

---

# Predicting the behavior of interacting humans by fusing data from multiple sources

---

Erik J. Schlicht<sup>1</sup>, Ritchie Lee<sup>2</sup>, David H. Wolpert<sup>3,4</sup>,  
Mykel J. Kochenderfer<sup>1</sup>, and Brendan Tracey<sup>5</sup>

<sup>1</sup> Lincoln Laboratory, Massachusetts Institute of Technology, Lexington, MA 02420

<sup>2</sup> Carnegie Mellon Silicon Valley, NASA Ames Research Park, Moffett Field, CA 94035

<sup>3</sup> Information Sciences Group, MS B256, Los Alamos National Laboratory, Los Alamos, NM 87545

<sup>4</sup> Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501

<sup>5</sup> Stanford University, Palo Alto, CA 94304

## Abstract

Multi-fidelity methods combine inexpensive low-fidelity simulations with costly but high-fidelity simulations to produce an accurate model of a system of interest at minimal cost. They have proven useful in modeling physical systems and have been applied to engineering problems such as wing-design optimization. During human-in-the-loop experimentation, it has become increasingly common to use online platforms, like Mechanical Turk, to run low-fidelity experiments to gather human performance data in an efficient manner. One concern with these experiments is that the results obtained from the online environment generalize poorly to the actual domain of interest. To address this limitation, we extend traditional multi-fidelity approaches to allow us to combine fewer data points from high-fidelity human-in-the-loop experiments with plentiful but less accurate data from low-fidelity experiments to produce accurate models of how humans interact. We present both model-based and model-free methods, and summarize the predictive performance of each method under different conditions.

## 1 Introduction

The benefit of using high-fidelity simulations of a system of interest is that it produces results that closely match observations of the real-world system. Ideally, algorithmic design optimization would be performed using such accurate models. However, the cost of high-fidelity simulation (computational or financial) often prohibits running more than a few simulations during the optimization process. The high-fidelity models

built from such a small number of high-fidelity data points tend not to generalize well. One approach to overcoming this limitation is to use multi-fidelity optimization, where both low-fidelity and high-fidelity data are used together during the optimization process (Robinson et al., 2006).

When the system of interest involves humans, traditionally human-in-the-loop (HITL) experimentation has been used to build models for design optimization. A common approach in HITL experimentation is to make the simulation environment match the true environment as closely as possible (Aponso et al., 2009). A major disadvantage of this approach is that designing a good high-fidelity HITL simulation requires substantial human and technological resources. Experimentation requires highly trained participants who need to physically visit the test location. As a consequence, collecting large amounts of high-fidelity data under different conditions is often infeasible.

In many cases, one could cheaply implement a lower fidelity simulation environment using test subjects who are not as extensively trained, allowing for the collection of large amounts of less accurate data. Indeed, this is the idea behind computational testbeds like Mechanical Turk (Kittur et al., 2008; Paolacci et al., 2010). The goal of this paper is to extend multi-fidelity concepts to HITL experimentation, allowing for the combination of plentiful low-fidelity data with more sparse high-fidelity data. This approach results in an inexpensive model of interacting humans that generalizes well to real-world scenarios.

In this paper, we begin by outlining the self-separation scenario used as a testbed to study multi-fidelity models of interacting humans. In Section 3 we introduce the multi-fidelity methods used in this paper. In Section 4, we present model-free approaches to multi-fidelity prediction, and in Section 5 we present model-

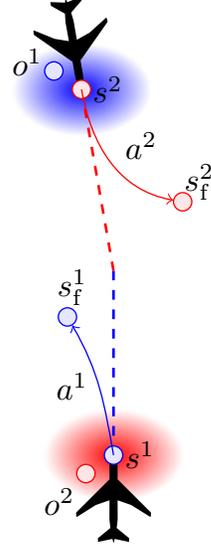
based methods. Results, summarized in Section 6, show the benefit of incorporating low-fidelity data to make high-fidelity predictions and that model-based methods out-perform model-free methods. Section 7 concludes with a summary of our findings and future directions for investigation.

## 2 Modeling Interacting Humans

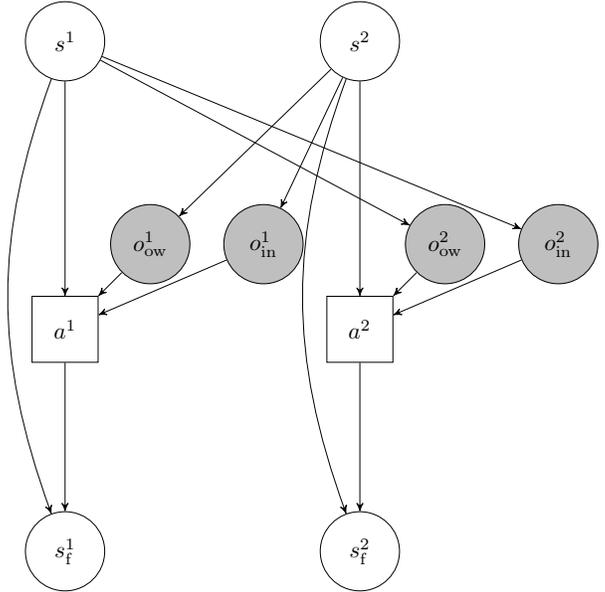
In order to ground our discussion of multi-fidelity methods in a concrete scenario, we will begin by presenting our testbed for studying the behavior of interacting humans. We model the interaction of two pilots whose aircraft are on a near-collision course. Both pilots want to avoid collision (Fig. 1a), but they also want to maintain their heading. For simplicity, we model the pilots as making a single decision in the encounter. Their decisions are based on their beliefs about the other aircraft’s position and velocity, their degree of preference for maintaining course, and their prediction of the other pilot’s decision.

We use a combination of Bayesian networks and game-theoretic concepts (Lee and Wolpert, 2012) to model this scenario. The structure of the model is shown in Fig. 1b. The state of aircraft  $i$  is a four-dimensional vector representing the horizontal position and velocity information. The initial distribution over states is described in Section 3. The action  $a^i$  taken by the pilot of aircraft  $i$  represents the angular change in heading, away from their current heading. Each pilot observes the state of the other aircraft based on their view out the window  $o_{ow}^i$  and their instrumentation  $o_{in}^i$ . Each pilot has a weight  $w^i$  in their utility function reflecting their relative desire to avoid collision versus maintaining heading. We consider the situation where each player knows their own type (i.e., the value of the weight term in their utility function) in addition to the type of the other player (Myerson, 1997).

As described in this section, the pilots chose their actions based on their observations and utility function. After both pilots select an action, those actions are executed for 5 s, bringing the aircraft to their final states  $s_f^i$ . Without loss of generality we describe the scenario from the perspective of one of the pilots, Player 1, with the other aircraft corresponding to Player 2. The model assumptions, detailed below, are symmetric between players. It should be noted that the approaches described in this paper are not limited to cases in which there are only two humans interacting, although scenarios with more humans would increase the amount of data required to make accurate predictions.



(a) Example geometry with model variables depicted.



(b) Model structure. Shaded circles are observable variables, white circles are unobservable variables, squares are decision nodes, and arrows represent conditional relationships between variables.

Figure 1: Visual self-separation game.

## 2.1 Visual inference

Player 1 infers the state of the intruder based on visual information out the window and instrumentation. Player 1 is not able to exactly infer the state of Player 2 as humans are not able to perfectly estimate position and velocity based on visual information (Graf et al., 2005; Schlicht and Schrater, 2007), and the instruments also have noise associated with their measurements (Fig. 1b).

We model the out-the-window visual observation as Gaussian with a mean of the true state of the intruder aircraft and covariance matrix given by  $\text{diag}(900 \text{ ft}, 900 \text{ ft}, 318 \text{ ft/s}, 318 \text{ ft/s})$ . The instrument observation is Gaussian with a mean of the true state of the intruder with covariance given by  $\text{diag}(600 \text{ ft}, 600 \text{ ft}, 318 \text{ ft/s}, 318 \text{ ft/s})$ . The out-the-window velocity uncertainties are derived from the visual psychophysics literature (Weis et al., 2002; Graf et al., 2005).

## 2.2 Decision-making

Theoretical models of multi-agent interaction offer a framework for formalizing decision-making of interacting humans. Past research has focused on game-theoretic approaches (Myerson, 1997) and their graphical representations (Koller and Milch, 2003; Lee and Wolpert, 2012) where agents make a single decision given a particular depth of reasoning between opponents. Such approaches have been successfully used for predicting human decision-making in competitive tasks (Camerer, 2003; Wright and Leyton-Brown, 2010, 2012).

In cases when it is important to capture how people reason across time, sequential models of human interaction can be used (Littman, 1994). Such models can reflect either cooperative settings where agents are attempting to maximize mutual utility (Amato et al., 2009) or competitive contexts where agents are attempting to maximize their own reward (Doshi and Gmytrasiewicz, 2005, 2006). Graphical forms of sequential decision processes have also been developed (Zeng and Xiang, 2010).

Since humans often reason over a relatively short time horizon (Sellitto et al., 2010), especially in sudden, stressful scenarios, we model our interacting pilots scenario as a one-shot game. Like Lee and Wolpert (2012), we use level- $k$  relaxed strategies as a model for human decision-making. A level-0 strategy chooses actions uniformly at random. A level- $k$  strategy assumes that the other players adopt a level- $(k - 1)$  strategy (Costa-Gomes et al., 2001). In our work, we assume  $k = 1$  because humans tend to use shallow depth-of-

reasoning (Wright and Leyton-Brown, 2010), and increasing  $k > 1$  had little effect on the predicted joint-decisions for our scenario.

The focus on bounded rationality stems from observing the limitations of human decision making. Humans are unable to evaluate the probability of all outcomes with sufficient precision and often make decisions based on adequacy rather than by finding the true optimum (Simon, 1956, 1982; Caplin et al., 2011). Because decision-makers lack the ability and resources to arrive at the optimal solution, they instead apply their reasoning only after having greatly simplified the choices available. This type of sampling-based, bounded-rational view of perceptual processing has been used in computational models of human visual tracking performance (Vul et al., 2010) and has been shown to predict human decision-making (Vul et al., 2009).

To align our model with a bounded-rational view of human performance, we assume that Player 1 makes decisions based on  $m'$  sampled intruder locations and  $m$  candidate actions sampled from a uniform distribution over  $\pm 1$  radian. In other words,

$$o^{(1)}, \dots, o^{(m')} \sim p(o_{\text{ow}}^1 | s^2)p(o_{\text{in}}^1 | s^2) \quad (1)$$

$$a^{(1)}, \dots, a^{(m)} \sim \mathcal{U}(-1, 1) \quad (2)$$

Player 1 selects the sampled action that maximizes the expected utility over the  $m'$  sampled states of the other aircraft:

$$a = \arg \max_i \sum_j w^1 d(s_{\text{fl}a^{(i)}}^1, s_{\text{fl}o^{(j)}}^2) - (1 - w^1)|a^{(i)}| \quad (3)$$

In the equation above,

- $s_{\text{fl}a^{(i)}}^1$  is the final state of Player 1 after performing action  $a^{(i)}$ ,
- $s_{\text{fl}o^{(j)}}^2$  is the expected final state of Player 2 given observation  $o^{(j)}$  and assuming a random heading change,
- $d(\cdot, \cdot)$  represents Euclidean distance, and
- $w^1$  is the relative desire of Player 1 to avoid the intruder versus maintaining their current heading.

The first term in the utility function rewards distance between aircraft, and the second term penalizes the magnitude of the heading change. Fig. 2 shows how joint-actions change across different  $w^1$  and  $w^2$  combinations. When weights are relatively low (e.g.,  $w^1 = 0.80$ ,  $w^2 = 0.80$ ), joint-actions are centered around  $(0, 0)$ , indicating pilots tend to make very small heading change maneuvers. Conversely, when weights are relatively high (e.g.,  $w^1 = 0.98$ ,  $w^2 = 0.98$ ), pilots tend to make large heading change maneuvers.

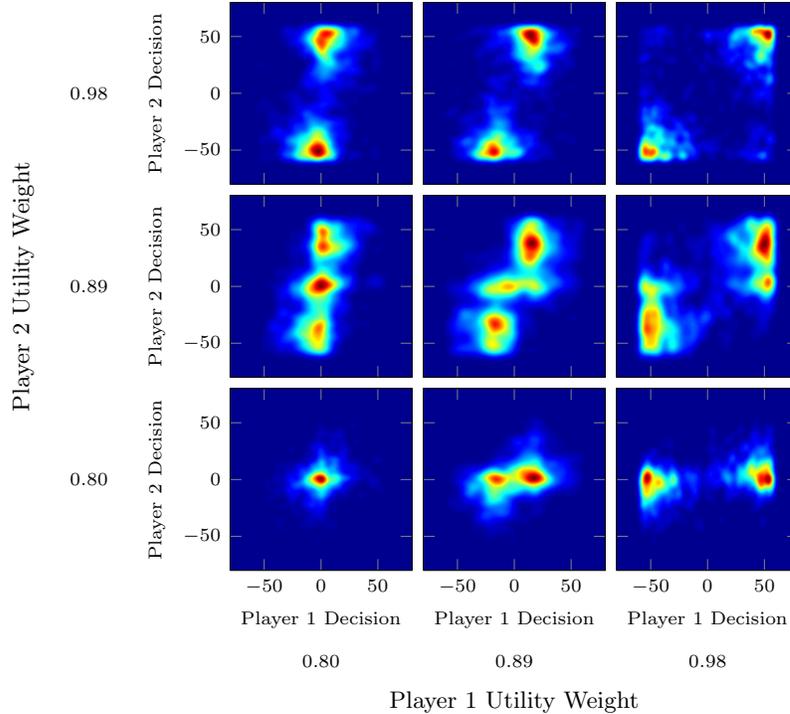


Figure 2: Influence of utility weights on joint-decision densities. Red regions depict areas with high probability joint-decisions (i.e., heading change, in degrees), while blue regions represent low probability joint-decisions.

### 2.3 Executing Actions

After choosing an appropriate action, the pilots then execute the maneuver. Although it is known that humans have uncertainty associated with their movements (Harris and Wolpert, 1998), we simulate the case in which pilots perfectly execute the action selected by level- $k$  relaxed strategies, since uncertainty in action execution is not a focus of this study.

Recall that the action of the player is the change in heading angle of the aircraft, away from the current heading. From the initial state and the commanded heading change of the aircraft, the final state is obtained by simulating the aircraft for 5 seconds assuming point-mass kinematics, no acceleration, and instantaneous heading change.

## 3 Multi-fidelity Prediction

There are two ways one could distinguish a low-fidelity and high-fidelity model of this encounter. The first is using “low-fidelity humans” versus “high-fidelity humans,” where low-fidelity humans would represent people who have been trained in flight simulators but who are not commercial pilots. The second way is to have both a low-fidelity and high-fidelity simulation environment, say having low-fidelity environment being a flight simulator that does not perfectly match

the conditions of an actual commercial jet. For this paper, our low-fidelity simulators are the same as the high-fidelity simulation except there is no instrument panel. The low-fidelity simulation has the  $o_{in}$  nodes removed in Fig. 1b.

We used separate training and testing encounters, all sampled with Player 2 being in Player 1’s field of view, and a heading such that a collision occurs if an avoidance maneuver is not performed (dashed lines in Fig. 1a). Training encounters were initialized with Player 2 randomly approaching at  $-45^\circ$ ,  $0^\circ$ , and  $+45^\circ$ . Test encounters were initialized with Player 2 randomly approaching at  $-22.5^\circ$  and  $+22.5^\circ$ . Initial headings for both train and test encounters were sampled from a Gaussian distribution with a mean around a heading direction that would result in a collision with a standard deviation of  $5^\circ$ .

The goal of multi-fidelity prediction is to combine the data in the high and low-fidelity training encounters to predict the joint-decisions of pilots in the high-fidelity game. The next two sections describe different approaches to achieving this type of prediction, and the notation is summarized in Table 1.

Table 1: Notation used for multi-fidelity prediction

	Description
$A$	Observed joint-actions ( $a^1, a^2$ )
$\hat{A}$	Predicted joint-actions ( $\hat{a}^1, \hat{a}^2$ )
$S$	Encounter geometry ( $s^1, s^2$ )
$N$	Number of encounters
$w$	Joint utility-weights ( $w^1, w^2$ )
$z$	Regression weight
$(.)^{tr}$	Superscript indicating training encounters
$(.)^{te}$	Superscript indicating test encounters
$(.)_l$	Subscript indicating low-fidelity game
$(.)_h$	Subscript indicating high-fidelity game

## 4 Model-Free Prediction Approaches

A model-free approach is one where we do not use any knowledge of the underlying game or the decision-making process. This section discusses a traditional model-free approach to predicting joint-actions as well as a model-free multi-fidelity method.

### 4.1 Locally Weighted, High-fidelity

The simplest approach to model-free prediction is to use the state and joint-action information from the high-fidelity training data to predict the joint-actions given the states in the testing data. Using only the high-fidelity training data, we trained a high-fidelity predictor  $R_h$  that predicts joint-actions  $A_h^{tr}$  given the training encounters  $S_h^{tr}$ . Then, the test encounters were used as inputs to the predictor, to predict the joint-actions at the new states  $\hat{A}^{te} = R_h(S^{te})$ . The regression model we considered was locally weighted (LW) regression, where the predicted high-fidelity joint-decisions for a particular test encounter is the weighted combination of high-fidelity joint-decisions from the training encounters. The weights are determined by the distance between the training encounters and the test encounter:

$$\hat{A}_i^{te} = \sum_j^{N_h^{tr}} z_{i,j} A_{h,j}^{tr} \quad (4)$$

$$z_{i,j} = \frac{e^{-d_{i,j}}}{\sum_j e^{-d_{i,j}}} \quad (5)$$

where  $d_{i,j}$  is the standardized Euclidean distance between the state in the  $i$ th testing encounter and the  $j$ th training encounter. This approach does not use any of the low-fidelity data to make predictions, so the quality of the predictions is strictly a function of the amount of high-fidelity training data. This method represents how well one might do without taking ad-

vantage of multi-fidelity data, and serves as a baseline for comparison against the multi-fidelity approach.

### 4.2 Locally Weighted, Multi-fidelity

Our model-free multi-fidelity method uses an approach similar to the high-fidelity method, but it also takes advantage of the low-fidelity data. We use LW regression on the low-fidelity training data to obtain a low-fidelity predictor  $R_l$  that predicts joint-actions  $\hat{A}_l^{tr}$  for the training encounters  $S_l^{tr}$ .

We use the the low-fidelity predictor to augment the high-fidelity training data. The augmented input  $(S_h^{tr}, R_l(S_h^{tr}))$  is used to train our augmented high-fidelity predictor  $R_{h'}$  that predicts joint-actions based on the high-fidelity data. The test encounters  $S^{te}$  were used as inputs to the augmented high-fidelity predictor to predict the joint-action for the test encounters as  $\hat{A}^{te} = R_{h'}(S^{te}, R_l(S^{te}))$ . In essence, we are using the predictions from the low-fidelity predictor as features for training the high-fidelity predictor.

## 5 Model-Based Prediction Approaches

The model-free methods described above ignore what is known about how the pilots make their decisions. In many cases, we have a model of the process with some number of free parameters. In our case, we model the players interacting in a level- $k$  environment, and treat the utility weights  $w_h$  as unknown parameters that can be learned from data, allowing the prediction of behavior in new situations. Such an approach is used in the inverse reinforcement learning literature (Baker et al., 2009; Ng and Russell, 2000), where the goal is to learn parameters of the utility function of a single agent from experimental data. This section outlines three model-based approaches where the data comes from experiments of varying fidelities. The first two are based on *maximum a posteriori* (MAP) parameter estimates, and the third is a Bayesian approach.

### 5.1 MAP High-fidelity

To demonstrate the benefit of multi-fidelity model-based approaches, we use a baseline method that exclusively relies on high-fidelity data to make predictions. Since we have a model of interacting humans, we can simply use the test encounters  $S^{te}$  as inputs into the model and use the resulting joint-actions as predictions. However, to make accurate predictions, we must estimate the utility weights used by the pilots. To find the MAP utility weights  $w_h^*$ , we first need to estimate  $p(A | S, w)$ , which is the probability of the joint-actions  $A$  given encounters  $S$  and weight  $w$ .

As an approximation, we estimate  $p(A | S, w)$  by simulating the high-fidelity game using a set of 1000 novel encounters ( $S^n$ ), under different utility weight combinations. We stored the resulting joint-actions for the  $j$ th utility weight combination, and then estimated  $p(A_h^n | w_h^j) = \sum_S p(A_h^n | S^n, w_h^j)$  using kernel density estimation via diffusion (Botev et al., 2010). Fig. 2 shows how  $p(A_h^n | w_h)$  changes across a subset of  $w_h$  settings.

We can find the MAP utility weight combination by using the joint-actions  $A_h^{tr}$  we observe for our training encounters  $S_h^{tr}$ . For each joint-action from the training data, we find the nearest neighbor joint-action in  $A_h^n$ . We sum the log-likelihood of the nearest-neighbor joint-actions for the training encounters, under each utility weight combination. The MAP utility weight combination is defined by

$$w_h^* = \arg \max_{w_h} [\ln p(w_h) + \sum_n \ln p(A_h^n | w_h)] \quad (6)$$

where  $p(w_h)$  is the prior of weight vector  $w_h$ . In our experiments, we assume  $p(w_h)$  is uniform over utility weight combinations. Once we have estimated the MAP utility weights, we can simulate our high-fidelity game using the test encounters and the MAP utility weights to obtain the predicted joint-action. The predicted joint-action is obtained by generating  $N_s = 10$  samples from  $p(A_h^{te} | S_i^{te}, w_h^*)$  and averaging them together:

$$A^{(1)}, \dots, A^{(N_s)} \sim p(A_h^{te} | S_i^{te}, w_h^*) \quad (7)$$

$$\hat{A}_{h,i}^{te} = \frac{1}{N_s} \sum_{\ell=1}^{N_s} A^{(\ell)} \quad (8)$$

The amount of high-fidelity data impacts the quality of the utility-weight estimate ( $w_h^*$ ), thereby limiting its effectiveness in situations when there is little data. The method described next overcomes this limitation by fusing both high- and low-fidelity data to find the utility weights to use for prediction.

## 5.2 MAP Multi-fidelity

In situations when there is little high-fidelity data to estimate the utility weights, it is beneficial to fuse both low- and high-fidelity data to increase the reliability of the estimate. In this approach, the predicted high-fidelity joint-actions for test encounters ( $\hat{A}^{te}$ ) are being computed according to Eq. (8), and our MAP utility weight is still found by maximizing Eq. (6). However, instead of only using the high-fidelity data to estimate the utility weights, we use both the low-fidelity and the high-fidelity data to estimate the weights. Specifically,

we find the log-likelihood of the nearest-neighbor joint-action  $A^n$  for both the low- and high-fidelity joint-actions  $A_h^{tr}$  and  $A_l^{tr}$ . Once we have estimates for the utility weights in the decision model, we can use the test encounters  $S^{te}$  as inputs to the high-fidelity model to obtain predictions  $\hat{A}^{te}$ , similar to what was done in Eq. (8).

Although this is a straightforward way to use both low- and high-fidelity training data, it assumes that the decision makers in the low-fidelity and high-fidelity encounters are identical. While it is plausible that the decision makers have similar utility functions to make this a valid assumption, it could also be the case that highly trained humans have very different utility functions from lesser-trained ones, or that changes in the simulation environment also cause changes in the utility function (for example, due to the decreased sense of immersion). Therefore, we would like a method that relaxes the equal utility assumption, allowing us to make predictions for games that are different for low- and high-fidelity.

## 5.3 Bayesian Multi-fidelity

The approaches described above use the test encounters as inputs into the high-fidelity model and sample joint-actions from  $p(A_h^{te} | S_h^{te}, w_h^*)$  using the MAP utility weight estimate  $w_h^*$  to compute the predicted joint-actions  $\hat{A}_h^{te}$ . In contrast, a Bayesian approach to multi-fidelity makes its predictions by sampling joint-actions from  $p(A_h^{te} | S^{te}, w_h^j)$  under *each* of the  $j$  high-fidelity utility weight combinations, and then weighting the predicted joint-actions by the probability of the utility weights, given the low- and high-fidelity joint-actions observed from the training encounters:

$$\hat{A}_{h,i}^{te} = \sum_j p(A_{h,k}^{te} | S_i^{te}, w_h^j) p(w_h^j | A_l^{tr}, A_h^{tr}) \quad (9)$$

where

$$p(w_h^j | A_l^{tr}, A_h^{tr}) = p(w_h^j | A_h^{tr}) \sum_k p(w_l^k | A_l^{tr}) p(w_l^k, w_h^j)$$

The probability distribution  $p(w_h^j | A_h^{tr})$  was estimated using the method outlined in Section 5.1. The low-fidelity version of this distribution  $p(w_l^k | A_l^{tr})$  was estimated in a similar manner for each of the  $k$  low-fidelity utility weight combinations by simulating the low-fidelity game using the same novel encounters  $S^n$  and kernel density estimation via diffusion methods.

Finally,  $p(w_l, w_h) = p(w_l^1, w_l^2, w_h^1, w_h^2)$  is a prior probability over low- and high-fidelity utility weights, and can be selected based on the degree of similarity between the high-fidelity and low-fidelity simulations. For our case, we assume a Gaussian prior with a mean

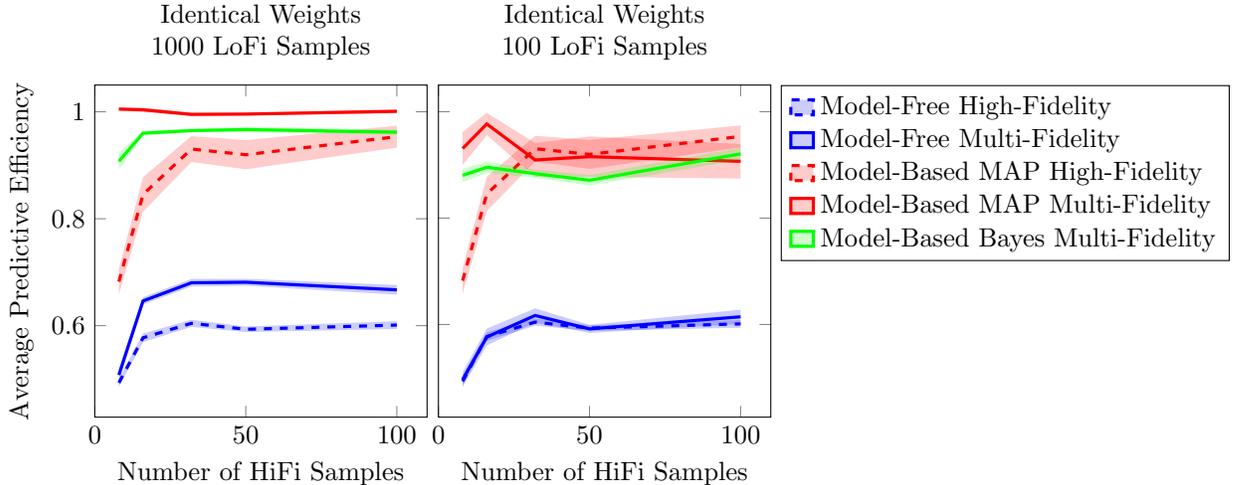


Figure 3: Predictive performance curves for various methods across different numbers of high-fidelity training samples. Lines depict mean predictive efficiency across the 10 simulations per sample-size condition, and the shaded-areas represent standard error. Results are shown for conditions in which the low- and high-fidelity utility weights are identical.

corresponding to the ground-truth utility weights (Section 6) and covariance given by:

$$\begin{bmatrix} .0017 & 0 & .0013 & 0 \\ 0 & .0017 & 0 & .0013 \\ .0013 & 0 & .0017 & 0 \\ 0 & .0013 & 0 & .0017 \end{bmatrix}$$

This covariance assumes that there is variation among players at the same fidelity, but also that the weights of the players are similar (but not identical) across fidelities. However, it assumes that the utilities of the two players within any specific encounter are independent.

A Bayesian approach to multi-fidelity allows for predictions to be made that account for the uncertainty in how well the utility weights account for the joint-actions observed in training. Such an approach also relaxes the assumption of identical utility weights across games through a prior distribution that encodes the coupling of these parameters.

## 6 Results

In order to assess the performance of both model-free and model-based approaches to multi-fidelity prediction, we evaluated each method with different amounts of high-fidelity training data. We assessed the benefit of incorporating an additional 100 or 1000 low-fidelity training examples, depending on the condition.

Since novices may employ different strategies than domain experts, we also simulated the scenario under different ground-truth utility weight relationships be-

tween the low- and high-fidelity games. For the scenario where experts and novices employ similar behavior, the ground-truth utility weights were set to:

$$w_l^{gt} = \{w^1 = 0.89, w^2 = 0.90\} \quad (10)$$

$$w_h^{gt} = \{w^1 = 0.89, w^2 = 0.90\} \quad (11)$$

For the scenario where there is a small difference in how novices act, the ground-truth low-fidelity utility weights were changed to:

$$w_l^{gt} = \{w_l^1 = 0.88, w_l^2 = 0.89\} \quad (12)$$

For the scenario where novices perform much differently than experts, the low-fidelity ground-truth utility weights were:

$$w_l^{gt} = \{w_l^1 = 0.80, w_l^2 = 0.81\} \quad (13)$$

Performance is measured in terms of “predictive efficiency,” which is a number typically between 0 and 1. It is possible to obtain a number above 1 because the normalization factor is an estimated lower-bound of the test-set error. The test-set error is given by

$$D = \sum_i d(\hat{A}_i, A_i^{te}), \quad (14)$$

where  $d(\cdot, \cdot)$  is the Euclidean distance,  $\hat{A}_i$  is the predicted action for the  $i$ th encounter, and  $A_i^{te}$  is the joint-action of the  $i$ th test encounter. A lower-bound on the test-set error  $D^{lb}$  can be approximated by estimating the error when using the ground-truth model in conjunction with Eq. (8). The predictive efficiency is given by  $D^{lb}/D$ .

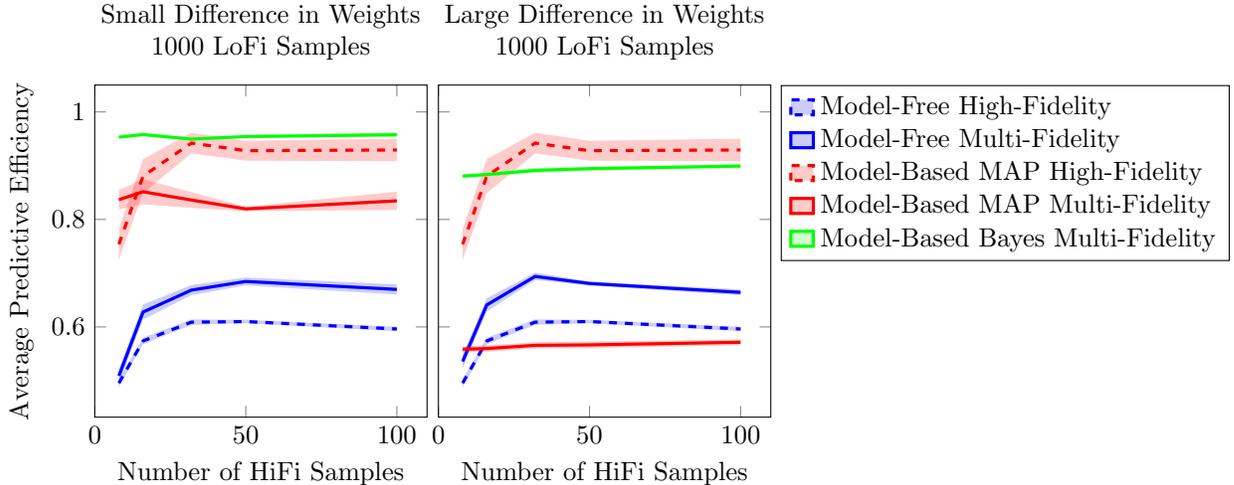


Figure 4: Predictive performance curves for various methods across different numbers of high-fidelity training samples. Lines depict mean predictive efficiency across the 10 simulations per sample-size condition, and the shaded-areas represent standard error. Results are shown for conditions in which the low- and high-fidelity utility weights have small (left) and large (right) differences between them.

Figure 3 shows how average predictive efficiency changes as a function of the amount of low- and high-fidelity training data for the identical utility weight condition. Figure 4 shows how average predictive efficiency is impacted by the level of discrepancy between low- and high-fidelity utility weights.

Multi-fidelity model-free approaches (blue solid lines) predicted better than methods that used only high-fidelity data (blue dashed lines), except for the case in which there was little (100 samples) low-fidelity available to train the model (Figure 3).

Model-based multi-fidelity approaches had better predictive performance (red and green solid lines) than high-fidelity methods (red dashed line) in cases when there was a large amount of low-fidelity data (1000 samples), and there was little or no difference in the utility weights used by the low-fidelity and high-fidelity humans in the task (Figures 3 and 4).

## 7 Conclusions and Further Work

We developed a multi-fidelity method for predicting the decisions of interacting humans. We investigated the conditions under which these methods produce better performance than exclusively relying on high-fidelity data. In general, our results suggest that multi-fidelity methods provide benefit if there is a sufficient amount of low-fidelity training data available and the differences between expert and novice behavior is not too large. Future effort will investigate these distinctions in greater detail.

This approach can also be extended to many domains beyond aviation. For example, work is currently being conducted that uses a ‘power-grid game’ to produce low-fidelity data to train algorithms that detect attacks on the system. This data could be combined with relatively sparse high-fidelity data (e.g., historical data involving known attacks) to increase the predictive performance of the model.

## Acknowledgments

The Lincoln Laboratory portion of this work was sponsored by the Federal Aviation Administration under Air Force Contract #FA8721-05-C-0002. The NASA portion of this work was also sponsored by the Federal Aviation Administration. Opinions, interpretations, conclusions and recommendations are those of the author and are not necessarily endorsed by the United States Government. We would like to thank James Chryssanthacopoulos of MIT Lincoln Laboratory for his comments on early versions of this work.

## References

- Amato, C., Bernstein, D. S., and Zilberstein, S. (2009). Optimizing fixed-size stochastic controllers for POMDPs and Decentralized POMDPs. *Autonomous Agents and Multi-Agent Systems*, 21:293–320.
- Aponso, B., Beard, S., and Schroeder, J. (2009). The NASA Ames vertical motion simulator—a facility engineered for realism. In *Royal Aeronautical Society Spring Flight Simulation Conference*.

- Baker, C., Saxe, R., and Tenenbaum, J. B. (2009). Action understanding as inverse planning. *Cognition*, 113:329–349.
- Botev, Z., Grotowski, J., and Kroese, D. (2010). Kernel density estimation via diffusion. *The Annals of Statistics*, 38:2916–2957.
- Camerer, C. (2003). *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press.
- Caplin, A., Dean, M., and Martin, D. (2011). Search and satisficing. *American Economic Review*, 101(7):2899–2922.
- Costa-Gomes, M., Crawford, V., and Broseta, B. (2001). Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235.
- Doshi, P. and Gmytrasiewicz, P. (2005). A particle filtering based approach to approximating interactive POMDPs. In *AAAI Conference on Artificial Intelligence*.
- Doshi, P. and Gmytrasiewicz, P. (2006). On the difficulty of achieving equilibria in interactive POMDPs. In *AAAI Conference on Artificial Intelligence*.
- Graf, E. W., Warren, P. A., and Maloney, L. T. (2005). Explicit estimation of visual uncertainty in human motion processing. *Vision Research*, 45:3050–3059.
- Harris, C. and Wolpert, D. (1998). Signal-dependent noise determines motor planning. *Nature*, 394:780–784.
- Kittur, A., Chi, E., and Suh, B. (2008). Crowdsourcing user studies with Mechanical Turk. In *ACM Conference on Human Factors in Computing Systems*.
- Koller, D. and Milch, B. (2003). Multi-agent influence diagrams for representing and solving games. *Games and Economic Behavior*, 45:181–221.
- Lee, R. and Wolpert, D. H. (2012). Game theoretic modeling of pilot behavior during mid-air encounters. In *Decision Making with Imperfect Decision Makers*. Springer.
- Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *International Conference on Machine Learning*.
- Myerson, R. B. (1997). *Game Theory: Analysis of Conflict*. Harvard University Press.
- Ng, A. and Russell, S. J. (2000). Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*.
- Paolacci, G., Chandler, J., and Ipeirotis, P. G. (2010). Running experiments on Amazon Mechanical Turk. *Judgement and Decision Making*, 5:411–419.
- Robinson, T., Willcox, K., Eldred, M., and Haines, R. (2006). Multifidelity optimization for variable-complexity design. In *AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*.
- Schlicht, E. J. and Schrater, P. R. (2007). Effects of visual uncertainty on grasping movements. *Experimental Brain Research*, 182:47–57.
- Sellitto, M., Ciaramelli, E., and di Pellegrino, G. (2010). Myopic discounting of future rewards after medial orbitofrontal damage in humans. *Journal of Neuroscience*, 30:16429–16436.
- Simon, H. (1956). Rational choice and the structure of the environment. *Psychological Review*, 63:129–138.
- Simon, H. (1982). *Models of bounded rationality*. MIT Press.
- Vul, E., Frank, M., Alverez, G., and Tenenbaum, J. B. (2010). Explaining human multiple object tracking as resource-constrained approximate inference in a dynamic probabilistic model. In *Advances in Neural Information Processing Systems*.
- Vul, E., Goodman, N., Griffiths, T. L., and Tenenbaum, J. B. (2009). One and done? Optimal decisions from very few samples. In *COGSCI Annual Meeting of the Cognitive Science Society*.
- Weis, Y., Simoncelli, E. P., and Adelson, E. H. (2002). Motion illusions as optimal percepts. *Nature Neuroscience*, 5:598–604.
- Wright, J. R. and Leyton-Brown, K. (2010). Beyond equilibrium: Predicting human behavior in normal-form games. In *AAAI Conference on Artificial Intelligence*, Atlanta, Georgia.
- Wright, J. R. and Leyton-Brown, K. (2012). Behavioral game-theoretic models: A bayesian framework for parameter analysis. In *International Conference on Autonomous Agents and Multiagent Systems*.
- Zeng, Y. and Xiang, Y. (2010). Time-critical decision making in interactive dynamic influence diagram. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*.