APPENDIX I
HUMAN'S BAYESIAN GOAL PREDICTION

The human assumes the robot is noisily rational in choosing its actions for an intended goal (or that the robot is exponentially more likely to choose an action if it has a higher Q-value):

$$p(u_R^t \mid x_R^t; \theta) = \frac{e^{\beta_H Q(x_R^t, u_R^t; \theta)}}{\int_{u_R'} e^{\beta_H Q(x_R^t, u_R'; \theta)}}, \tag{22}$$

where β_H is the rationality coefficient (sometimes called the "inverse temperature" parameter or the "model confidence"). In general, the integral in the denominator can be challenging to compute, but the simulated human makes use of the robot's baseline LQR controller to define the reward function as the instantaneous LQR cost:

$$r_R(x_R, u_R; \theta) = -(x_R - \theta)^T Q (x_R - \theta) - u_R^T R u_R, \tag{23}$$

so the $Q-$function is then the negative optimal cost-to-go:

$$Q_R(x_R, u_R; \theta) = r(x_R, u_R; \theta) - (x' - \theta)^T P (x' - \theta) \tag{24}$$

where $x' = Ax_R + Bu_R$ and P is the solution to the discrete-time algebraic Ricatti equation (DARE):

$$P = A^T P A - A^T P B (R + B^T P B)^{-1} B^T P A + Q. \tag{25}$$

Following prior work [33], this allows us to compute an exact form of the denominator of (22):

$$\begin{aligned} &\int e^{\beta_R Q_R(x_R, u_R; \theta)} \\ &= e^{-\beta_R (x_R - \theta)^T P (x_R - \theta)} \sqrt{\frac{(2\pi)^m}{\det(2\beta_R R + 2\beta_R B^T P B)}}. \end{aligned} \tag{26}$$

The human then uses this likelihood function (and Bayes Rule) to update its belief given a new observation (x_R^t, u_R^t):

$$\begin{aligned} b_H^t(\theta_i) &= p(\theta_i \mid x_R^{0:t}, u_R^{0:t}) \\ &= \frac{p(u_R^t \mid x_R^t; \theta_i) p(\theta_i \mid x_R^{0:t-1}, u_R^{0:t-1})}{\sum_{\theta'} p(u_R^t \mid x_R^t; \theta') p(\theta' \mid x_R^{0:t-1}, u_R^{0:t-1})} \end{aligned} \tag{27}$$

APPENDIX II
LEARNING-BASED CBP DETAILS

The basic form of these models is that they take in a trajectory history for all agents to as well as the *future trajectory* of the robot and outputs a prediction about the non-ego agents (the human in our case). We structure our learning-based model similarly, it takes in the human and robot past trajectories $x_H^{t-k:t}, x_R^{t-k:t}$ as well as the robot's *future plan* $x_R^{t+1:t+T}$. The trajectories are fed into LSTM layers with attention, and finally all pieces are concatenated with the set of goals $\{\theta_1, \ldots, \theta_N\}$ and passed into linear layers to output the probability of the human reaching each goal $[P(\theta_1), \ldots, P(\theta_N)]$. The network is trained to output the empirical probability of the human reaching each potential goal conditioned on the robot's plan; it is trained with an MSE loss.

These papers have generally been in spaces where there are large pre-existing datasets to train on, like in social navigation and autonomous driving. We're instead focusing on human-robot collaborative tasks such as manufacturing and in-home robots, so such datasets are hard to come by. As a result, we create our own dataset for our simulated human-robot collaboration task. The dataset consists of 3.5 million data points, which corresponds to approximately 97 hours of data collected for this particular human-robot interaction. This kind of data would be impractical to collect on a real human-robot interaction, so we use this as a baseline in our simulations but not in our user studies. We use the same goal-selection method (14) for this approach as our proposed controller, which works since this method is also a conditional prediction method.

APPENDIX III
SAFE TRAJECTORY GENERATION

We propose a long-term safe controller to ensure the probability of safety over the time horizon H is always above the desired threshold $1-\epsilon$, as stated in the objective (3). The controller is summarized in Algorithm 2. The key procedures at each time step are: 1. estimate the probability of safety for the current state 2. find the closest state that is safe enough if the initial state is not 3. execute goal-pursing control plus a potential field safe control that drives the robot to safe regions. Note that when estimating the safety probability, we assume a greedy human that has no robot-avoidance control to cover the worst case.

Remark 3.1: Assuming the human intention prediction $p(\theta_H^{post})$ is correct, the proposed long-term safe control (Algorithm 2) is guaranteed to meet the design objective (3). Similar theoretical derivations can be found in [30], [39].

To achieve more efficient real-time implementation, we use Gaussian distribution to approximate the uncertainty in human's motion at each time step t. Specifically, the mean of the Gaussian is the expected nominal motion of the human x_H and the variance will be some estimated value σ^2. For trajectory generation, we are interested in the long-term safety of the human-robot interaction in the sense that we want the generated trajectories to be safe at all time within a horizon H. Given the initial level of uncertainty $\sigma^{(i)}$ for each human mode i, we know that the uncertainty at time step t is $\sigma^{t(i)} = \sqrt{t}\sigma^{(i)}$. For each possible goal $\theta_H^{(i)}$ of human, let $x_H^{0:H(i)}$ be the nominal trajectory of the human starting from x_H^0 pursuing $\theta_H^{(i)}$. The probability of human choosing $\theta_H^{(i)}$ is given by $\theta_H^{(i)post}$. The goal is to generate $x_R^{0:H}$ such that the following safety condition is satisfied

$$\phi_{d_{\min} + \sigma^{t(i)}}(x^t) \geq 0, \forall t < H, \ \forall i. \tag{28}$$

For robot trajectory generation, we introduce the potential field controller to generate candidate trajectories that are more likely to be safe. The idea of potential field controller is that it will impose repelling forces to the controlled agent if it is close to the unsafe region. For example, if we want the robot's position x_R to be away from the human's position

	Naive	Reactive	Proactive	F-score	p-value
"I felt like the robot accounted for the [goal] I wanted to pick when it was picking a [goal]."	2.1	**4.0**	3.7	$F(2,40) = 12.4$	$p < 0.01$
"I often changed which [goal] I picked initially because of the robot."	3.7	**1.7**	2.7	$F(2,40) = 18.5$	$p < 0.01$
"I felt like the robot and I scored as many points as we could."	3.6	**4.1**	3.0	$F(2,40) = 4.0$	$p = 0.02$
"The robot influenced me to pick good [goals] for the team."	3.0	2.6	3.2	$F(2,40) = 1.8$	$p = 0.18$
"The robot's choice of [goals] made me choose worse [goals] for the team."	3.3	**1.8**	2.5	$F(2,40) = 10.0$	$p < 0.01$
"The robot was easy to collaborate with."	3.3	4.1	3.4	$F(2,40) = 2.9$	$p = 0.06$
"I felt like the robot picked the best [goals] to grab for the team."	3.2	**4.0**	2.7	$F(2,40) = 8.1$	$p < 0.01$
"I felt like the robot hindered the team's performance."	2.8	2.2	3.0	$F(2,40) = 2.6$	$p = 0.08$

TABLE V: Responses to subjective survey questions (5: Strongly Agree, 1: Strongly Disagree).

Algorithm 1 Safe Robot Control Pipeline with Model-Based CBP

Given: $\{G_1, ..., G_N\}$ ▷ goal locations
Given: x_H^0, x_R^0 ▷ agent starting positions
$b_R^0(\theta_H^{prior}) \leftarrow$ uniform $\hat{b}_H^0(\theta_R) \leftarrow$ uniform ▷ initial robot belief and mental model
while $t < H$ **do** ▷ H is the trajectory horizon
  $b_R^t(\theta_H^{prior}) \leftarrow (x_H^t, x_R^t, u_H^t)$, $\hat{b}_H^t(\theta_R) \leftarrow (x_R^t, u_R^t)$ ▷ robot updates nominal belief and mental model
  $b_R^t(\theta_H^{post} \mid \theta_R) \leftarrow \sum_{\theta_H^{prior}} p(\theta_H^{post} \mid \theta_H^{prior}, x_H^{0:t}, x_R^{0:t}, u_H^{0:t}, \theta_R) b_R^t(\theta_H^{prior})$ ▷ CBP belief update
  $p(\theta_H^{post}) = \hat{b}_H^t(\theta_R) b_R^t(\theta_H^{post} \mid \theta_R)$ ▷ robot computes overall goal probabilities
  $\mathcal{X}_R \leftarrow trajGen(x_H^t, x_R^t, p(\theta_H^{post}))$ ▷ generate candidate safe trajectories
  $\mathbf{x}_R^*, \theta_R^* \leftarrow \underset{(\mathbf{x}_R, \theta_R) \in \mathcal{X}_R}{\text{argmin}} \; J(\mathbf{x}_R, b_R^t(\theta_H^{post} \mid \theta_R), b_R^t(\theta_H^{prior}))$ ▷ choose (traj, goal) pair that minimizes cost
  $u_R^t \leftarrow \mathbf{x}_R^{u,0}$ ▷ choose first control action from trajectory
  $x_R^{t+1} \leftarrow f_R(x_R^t, u_R^t)$
end while

x_H, we will have the potential field control for the robot to be

$$u_{\text{pf}} = \frac{\gamma}{d^2}(C_H x_H - C_R x_R) = K_{\text{pf}}(x_H - x_R), \tag{29}$$

where d is the distance between human and robot and γ is the repel force of the potential field controller. For simplicity we assume $C_H = C_R$ and use K_{pf} to denote the coefficient. Similarly, we can define o to be some obstacles to be avoided and use (29) to find the safe control by replacing x_H with o.

The overall safe trajectory generation algorithm in shown in Algorithm 3, where we use potential field controller to avoid different possible human motions given different goals, and use synthetic obstacles in the state space to diverse the trajectory. Safety is identified at each time step t with uncertainty level $\sigma\sqrt{t}$, which characterizes the growing uncertainty bound over time. Experiments show that synthetic obstacles close to the initial position of the robot will give more diverse trajectories.

APPENDIX IV
SUBJECTIVE QUESTIONS FOR USER STUDY

We ran a one-way repeated measures ANOVA for the effect of the robot type on each survey question listed in Table V and ran post-hoc tests with Bonferroni tests. As noted in the main text, we found significant differences for the "Accounted" and "Changed" questions. We additionally found significant differences for the questions about scoring as many points as they could, influencing negatively, and picking the best goals for the team. Participants rated that they felt like the team scored as many points as possible more with the reactive robot than the proactive robot ($p < 0.01$), which is interesting given that participants statistically performed the *worst* with the reactive robot. Participants also rated that they choose worse goals with the naive robot than the reactive robot ($p < 0.01$) and thought both the CBP and reactive robots were not making them choose worse goals. They additionally felt like the reactive robot picked better goals for the team than the CBP robot ($p < 0.01$), which is again interesting because it does not align with the actual performance on the task. This seems to indicate that the peoples' subjective opinions of a collaborator may not be based precisely on performance, but on other social factors such as perceived intelligence or predictability. We also see that on these subjective questions, the reactive robot is most often the highest rated, but when forced to choose a preferred collaborator, they tended to choose the CBP model. This may be because they felt it was the most capable collaborator overall, even though they felt that the reactive robot was responding well to their strategies.

Algorithm 2 Long-term Safe Control

procedure SAFECONTROL($x_H^t, x_R^t, \theta_R, \theta_H, H$)
  $F \leftarrow$ long_term_safe_prob $(x_H^t, x_R^t, \theta_H, H)$
  if $F > 1 - \epsilon$ **then**
    $u_R^t \leftarrow K(x_R^t - \theta_R)$ ▷ goal pursuing control
  else
    $x_{\text{safe}} \leftarrow$ find_safe_state $(x_H^0, x_R^0, \theta_H, H, \epsilon)$
    $u_R^t \leftarrow K(x_R^t - \theta_R) - K_{\text{pf}}(x_R^t - x_{\text{safe}})$ ▷ potential field safe control
  end if
  return u_R^t ▷ return trajectories
end procedure
procedure LONGTERMSAFEPROB($x_H^0, x_R^0, \theta_R, \theta_H, H$)
  Given: n ▷ number of episode to estimate safety probability
  Initialize $\{F^{(1)}, F^{(2)}, \cdots, F^{(n)}\} = \mathbf{1}$ ▷ record safety of each trajectory
  for $k = 1 : n$ **do**
    $\theta_H^* \leftarrow$ sample(θ_H) from $p(\theta_H^{post})$
    for $t = 1 : H$ **do**
      $u_R^t \leftarrow K(x_R^t - \theta_R)$ ▷ goal pursuing control for robot
      $u_H^t \leftarrow K(x_H^t - \theta_H^*)$ ▷ goal pursuing control for human
      $x_R^{t+1} \leftarrow$ step(x_R^t, u_R^t)
      $x_H^{t+1} \leftarrow$ step(x_H^t, u_H^t) ▷ human dynamics with disturbance
      if $\|x_R^t - x_H^t\| < d_{\text{min}}$ **then**
        $F^{(k)} = 0$
        break
      end if
    end for
  end for
  return mean(F)
end procedure
procedure FINDSAFESTATE($x_H^0, x_R^0, \theta_R, \theta_H, H, \epsilon$)
  Given: n ▷ number of sampling states for a certain radius
  Initialize $r = 1$, $s = 0$ ▷ searching radius and indicator of safe state found
  while $s = 0$ **do**
    for $k = 1 : n$ **do**
      $x_R' \leftarrow$ uniform_sample (x_R, r, k) ▷ sample near robot's position with radius r
      $F \leftarrow$ long_term_safe_prob $(x_H^0, x_R', \theta_H, H)$
      if $F > 1 - \epsilon$ **then**
        $s = 1$ ▷ mark safe state found
        return $\{x_R', F\}$ ▷ return the safe state and its safety probability
      end if
    end for
  end while
end procedure

Algorithm 3 Candidate Trajectory Generation

procedure TRAJGEN($x_H^0, x_R^0, \theta_R, \theta_H^{prior}$)
  Given: $\{\sigma^{(1)}, \sigma^{(2)}, ..., \sigma^{(N)}\}$ ▷ uncertainties in each mode of the human
  $\{o_1, o_2, ..., o_M\} \leftarrow$ sample_obstacle(x_R) ▷ generate synthetic obstacles
  Initialize $\{s_1, s_2, ..., s_M\} = \mathbf{1}$ ▷ record safety of each trajectory
  for o_m in $\{o_1, o_2, ..., o_M\}$ **do** ▷ loop over all synthetic obstacles
    for $t = 1 : H$ **do**
      $u_R^t \leftarrow K(x_R^t - \theta_R)$ ▷ goal pursuing control
      $u_R^t \leftarrow u_R^t + K_{\text{pf}}(x_R^t - o_m)$ ▷ potential field control against obstacle
      for $\theta_H^{(i)}$ in $\{\theta_H^{(1)}, ..., \theta_H^{(N)}\}$ **do** ▷ loop over all human's goals
        if $\|x_R^t - x_H^{t(i)}\|_2 \leq d_{\min} + \sigma^{(i)}\sqrt{t}$ **then** ▷ check safety
          $s_i \leftarrow 0$
          break
        else
          $u_R^t \leftarrow u_R^t + \theta_H^{(i)prior} K_{\text{pf}}(x_R^t - x_H^{t(i)})$ ▷ potential field control against each human mode
          $u_H^{t(i)} \leftarrow K(x_H^{t(i)} - \theta_H^{(i)})$ ▷ goal pursuing control for human
          $x_H^{t+1(i)} \leftarrow \text{step}(x_H^{t(i)}, u_H^{t(i)})$
        end if
      end for
      $x_R^{t+1} \leftarrow \text{step}(x_R^t, u_R^t)$
    end for
    if $s_i = 1$ **then**
      $\{X_R, U_R\} \leftarrow [\{X_R, U_R\}, \{x_R, u_R\}]$ ▷ record safe trajectory
    end if
  end for
  return $\{X_R, U_R\}$ ▷ return trajectories
end procedure