@article{rakita2019shared,
 title={Shared control--based bimanual robot manipulation},
 author={Rakita, Daniel and Mutlu, Bilge and Gleicher, Michael and Hiatt,
Laura M},
 journal={Science Robotics},
 volume={4},
 number={30},
 pages={eaaw0955},
 year={2019},
 publisher={Science Robotics}
}

Preprint -- accepted version

link to paper: https://robotics.sciencemag.org/content/4/30/eaaw0955

Shared-Control-Based Bimanual Robot Manipulation

Daniel Rakita,[1*] Bilge Mutlu,[1] Michael Gleicher,[1] Laura M. Hiatt[2]

[1]Department of Computer Sciences, University of Wisconsin–Madison, Madison, Wisconsin
[2]Naval Research Laboratory, Washington, District of Columbia

[*]To whom correspondence should be addressed; E-mail: rakita@cs.wisc.edu.

Human-centered environments provide affordances for and require the use of two-handed, or "bimanual," manipulations. Robots that are designed to function in, and physically interact with, these environments have not been able to meet these requirements, because standard bimanual control approaches have not accommodated the diverse, dynamic, and intricate coordinations between two arms in order to complete bimanual tasks. In this work, we enable robots to more effectively perform bimanual tasks by introducing a bimanual shared-control method. The control method moves the robot's arms to mimic the operator's arm movements, but provides on-the-fly assistance to help the user complete tasks more easily. Our method utilizes a *bimanual action vocabulary*, constructed by analyzing how people perform two-hand manipulations, as the core abstraction level for reasoning about how to assist in bimanual shared-autonomy. The method infers which individual action from the bimanual action vocabulary is occurring using a sequence-to-sequence recurrent neural network architecture and turns on a corresponding *assistance mode*, signals introduced into the shared-control loop designed to make the performance of

a particular bimanual action easier or more efficient. We demonstrate the effectiveness of our method through two user studies that show that novice users can control a robot to complete a range of complex manipulation tasks more successfully using our method compared to alternative approaches. We discuss the implications of our findings for real-world robot control scenarios.

1 Introduction

Human-centered environments are tailored for two-handed, or "bimanual," manipulations. Ever since neural structures in the brains of human ancestors started evolving to facilitate *interactions* between the two hands, societies began adapting their settings around these abilities, affording complex tool-use, manual labor, meal preparation, and communicative gestures (*1–3*). These societal and evolutionary underpinnings of bimanual processes are clearly evident in day-to-day environments and activities, such as when securing a jar with one hand while twisting its lid with the other to open the jar; when securing a bowl in place with one hand while stirring with the other; when lifting a laundry basket with both hands holding the handles on either side; and when passing plates from one hand to another when setting the table for dinner. While human-centered environments afford, and often require, bimanual manipulations, robots that are designed to function in, and physically interact with, these environments have not been able to meet these requirements.

1.1 How do *robots* currently approach bimanual tasks?

Robot platforms have historically been designed with *single-arm* abilities. Consequently, enabling such robots to perform bimanual tasks requires modifying either the task or the environment. For example, if a robot was tasked with unscrewing the cap off of a water bottle, the bottle would need to be secured to a table ahead of time in order for the robot to execute the task with

a single arm. This single-arm limitation has been shown to make robot actions more difficult to interpret and relate to (*4, 5*) and poses a significant barrier to realizing the full potential of robots for functioning in human environments and assisting people in day-to-day tasks (*6, 7*).

When, alternatively, two robot arms *are* utilized in current manipulation and control methods, the bimanual problem is generally reduced to *concurrent instances of single-arm approaches* or *single-function mechanisms* (for a review, see Smith et al. (*8*). We also provide a Related Works section in Supplemental Materials). Whether it be for multi-arm grasping (*9*); multi-arm motion planning (*10–12*); kinematic controllers (*13–16*), impedance controllers (*17–19*), hybrid-mode controllers (*20–22*); teleoperation interfaces (*23–26*), or active vision (*27–29*), the two robot arms in current methods either exhibit independent behavior with only limited coordination, such as collision avoidance, or show only single-function bimanual abilities, such as only being able to grasp and stabilize an object with two hands without accommodating other bimanual skills such as passing objects from one hand to the other.

1.2 How do *people* approach bimanual tasks?

While robots have not been able to realize the full breadth of bimanual manipulations in human-centered environments, people instinctively perform many such manipulations in day-to-day life. Understanding the differences between current bimanual robot control approaches and the way that the human *brain* considers two-handed manipulations, might reveal factors that contribute to the bimanual manipulation ability gap between people and robots.

Much prior work in neuroscience, neurophysiology, and rehabilitation suggests that current robot control and manipulation paradigms, *i.e.*, considering two-handed control as concurrent instances of single-arm approaches or as single-function mechanisms, do *not* reflect how the brain considers bimanual manipulations. For example, studies have indicated that the brain does not command bimanual manipulations by simply superimposing two independent single-

arm representations (*30, 31*). Instead, dedicated regions of the brain, such as the supplementary motor area (SMA) (*32*) and primary motor cortex (*33*) exhibit unique neural patterns specific to bimanual manipulations (*34*). This effect is illustrated in a study by Ifft et al. where the authors successfully controlled the bimanual arm motions of rhesus monkeys using a brain-machine interface (BMI) by targeting the areas of the brain specific to bimanual movements as opposed to separately targeting the brain regions associated with right- and left-arm unimanual movements (*34*). A leading theory for describing the cognitive bases for bimanual actions, called *internal model theory*, posits that the brain maintains a centralized symbolic representation for bimanual movements arbitrated by specialized brain regions (*35*).

Prior work has also shown the remarkable *dynamism* and *flexibility* of brain activity when performing bimanual tasks (*36, 37*). This dynamic nature of brain networks during bimanual activity facilitates *switching functions* to accommodate various environmental constraints, task difficulty levels, and spatiotempoeral relationships between the two arms (*3*). All leading theories for modeling bimanual behavior, including dynamical systems theory (*38, 39*), muscle synergy theory (*40*), internal model theory (*35*), and optimal feedback control theory (*41*), agree that the nature of coordination between the arms during bimanual movements dynamically changes depending on current task constraints (*42*).

Swinnen et al. combine the concepts of a *centralized action semantics* in the brain during bimanual movements and the *dynamic function switching* dependant on the bimanual task to describe a "gestalt" phenomenon where the individual motions of each arm are promoted to achieve more than the sum of their parts (*3*). This body of work suggests that any bimanual motion planning, control, or manipulation method that does not consider the centralized action semantics or dynamic function switching involved in bimanual actions, such as the current methods outlined above, will fail to achieve this proposed gestalt effect and be limited in applicability and scope.

1.3 Our Solution

The goal of the current work is to extend the abilities of robots to effectively function in human-centered environments by achieving the gestalt effect involved in human bimanual manipulations. To illustrate, consider a bimanual robot platform that is installed in a home environment to provide assistance for an older adult. The robot would need to perform a wide variety of bimanual tasks in this scenario, such as opening pill bottles, carrying a laundry basket, or stirring a meal while keeping the pan stable on the stove. Each of these tasks is composed of various "bimanual actions," individual types of two-handed movements that would benefit from a particular control strategy. Our goal is to capture these diverse, dynamic, and intricate actions and interactions between the hands that commonly occur throughout bimanual manipulations in such tasks and to enable control mechanisms that support these interactions. For example, when carrying a laundry basket, the *correspondence* between the two hands, specifically the fixed translation and rotation offset of the hands as dictated by the laundry basket between them, is more critical than the individual motions of each independent hand. Thus, a central premise throughout our work is that a successful bimanual control method will leverage *the higher-level actions and interactions between the two hands*, which often take precedence over the independent behavior of each hand.

To develop such control for robots that considers the higher-level actions and interactions between the hands, we argue that three technical challenges must be addressed:

1. How should the robot *organize* the wide range of bimanual manipulations in a manner that allows us to provide mechanisms to support them?

2. How should the robot *identify* which bimanual coordination is needed so that it can apply the appropriate control strategy?

3. What *control strategy* should the robot implement for each bimanual coordination type?

Our current work explores bimanual manipulation in a *real-time control* scenario. Specifically, we formulate the problem as a bimanual *shared-control* method, *i.e.*, a control method that aims to reduce the tedium or difficulty of direct-control by enabling the robot to handle some aspects of the control process (*43*). Using such a method, the robot is able to arbitrate between a user's command inputs and its own underlying motion policies and understanding of bimanual tasks. Our methods identify which action the user is performing and adapt the control algorithm to provide assistance in executing the action by adapting the robot's movements. For example, in the laundry basket carrying example provided above, even if the user does not exactly maintain the fixed-offset task constraint while specifying their motion inputs, the robot could utilize its *understanding* of the underlying bimanual coordination to maintain a fixed offset between its end-effectors such that it does not drop or break the laundry basket. An overview of our work can be seen illustrated in Figure 1. While our current work frames our proposed approach in real-time control, we expect our core ideas to apply more generally, for example, to a system in which the inputs come from an autonomous robot's perception and planning algorithms instead of a human operator.

1.3.1 How should the robot *organize* the range of bimanual manipulations?

To address the first technical challenge, we explored the idea that there exists a small set of "bimanual action" classes that can abstractly represent a wide range of possible bimanual manipulations. Supported by the theories discussed above, particularly the theory that the brain maintains a centralized action semantics in specialized brain regions (*3, 35*), our proposed method represents bimanual actions as labeled elements in a *bimanual action vocabulary* that spans possible forms of coordination that the two arms may try to achieve in a task.

Given the premise that a concise set of actions and interactions between the hands will often take precedence over the independent actions of each hand during bimanual manipulations, as

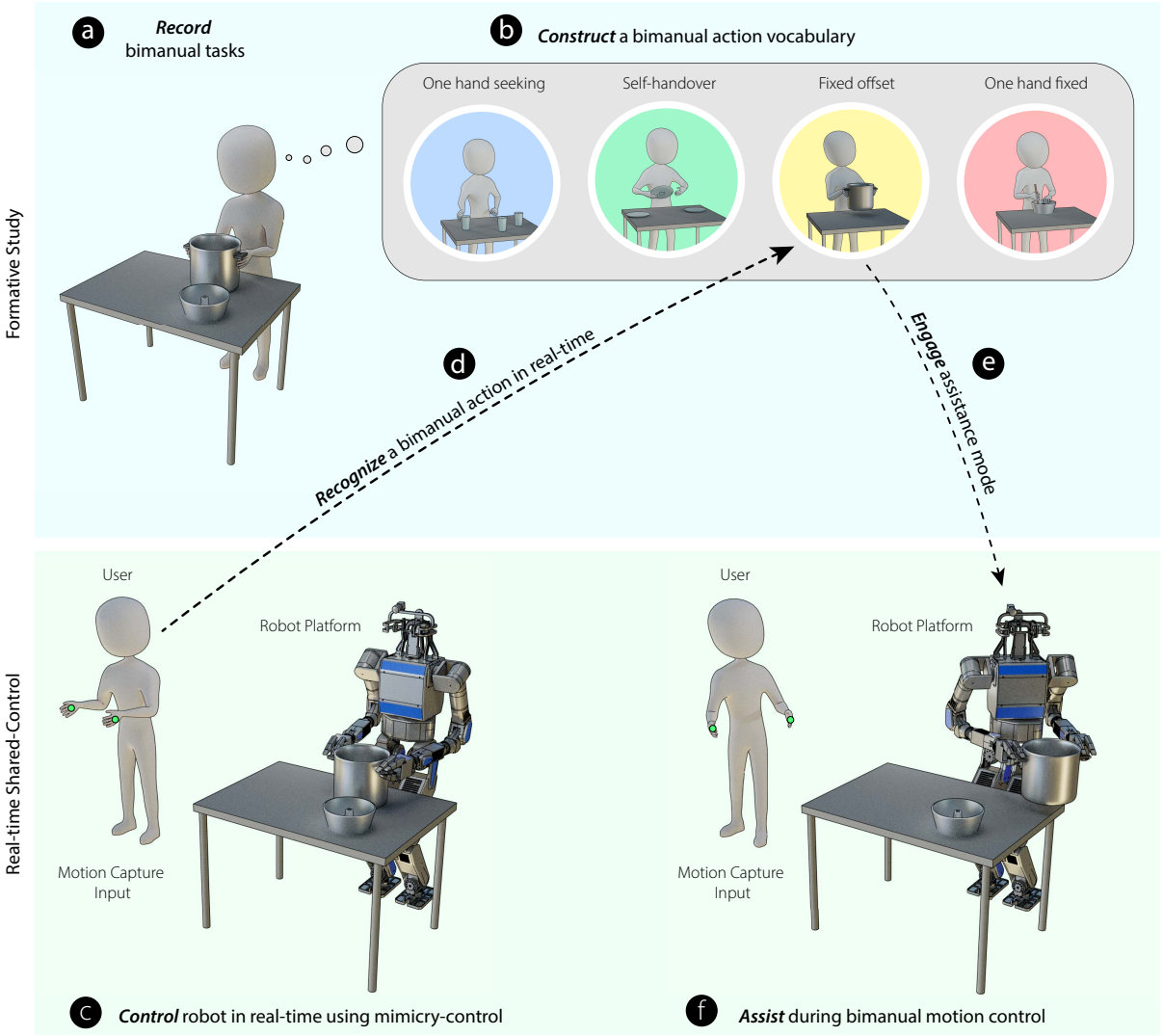

Figure 1: In this work, we present a shared-control method for effective bimanual robot manipulation. (a) We constructed a motion dataset of people performing two-handed tasks and (b) extracted high-level kinematic patterns from the data to build a compact and lightweight *bimanual action vocabulary* that sufficiently spans the space of two-handed actions. (c) While the user is controlling the robot, (d) the method infers which action from the bimanual action vocabulary is most likely being specified by the user and (e) engages an appropriate *assistance mode* (f) to help during the respective bimanual action.

supported by prior work above, a natural question arises: *what are these actions and interactions between the two hands*? Prior work has successfully isolated a set of control semantics in

the brains of rhesus monkeys using surgically implanted electrodes to activate their bimanual actions (*34*). We argue that, if the brain has a representation of central bimanual actions that it uses to organize the interactions, constraints, cooperative movements, and asymmetric movements between the user's hands, a theory supported by much prior work (*3, 35, 40, 42*), then we should be able to *isolate*, *recognize*, and *label* such features as *distinct patterns in the user's hand motions* when performing bimanual tasks. To explore this premise, we first conducted a formative study in which we recorded and analyzed the hand motions of human participants through various bimanual tasks. Through a kinematic pattern analysis, we distilled this space down to a *bimanual action vocabulary* designed to characterize the bimanual manipulation space in a flexible and comprehensible manner. An evaluation of the vocabulary shows that it can serve as an effective abstraction level to specify to the robot how bimanual tasks should be accomplished.

1.3.2 How should the robot *identify* which action is needed?

Our solution to the second technical challenge involves classifying which bimanual action from the bimanual action vocabulary is most likely being specified by the user at a given time. The method observes the recent stream of the user's motion inputs as a state model and uses a sequence-to-sequence recurrent neural network to infer the most probable bimanual action being specified. This solution is analogous to how the brain considers bimanual neural processes, according to the internal-model theory of bimanual action specification (*35*). The method recognizes from a centralized action semantics how the two hands are likely to be coordinating, and is subsequently able to modulate the control processes based on the currently inferred task constraints.

1.3.3 What *control strategy* should the robot implement for each bimanual action?

Our solution to the third technical challenge involves a control approach that enables the robot to move its arms in a coordinated fashion following three phases. First, it captures the poses (position and rotation) of the user's hands at each update so that it can *map* the user's hand motions onto the robot's end-effectors in real-time. This step, called *motion-retargeting*, involves mapping motion from one articulated figure (*e.g.*, a teleoperator, motion-capture actor, etc.) to a potentially vastly dissimilar articulated figure (*e.g.*, a robot, animated character, etc.), such that important motion or pose properties are maintained (*44, 45*). This process allows users to specify what actions they want the robot to perform by simply and naturally providing desired motions with their own arms, an approach that we have termed *mimicry-control* in prior work (*46, 47*). Second, based on the recent stream of the user's hand poses, the method utilizes our solution to the second technical challenge in the control loop to infer which bimanual action from the *bimanual action vocabulary* that the user is most likely trying to specify to the robot using a sequence-to-sequence recurrent neural network. And third, given the inferred bimanual action at the current time, the method dynamically engages an appropriate *assistance mode*, signals introduced into the shared-control loop designed to make the performance of a particular bimanual action easier or more efficient. Thus, while the user *feels* as if they have direct control over both of the robot's arms, the robot subtly *overrides* their direct inputs in order to meet task constraints between the two hands given the currently inferred bimanual action. Two examples of assistance modes in our method (outlined in more detail below) are constraining the two hands to maintain a fixed translation and rotation offset when lifting a rigid object with two hands and ensuring that the robot's end effectors are at the correct distance from each other when performing a self-handover.

To establish the effectiveness of our proposed solutions, we have evaluated our bimanual shared-control method through two laboratory studies involving naïve human participants. The

first study compared our method against alternative control approaches, including an unassisted bimanual approach, as well as a state-of-the-art single-arm control interface. The second study provides details on the relative contributions of each assistance mode used in our method. The results of these studies provide insight into the potential impact of introducing bimanual robot operation on real-world robot control scenarios and other related human-robot collaboration domains, including fully autonomous bimanual manipulations and bimanual robot teaching.

The contributions of our overall work include (1) introducing an appropriate abstraction level for how people instinctively perform tasks with two hands, (2) enabling robots to utilize this abstraction to interpret and reason about bimanual manipulations in real-world environments, and (3) making robot control in real-world, human-centered environments easier and more effective, even for novice users, by providing users with the ability to control the robot analogously to how they would naturally perform tasks with two hands themselves.

2 Results

This section presents the technical solutions to the three challenges outlined above and the findings from their evaluation. Further technical, implementation, and experimental details are provided in Supplementary Materials.

2.1 Solution to Challenge 1: Bimanual Action Vocabulary

A central premise throughout our work is that a successful bimanual shared-control method will tune the control behavior based on the higher-level *actions* and *interactions* between the two hands. To realize this robot behavior, we must first discover and define these actions and interactions between the two hands. Our strategy to discover these bimanual actions and interactions in this work is: (1) Record a dataset consisting of participants' hand poses over time while executing various bimanual tasks; (2) Filter the dataset such that data signals are not conflated

with extraneous noise; (3) Analyze the dataset to assess what high-level patterns emerge when people perform bimanual tasks; (4) Organize the observed bimanual patterns into a bimanaul action vocabulary, such that the elements in the action vocabulary are robust, interpretable, and cover a wide breadth of common bimanual manipulations.

In this section, we describe the kinematic pattern analysis we conducted in order to distill the space of bimanual manipulations down to a bimanual action vocabulary.

2.1.1 Formative Study

To form a bimanual action vocabulary, we first assessed how *people* perform bimanual tasks. We ran a participant study where we collected a bimanual manipulation dataset by recording the 6-degree of freedom (translation and rotation) pose information of each participants' hands while they perform a set of two-handed tasks. Details about how we collected and filtered our dataset can be found in Supplemental Materials. Our whole dataset is structured as:

$$D := \{(\mathbf{p}_1^d, \mathbf{p}_1^n, \mathbf{q}_1^d, \mathbf{q}_1^n), (\mathbf{p}_2^d, \mathbf{p}_2^n, \mathbf{q}_2^d, \mathbf{q}_2^n), ..., (\mathbf{p}_T^d, \mathbf{p}_T^n, \mathbf{q}_T^d, \mathbf{q}_T^n)\}$$

Here, $\mathbf{p}_t^d$, $\mathbf{q}_t^d$, $\mathbf{p}_t^n$, and $\mathbf{q}_t^n$ are the positions and orientations of the dominant and non-dominant hand at time step t, respectively, and T denotes the final time step.

2.1.2 Feature Construction

Given the collected and filtered dataset, our goal is to use the data to categorize actions and interactions that occur between peoples' hands during bimanual manipulations.

We transform the data to encode each pose in a manner that allows for analysis of the interactions between hands. For each hand-pose (pair of 6-DOF hand configurations), we compute six different scalar valued features that measure various relationships between the hands, independent of their absolute position. The values use only relative coordinates such that models

learned through our analyses are not tied to the absolute coordinate frame of the motion capture setup. The six scalar values in a are (1) the distance between the hands (denoted as *hand offset*); (2) the rate of change of the distance between the hands (denoted as *hand offset velocity*); (3) the *dominant hand's translational velocity*; (4) the *dominant hand's rotational velocity*; (5) the *non-dominant hand's translational velocity*; and (6) the *non-dominant hand's rotational velocity*. A feature at a single time-point, $\mathbf{s}_t$ is structured as:

$$\mathbf{s}_t = \begin{bmatrix} ||\mathbf{p}_t^d - \mathbf{p}_t^n|| \\ ||\mathbf{p}_t^d - \mathbf{p}_t^n|| - ||\mathbf{p}_{t-1}^d - \mathbf{p}_{t-1}^n|| \\ ||\mathbf{p}_t^d - \mathbf{p}_{t-1}^d|| \\ ||disp(\mathbf{q}_t^d, \mathbf{q}_{t-1}^d)|| \\ ||\mathbf{p}_t^n - \mathbf{p}_{t-1}^n|| \\ ||disp(\mathbf{q}_t^n, \mathbf{q}_{t-1}^n)|| \end{bmatrix} \tag{1}$$

Here, velocities are approximated using backwards finite differencing, and $disp$ is the standard displacement operator for quaternions: $disp(\mathbf{q}_1, \mathbf{q}_2) = log(\mathbf{q}^-1_1 * \mathbf{q}_2)$ ((*48*)). While we show that these six features are sufficient for representing interactions between the two hands in our kinematic pattern analysis, we note that this set of features is only one of many possible sets. For example, one could supplement this set of features with force and moment information in order to also characterize the dynamics of bimanual manipulation in an action vocabulary.

Because the features $\mathbf{s}_t$ only encode a single, discrete hand pose event, we also *window* many such events together in a long, concatenated vector to encode motion over time:

$$\mathbf{f}_t = \begin{bmatrix} \mathbf{s}_{t-\omega/2} \\ \mathbf{s}_{t-\omega/2+1} \\ \vdots \\ \mathbf{s}_t \\ \vdots \\ \mathbf{s}_{t+\omega/2-1} \\ \mathbf{s}_{t+\omega/2} \end{bmatrix} \tag{2}$$

In our analyses, a single feature encodes a second of motion ($\omega = 80$).

2.1.3 Principal Components Analysis

Given the feature vectors $\mathbf{f}_t$, our goal is to analyze these vectors such that high-level patterns characterizing the actions and interactions between the hands emerge. Our conjecture is that, while our feature-vector space has many, high-dimensional data points, there is a set of just a *few* component vectors that can effectively span this space. If such a small set of components exists, this would indicate that only a few central kinematic actions arise when people do bimanual tasks.

To search for such a set of bimanual actions, we use principal components analysis (PCA), a common statistical technique for doing dimensionality reduction (*49*). The scree plot at the top of Figure 2 indicates the explained variance with respect to each principal component. We observe that most of the variance in the bimanual dataset can be explained with just the first seven principal components, before hitting an inflection point on the scree plot and leveling out. Figure 2b illustrates the resulting principal components from the PCA to give a sense of what the dimensions of the bimanual action-vocabulary space represent. In the following section, we overview how we utilize these principal components in our subsequent manual analysis.

2.1.4 Bimanual Actions from Analysis

In this section, we attach semantic meaning to each of our principal components using a manual, post-hoc analysis in order to construct a bimanual action vocabulary. We manually organized and labeled the top PCA components in order to clarify what these components actually mean in the context of a bimanual manipulation task. Our analysis first involved finding points within our motion dataset that correspond well with the particular principal components. For example, when assessing the high-level semantic action corresponding to principal component 1 (the blue curve in Figure 2b), we found points close to represented as $[a, 0, 0, 0, 0, 0, 0]^T$ using the principal components as a basis, for some real value a. Then, we reviewed the study videos to

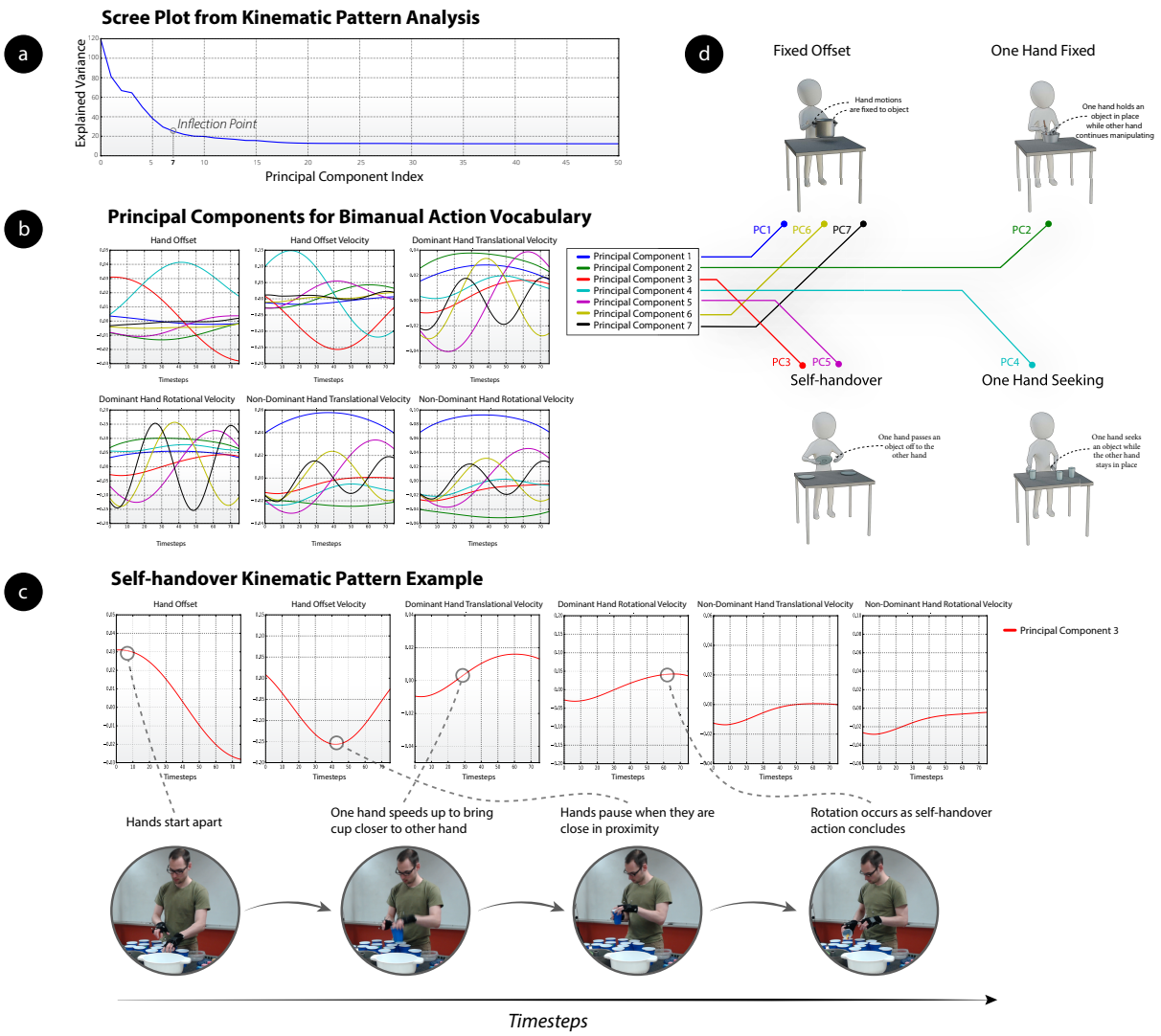

Figure 2: (a) Scree plot from our kinematic pattern analysis using PCA. The inflection point indicates that seven principal components cover much of the variance in the bimanual action space. (b) The first seven principal components (displayed as colored lines in the graphs) shown over 80 time-steps per each of our six kinematic features (hand offset, hand offset velocity, etc.). (c) Illustration of how the third principal component (the red lines in Figure 2b) connect to the self-handover bimanual action. The dotted lines point to particular landmarks over the different kinematic features that characterize the self-handover action. (d) The seven principal components are grouped into four high-level "words" in our bimanual action vocabulary: fixed offset, one hand fixed, self-handover, and one hand seeking.

interpret and further organize how the participants' hands were interacting with one another at these representative points. This type of analysis would be infeasible to do in an automated way due to the highly contextualized nature of the assessment. The resulting bimanual actions from our analysis are explained here and can be seen summarized in Figure 2:

1. *Fixed Offset* — Principal components 1, 6, and 7 all correspond to a similar kinematic pattern where the offset between the hands does not exhibit much change, and the translational and rotational velocities of both of the hands follow similar characteristics. Upon further investigation, we found that these motions happened when participants were holding an object with two hands, such that the offset of the hands was dictated by the object being moved. The hands had to follow similar translational and rotational velocity profiles since the rigid object between the hands constrained each hand from exhibiting its own independent translation or rotation.

2. *One hand fixed* — Principal component 2 corresponds to the two hands working in close proximity, where one hand is relatively stationary, while the other hand maintains a high translational and/or rotational velocity. We found that these motions occurred when one hand was holding an object in place while the other hand did some manipulation with respect to this object. An example of such an action is holding a bowl in place with one hand while stirring in the bowl with the other hand.

3. *Self-handover* — Principal component 3 corresponds to the hands coming together quickly, before easing up at the end of the action. Principal component 5 consists of the hands starting together and moving apart at a moderate speed. Upon further investigation, we found that these two components occur sequentially when participants are initiating and completing a self-handover, respectively.

4. *One hand seeking* — Principal component 4 correspond to the hands moving apart at a

fast rate, with one hand maintaining a high translational and rotational velocity, while the other hand remains relatively still. When investigating further, we observed that this action occurred when participants were reaching for an object in the workspace. We note that the top principal components here do not exhibit any motions where the hands move apart while *both* hands maintain a high velocity. We believe this indicates that people rarely reach for separate objects simultaneously with both hands, instead only reaching with one hand at a time. Thus, our bimanual action here is *one hand seeking* for an object.

2.2 Solution to Challenge 2: Bimanual Action Inference

In order to appropriately provide support mechanisms in the bimanual controller for the bimanual actions specified above, the method must have some way to discern which bimanual action is *currently happening*. To do this, we introduce a bimanual action inference method using a sequence-to-sequence recurrent neural network architecture. We chose to use a recurrent neural network architecture in this scenario because of its reported successes in inferring events in time-series data in prior work (*50, 51*). Specifically, we take advantage of the temporal structure in this data stream by using a recurrent neural network using a single long short-term memory (LSTM) layer (*52*). For full implementation and analysis details on our bimanual action inference approach, refer to the Supplemental Materials.

2.3 Solution to Challenge 3: Shared-Control-Based Bimanual Telemanipulation

Our next goal is to utilize the understanding of how people perform bimanual tasks in order to afford an effective shared-control method for bimanual manipulators. A high-level overview of our shared-control method is provided in the Introduction.

2.3.1 Real-time Motion Retargeting using RelaxedIK

We remap the motions of the user's hands onto the robot's hands on-the-fly through a process called *motion retargeting*. At each update, our shared-control method calculates joint angles during the retargeting step through a process called *inverse kinematics (IK)* (see Aristidou et al. (*53*) for a review of common IK methods). Prior work shows that using a standard IK solver for human-to-robot motion-retargeting is not effective, as the resulting sequence of joint angle solutions exhibit infeasible motion qualities, such as self-collisions, kinematic singularities, and joint-space discontinuities (*46*).

To address these problems, we use an optimization-based inverse kinematics solver, called *RelaxedIK*, that is able to handle trade-offs between many objectives on-the-fly (*54*). The key insight in this method is that exactly matching the end-effector pose goals does not have to be a hard constraint, and instead other goals, such as smooth joint motion, kinematic singularity avoidance, or self-collision avoidance, could be more important in certain situations. Certain sub-goals are considered less important on-the-fly and are automatically *relaxed* or de-emphasized. RelaxedIK has been shown to be successful for human-to-robot motion-retargeting in prior work (*28, 46, 47*). An abbreviated description of the problem formulation, structure, and notation behind RelaxedIK can be found in Supplemental Materials. While our current implementation utilizes RelaxedIK to perform the on-the-fly motion retargeting, other motion optimization frameworks that handle multiple kinematic chains or humanoid structures, such as the Stack-of-Tasks (*55*), may also successfully support our presented methods.

2.3.2 Bimanual Assistance Modes

In this section, we overview how we assist in our shared-control method for each action in our bimanual action vocabulary. All mathematical details for each assistance mode can be found in Supplemental Materials.

1. *Fixed Offset*: The fixed offset bimanual action occurs when a user has picked up and is moving an object with both hands. In this situation, the translation and rotation of the user's hands are constrained to the object being manipulated. The high-level idea behind the fixed offset assistance mode is to keep the robot's end-effectors at the same distance with the same relative translation and rotation offset throughout the whole bimanual action such that the rigid object is successfully moved with the cooperating hands. Without this fixed spatiotemporal offset between the hands, it would be difficult for users to provide sufficient independent inputs from both hands that meet the task constraints.

   To provide assistance throughout this mode, we approximate a coordinate frame pose of the object being manipulated throughout the fixed offset action, then add objectives of high importance to both of the robot's end-effectors to maintain the poses of the robot's hands with respect to the proxy object frame.

2. *One Hand Fixed*: The one hand fixed action typically occurs when one hand is holding an item in place such that the other hand can finely perform manipulations with respect to the static object. To assist during this action, we encourage the hand detected to be stationary in the manipulation to *remain fixed*. For example, if the robot were pouring a liquid into a cup, the end-effector holding the cup would remain static; even if the user exhibited small motion perturbations in the input signal, the robot would ignore this noisy behavior in favor of keeping the hand fixed, making the pouring action by the other hand easier to execute.

3. *Self-handover*: A self-handover action occurs when an object is passed from one hand to the other. To assist with this action, we make two adjustments to the control algorithm. Our first adjustment is to gradually *decelerate* the robot's end-effectors as they come together when the action is first detected. This assistance is designed to mimic the velocity

profile observed when people do self-handovers, as highlighted in our kinematic pattern analysis.

Our second adjustment is to ensure that the robot's end-effectors are close together when the user's hands are close together. Without this assistance, the user would need to cross their arms or keep their arms far apart during a handover if the robot's arms have a different scale and geometry from the operator's arms. To correct for this, we shift more importance in the control algorithm to the *absolute distance* between the robot's end-effectors when the user's hands are close together. Thus, a self-handover action being specified by the user will maintain a strong motion correspondence with the robot when it executes the self-handover action.

4. *One Hand Seeking*: The one hand seeking action occurs when one hand is reaching out for an object while the other hand is not active in the manipulation. We assist during this action by placing more *relative importance* on matching the position and rotation end-effector goals on the *seeking* hand, meaning that small position and rotation errors are considered more allowable on the *non-seeking* hand.

   Because RelaxedIK is an optimization-based inverse kinematics solver that can make trade-offs between many objectives, the goal of this assistance is to provide the solver with a sense of *importance* of one hand versus the other. Through this process, more effective joint-configurations can be solved at each update that exhibit better matching of the hand that is important as opposed to matching the hand that is not currently contributing to the task.

2.4 Evaluation of the Proposed Approach

In this section, we outline the results of our two user studies conducted to evaluate our method against alternative approaches for telemanipulation. All details about the design of the two studies, including the hypotheses tested, tasks, procedure, measures, and data-analysis methods can be found in the Materials and Methods section.

Study 1: Effects of Bimanual Assistance—In Study 1, we evaluated the performance of our bimanual shared-control method against alternative approaches on a set of complex manipulation tasks. We present results based on a repeated-measures ANOVA analysis. All pairwise comparisons used Tukey's HSD test to control for Type I error in multiple comparisons. Results can be seen summarized in Figure 3.

Study 2: Relative Contributions of our Assistance Modes— In Study 2, we evaluated the relative contribution of each of our bimanual assistance modes. We present results based on a repeated-measures ANOVA analysis. All pairwise comparisons used Tukey's HSD test to control for Type I error in multiple comparisons. Results can be seen summarized in Figure 4.

3 Discussion

In this work, we extended robot manipulation abilities to include two arms by formalizing a shared-control method designed to afford effective execution of bimanual tasks. Below, we suggest the main takeaways from our work based on results from our two user studies, provide more specific discussion of the results of our studies, and overview limitations and extensions for our work.

3.1 Overview of Takeaways

Based on results from our two user studies, we suggest four main takeways from our work (below). A more thorough discussion of our study findings can be found in Supplemental

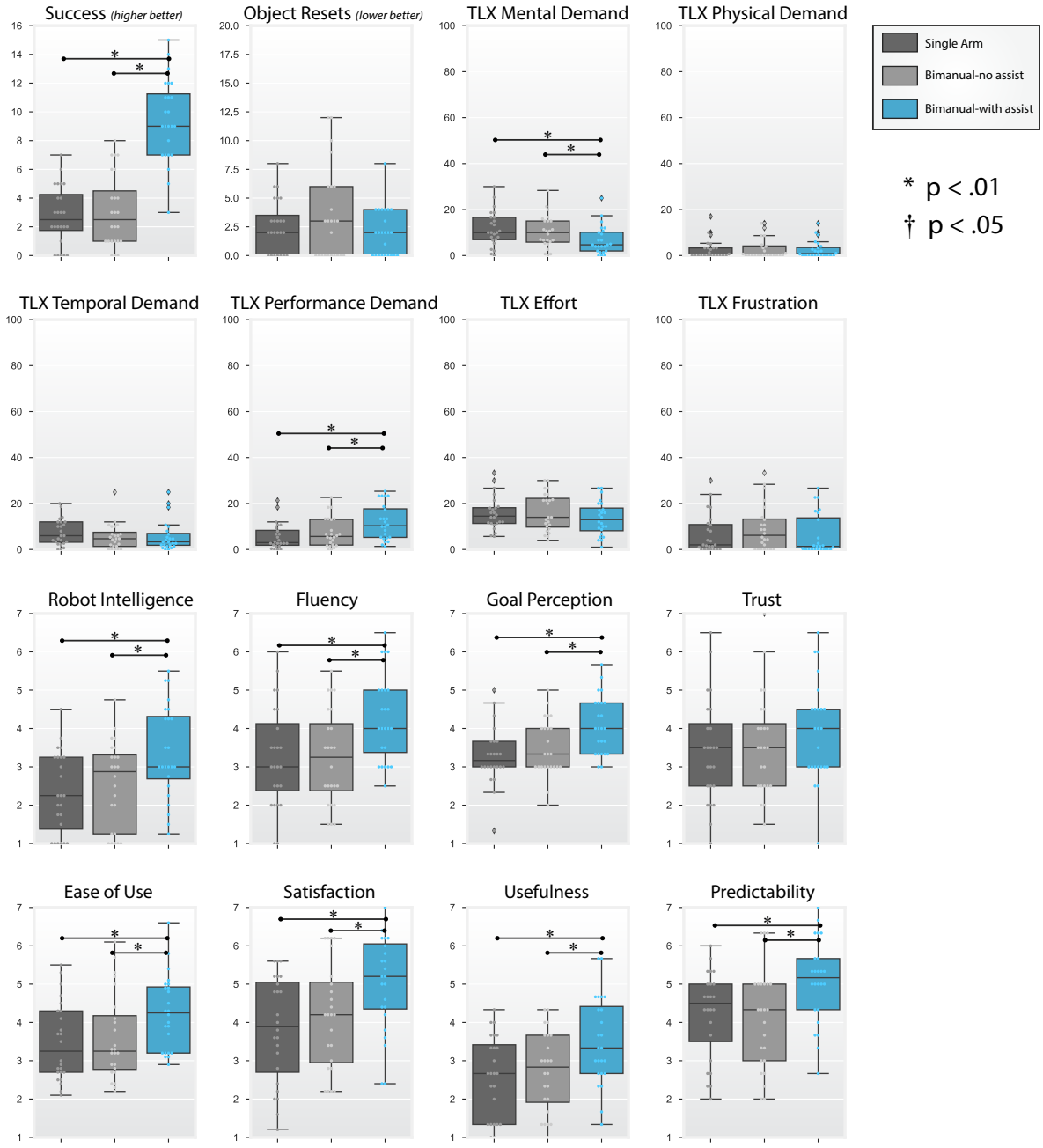

Figure 3: Tukey boxplots overlaid on data points from objective and subjective measures, displaying results from Study 1. Error bars designate standard error of the mean. This study used a sample size of 24 participants ($N = 24$), and p values are indicated by the * and † symbols.

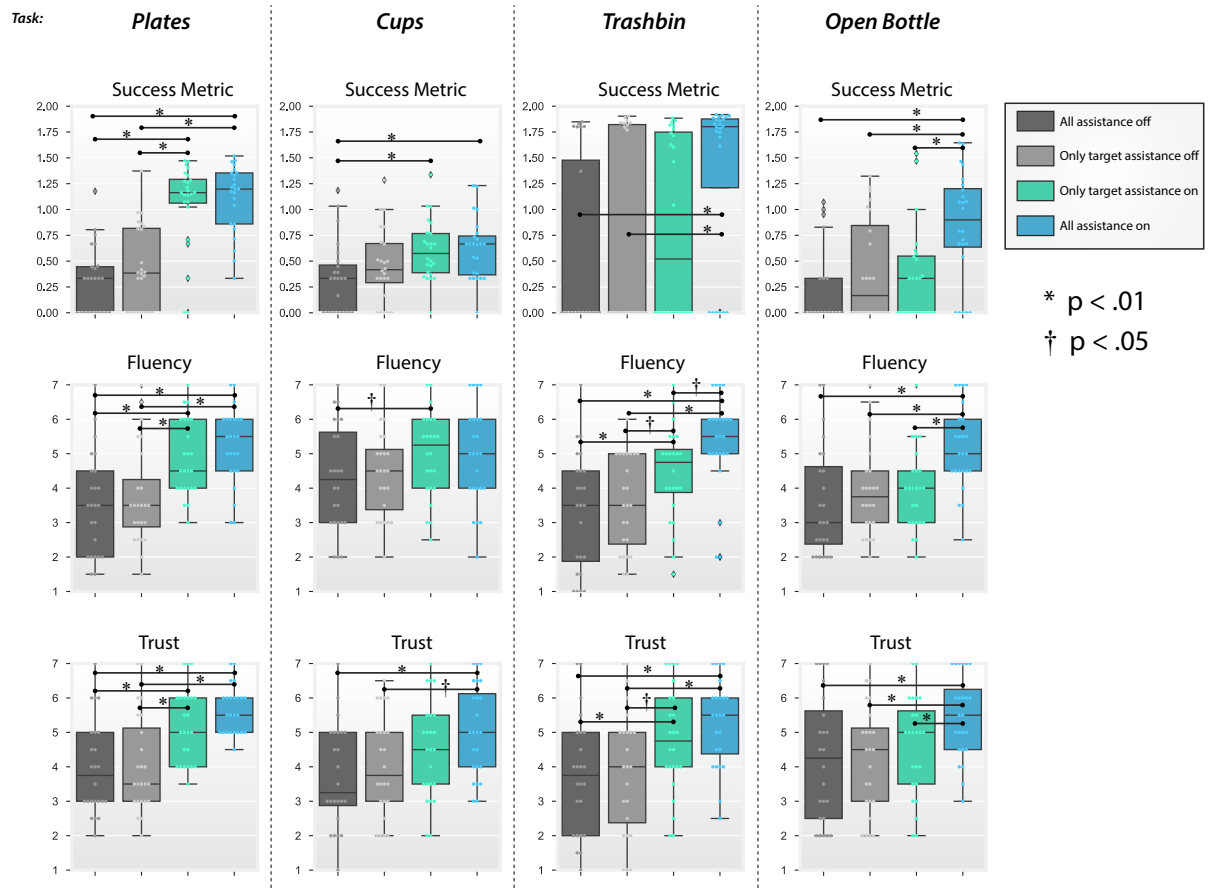

Figure 4: Tukey boxplots overlaid on data points from objective and subjective measures, displaying results from Study 2. Error bars designate standard error of the mean. This study used a sample size of 24 participants ($N = 24$), and p values are indicated by the * and † symbols.

Materials.

(1) The benefits observed by a bimanual robot control platform are elicited from reasoning about and leveraging the *actions* and *interactions* between the two arms, rather than simply having two arms involved in the manipulation. In other words, we observed no benefits in using two robot arms over a single robot arm in our robot control scenario when task intuition and corresponding assistance provided by our bimanual action vocabulary was removed. Thus, we argue that the robot having this intuition about how two arms interact with each other in tasks

is paramount when two arms are present.

(2) The use of "assistance modes" rooted in a bimanual action vocabulary provides task success and user perception benefits across a wide range of manipulation tasks.

(3) A particular "assistance mode" may have individual bearing on a particular task when used independently, though they appear to have a "gestalt effect", where even more benefits are present when multiple assistance modes are afforded to the user. Thus, the ability to dynamically switch between assistance modes, such as when using our proposed sequence-to-sequence recurrent neural network inference method, is an integral part of a successful bimanual shared-control approach.

(4) A bimanual robot control platform is capable of affording task performance and user perception benefits over a single-arm robot platform when used for complex manipulation tasks in human-centered environments.

3.2 Limitations & Extensions

Our bimanual shared-control method has limitations that suggest many future extensions to our work. First, our method currently only considers the kinematics of the bimanual platform, rather than also considering the dynamics of manipulations. Extensions of our proposed solutions could supplement our motion pattern analysis to include force and torque data, and extend our bimanual action vocabulary to include these extra dimensions. Using the new dynamics-infused bimanual action vocabulary, we could in turn extend our assistance modes to compliant control laws that dictate how the arms should work in tandem when manipulating objects. Force information could also be reflected back to the teleoperator using haptics.

Also, our method only considers motion at the level of the user's hand poses, though we speculate much of human-level dexterity when using two hands is at the level of the user's fingers. Including high fidelity finger motion and dynamics information at the users' fingertips

could provide rich data to bolster both our bimanual action vocabulary and resulting assistance modes. We also note that the assistance modes in our current method are hand-designed and based on heuristic choices. Extensions of our work could explore ways to automate the mapping of the various control processes, such as identifying control laws directly from human motion and force signals that could generalize to various robot platforms.

A key lesson from our work is that explicitly supporting bimanual actions through dual-arm control algorithms provides value over controlling arms individually. While our experiments only show the benefits relative to a specific single-arm controller, we believe that the control methods we introduced, as well as the general principles of bimanual action selection, can be used to extend other single-arm control approaches. Similarly, we believe that our action-based control methods could be incorporated to enhance other bimanual control schemes.

Lastly, our current method does not consider the user's view of the environment when controlling the robot's arms. Our current studies only considered cases where the user was co-located with the robot such that the user had direct line-of-sight of the workspace; however, in real-world use cases, it would be common for the robot to be deployed in a remote location, making the visibility problem a pertinent consideration. Our recent work in remote telemanipulation has stressed the importance of situational awareness when controlling a robot arm remotely (*28*). This prior work, however, does not consider the remote visibility problem when controlling two arms simultaneously. Methods that autonomously provide an effective view of a remote environment for a user controlling a bimanual robot platform would bolster our current shared-control approach.

4 Materials and Methods

4.1 Study Design

In this section, we describe the design of our two studies, including the hypotheses tested and study design, tasks, procedure, measures, and data-analysis methods. Implementation details for the prototype system used throughout our evaluations can be found in Supplemental Materials. The evaluations conducted in this work were approved by the Institutional Review Board at the Naval Research Laboratory in Washington, D.C.

Procedure— Both of our studies followed a common procedure. Following informed consent, participants were provided with information on the goals of the study and were invited to ask any questions they had. The participants first put on a pair of velcro motion-capture gloves, stood in a fixed location next to the robot, and waited in a comfortable initial pose with their palms level to the floor and fingers facing forward. The standing spot was selected to provide a sufficient vantage point for all sub-tasks and to ensure that participants would be out of the robot's range of motion at all times for safety. The experimenter guided the participants through a practice phase on how to control the Hubo robot. The system was initialized by the experimenter, then the experimenter counted down from five to signal when the participant would have control. Once the system counted down, the participant could move his/her arms and hands in free-space to practice using the control system by picking up an empty soda bottle for up to four minutes. Note that the the participants could practice moving both arms during the training phase, though the practice task just required a single arm. This training strategy was used so that the user could get accustomed to initializing the system and moving the robot's arms without focusing on the more pertinent aspects of bimanual assistance.

After the participants felt sufficiently comfortable using the control system, they took a short break while the experimenter set up the task (the task(s) for each study are outlined below). The

participants alternated between performing the tasks under different conditions and completing a questionnaire regarding their last robot control experience. This procedure was repeated until all tasks per condition were completed. After finishing all tasks, the participants completed a demographic questionnaire and were then debriefed on the details of the study.

Study 1 Experimental Design and Tasks— Study 1 followed a 3×1 within-subjects design. The participants used three control methods (*single-arm control*, *bimanual control-non-assisted*, *bimanual control-assisted*) to complete a breakfast-making task. The *single-arm control* case was included as a state-of-the-art telemanipulation method comparison from prior work by Rakita et al. (*46*), wherein the authors reported the success of this approach over other interfaces such as a 6-DOF stylus device and touch screen interface. We included this comparison to assess how this previously reported on interface extends to more dexterous manipulations and an experimental setup that is more reflective of standard human environments where two hands are often helpful. The *bimanual control-non-assisted* condition applies the method seen in the prior work by Rakita et al. (*46*) on two arms, essentially treating the bimanual problem as just two independent instances of single arm mimicry-control simultaneously. We included this condition to see how having two arms during robot control, even when the two arms do not have a sense of each other or the actions and interactions involved in bimanual actions, will affect task performance and user perceptions. Lastly, the *bimanual control-assisted* condition implements all of our bimanual assistance modes as inspired by our *bimanual action vocabulary*, presented throughout our work. The conditions were presented in a counterbalanced order between participants. A participant can be seen executing various tasks from Study 1 in Figure 5.

To ensure the generalizability of our findings to a wide range of telemanipulation tasks in a standard human-centered environment, we developed a breakfast-making task consisting of an array of sub-tasks. We chose this task because one of the motivating research domains

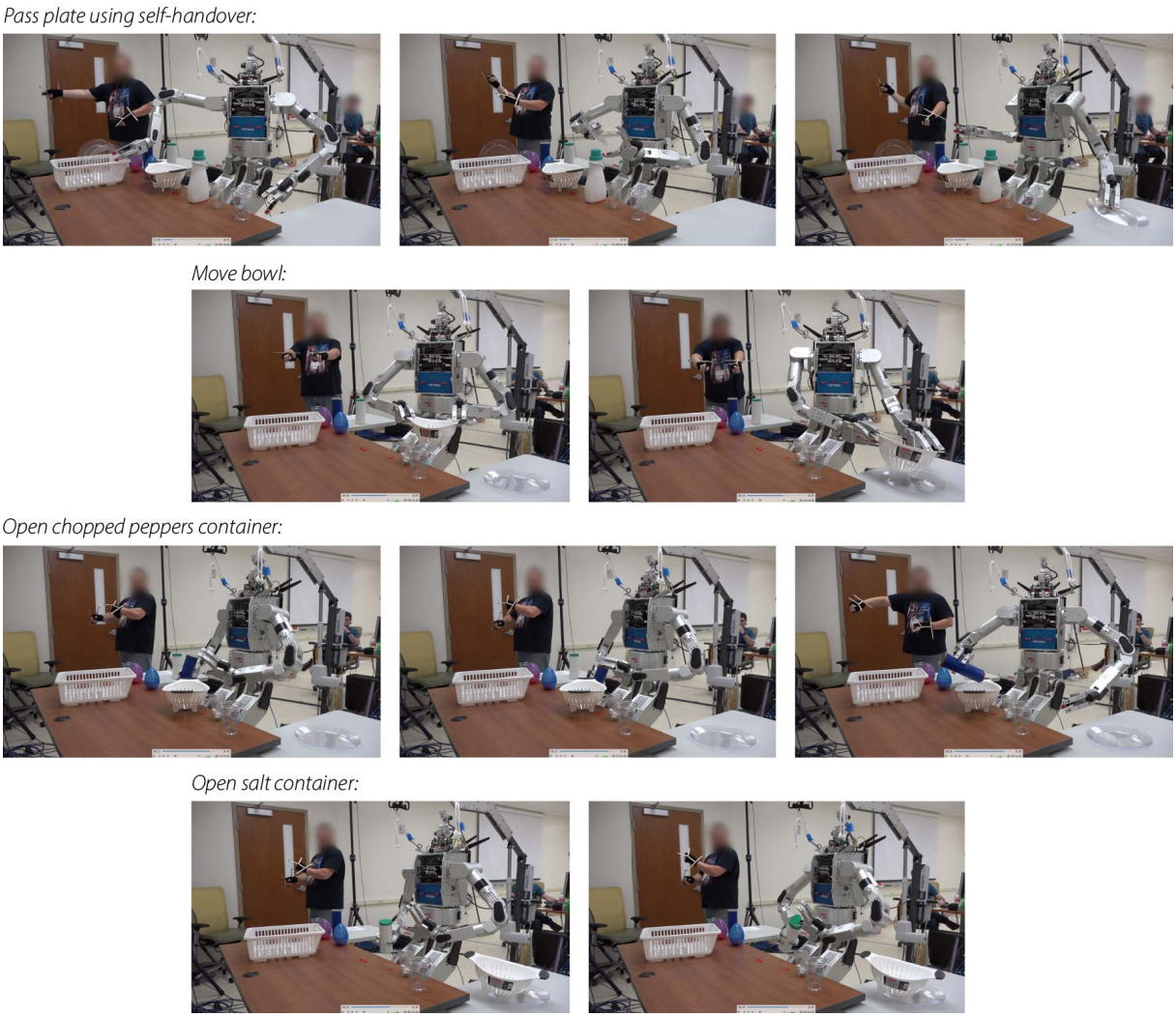

Figure 5: A novice user executing various sub-tasks from Study 1. The sub-tasks in Study 1 followed a breakfast-making theme, matching our motivated use-case of our bimanual shared-control method being used in remote home-care or telenursing scenarios.

for our work is remote home-care; thus, showing benefits in a simulated domain may indicate the potential of our methods to be used in such a scenario. The task as a whole consisted of an array of subtasks, designed to test different manipulation abilities. Specifically, the task involved cracking open two large prop eggs and releasing the contents of the eggs into a bowl, removing the top of a "chopped peppers container" (a snack canister) and pouring the contents

into a bowl, flipping the top off of a "salt container" (a bottle of disinfecting wipes) and pouring the contents into a bowl, "mixing egg yolks" by pouring the contents of one cup into another and forth three times, moving the bowl using two hands (if available) from one table to the other table, unscrewing the top off of an "orange juice container" (a laundry detergent bottle) and pouring the contents into a cup, and removing three plates from a drying rack and setting the table to the left of the robot. Participants were asked to complete as many of the subtasks as possible in 10 minutes given the current control condition.

Study 1 Measures & Analyses— To assess participant performance in study 1, we measured *binary success* over the 15 subtasks involved in the breakfast-making task.

To measure participants' perceptions about their robot control experience under various assistance conditions, we administered a questionnaire including eight scales to measure *predictability*, *robot intelligence*, *fluency*, *goal perception*, *trust*, *ease-of-use*, *satisfaction*, and *usefulness*, as well as the NASA-TLX questionnaire. The items and Cronbach's alpha value for each scale are featured in Table 1.

Study 1 Hypotheses— Our central hypothesis in Study 1 was that *bimanual control-assisted* would outperform *bimanual control-non-assisted* on all objective and subjective measures, and *bimanual control-non-assisted* would outperform *single-arm control* on all objective and subjective measures. We believed *bimanual control-assisted* would outperform *bimanual control-non-assisted* because our assistance modes were designed to help with the intricate actions and interactions that occur between the two hands, rather than just considering the bimanual control problem as two separate instances of single-arm control. Further, we believed *bimanual control-non-assisted* would outperform *single-arm control* on all objective and subjective measures because we thought having two hands, even if they do not correspond with each other or provide assistance, would still be beneficial for some of the complex manipulations incorporated in the breakfast-making task.

Study 1 Participants— We recruited 24 volunteers (16 male, 8 female) from the campus of the Naval Research Laboratory in Washington, D.C. Participant ages ranged from 18-60 (M = 35.21, SD = 14.11). Participants reported a moderate familiarity with robots (M = 3.79, SD = 2.02 measured on a seven-point scale). Eight participants had participated in a prior robotics study.

Study 2 Experimental Design and Tasks— Our Study 2 consisted of four separate 4×1 within-subjects experiments. Our goal in these experiments was to assess the relative contribution of each of our bimanual assistance modes (fixed offset, one hand fixed, self-handover, one hand seeking) toward the performance benefits reported in Study 1. For example, we wanted to assess whether our assistance mode designed to help with the self-handover action independently affords performance benefits, only elicits performance gains when used in concert with other assistance modes, or does not contribute to performance benefits at all.

To isolate the effect of each of the assistance types, each of the four experiments in Study 2 consisted of a single task designed to target one of our assistance modes (fixed offset, one hand fixed, self-handover, and one hand seeking). The tasks for each assistance type are as follows:

1. Fixed offset: Participants moved a trash bin from a table in front of the robot to a table to the left of the robot. The trash bin was to be moved just by resting the robot's hands on either side of the bin (the robot's grippers were deactived for this task).

2. One Hand Fixed: Participants opened and closed the top of a disinfecting wipes bottle three times. The bottle had a flip-up top, such that an effective strategy to open and close it involved holding the bottle steadily in place with one hand and using the the robot's other hand to manipulate the bottle.

3. Self-handover: Participants retrieved three plates one at a time off of the table with the robot's right hand, passed the plate from right hand to left hand, and dropped the plate

into a drying rack to the left of the robot.

4. One hand seeking: Participants stacked six cups into a single stack. The cups began organized in two rows of three, placed close enough together such that both hands had to be used throughout the task such that other cups in the grid were not knocked over.

Each of the tasks above were performed with four assistance variations: (1) *All assistance modes off*; (2) *Only the targeted assistance mode engaged*; (3) All assistance modes engaged *except* for the targeted assistance mode engaged (referred to as *Only targeted assistance off*); and (4) *All assistance modes engaged*. Throughout Study 2, the presented order of the four tasks was counterbalanced, and the order of the four assistance variations within those tasks was randomized. Participants had a maximum time of two minutes for each of the sixteen task trials.

Study 2 Measures & Analyses— To assess participant performance in Study 2, we used a compound objective measure that captured both success and timeliness in performing a task. This measure takes the general form $\frac{s}{s_{max}} + (\frac{t_{max}-t}{t_{max}} * \frac{s}{s_{max}})$, where t_{max} is a maximum time allowed for a particular trial, t is the time it took a participant to complete s subtasks in the trial, and s_{max} is the maximum number of subtasks to complete in a trial task. The possible range for this metric is 0 – 2.

To measure participants' perceptions about their robot control experience under various assistance conditions, we administered a questionnaire including the *fluency* and *trust* scales by Hoffman (*56*) found in Table 1.

Study 2 Hypotheses— Our central hypothesis in Study 2 is that *All assistance modes engaged* and *Only targeted assistance mode engaged* would outperform both *Only targeted assistance off* and *All assistance modes off* in all objective and subjective measures. We believed this would be the case because each task was specifically designed to target a particular assistance

Predictability (α = 0.73)

(1) The robot consistently moved in a way that I expected.
(2) The robot's motion was not surprising.
(3) The robot responded to my motion inputs in a predictable way.

Fluency (α = 0.64)

(1) The robot and I worked fluently together as a team.
(2) The robot contributed to the fluency of the interaction.

Trust (α = 0.64)

(1) I trusted the robot to do the right thing at the right time.
(2) The robot was trustworthy.

Goal Perception (α = 0.76)

(1) The robot perceives accurately what my goals are.
(2) The robot does not understand what I am trying to accomplish.
(3) The robot and I are working towards mutually agreed upon goals.

Usefulness (α = 0.75)

(1) It helps me be more effective.
(2) It is useful.
(3) It makes the things I want to accomplish easier to get done.

Robot Intelligence (α = 0.74)

(1) The robot was intelligent.
(2) The robot was able to independently make decisions through the task.
(3) The robot had an understanding of the task.
(4) The robot had an understanding of my goal during the task.

Satisfaction (α = 0. 86)

(1) I am satisfied with it.
(2) I would recommend it to a friend.
(3) It is fun to use.
(4) It works the way I wanted it to work.
(5) It is wonderful.

Ease of Use (α = 0.87)

(1) It is easy to use.
(2) It is simple to use.
(3) It is user friendly.
(4) It is flexible.
(5) It is effortless.
(6) I can use it without written instructions.
(7) I don't notice any inconsistencies as I use it.
(8) Both occasinal and regular users would like it.
(9) I can recover from mistakes quickly and easily.
(10) I can use it successfully every time.

mode; thus, the conditions where the target mode is present, *even* if the other assistance modes are turned off, should outperform the alternatives without the assistance engaged.

Study 2 Participants— We recruited 24 volunteers (13 male, 11 female) from the campus of the Naval Research Laboratory in Washington, D.C. Participant ages ranged 18–64 ($M = 31.92$, $SD = 12.53$). Participants reported a relatively high familiarity with robots ($M = 4.21$, $SD = 1.89$ measured on a seven-point scale). Six participants had participated in a prior robotics study.

Supplementary materials

Supplementary Text

Fig. S1 Motion Dataset Study

Fig. S2 Notation for Technical Details

Movie S1 Shared-Control-Based Bimanual Robot Manipulation

References

1. C. P. Van Schaik, R. O. Deaner, M. Y. Merrill, *Journal of Human Evolution* **36**, 719 (1999).

2. E. A. Franz, *Taking action: Cognitive neuroscience perspectives on intentional acts* pp. 259–88 (2003).

3. S. P. Swinnen, N. Wenderoth, *Trends in cognitive sciences* **8**, 18 (2004).

4. A. Edsinger, C. C. Kemp, *Recent progress in robotics: Viable robotic service to human* (Springer, 2007), pp. 345–355.

5. A. Albers, S. Brudniok, J. Ottnad, C. Sauter, K. Sedchaicharn, *Humanoid Robots, 2006 6th IEEE-RAS International Conference on* (IEEE, 2006), pp. 308–313.

6. C. C. Kemp, A. Edsinger, E. Torres-Jara, *IEEE Robotics & Automation Magazine* **14**, 20 (2007).

7. M. Fuchs, *et al.*, *2009 IEEE International Conference on Robotics and Automation* (IEEE, 2009).

8. C. Smith, *et al.*, *Robotics and Autonomous systems* **60**, 1340 (2012).

9. M. Gienger, M. Toussaint, C. Goerick, *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on* (IEEE, 2008), pp. 2758–2764.

10. Y. Koga, J.-C. Latombe, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation* (IEEE Comput. Soc. Press).

11. Y. Koga, J.-C. Latombe, *Proceedings 1992 IEEE International Conference on Robotics and Automation* (IEEE Comput. Soc. Press).

12. S. M. LaValle, *Planning algorithms* (Cambridge university press, 2006).

13. S. S. Mirrazavi Salehian, N. Figueroa, A. Billard, *The International Journal of Robotics Research* **37**, 1205 (2018).

14. F. Caccavale, P. Chiacchio, S. Chiaverini, *IEEE Transactions on Automatic Control* **44**, 85 (1999).

15. F. Caccavale, P. Chiacchio, S. Chiaverini, *Automatica* **36**, 879 (2000).

16. P. Ögren, C. Smith, Y. Karayiannidis, D. Kragic, *IFAC Proceedings Volumes* **45**, 747 (2012).

17. S. A. Schneider, R. H. Cannon, *IEEE Transactions on Robotics and Automation* **8**, 383 (1992).

18. R. Bonitz, T. C. Hsia, *IEEE Transactions on Robotics and Automation* **12**, 78 (1996).

19. F. Caccavale, P. Chiacchio, A. Marino, L. Villani, *IEEE/ASME Transactions On Mechatronics* **13**, 576 (2008).

20. P. Hsu, *IEEE Transactions on Robotics and Automation* **9**, 400 (1993).

21. T. Yoshikawa, *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on* (IEEE, 2000), vol. 1, pp. 369–376.

22. W. Gueaieb, F. Karray, S. Al-Sharhan, *IEEE/ASME Transactions on Mechatronics* **12**, 109 (2007).

23. R. C. Goertz, R. A. Olsen, W. M. Thompson, Electronic master slave manipulator (1958). US Patent 2,846,084.

24. M. DeLouis, *et al.*, Defense related robotic systems (2015). US Patent 9,144,909.

25. M. Selvaggio, F. Abi-Farraj, C. Pacchierotti, P. R. Giordano, B. Siciliano, *IEEE Robotics and Automation Letters* **3**, 4249 (2018).

26. M. Laghi, *et al.*, *2018 IEEE-RAS 18th International Conference on Humanoid Robots (Humanoids)* (IEEE, 2018).

27. E. Zereik, A. Sorbara, G. Casalino, F. Didot, *Recent Advances in Space Technologies, 2009. RAST'09. 4th International Conference on* (IEEE, 2009), pp. 710–715.

28. D. Rakita, B. Mutlu, M. Gleicher, *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2018), pp. 325–333.

29. D. Nicolis, M. Palumbo, A. M. Zanchettin, P. Rocco, *IEEE Robotics and Automation Letters* **3**, 796 (2018).

30. U. Rokni, O. Steinberg, E. Vaadia, H. Sompolinsky, *Journal of Neuroscience* **23**, 11577 (2003).

31. O. Steinberg, *et al.*, *European Journal of Neuroscience* **15**, 1371 (2002).

32. J. Tanji, K. Okano, K. C. Sato, *Journal of neurophysiology* **60**, 325 (1988).

33. O. Donchin, A. Gribova, O. Steinberg, H. Bergman, E. Vaadia, *Nature* **395**, 274 (1998).

34. P. J. Ifft, S. Shokur, Z. Li, M. A. Lebedev, M. A. Nicolelis, *Science translational medicine* **5**, 210ra154 (2013).

35. A. Yokoi, M. Hirashima, D. Nozaki, *Journal of Neuroscience* **31**, 17058 (2011).

36. D. J. Serrien, P. Brown, *European Journal of Neuroscience* **17**, 1098 (2003).

37. F. G. Andres, *et al.*, *Brain* **122**, 855 (1999).

38. J. Kelso, *American Journal of Physiology-Regulatory, Integrative and Comparative Physiology* **246**, R1000 (1984).

39. J. P. Scholz, *Physical therapy* **70**, 827 (1990).

40. N. Jarrassé, A. T. Ribeiro, A. Sahbani, W. Bachta, A. Roby-Brami, *Journal of neuroengineering and rehabilitation* **11**, 113 (2014).

41. J. Diedrichsen, R. Shadmehr, R. B. Ivry, *Trends in cognitive sciences* **14**, 31 (2010).

42. S. Kantak, S. Jax, G. Wittenberg, *Restorative neurology and neuroscience* **35**, 347 (2017).

43. G. Niemeyer, C. Preusche, G. Hirzinger, *Springer Handbook of Robotics* (Springer, 2008), pp. 741–757.

44. M. Gleicher, *Proceedings of the 25th annual conference on Computer graphics and interactive techniques* (ACM, 1998), pp. 33–42.

45. H. J. Shin, J. Lee, S. Y. Shin, M. Gleicher, *ACM Transactions on Graphics (TOG)* **20**, 67 (2001).

46. D. Rakita, B. Mutlu, M. Gleicher, *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2017), pp. 361–370.

47. D. Rakita, B. Mutlu, M. Gleicher, L. M. Hiatt, *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction* (ACM, 2018), pp. 23–31.

48. J. Lee, *IEEE Computer Graphics and Applications* **28**, 75 (2008).

49. I. Jolliffe, *International encyclopedia of statistical science* (Springer, 2011), pp. 1094–1096.

50. J. Liu, A. Shahroudy, D. Xu, G. Wang, *European Conference on Computer Vision* (Springer, 2016), pp. 816–833.

51. J. Walker, K. Marino, A. Gupta, M. Hebert, *Computer Vision (ICCV), 2017 IEEE International Conference on* (IEEE, 2017), pp. 3352–3361.

52. S. Hochreiter, J. Schmidhuber, *Neural computation* **9**, 1735 (1997).

53. A. Aristidou, J. Lasenby, Y. Chrysanthou, A. Shamir, *Computer Graphics Forum* (Wiley Online Library, 2018), vol. 37, pp. 35–58.

54. D. Rakita, B. Mutlu, M. Gleicher, *Proceedings of Robotics: Science and Systems* (Pittsburgh, Pennsylvania, 2018).

55. N. Mansard, O. Stasse, P. Evrard, A. Kheddar, *2009 International Conference on Advanced Robotics* (IEEE, 2009), pp. 1–6.

56. G. Hoffman, *International conference on human-robot interaction (HRI), workshop on human robot collaboration* (2013), vol. 381, pp. 1–8.

57. J.-C. Latombe, *Robot motion planning*, vol. 124 (Springer Science & Business Media, 2012).

58. J. Kuffner, S. LaValle, *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings* (IEEE).

59. L. Kavraki, P. Svestka, M. H. Overmars, *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*, vol. 1994 (Unknown Publisher, 1994).

60. D. Damen, *et al.*, *arXiv preprint arXiv:1804.02748* (2018).

Acknowledgments: We thank Bill Adams, Magdalena Bugajska, and Greg Trafton for their valuable help and feedback throughout the development of this work. The views and conclusions contained in this paper do not represent the official policies of the U.S. Navy. **Funding**: Funding for this work was provided by the Office of Naval Research through an award to Laura M. Hiatt. This work was also supported in part by the National Science Foundation under award 1830242 and the University of Wisconsin–Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation. **Author Contributions**: D.R. investigated the presented bimanual shared-control approach, implemented the proposed methods, ran the evaluations, and led the writing of the paper. B.M. assisted with concept formation, experimental design, and writing of the paper. M.G. assisted with concept formation, technical formulations, and writing of the paper. L.M.H. assisted with concept formation, experimental design, overseeing the physical setup of the prototype system, technical formulations, and writing of the paper. **Competing Interests**: The authors declare that they have no competing interests. **Data and materials availability**: All data needed to evaluate the conclusions in the paper are present in the paper or the Supplementary Materials. Contact D. Rakita for more information.

5 Supplemental Materials

5.1 Related Works

Our bimanual shared-control methods draws on prior work, especially in controls, motion planning, and teleoperation interfaces. In this section, we discuss pertinent previous works and highlight how our current work is related (for a comprehensive review of multi-arm robotics work, see Smith et al. (*8*)). A review of this body of work indicates that there are two main differences between our current work and the discussed prior works. First, the two robot arms in some prior bimanual manipulation methods exhibit independent behavior with only limited coordination. For example, these works may feature two robot arms working in close proximity, but they do not plan complex actions together such as manipulating objects using both arms. Second, other bimanual manipulation works mainly present single-function bimanual abilities, such as only being able to grasp and stabilize an object with two hands without accommodating other bimanual skills such as passing objects from one hand to the other. Thus, these prior works only need to demonstrate the efficacy of their independent solution without showcasing how it fits within to the wider context of bimanual manipulation abilities. In contrast, our work is focused on robots performing dexterous coordinations between the two arms inspired by how people perform bimanual manipulations, as well as endowing robots with a wide range of bimanual skills and the ability to fluently switch between these various control strategies based on inferred task context.

Controls— Many previous control algorithms have been proposed for bimanual robot platforms. These control algorithms fall into three main categories: *kinematic controllers*, where the output is joint-angle positions, *force controllers*, where the outputs drive the joint torques of the robot, and *hybrid controllers*, where the outputs may be joint-angle positions or joint torques subject to the current situation.

Mirrazavi et al. (*13*) present a method that allows two arms to move on-the-fly in order to intercept a large object being handed over by a human collaborator. Our current work is inspired by the dexterous, on-the-fly coordination between two arms in order to manipulate objects, though our focus is on the shared-control aspects of switching between control strategies based on inferred bimanual action contexts as opposed to the singular task of stabilizing and intercepting large objects. The method from Mirrazavi et al. also features an optimization-based inverse kinematics solver such that the two arms do not collide with each other. Our method also uses an optimization-based inverse kinematics solver that is able to avoid self-collisions, though instead of approximating the geometry of the robot with spheres and encouraging these spheres to stay far apart, our method uses a data-driven approach that considers the full geometry of the robot *a priori* such that the robot can learn how close or far a given configuration is to exhibiting a collision. Then, a feed-forward pass through the trained neural network that calculates a distance to a collision state is two orders of magnitude faster than the original distance calculation function, making this sufficiently fast enough to serve as a single term in a real-time motion optimization framework.

Work by Schneider et al. (*17*) present an impedance controller for multi-arm systems that considers the impedance of the object being carried by the multiple arms, as opposed to the impedance experienced at each of the arms' end-effector points. Considering how the carried object's forces should behave with respect to the environment would be important for tasks such as assembly of large parts transported by multiple arms. Bonitz and Hsia (*18*) present an impedance controller for multi-arm systems that instead considers the internal forces of each of the manipulators. Subsequent work by Caccavale et al. (*19*) presents a controller that is able to consider *both* the impedance of the carried objects as well as the internal forces of each of the manipulators carrying the object. This combined approach has the advantage that the internal forces exerted in each manipulator will not over-accumulate stress, while the risk

of object slip occurring between the two end-effectors is also reduced. Gueaieb et al. (*22*) introduce a hybrid control method that is able to switch control strategies between position and force-based control based on model uncertainties and disturbances from the environment. Our method does not consider object or internal dynamics when manipulating objects, as we only drive joint-positions when controlling the bimanual platform. A main focus of our work is to motivate the need to *switch* bimanual control types based on the bimanual action currently happening. We believe that potential extensions of our work would benefit from considering object and internal dynamic when switching control types based on context. For example, when our method recognizes that an object is being carried with both arms and our fixed-offset assistance controller is activated, we only fix the kinematic (translation and rotation) offset between the robot's end-effectors. However, switching to an impedance-based controller such as those mentioned above could reduce slip and adapt based on the forces exerted on the carried object, and may be even more successful in conjunction with the rest of our proposed approach.

Motion Planning— The problem statement for motion planning is generally to find a feasible path from an initial state q_{init} to a goal state q_{goal}, subject to a set of constraints (for a full review on motion planning, see Latombe (*57*)). Motion planning problems are often posed as graph search problems, such that robot configurations serve as nodes in the graph and edges define feasible, traversable motion in configuration-space (*58, 59*). While many bimanual motion planning problems can be subsumed by general motion planning problems by increasing the state-space to include the degrees-of-freedom from both arms, specialized methods have been developed that explicitly handle the coordination problems unique to multi-arm motion planning.

Koga and Latombe present a motion planning solution that addresses the object transport problem using two robot arms, i.e., the problem of moving an object from one location to another in task space using two cooperating arms (*11*). In this work, the robots are able to

avoid collisions and can handle re-grasps of the objects to better serve the motion planning. The authors extended this work to include more than two manipulators (*10*). LaValle also presents various strategies for multiple-robot motion planning in his comprehensive overview of planning algorithms (*12*). Our current work is inspired by the coordination between multiple arms exhibited by these motion planners, though our focus is on real-time control without any look-ahead information about a complete path. Thus, our work utilizes a real-time motion optimization framework where the robot is able to adapt and react in a per-update fashion to current needs.

Teleoperation Interfaces— Much prior work has investigated effective ways for users to control robot manipulators. While a lot of these prior works have focused on single arm control, bimanual robot control presents its own set of challenges, and thus has facilitated its own set of solutions in the literature. Many works in teleoperation draw on the influential mechanical systems constructed by Raymond Goertz in the 1950's (*23*). These systems often featured two robot arms to manipulate radioactive material in an adjacent, secured room. While interfaces were developed such that a single user could control both arms, these early systems often involved two people controlling the arms separately. Our current work is inspired by this early seminal work by Goertz. Our goal is to extend the dexterity exhibited by bimanual robot platforms in this early work by using shared-control aspects as well as a more thorough understanding of how people use two hands to complete tasks.

A recently proposed teleoperation interface from the company RE2 robotics features a bimanual setup where the robot's arms are controlled by a specialized controller (*24*). The controller is a mini replica of the controlled robot platform, and the motion of the controlled device is determined through joint-to-joint remapping from the replica's motion. The replica controller is large enough such that the operator can rest his/her arms on them, thus giving the illusion that the robot's arms are imitating the motion of the operator's arms. This system has been shown

to be effective at allowing users to perform dexterous bimanual manipulations. While the joint-to-joint mapping control strategy leads to reliable and predictable motion from the controlled device, this type of control only works for the robot platform that the controllers were designed for, making this method infeasible to generalize to any bimanual robot platform. While we would have been interested to compare our current methods to this system in our evaluation, the specialization of the hardware needed for this setup made such a comparison impractical. However, we believe that the methods presented throughout our work, especially the shared-control aspects that are inspired by how people perform bimanual actions, would integrate well with the promising system presented by RE2.

Recent work by Laghi et al. (*26*) present a shared-control strategy for bimanual robot platforms. The method allows users to control both arms independently, and users are also able to manually switch to a mode where the arms exhibit a fixed offset when carrying something together. Our current work also proposes methods that add shared-control elements to a bimanual control system, but our approach seeks to *automatically* switch between bimanual control modes based on the inferred action context. We also present more bimanual control strategies to dynamically switch between, all of which are inspired by how people perform bimanual actions through our kinematic pattern analysis.

5.2 Motion Dataset Collection Details

We recruited twelve participants (7 males, 5 females) from the University of Wisconsin–Madison campus with ages 18–22 (M = 20.67, SD = 1.23). Eleven participants were right-hand dominant, and one participant was left-hand dominant. The study took 25 minutes, and each participant received $5 USD. The motion dataset study was approved by the Institutional Review Board at the University of Wisconsin–Madison.

Participants completed a set of twenty tasks, designed to simulate a meal preparation sce-

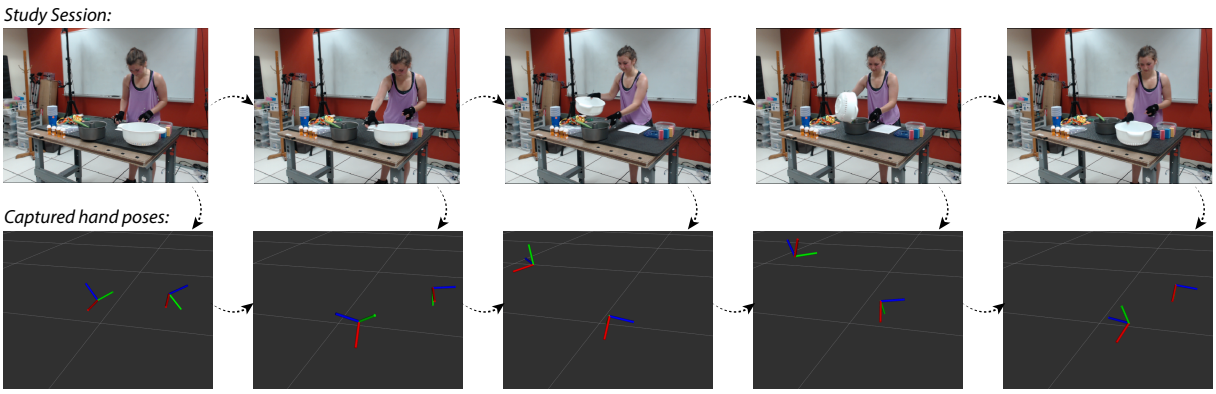

Figure 6: (Top) We collected a dataset by recording participants performing various bimanual manipulations, (bottom) in order to capture the position and rotation of the participants' hands throughout the tasks. Using this dataset, our goal was to distill the space of bimanual manipulations down to just a few bimanual actions that still span the bimanual manipulation space.

nario in a kitchen. In order to select a representative set of two-handed tasks that people would typically execute in their day-to-day lives, we designed the tasks using the object and action labels reported on in the Epic-Kitchens Dataset (*60*). The Epic-Kitchens Dataset consists of 55 hours of labeled, egocentric-view videos of people performing tasks around their own kitchens. Our tasks were constructed by combining the top ten action reported on in the dataset (put, take, wash, open, close, cut, mix, pour, remove, turn-on, turn-off) with common object labels reported on in the dataset (breads, fruits, meats, vegetables, appliances, kitchenware, etc.), ensuring that our dataset would consist of manipulation actions that commonly occur in peoples' everyday lives.

We recorded participant hand data using an Optitrack motion capture system. Our data consists of over 500,000 discrete hand poses from 111 minutes of bimanual manipulations sampled at 80 *Hz*. Our whole dataset is structured as:

5.3 Motion Datset Filtering Details

In this section, we provide more details on the filtering process used on our bimanual dataset. A bilateral filter preserves the signal's shape around extremal points, meaning rich motion information found in higher derivatives such as velocities or accelerations will remain intact; only the perturbations that do not match the motion signature in the signal's surrounding neighborhood are smoothed. In contrast, a more standard low-pass filter averages the whole curve based on a constant kernel, which will constrict the whole signal and cause it to move away from extremal points.

We pre-process the dataset using the following procedure:

$$P := \{\ (\phi(\mathbf{p}_t^r),\ \phi(\mathbf{p}_t^l),\ exp(\phi(log(\mathbf{q}_t^r))),\ exp(\phi(log(\mathbf{q}_t^l)))\ \},\ \ \forall t \tag{3}$$

$$\phi(\mathbf{x}_t) = \frac{1}{Z} \sum_{\mathbf{x}_v s.t. v \in [t-\Theta, t+\Theta]} x_v\ \gamma_r(||x_v - x_t||)\ \gamma_s(|v-t|) \tag{4}$$

$$Z = \sum_{\mathbf{x}_v s.t. v \in [t-\Theta, t+\Theta]} \gamma_r(||x_v - x_t||)\ \gamma_s(|v-t|) \tag{5}$$

$$\gamma_{\{r,s\}}(x) = \frac{1}{\sigma_{\{r,s\}}\sqrt{2\pi}}\ exp(-0.5 * [\frac{x}{\sigma_{\{r,s\}}}]^2) \tag{6}$$

Here, log and exp are the standard log-map and exponential-map operators for quaternions. The log-map operator maps a quaternion $\mathbf{q} \in \mathbb{S}^3$ to a rotation vector $\mathbf{r} \in \mathbb{R}^3$. The exponential-map exponentiates a rotation vector back to a corresponding quaternion. For more details on representing rotations and orientations, see work by Lee (*48*). The $\gamma_s(.)$ and $\gamma_r(.)$ functions are Gaussians that define the *spatial* and *range* falloff of the filter, respectively. Note that we do not filter across the boundaries between separate participants. Each participant's data is filtered separately and concatenated together into the dataset.

5.4 Bimanual Action Inference Implementation Details

The neural network in our inference step reasons about the same kinematic features seen in our previously discussed kinematic pattern analysis. We decided not to use the PCA components specified in our kinematic pattern analysis in our inference step, and instead just utilized the same kinematic features that were used to perform the PCA. We made this decision because the windowing method used in the PCA may present a temporal ambiguity. In other words, the RNN could be confused if the same bimanual action appears shifted over in the window. Using the core kinematic features instead of the PCA components allows the RNN to learn the temporal characteristics of the data itself, instead of imposing this temporal structure as filtered through the PCA process. A single input into the neural network is structured as:

$$\begin{bmatrix} \Phi_{t-\omega} \\ \Phi_{t-\omega+1} \\ \vdots \\ \Phi_t \end{bmatrix} \in \mathbb{R}^{\omega \times 6}, \quad \Phi_t = \begin{bmatrix} ||\mathbf{p}_t^d - \mathbf{p}_t^n|| \\ ||\mathbf{p}_t^d - \mathbf{p}_t^n|| - ||\mathbf{p}_{t-1}^d - \mathbf{p}_{t-1}^n|| \\ ||\mathbf{p}_t^d - \mathbf{p}_{t-1}^d|| \\ ||disp(\mathbf{q}_t^d, \mathbf{q}_{t-1}^d)|| \\ ||\mathbf{p}_t^n - \mathbf{p}_{t-1}^n|| \\ ||disp(\mathbf{q}_t^n, \mathbf{q}_{t-1}^n)|| \end{bmatrix}^T \tag{7}$$

The labeled outputs are a tall matrix of stacked one hot vectors, where each row only contains one value that is one and all other inputs are zero. When labeled over the whole window, these outputs denote which bimanual action is happening at the time point corresponding to that same row in the input kinematic features. The label matrix corresponding to each label instance is of size $\omega \times 7$, where columns correspond to one sub-action in the *bimanual action vocabulary*. Specifically, the columns correspond to the actions of fixed offset, right hand fixed, left hand fixed, self handover, right hand seeking, left hand seeking, and a null action that signifies none of the aforementioned actions are present. Thus, for example, a row of $[1, 0, 0, 0, 0, 0, 0]$ in the output matrix would mean that a fixed offset bimanual action is happening, given the input kinematic features Φ_t at time t.

We take advantage of the temporal structure in this data stream by using a recurrent neural network, specifically using a single long short-term memory (LSTM) layer (*52*), with an output dimensionality of 30. We trained the neural network using the Adam optimizer with a mean squared error loss. A data set of over 30,000 input and output pairs was used to train the model, which was collected by having three volunteers (colleagues of the authors at the Naval Research Laboratory) use the robot control system to complete a set of manipulations. Output bimanual action classes were hand labeled by the authors.

The neural network outputs a sequence of probabilities that each action is the current one over a certain window of time. To distill this sequence of predictions down to a single classification at the given time t, we take a plurality vote of the estimates for each class throughout the whole window and classify based on the maximum sum value. Taking this plurality vote over the window helps avoid the network being too reactive to small perturbations or noise in the motion capture data; only when the network builds up enough consistent confidence over a window of time will it decide that a new bimanual action is present. Using the overall learned model and this voting strategy, the network achieves classification accuracy of 94.3% on 5,000 unseen test-set data.

Our overall system was designed such that the presence of some misclassifications of bimanual actions by the neural network generally do not hinder the control experience. The controllers gracefully switch between one another such that an abrupt misclassification does not disrupt the control process. Also, because our underlying optimization-based inverse kinematics solver leads to differential motion increments and strongly enforces smooth velocities, acceleration, and jerks, the control process remains smooth over time regardless of the underlying motion strategy.

5.5 System Implementation Details

We realized our method in a system designed to provide sufficient performance and safety in order to demonstrate its benefits in a user study. The paragraphs below describe details of our prototype system.

To capture the user's hand configuration at each system update, we used a Vicon motion-capture system involving eight cameras placed in a semi-circle around the workspace. The users wore a velcro glove on their right hand with five passive markers secured to it. The marker pattern was recognized as a rigid body, serving as a proxy for the user's hand configuration. The Vicon system provided hand-transform data at a rate of 100 Hz. We expect any motion capture system that can track positions and rotations of the user's hand to support our method.

The robot platform used in our prototype system was a DRC-Hubo+ humanoid robot from Rainbow Robotics. We controlled the right and left arm of the robot, as well as the revolute waist joint, resulting in 15-DOF total. The robot platform has an anthropomorphic design and includes a three-finger gripper on both hands. The user opened and closed the grippers using a remote held in each hand.

System Overview—Our prototype system was set up as a distributed system over a network of computers that utilized ROS for communication. The motion-capture system sent transform information to a computer that solved the 15-DOF RelaxedIK nonlinear optimization problem, recurrent neural network pass for bimanual action inference, and end-effector goal location optimization at each update. This control loop as a whole ran at approximately 30 Hz on a computer with an Intel i7-8700k 3.7 GHz processor, 32GB RAM, and Nvidia GTX 1080Ti graphics card.

5.6 RelaxedIK Overview

In this section, we provide a brief overview of RelaxedIK, the optimization-based IK solver used in our motion retargeting step. For full details, refer to our prior work (*54*).

The IK problem in RelaxedIK is formulated as a constrained optimization:

$$\begin{aligned} \Theta = \arg\min_{\Theta} \ \ & \mathbf{f}(\Theta) \text{ s.t. } \mathbf{c}_i(\Theta) \ \geq \ \mathbf{0}, \ \ \mathbf{c}_e(\Theta) \ = \ \mathbf{0} \\ & l_i \leq \Theta_i \leq u_i, \forall i \end{aligned} \tag{8}$$

where Θ is the $n-$vector of robot joint values (n is the robot degrees of freedom); $\mathbf{c}_i(\Theta)$ is a set of inequality constraints; $\mathbf{c}_e(\Theta)$ is a set of equality constraints; l_i and u_i values define the upper and lower bounds for the robot's joints; and $\mathbf{f}$ is a scalar objective function. Our challenge is to encode our bimanual-motion assistance modes as goals within the constraints and objectives.

The objective function is expressed as a weighted sum of individual goals, such as end-effector position matching, end-effector orientation matching, minimum jerk joint motion, and distance to singularity. It is formalized it as follows:

$$\mathbf{f}(\Theta) = \sum_{i=1}^{k} w_i \ * h_i(\Theta, v(t)) \ * f_i(\Theta, \Omega_i) \tag{9}$$

Here, w_i is a static weight value for each term, which allows the programmer to incorporate prior knowledge about what terms are most important for a given task. The $h_i(\Theta, v(t))$ represents a dynamic weighting function, that can depend on the current robot configuration, Θ, or other time-varying values in the function $v(t)$. Finally, $f_i(\Theta, \Omega_i)$ is an objective-term function that encodes a single sub-goal, with Ω_i being model parameters used to construct a particular loss function.

To facilitate combining objectives, Rakita et al. normalize each term using a parametric normalization function that is designed to scale each function to a uniform range (*54*). This function places a narrow "groove" around the goal values, a more gradual falloff away from the

groove in order to better integrate with other objectives, and exhibits a consistent gradient that points towards the goal. The normalization function is encoded as a Gaussian surrounded by a more gradual polynomial:

$$f_i(\Theta, \Omega_i) = (-1)^l exp(\frac{-(\chi_i(\Theta) - s)^2}{2c^2}) \ + \ r * (\chi_i(\Theta) - s)^4 \tag{10}$$

Here, the scalar values n, s, c, r form the set of model parameters Ω. Together, they shape the loss function to express the needs of a certain term. Here, $n \in \{0, 1\}$, which dictates whether the Gaussian is positive or negative. Negative Gaussian regions are areas of high "reward," while the optimization will push away from positive regions of high "cost." The value s shifts the function horizontally, and c adjusts the spread of the Gaussian region. The r value adjusts the transition between the polynomial and Gaussian regions, higher values showing a steeper funneling into the Gaussian region and lower values flattening out the boundaries beyond the Gaussian. The scalar function $\chi(\Theta(t))$ assigns a numerical value to the current robot configuration that will serve as input to the loss function. In the sections below, we will tune existing objectives or add new objectives $\chi(\Theta(t))$ to tailor RelaxedIK for particular bimanual actions.

The groove loss function above has various parameters that need to be specified. A wide range of parameters lead to successful performance using this solver. Parameter tuning is not a key part of our bimanual control process as the normalization procedure through the custom groove loss function encourages the optimization to exhibit expected motions for a large range of reasonable parameter values. These values were determined empirically in our prototype system and reported on below, and an extension to this work will automatically determine these parameters based on higher-level motion properties.

The full optimization formulation is comprised of seven core objective terms and two constraints. The objective terms encode the following kinematic goals: (1) End-effector position matching; (2) end-effector orientation matching; (3) minimized joint velocity; (4) minimized

joint acceleration; (5) minimized joint jerk; (6) minimized end-effector translational velocity; (7) and self-collision avoidance. The two constraints are designed to clamp joint velocities at each update and avoid kinematic singularities, respectively. These objectives and constraints are detailed in our prior work (*54*).

5.7 Bimanual Assistance Mode Mathematical Details

In this section, we provide mathematical details for all of our bimanual assistance modes, each corresponding to an action in our bimanual action vocabulary. Our assistance modes all adjust the RelaxedIK optimization-based inverse kinematics solver by adding objective terms or adjusting weighting parameters. Common notation used throughout this section can be found in Figure 7.

5.7.1 Fixed Offset

A central aspect of our fixed offset assistance mode is inferring a *coordinate frame* serving as a proxy for the object being manipulated between the two hands, denoted as $\mathbf{F}_t$. We estimate this pose at each update based on the poses of the robot's end-effector pose goals at each update, as dictated by the user's hand motion. To illustrate, suppose that the fixed offset bimanual action is detected to be occuring at time t and the robot's right and left end-effector goal positions in world coordinates at time t are $p[\mathbf{G}_t^r]$ and $p[\mathbf{G}_t^l]$, respectively. First, we assign the origin of the coordinate frame serving as the object proxy, denoted as $\mathbf{c}_t$, as the center point between the robot's end-effectors: $\mathbf{c}_t = 0.5 * p[\mathbf{G}_t^r] + 0.5 * p[\mathbf{G}_t^l)$. Next, we begin to construct the coordinate frame serving as the object proxy by first calling the vector that points from the robot's left end-effector to right end-effector the local y-axis: $\hat{y}_t = n(r(\mathbf{G}_t^l] - p[\mathbf{G}_t^r])$. Here, $n(.)$ denotes vector normalization. The rest of the coordinate frame can be constructed using cross products: $\hat{x}_t = \hat{y}_t \times [0, 0, 1]^T; \quad \hat{z}_t = \hat{x}_t \times \hat{y}_t$. This procedure to construct the proxy object coordinate

Notation	Explanation
$EE_0^{\{l,r\}}$	Initial end-effector pose in world coordinate system
$EE_t^{\{l,r\}}$	End-effector pose at time t in world coordinate system
$r(EE_t^{\{l,r\}})$	End-effector pose at time t, relative to initial robot pose
$H_0^{\{l,r\}}$	Initial user hand pose in world coordinate system
$H_t^{\{l,r\}}$	User hand pose at time t in world coordinate system
$r(H_t^{\{l,r\}})$	User hand pose at time t, relative to user's initial pose
$FK_t^{\{l,r\}}(\theta)$	Pose of end-effectors given θ (forward kinematics model)
$p[.]$	Extracts the position component of the input pose
$o[.]$	Extracts the orientation component of the input pose
d_t	Distance between the user's hands at time t
$G_t^{\{l,r\}}$	End-effector pose goals at time t in world coordinate system
$r(G_t^{\{l,r\}})$	End-effector pose goals at time t, relative to user's initial pose
$A_t^{\{l,r\}}$	Optimized pose goals at time t in world coordinate system
$r(A_t^{\{l,r\}})$	Optimized pose goals at time t, relative to user's initial pose
c_f	Position of the proxy object pose at start of fixed offset
c_t	Position of the proxy object pose at time t
F_f	Pose of the fixed offset proxy object at start of fixed offset
F_t	Pose of the fixed offset proxy object at time t
$\hat{x}_t, \hat{y}_t, \hat{z}_t$	x, y, and z components of the F coordinate frame at time t

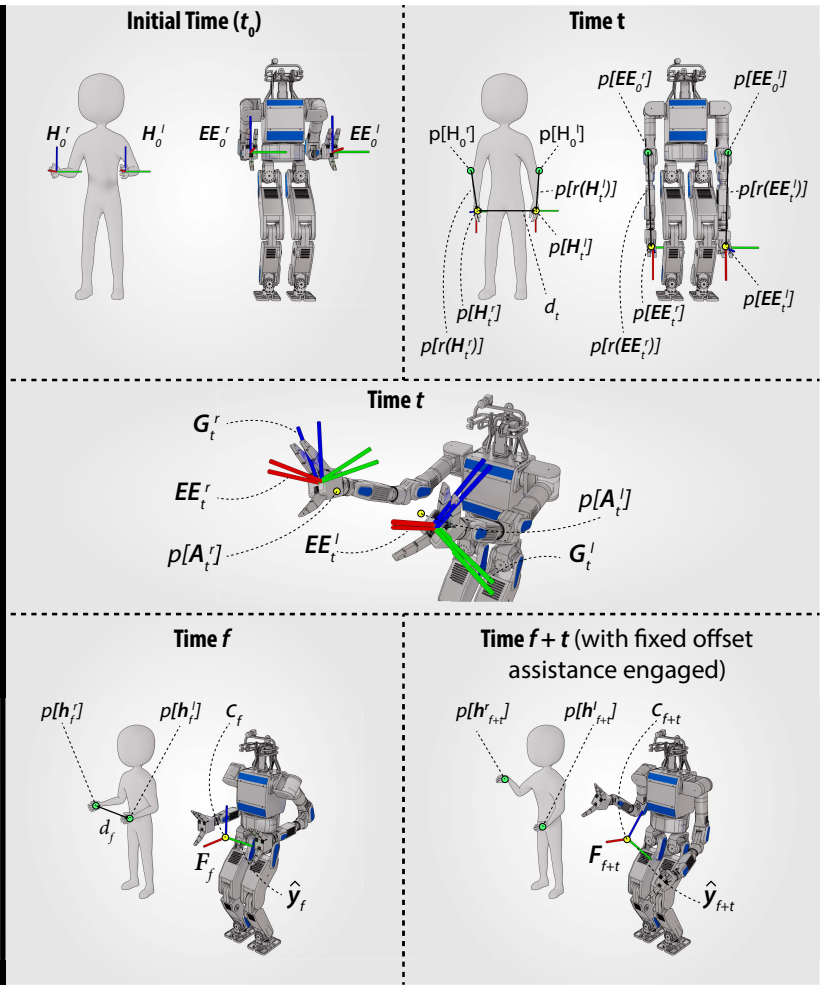

Figure 7: Common notation used throughout the technical details sections.

frame $\mathbf{F}_t$ repeats for the all timesteps t where the fixed offset bimanual action is still detected to be active.

Throughout the fixed offset action, the pertinent aspects of the hand's motions are that they maintain a fixed position and rotation offset while jointly moving the object held between the hands. Thus, while the fixed offset assistance is engaged, we remove the standard end-effector position matching and end-effector orientation matching objectives in RelaxedIK because matching independent pose goals of the separate end-effectors is not paramount when moving an object with two hands. Instead, we replace these objectives with three objective

terms to the weighted-sum objective function that are specifically tailored for the fixed offset action:

$$\chi_{f1}(\Theta) = || \, p[FK(\Theta)_t^r] - [c_t - (0.5 * d_f)\hat{y}_t] \, ||^2 + \\ ||p[FK(\Theta)_t^l] - [c_t + (0.5 * d_f)\hat{y}_t]||^2 \tag{11}$$

$$\chi_{f2}(\Theta) = (|| \, p[FK(\Theta)_t^r] - p[FK(\Theta)_t^l] \, || - d_f)^2 \tag{12}$$

$$\chi_{f3}(\Theta) = || \, disp(o[\mathbf{F}_t], o[FK(\Theta)_t^r]) - disp(o[\mathbf{F}_f], o[\mathbf{G}_f^r]) \, ||^2 + \\ || \, disp(o[\mathbf{F}_t], o[FK(\Theta)_t^l]) - disp(o[\mathbf{F}_f], o[\mathbf{G}_f^l]) \, ||^2 \tag{13}$$

Here, $\chi_{f1}(\Theta)$ places the end-effectors on either side of the object proxy, $\chi_{f2}(\Theta)$ keeps the end-effectors at the same distance as was seen when the fixed offset action was initialized, and $\chi_{f3}(\Theta)$ keeps the end-effector orientations the same rotational displacement from the proxy object as was seen when the fixed offset action was initialized. Also, as discussed above, $disp$ is the standard displacement operator for quaternions: $disp(\mathbf{q}_1, \mathbf{q}_2) = log(\mathbf{q}^-1_1 * \mathbf{q}_2)$ (*48*). In our prototype implementation, we placed very high weights on our fixed offset objectives ($w_{f1} = 150.0$, $w_{f2} = 1000.0$, $w_{f3} = 200.0$). With weights this high, the motion qualities defined by these objectives resemble constraints in the RelaxedIK framework.

5.7.2 One Hand Fixed

To assist with the one hand fixed bimanual action, we tune the control optimization such that the robot's end-effector corresponding to the user's static hand is discouraged from moving. This leave the static hand in place as best as possible such that the other hand can perform the manipulation with respect to that static hand deftly and precisely. To achieve this effect, we adjust the weight on the objective in the RelaxedIK framework that encourages minimized velocity in the robot's end-effector position space when the one hand fixed action is detected to

be happening:

$$\chi_e(\Theta) = ||F\dot{K}(\Theta)||_2 \tag{14}$$

Specifically, we change $w_e = 1.0$ to be $w_e = 5.0$ on the term associated with the static end-effector.

5.7.3 Self-handover

To assist with the self-handover bimanual action, we make two adjustments to the RelaxedIK optimization. First, we gradually ramp down the velocity in the robot's end-effector position space for *both* arms. This is similar to the process used for *one hand fixed* assistance, but the weights are adjusted for both hands. The reasoning behind this adjustment is to match the hand translation velocity profile highlighted in principal component 3 from our kinematic pattern analysis. We gradually change $w_e = 1.0$ to $w_e = 2.0$ using a linear ramp once the self-handover action is inferred. The linear ramp slides from $w_e = 1.0$ to $w_e = 2.0$ over the time-span of a second, matching the time window used in our kinematic pattern analysis.

The second assistance we incorporate for the self-handover action involves making sure that the correspondence between the user's hands and the robot's end-effectors match when the user's hands are in close proximity.

Prior work shows that control inputs relative to a user's starting pose is effective for a *mimicry-control* interface (*46*). However, because the robot may have a different scale and geometry compared to the user, relative control with two separate arms may result in the self-handover point for the robot not matching the self-handover point between the user's hands. To illustrate, the self-handover point for the robot may involve the user keeping their arms far apart, or may even involve the user crossing their arms in order to get the robot's hands close enough.

Our idea with this assistance is to use *relative* control when the arms are far apart, leading to the natural benefits seen in prior work when the hands are not in close proximity, and dynamically adapting such that the *absolute* distance between the hands takes precedence when the user's hands are close together. However, because these are conflicting goals, we solve separate optimization problem at each update to calculate candidate right and left end-effector goal poses that try to reconcile both of these objectives, denoted as $\mathbf{A}_t^r$ and $\mathbf{A}_t^l$, then calculate final goal poses, $\mathbf{G}_t^r$ and $\mathbf{G}_t^l$, by arbitrating between the relative hand poses, $r(\mathbf{H}_t^r)$ and $r(\mathbf{H}_t^l)$ and optimized goal poses depending on the distance between the user's hands. This procedure is framed as an unconstrained, non-linear programming problem with four objective terms, with the objective function defined as follows:

$$p[\mathbf{A}_t^r]^*, p[\mathbf{A}_t^l]^* = \underset{p[\mathbf{A}_t^r], p[\mathbf{A}_t^l]}{\arg\min} \sum_{i=1}^{4} \omega_i * \chi_i(p[\mathbf{A}_t^r], p[\mathbf{A}_t^l], \Omega) \tag{15}$$

$$\chi_1(p[\mathbf{A}_t^r], p[\mathbf{A}_t^l], \Omega) = (\, ||p[\mathbf{A}_t^r] - p[\mathbf{A}_t^l]|| - d_t \,)^2 \tag{16}$$

$$\begin{aligned} \chi_2(p[\mathbf{A}_t^r], p[\mathbf{A}_t^l], \Omega) = (\, p[r(\mathbf{H}_t^r)] - [(p[r(\mathbf{A}_t^r)] \,)^2 + \\ (\, p[r(\mathbf{H}_t^l)] - [(p[r(\mathbf{A}_t^l)] \,)^2 \end{aligned} \tag{17}$$

$$\chi_3(p[\mathbf{A}_t^r], p[\mathbf{A}_t^l], \Omega) = (\, ||p[\mathbf{H}_t^r] - p[\mathbf{A}_t^r]|| - ||p[\mathbf{H}_t^l] - p[\mathbf{A}_t^l]|| \,)^2 \tag{18}$$

$$\chi_4(p[\mathbf{A}_t^r], p[\mathbf{A}_t^l], \Omega) = || \, p[\dot{\mathbf{A}}_t^r] \, || + || \, p[\dot{\mathbf{A}}_t^l] \, || \tag{19}$$

Here, χ_1 is encouraging the candidate positions to match the distance between the robot's end-effectors with the absolute distance between the user's hands, χ_2 is encouraging the candidate positions to remain close to the user's relative hand position goals such that the results do not stray too far, χ_3 encourages the candidate positions to be equidistant from the respective

relative hand positions, and χ_4 minimizes Cartesian space velocity on the optimized candidate points such that the output positions move smoothly. In our prototype system, we used weight values of $\omega_1 = 12.0$, $\omega_2 = 1.0$, $\omega_3 = 0.5$, and $\omega_4 = 5.0$.

Once optimized candidate goal positions $p[\mathbf{A}_t^r]^*$ and $p[\mathbf{A}_t^l]^*$ have been calculated at the current update, the method must decide whether to use these candidate positions, or rather move in favor of the relative hand position inputs from the user $p[r(\mathbf{H}_t^r)]$ and $p[r(\mathbf{H}_t^l)]$. As discussed above, our premise here is that the method should adhere to the relative hand inputs when the user's hands are far apart, and favor the optimized goal positions $p[\mathbf{A}_t^r]^*$ and $p[\mathbf{A}_t^l]^*$ that match the absolute distances between the user's hands when the user's hands are close together. To achieve this goal, we use a falloff function to continuously slide between the goal points based on the distance between the user's hands:

$$p[\mathbf{G}_t^{\{r,l\}}] = (1 - \lambda^{0.5}) * p[\mathbf{A}_t^{\{r,l\}}]^* + \lambda^{0.5} * p[\mathbf{H}_t^{\{r,l\}}] \tag{20}$$

Here, λ is a value in the range $[0, 1]$, such that $\lambda = 0$ when the user's hands are touching, and $\lambda = 1$ when the user's hands are at (or exceeding) some maximum calibrated distance.

5.7.4 One Hand Seeking

When it is detected that one hand is seeking out an object, we assist by *lowering* the importance of precisely matching the position and rotation pose goal on the *other* end-effector. This allows the optimization to put more attention in precisely matching the position and rotation pose goals on the end-effector that is seeking, providing more fine and dexterous manipulations to be done with this hand.

The optimization has objectives $\chi_{pr}(\Theta)$ and $\chi_{pl}(\Theta)$, which encourages the position of the

robot's end-effectors to match given position goals. These both take the form of:

$$\chi_p(\Theta) = ||FK(\Theta) - \mathbf{p}_g||_2 \tag{21}$$

Here, $\mathbf{p}_g$ is the provided position goal, and $FK(\Theta)$ signifies the end-effector position given joint angles Θ, calculated by the robots forward kinematics model. Our method lowers either weight w_{pr} if the left hand is seeking or w_{pl} if the right hand is seeking. The weight is lowered such that the weight of the seeking hand's position matching is double the weight of the hand that is currently not pertinent.

5.8 Discussion of Study Findings

Our results in Study 1 show that our bimanual shared-control method significantly outperforms the alternative approaches in terms of task success. Our method was also perceived as less mentally demanding, more integrated with task performance, more intelligent, more fluent, more able to perceive the users' goals, more satisfying, more useful, and more predictable. These results suggest the promise of using our bimanual shared-control control system to complete various tasks.

We believe our bimanual shared-control method outperformed the alternative control approaches because our method aligns better with how *people* perform actions in human-centered environments. For example, we would expect our method to perform more favorably than a single-arm control approach because many tasks in human-centered environments benefit from having two arms. To illustrate, in our study, it was difficult for participants to unscrew the cap off of the orange juice container with only a single-arm, whereas there was a clear strategy of holding the container with one hand and unscrewing the cap with the other hand when two arms were available. Further, our bimanual shared-control method is able to *dynamically switch* between *centralized bimanual actions* as specificed in our bimanual action vocabulary,

aligning well with leading theories that describe how the brain interprets and executes bimanual actions (*3, 34, 35, 41*). Our results suggest that the more *biologically supported* way of arbitrating between different, predefined bimanual actions in robot control elicits performance and user perception benefits compared to a method where the two arms are simply two concurrent instances of single-arm control with minimal coordination, as is often done in prior bimanual manipulation and control approaches. Our results in Study 1 indicate that our method achieves more of the bimanual "gestalt" phenomenon described by Swinnen et al. (*3*), where the individual motions of each arm are promoted to achieve more than the sum of their parts, compared to alternative approaches.

Our results in Study 2 indicate that our assistance modes *do* have an effect on performance in each sub-task. We see that "all assistance on" significantly outperforms "all assistance off" in terms of task success across all tasks, which often elicited a feeling of more fluency and trust in the robot. However, the results for the effectiveness of each individual assistance mode differ across the various tasks.

In the plates task, we see that only "target assistance on" and "all assistance on" outperform "all assistance off" and "only target assistance off" in both task success, perceived fluency, and perceived trust. This suggests that the self-handover assistance mode, which was the assistance mode designed to be most pertinent in the plates task, is able to *independently* lead to improved performance and perceptions, even without the other modes being activated.

The other assistance modes do not appear to have the same independent effect on performance and perceptions. For instance, the one hand seeking assistance mode (the target assistance mode in the cups task) only outperformed "all assistance off" when it was used independently; the fixed offset assistance mode (the target assistance mode in the trashbin task) elicited benefits in perceived fluency and trust, but did not lead to significant advantages in performance over the baselines; and the one hand fixed assistance mode (the target assistance mode in the

open bottle task) did not outperform either baseline when it was the only assistance activated, while "all assistance on" did elicit performance and perception benefits over all alternatives in this task.

Our results in Study 2 suggest that, while all assistance modes appear to help across our various tasks, the task success and perception benefits are sometimes only present when the assistance modes are used *together*. Even when tasks were designed to be targeted for only a *single* assistance mode, other assistance modes still provided unexpected benefits. For example, while the open bottle task was designed to target the one hand fixed assistance mode, the other assistance modes helped throughout the task when activated, such as the one-hand-seeking assistance helping when reaching for the bottle and the self-handover assistance helping when the hands were brought close together to work in close proximity. Prior work by Swinnen et al. (*3*) provides some insight into why this effect could be present in our shared-control method. Referring to bimanual manipulation, the authors suggest that the more constraints act in coalition, the more stable a bimanual coordination pattern will be, while conversely, when constraints are in conflict with each other, performance will degrade. Thus, this suggests that because our method affords dynamic switching between a *coalition* of various assistance modes, acting as constraints in control patterns, bimanual performance will improve when *more* of these constraints are available.