Formalizing Assistive Teleoperation

Anca D. Dragan and Siddhartha S. Srinivasa

The Robotics Institute, Carnegie Mellon University

{adragan,siddh}@cs.cmu.edu

Abstract—In assistive teleoperation, the robot helps the user accomplish the desired task, making teleoperation easier and more seamless. Rather than simply executing the user's *input*, which is hindered by the inadequacies of the interface, the robot attempts to predict the user's *intent*, and assists in accomplishing it. In this work, we are interested in the scientific underpinnings of assistance: we formalize assistance under the general framework of *policy blending*, show how previous work methods instantiate this formalism, and provide a principled analysis of its main components: *prediction* of user intent and its *arbitration* with the user input. We define the prediction problem, with foundations in Inverse Reinforcement Learning, discuss simplifying assumptions that make it tractable, and test these on data from users teleoperating a robotic manipulator under various circumstances. We propose that arbitration should be moderated by the confidence in the prediction. Our user study analyzes the effect of the arbitration type, together with the prediction correctness and the task difficulty, on the performance of assistance and the preferences of users.

I. Introduction

We address the problem of teleoperating dexterous robotic manipulators to perform everyday manipulation tasks (Fig.1). In direct teleoperation, the user realizes their intent, for example grasping the bottle in Fig.1, by controlling the robot via an interface. Direct teleoperation is limited by the inadequacies and noise of the interface, making tasks, especially complex manipulation tasks, often tedious and sometimes impossible to achieve. In assistive teleoperation, the robot attempts to predict the user's intent, and augments their input, thus simplifying the task. Here, the robot faces two challenges when assisting: 1) predicting what the user wants, and 2) deciding how to use this prediction to assist.

We contribute a principled analysis of these two challenges. We introduce *policy blending*, which formalizes assistance as an arbitration of two policies: the user's input and the robot's prediction of the user's intent. At any instant, given the input, U, and the prediction, P, the robot combines them using a state-dependent arbitration function $\alpha \in [0, 1]$ (Fig.1(middle)). Policy blending with accurate prediction has a strong corrective effect on the user input (Fig.1,bottom). Of course, the burden is on the robot to *predict* accurately and *arbitrate* appropriately.

Prediction. Prior work in assistive teleoperation usually assumes that the robot knows the user's intent [1]–[9]. Other work assumes that the user is following one of a set of predefined paths or behaviors, and trains a classifier for prediction [10]–[14]. In many real-world scenarios, however, environments and goals change significantly, restricting the utility of fixed paths. For example, in the situation from Fig.1,
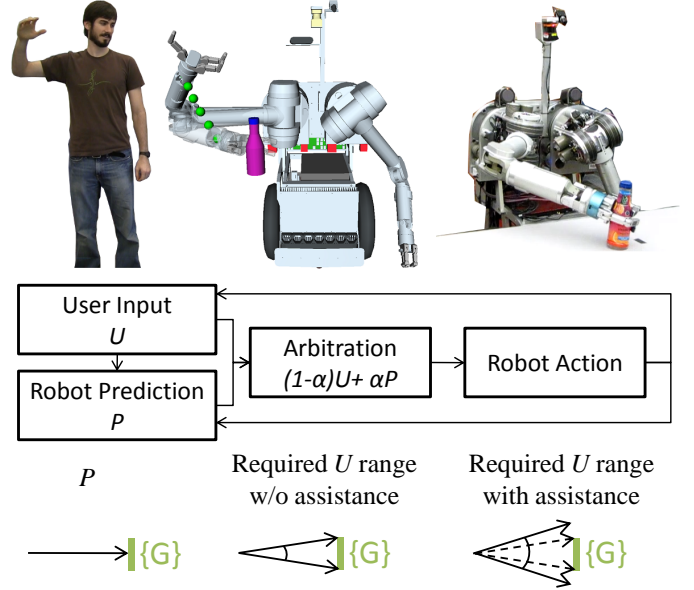

Fig. 1. (Top) The user provides an input U. The robot predicts their intent, and assists them in achieving the task. (Middle) Policy blending arbitrates user input and robot prediction of user intent. (Bottom) Policy blending increases the range of feasible user inputs (here, $\alpha = 0.5$).

the user must adapt to various locations of the goal object and its surrounding clutter.

In our work, we stray away from predefined paths, and instead formulate the prediction problem based on Inverse Reinforcement Learning [15]–[17]. We introduce several reductions that make it tractable, discuss how these reductions perform on real teleoperation data, and point out directions for improving prediction performance.

In this particular situation the user might be able to specify the intended goal (including the exact grasp) using other interfaces, like a GUI [18]. However, prediction via motion is often natural, faster, and seamless, enabling a user to, for example, easily change their mind and switch to another grasp or object, and can complement other interfaces.

Arbitration. Despite the diversity of methods proposed for assistance, from the robot completing the grasp when close to the goal [9], to virtual fixtures for following paths [14], to potential fields towards the goal [8], all methods can be seen as arbitrating user input and robot prediction. This common lens for assistance enables us to analyze the factors that affect its performance, and recommend design decisions for arbitration.

Prior work (detailed in Sec. II) compared more manual vs. more autonomous assistance modes [4]–[6] with surprisingly conflicting results in terms of what users prefer. Rather than

using autonomy as a factor, we introduce *aggressiveness*: arbitration should be moderated by the robot's confidence in the prediction, leading to a spectrum from very timid to very aggressive assistance, from small augmentation of user input even when confident to large augmentation even when unsure. Rather than analyzing the effect of aggressiveness (or autonomy) alone on the performance of assistance, we conduct a user study that analyzes how aggressiveness interacts with new factors, like *prediction correctness* and *task difficulty*, in order to help explain the seemingly contradictory findings from above.

Our formalism and analysis build on machine learning, control theory, and human-robot interaction to provide a new understanding of assistive teleoperation. The challenges we identify, particularly in predicting and expressing intent, do not only form the basis of assistive teleoperation, but lie at the foundation of human-robot collaboration in general.

II. PRIOR WORK

In 1963, Goertz [19] proposed manipulators for handling radioactive material that are able to turn cranks based on imprecise operator inputs, introducing one of the first instances of assistive teleoperation. Since then, research on this topic has proposed a great variety of methods for assistance, ranging from the robot having full control over all or some aspect of the motion [1]–[6], [10], [11], to taking control (or releasing it) at some trigger [9], [13], [20], to never fully taking control [4], [6]–[8], [14]. For example, Debus et al. [3] propose that the robot should be in full control of the orientation of a cylinder while the user is inserting it into a socket. In [9], the robot takes over to complete the grasp when close enough to the target. Crandal et al. [7] propose to mix the user input with a potential field in order to avoid obstacles.

Attempts to compare different modes of assistance are sometimes contradictory. For example, You and Hauser [4] found that for a complex motion planning problem in a simulated environment, users preferred a fully autonomous mode, where they only clicked on the desired goal, to more reactive modes of assistance. On the other hand, Kim et al. [5] found that users preferred a manual mode and not the autonomous one for manipulation tasks like object grasping.

Policy blending provides a unifying view of assistance, leading to an analysis which helps conciliate these differences. Table I shows how various methods proposed arbitrate user input and robot prediction (or simply robot policy, in cases where intent is assumed to be known). For example, potential field methods (e.g. [7], [8], [12]) that help the user avoid obstacles become blends of the user input with a policy obtained from the repulsive force field, under a constant arbitration function that establishes a trade-off. Virtual fixture-based methods (e.g. [6], [12]–[14]) that are commonly used to guide the user along a predefined path become blends of the user input with a policy that projects this input onto the path. The arbitration function dictates the intensity of the fixture at every step, corresponding to a normalized "stiffness/compliance" gain. However, the same framework also allows for the less studied case in which

TABLE I: PRIOR WORK.

Method	Prediction	Arbitration
[1]–[6], [18]	no	
[10], [11]	predefined paths/behaviors	
[4], [6]–[8]	no	
[12]	predefined paths/behaviors	
[9], [20]–[22]	no	
[13]	predefined paths/behaviors	
[12], [14]	predefined paths/behaviors	
[23]	fixed environment, goals (2D)	no
[24]	fully flexible (goal+policy) (2D)	no

the robot is able to generate a full policy for completing the task on its own, rather than an attractive/repulsive force or a constraint (e.g. [9], [20]). In this case, the arbitration is usually a binary switch from fully autonomous to fully manual behavior, although arbitrations with stages that trade off between the two (not fully taking control but still heavily correcting the user's input) are also possible. Instances of arbitration also appear outside the teleoperation domain, when mediating between two human input channels [25].

Analyzing assistance based on how arbitration is done, together new factors like prediction correctness and task difficulty, helps explain previously contradictory findings: our results show that aggressive assistance is preferable on hard tasks, like the ones from [4], where autonomy is significantly more efficient; opinions are split on easier tasks, like the ones from [5], where the autonomous and manual mode were comparable in terms of time to completion.

The same table shows how prior methods handle prediction of the user's intent. Aside from work that classifies which one of a *predefined* set of paths or behaviors the user is currently engaging [10], [11], most work assumes the robot has access to the user's intent, e.g. that it knows what object to grasp and how (except in [22], which deals with time delays in ball catching by projecting the input forward in time using a minimum-jerk model). Predicting or recognizing intent has received a lot of attention outside of the teleoperation domain, dating back to high-level plan recognition [26]. Predicting intended motion, however, is usually again limited to classifying behaviors, or is done in low-dimensional spaces [23], [24]. In the following section, which presents the building blocks of assistance, we present the general prediction problem, along with simplifying assumptions that make it tractable.

III. THE COMPONENTS OF ASSISTANCE

In what follows, Q denotes the robot's current configuration, U denotes the desired next configuration inputed by the user (e.g. through a GUI or a whole-body teleoperation interface, as in Fig.1, or by sending a velocity command to the robot[1]), and P denotes the configuration the robot predicts it should be at next. We denote the user's starting input as S, and the trajectory of user inputs until U as $\xi_{S \to U}$.

Each new scene has a (possibly continuous) set of accessible goals $\mathcal{G}$, known only at runtime to both robot and user. The robot does not know which goal the user is trying to reach.

A. Prediction

The robot must predict where the user would like it to move next, given $\xi_{S \to U}$, and any other cues, e.g. each goal's reachability, or a high-level description of the overall task.

We break this problem down into two successive steps:

1) **Goal Prediction** where we predict the most likely goal G^* given available data.
2) **Trajectory Prediction** where we predict how the user would want to move towards a predicted goal.

Goal Prediction. We formulate goal prediction as:

$$G^* = \arg \max_{G \in \mathcal{G}} P(G | \xi_{S \to U}, \theta) \qquad (1)$$

i.e. given $\xi_{S \to U}$ and any other available cues θ, the robot predicts the goal G^* that maximizes posterior probability.

Several simplifying assumptions help us to solve this problem. The strongest is *amnesic* prediction, which ignores all information except the current input U: $G^* = \arg \max_{G \in \mathcal{G}} P(G | U)$. There are many ways to estimate $P(G | U)$. For example, given a distance metric on goals d, we can assume that closer goals have higher probability:

$$G^* = \arg \min_{G \in \mathcal{G}} d(U, G) \qquad (2)$$

Under the Euclidean metric, $d(U, G) = ||U - G||$, the method predicts the goal closest in the robot's configuration space. Under $d(U, G) = ||\phi(U) - \phi(G)||$ (with ϕ denoting the forward kinematics function), the method predicts the goal closest in the robot's workspace.

Although intuitive, amnesic prediction does suffer from its amnesia. Where the user came from is often a critical cue for where they want to go. Prediction can also be *memory-based*, taking into account the trajectory $\xi_{S \to U}$ of user inputs (Fig.2):[2]

$$G^* = \arg \max_{G \in \mathcal{G}} P(G | \xi_{S \to U}) = \arg \max_{G \in \mathcal{G}} P(\xi_{S \to U} | G) P(G) \qquad (3)$$

In order to compute $P(\xi_{S \to U} | G)$, we need a model of how users teleoperate the robot to get to a goal. A possible assumption is that the user's input noisily optimizes a goal-dependent cost function C_G (one that depends, as an example,

Fig. 2. Even though G_1 is the closest goal to U in both cases, $\xi_{S \to U}$ indicates that G_2 is more likely in the situation on the right.

on the dot product between the user's velocity and the direction to the goal). Using the principle of maximum entropy, we can use C_G to induce a probability distribution over trajectories $\xi \in \Xi$ given a goal as $P(\xi | G) \propto \exp(-C_G(\xi))$, i.e. the probability of a trajectory decreases exponentially as its cost increases [16]. Given this distribution and if the cost is additive along the trajectory,

$$P(\xi_{S \to U} | G) = \frac{\exp(-C_G(\xi_{S \to U})) \int_{\xi_{U \to G}} \exp(-C_G(\xi_{U \to G}))}{\int_{\xi_{S \to G}} \exp(-C_G(\xi_{S \to G}))} \qquad (4)$$

In low-dimensional spaces, (4) can be evaluated exactly through soft-maximum value iteration [24]. In high-dimensional spaces, where this is expensive, an alternative is to approximate the integral over trajectories using Laplace's method. First, we approximate $C(\xi_{X \to Y})$ by its second order Taylor series expansion around $\xi_{X \to Y}^* = \arg \min_{\xi_{X \to Y}} C(\xi_{X \to Y})$:

$$C(\xi_{X \to Y}) \approx C(\xi_{X \to Y}^*) + \nabla C(\xi_{X \to Y}^*)^T (\xi_{X \to Y} - \xi_{X \to Y}^*) +$$
$$\frac{1}{2} (\xi_{X \to Y} - \xi_{X \to Y}^*)^T \nabla^2 C(\xi_{X \to Y}^*)(\xi_{X \to Y} - \xi_{X \to Y}^*) \quad (5)$$

Since $\nabla C(\xi_{X \to Y}^*) = 0$ at the optimum, we get

$$\int_{\xi_{X \to Y}} \exp(-C(\xi_{X \to Y})) \approx \exp(-C(\xi_{X \to Y}^*))$$
$$\int_{\xi_{X \to Y}} \exp(-\frac{1}{2}(\xi_{X \to Y} - \xi_{X \to Y}^*)^T H_{X \to Y}(\xi_{X \to Y} - \xi_{X \to Y}^*)) \quad (6)$$

with $H_{X \to Y}$ the Hessian of the cost function around $\xi_{X \to Y}^*$. Evaluating the Gaussian integral leads to

$$\int_{\xi_{X \to Y}} \exp(-C(\xi_{X \to Y})) \approx \exp(-C(\xi_{X \to Y}^*)) \frac{\sqrt{2\pi^k}}{\sqrt{|H_{X \to Y}|}} \quad (7)$$

and the optimal prediction G^* becomes

$$\arg \max_G \frac{\exp(-C_G(\xi_{S \to U}) - C_G(\xi_{U \to G}^*)) \sqrt{|H_{U \to G}|}}{\exp(-C_G(\xi_{S \to G}^*)) \sqrt{|H_{S \to G}|}} P(G) \quad (8)$$

If the cost function is quadratic, the Hessian is constant and (8) simplifies to

$$G^* = \arg \max_{G \in \mathcal{G}} \frac{\exp(-C_G(\xi_{S \to U}) - C_G(\xi_{U \to G}^*))}{\exp(-C_G(\xi_{S \to G}^*))} P(G) \quad (9)$$

---

[1]In the case of velocity inputs, the robot applies its motion model to obtain the configuration U.

[2]Although considering current velocity in amnesic prediction can help with this, this information is still local. As in Fig.3, incorporating global knowledge from the trajectory can be beneficial.
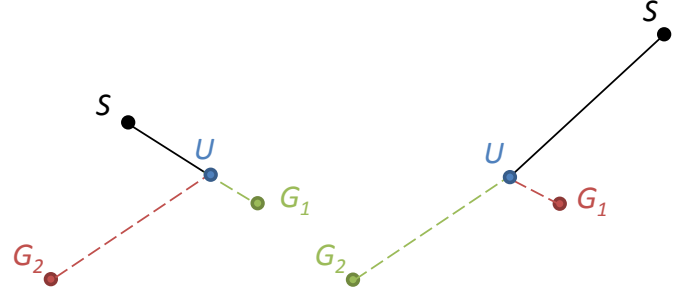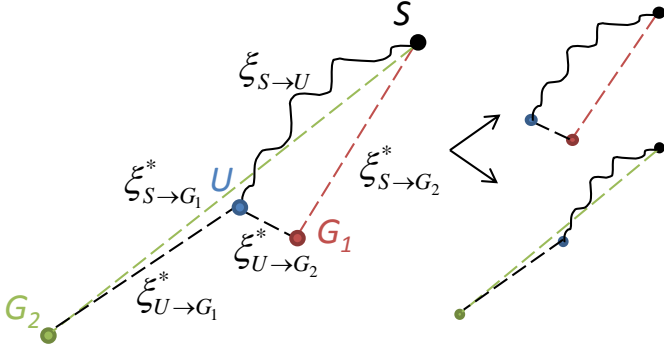
Fig. 3. An example user trajectory leading to two possible goals. A simple prediction implementing (9) with a cost function based on length compares the two situations on the right and determines that the user's trajectory is a smaller deviation from the optimal trajectory to G_2 than to G_1.

This prediction method implements an intuitive principle: if the user appears to be taking (even in the optimistic case) a trajectory that is a lot costlier than the optimal one to that goal, the goal is likely not the intended one.

In low-dimensional spaces, it is possible to learn C_G from user teleoperation data via Maximum Entropy Inverse Reinforcement Learning [16] (or via algorithms that assume no user noise, e.g. [15], [17]). In larger domains such as manipulation, simple guesses for C_G can still be useful for prediction. This is illustrated in Fig.3, which shows a toy problem with two possible goals and a user trajectory $\xi_{S \to U}$. Even with C_G as the sum of squared velocity magnitudes, this method still holds an advantage over the amnesic version. When C_G is the same for all goals, the cost of $\xi_{S \to U}$ is common across goals and does not affect the probability, leaving only the starting point S and the current input U as the crucial components of the trajectory so far. The comparison between G_1 and G_2 leads to the correct result, because the path to G_2 through U is much shorter relative to the optimal, $\xi^*_{S \to G_2}$, than the path to G_1 through U is relative to $\xi^*_{S \to G_1}$. Whether this very simplified memory-based prediction is still helpful in an analogous problem in the real world is one of the questions we answer in our user study.

A more sophisticated prediction method learns to choose G^* by also considering θ, and using the goal probabilities as features. From labeled data of the form $(F[\xi_{S \to U}, \theta, G] \to 0/1)$ — features computed relative to G, paired with whether G was the desired goal — a range of methods will learn to predict a goal "score" given feature values for a new situation. See [27] for goal prediction based on such feature constructs.
Trajectory Prediction. Once the robot predicts a goal G^*, it must also compute how the user wants it to move towards G from the current state Q. It can do so by computing a policy or a trajectory based on a cost function. Note that the cost function the robot must optimize in this stage is not necessarily the same as the cost function the human is optimizing during teleoperation: the idea behind assistive teleoperation is that the robot executes what the user actually wants rather than what the user commands. Approximations for this function

also range from very simple (e.g. the length of the trajectory, or a trade-off between the length and the distance from obstacles) to very complex (e.g. functions learned via Inverse Optimal Control from users physically moving the robot on a trajectory they would want the robot to take).

B. Arbitration

Given U and P, the robot must decide on what to do next. The arbitration function α, which makes this decision, can depend on a number of inputs, such as the distance to the goal or to the closest object, or even a binary switch operated by the user. We propose a simple principle: that arbitration must be moderated by how good the prediction is.
Timid vs. Aggressive. In trading off between not over-assisting (providing unwanted assistance) and not under-assisting (failing to provide needed assistance), the arbitration lies on a spectrum: On the one hand, the assistance could be very timid, with α taking small values even when the robot is confident in its prediction. On the other hand, it could be very aggressive: α could take large values even when the robot does not trust the predicted policy.
Inescapable Local Minima Do not Occur. In general, when arbitrating between two policies, we need to guarantee that inescapable local minima do not occur. In our case, these are states at which the arbitration results in the same state as at the previous time step, regardless of the user input.

Theorem 1: Let Q be the current robot configuration. Denote the prediction velocity as $p = P - Q$, and the user input velocity as $u = U - Q$. Arbitration never leads to inescapable local minima, unless $\forall u \neq 0$, $p = -ku$ for some $k \geq 0$, *and* $\alpha = \frac{1}{k+1}$ (i.e. the policy is always chosen to directly oppose the user's input, and the arbitration is computed adversarially, or $p = 0$ and $\alpha = 1$ for all user inputs).

Proof: Assume that at time t, a local minima occurs in the arbitration, i.e. $(1 - \alpha)(Q + u) + \alpha(Q + p) = Q$. Further assume that this minima is inescapable, i.e. $(1 - \alpha')(Q + u') + \alpha'(Q + p') = Q$, $\forall u'$, where p' and α' are the corresponding prediction and arbitration if u' is the next user input. $\Leftrightarrow (1 - \alpha')u' + \alpha'p' = 0$, $\forall u'$.
Case 1: $\forall u' \neq 0$, the corresponding $\alpha' \neq 0 \Rightarrow p' = -\frac{1-\alpha}{\alpha}u'$, $\forall u' \neq 0 \Rightarrow p' = -ku'$ and $\alpha = \frac{1}{k+1}$, with $k \geq 0$ (since $\alpha \in [0, 1]$) $\forall u' \neq 0$. Contradiction with the problem statement.
Case 2: $\exists u' \neq 0$ s.t. the corresponding $\alpha' = 0 \Rightarrow (1 - 0)u' + 0p' = 0 \Rightarrow u' = 0$. Contradiction with $u' \neq 0$.
$\Rightarrow \exists u'$ s.t. $(1 - \alpha')(Q + u') + \alpha'(Q + p') \neq Q$, ∎
Therefore, with an adversarial exception, the user can always take a next action that escapes a local minimum.
Evaluating Confidence. Earlier, we had proposed that the arbitration should take into account how good the prediction is, i.e. a measure of the confidence in the prediction, c, that correlates to prediction correctness. One way to evaluate c is to assume that the closer the predicted goal gets, the more likely it becomes that it is the correct goal: $c = \max(0, 1 - \frac{d}{D})$, with d the distance to the goal and D some threshold past which the confidence is 0. Alternately, confidence can be defined as the probability assigned to the prediction. If a cost function

is assumed, the match between the user's input and this cost should also factor in. If a classifier is used for prediction, then such a probability is obtained through calibration [28].

IV. A STUDY ON ASSISTANCE

Mathematically, arbitration can be any non-adversarial function of the robot's confidence in its prediction, from very timid to very aggressive. But assistive teleoperation is fundamentally a human-robot interaction task, and this interaction imposes additional requirements on arbitration: the robot must arbitrate in an efficient and user-preferred way. Therefore, we embarked upon a user study that analyzes the effect of the aggressiveness of arbitration on the performance of assistance – an analysis that we believe must incorporate other factors, like prediction correctness (users might not appreciate assistance if the robot is wrong) and task difficulty (users might appreciate assistance if the task is very hard for them). Although this analysis is our primary goal, we will also to test the performance of the simplifying assumptions from Sec. III-A on real data of users teleoperating the robot through or whole-body interface.

We tasked 8 users with teleoperating the robot to grasp an object from a table, as in Fig.1. There were always two graspable objects, and we gave the user, for every trial, the farther of the two as goal (an analogous situation to the one from Fig.3). We implemented a whole-body interface that tracks their skeleton (OpenNI, www.openni.org), yielding an arm configuration which serves as the user input U. The robot makes a prediction of the goal and the policy to it (that minimizes length in configuration-space), leading to P, and combines the two via the arbitration function α.

A. Goal 1: Factors that affect assistance

Hypotheses. We test the following two hypotheses:
1) Main effects: Prediction correctness, task difficulty, and aggressiveness of assistance each has a significant effect on task performance.
2) Interaction effects: Aggressive assistance performs better on hard tasks if the robot is right, while the timid assistance performs better on easy task if the robot is wrong.

Manipulated Variables. We manipulated prediction correctness by using a simple, easy to manipulate goal prediction method: the amnesic prediction based on workspace distance, which always selects the closest object. We setup wrong conditions at the limit of the robot being wrong yet rectifiable. We place the intended object further, guaranteeing wrong prediction until the user makes his preference clear by providing an input U closer to the correct goal. We setup right conditions by explicitly informing the robot of the user's intended goal.

We manipulated task difficulty by changing the location of the two objects and placing the target object in an easily reachable location (e.g. grasping the bottle in Fig.4(b) makes an easy task) vs. a location at the limit of the interface's reachability (e.g. grasping the box in Fig.4(b) is a hard task). This leads to four types of tasks: Easy&Right, Easy&Wrong, Hard&Right and Hard&Wrong.

Finally, we manipulated the aggressiveness of the assistance by changing the arbitration function, and used the distance-based measure of confidence from Sec. III-B. As the user makes progress towards the predicted object, the confidence increases. We had two assistance modes, shown in Fig.4(c): the timid mode increases the assistance with the confidence, but plateaus at a maximum value, never fully taking charge. On the other hand, the aggressive mode eagerly takes charge as soon as the confidence exceeds a threshold.

Subject Allocation. We chose a within-subjects design, enabling us to ask users to compare the timid and aggressive mode on each task. Each of our 8 participants (all students, 4 males and 4 females) executed both modes on each of the four types of tasks. To avoid ordering effects, we used a balanced Latin square for the task order, and balanced the order of the modes within each task.

Dependent Measures. We measure the performance of assistance in two ways: the amount of time each user took to complete the task under each condition, and each user's preference for the timid vs. the aggressive mode on each task type (on a 7 point Likert scale where the two ends are the two choices). We expect the two measures to be correlated: if an assistance mode is faster on a task, then the users will also prefer it for that task. We also asked the users additional questions for each condition, about how helpful the robot was, how much its motion matched the intended motion, and how highly they would rate the robot as a teammate.

Covariates. We identified the following confounds: the users' initial teleoperation skill, their rating of the robot without assistance, and the learning effect. To control for these, users went though a *training phase*, teleoperating the robot without assistance. This partially eliminated the learning effect and gave us a baseline for their timing and ratings. We used these as covariates, together with number of tasks completed at any point — a measure of prior practice.

B. Goal 2: Prediction based on real teleoperation data

Hypothesis. We test the following hypothesis:
3) On tasks in which the target object is not the closest one to the original human input configuration, replicating the situation from Fig.3, the memory-based prediction will identify the correct goal faster, yielding a higher success rate despite the simplifying assumptions it makes.

Manipulated Variables. We used the amnesic prediction during the study for its transparency, which enabled us to manipulate prediction correctness. We compared amnesic vs. memory-based prediction on the same data of the users teleoperating the robot under the different conditions, in a post-experimental stage. For memory-based prediction, we use workspace sum squared velocity as the cost C, leading to the simplification from (9).

Dependent Measures. We took each user trajectory and applied each of the two prediction methods at every point along. We measured the percent of time (success rate) across the trajectory the prediction identified the correct target object.

(a) Hard&Right   (b) Hard&Wrong   (c) Timid vs. Aggressive   (d) Time and Preference Comparison
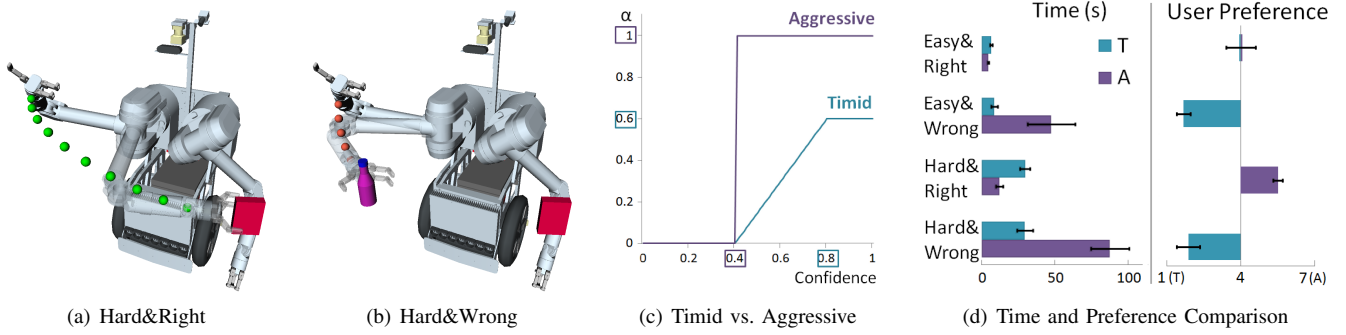
Fig. 4. From left to right: hard tasks, with the goal at the limit of the reachable area, for both right and wrong prediction, the arbitration functions, and the results of the study.

V. ANALYSIS AND DISCUSSION

Our first goal with this study was to identify the effect of different factors on the performance of assistance, and we do so in the following sections. Our secondary goal was to analyze two simplistic prediction methods (an amnesic and a memory-based one) on teleoperation data under different assistance modes. We discuss our findings in Sec. V-B.

A. Arbitration

Teleoperation Timing. The average time per task was approximately 28s. We performed a factorial repeated-measures ANOVA with Bonferroni corrections for multiple comparisons and a significance threshold of p = 0.05, which resulted in a good fit of the data ($R^2 = 0.66$). In line with our first hypothesis, we found main effects for all three factors: hard tasks took 22.9s longer than easy ones ($F(1, 53) = 18.45$, $p < .001$), tasks where the policy was wrong took 30.1s longer than when right ($F(1, 53) = 31.88$, $p < .001$), and the aggressive mode took overall 19.4s longer than the timid ($F(1, 53) = 13.2$, $p = .001$). We found a significant interaction effect between aggressiveness and correctness, showing that when wrong, being timid is significantly better than being aggressive. This is confirmed in Fig.4(d), which compares the means and standard errors on each task: the timid mode is better on both Easy&Wrong and Hard&Wrong. The timid mode performed about the same on Easy&Right, and, as expected, worse on Hard&Right (the time taken for aggressive is smaller than for timid for every user). Surprisingly, the interaction effect among all factors was only marginally significant ($F(1, 53) = 2.63$, $p = .11$). We believe that increasing our user pool would strengthen this effect.

To conclude based on this regression that the timid mode is overall better would be misleading, because it would assume that the robot is wrong in 50% of the tasks (in general, either by predicting he wrong goal, or by computing a motion that, for example, collides with an unseen obstacle). Our data indicates that the aggressive mode is overall more efficient if the robot is wrong in less than 16% of the cases. However, efficiency is only part of the story: as the next section points out, *some users are more negatively affected than others by a wrong robot policy.*

User Preferences. Fig.4(d) also shows the users' preferences on each task, which indeed correlated to the timing results (Pearson's $r(30) = .66$, $p < .001$). The outliers were users with stronger preferences than the time difference would indicate. For example, some users strongly preferred the timid mode on Hard&Wrong tasks, despite the time difference not being as high as with other users. The opposite happened on Hard&Right tasks, on which some users strongly preferred the aggressive mode despite a small time difference, commenting that they appreciated the precision of the autonomy. On Easy&Right tasks, the opinions were split and some users preferred the timid mode despite a slightly longer time, motivating that they felt more in control of the robot. Despite the other measures (helpfulness, ranking as a teammate, etc.) strongly correlating to the preference rating ($r(30) > .85$, $p < .001$), they provided similar interesting nuances. For example, the users that preferred the aggressive mode on Easy&Right tasks because they liked having control of the robot were willing to admit that the aggressive mode was more helpful. On the other hand, we also encountered users that preferred the aggressive mode, and even users that followed the robot's motion while aggressive, not realizing that they were not in control and finding the motion of the robot to match their own very well (i.e. the predicted policy P matched what they intended, resulting in seamless teleoperation).

Overall, although difference in timing is a good indicator of the preference, *it does not capture a user's experience in its entirety.* First, some users exaggerate the difference in preferences. Second, some users prefer the timid mode despite it being slightly less efficient. Third, *assistance shouldn't just be quick – it should also be intent-transparent.* Our users commented that "Assistance is good if you can tell that [the robot] is doing the right thing".

Detecting "Aggressive and Wrong". Being aggressive and wrong results in large penalties in time and user preference. Fortunately, it is also a state that can easily be identified and remedied by the user. Because prediction affects the user's behavior, when the robot eagerly starts heading towards the wrong target, the user rapidly attempts to get back control by moving against the robot's policy. This, in turn, decreases the predictor's confidence, causing the robot to following the user. This state can be detected early (Fig.6) by comparing the
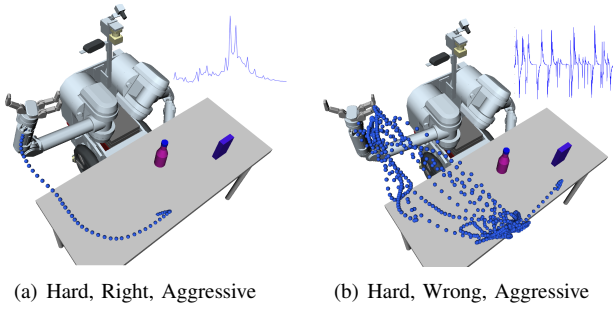
(a) Hard, Right, Aggressive   (b) Hard, Wrong, Aggressive

Fig. 6. A comparison of the user input trajectories for right vs. wrong. The graphs show the dot product between the robot's policy and the robot's actual velocity.

user's trajectory in the right and wrong case, along with the dot product between the robot's policy and its actual velocity.

B. Prediction

Results from the Study. A factorial ANOVA on the four manipulated factors obtained clean fit of the data, with $R^2 = 0.9$. Confirming our hypothesis, the factor corresponding to the prediction was significant ($F(1, 112) = 1020.95$, $p < 0.001$): memory-based prediction was significantly better at identifying the correct goal for a longer amount of time along the trajectory. The assistance being timid also made it significantly easier for the prediction method to output the correct answer ($F(1, 112) = 7.62$, $p < 0.01$): *users are not as precise about the motion in the aggressive mode, making prediction more difficult.* Fig.5(a) shows the means and standard errors for the memory-based prediction vs. the amnesic one, for each assistance type. Fig.5(b) and 5(c) compare the two methods on one of the trajectories, colored in each case according to whether the prediction was correct (green spheres) or incorrect (red spheres). While the amnesic variant only switches to the correct prediction at the very end, when the user input gets closer to the correct goal, the memory-based prediction is able to identify the goal much earlier.

In our study, the target object was always the farther one (to manipulate prediction correctness, i.e. create situations in which the robot is wrong until the user's input gets closer to the correct goal). However, this is disadvantageous to the amnesic method: if the target object were the closer one, this method would usually be 100% accurate. In fact, the amnesic success should be on average almost 0.5 higher. Nevertheless, even with this boost, memory-based remains the outperformer. Furthermore, prediction becomes more challenging as the number of goals increases, changing the results above.

A Thought Experiment. Despite the good performance of the memory-based prediction in the environment configurations from our experiment, we were also interested in exploring its limitations. We conducted a thought experiment by taking real environments and their trajectories from the users, and varying the location of the non-target object. This exposed a few failure modes of the workspace length-based prediction.

First, if the goals are collinear with the start, and the user trajectory deviates from this line, the method is biased towards the farther goal. We call the "null space effect" of using the
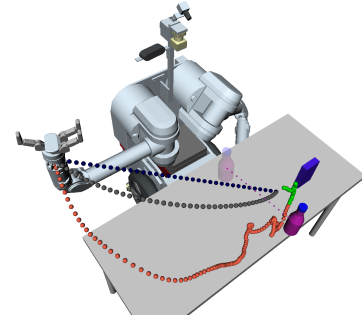

Fig. 7. The user trajectory colored according to the prediction for this artificially generated environment, the trajectory optimizing configuration space length in gray, and the one optimizing work space length in dark blue.

same C for each goal: given two goals G_1 and G_2 s.t. the optimal trajectory to G_2 passes through G_1, the two goals get equal probabilities until G_1 along the optimal path:

$$P(G_1|\xi^*_{S \to U}) = \frac{1}{Z} \frac{\exp\big(-C(\xi^*_{S \to U}) - C(\xi^*_{U \to G_1})\big)}{\exp\big(-C(\xi^*_{S \to G_1})\big)} = \frac{1}{Z}$$

$$P(G_2|\xi^*_{S \to U}) = \frac{1}{Z} \frac{\exp\big(-C(\xi^*_{S \to U}) - C(\xi^*_{U \to G_2})\big)}{\exp\big(-C(\xi^*_{S \to G_2})\big)} = \frac{1}{Z}$$

When a small deviation arises, it decreases $P(G_1| \ldots)$ more, and the method predicts G_2. Even humans are often confused in such situations for example, when interacting in a crowded room. We move towards someone at the back of the room in order to speak with them but are often intercepted along the way by others who predict that social motion incorrectly.

Second, given the trajectories in our data set, the method is in some cases incorrectly biased towards the rightmost object. An example is shown in Fig.7: the trajectory, collected in the environment from Fig.5, heads towards the artificially added obstacle and confuses the predictor. This example should be taken with a grain of salt, because the user would adapt the trajectory to the new environment. However, the more fundamental problem is the incorrect model of human behavior, as shown by Fig.7.

While the first issue could be addressed with priors that prefer closer goals in cases where the goals are "aligned", the second issue calls for learning a better model of how users teleoperate. From our experience, their cost function does not solely depend on lengths or increasing distance from obstacles: *different users develop different strategies for achieving tasks.* We saw some users developing a "move-one-joint-at-a-time" strategy, a way of dividing the problem. We saw differences in trajectories stemming from obstacle avoidance. These are challenges we need to address in order to improve both goal prediction, as well as prediction of the intended motion. Furthermore, we did not ask the users to move in a way that makes their intent clear to the robot. An interesting future experiment would consist of *analyzing how prediction improves when users are trained to provide more intent-expressive inputs.*
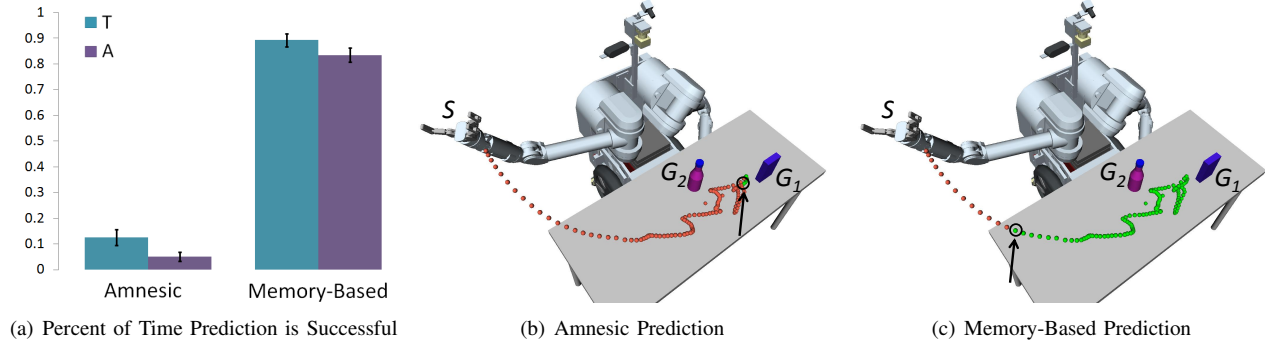
(a) Percent of Time Prediction is Successful   (b) Amnesic Prediction   (c) Memory-Based Prediction

Fig. 5.   A comparison of amnesic vs. memory-based prediction on data from our study.

VI. Conclusion

In this work, we presented a principled analysis of assistive teleoperation. We formalized assistance as policy blending, unifying prior work and providing common ground for future methods and comparisons of assistive teleoperation. We introduced aggressiveness, prediction correctness and task difficulty as factors that affect assistance performance and user preferences, and analyzed their interaction in a user study: arbitration must take into account the robot's confidence in itself (i.e. in the correctness of the predicted policy) and in the user (i.e. in how easy the task is for the user). The challenges we identified lie at the base of human-robot collaboration in general. The robot's motion must make its intent clear to the user, reassuring him of the correct prediction. The user's input must also make its intent clear to the robot, simplifying the prediction task. Furthermore, unlike for typical intent prediction tasks, the robot's prediction directly affects the user's behavior. This gives the learner the opportunity to improve its predictions by explicitly incorporating the user's reaction, and even by taking purposeful information-gathering actions to disambiguate among its hypotheses.

References

[1] L. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Virtual Reality Annual International Symposium*, 1993.
[2] P. Marayong, M. Li, A. Okamura, and G. Hager, "Spatial motion constraints: theory and demonstrations for robot guidance using virtual fixtures," in *ICRA*, 2003.
[3] T. Debus, J. Stoll, R. Howe, and P. Dupont, "Cooperative human and machine perception in teleoperated assembly," in *ISER*, 2000.
[4] E. You and K. Hauser, "Assisted teleoperation strategies for aggressively controlling a robot arm with 2d input," in *R:SS*, 2011.
[5] D.-J. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Port ande, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Trans. on Systems, Man and Cybernetics, Part A: Systems and Humans*, 2011.
[6] P. Marayong, A. M. Okamura, and A. Bettini, "Effect of virtual fixture compliance on human-machine cooperative manipulation," in *IROS*, 2002.
[7] J. Crandall and M. Goodrich, "Characterizing efficiency of human robot interaction: a case study of shared-control teleoperation," in *IROS*, 2002.
[8] P. Aigner and B. McCarragher, "Human integration into robot control utilising potential fields," in *ICRA*, 1997.
[9] J. Kofman, X. Wu, T. Luu, and S. Verma, "Teleoperation of a robot manipulator using a vision-based human-robot interface," *IEEE Trans. on Industrial Electronics*, 2005.
[10] Y. Demiris and G. Hayes, "Imitation as a dual-route process featuring predictive and learning components: a biologically plausible computational model," in *Imitation in animals and artifacts*, 2002.
[11] A. H. Fagg, M. Rosenstein, R. Platt, and R. A. Grupen, "Extracting user intent in mixed initiative teleoperator control," in *AIAA*, 2004.
[12] W. Yu, R. Alqasemi, R. Dubey, and N. Pernalete, "Telemanipulation assistance based on motion intention recognition," in *ICRA*, 2005.
[13] M. Li and A. Okamura, "Recognition of operator motions for real-time assistance using virtual fixtures," in *HAPTICS*, 2003.
[14] D. Aarno, S. Ekvall, and D. Kragic, "Adaptive virtual fixtures for machine-assisted teleoperation tasks," in *IEEE ICRA*, 2005.
[15] P. Abbeel and A. Y. Ng, "Apprenticeship learning via inverse reinforcement learning," in *ICML*, 2004.
[16] B. D. Ziebart, A. Maas, J. A. Bagnell, and A. Dey, "Maximum entropy inverse reinforcement learning," in *AAAI*, 2008.
[17] N. Ratliff, J. A. Bagnell, and M. Zinkevich, "Maximum margin planning," in *ICML*, 2006.
[18] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *HRI*, 2012.
[19] R. Goertz, "Manipulators used for handling radioactive materials," *Human factors in technology*, 1963.
[20] J. Shen, J. Ibanez-Guzman, T. C. Ng, and B. S. Chew, "A collaborative-shared control system with safe obstacle avoidance capability," in *RAM*, 2004.
[21] S. Loizou and V. Kumar, "Mixed initiative control of autonomous vehicles," in *ICRA*, 2007.
[22] C. Smith, M. Bratt, and H. Christensen, "Teleoperation for a ball-catching task with significant dynamics," *Neural Networks*, vol. 21, no. 4, pp. 604 – 620, 2008.
[23] D. Vasquez, T. Fraichard, O. Aycard, and C. Laugier, "Intentional motion on-line learning and prediction," *Machine Vision and Applications*, 2005.
[24] 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 *IROS*, 2009.
[25] S. J. Glynn and R. A. Henning, "Can teams outperform individuals in a simulated dynamic control task," *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 44, no. 33, pp. 141–144, 2000.
[26] C. F. Schmidt and J. D'Addamio, "A model of the common-sense theory of intention and personal causation," in *IJCAI*, 1973.
[27] A. Dragan, G. Gordon, and S. Srinivasa, "Learning from experience in manipulation planning: Setting the right goals," in *ISRR*, 2011.
[28] J. C. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," in *Advances in Large Margin Classifiers*, 1999.