2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Integrating Open-World Shared Control in Immersive Avatars

Patrick Naughton*[1], *Student Member, IEEE*, James Seungbum Nam*[2], *Student Member, IEEE*, Andrew Stratton[1], and Kris Hauser[1], *Senior Member, IEEE*

***Abstract*— Teleoperated avatar robots allow people to transport their manipulation skills to environments that may be difficult or dangerous to work in. Current systems are able to give operators direct control of many components of the robot to immerse them in the remote environment, but operators still struggle to complete tasks as competently as they could in person. We present a framework for incorporating open-world shared control into avatar robots to combine the benefits of direct and shared control. This framework preserves the fluency of our avatar interface by minimizing obstructions to the operator's view and using the same interface for direct, shared, and fully autonomous control. In a human subjects study (N=19), we find that operators using this framework complete a range of tasks significantly more quickly and reliably than those that do not.**

I. Introduction

Teleoperation allows humans to sense and act in remote locations that may be hazardous or difficult to access. Recently, several groups have developed robot avatars [4, 21, 22, 29] that provide immersive interfaces for operators to control an entire robot body and transport their presence to a remote location. These systems have proven that avatars enable novice operators to intuitively inspect, navigate, and manipulate the remote environment, but even state-of-the-art systems lag behind human proficiency [12].

This skill gap has long been identified as an issue for teleoperation, and researchers have proposed many assistance schemes to mitigate it, including virtual fixtures [1, 13, 24, 26], mode switches [25], and automated planning [5, 18, 19]. Assistance has been shown to help operators in structured lab settings, but several challenges remain before they can be deployed, such as "open-world" tasks (tasks where the number and/or types of objects in the robot's environment are not known a-priori) [32], predicting the operator's intent [20], evaluating and managing the operator's trust [20], and operator overload degrading the operator's fluency [8]. The open-world problem is particularly troublesome, since teleoperation is especially effective in leveraging human problem-solving and contextual understanding, but nearly all assistance methods are designed to work with predefined objects in semi-structured scenarios [5, 13, 25, 31]. Another major challenge is bridging assistance paradigms with the immersive paradigm. Existing avatars incorporate few assistive features [6, 21, 28], whereas shared control literature typically considers non-immersive mouse and keyboard interfaces [19, 24]. The question of how to integrate these schemes introduces several design challenges, such as how to allow the operator to quickly switch between control modes and configure different types of assistance without occluding the view of the remote environment.

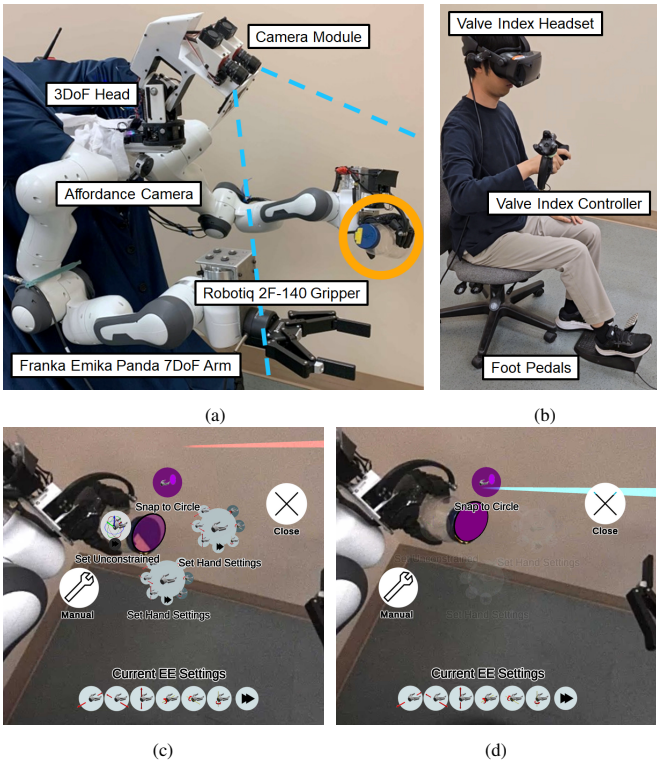


Fig. 1: Avatar robot unscrewing a jar (a) while controlled by an operator (b). The operator's view is overlaid with a predictive menu (c) that suggests possible assistive actions and shows corresponding affordances as augmented-reality objects (purple circle overlaying the jar lid). The operator can use a laser pointer attached to their controller to select one of the suggested actions (d). [Best viewed in color.]

The contribution of this work is the design and evaluation of a framework to incorporate open-world shared control into immersive robot avatars. To address the central design challenges highlighted above, we created an in-headset menu that allows the operator to launch and configure assistive actions using the same controllers they use to directly move the robot (Fig. 1). We implement assistive actions based on geometric affordances that are agnostic to object identity, allowing them to work in a wide range of scenarios. Affordances are rendered as augmented reality (AR) markers in the operator's immersive view when the user is con-

[1]P. Naughton, A. Stratton and K. Hauser are with the Department of Computer Science, University of Illinois at Urbana-Champaign, IL, USA. {pn10, ars21, kkhauser}@illinois.edu
[2]J. S. Nam is with the Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, IL, USA. sn29@illinois.edu
This work was supported by NSF Grant #2025782.
*Equal contribution. Corresponding author listed first.

figuring action targets. We further enhance the fluency of this interface using an "autocomplete" predictive menu that predicts the operator's intent in the context of the current scene and history [23]. We incorporate this framework into an avatar system and evaluate novice users on long-form tasks that require many uses of the assistive actions. Human subjects testing ($N = 19$) verifies that our approach, with and without the predictive menu, increases task success rates and system usability, and decreases task completion times and operator workload over standard direct control interfaces while preserving the operator's self-reported sense of presence in the remote environment.

II. Related Work

The recent ANA Avatar XPRIZE competition spurred rapid development of teleoperated avatar robots capable of transporting basic human manipulation skills to remote environments [12]. As the competition emphasized immersion and presence, most teams made very little or no use of shared control, instead opting to give as much direct control to the operator as possible. This choice makes the systems open-world, immersive, and intuitive, but users still struggle to perform tasks through the robot as proficiently as they would in-person [12]. Shared control methods could hypothetically assist in operator proficiency while preserving desirable aspects of immersion, but mechanisms for achieving such integration are not well studied.

Operator assistance for non-immersive interfaces has received much attention in the literature. A significant line of work addresses reaching for an object [7], especially when the operator's interface has fewer DoFs than the robot [11, 14, 15, 25]. In the avatar context, this is not normally a concern because the operator has access to high DoF input devices. Other research provides assistance for complex tasks but requires pre-programmed information about the environment and target objects [5, 13, 25]. For example, [25] presents a system that can perform complicated tasks like opening a door, but key frames of reference for specific objects are labeled by hand, and the state-machines describing transitions between different phases of the tasks are pre-specified. Our work seeks to relax this requirement and provide assistance in an open-world where the semantic identities and number of objects encountered in the environment are not known ahead of time. We achieve this by using more generic types of assistance, detecting affordances at runtime rather than hand labelling them at design-time.

The work of Pruks and Ryu [24] is most similar to our system. Similar to our system, their work uses off-the-shelf methods to segment the environment into geometric primitives and allows the operator to apply customizable virtual fixtures between features detected in the environment and features from the robot. However, they use a screen-and-mouse interface to specify virtual fixtures and a separate haptic device to input low-level motion commands, requiring the operator to switch between two input devices. In contrast, our system uses a consistent input interface for both specifying virtual fixtures and providing low-level commands. Our system also provides an immersive interface via a virtual reality headset, rather than a standard screen interface. Finally, we also present a framework for incorporating predictive assistance into our system, which [24] did not consider.

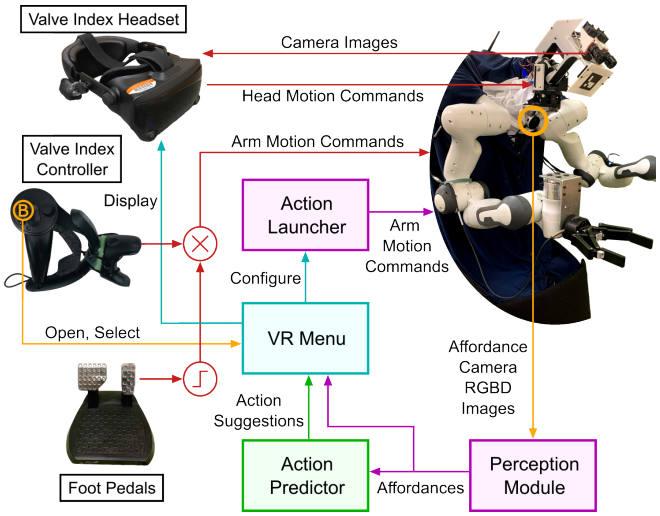


Fig. 2: System diagram showing how different interface elements control the robot. Operators use their own head and hand to control the robot's head and hand, and use a button on their hand controller to interact with the assistive menu. The Perception Module detects affordances in the environment to display possible assistive actions to the operator. [Best viewed in color.]

III. Interface Design

Suppose that an avatar robot has a library of assistive actions available which may include shared control and semi-autonomous actions. The key design question is *how to let the operator access and configure assistive actions without breaking immersion and maintaining or enhancing fluency?* Our approach is designed to satisfy the following objectives:

- O1. The operator must be able to quickly switch between direct, shared, and autonomous control modes.
- O2. The same control and feedback interfaces must be used for each level of control.
- O3. The operator should be able to see as much of the remote environment as possible even when configuring assistive actions.
- O4. The robot should determine which target objects for actions are available dynamically, i.e., from open-world perception applied to the robot's current context.
- O5. The interface should have a limited number of displays and widgets to minimize operator overload and facilitate faster learning.

We build our work on the TRINA avatar system [6], in which the robot is comprised of two Franka Emika Panda arms, a Robotiq 2F-140 parallel-jaw gripper, an anthropomorphic Psyonic Ability Hand, a Waypoint Vector omnidirectional wheeled base, and a custom-built three DoF neck and head assembly. A human operator controls TRINA using a virtual reality (VR) head-mounted display (HMD) that shows the view of TRINA's environment from stereo head cameras. They control the robot's head directly via HMD motion and use VR controllers to move the arms. The operator station is connected to the Internet via Ethernet and the robot is connected via WiFi or an Ethernet tether.

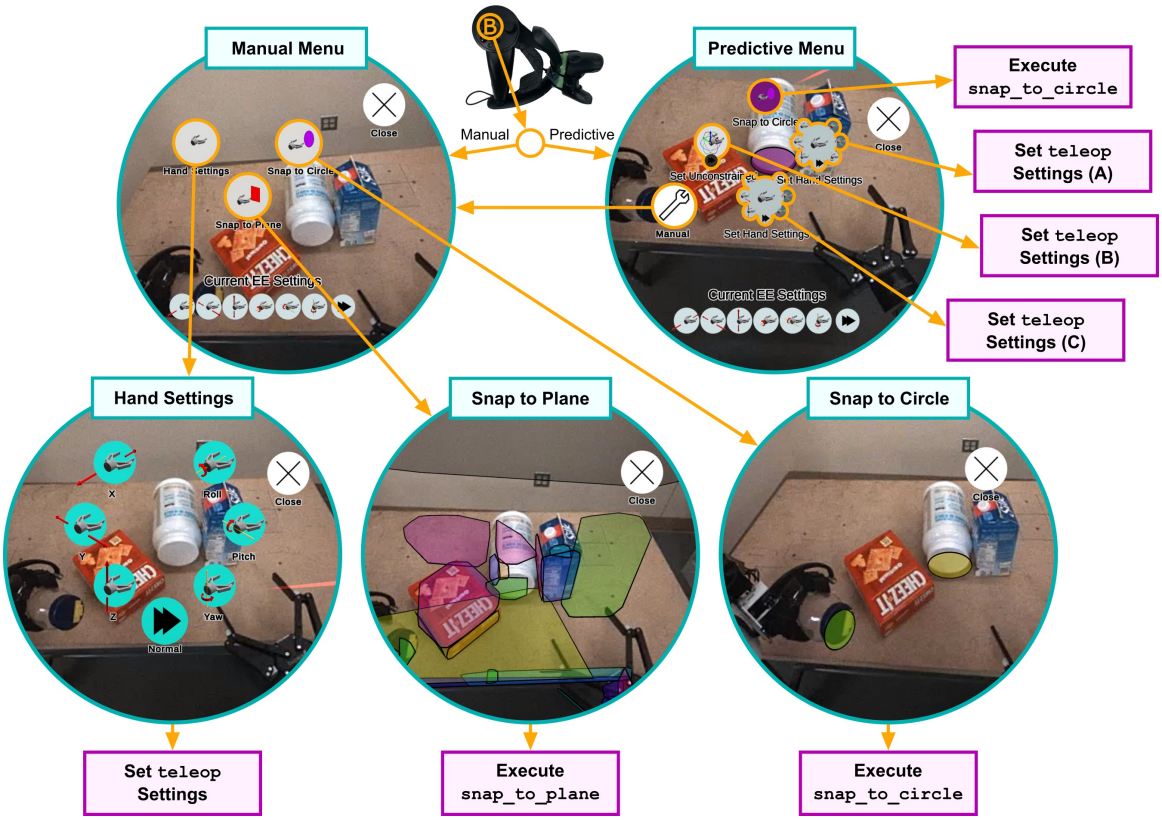


Fig. 3: Flow diagram showing how different menus are accessed. Depending on which interface type is being used, the B button will show the operator different interfaces: in manual mode, this button will directly show the manual menu, while in predictive mode, it will show the predictive menu. In the predictive menu shown here, each `teleop` icon gives the operator the option to choose a different set of constraints. Orange emphasis is added to highlight certain icons, and is not present in the actual menu. [Best viewed in color.]

Fig. 2 illustrates the major components of the proposed interface. Specifically, to satisfy O1 and O2, action selection functions are triggered with a single controller button. To satisfy O3, an unobtrusive VR Menu with a hierarchical pie system is overlaid atop the camera feed to configure and launch actions. For O4, the Perception Module continually recognizes geometric affordances in the robot's environment, which are rendered as selectable AR objects. For O5, we incorporate a machine learning-based Action Predictor to generate a Predictive Menu trained on expert demonstrations.

A. Direct Teleoperation (DT)

The default control mode is the direct teleoperation scheme described in [6]. To simplify novice operator training, in our experiments, we only activate the robot's right arm, parallel-jaw gripper, and head. The operator wears a VR HMD and the robot's head tracks the operator's head orientation. The operator uses a clutched system to control the arm: while holding down a foot pedal, the operator moves a VR controller, shown in Fig. 2, to move the robot's hand target. This motion is computed relative to the controller's pose when the operator first presses the pedal. A lower-level controller then attempts to reach this target. The operator can also velocity-control the parallel-jaw gripper using a joystick on the controller, pushing it right to inch the gripper closed, and left to inch it open.

The robot estimates the net force applied to its end effector to provide force feedback via two modalities: First, the controller vibrates with an intensity proportional to the estimated force magnitude (clipped between 10 and 30 N). Second, a virtual red hemisphere around the operator's controller shows the direction of the applied force, and becomes more opaque as the magnitude of the force increases.

B. Manual Menu (MM)

Using the direct teleoperation interface alone, operators can achieve some manipulation tasks [6], but complicated tasks, such as writing, are still quite difficult. To aid the operator, we created an interface to allow them to execute assistive actions. Guided by previous research [17], we designed a hierarchical pie menu fixed to the operator's head, shown in Fig. 3. By making the menu hierarchical, we minimize the number of simultaneously displayed icons to keep the operator's view of the remote environment unobstructed. The operator interacts with the menu using a "laser pointer" emanating from their controller to point at different icons, and clicks the B button on their controller to select them. The operator can bring up this menu by clicking the B button at any time and can close it by selecting the "Close" icon. This menu design allows the operator to configure the menu using the same interface they use to provide low-level commands to the robot, eliminating any need to switch between interfaces during operation. Clicking other icons gives the operator access to different submenus.

The "Hand Settings" submenu allows the operator to edit constraints and the sensitivity mode of the arm by selecting

any of the icons to toggle their state. The "Snap to Plane" and "Snap to Circle" submenus display the most recently detected affordances of each type, shown in Fig. 3. Each affordance is rendered as an AR object in the virtual world, displayed so that it appears aligned with the object it was detected from, with a random hue at 30% opacity. By performing this alignment, the menu leaves the operator's view essentially unobstructed, integrating information about affordances with the operator's existing view of the environment. When the operator hovers over an affordance with their laser pointer, that affordance becomes opaque. Selecting an affordance will send it to the robot, which will then execute the corresponding action.

Whenever the operator selects an action, "Executing Action" followed by "Action Succeeded" or "Action Failed" is displayed depending on its status. If an action fails, the arm maintains the position it had when the failure occurred. The operator can also cancel actions by pressing their foot pedal, which gives them direct control over the arm as usual.

C. Predictive Menu (PM)

While the manual menu provides access to all possible actions, it can be overwhelming and slow, especially for novice users. To alleviate this, we designed a third interface that uses an action predictor, described in section V, to predict the operator's intent and present them with a reduced menu that only includes the four most likely actions. If the operator's desired action is not in this set, they can still access the manual menu as a fallback. With this menu, when the operator clicks B, the top four actions are shown instead of the manual menu, as shown in Fig. 1c and Fig. 3. Whenever the operator hovers over an icon corresponding to an action, all other icons (and affordances) dim to 10% opacity. Selecting any icon closes the menu and sends the action to the robot which then executes it.

We assume that the robot is the only agent in the scene and that all manipulations are quasistatic. As a result, the state of the world only changes when the robot is executing an action. Therefore, we design the robot to run the action predictor to produce the next set of suggestions when it first starts up, and after any action is completed. While these assumptions do not strictly hold in all experiments, they are good enough approximations to produce accurate predictions while not having to compute new predictions in every frame.

IV. Assistive Actions

We implemented three kinds of assistive actions: constrained teleoperation, snapping to planes, and snapping to circles. The use of geometric affordances to provide assistance allows the use of these actions in an open-world context, where the semantic meaning of objects in the environment is unknown. The constrained teleoperation and plane snapping actions were previously described in [23], and so are only briefly covered here.

The constrained teleoperation action, `teleop(sens, x, y, z, roll, pitch, yaw)` accepts 7 Boolean parameters modifying the operator's direct control of the arm. During this action, the operator controls the gripper's target pose by moving a VR controller with their own arm. When the `sens` parameter is `true`, the arm's end-effector motion is isotropically scaled to 0.25 of the operator's input motion to enable precise manipulation. The remaining parameters toggle constraints on the end-effector motion, activating guidance virtual fixtures to simplify operation [1].

The plane snapping action, `snap_to_plane(p)` accepts a plane detected from a point-cloud of the environment by a clustering method [9]. This point-cloud is sensed by the "affordance camera" shown in Fig. 2, an Intel RealSense L515 mounted below the robot's neck, pointed at the center of the robot's workspace. The plane extraction algorithm updates the set of detected planes once every 5 seconds. This action aligns the forward direction of the gripper with the normal of the detected plane and moves it so that its tool tip is d_s m away from the plane to prepare the operator to perform manipulation on or near the plane's surface. For the tasks considered here we found $d_s = 0.15$ m to work well. Fig. 4 illustrates this process in 2D. The robot uses a sampling-based planner to find a path to reach this target or reports that no path was found after 10 s.

Lastly, the `snap_to_circle(c)` action accepts a circle detected from the environment, aligns the gripper's forward direction with the circle's axis, and centers the gripper on the circle to prepare the operator to perform rotating manipulations about the circle's axis. Our system detects circles from RGBD images from the affordance camera once every 5 seconds. The system segments the RGB image using the Segment Anything Model (SAM) [16] and converts the RGBD image into a point-cloud. For each image mask, the corresponding points are selected, and the plane supported by the most points is found. The inliers of this plane are computed as the points in the mask within $d_{\text{in}} = 5$ mm of the plane and projected to the plane. The convex hull of these projected points is found and the circle is discarded if this hull's "circularity" ($\frac{4\pi \cdot \text{Area}}{\text{Perimeter}^2}$ [2]) is below $c_{\text{min}} = 0.9$. The minimum enclosing circle of the hull is computed and circles with radii greater than $r_{\text{max}} = 7$ cm are discarded. To remove duplicates, this candidate circle is compared against previously detected circles. Circles are considered similar if the masks from which they were detected overlap, their centers are within $\Delta_{\text{c}} = 5$ cm, and their radii are within $\Delta_{\text{rad}} = 1$ cm. Among similar circles, the one with the largest ratio of inliers to points in the mask is kept. Once a circle has been selected, the robot computes a target end-effector pose in the same manner as the `snap_to_plane` action, additionally moving the target so that the projection of the tool tip to the plane of the circle coincides with the circle's center. Fig. 4 demonstrates this action in 2D.

V. Intent Prediction

To populate the predictive menu, we require an action predictor that can predict multiple likely actions. Additionally, since the set of affordances is not known until runtime, the predictor must be open-world, i.e. able to predict over an open set of objects. We employ the structured prediction

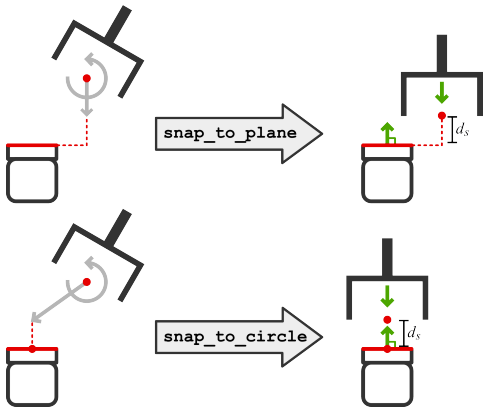


Fig. 4: 2D illustration of the `snap_to_plane` and `snap_to_circle` actions. Both align TRINA's gripper with the normal of the selected affordance, but `snap_to_circle` centers the gripper on the circle while `snap_to_plane` only moves it closer to the plane. Here we used $d_s = 0.15$ m. [Best viewed in color.]

method of [23] as it was found to have strong performance in open-world scenarios on similar tasks.

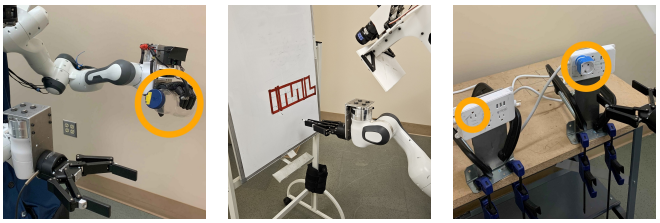
Actions are defined by a type and a collection of parameters, $\overline{\psi}$, which may be different for each action type. We limit the set of n types *a priori* and dynamically detect the set of feasible parameters for each type, corresponding to detected affordances. To predict an action given the robot's current context vector, x, the method uses n parameter scoring neural networks, $\{G^{(i)}(x, \overline{\psi})\}_{i=1}^{n}$, and an action network, $A(x)$. $A(x)$ produces an n-dimensional output vector with each element representing the overall score for an action type. Each $G^{(i)}(x, \overline{\psi})$ predicts a scalar score for parameter collections of a particular action type. To score a complete action, the appropriate scores are summed, $s = e_i^\intercal A(x) + G^{(i)}(x, \overline{\psi})$, where e_i is the ith standard basis vector.

To train and evaluate our predictor, three expert operators (paper authors) collected a dataset of 150 action sequences across three different tasks: unscrewing a jar lid, writing "IML" on a whiteboard, and plugging a cord into an electrical socket. Each sequence was collected in a highly cluttered environment that contained many different distractor objects with varied compositions and arrangements. The specific target objects used were also modified (for example, varying which jars were used). The scoring function was trained using a maximum margin loss function to output high scores for actions observed in the demonstrations [23].

VI. Experiments

Human subjects studies were conducted to evaluate differences between the DT, MM, and PM interfaces. All procedures were reviewed and approved by the UIUC IRB on Feb. 20, 2023. We formulated the following *a priori* hypotheses about the system:

- **H1**: There is a difference in the proportion of tasks operators complete when using each interface.
- **H2**: There is a difference in the operators' total task completion times when using each interface.
- **H3**: There is a difference in the operator's sense of presence when using each interface.



(a) Jar (b) Whiteboard (c) Plug

Fig. 5: The three testing tasks. Target objects are highlighted with orange circles. [Best viewed in color.]

To test our hypotheses, we designed a human subjects study to test novices' use of each interface. We considered three tasks: unscrewing a jar lid held in TRINA's left hand, writing "IML" on a whiteboard, and plugging in an electrical plug. Setups for these tasks are shown in Fig. 5. The predictor was trained on expert demonstrations of the same tasks. These tasks were chosen to be representative of multi-stage tasks in which assistance is useful but solution strategies are somewhat flexible; novice strategies can differ significantly from one another and the expert demonstrations.

We recruited 20 student participants from the University of Illinois at Urbana-Champaign campus, 19 of whom completed the entire procedure. One participant requested to end the experiment during training due to nausea. Of the 19 participants, 11 were male, 7 were female, and one preferred not to say. Subjects were of age 19–32 (mean: 24) and self-reported their familiarity with robotics and controlling robots on average as 5.4 and 4.4 on a 7-point Likert scale [27] respectively. None of the subjects had used TRINA before.

1) Basic Training: Subjects were trained to use the direct teleoperation interface and were introduced to several possible fault states. For example, if excessive force was applied to the arm, the subject would momentarily lose control of it. Subjects were given suggestions about how to resolve each of these faults. The assistive functionalities were demonstrated using the manual (MM) and predictive (PM) menus.

2) Task Introduction: Subjects were shown the three testing tasks and completed the tasks in-person to familiarize themselves with the specific features of the target objects. A researcher explained how task completion would be graded, and that subjects should try to complete tasks as quickly as possible with 5 min at most for each task. For the jar, the task was completed when the lid no longer was touching the jar body. For the whiteboard, the required writing was split into 19 segments and credit was given for each completed segment. For the plug, the task was completed when the subject had fully inserted the plug into the target socket.

3) Training Tasks: Subjects were coached through using the MM and PM on two training tasks which demonstrated each of the assistive actions in context. In the first task, a researcher handed TRINA a capped Expo marker, and the subject had to use TRINA to insert the tip of the marker into a square hole. Subjects were told to snap to the plane of the hole and turn off all rotational DoFs before inserting the marker into the hole. In the second task, subjects had to grasp and turn a dial for three full rotations. They were instructed to first snap to the circle of the dial, disable all

TABLE I: Differences between each interface across all tasks. *, **, and *** denote $p \leq 0.05$, $p \leq 0.01$, and $p \leq 0.001$ respectively.

	Condition	Success (%) (↑)	Time (s) (↓)	Usability (↑)	Workload (↓)	Presence (↑)
Avg ± Std	DT	42.7 ± 30.5	756 ± 179	4.32 ± 1.04	5.29 ± 1.14	4.53 ± 1.68
	MM	68.5 ± 28.9	672 ± 183	**5.17 ± 0.56**	4.21 ± 1.26	**5.00 ± 1.29**
	PM	**75.8 ± 24.2**	**650 ± 152**	5.01 ± 0.74	**3.90 ± 1.13**	4.74 ± 1.33
Friedman W-Score		0.4014	0.2696	0.2647	0.3836	0.0269
Friedman p-value		***0.0005	**0.0060	**0.0065	***0.0007	0.6004
Post-hoc p-value	DT vs. MM	**0.0066	0.0611	**0.0015	**0.0053	0.1308
	DT vs. PM	***0.0004	*0.0115	**0.0061	***0.0004	0.5202
	MM vs. PM	0.4844	0.5412	0.2882	0.2958	0.3543

but the `x` and `roll` DoFs to grasp the dial, and finally have only `roll` enabled to turn the dial.

4) Testing Procedure: On average, training took ∼90 min. After training, the order of conditions (DT, MM, and PM) was randomized. For each condition, subjects completed the tasks in the order of jar, whiteboard, then plug. Subjects were given 3 and 1 min remaining warnings. To minimize variance between the subjects, the placement of the target objects in the scene was kept consistent, and there were no distractor objects. Additionally, the jar and plug were modified to make the tasks slightly easier for novices: bright tape was added to the lid of the jar, and a socket adapter was used as the plug instead of an electrical cord. Blue tape was also added to the adapter to make it easier to see. After attempting all of the tasks in a given condition, subjects filled out a questionnaire about their experience, measuring the system's usability [3], the subject's workload [10], and their self-reported feeling of presence in the remote environment. All questions were rated on a 7-point Likert scale. Subjects would then immediately proceed to the next condition.

VII. Results and Discussion

Subject performance was measured by the proportion of tasks completed and the time taken. Success metrics are computed as $(\text{Did jar} + \text{Segments completed}/19 + \text{Did plug})/3$. If a subject failed a task early, their time was recorded as the maximum time. We ran a Shapiro–Wilk test [30] on the performance metrics for each condition and found significant deviations from normality. To test **H1**, **H2**, and **H3** we ran separate Friedman tests [27] on the subjects' success rates, completion times, and reported senses of presence, which revealed significant differences between the conditions for success rates ($p = 0.0005$) and completion times ($p = 0.0060$), but not for senses of presence ($p = 0.6004$). Post-hoc pairwise two-sided Wilcoxon-signed-rank testing [27] found a significant increase in success rate for DT vs. MM ($M = 25.9\%, SD = 33.6\%, p = 0.0066$) and DT vs. PM ($M = 33.1\%, SD = 26.6\%, p = 0.0004$), and a decrease in completion time for DT vs. PM ($M = 105$ s$, SD = 170$ s$, p = 0.0115$). Table I shows these results and includes results of exploratory analysis performed on other subjective measures, indicating that the presented interfaces also improve usability and workload.

These results provide support for **H1** and **H2**, indicating that the presented system can significantly improve novice operators' ability to perform several tasks quickly and accurately. We also found that the predictive menu generally has a larger impact on both objective and subjective metrics than the manual menu, despite its relatively low accuracy of 60.% on novice actions. We expect this impact to further increase as the number of possible actions and the accuracy of the predictor rise. The lack of support for **H3** suggests that this menu system preserves the operator's sense of presence despite introducing non-physical visual elements; in fact, both MM and PM received higher average presence scores than DT. We attribute this to the minimally invasive nature of the hierarchical pie menu and affordances registered to the remote environment. We further found that both the MM and PM interfaces tend to increase the system's usability and decrease the operator's workload. Users can easily understand how to interact with both kinds of menus and use them to decrease the required cognitive effort to complete manipulation tasks.

Our results show that contrary to conventional wisdom, designers of avatar robots need not choose between an immersive interface and using shared control: it is possible to achieve both in a single system. When integrating these two control paradigms, we suggest designers follow the philosophy presented here. For example, for shared control actions that reference the robot's environment, directly overlaying visual elements corresponding to those actions onto the operator's existing view lets the operator launch those actions while still focusing on their desired task. The manual menu presented here keeps the number of simultaneously presented icons low using a hierarchy, and this can be further improved for systems with large numbers of actions by using a predictive menu.

VIII. Conclusion

Our unified interface demonstrates a route for robot avatars to harness the "best of both worlds" between immersive teleoperation and assistive actions. Our interface gives avatar operators intuitive access to assistive actions with dynamic affordance detection and AR overlays in an unobtrusive menu, and experiments showed that our approach improves operator fluency on three multi-step tasks without degrading immersion. In future work, we would like to expand the set of assistive actions to include automatic grasping and tool-centric shared control. We also wish to study how the interface affects operator performance in longer-form tasks, and to develop action predictors that adapt to individual operators online.

REFERENCES

[1] S. A. Bowyer, B. L. Davies, and F. Rodriguez y Baena, "Active Constraints/Virtual Fixtures: A Survey," *IEEE Trans. Robotics*, vol. 30, no. 1, pp. 138–157, Feb. 2014.

[2] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[3] J. Brooke, "Sus: A quick and dirty usability scale," *Usability evaluation in industry*, vol. 189, no. 3, pp. 189–194, 1996.

[4] J. van Bruggen, C. Brekelmans, R. Lieftink, D. Dresscher, and J. van Erp, "I-botics avatar system: Towards robotic embodiment," Jun. 2023.

[5] S. Bustamante, G. Quere, D. Leidner, J. Vogel, and F. Stulp, "CATs: Task Planning for Shared Control of Assistive Robots with Variable Autonomy," in *2022 International Conference on Robotics and Automation (ICRA)*, May 2022, pp. 3775–3782.

[6] J. M. Correia Marques, P. Naughton, J.-C. Peng, Y. Zhu, J. S. Nam, Q. Kong, X. Zhang, A. Penmetcha, R. Ji, N. Fu, V. Ravibaskar, R. Yan, N. Malhotra, and K. Hauser, "Immersive Commodity Telepresence with the AVATRINA Robot Avatar," en, *International Journal of Social Robotics*, Jan. 2024. [Online]. Available: https://link.springer.com/10.1007/s12369-023-01090-1.

[7] A. D. Dragan, S. Siddhartha Srinivasa, and K. Kenton Lee, "Teleoperation with Intelligent and Customizable Interfaces," *J. Human-Robot Interaction*, vol. 2, no. 2, pp. 33–79, Jun. 2013.

[8] M. Fallon, S. Kuindersma, S. Karumanchi, M. Antone, T. Schneider, H. Dai, C. P. D'Arpino, R. Deits, M. DiCicco, D. Fourie, *et al.*, "An architecture for online affordance-based perception and whole-body planning," *J. Field Robotics*, vol. 32, no. 2, pp. 229–254, 2015.

[9] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast plane extraction in organized point clouds using agglomerative hierarchical clustering," in *IEEE Int. Conf. Robotics and Automation*, May 2014, pp. 6218–6225.

[10] S. G. Hart and L. E. Staveland, "Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research," en, in *Advances in Psychology*, vol. 52, Elsevier, 1988, pp. 139–183. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0166411508623869 (visited on 07/27/2022).

[11] K. Hauser, "Recognition, prediction, and planning for assisted teleoperation of freeform tasks," *Autonomous Robots*, vol. 35, no. 4, pp. 241–254, 2013.

[12] K. Hauser, E. Watson, J. Bae, J. Bankston, S. Behnke, B. Borgia, M. G. Catalano, S. Dafarra, J. B. F. Van Erp, T. Ferris, J. Fishel, G. Hoffman, S. Ivaldi, F. Kanehiro, A. Kheddar, G. Lannuzel, J. F. Morie, P. Naughton, S. NGuyen, P. Oh, T. Padir, J. Pippine, J. Park, J. Vaz, D. Pucci, P. Whitney, P. Wu, and D. Locke, "Analysis and Perspectives on the ANA Avatar XPRIZE Competition," en, *International Journal of Social Robotics*, Jan. 2024. [Online]. Available: https://link.springer.com/10.1007/s12369-023-01095-w.

[13] K. Huang, D. Chitrakar, F. Rydén, and H. J. Chizeck, "Evaluation of haptic guidance virtual fixtures and 3D visualization methods in telemanipulation—a user study," en, *Intelligent Service Robotics*, vol. 12, no. 4, pp. 289–301, Oct. 2019. [Online]. Available: http://link.springer.com/10.1007/s11370-019-00283-w.

[14] S. Javdani, S. Srinivasa, and A. Bagnell, "Shared Autonomy via Hindsight Optimization," in *Robotics: Science and Systems XI*, Robotics: Science and Systems Foundation, Jul. 2015. [Online]. Available: http://www.roboticsproceedings.org/rss11/p32.pdf.

[15] H. J. Jeon, D. Losey, and D. Sadigh, "Shared Autonomy with Learned Latent Actions," in *Proceedings of Robotics: Science and Systems*, Corvalis, Oregon, USA, Jul. 2020.

[16] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.

[17] R. Komerska and C. Ware, "A study of haptic linear and pie menus in a 3D fish tank VR environment," en, in *12th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2004. HAPTICS '04. Proceedings.*, Chicago, IL, USA: IEEE, 2004, pp. 224–231. [Online]. Available: http://ieeexplore.ieee.org/document/1287200/.

[18] A. Leeper, K. Hsiao, M. Ciocarlie, I. Sucan, and K. Salisbury, "Methods for collision-free arm teleoperation in clutter using constraints from 3D sensor data," in *IEEE-RAS Int. Conf. Humanoid Robots*, Oct. 2013, pp. 520–527.

[19] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *J. Human-Robot Interaction*, ACM Press, 2012, pp. 1–8.

[20] G. Li, Q. Li, C. Yang, Y. Su, Z. Yuan, and X. Wu, "The Classification and New Trends of Shared Control Strategies in Telerobotic Systems: A Survey," *IEEE Transactions on Haptics*, vol. 16, no. 2, pp. 118–133, Apr. 2023.

[21] R. Luo, C. Wang, C. Keil, D. Nguyen, H. Mayne, S. Alt, E. Schwarm, E. Mendoza, T. Padır, and J. P. Whitney, *Team Northeastern's Approach to ANA XPRIZE Avatar Final Testing: A Holistic Approach to Telepresence and Lessons Learned*, Mar. 2023. [Online]. Available: http://arxiv.org/abs/2303.04932.

[22] J. M. Marques, J.-C. Peng, P. Naughton, Y. Zhu, S. Nam, and K. Hauser, "Commodity Telepresence with the AvaTRINA Nursebot in the ANA Avatar XPRIZE Finals," in *ICRA 2nd Workshop Toward Robot Avatars*, Jun. 2023.

[23] P. Naughton and K. Hauser, "Structured Action Prediction for Teleoperation in Open Worlds," en, *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3099–3105, Apr. 2022. [Online]. Available: https://ieeexplore.ieee.org/document/9691823/.

[24] V. Pruks and J.-H. Ryu, "Method for generating real-time interactive virtual fixture for shared teleoperation in unknown environments," *The International Journal of Robotics Research*, p. 02 783 649 221 102 980, Jun. 2022. [Online]. Available: https://doi.org/10.1177/02783649221102980.

[25] G. Quere, A. Hagengruber, M. Iskandar, S. Bustamante, D. Leidner, F. Stulp, and J. Vogel, "Shared control templates for assistive robotics," in *IEEE Int. Conf. Robotics and Automation*, 2020, pp. 1956–1962.

[26] L. Rosenberg, "Virtual fixtures: Perceptual tools for telerobotic manipulation," in *Proceedings of IEEE Virtual Reality Annual International Symposium*, Sep. 1993, pp. 76–82.

[27] S. Sarantakos, *Social research*. Bloomsbury Publishing, 2017.

[28] M. Schwarz, C. Lenz, R. Memmesheimer, B. Pätzold, A. Rochow, M. Schreiber, and S. Behnke, *Robust Immersive Telepresence and Mobile Telemanipulation: NimbRo wins ANA Avatar XPRIZE Finals*, Mar. 2023. [Online]. Available: http://arxiv.org/abs/2303.03297.

[29] M. Schwarz, C. Lenz, A. Rochow, M. Schreiber, and S. Behnke, "NimbRo Avatar: Interactive Immersive Telepresence with Force-Feedback Telemanipulation," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Prague, Czech Republic: IEEE, Sep. 2021, pp. 5312–5319. [Online]. Available: https://ieeexplore.ieee.org/document/9636191/.

[30] S. S. Shapiro and M. B. Wilk, "An Analysis of Variance Test for Normality (Complete Samples)," *Biometrika*, vol. 52, no. 3/4, pp. 591–611, 1965. [Online]. Available: https://www.jstor.org/stable/2333709.

[31] C. Wang, S. Huber, S. Coros, and R. Poranne, "Task autocorrection for immersive teleoperation," in *IEEE Int. Conf. Robotics and Automation*, May 2021.

[32] S. N. Young and J. M. Peschel, "Review of Human–Machine Interfaces for Small Unmanned Systems With Robotic Manipulators," *IEEE Transactions on Human-Machine Systems*, vol. 50, no. 2, pp. 131–143, Apr. 2020.