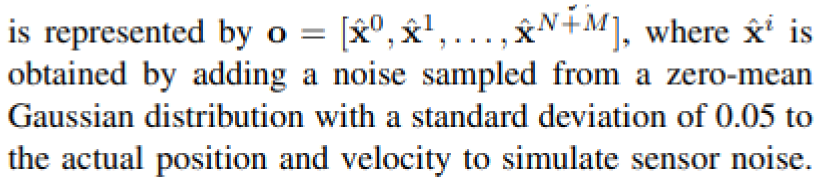
2. POMDP
   1. state: x are pos, vel, and type, z is intention



The joint state is represented by

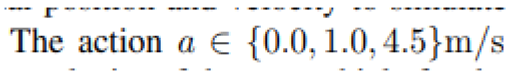
$$\mathbf{s} = \left[\mathbf{x}^0, (\mathbf{x}^1, \mathbf{z}^1), \ldots, (\mathbf{x}^N, \mathbf{z}^N), \mathbf{x}^{N+1}, \ldots, \mathbf{x}^{N+M}\right].$$

   2. observation:



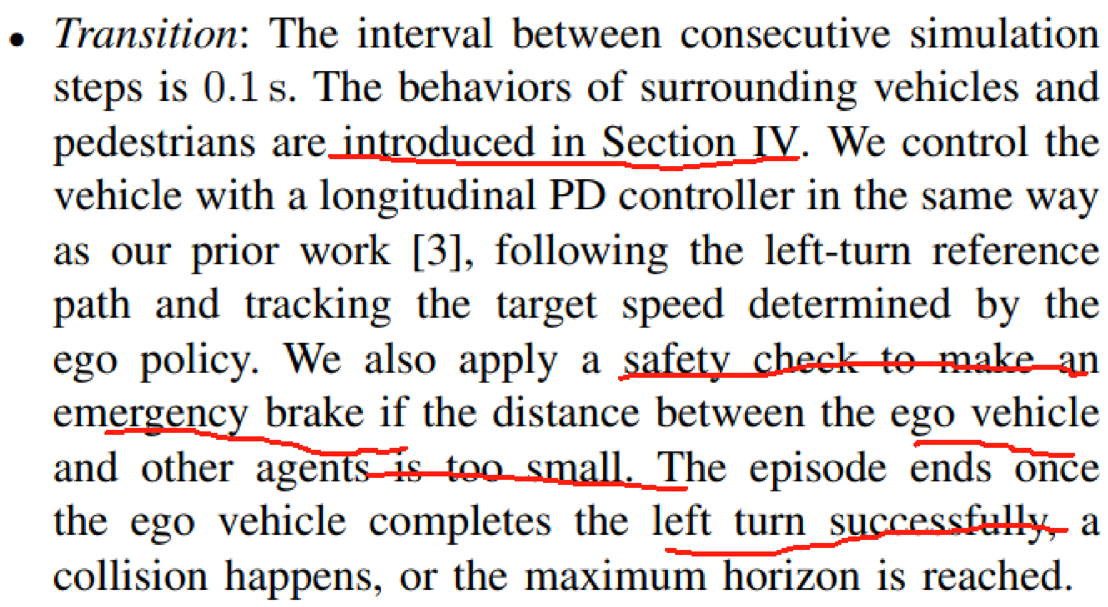
is represented by $\mathbf{o} = [\hat{\mathbf{x}}^0, \hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{N+M}]$, where $\hat{\mathbf{x}}^i$ is obtained by adding a noise sampled from a zero-mean Gaussian distribution with a standard deviation of 0.05 to the actual position and velocity to simulate sensor noise.

   3. action:



The action $a \in \{0.0, 1.0, 4.5\}\mathrm{m/s}$

   4. transition



- *Transition*: The interval between consecutive simulation steps is $0.1\,\mathrm{s}$. The behaviors of surrounding vehicles and pedestrians are introduced in Section IV. We control the vehicle with a longitudinal PD controller in the same way as our prior work [3], following the left-turn reference path and tracking the target speed determined by the ego policy. We also apply a safety check to make an emergency brake if the distance between the ego vehicle and other agents is too small. The episode ends once the ego vehicle completes the left turn successfully, a collision happens, or the maximum horizon is reached.

5. reward

$$R(s,a) = \mathbf{1}\{s \in S_{\text{goal}}\} r_{\text{goal}} + \mathbf{1}\{s \in S_{\text{col}}\} r_{\text{col}} + r_{\text{speed}}(s),$$

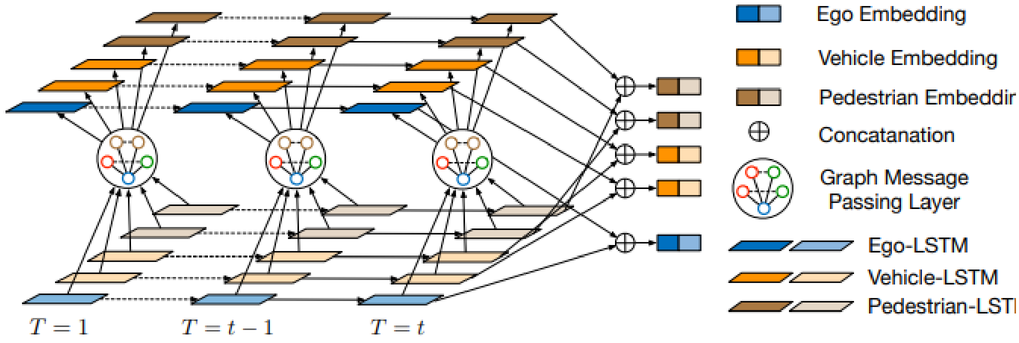
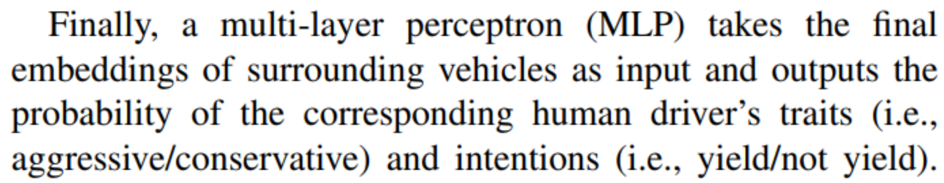
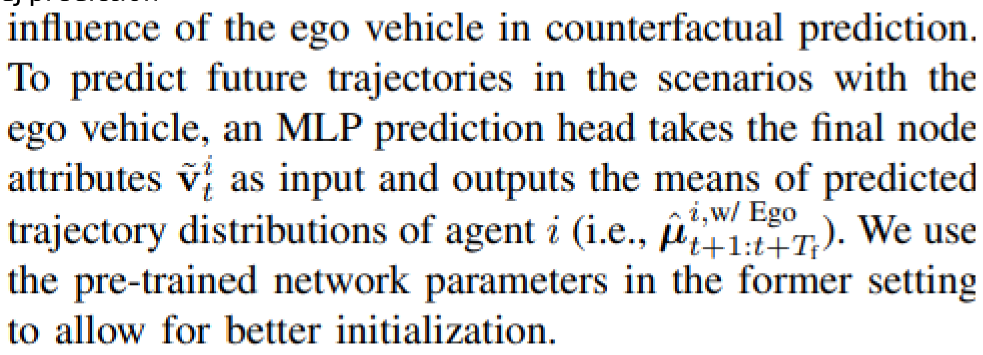
3. DRL
    1. Internal State Inference
        1. goal $p\left(\mathbf{z}_t^i \mid \mathbf{o}_{1:t}\right),$
        2. use GT from simulator, use classification models
    2. Graph-Based Representation Learning
        1. 

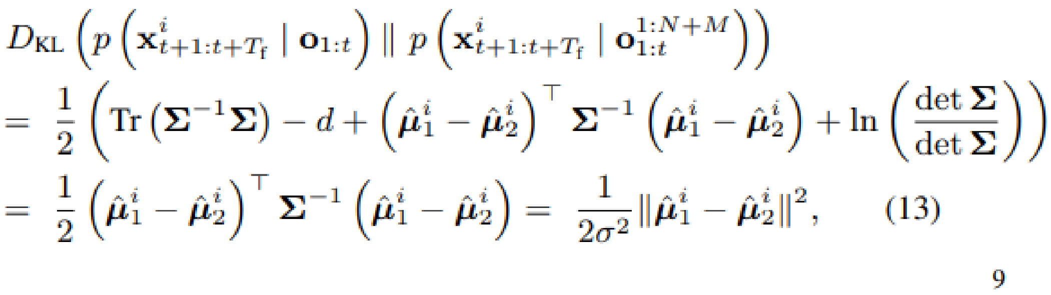
           Fig. 2. A general diagram of the graph-based encoder, which consists o two LSTM layers and a graph message passing layer in between. We us different LSTM networks to extract features for the ego vehicle, surroundin vehicles, and pedestrians, respectively. The outputs of the two LSTM laye are concatenated to generate their final embeddings. Best viewed in color.
        2. 

           Finally, a multi-layer perceptron (MLP) takes the final embeddings of surrounding vehicles as input and outputs the probability of the corresponding human driver's traits (i.e., aggressive/conservative) and intentions (i.e., yield/not yield).
    3. traj prediction

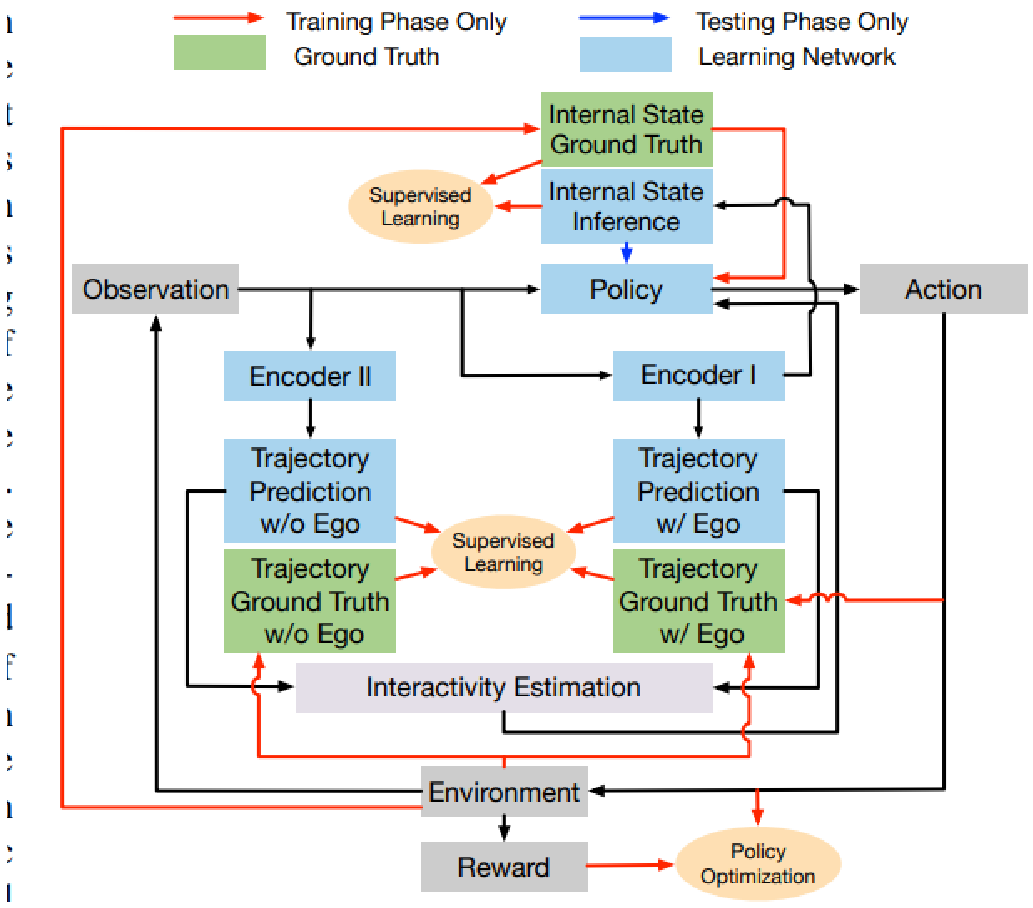
        

        influence of the ego vehicle in counterfactual prediction. To predict future trajectories in the scenarios with the ego vehicle, an MLP prediction head takes the final node attributes $\tilde{\mathbf{v}}_t^i$ as input and outputs the means of predicted trajectory distributions of agent i (i.e., $\hat{\boldsymbol{\mu}}_{t+1:t+T_f}^{i,\text{w/ Ego}}$). We use the pre-trained network parameters in the former setting to allow for better initialization.

        1. use GNN as encoder, mlp output mean estimation, freezed during training the system as a whole.
    4. Interactivity Estimation
        1. function in terms of KL(traj w/ego, w/o ego)

5. 

$$\begin{aligned}
&D_{\mathrm{KL}}\left(p\left(\mathbf{x}^i_{t+1:t+T_{\mathrm{f}}} \mid \mathbf{o}_{1:t}\right) \parallel p\left(\mathbf{x}^i_{t+1:t+T_{\mathrm{f}}} \mid \mathbf{o}^{1:N+M}_{1:t}\right)\right) \\
&= \frac{1}{2}\left(\mathrm{Tr}\left(\boldsymbol{\Sigma}^{-1}\boldsymbol{\Sigma}\right) - d + \left(\hat{\boldsymbol{\mu}}^i_1 - \hat{\boldsymbol{\mu}}^i_2\right)^\top \boldsymbol{\Sigma}^{-1}\left(\hat{\boldsymbol{\mu}}^i_1 - \hat{\boldsymbol{\mu}}^i_2\right) + \ln\left(\frac{\det \boldsymbol{\Sigma}}{\det \boldsymbol{\Sigma}}\right)\right) \\
&= \frac{1}{2}\left(\hat{\boldsymbol{\mu}}^i_1 - \hat{\boldsymbol{\mu}}^i_2\right)^\top \boldsymbol{\Sigma}^{-1}\left(\hat{\boldsymbol{\mu}}^i_1 - \hat{\boldsymbol{\mu}}^i_2\right) = \frac{1}{2\sigma^2}\left\|\hat{\boldsymbol{\mu}}^i_1 - \hat{\boldsymbol{\mu}}^i_2\right\|^2, \qquad (13)
\end{aligned}$$

6. 
Training Phase Only
Testing Phase Only
Ground Truth
Learning Network
Internal State Ground Truth
Internal State Inference
Supervised Learning
Observation
Policy
Action
Encoder II
Encoder I
Trajectory Prediction w/o Ego
Trajectory Prediction w/ Ego
Supervised Learning
Trajectory Ground Truth w/o Ego
Trajectory Ground Truth w/ Ego
Interactivity Estimation
Environment
Reward
Policy Optimization