2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 14-18, 2024. Abu Dhabi, UAE

Fast Explicit-Input Assistance for Teleoperation in Clutter

Nick Walker[1], Xuning Yang[2], Animesh Garg[2,3], Maya Cakmak[1], Dieter Fox[1,2], Claudia Pérez-D'Arpino[2]

***Abstract*— The performance of prediction-based assistance for robot teleoperation degrades in unseen or goal-rich environments due to incorrect or quickly-changing intent inferences. Poor predictions can confuse operators or cause them to change their control input to implicitly signal their goal. We present a new assistance interface for robotic manipulation where an operator can explicitly communicate a manipulation goal by pointing the end-effector. The pointing target specifies a region for local pose generation and optimization, providing interactive control over grasp and placement pose candidates. We evaluate this explicit pointing interface against an implicit inference-based assistance scheme and an unassisted control condition in a within-subjects user study (N=20), where participants teleoperate a simulated robot to complete a multi-step singulation and stacking task in cluttered environments. We find that operators prefer the explicit interface, experience fewer pick failures and report lower cognitive workload. Our code is available at: `github.com/NVlabs/fast-explicit-teleop`.**

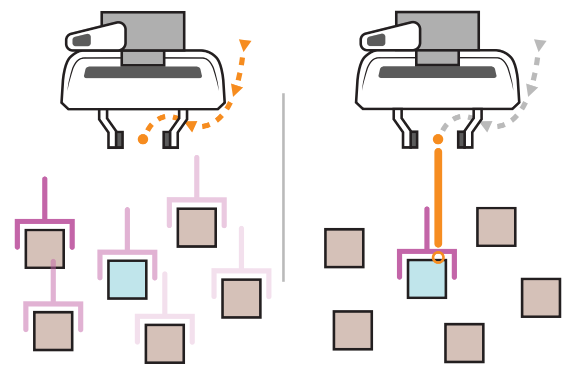

Fig. 1: Implicit assistance (left) funnels the operator toward the goal predicted based on (for instance) the recent trajectory. The operator is not intended to change their input to influence the assistance. Explicit assistance (right) affords the operator direct control over the inferred goal by pointing the gripper toward the object of interest. A local optimization selects a feasible, collision free pose.

I. Introduction

Robot telemanipulation is widely useful but demanding, even for skilled operators. Acting in the world through a foreign embodiment with limited perception requires the user to reason not only about the task at hand but about the abilities and limitations of the robot, as well as the state of the environment. Assistive teleoperation interfaces can reduce this burden by automating parts of the robot's behavior, increasing safety and comfort for everyone from operators conducting tight-tolerance assembly in manufacturing to home users of assistive robots. Teleoperation is also being used for data collection of human demonstrations, both with simulated [1–4] and real robots [5–7], to build datasets for use with imitation learning [1,8] and offline reinforcement learning [9,10]. Improvements to interfaces are required in order to make on-line teleoperation faster and more intuitive, as well as to improve the quality of trajectories for robot learning [11,12]. Grasping and placing objects precisely and smoothly is still difficult for operators due to perception and haptic gaps [13,14]. Grasps often fail when small clearances aren't respected, and the limited visual cues afforded to operators can cause them to press objects down further than necessary when placing.

The foundation of most assistive teleoperation systems is prediction [15–17]. Inferring, for instance, the operator's desired trajectory or end-effector goal based on their recent trajectory and context (i.e. scene, object, task) enables the automation of subsequent actions. Performant prediction systems can engage assistance fluently in proportion to their own confidence. The user teleoperates as they would without assistance. Their control over the predictions is *implicit*, arising from how their state or actions correspond with some model of possible intended behavior.

But the benefits of implicit inference-based assistance are difficult to realize in practice. Human environments pose challenges for on-line trajectory and goal prediction [18,19]. In clutter, where there are numerous possible target objects in close physical proximity, it is inherently difficult to predict manipulation targets as many goals may be consistent with a user's state or historical input. Poor predictions can lead the operator to modify their behavior in an attempt to better signal their goal—a confusing interaction, as the operator's mental model of the predictor is likely incorrect. In such situations, it is preferable to provide an *explicit* interface that accommodates the user's desire to exert direct control over the predicted intent. Explicit input interfaces usually involve modal goal-specification interactions which aren't suitable for on-line interaction [], or additional input modalities, like natural language, that introduce complexity and potentially increase burden.

Our proposed interface for pick and place manipulation, shown in Fig. 1, leverages "pointing" of the end effector as an explicit input method, requiring neither an additional input modality nor a modal interaction. The approach offers assistance for a possible grasp or placement pose via optimization in a small region around where a ray from the gripper to the target object (grasping) or from the object in the gripper (placing) meets the scene geometry. Parallel computation

* This work was done during an internship at NVIDIA.
[1] University of Washington, [2] NVIDIA. [3] Georgia Institute of Technology. nswalker@cs.uw.edu

allows the system to rank and filter many possible collision-free candidates and present suggestions that are responsive to the user's input at high frequency.

We implemented our proposed explicit interface on a simulated Franka Emika Panda robot and conducted a user study comparing it to an implicit-input assistive teleoperation method on pick-and-place stacking tasks with clutter. We find that operators prefer the explicit interface, experience fewer pick failures and report lower cognitive workload. Our implementation of explicit assistance and study conditions in NVIDIA Omniverse Isaac Sim is available at `github.com/NVlabs/fast-explicit-teleop`.

II. RELATED WORKS

The design space for assistive teleoperation spans various types of operator input, different forms of assistance and a spectrum of manual to automatic engagement [15].

Most assistive teleoperation methods use a form of implicit input to autonomously generate improved robot actions. Early methods maintained a probability distribution over possible goals given users' recent actions and overrode user control with actions more in line with optimal trajectories to the inferred goal [15–17]. When available, data enables the use of sophisticated predictive models like trajectory forecasting transformers [20] or multi-modal diffusion policies [21]. When a task reward is available, it is possible to use human-in-the-loop deep reinforcement learning [22]. While some of these methods produce assistance based only on the current state, user interactions with the assistance are characteristically implicit as the user is not intended to control the state with the aim of modifying the assistance.

Human-in-the-loop autonomous systems commonly allow operators to explicitly specify goals, preview generated trajectories and supervise execution [23,24]. Most frequently, these interfaces use keyboard and mouse control over 6DOF interactive pose markers, enabling precise goal specification at the expense of fluency, making them unsuitable on-line continuous teleoperation.

Assistance can also come in the form of augmented control input schemes. [25] used demonstrations to learn a task-specific low-dimensional control mapping, enabling operators to control a robot arm using only a 2D joystick. [26] showed that such task-specific mappings can also be generated conditionally based on a language description of a task in a way that also allows natural language corrections during execution.

Another approach is to dynamically constrain actions to, for example, avoid collisions with obstacles [27], or reject probable inadvertent input in a fine manipulation task. [28] introduced the concept of "virtual fixtures," registered geometric overlays, typically specified beforehand using task knowledge, which produce sensory cues or alter control behavior as operators move through them. These fixtures restrict motion within a region, like a virtual ruler confining end-effector motion to a line.

III. FAST EXPLICIT-INPUT ASSISTANCE

We are interested in generating actions to assist a teleoperator. Abstractly, the generation of these *actions*—which may be poses, configurations or trajectories—is the result of an optimization based on *state* information and *context*:

$$\text{actions} = \operatorname*{arg\,max}_{\text{option}\in\mathcal{A}} f(\text{option}, \text{state}, \text{context}) \tag{1}$$

The defining decisions we make about the implementation of Eq. 1 that result in an effective explicit-input interaction are:

- to use transparent and controllable state information;
- to prioritize smoothness of the assistance with respect to state in the selection of f. Both the average and maximum variations in assistance for small state changes affect usability, as abrupt changes can be disorienting;
- and to treat the resulting action as a suggestion subject to user review and refinement.

Conventional inference-based assistance systems attempt to represent the space of possible goal poses or next-actions in $\mathcal{A}$. They select for f a model of the likelihood of the goal conditioned on the pose or recent trajectory of the robot.

We similarly choose to produce useful poses for the operator, but we disregard the opaque history of the operator's actions and instead rely on immediately controllable present-state information. We leverage an intuitive "pointing" metaphor to allow the user to specify the anchor for a local optimization of an assistance pose. We define the optimization to be amenable to parallelization, ensuring it can compute at interactive speeds. The result is a pose suggestion that the user can ignore or modify by pointing the gripper before affirmatively accepting.

A. Pointing as Ray Control

Our experience is that the most understandable and controllable aspect of state is where the end-effector is pointing. Although pointing is governed by all six degrees of freedom (DoF) of the end effector pose—which we denote as $\boldsymbol{e}_e \in \mathrm{SE}(3)$ located between the fingers—it particularly emphasizes control of the axis component $\boldsymbol{v} \in R^3$ of the axis-angle $(\boldsymbol{v}, \theta)$ representation of the SO(3) orientation. For convenience, we will also leverage the $R^{4\times4}$ transformation matrix representation of the pose $\boldsymbol{e}_e$ consisting of rotation matrix $\boldsymbol{R} = [\boldsymbol{r}_x, \boldsymbol{r}_y, \boldsymbol{r}_z] \in R^{3\times3}$ and translation component $\boldsymbol{p}_e \in R^3$. We assign the $\boldsymbol{r}_z$ component outwards from the gripper, $\boldsymbol{r}_y$ perpendicular and along the axis of closing, and $\boldsymbol{r}_x$ perpendicular and away from the gripper camera. These axes are labeled on Fig. 2.

Pointing the axis $\boldsymbol{r}_z$ is familiar for operators not only because it is a necessary component of most manipulation tasks, but also because it is a means to change the view of the "eye-in-hand" camera that is often available. When unobstructed, this view is an innate visualization of the pointing input upon which crosshairs or a rendered lines can directly show the pointing axis.

The pointing target is the point $\boldsymbol{p}_t \in R^3$ at which the ray extending from the end effector position $\boldsymbol{p}_e$ along $\boldsymbol{r}_z$

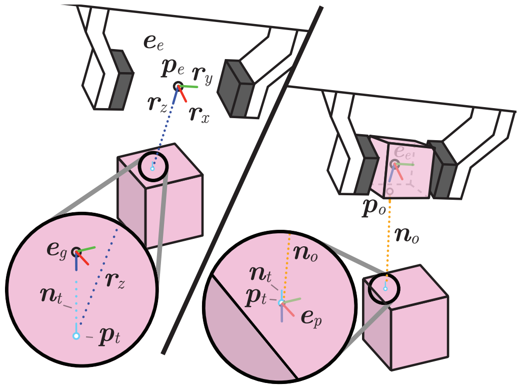

Fig. 2: Our realizations of explicit grasping (left) and placing assistance (right) both center on the interaction of a ray from the gripper with scene geometry. A projected anchor pose is calculated then used to select amongst a set of candidate assistance poses.

contacts the scene geometry, and $\boldsymbol{n}_t$ is the surface normal at the target point. These quantities can be approximated using depth data or a geometric representation that the robot has access to. They are simple to visualize by (for example) highlighting the point in a 2D view and drawing a tick mark in the normal direction.

Previous works have focused on the proximity of the end effector to assistance candidates, but relative distinctions in distance are difficult to assess based on a remote 2D view. Proximity is chiefly a function of the 3DoF end effector position $\boldsymbol{p}_e$, whereas the axis $\boldsymbol{v}$ of the gripper orientation is characterized by just the 2 spherical coordinates, azimuth and elevation. Pointing does still usefully encode a nearness bias, since the area at which one can point at a given object is inversely proportional to the square of the distance to the object. In other words, it is easy to point at things nearby, grows more difficult for things further away, and quickly becomes effectively impossible beyond a point. This bias is reinforced by the fact that one can only point at what can be seen and further objects are subject to greater occlusion.

B. Grasp Pointing

In order to suggest a possible grasp pose to the operator using our pointing interface we must define a mapping from the 6D pointing control to a 6D grasp pose. We denote the final grasp assistance suggestion as $\boldsymbol{e}_g^*$.

A direct mapping would be to simply displace the current gripper orientation along the ray to some fixed offset of the target point, making no modification to the gripper orientation. However, our experience is that users often point at oblique angles but nonetheless desire an approach orthogonal to the object surface. Instead of using $\boldsymbol{r}_z$, we use the negation of the surface normal $\boldsymbol{n}_t$ at the target point $\boldsymbol{p}_t$.

Users generally expect the angle θ about the axis to be the one that is "most similar" to their current orientation. To encode this geometrically, we project a reference vector anchored to the gripper onto the plane defined by the intersection point $\boldsymbol{p}_t$ and the normal vector $\boldsymbol{n}_t$. Any reference vector may be selected, however it is preferable to use one that is unlikely to be perpendicular to the plane, like $\boldsymbol{r}_x$ or $\boldsymbol{r}_y$. The minimal rotation is the geodesic between the current and the projected reference vector.

The resulting grasp anchor pose $\boldsymbol{e}_g$ provides an intuitive, cursor-like interaction when the gripper ray is swept across the scene. It is unlikely to be a satisfactory grasp on its own, however, because an orthogonal approach may be inappropriate for the object, or the position may cause contact with the object or other scene geometry. A generative grasp model can be used to provide a set $\mathcal{A}_{(\boldsymbol{e}_g)}$ of candidates near the anchor. The specification of "near" governs the smoothness of the assistance interaction, with smaller thresholds ensuring that the resulting poses do not change substantially as the cursor moves but necessarily excluding more suitable grasps that are too far away. Each candidate can be computed and scored independently, making this step highly parallelizable. The result nearest the anchor should be taken as grasping suggestion $\boldsymbol{e}_g^*$. Generally the quality and smoothness of the assistance improve as more candidates are considered so long as the computation runs at interactive rates.

C. Placement Pointing

As with grasp pointing, we seek a mapping from the 6D pointing control to a 6D end-effector placement pose, $\boldsymbol{e}_p^*$.

The object may have been grasped in an arbitrary orientation, so a direct mapping that translates the current pose along the gripper axis $\boldsymbol{r}_z$ toward the target point is unlikely to be useful for stably placing the object. Instead, we observe that the object was likely picked from a stable pose where it rested on a support facet defined by some point $\boldsymbol{p}_o$ and normal $\boldsymbol{n}_o$ pointing in the gravity direction. At the moment of the pick, the orientation of normal $\boldsymbol{n}_o$ can be recorded with respect to the end effector pose $\boldsymbol{e}_e$, and a point $\boldsymbol{p}_o$ on the object facet can be estimated by projecting the end-effector position $\boldsymbol{p}_e$ at the moment of the pick onto the scene geometry revealed after the object is lifted.

It is now intuitive to map the control of the resulting plane $(\boldsymbol{p}_o, \boldsymbol{n}_o)$; the user principally controls the axis $\boldsymbol{n}_o$ to select a pose constrained to place the facet point $\boldsymbol{p}_o$ at the target point $\boldsymbol{p}_t$ and to align the object normal $\boldsymbol{n}_o$ opposite the target normal $\boldsymbol{n}_t$. Similar to the grasp mapping, the undetermined rotation of the object about the target normal is specified by finding the geodesic from a reference vector on the end effector (like $\boldsymbol{r}_x$ or $\boldsymbol{r}_y$) to the same vector projected onto the target plane.

The resulting placement anchor pose $\boldsymbol{e}_p$ is a direct, cursor-like projection of the grasped object into a placement, and is used in a similar manner as the grasp anchor pose. The anchor itself may not be a feasible placement pose if it puts the object or the gripper into contact with the scene. Candidates $\mathcal{A}_{(\boldsymbol{e}_p)}$ can be generated in the local region around the anchor using any generative object placement method, with the candidate nearest the placement anchor pose serving as the suggestion $\boldsymbol{e}_p^*$ to the user.

D. Snapping

As a consequence of prioritizing responsiveness, the range of inputs which our methods map to any particular assistance

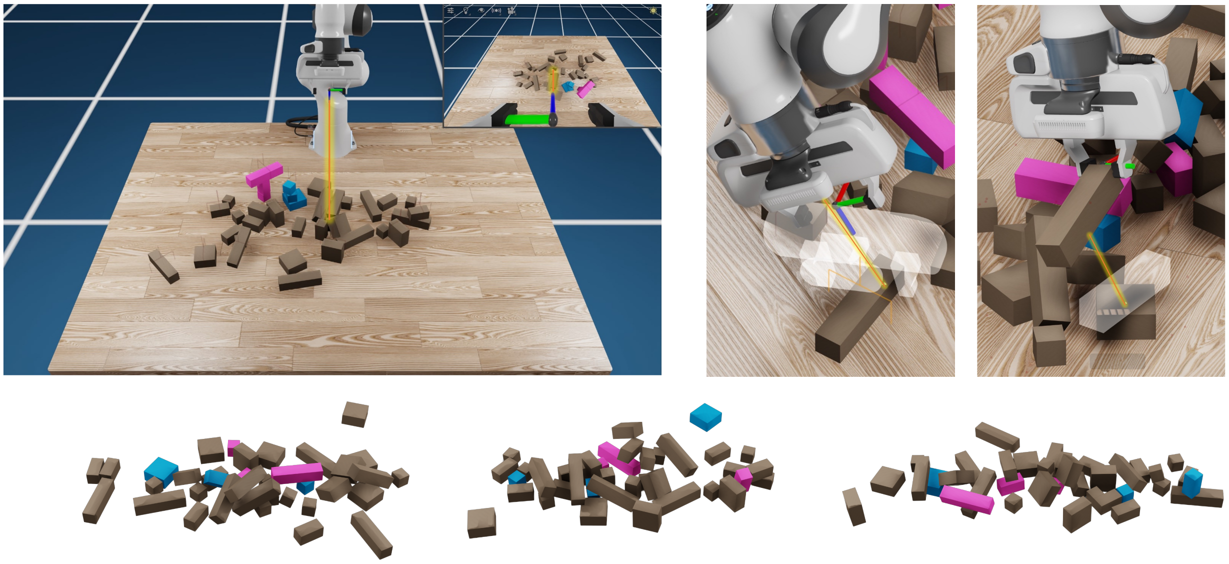

Fig. 3: The operator controls the robot while looking at two camera views displayed picture-in-picture (left). Assistance suggestions are shown as a "ghost gripper" for grasping and a "ghost shape" for placing actions (right). Ray visualizations are exaggerated for legibility in print. The experimental task involved participants extracting and stacking blue and pink blocks that were initially scattered in one of three clutter configurations (bottom).

anchor pose $\boldsymbol{e}_g$ or $\boldsymbol{e}_p$ is small. Certain "easy" poses like a perfectly aligned side-grasp might be frustratingly difficult to specify. We use *snapping* to nudge the generated assistance toward these preferred poses, providing the flexibility to control the grasp suggestion (as is typically needed in cluttered scenes) or to easily snap into commonly used grasps when feasible. The behavior of snapping is demonstrated in the accompanying video.

Snaps are encoded by one or more potential fields $\phi(\cdot)$ over poses. After anchor poses $\boldsymbol{e}_g$ or $\boldsymbol{e}_p$ are calculated, a local optimization over ϕ occurs, checking to see if there is a lower potential pose within an ϵ distance threshold that would breach potential threshold γ. If so, candidates from $\mathcal{A}_{(\boldsymbol{e}_g)}$ or $\mathcal{A}_{(\boldsymbol{e}_p)}$ are ignored and the snap pose is provided as the suggestion.

In practice, we find that specifying a set of poses that align with object centroids coupled with proximity potential $\phi(e^*) = \min_{G_i \in G} d(\boldsymbol{e}^*, G_i)$ is useful for picking and placing and requires no additional task context.

Following [29], we define the distance between the poses $\boldsymbol{x}, \boldsymbol{y} \in \mathrm{SE}(3)$ with position components $\boldsymbol{p}_x, \boldsymbol{p}_y \in R^3$ and rotational components $\boldsymbol{R}_x, \boldsymbol{R}_y \in R^{3\times 3}$, as

$$d(\boldsymbol{x}, \boldsymbol{y})^2 = ||\boldsymbol{p}_x - \boldsymbol{p}_y||_2 + 2\beta^2(1 - \frac{tr(\boldsymbol{R}_y^{-1}\boldsymbol{R}_x)}{3})), \quad (2)$$

where β weights the translation and rotation contributions to the distance.

IV. Experiment

We conducted a within-subjects user study where participants completed stacking tasks without assistance (**CON**), with implicit inference-based assistance suggestions (**IMP**), and with explicit-input assistance suggestions (**EXP**).

Participants completed a multi-step singulation and stacking task where they created multiple stacks of particularly-colored blocks from a cluttered pile. The task was designed to have few prescribed steps and many possible intermediate goals.

We expected that participants would:

- **H1** : be most effective at completing the task using EXP.
- **H2** : make most use of suggestions provided by EXP
- **H3** : report the lowest workload when using EXP.
- **H4** : feel that the suggestions from EXP better match their preferences.
- **H5** : feel that they understand the behavior EXP better than that of IMP.

A. System

Participants interact with a Franka Panda robot simulated in NVIDIA Omniverse Isaac Sim. Grasp sampling and collision checking operations are GPU accelerated using NVIDIA Warp [30].

a) Input: Users provide input using a 6DOF mouse, a spring-suspended puck that they can displace in three spatial dimensions while simultaneously panning, tilting, or twisting to provide 3D rotation [31].

b) Robot Control: User input is interpreted as a twist goal for the robot's end-effector. We integrate the twist over a fixed timestep, apply the resulting transformation to the current end-effector pose, and provide the result as a pose goal to the robot controller, a Riemannian Motion Policy implemented in RMPFlow [32]. To avoid large accelerations, the pose goal is passed through a low pass filter.

c) Camera Control: Users operate the robot while monitoring a fixed view, showing most of the robot and

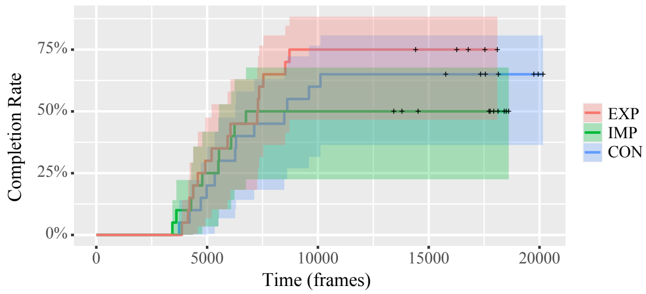

Fig. 4: Survival analysis (↑) of participant's completion of the task over time. Lines plot percentage of participants that completed the task at the time and Xs mark termination without completion. Differences lie within the 95% confidence interval, with a trend that the probability of having completed the task grows most quickly for the explicit input interface and reaches a higher peak.

the workspace, and a dynamic view affixed to the gripper pointing towards the fingers. As shown in 3, one view is foregrounded at a time, and user input is interpreted in the frame of the foregrounded view.

d) Assistance: Offers of assistance are visualized as "ghosts," as shown in Fig. 3. Holding a button on the 3D mouse engages the assistance, forwarding the suggested pose as a goal for the controller with an additional preprocessing step to ensure poses are approached from the front.

e) Explicit Assistance Condition (EXP): We implement our grasp pointing assistance approach using a simple approach-vector parameterized sampling scheme, looking for the nearest non-colliding pose amongst 7125 translated and rotated candidates around the grasp anchor pose. The samples are distributed in a fixed 2cm diameter, 1cm thick disc pattern. We did not use a placement sampler as we assessed that direct control over the placement anchor pose was sufficient for the experimental task. Raycasting is performed against a mesh representation of the scene. We generate axis aligned grasp and placement poses and use them to define a snapping potential as described in Sec. III-D.

f) Implicit Inference-Based Assistance Condition (IMP): Following [15], we attempt to infer the user's goal by selecting the most-probable goal G^* from a predefined set of candidates $\mathcal{G}$ based on a recent window of the robot's trajectory $\xi_{S\to U}$ from start pose S to current pose U:

$$G^* = \arg\max_{G\in\mathcal{G}} \left(\frac{e^{-C_G(\xi_{S\to U})-C_G(\xi^*_{U\to G})}}{e^{-C_G(\xi^*_{S\to G})}} \cdot e^{-d(U,G)} \right) \quad (3)$$

The first term assigns greater likelihood to goals for which the user's trajectory, completed optimally by $\xi^*_{U\to G}$, has cost similar to the cost of the optimal trajectory $\xi^*_{S\to G}$. The second term serves as a prior, assigning more mass to goals that are closer to current pose. We use $C_G(\xi_{X\to Y}) = d(X,Y)^2$ and reset S if 2 seconds pass with no control input. The same set of axis-aligned grasp and placement poses used for snaps are used as $\mathcal{G}$, and collision checking is performed across this set to ensure no in-collision poses are offered.

TABLE I: Comparison of Condition Preference Counts ↑

	A	B	C_A	C_B	$\frac{C_A}{C_A+C_B}$% (CI)	p
EQ1	EXP	CON	11	1	92 (62, 100)	**.010**
	"	IMP	"	8	60 (34, 80)	.648
	CON	"	1	"	11 (0, 48)	.078
EQ2	EXP	CON	10	3	79 (49, 95)	.277
	"	IMP	"	7	61 (33, 82)	.688
	CON	"	3	"	30 (7, 65)	.688
EQ3	EXP	CON	14	1	94 (68, 100)	**.003**
	"	IMP	"	5	75 (49, 91)	.127
	CON	"	1	"	17 (0, 64)	.219
EQ4	EXP	CON	14	2	88 (62, 98)	**.013**
	"	IMP	"	4	79 (52, 94)	.061
	CON	"	1	"	33 (4, 78)	.687

B. Procedure

Participants were told they would use a 3D mouse to control a robot with three different systems, some of which would provide suggestions they could use to help them complete tasks. Each session began with an interactive 3D mouse tutorial, followed by a robot control tutorial where they had to grasp and lift a block, and finally an assistance tutorial which demonstrated what suggestions of assistance would look like and how to use them.

For each condition, participants were given a brief verbal introduction to how the system would behave and asked to "warm up" by stacking a block. Once satisfied that they understood the system, participants completed a single stack task for 3 minutes, then had a maximum of 7 minutes to complete the multi-step stacking task. A post-interaction survey included the NASA-TLX questionnaire [33], three agreement questions regarding their sense of control over the suggestions (reported as assistance composite) and one regarding their sense of understanding. Rating questions were represented using 7 point scales.

A final set of forced-choice questions probed which system "felt easiest to use" (EQ1), and which system had the suggestions that "made it easiest to do the task the way [they] wanted to" (EQ2) which they best understood "why [the suggestions] behaved the way they did" (EQ3), and "felt most in control of" (EQ4). Finally, participants completed demographic questions and rated their familiarity with robots, operating robot arms, 3D mice, and playing video games. Sessions lasted between 45-60 minutes total.

a) Participants: We recruited 20 participants (18 male, 2 female, aged 19-39 *M*=25.1, *SD*=5.45) from the University of Washington under an IRB approved study plan. Many participants were roboticists, rating their familiarity with robots highly (*M*=4.80, *SD*=2.08, 7 point scale). Only two participants reported being familiar with 3D-mice (rating >4 on 7 point scale). All participants were right handed.

C. Methods

We analyze logged events, survey data and supplemental annotations using generalized linear mixed models to account for inter- and intra-participant variance. The effect

TABLE II: NASA-TLX scores ↓

A	B	$M_A(SE_A)$	$M_B(SE_B)$	$M_A - M_B(CI)$	p
EXP	IMP	3.42 (.25)	3.77 (.25)	-.36 (-.97, .25)	.335
"	CON	"	4.33 (.25)	-.92 (-1.53, -.31)	**.002**
IMP	"	3.77 (.25)	"	-.56 (-1.17, .05)	.079

TABLE III: Assistance Subjective Scores ↑

Question	EXP M_A	EXP SE_A	IMP M_B	IMP SE_B	$M_A - M_B(CI)$	p
Composite	4.70	.253	3.44	.253	1.25 (.52, 1.99)	**.002**
Understanding	4.62	.274	4.29	.274	.33 (-.47, 1.14)	.398

TABLE IV: Failure Counts ↓

	A	B	$M_A(SE_A)$	$M_B(SE_B)$	$M_A/M_B(CI)$	p
Pick	EXP	IMP	1.13 (.32)	2.48 (.59)	.46 (.23, .91)	**.028**
	"	CON	"	4.22 (1.15)	.27 (.10, .75)	**.008**
	IMP	"	2.48 (.59)	"	.59 (.27, 1.27)	.242
Place	EXP	IMP	.55 (.18)	.59 (.18)	.93 (.38, 2.29)	.980
	"	CON	"	1.22 (.30)	.45 (.20, .98)	**.043**
	IMP	"	.59 (.18)	"	.48 (.23, 1.04)	.065

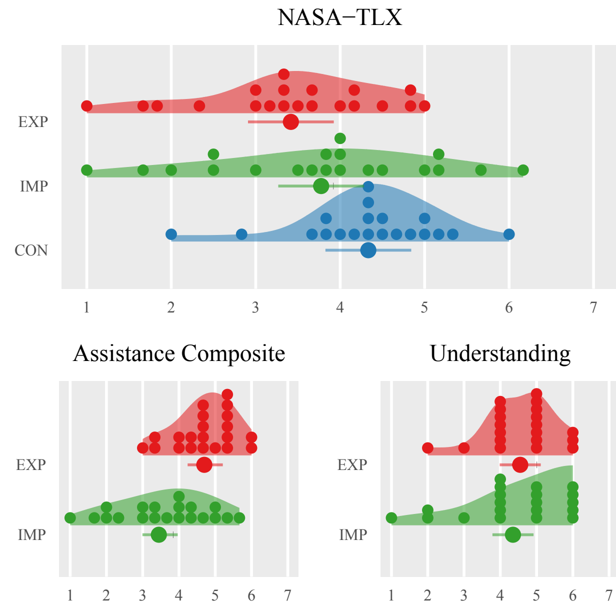

Fig. 5: Raw data for subjective scores collected on 7 point scale with density estimates overlaid. Point and bar show estimated marginal mean with 95% confidence interval.

of an experimental condition is given as either a ratio or difference of the estimated marginal mean against another contrasting condition, and significance is determined using 95% confidence intervals. We conducted survival analysis to characterize task completion rates over time. Statistical details are reported in the supplementary materials.

D. Results

H1: Participants experienced significantly fewer failed picks in EXP when compared to IMP or CON, and there was a trend indicating that they experienced fewer place failures as well, as shown in Tab. IV. There were trends indicating that users of the explicit interface complete the task with higher frequency and stack objects more quickly, as shown in Fig. 4, however the differences are not statistically significant.

H2: There was no measurable difference in the duration or number of engagements of the assistance between the implicit and explicit interfaces. Qualitatively, we observed that some participants made use of the explicit assistance system without engaging it.

H3: Mean workload was lowest for the explicit condition, however the difference was only significant when compared to the control condition. The implicit input condition was rated as higher workload than the explicit condition and lower than no assistance at all, however these differences were not statistically significant, as shown in Tab. II.

H4: participants indicated that the explicit assistance interface was more controllable, rating it 1.25 points (CI .52, 1.99) more highly on average on our assistance composite scale (reported in Tab. III and Fig. 5).

H5 participants rated their understanding higher on average, but the difference was not statistically significant as shown in Tab. III and Fig. 5.

V. Discussion and Limitations

We designed an explicit-input teleoperation interface that is interpretable, responsive, unobstrusive and capable. These design principles have inherent tradeoffs. For example, making assistance more capable may result in a less responsive and less usable system. Reducing latency is only desirable if interpretability can be maintained, a trade-off that often appears when considering how to configure anytime sampling-based planners.

Our implementation is deployed in simulation, making it applicable to simulated data collection or robot teaching interactions. Porting our system to teleoperation of a real robot would require the integration of appropriate generative grasp- and placement-pose models, as well as object state estimation or point cloud-based occupancy checking methods. Our experimental assessment of the interface informs and motivates the future development of physical implementations. Future work should also explore placement assistance with objects and support surfaces that are not well-approximated as planes.

VI. Conclusion

We contribute a new framing for assistance interactions based on explicit input and two new algorithms and interfaces for on-line teleoperation, designed to leverage GPU-based parallel computation to calculate grasping and placing feasible options on-line—even in clutter. Our work goes beyond individual picks by also considering assistance during placement, thus offering a complete workflow for multi-step pick and place tasks. The results of our study highlight the promise of this new kind of assistance interaction, and motivate us to further explore how accelerated computation can augment teleoperation.

References

[1] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei, "Roboturk: A crowdsourcing platform for robotic skill learning through imitation," in *Proceedings of The 2nd Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Billard, A. Dragan, J. Peters, and J. Morimoto, Eds., vol. 87. PMLR, Oct. 2018, pp. 879–893.

[2] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard, "Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7327–7334, 2022.

[3] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[4] C. Li, R. Zhang, J. Wong, C. Gokmen, S. Srivastava, R. Martín-Martín, C. Wang, G. Levine, M. Lingelbach, J. Sun, *et al.*, "Behavior-1k: A benchmark for embodied ai with 1,000 everyday activities and realistic simulation," in *Conference on Robot Learning*. PMLR, 2023, pp. 80–93.

[5] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. S. Ryoo, G. Salazar, P. R. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. H. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "RT-1: Robotics Transformer for Real-World Control at Scale," in *Robotics: Science and Systems (RSS)*, 2023.

[6] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, X. Chen, K. Choromanski, T. Ding, D. Driess, A. Dubey, C. Finn, P. Florence, C. Fu, M. G. Arenas, K. Gopalakrishnan, K. Han, K. Hausman, A. Herzog, J. Hsu, B. Ichter, A. Irpan, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, L. Lee, T.-W. E. Lee, S. Levine, Y. Lu, H. Michalewski, I. Mordatch, K. Pertsch, K. Rao, K. Reymann, M. Ryoo, G. Salazar, P. Sanketi, P. Sermanet, J. Singh, A. Singh, R. Soricut, H. Tran, V. Vanhoucke, Q. Vuong, A. Wahid, S. Welker, P. Wohlhart, J. Wu, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich, "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in *Conference on Robot Learning*. PMLR, 2023.

[7] T. Zhang, Z. McCarthy, O. Jow, D. Lee, X. Chen, K. Goldberg, and P. Abbeel, "Deep imitation learning for complex manipulation tasks from virtual reality teleoperation," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5628–5635.

[8] C. Wang, L. Fan, J. Sun, R. Zhang, L. Fei-Fei, D. Xu, Y. Zhu, and A. Anandkumar, "Mimicplay: Long-horizon imitation learning by watching human play," in *Conference on Robot Learning*, 2023.

[9] E. Rosete-Beas, O. Mees, G. Kalweit, J. Boedecker, and W. Burgard, "Latent plans for task-agnostic offline reinforcement learning," in *Conference on Robot Learning*. PMLR, 2023, pp. 1838–1849.

[10] G. Zhou, L. Ke, S. Srinivasa, A. Gupta, A. Rajeswaran, and V. Kumar, "Real world offline reinforcement learning with realistic data source," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 7176–7183.

[11] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín, "What matters in learning from offline human demonstrations for robot manipulation," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, A. Faust, D. Hsu, and G. Neumann, Eds., vol. 164. PMLR, 08–11 Nov 2022, pp. 1678–1690.

[12] S. Belkhale, Y. Cui, and D. Sadigh, "Data quality in imitation learning," in *Proceedings of the 37th International Conference on Neural Information Processing Systems*, ser. NIPS '23. Red Hook, NY, USA: Curran Associates Inc., 2024.

[13] Y. Zhu, B. Jiang, Q. Chen, T. Aoyama, and Y. Hasegawa, "A shared control framework for enhanced grasping performance in teleoperation," *IEEE Access*, vol. 11, pp. 69 204–69 215, 2023.

[14] Y. Qin, W. Yang, B. Huang, K. Van Wyk, H. Su, X. Wang, Y.-W. Chao, and D. Fox, "Anyteleop: A general vision-based dexterous robot arm-hand teleoperation system," *Proceedings of Robotics: Science and Systems*, 2023.

[15] A. D. Dragan and S. S. Srinivasa, "Formalizing teleoperation assistance," in *Robotics: Science and Systems*, 2008.

[16] A. B. Shervin Javdani, Siddhartha Srinivasa, "Shared autonomy via hindsight optimization," *Proceedings of Robotics: Science and Systems*, 2015.

[17] S. Nikolaidis, Y. X. Zhu, D. Hsu, and S. Srinivasa, "Human-robot mutual adaptation in shared autonomy," in *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 2017, pp. 294–302.

[18] P. A. Lasota and J. A. Shah, "A multiple-predictor approach to human motion prediction," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 2300–2307.

[19] C. Pérez-D'Arpino and J. A. Shah, "Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification," in *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2015, pp. 6175–6182.

[20] H. M. Clever, A. Handa, H. Mazhar, K. Parker, O. Shapira, Q. Wan, Y. Narang, I. Akinola, M. Cakmak, and D. Fox, "Assistive tele-op: Leveraging transformers to collect robotic task demonstrations," in *4th NeurIPS Robot Learning Workshop on Self-Supervised and Lifelong Learning*, 2021, workshop.

[21] T. Yoneda, L. Sun, G. Yang, B. C. Stadie, and M. R. Walter, "To the noise and back: Diffusion for shared autonomy," in *Robotics: Science and Systems XIX, Daegu, Republic of Korea*, 2023.

[22] S. Reddy, A. D. Dragan, and S. Levine, "Shared autonomy via deep reinforcement learning," *Proceedings of Robotics: Science and Systems*, 2018.

[23] A. E. Leeper, K. Hsiao, M. Ciocarlie, L. Takayama, and D. Gossow, "Strategies for human-in-the-loop robotic grasping," in *Proceedings of the Seventh Annual ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 1–8.

[24] C. Pérez-D'Arpino, R. P. Khurshid, and J. A. Shah, "Experimental assessment of human–robot teaming for multi-step remote manipulation with expert operators," *J. Hum.-Robot Interact.*, vol. 13, no. 3, Aug. 2024.

[25] D. P. Losey, H. J. Jeon, M. Li, K. Srinivasan, A. Mandlekar, A. Garg, J. Bohg, and D. Sadigh, "Learning latent actions to control assistive robots," *Auton. Robots*, vol. 46, no. 1, p. 115–147, Jan. 2022.

[26] Y. Cui, S. Karamcheti, R. Palleti, N. Shivakumar, P. Liang, and D. Sadigh, "No, to the right: Online language corrections for robotic manipulation via shared autonomy," in *Proceedings of the 2023 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 93–101.

[27] M. Rubagotti, T. Taunyazov, B. Omarali, and A. Shintemirov, "Semi-autonomous robot teleoperation with obstacle avoidance via model predictive control," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2746–2753, 2019.

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

[29] C. Mazzotti, N. Sancisi, and V. Parenti-Castelli, "A measure of the distance between two rigid-body poses based on the use of platonic solids," in *ROMANSY 21 - Robot Design, Dynamics and Control*, V. Parenti-Castelli and W. Schiehlen, Eds. Cham: Springer International Publishing, 2016, pp. 81–89.

[30] M. Macklin, "Warp: A high-performance python framework for gpu simulation and graphics," https://github.com/nvidia/warp, Mar. 2022, nVIDIA GPU Technology Conference (GTC).

[31] V. Dhat, N. Walker, and M. Cakmak, "Using 3d mice to control robot manipulators," in *Proceedings of the 2024 ACM/IEEE International Conference on Human-Robot Interaction*, ser. HRI '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 896–900.

[32] C.-A. Cheng, M. Mukadam, J. Issac, S. Birchfield, D. Fox, B. Boots, and N. Ratliff, "Rmpflow: A geometric framework for generation of multitask motion policies," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 968–987, 2021.

[33] S. G. Hart and L. E. Staveland, "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in *Human Mental Workload*, P. A. Hancock and N. Meshkati, Eds. North-Holland, 1988, pp. 139–183.