Robotics: Science and Systems 2025
Los Angeles, CA, USA, June 21–25, 2025

Interface-Level Intent Inference for Environment-Agnostic Robot Teleoperation Assistance

Larisa Y.C. Loke[*§], Brenna D. Argall[*†‡§]
[*]Department of Mechanical Engineering, Northwestern University, Evanston, IL
[†]Department of Computer Science, Northwestern University, Evanston, IL
[‡]Department of Physical Medicine and Rehabilitation, Northwestern University, Chicago, IL
[§]Shirley Ryan AbilityLab, Chicago, IL, USA
larisaycl@u.northwestern.edu, brenna.argall@northwestern.edu

Abstract—**In robot teleoperation, humans issue control signals through interfaces that require physical actuation. This interface-level interaction largely goes unmodeled within the field, yet the interpretation of an interface-level command can differ from what was intended by the user, as a result of diminished human ability or inadequate mappings from raw interface signals to robot control signals. *Interface-aware* systems aim to address this limitation in robot teleoperation by explicitly considering the impact of a control interface on user input quality when interpreting interface signals for robot control. This work presents an interface-aware formulation for the direct inference of intended interface-level commands given known interaction characteristics of a control interface using data-driven modeling, allowing for teleoperation assistance without knowledge of the human's policy. In our specific implementation, we tailor the formulation to model a user's operation of a sip/puff interface using a network of Gated Recurrent Units, chosen for their ability to model temporal patterns and suitability for data-scarce domains. The resulting model is agnostic to the robot being controlled, which allows for its use in task- and environment-agnostic robot teleoperation assistance. We deploy this model in two variations of assisted teleoperation frameworks using a 1-D sip/puff interface to control a 7-DoF robotic arm, and conduct a human subjects study with spinal cord injured participants to evaluate the efficacy of our method. Our proposed task- and environment- agnostic formulation is effective in reducing collisions during teleoperation, and is preferred by users over teleoperation baselines for ease and intuitiveness of robot operation.**

I. Introduction

In robotic systems, uncertainty modeling primarily addresses uncertainties in sensing and actuation. However, in human-robot interaction (HRI) systems, where the human is fully or partially in control of the robot, a third critical source of uncertainty can present—uncertainty in the human interaction with the control interface.

When humans operate a robotic device, they must physically actuate an interface to issue a robot control command that achieves their intended task-level robot action—whether by joystick deflection, button press, or even electrical activity of the brain. However, once a human-issued control command has been processed and passed on to robot control, *how the interface is operated*, and how its mapping to the robot control space affects the measured user input, often is *not explicitly taken into consideration* when reasoning about the human command—it is treated the same by the control and autonomy pipeline, regardless of its source.

The impact of this oversight is especially severe for individuals with motor impairments who benefit from operating assistive devices in their daily lives—the control interface options that are accessible also often are limited in dimensionality and continuity, which can be mentally demanding to use for high dimensional control and can be difficult to actuate precisely, resulting in a discrepancy between intended inputs and measured user inputs. This can be further exacerbated by symptoms of neuromuscular injury such as tremors or spasticity, as well as fatigue.

In order to make controlling assistive devices easier, and enable greater physical independence for individuals with motor impairments, there is work that explores the inclusion of robotics autonomy, often in the form of shared control. Shared-control paradigms consider the human control command at many steps along the autonomy's decision-making pipeline—such as inferring the human's high-level goal [31] or determining when autonomy should step in or adapt how it assists [28]. Thus, any deviations in magnitude, timing, or direction [27] between the true signal intended by the human and that received by the robotics autonomy, as measured through the interface, can have rippling effects throughout a shared-control system.

By modeling the variability in user inputs and understanding the mapping from interface to robot actions, we can more effectively interpret human control commands [12, 15]. Such an interface-aware system models, for a given person, (a) how interface-level control actions are mapped to robot task-level actions, and (b) how intended interface-level control actions are distorted when measured through the interface. This modeling enables robotics autonomy to reason about deficiencies in human teleoperation, and provide customized assistance at the level of the *interface* action, in contrast to typical shared-control frameworks that provide assistance at the level of *robot* control commands—for example, robot end-effector pose or mobile base wheel speeds [1]. A critical step of such assistance is to *infer* the intended interface-level action. Exactly how to formulate this inference, and how to model the use of an interface by a human operator, remains however minimally addressed.

This paper dives more deeply into the question of the inference of interface-level commands, and how this inference can be used in an interface-aware teleoperation framework to provide task- and environment-agnostic robot teleoperation assistance. This work presents the following contributions:

1) A relaxed formulation of interface-aware assistance, that removes all assumptions about the user's policy.
2) A data-driven modeling method, and accompanying specifications for a data collection task, that more closely mimics real-world use of a given interface control map.
3) An end-user study in which individuals with spinal cord injury evaluate our interface-aware assistance against teleoperation baselines.

The rest of the paper is organized as follows. Section II provides an overview of related literature. Section III details our proposed interface-aware formulation, as well as the modeling and data collection methods. Section IV presents the human subjects study experimental methodology. Results and discussion are presented in Section V, with limitations in Section VI and conclusions presented in Section VII.

II. Related Work

In this section, we present a summary of related research on uncertainty handling in control interfaces, teleoperation, and shared-control for robotics. We also overview the mathematical formulation of interface-awareness, and initial efforts to apply this framework to the control of a 7-DoF robotic arm.

A. Handling Uncertainty in Interface Signals

In clinical settings, assistive device interfaces are customized to a user to account for physical variations in interface actuation. These customizations often take the form of manual adjustments and tuning of preset configurations and thresholds. For joysticks, this may include uniform scaling along principal axes, and adjustment of deadzones [13]; for one-dimensional (1-D) interfaces such as the sip/puff, signal thresholds for interface actions can be adjusted [2]. However, these adjustments often need to be done by clinician experts, or require additional hardware to complete [10], making it difficult to adapt to the varying needs of assistive device users.

More automated signal-processing methods, featured in non-commercial interfaces, often focus on techniques to compensate for the complex and variable properties in physiological signals such as EEG or EMG [17, 30], or high-dimensional body motion data [20]. By contrast, the work we present in this paper models the operation of a conventional (commercial) interface by a specific user, for the purpose of discerning discrepancies between intended and measured interface-level commands.

B. Teleoperation

Robot teleoperation typically employs a control interface with *interface-level actions* that are distinct from the workspace of the robot platform, introducing a requirement for the human operator to learn a *map* from interface-level actions to robot control commands and motions.

Within the domain of assistive robots, control maps become increasingly challenging to operate as there is often a dimensionality mismatch between the control interface and the robotic platform. To enable accessibility for users with motor impairments, assistive device interfaces often employ simplified or compensatory control schemes, such as limiting fine control over device motion or speed modulation [6]. Such limitations can result in choppy or imprecise movements, and an increased cognitive and physical operational effort [7], in comparison to more traditional direct or continuous input methods (such as the ubiquitous joystick). As a result, challenges in learning and adapting to the control map, which can be mentally and physically taxing, ultimately can impact a user's ability to effectively operate the assistive device [18].

These challenges are exacerbated when controlling higher-dimensional assistive devices, such as 6-DoF assistive robotic arms. There are no standard interfaces for simultaneous 6-D control of end-effector pose in $SE(3)$. A common solution involves mapping the movements of the human operator's body to the robot in an anthropomorphic manner [24], or partitioning the robot's control space into *modes* [19, 23] that allow for a single interface-level action to be mapped to multiple robot task-level actions via *mode switching*. Although modal control is widely used, it introduces added cognitive load in learning and using the control map. One common approach to help alleviate this burden is to share control between the human operator and a robotics autonomy system [1].

C. Shared-Control Robotic Assistance

Shared-control frameworks typically strive to improve the overall quality of robot operation (performance and safety) while maintaining human agency, by coordinating control responsibility between the human and the robot [26]. To achieve this, elements that shared-control systems commonly embody include: (a) inference: the system accurately infers the human operator's intent; (b) arbitration: the system modulates how control is shared; and (c) communication: the system provides feedback to the human regarding the state of autonomy [22].

Intent inference for shared-autonomy generally relies on analyzing control commands received by the robot [16], proprioceptive sensors on the robot [9], force sensors on the human [29], or physiological signals from the human [4, 17, 30], often with the purpose of inferring task-level actions or goals. Interface-aware robotic assistance [12, 15] instead seeks to infer interface-action level intent based on interface activation.

D. Interface-Aware Robotic Assistance

The original formulation of interface-awareness [12] aims to estimate $p(a^t|\phi_m^t)$, in order to infer the human operator's intended task-level action a^t given the measured interface-level action ϕ_m^t. Using Bayes' theorem, the relationship between conditional probability distributions can be derived to be:

$$p(a^t|\phi_m^t) \propto \eta \sum_{x^t \in \mathcal{X}} p(a^t|x^t)p(x^t) \sum_{\phi_i^t \in \Phi} p(\phi_m^t|\phi_i^t)p(\phi_i^t|a^t) \tag{1}$$

where ϕ_i^t is the intended interface-level command, x^t is the state of the world and robot, and η is a normalizing factor.

There are intuitive interpretations of the conditional probability distributions in Equation 1. $p(\phi_i^t|a^t)$ is the user's *internal model* of the control mapping from task-level actions to the interface-level actions that achieve them (e.g., to move the robot forward, deflect the joystick forward), which captures uncertainties in the user's understanding of the robot control map. $p(\phi_m^t|\phi_i^t)$ is the user's *input distortion model*, which captures biases and deficits in the user's ability to issue their intended commands through the interface. $p(a^t|x^t)$ is the user's *control policy*: the task-level action a^t they intend to take given the state x^t.

In practice, these conditional probability distributions can be built from user teleoperation data, which can be collected through simulated tasks and/or through interaction with the control interface and robot hardware [15]. Previous approaches make use of direct statistical learning—*histogram binning* of user teleoperation data—to build the models $p(\phi_i^t|a^t)$ and $p(\phi_m^t|\phi_i^t)$, which enables the evaluation of $p(a^t|\phi_m^t)$ given a known user policy $p(a^t|x^t)$ [12]. Limitations of this method include (a) its inability to model temporal patterns in teleoperation, (b) practical difficulties in abstracting the latent variable ϕ_i when collecting user teleoperation data, and (c) the need to know or estimate the user's policy $p(a^t|x^t)$, which is challenging in real world scenarios [15]. In this work, we introduce a formulation for interface-aware robotic assistance that aims to address each of these limitations.

III. Technical Approach

A problematic component of Equation 1 is the distribution $p(a^t|x^t)$. It represents an estimate of the human's control policy, and knowledge of the human's policy is a strong assumption that is unrealistic in many domains, especially if one wants to loosen requirements on the robot agent's task and environment knowledge. Prior works address this issue by minimizing the assumptions baked into the estimate of $p(a^t|x^t)$, with only the presumption of robot safety on the part of the human policy [15], or by operating within a simplified simulation environment while performing tasks with a single unambiguous goal, so that only one goal-achieving action a^t exists at each state x^t [12].

We present an alternate formulation of interface-awareness that (a) removes the need for assumptions on the human's control policy and (b) additionally accounts for known characteristics and patterns of user interaction with the control interface in the modeling approach.

A. Task- and Environment-Agnostic Interface-Awareness

In this work, we augment the user's input distortion noise model (previously $p(\phi_m^t|\phi_i^t)$) to explicitly represent the mapping $g(\cdot)$ from raw interface signal ρ^t to the interface-level action ϕ_m^t (Fig. 1, blue). For many interfaces, this map is more than just a simple linear scaling: for example, a sip/puff device uses thresholds to map to a discrete set of ϕ_i^t (inhalation/exhalation pressure that is strong/weak), and joysticks have center deadzones within which all positions map to a value of zero.

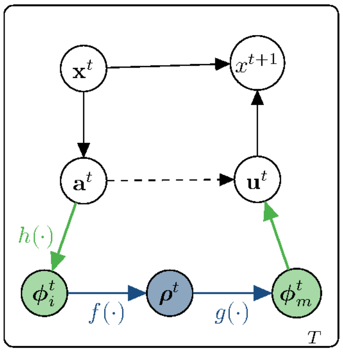

Fig. 1. User-robot interaction via a control interface depicted as a probabilistic graphical model. The typical model of teleoperation is simply $a^t \rightarrow u^t$ (dashed edge), while an interface-aware model additionally captures the physical interaction with the interface (green nodes) [12], as well as human understanding of interface control $h(\cdot)$. Our extension (blue node) provides enough context to perform direct inference of ϕ_i^t from ρ^t through a modeling of $f^{-1}(\cdot)$, without any representation of the human's policy $p(a^t|x^t)$.

We then recast the inference problem from one of estimating $p(a^t|\phi_m^t)$ to one of estimating $f^{-1}(\cdot)$: that is, of directly estimating the inverse function mapping raw signals ρ^τ to intended interface-level action ϕ_i^t. Here t is the time index of mapped interface-level commands ϕ_m^t, while τ is the time index of raw interface signals ρ^τ, which often are sampled at a higher frequency than ϕ_m^t, and so $d\tau \leq dt$.

As we will see in Section III-C, taking into account the raw interface signal, rather than only its mapping to the interface-level action (as was done in prior work [12, 15]), is crucial for reconstructing an accurate and sufficiently rich representation of a human's interface use. The primary factors that account for discrepancies between ϕ_i^t and ϕ_m^t are (a) human error or diminished ability, and (b) suboptimal mappings $g(\cdot)$ and $h(\cdot)$. In this work we consider the latter to be fixed (by interface and control paradigm design), and focus our efforts on the former.

A critical aspect of our model $f^{-1}(\cdot)$ is the incorporation of history: critical both in building its representation and for its use for inference. In detail, we consider a window of raw interface signals $\rho^{\tau-\kappa:\tau} = [\rho^{\tau-\kappa} : \rho^\tau]$ as well as the prior interface-level action $\phi_m^{t-1} = g(\rho^{\tau=t-1})$. Observing a window of raw interface signals captures a representation of the noise in that signal, while including history in the form of ϕ_m^{t-1} aims to capture patterns in interface use when issuing consecutive task-level commands.

With a user-specific model of $f^{-1}(\cdot)$, we can assist in robot teleoperation without knowledge of the user's control policy $p(a^t|x^t)$ (in contrast to [12]) or the environment state (in contrast to [15]). (How to build the model $f^{-1}(\cdot)$ is detailed next, in Sections III-C–III-D).

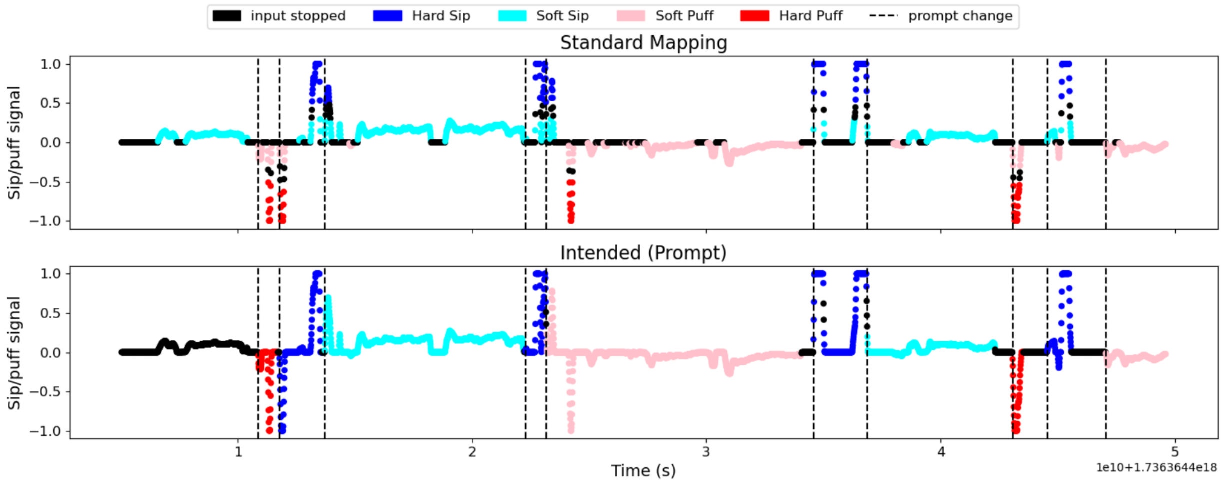

Fig. 2. A comparison of measured interface actions ϕ_m^t using thresholding (top) and ground-truth intended (bottom) interface actions (ϕ_i^t) during the data collection task (Sec. III-D). Notably, we observe instances of: unintended mode switches (wrong direction; accidental overshoot of *soft* threshold); the pressure signal passing through the *soft* pressure value range to get to the *hard* range; and inconsistent *soft* actions (that intermittently drop into the *null* range).

Algorithm 1 Interface-Aware Action Assistance

```
1: function INTERFACE-LEVEL ACTION ASSISTANCE($\rho^{\tau-\kappa:\tau}$, $\phi_m^t$, $\phi_a^{t-1}$)
2:     { $\phi_i^t$, $H$ } $\leftarrow$ INFER INTENDED ACTION($\rho^{\tau-\kappa:\tau}$, $\phi_a^{t-1}$)
3:     if $H < \epsilon$ then                    ▷ Model uncertainty is low
4:         $\phi_a^t \leftarrow \phi_i^t$                ▷ Command inferred action
5:     else
6:         $\phi_a^t \leftarrow \phi_m^t$               ▷ Command measured action
```

Algorithm 1 describes our task- and environment-agnostic interface-aware assistance, where assistance takes place at the interface-level action. Given the window of previous interface signals up to the current signal $\rho^{\tau-\kappa:\tau}$, and the previously commanded interface-level action ϕ_a^{t-1}, we use the trained user-specific model to infer the intended interface-level action at the current timestep ϕ_i^t (Line 2). If model inference uncertainty (defined in Sec. III-C) is low, we pass the inferred interface-level action to the robot control system (lines 3-4). If model inference uncertainty instead is high, we pass the measured interface-level action ϕ_m^t (lines 5-6).

B. Data-driven Modeling of Physical Interface Operation

Here we detail how to build the model $f^{-1}(\cdot)$. Our specific implementation tackles the sip/puff device, a 1-D interface typically used to operate assistive devices such as powered wheelchairs by persons with very limited upper body mobility. This interface is operated via respiration, and the continuous-valued pressure signal (ρ^τ, sampled at 120 Hz) is mapped via thresholding to one of five interface-level actions: Φ={*hard-sip, soft-sip, soft-puff, hard-puff, null*}, sampled at 60 Hz. It most commonly is used under a Latch control paradigm for powered wheelchair operation, where a single *hard-puff (hard-sip)* action is used to start (stop) linear motion, while a sustained *soft* action operates rotational motion.

In this work, we map sip/puff interface-level actions to the modal control of a 7-DoF robotic arm. Under this control scheme, 7 control modes exist: one mode for each dimension (6) of the end-effector position and orientation in $SE(3)$, and one for gripper operation. The actions ϕ_i ={*hard-puff, hard-sip*} activate (clockwise/counterclockwise) switches between control modes (with a fixed cyclical order, described in Fig. 5), and actions ϕ_i ={*soft-puff, soft-sip*} control (positive/negative) movement within a given control dimension.

We furthermore can characterize the operation of this interface-robot combination according to the following empirical observations:

- Duration: Pressure signals corresponding to *hard* actions tend to be short in nature, as they issue a discrete mode-switch action. In contrast, pressure signals corresponding to *soft* (motion) actions are longer in nature.
- Over/undershoot: It is difficult to issue longer duration (*soft*) commands with consistency, with over or undershoot of the pressure value range as a result.
- Lag: The time lag or interval between consecutive commands being issued is dependent on the similarity in pressure value between the previous and current command.
- Unintended motion: To issue a *hard* action, the pressure signal must pass through the corresponding *soft* action pressure value range (due to it being a 1-D signal with multiple thresholds), which may result in unintended robot motion during an intended mode switch action.
- Unintended mode switch: When accidental *hard* actions are issued, an unintended mode switch occurs, necessitating a *corrective* mode switch (*hard*) action, which can be fatiguing during prolonged robot arm operation.

Examples of these characteristics are visualized in Fig. 2.

These characteristics motivate the need for (a) the incorporation of temporal characteristics of the pressure signal, via data-driven sequence modeling of $f^{-1}(\cdot)$, and (b) a data collection task specifically designed to simulate actual interface use for device control.

C. Modeling Raw Signals → Intended Interface Commands

Given the need for the effective modeling of *sequences* of interface signals, in order to incorporate temporal information, we look to machine learning methods—specifically, Recurrent Neural Networks (RNNs), which are designed for processing sequential data [11]. Given the characteristics of sip/puff interface operation, we choose to use Gated Recurrent Units (GRUs) to learn a user-specific mapping from ρ^{τ} to ϕ_i^t. GRUs are chosen for their simplicity compared to other sequential modeling techniques such as Long Short-Term Memory (LSTM), which allows for shorter training times and are shown to outperform LSTMs on low-complexity problems with small datasets and shorter-term dependencies [5].

We train a multi-layer GRU-based neural network classifier which is made up of three GRU layers, a final linear layer, and a softmax activation that maps the output of the linear layer to the output classes Φ. The input to the classifier is described in Section III-D. We use the Adam optimizer during training. To motivate learning that takes into account interface operation characteristics as well as practical usability, we design a loss function that incorporates the following metrics:

- Cross-entropy loss, $\mathcal{L}_{CE}$: The difference between the predicted probability distribution and the ground truth distribution, commonly used for classification tasks.
- Weighted accuracy, A_w: Percentage of *confident* and *correct* predictions, rewarding the model for being confidently correct.
- Confidently wrong penalty, P: Percentage of *confident* and *wrong* predictions, penalizing the model for being confidently wrong—notably, this aims to mitigate unwanted corrections to the user's input during model deployment.

Here *confidence* is determined from the normalized entropy H of the model's output, where $H < \epsilon$ is designated to be confident (certain). (In our implementation, $\epsilon = 0.1$.) The additional loss terms A_w and P *explicitly* encode the selected confidence threshold into the loss function. (In contrast to cross-entropy loss, which motivates higher prediction confidence however without a specific target.) This explicit encoding is critical for real-time use, as the threshold governs whether measured interface actions are passed through to the robot or replaced with inferred actions.[1] The resulting loss function is as follows:

$$\mathcal{L} = \mathcal{L}_{CE} - \alpha A_w + \beta P \tag{2}$$

where α and β are tunable weights on the loss terms.

The classifier is implemented in PyTorch [3], and model architecture and learning hyperparameters are optimized using Ray Tune [21] to maximize the weighted accuracy A_w as well as the F_β score of the classifier. Model architecture and hyperparameter tuning is done on sip/puff datasets collected from volunteers (lab members) with a range of experience using a sip/puff interface. The final optimized model architecture and hyperparameters are detailed in Table I. These same hyperparameters are used when training each SCI user model.

TABLE I
FINAL MODEL ARCHITECTURE AND HYPERPARAMETERS

	Descriptor	Value
Model Architecture	Hidden size	60
	Number of GRU layers	3
	Dropout for regularization	0.2
	α, β	1, 2
Hyper-parameters	Initial learning rate (LR)	0.0001
	LR scheduler	ReduceLROnPlateau
	LR decay rate	0.8
	Weight decay for regularization	0.0005

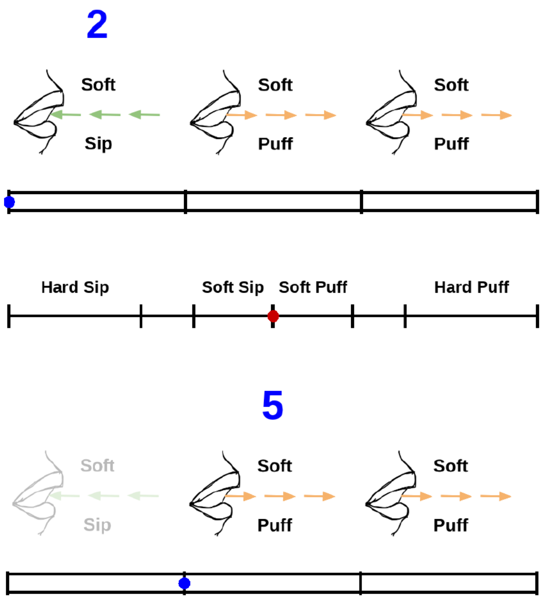

Fig. 3. Data collection task. Interface-level action prompts are displayed in groups of three, each with a countdown timer. A progress indicator (blue dot) also serves as a proxy for the control result of the interface-level command. *Top:* The task first is used for familiarization with the interface, during which visual feedback of the user's raw pressure signal ρ is provided via the position of the red dot. *Bottom:* After familiarization, this feedback is removed for the data collection phase, since it is not available when operating the robot arm.

D. Data Collection Tasks

For the GRU network to model $f^{-1}(\cdot)$ for a specific user, we need to collect representative labeled data of the user interacting with the control interface. We design a data collection task that, critically, is *responsive to known interface operation characteristics* (identified in Section III-C). Specifically, the task aims to elicit these characteristics during data collection, so that they are represented within the training dataset. We furthermore apply data processing and augmentation techniques to produce a dataset suitable for GRU training.

The data collection prompts consist of a sequence of three interface-level actions presented to users on a screen (Fig. 3).

[1] Preliminary ablations with the threshold ϵ and combinations of loss terms showed that ϵ can be lowered when the additional terms are included, as compared to only cross-entropy loss, likely because the explicit encoding of ϵ drives the model towards making confident predictions that meet the threshold. A lower threshold ϵ is useful in safety-critical settings, such as powered wheelchair driving, as model predictions should be extremely confident before replacing human commands.

Users are asked to provide the respiration action that would produce the prompted action. The criteria for prompt completion differs depending on the interface-level action, to reflect the differences in usage when issuing a motion command (continuous *soft* actions) versus a mode switch (short and discrete *hard* actions).

- For *soft* actions ϕ_i, users are tasked with providing the corresponding pressure command ρ for a set length of time, during which a blue dot moves left→right across the screen (through the channel) towards the next prompt.
- For *hard* actions ϕ_i, once users have successfully provided the corresponding pressure command ρ, the prompt is considered complete.

For both, the blue dot also serves as a sort of proxy for the effect of the interface-level action on the robot control, where *hard* actions (mode switches) result in abrupt changes in robot control while *soft* actions (motion commands) result in robot movement. Notably, this proxy aims to simulate the feedback that users would receive during real robot teleoperation. (Similar feedback is absent in prior work [12].)

In total, the task consists of 32 sets of 3 prompts. We generate prompts by taking the Cartesian product of the set of actions Φ 3 times, then randomly sample the resulting set until we get an action-prompt-balanced set of 32. With 32 sets, data collection takes approximately 6 minutes to complete, excluding three breaks (every 8 sets) given to users during the task. In our study, data collection (including breaks) took $431s$ on average ($\sim 7min$), and did not appear taxing for participants.

The recorded data timeseries is partitioned into rolling windows of 10 timesteps of ρ^τ. (Thus $\kappa = 9$, Alg. 1). The ground truth (intended) previous interface-level action in Φ is concatenated to the window as a tiled 5-vector, where the representation of the action is embedded in the range of ρ. At deployment time, we do not have access to the ground truth previous action, and so we would use the executed interface-level action ϕ_a^{t-1} from the previous timestep.

The resulting dataset consists of input tensors of size 10 by 6, and output tensors of size 4 (one-hot encoding of the actions in Φ). Since *hard* action prompts are shorter, the collected user data has fewer time samples of *hard* action raw interface signals. The dataset is upsampled with noise added to ρ^τ to ensure a balanced dataset.

IV. Experimental Methodology

We conduct a human subjects study to evaluate the efficacy of our proposed task- and environment-agnostic interface-aware assistance, and compare it with unassisted teleoperation and environment-aware collision avoidance assistance.

A. Participants

A total of 8 participants with cervical-level spinal cord injury (SCI) (5 male, 3 female) aged 28 to 62 (mean 47.6±13.2) were recruited. Participants had varying levels (C2 to C7) and types (complete and incomplete) of injury to the spinal cord. Detailed participant demographics are provided in Appendix VIII. All participants gave their informed consent to participate in the experiment, which was approved by Northwestern University's Institutional Review Board (STU00211758).

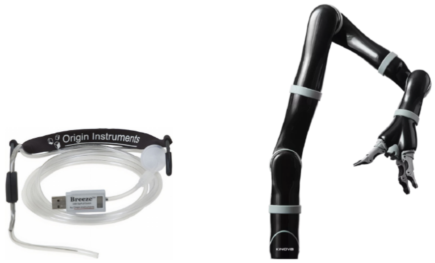

Fig. 4. Left: Sip/Puff Breeze USB interface with headset from Origin Instruments Corp. Right: Kinova Jaco 7-DoF robotic arm.

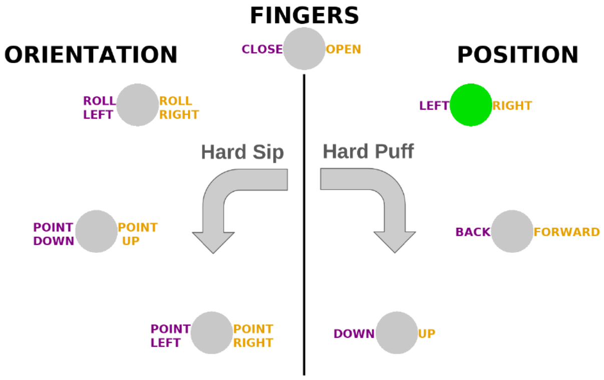

Fig. 5. Cyclical screen-based mode display, with the current control mode of the robot indicated (green circle). Text to the left/right (purple/yellow) of each mode (circle) describes the end-effector motion that corresponds to *soft-sip/puff* actions. When users issue a *hard-sip/puff* to mode switch, the green circle jumps to the next mode in the corresponding direction (counterclockwise/clockwise), with wrapping.

B. Hardware and Materials

We use a Sip/Puff Breeze™ interface from Origin Instruments Corporation (Texas, USA) to control a 7-DoF Kinova Jaco robotic arm (Quebec, Canada), as seen in Fig. 4. While controlling the arm, participants also are shown the circular mode display in Fig. 5 on a screen. The mode display is specifically designed to incorporate visual cues for the user regarding the sip/puff-Jaco control scheme.

C. Study Procedure

1) Participant Training: Participants are trained to use the sip/puff to issue interface-level actions as defined by a *standard thresholding map*. As is the standard in clinical practice, this map sets thresholds that dictate the mapping of a pressure value to a *hard/soft-sip/puff*. A real-time visualization of the ρ signal is shown to participants, and the four raw signal value ranges corresponding to the interface-level actions are explained. Participants are then given time to familiarize themselves with and practice the four actions.

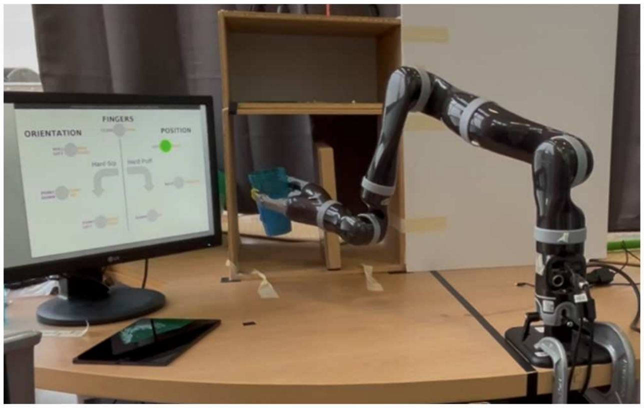

Fig. 6. ADL Task. The environment setup places a cup upside-down in the center of the upper shelf, and initializes the robot arm pose. The task proceeds as: (1) move the end-effector into the upper shelf, (2) grasp and lift the cup, (3) remove the cup from the upper shelf, (4) flip the cup right-side-up, (5) position the cup inside the left-hand compartment of the lower shelf, (6) open the gripper. A trial is considered successful if the cup is placed (gripper opened) in the target pose and orientation (inside lower shelf, right-side-up) within the time limit. A trial is considered unsuccessful if participants exceed the time limit, drop the cup, or do not achieve the goal pose and orientation.

After this, participants are introduced to the data collection task detailed in Section III-D. They are given time to practice the task with and without the raw pressure signal feedback (familiarization), after which the data collection phase commences. When data collection is complete, the GRU network is trained on the participant's data to build a user-specific model of interface use ($\sim$*5min*).

Participants then are trained to operate the robot arm using the interface, first by a verbal and visual introduction to the 7 different control modes and mode display (Fig. 5), followed by hands-on practice for mode switching (*hard* actions) and issuing motion commands (*soft* actions).

2) Evaluation Tasks: Participants are tasked with completing an Activities of Daily Living (ADL) task of picking and placing a cup in a different position and orientation (Fig. 6) within a 7 minute time limit. Training involved coaching the participant to complete the task as subtasks, followed by two rounds of timed practice completing the task using the standard map with no assistance. Participants are instructed to complete the task as quickly as possible while avoiding collisions. This task is chosen for its complexity because it requires (a) entering/navigating a tight space, and (b) orienting/pivoting the end-effector, both of which are considered difficult for human teleoperation as well as robot path planners [8].

Participants then execute the ADL task under four different combinations of control mappings and assistance conditions—the standard map and user-specific interface-aware map, with and without collision avoidance assistance:

- Standard Map (*Std-Map*): The raw 1-D signal of the sip/puff interface is mapped to interface-level actions via linear thresholding, as is the clinical standard.
- Interface-Aware Map (*IA-Map*): A sequence of raw 1-D signals from the sip/puff interface are mapped to interface-level actions using a GRU-based model trained on a specific user's interface actuation data, taking into account temporal characteristics and command history.
- Standard Map with Safety Assistance (*Std-Map+Safety*): The Std-Map, augmented with collision avoidance assistance.
- Interface-Aware Map with Safety Assistance (*IA-Map+Safety*): The IA-Map, augmented with collision avoidance assistance.

Collision avoidance (*Safety*) assistance is provided by representing the task environment as a potential field, wherein collision objects (i.e., tabletop, shelves, etc.) have repulsive potentials with respect to the end-effector position. Self collisions are not represented. No attractive potentials are defined, to keep the assistance task-agnostic.

Participants are *not* told which teleoperation condition is active during each trial, and the presentation order of conditions is randomized and balanced across participants.

3) Questionnaires: After each ADL trial, participants are asked to fill out a Raw NASA-TLX survey [14] to assess their self-perceived workload for the trial, as well as a questionnaire of their perception of the teleoperation condition:

1) An autonomous assistance system was assisting me to perform the task.
2) The autonomous assistance system helped me to complete the task faster.
3) The autonomous assistance helped me complete the task more safely.
4) The autonomous assistance helped me to reduce unwanted commands.
5) The autonomous assistance helped me to reduce the number of mode switches.
6) I preferred doing the task with assistance.

Question 1 is a True/False question, and questions 2-6 are 7-point Likert scales ranging from Strongly Disagree (1) to Strongly Agree (7). At the end of all four trials, participants are asked to rank the teleoperation conditions in order of ease and intuitiveness of robot control.

V. Results and Discussion

In this section, we present results relating to our user-specific interface-aware map testing, as well as ADL task completion metrics and questionnaire responses.

A. Interface-Aware Map Testing

We first compare the offline performance of the two interface maps without any collision assistance: the *Standard Map (Std-Map)* and *Interface-Aware Map (IA-Map).*

Fig. 7 presents the mapping accuracy of each approach: that is, the accuracy of each map outputting the interface-level command as prompted during the data collection task (ground truth label) given the raw measured interface-level signal. Data fed through each map consists of the 20% hold out from the data collection task, where for the IA-Map the remaining 80% was used for training the GRU model.

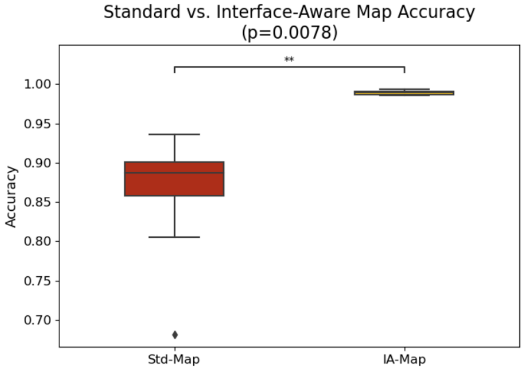

Fig. 7. Interface-level command mapping accuracy by the standard map compared to the user-specific interface-aware model prediction, evaluated on the test dataset collected from the data collection task. Interface-aware map predictions are significantly more accurate than the standard map across all participants. Statistical significance is computed using a permutation test on the difference in means of prediction accuracy between the two maps for all participants, validated by a strong effect (Cliff's Delta [25] $\delta = 1$).

TABLE II
MEAN PERCENTAGE OF CORRECT/WRONG AND CONFIDENT/UNCERTAIN INTERFACE-AWARE MODEL PREDICTIONS ACROSS ALL PARTICIPANTS

	Percentage of Predictions			
	Correct Confident	Correct Uncertain	Wrong Confident	Wrong Uncertain
Hard Puff	98.1%	0%	0%	1.9%
Soft Puff	99.8%	0.1%	0%	0.1%
Soft Sip	99.8%	0.1%	0%	0.1%
Hard Sip	97.8%	0%	0%	2.2%

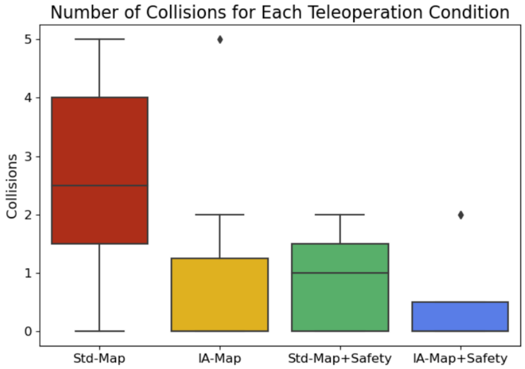

Fig. 8. Number of collisions for each assistance condition, averaged over participants. A given participant is represented by a colored marker. With any form of assistance (interface-aware map and/or safety assistance), collisions are generally reduced.

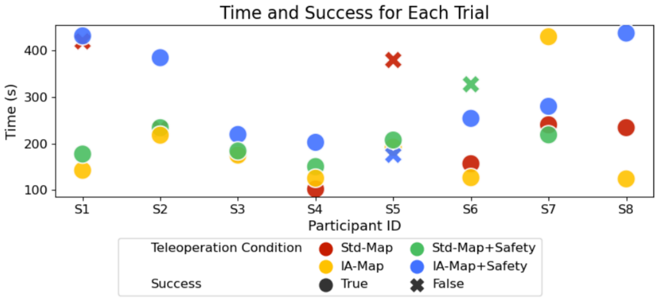

Fig. 9. Time taken and success of each trial. Successful trials are marked by circles while unsuccessful trials are marked with crosses.

Across our population of 8 participants, the interface-aware maps have significantly ($p < 0.01$) superior performance compared to the standard maps, both in the average over all participants and for each participant individually. These results suggest that the method of modeling employed by the IA-Map is able to capture nuances in the interface signal, and model interface-level intent more accurately than the standard map.

Next, we take a closer look at the interface-aware GRU model prediction results. Table II shows the percentage of correct/wrong and confident/uncertain model predictions averaged across all participants during testing, wherein confidence is determined by the normalized entropy H as detailed in Section III-C. Notably we see that, across all participants, the interface-aware model makes zero *Wrong-Confident* predictions, and 98-99% of predictions are *Correct-Confident*.

The ability to predict correctly with confidence is a key property of the interface-aware map. At run-time, the decision of whether to use ϕ_m, the measured interface-level action, or ϕ_i, our estimation of the intended interface-level action, is made based on the value of H. By design, only *Confident* predictions lead us to replace ϕ_m with ϕ_i. Moreover, and perhaps even more important, is that all incorrect predictions occur with uncertainty: *Wrong-Confident* predictions would result in an *unintended* interface-level action replacing the measured interface-level action for robot control, and are potentially worse than doing nothing at all and simply allowing the measured command to pass through. Thus, minimizing the instance of *Wrong-Confident* predictions, ideally down to zero, as is done here, is of critical import.

B. ADL Task Performance

We present ADL task completion metrics over the four different combinations of mappings and assistance conditions (teleoperation conditions). Due to the small sample size (8 participants), we do not include statistical analyses in our discussion but instead present descriptive analyses of ADL the task metrics.

Fig. 8 shows the number of collisions, averaged over participants' ADL trials. We see that any form of assisted teleoperation (*IA-Map, Std-Map+Safety,* and *IA-Map+Safety*) reduces the number of collisions. Notably, the *IA-Map* condition is effective at reducing the number of collisions a person experiences while completing the task, even *without* any encoded knowledge of the environment (which the *Safety* conditions both require).

Fig. 9 shows the success rate and time taken to complete trials for each of the participants under the four conditions. In general, we see a range of variability in task completion times across participants. The *IA-Map* and *Std-Map+Safety* conditions result in similar task completion times in 5 of the 8 participants, with the *IA-Map* having the shortest task completion time in 5 of 8 participants.

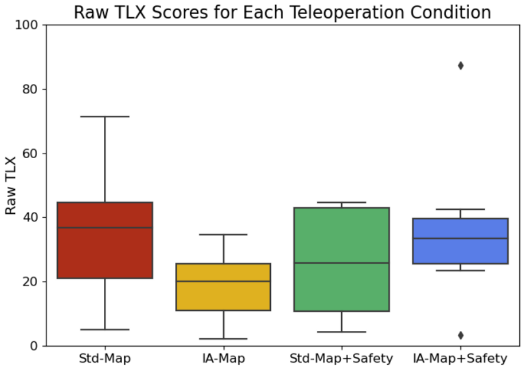

Fig. 10. Raw NASA-TLX scores for each teleoperation condition, averaged over participants.

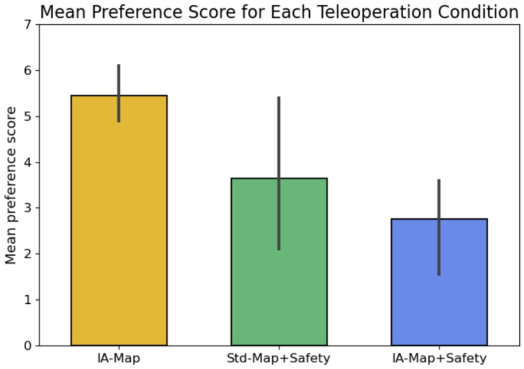

Fig. 11. Preference scores for each assisted teleoperation condition in which a participant identified the presence of assistance, averaged over participants.

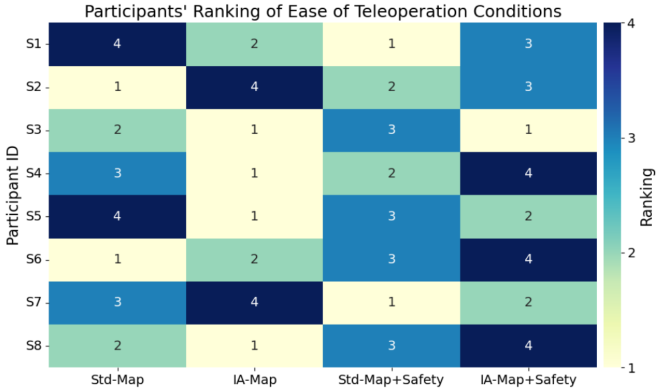

Fig. 12. Participants' ranking of the ease of robot control for task completion, for all teleoperation conditions.

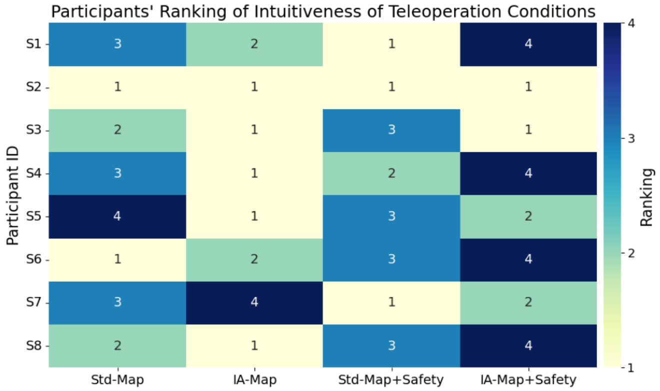

Fig. 13. Participants' ranking of the intuitiveness of robot control for all teleoperation conditions.

C. ADL Task Questionnaire Results

We present participants' self-reported NASA-TLX workload scores and preferences, over the four different combinations of mappings and assistance conditions during the ADL task.

Fig. 10 shows participants' self-reported workload scores from the Raw NASA-TLX questionnaire. We see that, in general, participants report consistently lower workload when operating under the *IA-Map* condition, and elevated workload under the *Std-Map* and *IA-Map+Safety* conditions.

Interestingly, the *Std-Map* makes no modifications to the raw user command, while the *IA-Map+Safety* arguably makes the 'most' modifications, by having both interface-aware and safety layers. This result suggests that assistance that is too great, or too complex, might be counterproductive or unintuitive (as we will see in Fig. 13), perhaps due to control transparency. If true, there likely exists a sweet spot in the magnitude or complexity of teleoperation assistance that users can be expected to comfortably use.

We next evaluate the post-trial surveys that query participants for their perception of assistive autonomy during teleoperation. For all conditions, 50% of participants answered that they thought there was assistance during the task. For these responses, under the conditions where teleoperation assistance was present, we aggregate the scores of questions 2-6 (Sec. IV-C3) to compute each participant's *mean preference score* for each assistance system. The results are presented in Fig. 11.

We see that, of the three assisted teleoperation conditions, participants prefer *IA-Map* the most and *IA-Map + Safety* the least. Based on anecdotal vocalizations during study sessions, participants can become frustrated by conditions that included collision avoidance *Safety* assistance, citing the following issues:

- Too conservative: *Safety* assistance imposes a buffer zone that does not allow scraping the end-effector or bumping collision objects, with which participants can feel comfortable if it allows them to complete the task.
- Opaque: When participants did not know when and why the assistance was engaging they were unable to work with, rather than against, it, making operation difficult.

Finally, we ask participants to rank each of the four trials according to *ease of use* (Fig. 12) and *intuitive operation* (Fig. 13). Participants are given the option to rank multiple conditions as equal if they did not perceive a difference. We find most often *IA-Map* is ranked as the easiest and most intuitive teleoperation condition, while *IA-Map+Safety* most often is ranked the least. This result reinforces the idea of a sweet spot in the magnitude or complexity of teleoperation assistance. Moreover, when coupled with the issues participants raised about *Safety* assistance, this suggests that the user-specific interface-aware model powering the *IA-Map* sufficiently models the user's interface operation characteristics to allow for an easier and more intuitive robot teleoperation experience *without* additional layers of robotics autonomy.

VI. Limitations

Due to the small sample size of participants with spinal cord injury in this human subjects study, we are unable to draw conclusions with statistical significance with regards to ADL task metrics and questionnaire results. Additionally, due to constraints on study session length to avoid participant fatigue, we only conduct a single trial for each of the assisted teleoperation conditions, and results for each of these trials may have been affected by factors other than the assistance condition such as learning effects or fatigue.

The standard map sip/puff thresholds were kept constant between participants. During the study design process, we consulted with clinical practitioners who work with sip/puff users, and found there to be a lack of a repeatable, standardized method for threshold tuning, especially for non-practitioners. We thus chose to omit manual threshold tuning to keep the study protocol standardized across all participants.

Participants tended to prefer conditions without *Safety* assistance. A version of *Safety* that incorporates the ability to override could perhaps have addressed frustrations that participants voiced regarding the assistance.

This work presents a generalized framework for interface intent modeling, with a specific implementation for a sip/puff device. The scalability of interface intent modeling is dependent on the interface being modeled, since different interfaces have different interface operation characteristics and *interface action* spaces Φ, and accordingly will require different design and implementation choices, for both the data collection method and selection of a suitable machine learning technique to model $\rho^{\tau} \rightarrow \phi_i^t$. It thus is difficult to say how the computation and performance requirements of interface-aware systems will scale with higher dimensional interfaces, for which further exploration is needed.

VII. Conclusion

We have presented a formulation of interface-aware teleoperation assistance that removes any need for assumptions about the human policy or information about the environment. Our interface-aware, environment-agnostic teleoperation assistance was evaluated in a human subjects study, where participants with spinal cord injury were tasked with operating a sip/puff interface to control a robotic arm. We further compared the performance of our formulation to a variant that was environment-aware, as well as formulations that were interface-agnostic. The results showed our proposed assistance to outperform standard teleoperation mappings as well as environment-aware collision avoidance systems in improving teleoperation safety, task completion times, as well as reducing operator workload. Our method also was found to be most preferred by participants in the study. These findings point to the utility of an interface-aware framing for teleoperation assistance, that furthermore can circumvent the need for assistance from robotics autonomy that depends on environment sensing and perception.

Acknowledgments

The authors would like to thank Corinne Fischer, OTR/L, Occupational Therapist at Shirley Ryan AbilityLab, for assisting with study sessions and participant recruitment. This material is based upon work supported by the National Science Foundation under Grant CMMI-2208011. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

References

[1] David A. Abbink, Tom Carlson, Mark Mulder, Joost C.F. De Winter, Farzad Aminravan, Tricia L. Gibo, and Erwin R. Boer. A topology of shared control systems-finding common ground in diversity. *IEEE Transactions on Human-Machine Systems*, 48(5):509–525, 2018.

[2] Benjamin Aigner, Veronika David, Martin Deinhofer, and Christoph Veigl. Flipmouse: A flexible alternative input solution for people with severe motor restrictions. In *Proceedings of the International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion (DSAI)*, 2016.

[3] Jason Ansel, Edward Yang, Horace He, Natalia Gimelshein, Animesh Jain, Michael Voznesensky, et al. PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation. In *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, 2024.

[4] Reuben M Aronson and Henny Admoni. Eye gaze for assistive manipulation. In *Companion of the ACM/IEEE International Conference on Human-Robot Interaction (HRI)*, 2020.

[5] Roberto Cahuantzi, Xinye Chen, and Stefan Güttel. A Comparison of LSTM and GRU networks for learning symbolic sequences. In *Intelligent Computing: Proceedings of the Computing Conference (SAI)*, 2023.

[6] Tom Carlson, Guillaume Monnard, Robert Leeb, and José del R Millán. Evaluation of proportional and discrete shared control paradigms for low resolution user inputs. In *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011.

[7] Kenneth A Chizinsky, Heidi M Horstmann, Simon P Levine, and Daniel J Koester. Modification of a proportional joystick to incorporate switch outputs for accessories. *Assistive Technology*, 1(4):101–105, 1989.

[8] Marco Costanzo, Simon Stelter, Ciro Natale, Salvatore Pirozzi, Georg Bartels, Alexis Maldonado, and Michael Beetz. Manipulation planning and control for shelf replenishment. *IEEE Robotics and Automation Letters (RA-L)*, 5(2):1595–1601, 2020.

[9] Mustafa Suphi Erden and Tetsuo Tomiyama. Human-intent detection and physically interactive control of a robot without force sensors. *IEEE Transactions on Robotics (T-RO)*, 26(2):370–382, 2010.

[10] Michael Gillham, Matthew Pepper, Steve Kelly, and Gareth Howells. Feature determination from powered wheelchair user joystick input characteristics for adapting driving assistance. *Wellcome Open Research*, 2:93, 2018.

[11] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[12] Deepak Gopinath*, Mahdieh Nejati-Javaremi*, and Brenna Argall. Customized handling of unintended interface operation in assistive robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[13] Alcinto S Guirand, Brad E Dicianno, Harshal Mahajan, and Rory A Cooper. Tuning algorithms for control interfaces for users with upper-limb impairments. *American Journal of Physical Medicine & Rehabilitation*, 90(12): 992–998, 2011.

[14] Sandra G Hart. NASA-task load index (NASA-TLX); 20 years later. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting (HFES)*, 2006.

[15] Mahdieh Nejati Javaremi, Larisa Y.C. Loke, and Brenna D. Argall. Interface-aware assistance for 7-dof robot arm teleoperation: Case studies on feasibility. In *Proceedings of the International Symposium on Experimental Robotics (ISER)*, 2023.

[16] Zongyao Jin, Prabhakar R Pagilla, Harshal Maske, and Girish Chowdhary. Task learning, intent prediction, and adaptive blended shared control with application to excavators. *IEEE Transactions on Control Systems Technology*, 29(1):18–28, 2020.

[17] Reva E Johnson, Konrad P Kording, Levi J Hargrove, and Jonathon W Sensinger. EMG versus torque control of human–machine systems: Equalizing control signal variability does not equalize error or uncertainty. *IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE)*, 25(6):660–667, 2016.

[18] Gerard Kelliher, Jing Farrelly, Qasim Afridi, Franklin Arubi, and Bridget Kane. Designing control interfaces for powered wheelchair users. In *Proceedings of the Irish Human Computer Interaction Conference (iHCI)*, 2010.

[19] Dae-Jin Kim, Ryan Lovelett, and Aman Behal. An empirical study with simulated ADL tasks using a vision-guided assistive robot arm. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2009.

[20] Jongmin Lee, Temesgen Gebrekristos, Dalia De Santis, Mahdieh Nejati-Javaremi, Deepak Gopinath, Biraj Parikh, et al. Learning to control complex robots using high-dimensional body-machine interfaces. *ACM Transactions on Human-Robot Interaction (THRI)*, 13(3):1–20, 2024.

[21] Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E. Gonzalez, and Ion Stoica. Tune: A Research Platform for Distributed Model Selection and Training. *ArXiv*, abs/1807.05118, 2018.

[22] Dylan P Losey, Craig G McDonald, Edoardo Battaglia, and Marcia K O'Malley. A review of intent detection, arbitration, and communication aspects of shared control for physical human–robot interaction. *Applied Mechanics Reviews*, 70(1):010804, 2018.

[23] Dylan P Losey, Hong Jun Jeon, Mengxi Li, Krishnan Srinivasan, Ajay Mandlekar, Animesh Garg, et al. Learning latent actions to control assistive robots. *Autonomous Robots*, 46(1):115–147, 2022.

[24] Matteo MacChini, Fabrizio Schiano, and Dario Floreano. Personalized telerobotics by fast machine learning of body-machine interfaces. *IEEE Robotics and Automation Letters (RA-L)*, 5(1):179–186, 2019.

[25] Kane Meissel and Esther S Yao. Using Cliff's delta as a non-parametric effect size measure: an accessible web app and R tutorial. *Practical Assessment, Research, and Evaluation*, 29(1), 2024.

[26] Selma Musić and Sandra Hirche. Control sharing in human-robot team interaction. *Annual Reviews in Control*, 44:342–354, 2017.

[27] Mahdieh Nejati Javaremi, Michael Young, and Brenna D Argall. Interface operation and implications for shared-control assistive robots. In *Proceedings of the IEEE International Conference on Rehabilitation Robotics (ICORR)*, 2019.

[28] Luka Peternel, Nikos Tsagarakis, Darwin Caldwell, and Arash Ajoudani. Robot adaptation to human physical fatigue in human–robot co-manipulation. *Autonomous Robots*, 42(5):1011–1021, 2018.

[29] Fernando Trincado-Alonso, Antonio J del Ama-Espinosa, Guillermo Asín-Prieto, Elisa Piñuela-Martín, Soraya Pérez-Nombela, Ángel Gil-Agudo, et al. Detection of subject's intention to trigger transitions between sit, stand and walk with a lower limb exoskeleton. In *Wearable Robotics: Challenges and Trends: Proceedings of the International Symposium on Wearable Robotics*, 2017.

[30] John Williamson, Roderick Murray-Smith, Benjamin Blankertz, Matthias Krauledat, and K-R Müller. Designing for uncertain, asymmetric control: Interaction design for brain–computer interfaces. *International Journal of Human-Computer Studies*, 67(10):827–841, 2009.

[31] Kelilah L Wolkowicz, Robert D Leary, Jason Z Moore, and Sean N Brennan. Discriminating spatial intent from noisy joystick signals for wheelchair path planning and guidance. In *Proceedings of the ASME Dynamic Systems and Control Conference (DSCC)*, 2018.

VIII. Participant Demographics

Participant ID	Age	Sex	SCI level	Sip/puff experience	Robotic devices		Enjoys using new technology[3]	Uses computer daily[3]	Plays video/computer games frequently[3]
					Experience[1]	Comfort[2]			
S1	61	Male	C4-6	None	5	5	5	5	3
S2	54	Female	C5	None	1	3	5	5	1
S3	41	Male	C2-3	Yes, previously learned to use, but not anymore	2	4	5	5	5
S4	30	Male	C6-7	None	3	5	5	5	1
S5	62	Male	C-unknown	None	3	4	5	5	1
S6	28	Male	C6-7	None	1	3	4	5	1
S7	53	Female	C-unknown	None	1	3	5	5	5
S8	52	Female	C5-7	None	3	4	5	5	5

[1] 5-point Likert scale from Not Experienced (1) to Extremely Experienced (5).

[2] 5-point Likert scale from Not Comfortable (1) to Extremely Comfortable (5).

[3] 5-point Likert scale from Strongly Disagree (1) to Strongly Agree (5).