On the Utility of Model Learning in HRI

Rohan Choudhury*
California Institute of Technology
rchoudhury@caltech.edu

Gokul Swamy*
UC Berkeley
gokul.swamy@berkeley.edu

Dylan Hadfield-Menell
UC Berkeley
dhm@eecs.berkeley.edu

Anca D. Dragan
UC Berkeley
anca@berkeley.edu

***Abstract*—Fundamental to robotics is the debate between model-based and model-free learning: should the robot build an explicit model of the world, or learn a policy directly? In the context of HRI, part of the world to be modeled is the human. One option is for the robot to treat the human as a black box and learn a policy for how they act directly. But it can also model the human as an agent, and rely on a "theory of mind" to guide or bias the learning (grey box). We contribute a characterization of the performance of these methods under the optimistic case of having an ideal theory of mind, as well as under different scenarios in which the assumptions behind the robot's theory of mind for the human are wrong, as they inevitably will be in practice. We find that there is a significant sample complexity advantage to theory of mind methods and that they are more robust to covariate shift, but that when enough interaction data is available, black box approaches eventually dominate.**

***Index Terms*—theory of mind, inverse RL, model-based RL, model-free RL, sample complexity**

I. Introduction

An age-old debate that still animates the halls of computer science, robotics, neuroscience, and psychology departments alike is that between model-based and model-free (reinforcement) learning. Model-based methods work by building a model of the world – the dynamics that tells an agent how the world state will change as a consequence of its actions – and optimizing a cost or reward function under the learned model. In contrast, model-free methods never attempt to explicitly learn how the world works. Instead, the agent learns a policy directly from acting in the world and learning from what works and what does not. Model-free methods are appealing because the agent implicitly learns what it needs to know about the world, and *only* what it needs. Model-based methods are appealing because knowing how the world works might enable the agent to *generalize* beyond its experience, and possibly be able to explain why a decision is the best one.

In neuro- and cognitive science, the debate is about which paradigm best describes human learning [1], [2]. On the other side of campus, in AI and robotics, the debate is instead about which paradigm enables an agent to perform its task best. As of today, model-free methods have produced more successes [3]–[5], but some efforts are shifting towards model-based methods as well [6], [7].

In the context of Human-Robot Interaction (HRI), which is our focus in this work, the debate has a different nuance. For robots that do not work in isolation, but in worlds that contain people, the dynamics of the world is no longer just about how physical state changes, but also how *human* state changes – what the human will do next, and how that is influenced by the robot's action. There is thus a lot of richness in what it means to be model-based. On the one hand, the robot can learn a model by observing human state transitions and fitting a policy to them, as it can with any other part of the environment (Fig. 1,b). This is a black box approach to system identification. But on the other hand, the robot can structure its model and explicitly reason about the human differently: humans, unlike objects in the world, have *agency*, and treating them as such means using what Gopnik called a "Theory of Mind" (ToM) [8]: a set of assumptions about how another agent works, including how they decide on their actions (Fig. 1,c). Rather than black box, this is a gray box approach.

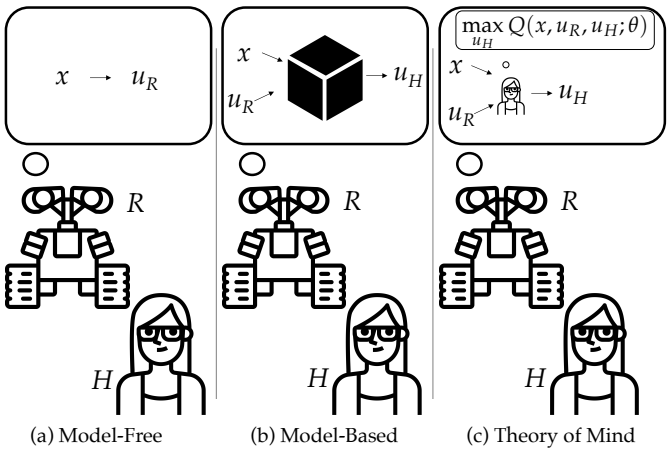

Fig. 1: We characterize the performance of three HRI paradigms that impose increasingly more structure: (a) in model-free, the robot learns a policy for how to act directly, without modeling the human; (b) in black-box model-based, the robot learns a policy for how the human acts, and uses it when optimizing its reward; (c) in theory of mind, the robot further assumes that the human optimizes a reward function with unknown parameters.

When it comes to our own interaction with other people, cognitive science research has amassed evidence that we might use such a theory of mind [9]–[12] – in particular, that we assume that others are approximately *rational*, i.e. they tend to make the decisions that are approximately optimal under some objective (or utility, or reward) [13]. But even if humans do use such tools in interaction, it is not at all clear that robots ought to as well. For robots and interaction, methods from all three paradigms exist: model-free [14]–[16], regular model-based [17], and ToM-based [18]–[22].

* These authors contributed equally to this work.

Ideally, we'd want to settle the debate for HRI by seeing what works best in practice. Unfortunately, several factors make it very difficult to get a definitive answer: 1) we do not yet have the best ToM assumptions we could get, as research in the psychological sciences and even economics is ongoing; as a result, any answer now is tied to the current models we have, not the ones we could have; 2) making the comparison requires immensely expensive evaluations with robots acting in the real world and failing in their interactions around real people – this is very difficult especially for safety-critical tasks; 3) the answer might depend on the amount of data that can be available to the robot, which is also hard to predict – therefore, what we need to know is which paradigm to use as a function of the data available.

In this work, rather than attempting the practical question of which paradigm the robot should use, our idea was to turn to a more scientific question. *We do not know how wrong the eventual ToM model will still be, and we do not know how much data we can afford, but we can compare the performance of these paradigms under different possible scenarios.*

We take inspiration from recent work in learning for control [23] that studied the sample complexity of model-based and model-free methods for a very simple system: the linear quadratic regulator. Their idea was that performance in a controlled simple system that we have ground truth for is informative – if a method struggles even on this system, what chance does it have in the real world?

We thus set up a simplified HRI system: we have a robot that needs to optimize a reward function in the presence of a human who optimizes theirs, in response to the robot's actions. We instantiate this in an autonomous driving domain. This simplification from the real world enables us to start answering two fundamental questions: 1) how big the benefit of ToM would actually be, even in the optimistic case of having the perfect set of assumptions about human decision making; and 2) how exactly this benefit decreases as our assumptions become increasingly incorrect.

We contribute evidence which suggests that if we had a perfect theory of mind, we'd greatly reduce the sample complexity of learning compared to black-box model-based learning, without requiring human-robot interaction data, and relying solely on offline (off-policy) demonstrations. Furthermore, model-based learning has drastically lower sample complexity than model-free learning, consistent with the findings in [23]. ToM can be surprisingly robust in some cases to wrong assumptions: in our driving domain, if we are wrong about the way the person is making predictions about the robot, ToM will still learn a useful model overall. However, when the person deviates too much from the ToM, black-box model-based is unencumbered by the assumptions and ends up dominating. The caveat is the amount (and type) of data required. In low-data regimes, ToM is still better, even without requiring on-policy exploration. Further, ToM transfers better: when we used a trained model to interact in a new HRI system, the ToM model performed better.

These findings should be of course taken with a grain of salt. The differences we introduce between the ToM and the ground truth "human" might not be representative of what differences we will see in real life (despite our efforts to create an analogy). We also do this for one particular task, with one particular ToM instantiation. However, we find the results useful in giving us at least a glimpse at what we might expect. We are also encouraged that the results are consistent with the lens of bias vs. variance in machine learning in general. The distinction between a ToM-based and a black-box model-based method is that ToM imposes additional structure on the problem: a bias. On the one hand, bias is useful because it can prevent overfitting, and we see this happen even with bias that is not completely correct in low data regimes. In high data regimes, incorrect bias leads to underfitting, and less biased methods shine.

Overall, we are excited to contribute a quantitative comparison of these paradigms for HRI, along with an analysis of their degradation as we make the wrong modeling assumptions, decrease the amount of data available, or restrict the ability to collect data on-policy.

II. Interaction Learning Algorithms

A. Notation

For all of the following methods and experiments, we denote the human plan at time t by $\mathbf{u}_H^t$, and the robot plan at time t by $\mathbf{u}_R^t$. We also denote the action executed by the human at time t by u_H^t, and for the robot u_R^T. Both the human and robot states at a time t are denoted by x_H and x_R.

B. Robot Objective

The robot's goal is to optimize a reward function $r_R(x_R, x_H, u_R, u_H)$ that depends on both its state and action, as well as the human's state and action.

C. Theory-of-Mind-Based Learning

In line with previous work [10]–[12] , our ToM will assume that the human optimizes a reward function. The robot will focus the learning on figuring out this reward via inverse reinforcement learning, and the ToM-based method will plan the robot's actions using the learned model.

In particular, the reward will have the form

$$r_H(x_H, x_R, u_H, u_R) = \theta^T \phi(x_H, x_R, u_H, u_R)$$

where θ is a vector of weights and ϕ is a feature map from the current state of the system, which depends on the robot's state and action as well, akin to the robot's reward. We describe the particular features we assumed in more detail in Sec. III-A.

To optimize it, the person needs to reason about the interdependency between their future actions and robot's. Work on ToM has investigated different ways to capture this, from infinite regress (I think about you thinking about me thinking about you..) to capping the regress to one or two levels. Our particular instance of ToM is based on prior work which avoids regress by giving the human access to the robot's future plan [22], and is given by

$$\mathbf{u}_H^*(\mathbf{u}_R) = \arg\max_{\mathbf{u}_H} \mathcal{R}_H(x_H, x_R, \mathbf{u}_H, \mathbf{u}_R) \qquad (1)$$

with $\mathcal{R}$ denoting the cumulative reward, i.e. the sum of rewards over a finite horizon of the length of the trajectories $\mathbf{u}_H$ and $\mathbf{u}_R$. This assumption is very strong, i.e. that the human can read the robot's mind. Part of our goal is to simulate the human as instantiating other approaches, and teasing out how useful or not ToM still is when its assumptions are wrong. This particular ToM instance will also assume that the person computes this at every step, takes the first action, observes the robot's new plan, and replans.

To leverage this ToM model, the robot needs to know the human reward function r_H. In our experiments, we create a dataset of demonstrations $\mathcal{D}$ by placing our ground-truth human (be it a human perfectly matching the ToM model or one that does not) around another vehicle executing various trajectories, and recording the human's response. In the real world, this data would be collected from people driving in response to other people. We then run inverse reinforcement learning [22], [24]–[26] on $\mathcal{D}$ to obtain weights θ.

Finally, given the human reward function described by the learned weights θ, the robot optimizes its own plan:

$$\mathbf{u}_R^* = \arg\max_{\mathbf{u}_R} \mathcal{R}_R(x_R, x_H, \mathbf{u}_R, \mathbf{u}_H^*(\mathbf{u}_R))$$

with $\mathcal{R}_R$ the robot's cumulative reward. The robot plans, takes the first action, observes the next human action, and replans at every step. We use a Quasi-Newton optimization method [27] and implicit differentiation to solve this optimization problem.

D. Black-Box Model-Based Learning

The Theory-of-Mind method models the human's actions as explicitly optimizing some cost function. An alternative is to learn this function directly from data via, e.g., a conditional neural network, as in [17].

Vanilla. In the "vanilla" black-box model-based approach, we collect a training dataset of human-robot interactions as in the ToM approach and fit a neural network to it. This gives a model of human behavior that achieves low validation error. We generate a dataset $\mathcal{D}$ of generic demonstrations in the same manner as the ToM learner. We fit a neural network f to $\mathcal{D}$. This allows us to estimate the human plan $\mathbf{u}_H$, given the human and robot histories, the current state of the system, and the robot plan:

$$\mathbf{u}_H = f(H_R, H_H, x_R, x_H, \mathbf{u}_R).$$

H_R and H_H are the state-action histories of the robot and human over a finite time horizon. The model f is trained by minimizing the loss between $f(.)$ and the observed data $\mathbf{u}_H$ in $\mathcal{D}$. Specifically, we use a neural network with three fully connected layers, each with 128 weights and ReLU activations, and train the network using ADAM [28].

Given this learned model, the vanilla model-based method generates a plan with trajectory optimization, just like the ToM method:

$$\mathbf{u}_R^* = \arg\max_{\mathbf{u}_R} \mathcal{R}_R(x_R, x_H, \mathbf{u}_R, f(H_R, H_H, x_R, x_H, \mathbf{u}_R)).$$

The distinction is whether the prediction about $\mathbf{u}_H$ comes from optimizing $\mathcal{R}_H$, or from the black box predictor f.

Covariate-Shift-Robust. The vanilla approach ignores the fact that predicting the human's behavior is a *sequential* problem. In fact, even if f attains a test error rate of ϵ it can still exhibit prediction errors that are $O(\epsilon T^2)$ when rolled out over a horizon T [29]. The trajectories generated by optimizing against a fixed model induce *covariate shift* that reduces the accuracy of the model.

To deal with this, we adopt a typical approach in system identification: we alternate between fitting a model, and using it to act and collect more interaction data. This comes at a cost: the need for data collected on-policy, from interacting with the human, rather than from (off-policy/offline) demonstrations.

We refer to this as covariate-shift robust model-based learning. In this method, we again use a neural network f to predict the human actions. But unlike the previous model-based method, which relies on training data collected beforehand, here we train f *iteratively*.

Of course, needing interaction with the human can be prohibitive, especially if a) a lot of interaction data is needed, or b) the robot does not perform well initially, when its model is not yet good, which can harm adoption or lead to safety concerns.

E. Model-Free Learning

A final approach is to employ fully model free methods, such as policy gradients or DQNs. These methods are quite general and fast online as they make no assumptions about the environment and don't explicitly plan online. We use Proximal Policy Optimization (PPO) [5], a model free reinforcement learning algorithm that has had strong results in other continuous control tasks. Although more sample efficient approaches exist, we selected PPO because it has been adopted as a sort of baseline for continuous control tasks.

PPO works by computing clipped gradients of expected reward with respect to the parameters of a policy. This gradient is estimated with rollouts using the current policy parameters in the environment. The algorithm alternates between rolling out trajectories and performing gradient updates. See [5] for a more complete explanation of the approach. [1]

III. Performance under Correct Modeling Assumptions

We begin with comparing these methods in a driving domain.

[1]We used a fully connected network with 2 hidden layers of width 128. We used a large batch size of 8192 frames to give PPO the best chance to reach high final performance. The primary difficulty in applying PPO to this setting is that the human simulator we implemented for testing what happens where ToM assumptions are exactly right, does not, strictly speaking, fit into the environment model used for reinforcement learning. Because the human reacts to what the robot *plans* to do in the future, the environment is different depending on the current policy parameters. We implement this by adding the robot's policy parameters to the cost function the human optimizes. The human then optimizes their trajectory, taking gradients through the robot's policy. We also test PPO against human simulators that do not require access to the robot's plan, and discuss the result in the last section.

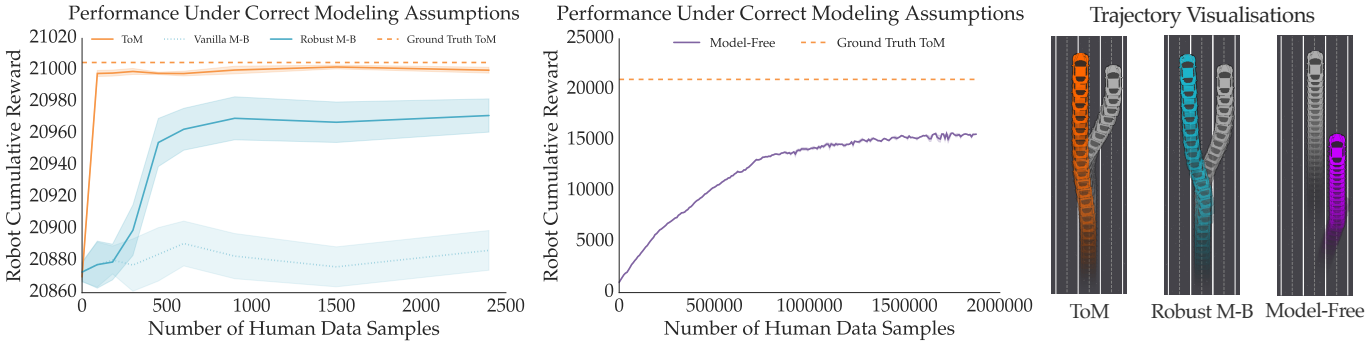

Fig. 2: The test rewards of the interaction learning algorithms on the scenario with the ground truth human simulator. The ToM learner has the smallest sample complexity and best performance, followed by the covariate-shift robust model based method. The 'vanilla' model-based method does poorly. The ToM is able to pass the human car with the least movement out of its lane, and thus obtains the highest reward.

A. Experiment Design

(Ground Truth) Human Simulator. For our driving domain, states are tuples of the form (x, y, v, α), where x and y are the positional coordinates, v the speed, and α the heading. The actions are $u = (a, \omega)$, where a is a linear acceleration, and ω an angular velocity.

The ground truth human simulator plans forward over a finite time horizon T (in all experiments, $T = 5$) by optimizing over a linear combination of features:

Car Proximity: This cost is based on the distance between the robot and human cars, which represents human's desire to not hit another vehicle. Given the human state $(x_H, y_H, v_H, \alpha_H)$ and the robot state $(x_R, y_R, v_R, \alpha_R)$, this cost is given by $\mathcal{N}((x_R, y_R)|(x_H, y_H, \sigma^2_{car})$.

Lane Edge Proximity: This cost is based on the distance to the nearest lane edge. This represents how humans generally prefer to stay in the middle of their lane. Letting the left edge of some lane be L_l and the right edge R_l, the lane cost is given by: $\mathcal{N}(L_1|x_t, \sigma^2_{lane}) + \mathcal{N}(R_l|x_t\sigma^2_{lane})$.

Forward Progress: This cost is based on the vertical distance between the next state and the current state, representing how humans want to go forward when driving. This cost is given by: $-(y_{t+1} - y_t)$.

Bounded Control: This cost is based on accelerating or trying to turn more quickly than certain bounds, representing how humans prefer smoother rides and cars have actuator limits. The cost is given by $\exp(a - a_{max}) + \exp(\omega - \omega_{max})$.

Offroad: This cost represents how drivers want to stay on the road when driving. Letting the left edge of the road being R_l and the right edge R_r, the offroad cost is given by $\exp(x_t - R_l) + \exp(R_r - x_t)$.

The weights on these features were tuned to produce plausible/natural driving in a series of scenarios. In addition to the features and weights, the ground truth human simulator is given the plan of the robot $\mathbf{u}_R$. It then solves the cost minimization in (1). Importantly, this particular simulator exactly matches the assumptions made by the ToM learner (Section II-C), in both the features used and planning method. The next section modifies the simulator so that the ToM assumptions are wrong.

Environment. This experiment environment consists of the human and robot car on a road, with the robot beginning behind the human. The robot has a similar reward to that of the human, incentivizing it to make progress, avoid collisions, keep off of the lane boundaries, and stay on the road. However, we set the weight for forward progress to 10 times its value for the Ground Truth Human Simulator to incentivize the robot to be more aggressive.

We chose this as our environment because for all its simplicity, it can actually capture sophisticated interaction: given our ground truth human, to do well, the robot should not actually just stay in its lane and brake, nor should it go very far out of its lane and overtake: the optimal behavior for the robot in this environment is to *influence the person to make space, thus needing to get minimally away from the center of the lane*. Of course, we do not argue that this is what real robots out into the world should do, but we use is it as an interesting challenge for HRI, because it requires being able to account for robot's *influence* on the human.

Manipulated Variables. We manipulate two variables: the interaction learning algorithm (with the options described in the previous section) and the amount of data (number of samples) the learner gets access to. The u_R and u_H collected in training data and used in learning algorithms are not *plans*, but rather *actions* that have been executed by the robot and human, since in reality, humans can only react to and learn from actions they can physically observe.

Dependent Measures. In each experiment, we measure the reward the robot accumulates after training over 25 test environments drawn from the same initial state distribution as the training environments. We train ToM and model-based 30 times with each data sample size and measure test reward for each trained model. We train model-free only once due to the several orders of magnitude larger amount of data required.

B. Analysis

We plot the results and visualize trajectories from each paradigm in Fig. 2. We see that the ToM learner has the lowest sample complexity, as well as the highest numerical performance. Given that the ToM learner has the exact reasoning model used by the ground truth human simulator,

this is expected. The comparatively low sample complexity demonstrates how inverse reinforcement learning is capable of determining satisfactory weights after seeing very little data if all of its assumptions are satisfied. Looking at the trajectory the robot produces from one of the 25 initial states, since the ToM learner has the 'perfect model', it is able to force the human robot of out of the center lane while staying on an almost straight course itself.

On the other hand, the 'vanilla' model based method, which learns from the same demonstrations used by the ToM learner, is unable to learn to pass the other car. One explanation for this behavior is the difference between the training and test distributions - at test time the robot plans with the learned model, generating different kinds of trajectories with different human responses than in training.[2] The ToM model can better cope with this because it has learned the correct weights of various features in the reward function, and can rely on trajectory optimization to produce the corresponding human trajectories in new situations, given new robot trajectories.

When using online interaction data as opposed to offline demonstrations, we see that the covariate-shift robust model-based method is able to eventually almost match the performance of ToM. It also is able to learn to pass the human car, but does so less smoothly than the ToM learner, having to switch lanes and allowing the human to move less out of the way. While this seems nicer to the person, remember that this is not the goal in this particular domain – we are telling the robot to optimize purely selfishly for its own reward. We would expect a similar difference between ToM and the robust model-based method for a different robot reward that encourages courtesy or the progress of the person. We attribute the robust model-based method's inability to catch up to the cumulative reward achieved by the ToM to the fact that despite the iterative training, it is possible to still converge to an inaccurate human model – the model will be accurate in predicting the kind of trajectories the robot ended up prompting from the human but not necessarily to the ground-truth optimal trajectories, or the trajectories nearby that the robot is using iteratively as it is running its optimization at test time. Another important point is the sample complexity of this model – it requires 2 orders of magnitude more data than the ToM model.

We used the PPO2 implementation in the Open AI Baselines repository [30] as our model-free algorithm. This method takes several orders of magnitude more than the model-based method and is not able to match its performance. This can likely be attributed to the fact that the model-free method is not handed the dynamics of the system and therefore cannot take advantage of online planning, which is fundamental to the way model-free methods operate.

There is a stark difference between the methods even with 0 samples: even without any data, ToM and model-based optimize the robot's trajectory (under essentially a random human), and can figure out how to make progress (albeit not how to avoid the person). The model-free method has to first learn the dynamics of the system. We attempted to counter this handicap by pretraining the model-free policy on environments where the robot acted in isolation and not counting these samples because they did not require any human interaction. This did not help much as introducing the person left the robot just as confused as when it started. We also tried annealing the proportion of episodes per batch that were in isolation from 1 to 0 (again not counting the isolated episodes) but saw similar training curves.

[2]This might be different depending on how closely the training data matches what the robot's behavior is when planning with the learned model.

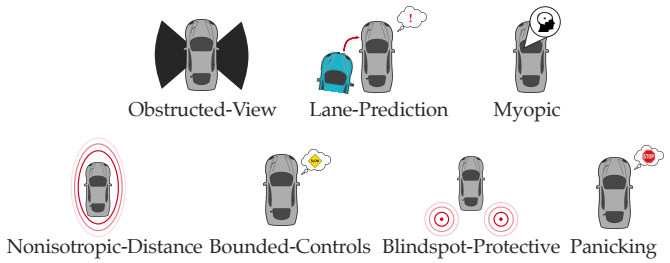

Fig. 3: The various modifications made to the ground truth human simulator. The first row corresponds to modification of planning methodology, while the first three elements of the second row correspond to changes in reward features. The last corresponds to an irrational planning heuristic humans might use while under pressure.

Takeaways. Overall, what we find confirms intuition: if we have a good model, learning its parameters leads to good performance compared to learning from scratch. More surprising is the poor performance of the vanilla model-based method: to get model-based methods to work, it seems like they need to be interactive. What this says is that we might not be able to use black-box models learned based on human-human interaction data: we might need human-robot interaction data, and in particular data obtained from interaction as the robot is still learning. This can be prohibitively expensive or dangerous in many scenarios.

IV. Performance under Incorrect Modeling Assumptions

From what we have found in the previous section, Theory of Mind is appealing because it has low sample complexity and does not require human-robot (on-policy) interaction data for training – human-human data could be sufficient. However, the bias introduced with this approach could lead to underfitting. To quantify this, we compare the ToM and model-based methods when we modify the human simulator. Because of the tremendous difference in terms of performance and sample complexity of the model-free method, we chose to omit it to focus on the aforementioned comparison.

A. Human Simulators that Contradict Modeling Assumptions

Inconsistency in how humans plan, the "features" they might care about, or unexpected reactions to certain actions all violate the assumptions made by the ToM learner. We aimed to create ground truth modifications that are analogous to differences between reality and our ToM-based modeling – things that designers of these systems might get wrong. As

such, even though these are controlled experiments where we know the ground-truth, we think they provide some indication of real-world differences would look like under different hypotheses. We group these modifications into 3 categories:

1) Incorrect model of how the human plans: One way our model could be wrong is if it inaccurately captures the planning process – even if we assume the person is actually trying to optimize for a known reward, they might not optimize well or reason about the robot differently than ToM assumes.
"Obstructed-View" – Humans have blind spots. Our instance of ToM assumes humans have a 360° vision. In reality, drivers have blind spots. To model this we hide cars that are not in a double-cone from the human, with a vertex angle of 45°. Robots that do not model blind spots will thus take more risky maneuvers, expecting to be seen by the person when they are not.
"Lane-Prediction" – Humans can plan conservatively. Given the inherent risks in driving, humans may be more cautious in their planning than necessary, taking evasive maneuvers when there is any chance of danger. This simulator swerves out of the current lane if the robot angles itself slightly towards said lane. A robot that does not know this might not be able to influence the human as much is possible. Note that this no longer matches our ToM's assumption that the human gets access to the robot's plan.
"Myopic" – Humans might not plan ahead for as long as the robot assumes. Another assumption ToM makes about the human's reasoning is that it plans as far forward as the robot does. In reality, however, humans may be more myopic, and plan forward for a shorter time horizon than we assume. Our "Myopic" human simulator only plans forward for one step.

2) Incorrect model of what the human cares about: Another class of inconsistency in modeling deals with reward features.
"Nonisotropic-Distance" – Humans care about avoiding other cars, but we might not know how sensitive they are to getting close to different areas of another car. The original human simulator has a cost based on a Gaussian that takes in the Euclidean distance between the centers of the two cars, the "Nonisotropic-Distance" simulator modifies the cost contours to be longer than they are wide, as show in Figure 3. This models the fact that people are more comfortable with cars behind them than they are with them to their sides.
"Bounded-Controls"– We might not know people's preferences for speed or their control limitations. Human drivers have different preferences for how fast they turn or accelerate as well as cars with different control bounds. To model this, the ground truth human simulator's values of a_{max} and ω_{max} (see section 3.A) are reduced to $\frac{a_{max}}{2}, \frac{\omega_{max}}{2}$, reducing the capability of the human to react to the robot.
"Blindspot-Protective" – Humans might additionally care about not having another car in their blindspot. A subclass of modifications we have not yet considered is where the human might use features in planning that the robot might not know about. One example of this might be discomfort with having cars in one's blindspot. Drivers might speed up or slow down to avoid such an arrangement. This modification models this dislike by adding additional points of Gaussian cost where blindspots are, as illustrated in Fig. 3.

3) Human is using a simple heuristic: **"Panicking" – Humans might behave in irrational heuristic ways.** Humans might also use heuristics that are irrational. Our "Panicking" modification combines the slower speed of the "Bounded-Controls" modification with an additional heuristic of stopping immediately if another car is fewer than 2 car-lengths behind. This is inspired by a newer driver, whose inexperience causes him to drive slowly and panic when another car approaches. This modification is furthest from the ToM assumptions.

B. Analysis

Modifications that maintain ToM's superiority. The performance of the ToM method and robust model-based method across several modifications are show in Figure 4.

In the "Blindspot-Protective", "Obstructed-View", and "Lane-Prediction" modifications, we see that both ToM and robust model-based methods reach almost the same level as they did with the previous simulator. One explanation for this outcome is that when passing the human from behind, the robot is almost never in the human's blindspot and thus almost always in the field of view of the "Obstructed-View" car. The cars start in the same lane so the "Lane-Change" modification has no effect. In all three cases, the ToM has lower sample complexity than the model-based method, indicating it needs less data to converge to an optimum. Notably, the ToM learner also performs better than the model-based, even though its reasoning model does not account for these factors. Though both these models differ from the original model, their interaction with the robot car is for the most part identical - they plan through the same features ϕ with perfect information over the same time horizon. From these three experiments, we find that ToM still performs quite well and with low sample complexity, if its model of the human is close to the truth.

In the "Bounded-Control" experiment, both the ToM and model-based learners converge to optima that are lower than in the previous simulator. This can be attributed to the fact that the human cannot get out of the way quickly, and thus the robot car spends more time close to it. The relative performance of ToM and model-based stays the same.
Modifications that cross the tipping point. In the "Nonisotropic-Distance" experiment, the modification is drastic enough that model-based is able to surpass ToM. While ToM is superior initially, after 900 samples it gets outperformed by model-based when the later has enough data to learn a more accurate predictor.

The myopic modification breaks the assumptions of ToM even more. Fig. 5 shows that while ToM performs well initially, model-based surpasses at 500 samples and converges to a much better reward around 1000 samples. With no data, both learners perform around the same, simply driving forward and crashing into the other cars (points (1) and (2) in the Fig. 5). As is the case with other simulators, ToM converges to a stable solution at around 250 samples. This corresponds to

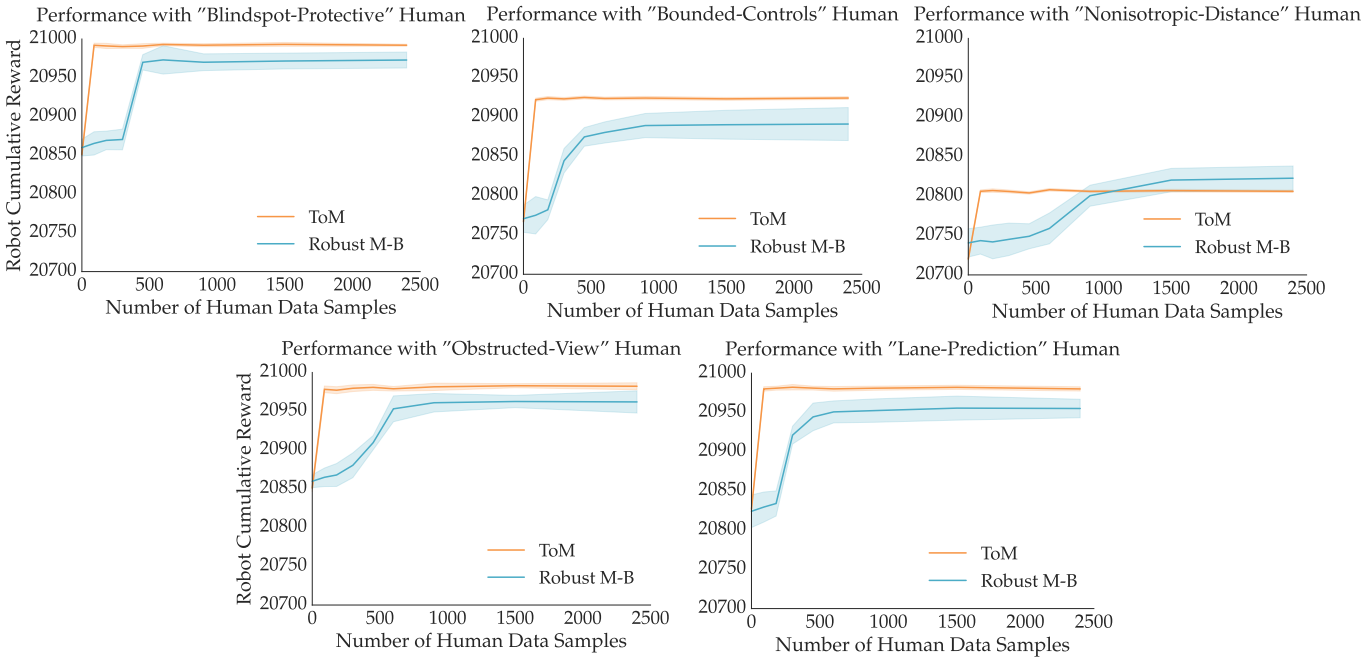

Fig. 4: ToM vs model-based on different simulators. ToM is robust to simulator modifications in some cases but is eventually surpassed by model-based when the difference between assumptions and reality is sufficiently large.

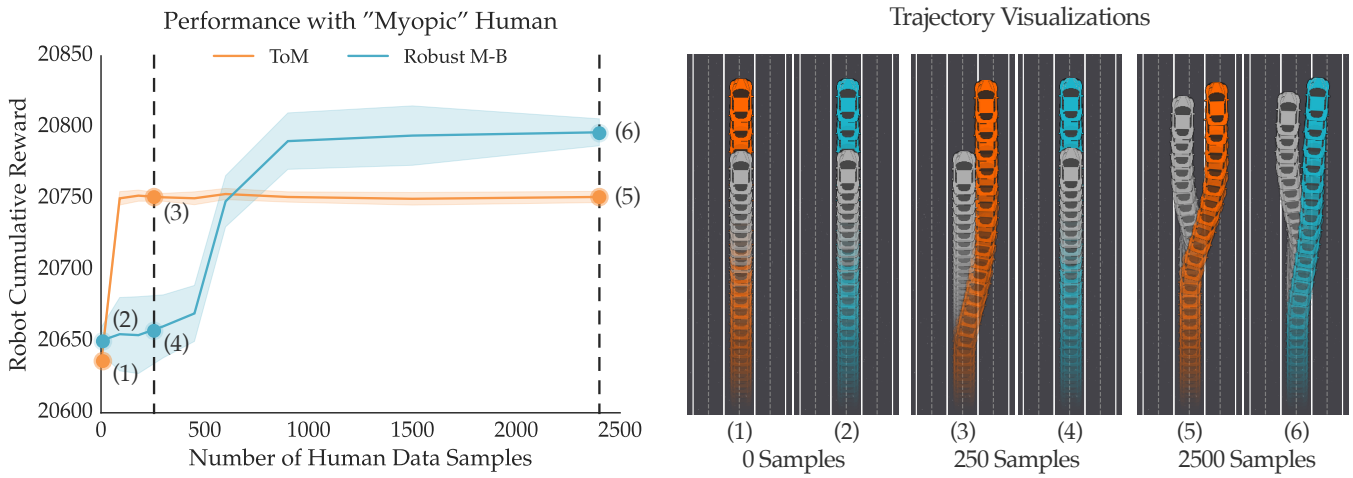

Fig. 5: The cumulative reward and taken trajectories of the interaction learning algorithms with the myopic human simulator. The ToM learner performs better in low-data regimes but the robust model-based method is able to eventually outperform the other method.

changing lanes to make unobstructed progress. At the same number of samples, the model-based method still hits the other car. However, when we get to 2500 samples, the ToM car is still performing around the same while the model-based car has learned it does not need to fully lane-change and can force the other car out of the lane that they share. Because of the difference between the assumptions of the ToM planner and the "Myopic" human, the model-based method is able to learn a model which the ToM does not have the capacity to represent.

Modifications where model-based dominates ToM. The most drastic difference from the ToM assumption we tested as the "Panicking" simulator modification. At the maximum number of samples, the ToM learner manages to barely skirt the stationary panicked human car while the model-based method is able to change lanes and smoothly pass the stopped car at full speed. We also observe that both methods result in lower reward than the ground truth simulator. This is due to the fact that both cars have to either change lanes or drive close to the other car, actions which have high cost.

Takeaways. We establish that there is a tipping point where ToM switches from being robust to being unable to model the human. Before this tipping point, ToM remains superior. At this tipping point, model-based eventually surpasses ToM. Past this tipping point, ToM is drastically inferior. Surprisingly, even some large inaccuracies in ToM fail to harm it enough,

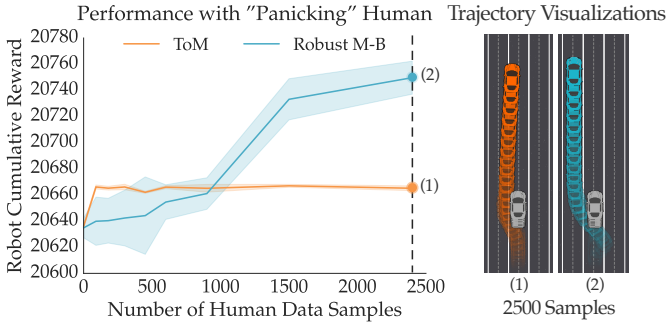

Fig. 6: Rewards and sample trajectories in the Panicking experiment. The robust model-based learner is able to change lanes and pass the human at speed, giving it a higher reward. The ToM learner skirts the human car but still is unable to completely avoid it and gets much lower reward.

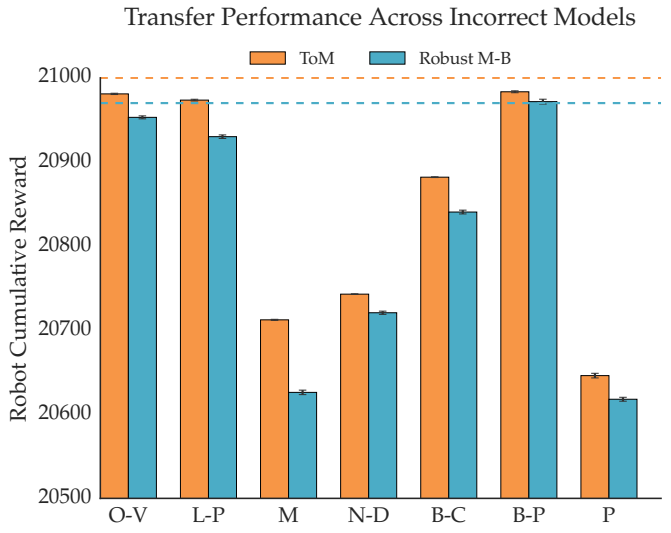

Fig. 7: Results from testing the ground truth models in the modified simulations. The horizontal lines correspond to performance with the ground truth simulator. Both types of models perform less well under drastic simulator modifications. ToM performs better than the robust model-based method across all such experiments.

especially in low-data regimes.

V. Transferability of Learned Models

Finally, we study the transferability of the models trained against the original human. We compare their performance when tested against the modified simulators from the previous section. Fig. 7 shows that all models transfer better when changes are small. As expected, all models perform worse than when they are trained on the correct human data.

Interestingly, ToM is consistently more transferable than the model-based method, even when we violate many of its cardinal assumptions. An explanations for this might be that because neural networks are a higher capacity model, they are less resilient to changes in distribution. This is analogous to overfitting to a narrow dataset with complex models in traditional supervised learning.

Takeaways. ToM seems to be more transferable across the board, even on situations where its assumptions are dramatically different from reality (so different that if model-based were to be re-trained, it would vastly surpass it). This is again explained by its resiliency to covariate shift.

VI. Discussion

We provided what is, to the best of our knowledge, the first comparison between model-free, black-box model-based, and Theory-of-Mind-based methods for interaction. We quantified the performance advantage of ToM-Based when we have made the right assumptions, as well its data collection advantage: it is the only paradigm which does not seem to require human-robot interaction data during learning, and can be trained on observed human-human data instead.

We also found that model-based methods can perform almost as well as ToM, so long as they are trained on-policy. However, they require much more data even when counting it the same as on-policy data (a couple orders of magnitude in our experiments).

Further, we found that model-free methods require several orders of magnitude more data. In a follow-up experiment, we removed the human simulator's dependence on the robot's future plan in order to make the problem easier for model-free methods, but found similar performance.

We also studied what happens as ToM's assumptions become increasingly wrong by emulating the kinds of deviations we might expect to encounter. We found that ToM is robust to small changes, but with large enough differences, model-based methods can vastly surpass ToM. Lastly, we saw that ToM methods transfer better.

Ultimately, our work does not answer the question of which type of model to use: it merely provides evidence for what we should expect given the relative amount of data we can get, whether we have the ability to interact during learning, and how accurate our assumptions about human behavior can get. Even though the evidence is in the context of a particular task, the core findings – the performance gap between ToM and robust model-based, the inability of vanilla (off-policy) model-based to reach the same performance, the utility of ToM with wrong assumptions in low-data regimes and of black-box model-based in high-data regimes – will find echoes in other tasks (but with different scales of data). We ourselves found tremendous value in seeing this evidence, and are excited to share it with the HRI community. We are also excited to research *hybrid* methods that might be able to take advantage of the best of each paradigm: using assumptions in low-data regimes, having flexibility when there is enough data.

Acknowledgment

We thank the members of the InterACT Lab at UC Berkeley. In particular, we are grateful for Kush Bhatia's feedback on building human simulators and Eli Bronstein's assistance on the black-box model-based component of this work.

This work is partially supported by NVIDIA and the Caltech Arjun Bansal and Ria Langheim Summer Undergraduate Research Fellowship.

REFERENCES

[1] J. Gläscher, N. Daw, P. Dayan, and J. P. O'Doherty, "States versus rewards: dissociable neural prediction error signals underlying model-based and model-free reinforcement learning," *Neuron*, vol. 66, no. 4, pp. 585–595, 2010.

[2] S. W. Lee, S. Shimojo, and J. P. ODoherty, "Neural computations underlying arbitration between model-based and model-free learning," *Neuron*, vol. 81, no. 3, pp. 687–699, 2014.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing atari with deep reinforcement learning," *arXiv preprint arXiv:1312.5602*, 2013.

[4] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, "Trust region policy optimization," in *International Conference on Machine Learning*, pp. 1889–1897, 2015.

[5] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[6] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *Advances in neural information processing systems*, pp. 64–72, 2016.

[7] C. Perez, "Predictive learning is the new buzzword in deep learning."

[8] A. Gopnik and H. M. Wellman, "10 the theory theory," *Mapping the mind: Domain specificity in cognition and culture*, p. 257, 1994.

[9] P. Carruthers and P. K. Smith, *Theories of theories of mind*. Cambridge University Press, 1996.

[10] G. Gergely and G. Csibra, "Teleological reasoning in infancy: The naıve theory of rational action," *Trends in cognitive sciences*, vol. 7, no. 7, pp. 287–292, 2003.

[11] B. Sodian, B. Schoeppner, and U. Metz, "Do infants apply the principle of rational action to human agents?," *Infant Behavior and Development*, vol. 27, no. 1, pp. 31–41, 2004.

[12] C. L. Baker, R. Saxe, and J. B. Tenenbaum, "Action understanding as inverse planning," *Cognition*, vol. 113, no. 3, pp. 329–349, 2009.

[13] G. S. Becker, *The economic approach to human behavior*. University of Chicago press, 2013.

[14] S. Nikolaidis and J. Shah, "Human-robot cross-training: computational formulation, modeling and evaluation of a human team training strategy," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pp. 33–40, IEEE Press, 2013.

[15] B. Busch, J. Grizou, M. Lopes, and F. Stulp, "Learning Legible Motion from Human–Robot Interactions," *International Journal of Social Robotics*, pp. 1–15, 2017.

[16] S. Reddy, S. Levine, and A. Dragan, "Shared autonomy via deep reinforcement learning," *arXiv preprint arXiv:1802.01744*, 2018.

[17] E. Schmerling, K. Leung, W. Vollprecht, and M. Pavone, "Multimodal probabilistic model-based planning for human-robot interaction," *arXiv preprint arXiv:1710.09483*, 2017.

[18] B. D. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. K. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pp. 3931–3936, IEEE, 2009.

[19] A. D. Dragan, K. C. Lee, and S. S. Srinivasa, "Legibility and predictability of robot motion," in *Proceedings of the 8th ACM/IEEE international conference on Human-robot interaction*, pp. 301–308, IEEE Press, 2013.

[20] H. Bai, S. Cai, N. Ye, D. Hsu, and W. S. Lee, "Intention-aware online pomdp planning for autonomous driving in a crowd," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 454–460, IEEE, 2015.

[21] S. Javdani, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization," *arXiv preprint arXiv:1503.07619*, 2015.

[22] D. Sadigh, S. S. Sastry, S. A. Seshia, and A. D. Dragan, "Planning for autonomous cars that leverage effects on human actions," in *Proceedings of Robotics: Science and Systems*, RSS '16, 2016.

[23] S. Tu and B. Recht, "Least-squares temporal difference learning for the linear quadratic regulator," *arXiv preprint arXiv:1712.08642*, 2017.

[24] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, (New York, NY, USA), p. 1, ACM, 2004.

[25] S. Levine and V. Koltun, "Continuous inverse optimal control with locally optimal examples," *CoRR*, vol. abs/1206.4617, 2012.

[26] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, pp. 1433–1438, AAAI Press, 2008.

[27] G. Andrew and J. Gao, "Scalable training of l1-regularized log-linear models," in *Proceedings of the 24th international conference on Machine learning (ICML)*, (Corvalis, Oregon), pp. 33–40, 2007.

[28] D. P. Kingma and J. Ba., "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[29] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635, 2011.

[30] P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, Y. Wu, and P. Zhokhov, "Openai baselines," 2017.