Learning Autonomous Viewpoint Adjustment from Human Demonstrations for Telemanipulation

RUIXING JIA, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
LEI YANG, TransGP and The University of Hong Kong, Pokfulam, Hong Kong SAR, China
YING CAO, ShanghaiTech University, Shanghai, China
CALVIN KALUN OR, The University of Hong Kong, Pokfulam, Hong Kong SAR, China
WENPING WANG, Texas A&M University, College Station, TX, USA
JIA PAN, The University of Hong Kong, Pokfulam, Hong Kong SAR, China

Teleoperation systems find many applications from earlier search-and-rescue to more recent daily tasks. It is widely acknowledged that using external sensors can decouple the view of the remote scene from the motion of the robot arm during manipulation, facilitating the control task. However, this design requires the coordination of multiple operators or may exhaust a single operator as s/he needs to control both the manipulator arm and the external sensors. To address this challenge, our work introduces a viewpoint prediction model, the first data-driven approach that autonomously adjusts the viewpoint of a dynamic camera to assist in telemanipulation tasks. This model is parameterized by a deep neural network and trained on a set of human demonstrations. We propose a contrastive learning scheme that leverages viewpoints in a camera trajectory as contrastive data for network training. We demonstrated the effectiveness of the proposed viewpoint prediction model by integrating it into a real-world robotic system for telemanipulation. User studies reveal that our model outperforms several camera control methods in terms of control experience and reduces the perceived task load compared to manual camera control. As an assistive module of a telemanipulation system, our method significantly reduces task completion time for users who choose to adopt its recommendation.

CCS Concepts: • **Computer systems organization → Robotics**;

Additional Key Words and Phrases: Human robot interaction, automatic camera placement, learning from demonstrations, teleoperation

ACM Reference format:
Ruixing Jia, Lei Yang, Ying Cao, Calvin Kalun Or, Wenping Wang, and Jia Pan. 2024. Learning Autonomous Viewpoint Adjustment from Human Demonstrations for Telemanipulation. *ACM Trans. Hum.-Robot Interact.* 13, 3, Article 32 (September 2024), 23 pages.
https://doi.org/10.1145/3660348

Authors' Contact Information: Ruixing Jia, The University of Hong Kong, Pokfulam, Hong Kong SAR, China; e-mail: ruixing@connect.hku.hk; Lei Yang (Corresponding author), TransGP and The University of Hong Kong, Pokfulam, Hong Kong SAR, China; e-mail: lyang@cs.hku.hk; Ying Cao, ShanghaiTech University, Shanghai, China; e-mail: caoying59@gmail.com; Calvin Kalun Or, The University of Hong Kong, Pokfulam, Hong Kong SAR, China; e-mail: klor@hku.hk; Wenping Wang, Texas A&M University, College Station, TX, USA; e-mail: wenping@cs.hku.hk; Jia Pan (Corresponding author), The University of Hong Kong, Pokfulam, Hong Kong SAR, China; e-mail: jpan@cs.hku.hk.

ACM 2573-9522/2024/9-ART32
https://doi.org/10.1145/3660348

1 Introduction

Teleoperation systems find applications in many real-world scenarios, from earlier search-and-rescue [19, 20] to more recent daily tasks such as object manipulation [12, 26], food feeding [21], or tele-nursing [38]. Usually, operators of these systems are presented with one or multiple feeds from the remote cameras/sensors, and they need to synthesize information about the scene based on these camera feeds to achieve certain manipulation tasks. However, kinematic differences between the robot and operator, limited viewing options, and depth ambiguities present significant challenges, necessitating complex training for users to become proficient. Recent research has introduced various means to reduce the cognitive burden of the operators either by providing informative visual cues via augmented reality [9, 33, 40] or more viewing options, such as multiple view selection [5, 6, 25, 38, 43] or continuous viewpoint adjustment [2, 17, 23, 27, 28, 32].

In this article, we focus on the latter means of continuous viewpoint adjustment. This line of work has been proven to be effective in reducing users' cognitive load and can facilitate human–robot interaction [45]. For example, a set of geometry-based heuristics was proposed to define the focus of the camera when the gripper approaches a target object [27], and manual adjustment for adapting viewpoints based on the task semantics or exploration was enabled [28]. These design choices are generally aligned with how humans adjust their views when performing a pick-and-place task [15]. However, it is not always desirable for humans to have a viewpoint fixating on the gripper across the entire pick-and-place task, which is adopted by Rakita et al. [27]. Instead, human eyes fixate on the objects to be grasped or in grip when it is to be released (e.g., placed onto a target location) as suggested by Lavoie et al. [15].

Through our pre-study, we observed that during the picking or the placing tasks, alignment between the gripper and the target object or between different targets is, on the one hand, crucial for the success of telemanipulation tasks and, on the other hand, demanding for the users. In addition to the target visibility considered in [27], it also requires specific viewing angles to reflect the relative relationship between the gripper and the target (for grasping) or between different target objects (for placing). These desirable viewing angles can mitigate the depth ambiguity of the scene when viewed from a two-dimensional screen and facilitate users to achieve the desired level of alignment. Providing a desired viewpoint to users is, therefore, crucial to the performance of telemanipulation tasks where a high level of alignment is required. To address this problem, we developed an algorithm for autonomously placing the camera viewport during a pick-and-place task. Our effort is complementary to previous works on autonomous camera control, extending the current state-of-the-art works to cover the alignment aspect for fine manipulation.

To automate the camera placement, we built *a viewpoint prediction model* using **deep neural networks (DNNs)**. Yet, training DNNs requires plenty of labeled data, while collecting the data from human operators is prohibitive. To reduce the human effort in data collection, we proposed two strategies. First, we make use of all viewpoints visited in a single trajectory to provide contrastive training examples. Second, observing that manually adjusting the camera viewpoint may exhaust a non-expert user, we employed an interactive system and allowed users to correct the automatically generated views for data collection.

To demonstrate the effectiveness of the proposed learning-based model, we built a physical setup of a bimanual telerobotics system similar to [27, 28]. A dynamic camera is held by a robotic arm to provide a real-time video stream of the scene viewed by the camera. The other robotic arm is used to perform the manipulation tasks. The user can see the remote workplace from the camera feed and interact with the telerobotics system by controlling both the camera and the gripper to perform the manipulation tasks. We conducted two user studies to validate the proposed viewpoint prediction model as *a standalone module* for camera adjustment and as *a part of an assistive telemanipulation*

system, respectively. In the first user study, we compared the viewpoint prediction model to three baseline methods for adjusting the camera viewpoint. From the experimental results, we found the proposed model could reduce the task load and lead to a better control experience. In the second user study, the proposed model serves as an assistive module for the presented telemanipulation system, where users can request assistance from the proposed model. We compared this assistive mode to the manual camera control baseline. We observed a strong correlation between user acceptance of the proposed system and its benefits in improving control experience and enhancing task efficiency. We release our implementation of the proposed method as well as the baseline methods in this link.[1]

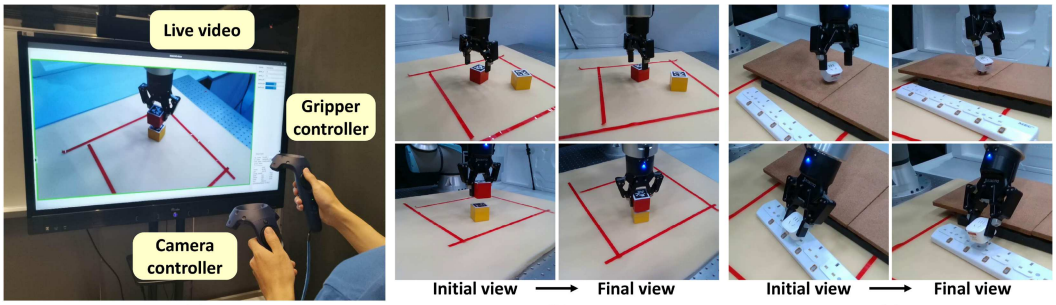
The contribution of this article is three-fold: (1) a viewpoint prediction model to learn from human operator's demonstrations to predict potential desirable viewpoints to assist users for telemanipulation tasks; (2) a contrastive learning strategy to fully utilize the camera motion trajectory for data labeling; and (3) a user study on our physical prototype that validated the efficacy of the proposed method.

2 Related Works

Teleoperation tasks are usually performed by one or more human operators, who synthesize information based on the streaming images on a screen display returned by the remote cameras/sensors. Due to the lack of stereo perception to tell the depth, ambiguity about the scene can be severe. Therefore, it is straightforward to use multiple cameras to provide alternative views [4, 6, 7, 14, 25, 31, 36, 37] for enhancing operators' performance. However, placing the camera in a suitable location or selecting a view from a candidate set to facilitate teleoperation tasks will increase the cognitive burden of the single operator or require coordination between multiple operators. Hence, natural means to interact with the telerobotics system, such as by eye gaze [21, 25], have been explored. Leveraging the technological advances in virtual reality [3, 23, 30, 41] or augmented reality [9, 33, 40] has also been widely discussed. In this study, we consider automating an external moving camera that has long been pursued to reduce the operator's cognitive burden in the teleoperation tasks [2, 17].

Recent works [5, 22, 27, 28, 43] extended the previous ones from several aspects. Nicolis et al. [22] proposed a dynamic camera control method to avoid nearby object occlusions. Rakita et al. [27] developed an autonomous camera system that can follow the gripper during manipulation. Their follow-up work [28] extended the previous one by adjusting the viewpoint to avoid occlusions due to the cluttered remote workplace. In both works, a set of heuristic rules was used to optimize the dynamic camera's pose while tracking the gripper. This design rationale for what to look at during the coarse movement of the gripper (i.e., transport the gripper from one place to another) aligns well with the observation from [15]. Similarly, Senft et al. [32] adopt a set of view heuristics to optimize and generate candidate viewpoints for a flying drone with a dynamic camera. Since the heuristics used in these studies are rules for optimizing the viewport to avoid *environmental occlusions*, we follow the empirical observation made in [15] for fine manipulation and extend this line of research through the discussion of how to select a viewpoint that potentially reduces the depth ambiguity and facilitates fine manipulation tasks, such as grasping or stacking. Our second difference from the previous works is the use of a learning-based model to predict the desirable viewpoints.

[1] https://github.com/rxjia/view_adjust



(a) (b) (c)

Fig. 1. A viewpoint prediction model was learned from human demonstrations to predict potentially desirable viewpoints for remote telemanipulation given the current manipulation state, automating the viewpoint adjustment which is demanding for users. (a) A user was remotely controlling the system for a telemanipulation task by watching a live video. (b) A task in the user study. (c) The ability of our method in a more realistic scenario.

Another line of recent research [5, 34, 43] has proposed leveraging the concept of environmental affordance and psychomotor aspects to evaluate the quality of viewpoints. This research encompasses a broader range of teleoperation tasks, including traversing and passing in addition to manipulation. Expert users were invited to evaluate a limited set of 30 views for each affordance concept corresponding to a teleoperation task.

Our work shares similarities with this line of research in that both aim to assess the *viewpoint utility* based on human evaluation. However, our work takes a step further, developing a data-driven approach that can evaluate the viewpoints through learning from human demonstrations. To achieve this, we proposed a novel contrastive learning strategy to operate in a low-data realm. This learning-based method also enables us to leverage additional information, such as how the gripper approaches the target object.

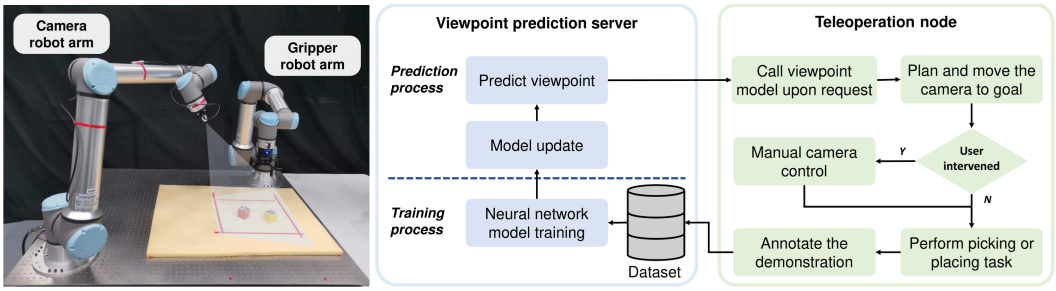
3 Methodology

In this section, we first give an overview of our telemanipulation system (Section 3.1). Then, we present a viewpoint prediction model that predicts desirable viewpoints for users (Section 3.2) trained with a contrastive learning scheme. Finally, we introduce the user interface and data collection in Section 3.3.

3.1 System Overview

Using one or more external cameras has become the usual practice for teleoperation. Our system follows this design rationale and adopts a similar setting presented in [27, 28] to use two robotic arms and only one dynamic camera for the telemanipulation task. In our physical prototype, one of the robotic arms is equipped with a parallel gripper while the other moves the camera to change the viewpoint continuously. Each of the robot arms can be manually controlled via an HTC Vive^{TM} controller (see Figure 1). Figure 2(a) shows the physical setup of the physical prototype. Our system also features a *viewpoint prediction model* that predicts potentially desirable viewpoints based on the manipulation states for users. The training process and how it interacts with the users during data collection are shown in Figure 2(b).

Demonstrated in previous works (e.g., [25]) and also observed in our pre-study, users prefer to control the gripper movement in the reference frame aligning with the viewing frame. Hence, we used the current viewing frame as the reference frame for gripper motion control. This leads to the design of an alternating control pattern where the users controlled either the gripper or the camera



(a) Physical setup (b) System architecture

Fig. 2. System overview. (a) Physical setup of the remote workplace for the telemanipulation tasks. (b) System architecture shows how the viewpoint prediction model is trained in an interactive data collection process.

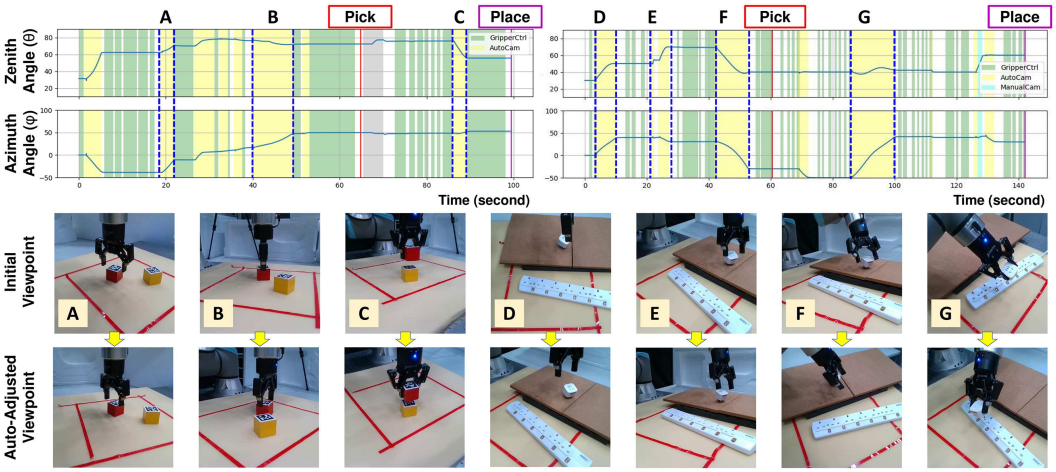


Fig. 3. Two samples of the control signals regarding the zenith (θ) and azimuth (ϕ) angles are shown. The yellowish and greenish blocks represent the control of the camera and of the gripper. Three and four pairs of live video snapshots (cropped and zoomed-in for readability) before and after the autonomous viewpoint adjustment are shown for the two samples, respectively. Pairs A, B, D, E, and F were captured before the pick event (the red line); pairs C and G were captured after that. Views obtained by the automatic adjustment can better reveal relative positions between the gripper and the target, reducing the depth ambiguity. For example, pair E reflects the gap between the gripper and the target object along the axis perpendicular to the tabletop; pair G shows the misalignment between the gripper and the target.

at one time, avoiding the issue of confounding viewpoint control mentioned in [18]. Our viewpoint prediction model is, therefore, inactivated when the gripper is moving and predicts desirable view changes upon the request by users when the gripper is stationary.

Several examples of the control signals and their corresponding camera motions are shown in Figure 3. For example, the view suggested by our automatic viewpoint adjustment in snapshot C shows that the two cubes are almost aligned. Hence, the user can move the gripper downwards without any horizontal adjustment. In snapshot F, the suggested view indicates that the user needs to adjust the gripper before picking the plug. The suggested view in snapshot G can better facilitate the plug-in task. See the attached video for more details. Next, we present the implementation details of the viewpoint prediction model.

3.2 Viewpoint Prediction Model

As demonstrated in [15], humans consistently direct their attention toward or near the object of interest during manipulation tasks, such as picking and placing. Hence, the moving camera of our system is set to look at the target object during the pick-and-place tasks. Consequently, the automatic placement of a moving camera reduces to determining a camera pose, or a viewpoint, which can be expressed as the coordinates of a spherical coordinate system with the origin at the target object.

One way to address this problem is to adopt a regression approach to train a model that predicts the coordinate of the viewpoint based on the manipulation state with a large amount of training data. However, the amount of data that can be collected from human demonstrations is limited. Hence, the model should be able to learn the utility of a viewpoint from this limited set of demonstration data. To achieve this, we propose a contrastive learning approach that leverages all viewpoints visited by the trajectories of human demonstrations. Specifically, we treat different viewpoints as contrastive examples and require our model to score all viewpoints in a candidate set. This way, both the target viewpoints collected and those implicitly rejected by human operators are exploited during the training, which leads to the superior performance of our contrastive approach compared to the regression approach as shown later. Next, we will first present the formulation of the training objective and then elaborate on how the contrastive examples of viewpoints are defined from our collected data.

3.2.1 Viewpoint Modeling. We consider the target viewpoint to be a sample from a discrete set of views $\mathcal{V} = \{\mathbf{v}_i\}$ uniformly sampled from a spherical surface centered at the target object.

We define the manipulation state, $\mathbf{s}$, to contain the following information: (1) current viewpoint $\mathbf{v}_{\text{cur}}$, (2) current gripper pose $\mathbf{g}_{\text{cur}}$, and (3) gripper movement $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{T}$ in previous seconds. The proposed viewpoint prediction model F takes as input a manipulation state, $\mathbf{s}$, and predicts a score ($p \in [0, 1]$) for each view $\mathbf{v}_i$ in $\mathcal{V}$. Formally, we have

$$p(\mathbf{v}_i|\mathbf{s}) = F(\mathbf{v}_i, \mathbf{s}), \quad \forall \mathbf{v}_i \in \mathcal{V}, \tag{1}$$

where $\mathbf{s} = [\mathbf{v}_{\text{cur}}, \mathbf{g}_{\text{cur}}, E(\mathcal{G})]$ and $\mathcal{G} = \{\mathbf{g}_i\}_{i=1}^{T}$ represents the gripper movement consisting of poses from previous T frames (spanning 2 seconds). All components of the manipulation state are represented in the local frame of the target object. To simplify the experimental settings, we assume that our system has access to the positions and orientations of the target objects in the remote workplace, but other existing methods [16, 42] can be used to obtain this piece of information in real-world applications.

Viewpoint prediction model F is implemented as a network of **multi-layer perceptrons (MLP)** with 3 hidden layers of 512 dimensions, and motion encoder E is also designed as an MLP with 1 hidden layer of 256 dimensions to aggregate the motion information. We chose this network architecture to predict the score of each candidate viewpoint as MLPs are proven to be effective in predicting a scalar field [35, 44]. We opt for a three-layer small network for its inference efficiency and due to the simplicity of our tasks.

We consider the target viewpoint to be the one with the highest score, or $\mathbf{v}^* = \arg\max_{\mathbf{v}_i} p(\mathbf{v}_i)$. Once the target viewpoint is predicted, a smooth path is planned on the spherical surface (to keep a constant distance $r = 0.6$ m between the object and the camera) to transport the camera viewpoint from the current pose $\mathbf{v}_{\text{cur}}$ to the target one $\mathbf{v}^*$.

3.2.2 Learning from Contrastive Examples. We detail how the collected trajectories from human operators are annotated into *positive* and *negative* examples for contrastive learning of the viewpoint prediction model.

Given a manipulation state $\mathbf{s}$, users will change the viewpoint and continue performing the manipulation task, either moving the gripper again and adjusting the view or performing the *final action* (e.g., trigger the picking or stacking actions) to finish the task. To collect training data on desirable views that may lead to the success of the manipulation task, we only focused on the camera trajectory, along with its previous manipulation state, right before the final action.

As training a neural network usually requires a large amount of data, we made *two core assumptions* about the collected data to get dense annotation for each camera pose in the camera trajectory. We assume that (1) the operators are rational in the sense that they will not choose another viewpoint if a good one is presented to them and (2) if a task fails, it is because there exists no good viewpoint along the camera trajectory for completing the task. We consider a viewpoint $\mathbf{v}$ desirable if it can lead to a successful completion of the task.

We define a *control sequence* as a tuple of $\{\mathbf{s}, \Gamma, \alpha\}$, where $\mathbf{s}$ is the manipulation state as described previously; $\Gamma = \tau_{i=0}^{K}$ denotes the trajectory that adjusts viewpoint τ_0 to τ_K right before the *final action*; α indicates the success ($\alpha = 1$) or failure ($\alpha = 0$) of a task. We refer to the views ($\tau_{i=0}^{K}$) visited by the trajectory Γ as seen views and the rest as unseen. If $\alpha = 1$ (i.e., the task was successfully completed), we labeled the candidate views from $\mathcal{V}$ that are close to the last viewpoint (angular distance D_T less than 10°) in the trajectory as desirable views. The rest of seen views were labeled as undesirable. This practice is based on Assumption 1; if the operator finds an intermediate view to be desirable, s/he should have stopped at this view instead of ending at an inferior view that appeared later. If otherwise the task failed (i.e., $\alpha = 0$), all seen views were considered undesirable, following the rationale that the user would have stopped at a desirable view to finish the task if there is any (based on Assumptions 1 and 2).

On the other hand, we also need to label the unseen views in the entire space of candidate views, $\mathcal{V}$. For these views, we simply labeled them as undesirable, but a different weight was applied to them in the training objective. This difference between the weights on the seen views and the unseen ones can be seen as the difference between annotation confidences on the seen and unseen views. Two example camera trajectories with labels are shown in Figure 4.

Thus, we write the training loss term for a success control sequence (i.e., $b = \{\mathbf{s}, \Gamma, \alpha = 1\}$) as

$$L^{+}(\mathbf{v}_i, b) = \begin{cases} 0.3 \cdot BCE(p(\mathbf{v}_i|\mathbf{s}), l_{-}), & D(\Gamma, \mathbf{v}_i) > D_T \\ 0.7 \cdot BCE(p(\mathbf{v}_i|\mathbf{s}), l_{+}), & D(\tau_K, \mathbf{v}_i) < D_T \\ 0.7 \cdot BCE(p(\mathbf{v}_i|\mathbf{s}), l_{-}), & \text{otherwise,} \end{cases} \tag{2}$$

and the loss term for a failure control sequence ($\alpha = 0$) as

$$L^{-}(\mathbf{v}_i, b) = \begin{cases} 0.3 \cdot BCE(p(\mathbf{v}_i|\mathbf{s}), l_{-}), & D(\Gamma, \mathbf{v}_i) > D_T \\ 0.7 \cdot BCE(p(\mathbf{v}_i|\mathbf{s}), l_{-}), & \text{otherwise.} \end{cases} \tag{3}$$

Here, we set $l_{+} = 1$ to represent the **ground-truth (GT)** label of a viewpoint if it leads to task success; otherwise, this viewpoint has the label $l_{-} = 0$. BCE is the binary cross entropy loss

$$BCE(y, \hat{y}) = -\left(y \log(\hat{y}) + (1 - y) \log(1 - \hat{y})\right), \tag{4}$$

where y and $\hat{y}$ denote the predicted score of a viewpoint and the GT label of this viewpoint, respectively. Finally, $D(\cdot, \mathbf{v}_i)$ is the distance of $\mathbf{v}_i$ to either the camera trajectory Γ or a viewpoint τ_k in the trajectory. The training objective for a batch of control sequences $B = \{b_j\}$ is minimized with respect to the network parameters Φ.

$$\Phi^{*} = \arg\min_{\Phi} \sum_{b_j \in B} \sum_{\mathbf{v}_i \in \mathcal{V}} \alpha L^{+}(\mathbf{v}_i, b_j) + (1 - \alpha) L^{-}(\mathbf{v}_i, b_j). \tag{5}$$

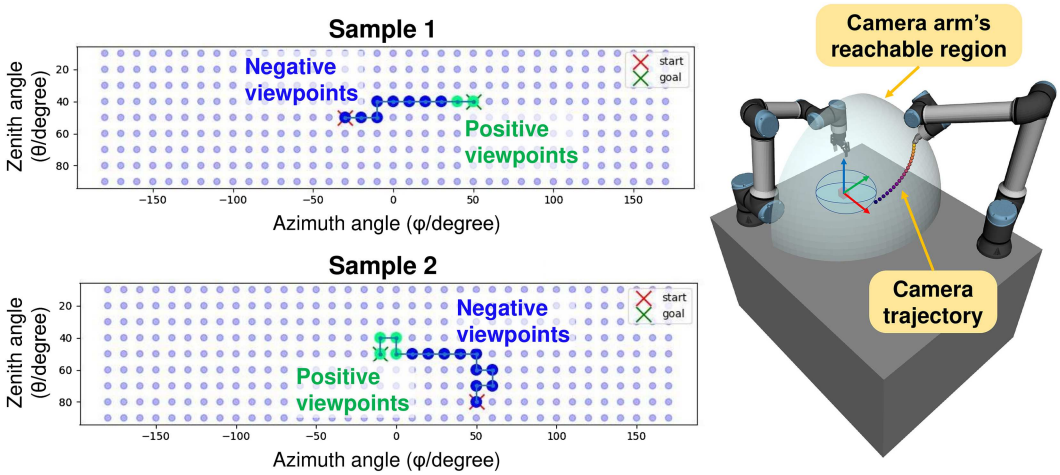


Fig. 4. Annotating each camera trajectory as contrastive examples. Viewpoints that are visited by the trajectory and closer to the destination of the trajectory are labeled as positive viewpoints. The rest of the visited viewpoints are labeled negative. Unseen viewpoints (transparent ones) in the ambient space are labeled negative with a smaller weight.

Table 1. Ablation Study on the Performances of Different Design Choices

Method	Mean (M) prediction error (°)	Error threshold		
		≤ 10°	≤ 20°	≤ 30°
Regression	19.8 (11.3)	22.8%	55.7%	65.4%
Contrastive-0s	12.3 (9.5)	59.4%	83.9%	94.3%
Contrastive-2s	11.9 (8.8)	61.1%	85.2%	96.0%

The average prediction errors with SDs in parentheses are shown. Our approach (Contrastive-2s) is compared to two alternatives (Regression-2s and Contrastive-0s). Our approach outperforms the other two, validating our design choices.

Evaluation on the Contrastive Learning Scheme. We conducted an ablation study to validate the use of the proposed contrastive learning scheme against the regression approach. For our model, the viewpoint with the highest prediction score is considered the predicted target view, while the regression approach (*Regression*) directly predicts the coordinates of target viewpoints given the same input as ours.

The performance of different methods was measured as the angular distance between the predicted target viewpoints and the user-selected viewpoints. We collected 400 control sequences and randomly split them into five folds. We conducted a five-fold cross-validation on these data, that is, we trained the model on four folds and tested it on the remaining one. This was repeated five times, each with a different fold as the test set. The performances of different design choices are reported in Table 1. The average test error of our method (*Contrastive-2s*) is 11.9° against 19.8° for *Regression*, validating our design choice.

3.3 Prototype Implementation and Data Collection

Prototype and User Interface. At the backend, the telemanipulation system uses CollisionIK [29] to generate the joint position commands, which were sent to the joint position controller of **Universal**

Robot (UR) Robot Operating System (ROS) Driver [1] to control two six-df UR robot arms (UR16e for manipulation and UR10e for camera control). The gripper for manipulation is a parallel gripper (ROBOTIQ 2F-85), and an Intel Realsense D435 sensor is used as the remote camera. Our system runs on a Linux desktop with an Intel i7-11700K @ 3.60 GHz CPU and 32 GB RAM. The programs communicate via the ROS. The live video is of size $1,280 \times 720$. The streaming video showing the remote workplace captured by a moving camera was displayed on a 52″ screen monitor. Two HTC Vive controllers were used to control the gripper and the camera motions (see Figure 1(a)).

As for the neural network, we implemented it in PyTorch [24] and adopted the Adam optimizer [13] (with the initial learning rate of 10^{-3}).

We employed an alternating control strategy in which only the camera or the gripper can be controlled and moved by the users at one time. With this alternating control strategy, we ensure that the control frame defining the movement of the gripper in the remote scene aligns with the viewing frame of the camera for visualizing the remote scene [25]. Hence, even drastic changes in the gripper's movement will not introduce inconsistent movement to the users.

Gripper control is only enabled when the trigger button on the joystick controller for the gripper is pressed and held. The movement of the gripper is controlled by moving and rotating the joystick controller in the space. The control frame (coordinate system for controlling the gripper) is defined as the viewing frame when the gripper controller is triggered [25]. Owing to the alternating control strategy, the viewing frame will not change as the camera is stationary when the gripper is moving. To reduce the physical workload of participants in the user studies, we only enable four **Degrees of Freedom (DF)** movements of the gripper, namely, translation along three axes and the rotation DF about the z-axis (the direction of gravity). The gripper's opening/closing is controlled by clicking the menu button of the gripper controller.

Camera control is enabled by holding the trigger of the corresponding controller as well. Touching/pressing, for example, the southwest of the trackpad can generate a corresponding camera motion moving along the same direction in the spherical surface. We relaxed the camera position to be within a distance ± 0.1 m from the spherical surface (with a 0.6 m radius) centered at the target object. We specifically chose to use the trackpad to control the camera viewpoint because our camera movement is parameterized to a spherical coordinate system. When a user presses the camera controller's trigger without pressing the controller's trackpad, the viewpoint prediction model is invoked to predict a viewpoint given the aforementioned manipulation state. An assistive camera mode is provided to allow users to terminate this automatic camera placement process and manually control the camera motion by pressing the trackpad.

Data Collection. Our data collection is conducted in a simulation environment adapted from RLBench [11]. We train the viewpoint prediction model in an online fashion as we collect the data. This is shown in Figure 2(b) where two processes are run during the data collection. In the training process, an instance of the viewpoint prediction model is trained in the background. In the prediction process, another instance of the model is run to predict target viewpoints upon the user's request. For every five interactions, the prediction model instance will be updated by cloning the weights of the model from the background training process.

With this online training scheme, we employ a correction-based interface to facilitate the data collection. Upon receiving a recommended target viewpoint from the model, users can accept this viewpoint or reject it by manually adjusting the camera pose. Such a correction can be made anytime during the camera movement. The entire camera trajectory before the *final action* is then labeled as described in Section 3.2.2.

Since all components of a manipulation state **s** are represented in the local coordinate system of the target object, we made use of the symmetry of the target objects at runtime, if applicable, to augment the collected data. We rotated the local frame of the target object according to its rotational symmetry. After that, we transformed the poses of the gripper and the camera into these rotated frames and input them into the trained model to predict target viewpoints.

4 User Studies and Discussion

The results of two user studies are presented. The goal of User Study 1 is to compare our viewpoint prediction model with the other three methods for viewpoint adjustment to validate our design. In User Study 2, our viewpoint prediction model serves as an assistive module in our telemanipulation system prototype, which is compared to a manual camera control mode. The objective is to investigate how this assistive technique will be used and how it performs as compared to the telemanipulation system with a manually controlled camera. Note that the DNN for our viewpoint prediction model was fixed in both user studies.

4.1 User Study 1: Comparison with Multiple Baselines

Baseline Methods. In this study, three baseline methods (conditions) are compared with our viewpoint prediction model (denoted as Condition *Auto-Cam*) as follows.

Manu-Cam: In this *Manual Camera control* setting, users manually adjust the viewpoint to achieve the task in the feasible viewing region. The manual control camera is adapted from the semantically dictated viewpoint adaptations of [28], with two modifications. First, the camera focuses on the target object based on [15] for the alignment tasks. Second, the camera here is controlled by a trackpad rather than the joystick's motions in space.

Rand-Cam: This *Random Camera control* setting is included as a random control group to demonstrate the effectiveness of the proposed method for predicting desirable viewpoints. The only controlled variable between *Rand-Cam* and *Auto-Cam* is that the viewpoints of *Rand-Cam* are randomly sampled from the uniform distributions ($\theta \sim \mathcal{U}(L_\theta, U_\theta)$, $\phi \sim \mathcal{U}(L_\phi, U_\phi)$).

Dyna-Cam: This *Dynamic Camera control* setting is a rule-based camera control method presented in [27]. While the other methods have the camera focused on target objects, the camera of *Dyan-Cam* always looks at the gripper from a randomly initialized angle. Previous methods [27, 28] adjust the camera viewpoint only when the gripper is occluded. Since the gripper will not be occluded in our tasks, the viewing angle of this condition is fixed. That said, the gripper's motion allows the camera to move and hence provide different views of the remote scene.

The same feasible viewing region ($L_\theta = 30°$, $U_\theta = 80°$, $L_\phi = -50°$, $U_\phi = 90°$) is applied to all four conditions due to the mechanical limit of the camera robot arm. To evaluate the quality of the predicted target viewpoints, a black screen was displayed to mask out the intermediate camera feeds during the transition of the camera under conditions *Auto-Cam* and *Rand-Cam*. This way, the spatial perception is solely based on the stationary images captured from target viewpoints, preventing participants from leveraging the intermediate views for perceiving the remote scene under these two conditions. For conditions *Manu-Cam* and *Dyan-Cam*, there is no black screen applied when the camera moves.

Hypotheses. H1: The proposed method (*Auto-Cam*) can provide users with a better control experience than the other methods. *H2*: The proposed method (*Auto-Cam*) can reduce the task load as compared to the other methods. *H3*: The proposed method (*Auto-Cam*) can achieve higher efficiency in the designated tasks.

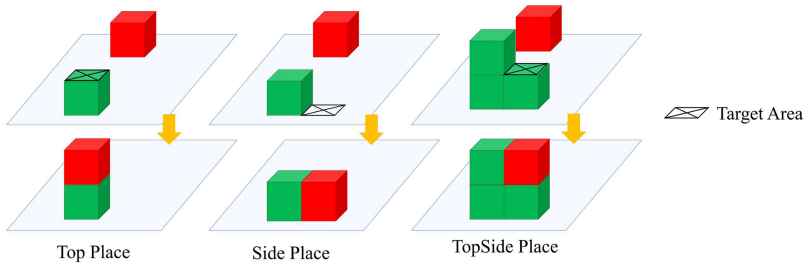


Fig. 5. In the pick-and-place tasks, the object in red is to be grasped and placed on the target area marked by an X, resulting in the target configurations shown in the bottom row.

Experimental Design. We designed a 4×1 within-participants experiment in which participants used the four camera control methods to complete a set of telemanipulation tasks in a counterbalanced order.

Study Tasks. The concerned objects were cubical wooden blocks with a size of 5 cm. The picking sub-task requires the participants to align the gripper with the block for a stable grasp, while the placing sub-task requires them to align the block in the gripper with different target configurations for stacking, as shown in Figure 5. Participants could only observe the remote workplace from the display showing the view of the single moving camera. Thus, the camera view needs to be adjusted from time to time to achieve these tasks.

Study Procedures. An experimenter first obtained informed consent, introduced a participant to the user study by explaining the goal, procedures, and tasks of this study, and guided the participant through an interactive session with the prototype system (Figure 2(a)). The experimenter initially provided users with an elucidation of the manipulation of the gripper's movements via the controller in *Manu-Cam* condition, including directional commands (e.g., move left or right), rotational instructions, and grasp. Since there are three ways of controlling the camera (with *Auto-Cam* and *Rand-Cam* being seen as the same for users), the participants were taught how to use the three control methods in the manipulation tasks. The experimenter explained how to use the *Manu-Cam* in the *top-place* task, the *Dyan-Cam* in the *side-place* task, and the *Auto-Cam* in the *top-side-place* task. The total tutorial took around 20 minutes: about 5 minutes for the introduction and 5 minutes for each task (or camera control method). Participants should be able to finish the task successfully in the tutorial session.

After the tutorial session, the participant was invited to perform the pick-and-place tasks. The four investigated conditions were presented in a counterbalanced order. The participant needs to complete the three tasks in each condition. Each task was repeated twice in succession, resulting in a total of six trials within a given condition. The initial states of the blocks and the gripper were fixed for each task, as shown in Figure 5. We randomly selected six initial viewpoints for each trial and initialized the camera arm to the corresponding viewpoint. The initial viewpoint of each task was the same for all four conditions.

After completing all trials under a condition, the participant filled out a subjective questionnaire regarding that condition and moved to the next one. A break was given to the participant after a condition. After finishing all conditions, the participant filled out a demographic survey, underwent a semi-structured interview with the experimenter, and received a coupon equivalent to 16 USD. It took approximately 100 minutes for each participant to finish the user study.

Table 2. Statements About the User's Perceived Control Experience

Goal understanding (Cronbach's $\alpha = 0.84$)	*Fluency* (Cronbach's $\alpha = 0.81$)
+ The robot perceives accurately what my goals are	+ The robot contributed to the fluency of the interaction
− The robot does not understand what I am trying to accomplish	+ The robot and I worked fluently together as a team
+ The robot and I are working toward mutually agreed upon	
Robot contributions (Cronbach's $\alpha = 0.71$)	*Ease-of-use* (Cronbach's $\alpha = 0.83$)
− I had to carry the weight to make the human–robot team better	+ The control method made it easy to accomplish the task
+ The robot contributed equally to the team performance	+ I felt confident controlling the robot
− I was the most important team member on the team	+ I could accurately control the robot
+ The robot was the most important team member on the team	

Statements related to Goal understanding, Fluency, and Robot contributions are adopted from [10], while those related to Ease-of-use are from [28].

Measures. We measured the alignment quality, success rate, and time to complete a trial. The alignment quality is measured by the **Intersection-over-Union (IoU)** of the top surface of the placed object and the target place area. A trial is considered successful if the IoU is larger than 0.3. Mistakenly triggering the pick-up action was viewed as failing to complete the trial.

The **National Aeronautics and Space Administration Task Load Index (NASA TLX)** [8] was adopted to measure the perceived workload. As shown in Table 2, we adopted questions commonly used in human-computer interaction from [10, 28] to measure perceived control experience. Specifically, the questions regarding *Goal Understanding*, *Fluency,* and *Robot contributions* are adopted from [10], while the questions regarding *Ease-of-use* are from [28]. All questions in these subjective questionnaires were evaluated with a seven-point Likert scale. For each measure in the perceived control experience, a higher rating indicates a better control experience.

Participants. Twenty-four participants with ages 23–49 (**Mean (M)**: 28; SD: 4.97) were recruited from a university campus. Eleven participants are female. Six participants have studied robotics but reported had never participated in similar research before. Two reporting previous experience in similar user studies were from majors other than robotics.

4.2 Results and Discussion of User Study 1

We analyzed all measured results using one-way repeated-measures analyses of variance. The camera control method was treated as the within-participants variable. We conducted a *post hoc* analysis, performing pairwise t-test between *Auto-Cam* and the other three conditions. The returned p values were adjusted using the Bonferroni correction by multiplying the p value by 3 (except *Goal Understanding*). The survey results are shown in Figure 6.

Perceived Control Experience. Our results partially support Hypothesis *H1*. Specifically, *Auto-Cam* outperforms all other three methods concerning *Fluency*. Additionally, *Auto-Cam* significantly improves the user control experience over *Manu-Cam* concerning *Robot contribution*, over *Dyan-Cam* in terms of *Ease-of-use*, and over *Rand-Cam* regarding *Goal Understanding*. Specifically, this

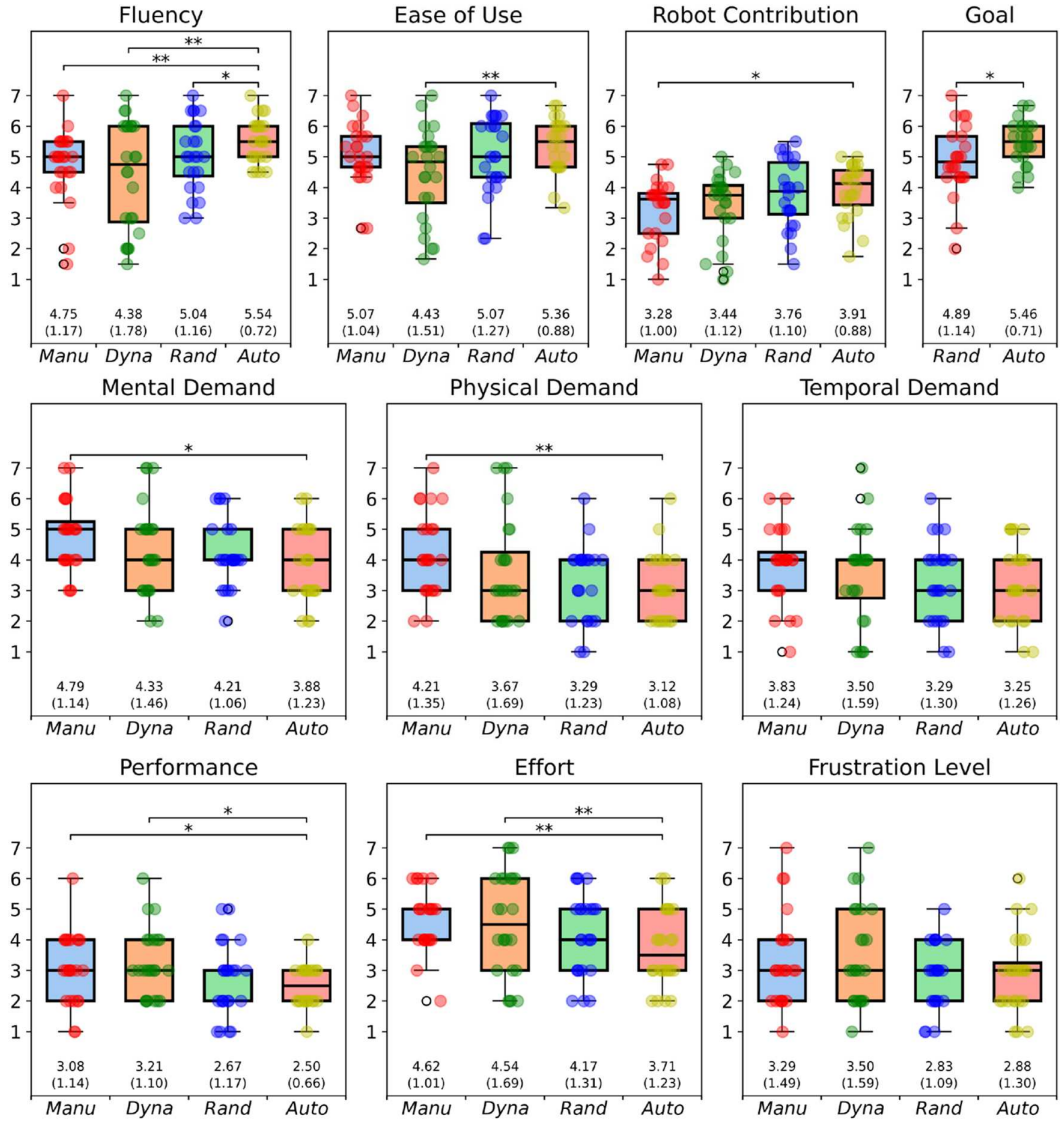


Fig. 6. Boxplots of the survey results of User Study 1. Average ratings and their SDs (in parentheses) are reported below the plots. Overall, participants perceived a *lower* task load and *improved* control experience with the proposed method (*Auto-Cam*). Different statistical significance levels are denoted with $^{*}p < 0.05$ and $^{**}p < 0.01$.

last comparison with *Rand-Cam* the random control group demonstrates that the proposed method (*Auto-Cam*) can encode task-related information to generate viewpoints preferable to users. Notably, our method still gains marginal improvement and exhibits smaller SDs for those aspects without statistically significant differences.

Perceived Task Load. Our results also partially advocate Hypothesis *H2*. Compared to *Manu-Cam*, our method (*Auto-Cam*) can significantly reduce the mental demand, physical demand, or effort

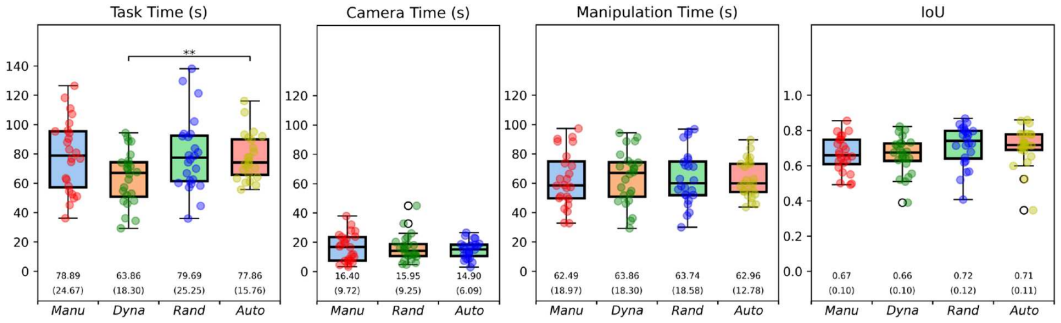


Fig. 7. User Study 1: Average time spent (in seconds) and IoU, along with their SDs (in parentheses) of each camera control mode, are reported below the plots.

required for completing the tasks, while improving the perceived performance. *Auto-Cam* also outperforms *Dyan-Cam* in terms of the perceived performance and effort. *Auto-Cam* receives slightly higher ratings than *Dyan-Cam* regarding mental and physical demand, indicating similar benefits of automatic camera control enjoyed by both methods. Since *Rand-Cam* and *Auto-Cam* differ only in how the viewpoint is synthesized, it is reasonable that they do not show significant differences in terms of the perceived task load.

Task Performance. Hypothesis *H3* was not confirmed in this user study. The results regarding task efficiency, which show no significant differences, align with the scores for Temporal Demand in the NASA-TLX survey. This may be attributed to the relatively low difficulty of the manipulation tasks designed for the user study.

Nevertheless, certain advantages of our method (*Auto-Cam*) over the other methods can be observed from Figure 7. First, our method achieved a shorter task time than both *Manu-Cam* and *Dyan-Cam*, both of which involve camera movements during the manipulation task. Second, when compared to *Dyan-Cam*, our method also yielded a lower manipulation time (total task time minus camera motion time).

Auto-Cam achieved the highest success rate (93.8%) compared to *Manu-Cam* (89.6%), *Dyna-Cam* (90.8%), and *Rand-Cam* (91.7%), while keeping a relatively high IoU (slightly failing behind *Rand-Cam*), as shown in Figure 7.

Post Hoc Power Analysis. The *post hoc* power analysis [39] with $1 - \beta = 0.80$ and $\alpha = 0.05$ found 24 participants are sufficient for most questions in User Study 1, with some exceptions such as Fluency in *Rand-Cam* vs. *Auto-Cam* ($p < 0.01$, $1 - \beta = 0.68$) and Effort in *Dyna-Cam* vs. *Auto-Cam* ($p < 0.01$, $1 - \beta = 0.75$).

User Preference. In the follow-up semi-structured interview of Study 1, we asked the subjects to rank each method according to their preference. The results are shown in Figure 8. Among them, 12 users liked *Auto-Cam* the most, while 8 liked *Manu-Cam*, 3 liked *Rand-Cam,* and 1 liked *Dyna-Cam* the most, respectively.

We also interviewed the participants about the positive and negative features of each camera control mode. For our *Auto-Cam* mode, seven participants thought *Auto-Cam* was "smart" as the recommended viewpoints were convenient for completing the tasks and requiring less effort compared to other camera control modes. But, on the other hand, four participants mentioned the predicted viewpoints were not what they wanted and two participants thought this method cost more time. This shows that our camera control method may satisfy the expectations of part of the participants.

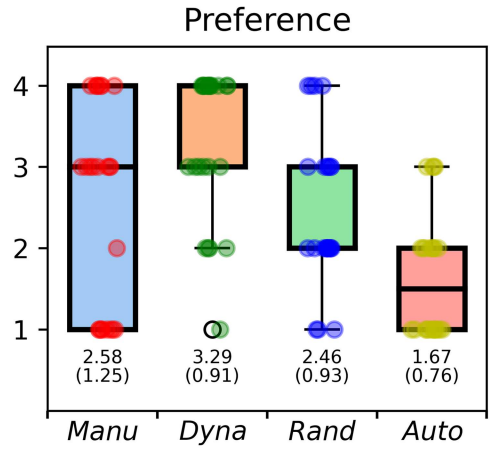


Fig. 8. User Study 1: Participants' preference toward different camera conditions. The number is the rank of different camera conditions: 1 indicates the highest rank and 4 means the lowest. Our method (*Auto-Cam*) was ranked first place by 12 out of 24 participants.

The *Rand-Cam* mode is similar to our *Auto-Cam* mode. From the interview, six participants were not satisfied with the viewpoints provided by *Rand-Cam*. But a bit contradictory, the overall IoU metric achieved in the *Rand-Cam* mode is slightly higher than that achieved in the *Auto-Cam* mode.

For the *Manu-Cam* mode, its controllability received most comments, both positive and negative. Those (10) who liked this mode usually mentioned they could navigate the camera to "go to the viewpoint they want," while those (7) against this mode complained about the effort required for controlling both the camera and the gripper.

Four participants thought the *Dyna-Cam* mode could boost the task efficiency (as no camera traveling time) while reducing their effort and workload; while more than half of the participants (13) found *Dyna-Cam* difficult to use as the camera view could reveal little about the spatial relationship (e.g., distance) between the gripper and the objects of interest.

In summary, half of the participants ranked *Auto-Cam* as their first preference among the four methods. However, subjective comments on each method are somewhat conflicting among the participants.

General Discussion. In summary, based on both the subjective questionnaire responses and the quantitative analysis of the results, the proposed camera control method (*Auto-Cam*) demonstrates the ability to enhance users' control experience, particularly in terms of control fluency. It also outperforms each of the other methods in a different aspect related to control experience. In terms of task load, our method significantly reduces the task load compared to *Manu-Cam*. However, it's worth noting that we did not observe a clear improvement in task efficiency.

It is intriguing to understand how participants were able to achieve the manipulation tasks in Condition *Dyan-Cam* with similar efficiency as other methods. Through our semi-structured post-experiment interviews, we found that the primary reason is as follows. In Condition *Dyan-Cam*, the camera's spatial position changes as the gripper moves to maintain a fixed viewpoint of the camera focused on the gripper. As they explained in the interviews, participants can exploit the limited camera movement and the disparity created by these continuous changes in perspective to perceive the three-dimensional remote scene and complete the tasks, albeit at the cost of expending more effort, resulting in lower ratings for *Fluency* and *Ease-of-use*.

Most participants considered both *Manu-Cam* and *Auto-Cam* to be easy to use, and as a result, no statistically significant difference in terms of *Ease-of-use* was observed between the two methods. However, based on the interviews, some participants believed that *Manu-Cam* offered higher control accuracy than *Auto-Cam* because it allowed them to position the camera at their preferred viewpoint precisely. Another frequently mentioned reason was that *Manu-Cam* permitted users to see intermediate views during the camera movement, which provided valuable information for a better understanding of the remote environment.

4.3 User Study 2: Assistive Camera Mode vs. Manual Camera Mode

In the previous user study, our primary focus was on evaluating the effectiveness of the proposed viewpoint prediction model. In that study, we masked out the intermediate views during camera transitions and did not allow users to override the suggested viewpoints provided by our system. In this second user study, we removed these constraints and integrated the viewpoint prediction model as an integral part of the telemanipulation system for evaluation.

Specifically, we implemented the proposed automatic camera placement method as an assistive technique (referred to as *Assist-Cam*), activating it only upon the user's request. Under this *Assist-Cam* condition, the participants had the option to reject the proposed viewpoint by *Assist-Cam* and manually adjust the camera viewpoint. We were especially interested in understanding how the proposed method, when utilized as an assistive technique, would be employed by users and how its performance would compare to the manual-controlled camera mode (i.e., *Manu-Cam*).

Hypotheses. Following hypotheses were investigated: *H1*: The proposed method (*Assist-Cam*) can provide users with a better control experience than Condition *Manu-Cam*. *H2*: The proposed method (*Assist-Cam*) can reduce task load for the users as compared with Condition *Manu-Cam*. *H3*: The proposed method (*Assist-Cam*) can achieve higher efficiency in the telemanipulation tasks than Condition *Manu-Cam*.

Study Tasks and Procedures. In this study, only the pick-and-place task with the *top-place* target configuration was considered. The same cubical wooden blocks were used.

An experimenter first obtained informed consent and introduced a participant to the user study by explaining the goal, procedures, and tasks of this study. The experimenter guided the participant through an interactive session with the telemanipulation system and explained to the participant how to use the controllers to achieve certain simplified goals (e.g., move left or right). The participant was given 8 minutes and then 4 minutes to interact with conditions *Manu-Cam* and *Assist-Cam*, respectively.

After the tutorial session, the participant was invited to perform the pick-and-place tasks. The two investigated conditions were presented to a participant in a counterbalance order. The participant performed the designated task four times. The initial states of the robot arms were set to the same, while the cubes were randomly placed in the remote workplace. After completing all four trials under each condition, the participant filled out a questionnaire regarding that condition and moved on to the next condition. In between two conditions, a break was given to the participant. After finishing all conditions, the participant filled out a demographic survey, underwent a semi-structured interview with the experimenter, and received a coupon equivalent to 16 USD. It took approximately 1 hour for each participant to finish the user study.

Measures. We adopt the same metrics to quantify the performance of each trial as described in User Study 1. The same subjective survey was conducted to measure the perceived workload and the perceived control experience. Note that the questions related to *Goal Understanding* were excluded for a fair comparison in this user study.

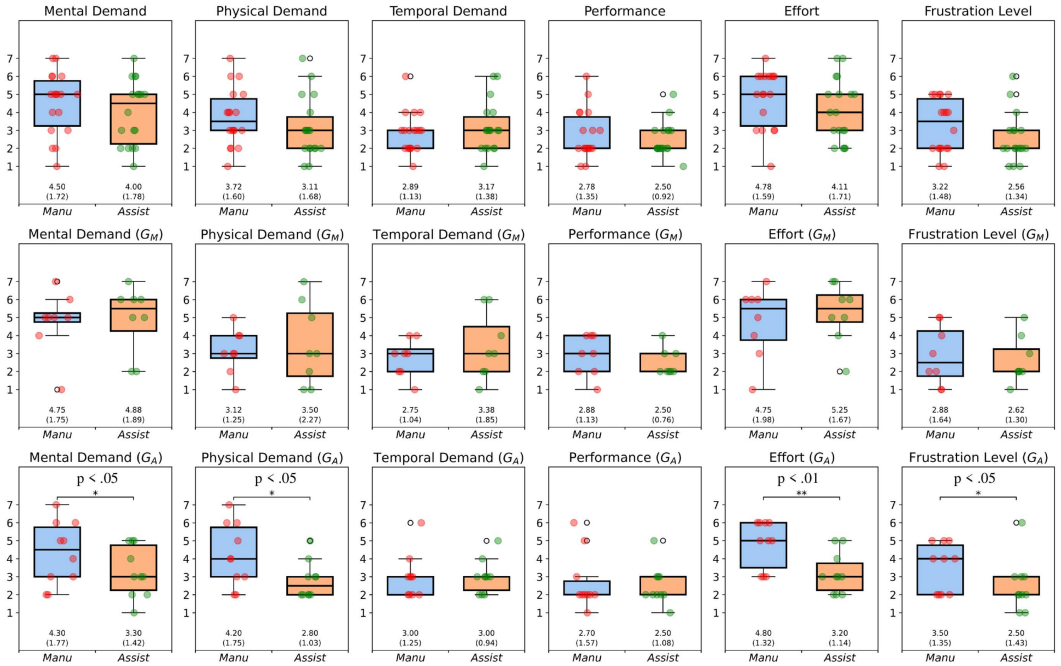


Fig. 9. Perceived task load results of Study 2 (higher score means higher load). Under Condition *Assist-Cam*, users in the G_A group experience lower levels of mental and physical load, report fewer feelings of frustration, and exert less effort in completing tasks.

Participants. Eighteen participants with ages 19–41 (mean: 26.4; SD: 5.9) were recruited from the same university campus. Eleven participants are male. Three participants have studied robotics but reported having never participated in similar research. Two reporting previous experience in similar user studies were from other engineering majors. There is no overlap between the two groups of participants from User Study 1 and this user study.

4.4 Results and Discussion of User Study 2

We performed paired t-tests for each measure between the two conditions, *Assist-Cam* and *Manu-Cam*, to check if there is a statistically significant difference. The results are shown in Figures 9 and 10.

Acceptance of the Assistive System. In this user study, we observed that some participants hesitated to accept the camera viewpoint recommended by *Assist-Cam*. These participants quickly assumed camera control after triggering *Assist-Cam*. To quantify this behavior, we computed the ratio of *Assist-Cam* time to the total camera control time: $r_{Auto} = t_{Auto}/(t_{Auto} + t_{Manu})$, where t_{Auto} and t_{Manu} represent the time the camera was controlled by *Assist-Cam* and the time controlled manually by participants, respectively. Based on the median value of r_{Auto} (0.67), we divided participants into two groups: G_A ($r_{Auto} \geq 0.67$, $N = 10$), who accepted *Assist-Cam*, and G_M ($r_{Auto} < 0.67$, $N = 8$), who more frequently overrode the suggested viewpoints provided by *Assist-Cam*. We found a significant statistical difference between the two groups of participants ($p < 0.001$), which validates the effectiveness of this clustering scheme for the participants.

Perceived Control Experience and Task Load. The statistics of the perceived control experience and task load are shown in Figures 10(a) and 9, respectively.

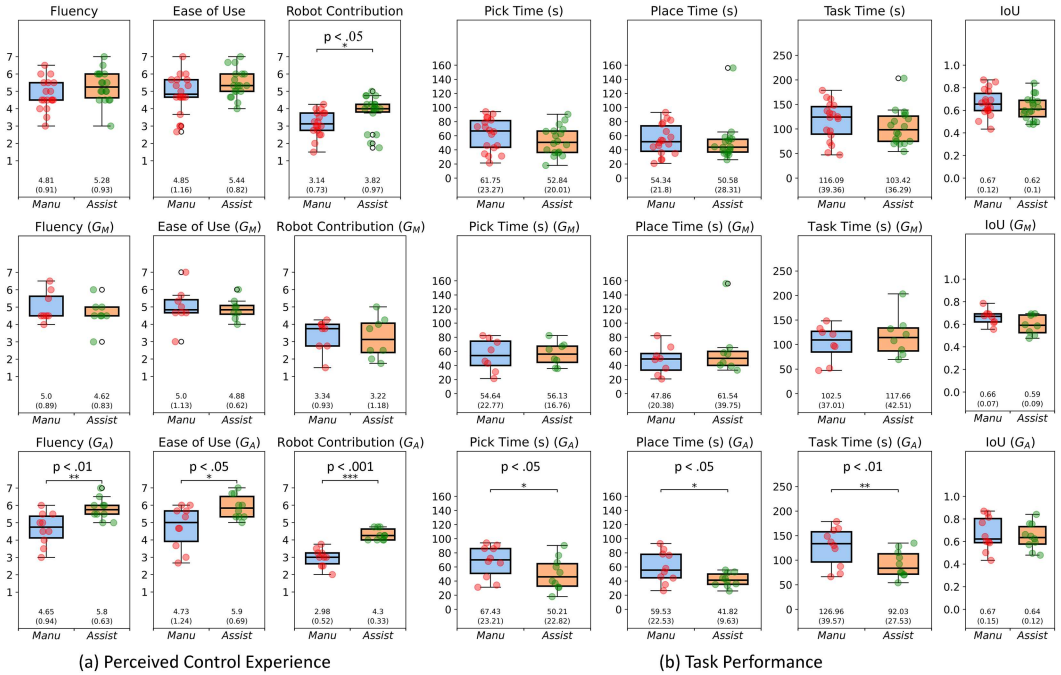


Fig. 10. Boxplots of the (a) perceived control experience and (b) task performance of User Study 2. Conditioned on the participants' preference, *Assist-Cam* exhibits significantly better control experience perceived by the participants in G_A (who rely more on *Assist-Cam* to control the camera viewpoint). Higher task efficiency was observed in group G_A as well, achieving a 25% reduction of the total task completion time.

When considering all participants without grouping, we found that *Assist-Cam* significantly improved the *Robot contribution*[2] over *Manu-Cam* and was rated favorably on average in terms of *Fluency* and *Ease-of-use*. When examining the group G_A, we observed that our method achieved better control experience in all aspects, as shown in Figure 10(a). Therefore, *H1* holds when participants accept the proposed system (G_A). Additionally, we observed statistically significant differences in terms of *Mental Demand*, *Physical Demand*, *Effort*, and *Frustration Level* among participants in G_A, partially confirming *H2* within the context of participants' preferences. Regarding participants in G_M, we observed that the performances when using *Assist-Cam* and *Manu-Cam* were very similar. This aligns with the fact that participants in this group primarily relied on the manual control mode within Condition *Assist-Cam*.

Task Efficiency and Success Rate. Hypothesis *H3* was confirmed as "true" for the group of participants (G_A) who accepted our proposed system. As indicated in Table 3, the proposed method (*Assist-Cam*) led to a reduction of more than 25% in the averaged task time for G_A, and significant differences were also observed for each sub-task. However, for group G_M, the task time under Condition *Assist-Cam* experienced a slight increase, yet no significant difference was observed.

When examining the entire population, we observed that the time spent on picking/placing tasks under Condition *Assist-Cam* was, on average, approximately 9/4 seconds shorter than *Manu-Cam*. However, it's important to note that *Manu-Cam* achieved a higher M IoU and success rate (IoU: *Manu-Cam* 0.67 vs. *Assist-Cam* 0.62; success rate: *Manu-Cam* 85.7% vs. *Assist-Cam* 82.9%) in terms

[2]Here, "robot" refers to the entire telemanipulation system.

Table 3. Task Efficiency

Groups	Pick	Place	Total
All	52.8/61.7 (14.4%, 0.71)	50.6/54.3 (6.9%, 0.58)	103.4/116.1 (10.9%, 0.20)
G_M	56.1/54.6 (−2.7%, 0.69)	61.5/47.9 (−28.6%, 0.19)	117.7/102.5 (−14.8%, 0.21)
G_A	50.2/67.4 (25.5%, 0.03)	41.8/59.5 (29.8%, 0.02)	92.0/127.0 (27.5%, 0.01)

We report M operation time (in seconds) with *Assist-Cam*/*Manu-Cam* modes and the relative differences computed as $(t_{Manu} - t_{Auto})/t_{Manu}$ % followed by the p value in parentheses. Total task time was largely reduced (>25%) among the participants in the group G_A who chose to adopt the proposed system.

Table 4. Time Distribution of Camera Control under Condition *Assist-Cam*

Groups	Each trial			Each camera movement		
	Total	*Manual*	$1 - r_{Auto}$	*Total*	*Manual*	*MPaI*
All	29.34 (21.17)	12.22 (13.36)	33.9% (33.0%)	4.81 (1.58)	1.50 (1.65)	29.4% (30.1%)
G_M	37.00 (24.85)	23.19 (12.52)	65.3% (21.0%)	4.50 (1.60)	2.95 (1.45)	57.4% (22.1%)
G_A	22.80 (14.91)	3.45 (4.77)	8.8% (11.6%)	5.05 (1.61)	0.35 (0.44)	7.2% (9.3%)

Statistics in the left columns show the manual camera time and total camera time averaged over each trial of a complete pick-and-place task; $1 - r_{Auto}$ represents the average percentage of manual camera control for a trial. Statistics in the right columns show the manual camera time and total camera time averaged over each camera movement; **Manual Percentage after Interrupt (MPaI)** represents the average percentage of manual camera control for one camera movement. All the M values with SDs in parentheses are provided.

of the overall participant population. Since in the *Manu-Cam* mode, more time and effort (both mentally and physically) were devoted to the pick-and-place task, the performances participants achieved can be considered as a ceiling performance for the task. We note that the performance obtained with our *Assist-Cam* mode did not drop much, showing that *Assist-Cam* can generally provide satisfactory viewing angles for users to complete the task.

Post hoc Power Analysis. We ran the post hoc power analysis [39] with $1 - \beta = 0.80$ and $\alpha = 0.05$ for examining if the number of participants is sufficiently large for group G_A. We found that 10 participants are enough for the perceived control experience questions, while it falls a bit short regarding the NASA-TLX questions in User Study 2. Future studies should recruit more participants to draw more conclusive results.

Further Discussion on User Preference. We analyzed the percentage of manual camera control that interrupts autonomous camera movement produced by *Assist-Cam*. The statistics are reported in Table 4. Participants in G_A had a notably low manual control ratio over the entire pick-and-place task trial, which was only 8.8%. In contrast, participants in G_M had a much higher ratio, with manual control accounting for as much as 65% of the camera time.

We also calculated a similar manual camera control ratio with respect to each camera movement in between two consecutive gripper movements. This ratio, referred to as **Manual Percentage after Interrupt (MPaI)**, indicates the extent to which the user-preferred viewpoint is away from the generated path. As shown in Table 4, the M MPaI for G_A is 7.2%, indicating that the manual control per camera movement is less than 1 second. This suggests that the user-preferred viewpoint was relatively close to the generated path for transporting the camera to the predicted viewpoint.

With the same viewpoint prediction model, participants in G_M spent an average of 2.95 seconds, accounting for 57% of the total time of one camera movement, to manually adjust the camera pose.

We have identified several possible reasons why participants chose to override the recommended camera viewpoints generated by *Assist-Cam* based on insights gained from participant interviews.

First, seven participants changed the camera movement because they believed the recommended viewpoints were not their desired viewpoints for the task, e.g., "not the angle I wanted to watch" or "cannot reach an angle I wanted." We acknowledge the fact that the optimal viewpoint is not unique, as noted in previous research [5], that our model's limitation of learning viewing preference/bias from the data we collected, and its occasional failure to recommend informative viewpoints that can reflect the relative spatial relationship between the gripper and the objects.

Second, four participants found the camera movement too slow and preferred to accelerate the process manually. One of these participants experienced the failure of the inverse kinematics solver for the camera robot arm's trajectory and manually controlled the camera to recover. This experience might have reduced the participant's faith in the proposed method, and we observed an increased ratio of manual control afterward. The remaining three participants mentioned that "the camera moves too slow" so they took over the control or "It takes a long time, but the camera still does not reach my desired viewpoint." Unfortunately, we are unable to reproduce the situations these participants experienced and examine how distant the recommended viewpoints are from their manually selected ones. This can be a consideration for future studies.

Third, during the user study, we observed that some participants preferred to repetitively check the remote scene from a sparse set of views before taking action to pick or place an object. This pattern was observed strongly at least in three participants' video data records. This pattern significantly differs from the data we collected to train the viewpoint prediction model, resulting in our method failing to provide desirable views.

Lastly, we reported an outstanding participant who directly adopted the manual mode throughout the entire *Assist-Cam* condition as the participant indicated his/her preference for the *Manu-Cam* mode. By analyzing this participant's data, we found that the participant experienced a higher task load than the rest of the participants. Among many other possible follow-up studies, an interesting direction is to explore if the bipolar trend observed in Study 2 is correlated to the participants' acceptance of novel techniques.

Result Visualization. Finally, we provide some qualitative results regarding the predicted viewpoints in the pick-and-place tasks in Figure 11. Four maps are depicted. Each map shows the frequency of each viewpoint (whose coordinate system is centered at the target object but has the axes aligned with the world coordinate) visited right before performing the picking or placing actions. From this figure, strong agreement between the distribution of the predicted viewpoints and that of the viewpoints manually chosen by the participants was observed. This agreement demonstrates the proposed model's ability to generate potentially desirable viewpoints for manipulation. As seen from the snapshots on the right, the proposed viewpoints are able to reduce the depth ambiguity effectively. For example, snapshots for the placing sub-tasks (D, E, F, and G) facilitate the alignment of the two objects by viewing them from a perspective view.

5 Concluding Remarks

In this article, we introduce the first learning-based method for automating camera placement for telemanipulation. Our method is trained effectively with a small set of human demonstrations using a contrastive learning strategy that leverages the viewpoints on a camera trajectory.

In the first user study, we validated the effectiveness of our camera control method in improving the control experience, particularly in terms of control fluency. Our proposed method demonstrates a

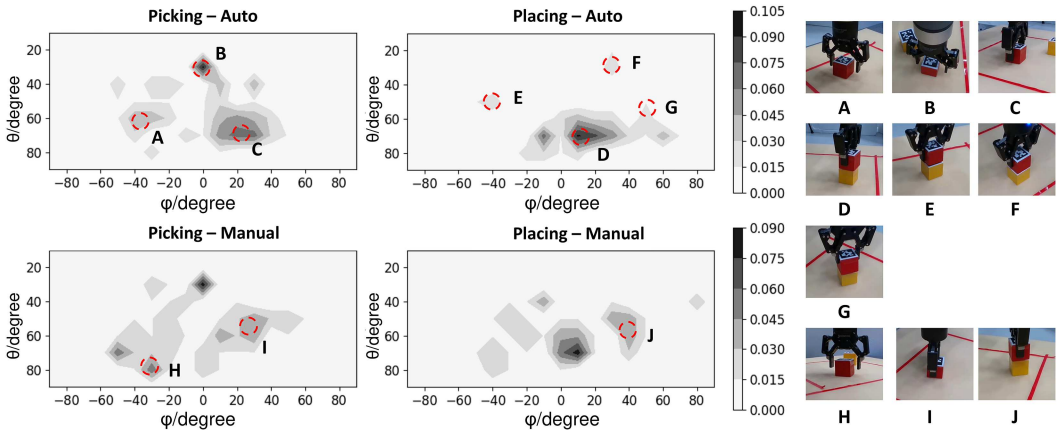


Fig. 11. Each map shows the frequency of each viewpoint being visited right before performing the picking or placing actions. Strong agreement was observed between the predicted results and the manual ones.

statistically significant difference from random camera control, highlighting its ability to recommend potentially desirable viewpoints to users. Additionally, our method proves to be effective in reducing perceived task load when compared to manual camera control.

In the second user study, we compared an assistive camera control mode based on our method to manual camera control. We observed a disparity among the participants in accepting the assistive camera control. Among the users who accepted it, the assistive camera control significantly enhanced the perceived control experience and reduced around 25% of the task completion time.

Limitations. While our viewpoint prediction model can generate several candidate viewpoints, it currently selects only the one with the highest score as the target viewpoint. Exploring the concept of a shared autonomy system is a promising avenue, where user input can serve as a hint to interpret the user's intent. Currently, we simplify the automatic camera placement problem by focusing on predicting a stationary destination for the camera. A potential direction for future exploration is to consider the value of intermediate views to users and formulate it as a path-planning problem.

References

[1] Universal Robots A/S. 2023. *Universal*_Robots_ROS_Driver. Retrieved from https://github.com/UniversalRobots/Universal_Robots_ROS_Driver

[2] Bernard G. Brooks and Gerard T. McKee. 2001. The visual acts model for automated camera placement during teleoperation. In *Proceedings of the 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164)*, Vol. 1. IEEE, 1019–1024.

[3] Andre Cleaver and Jivko Sinapov. 2023. Helping humans become better teachers for robots with augmented reality. Retrieved from https://openreview.net/forum?id=LXgHj3JZtB

[4] Francesco De Pace, Gal Gorjup, Huidong Bai, Andrea Sanna, Minas Liarokapis, and Mark Billinghurst. 2020. Assessing the suitability and effectiveness of mixed reality interfaces for accurate robot teleoperation. In *Proceedings of the 26th ACM Symposium on Virtual Reality Software and Technology*, 1–3.

[5] Jan Dufek, Xuesu Xiao, and Robin R. Murphy. 2021. Best viewpoints for external robots or sensors assisting other robots. *IEEE Transactions on Human-Machine Systems* 51, 4 (2021), 324–334.

[6] Francois Ferland, Francois Pomerleau, Chon Tam Le Dinh, and Francois Michaud. 2009. Egocentric and exocentric teleoperation interface using real-time, 3D video projection. In *Proceedings of the 4th ACM/IEEE International Conference on Human Robot Interaction*, 37–44.

[7] Anton Franzluebbers and Kyle Johnson. 2019. Remote robotic arm teleoperation through virtual reality. In *Proceedings of the Symposium on Spatial User Interaction*, 1–2.
[8] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Advances in Psychology*, Peter A. Hancock, Najmedin Meshkati, (Eds.), Vol. 52. Elsevier, 139–183.
[9] Hooman Hedayati, Michael Walker, and Daniel Szafir. 2018. Improving collocated robot teleoperation with augmented reality. In *Proceedings of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 78–86.
[10] Guy Hoffman. 2019. Evaluating fluency in human–robot collaboration. *IEEE Transactions on Human-Machine Systems* 49, 3 (2019), 209–218.
[11] Stephen James, Zicong Ma, David Rovick Arrojo, and Andrew J. Davison. 2020. RLBench: The robot learning benchmark & learning environment. *IEEE Robotics and Automation Letters* 5, 2 (2020), 3019–3026.
[12] David Kent, Carl Saldanha, and Sonia Chernova. 2017. A comparison of remote robot teleoperation interfaces for general object manipulation. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 371–379.
[13] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. arXiv:1412.6980. Retrieved from https://arxiv.org/abs/1412.6980
[14] Jonathan Kofman, Xianghai Wu, Timothy J. Luu, and Siddharth Verma. 2005. Teleoperation of a robot manipulator using a vision-based human-robot interface. *IEEE Transactions on Industrial Electronics* 52, 5 (2005), 1206–1219.
[15] Ewen B. Lavoie, Aïda M. Valevicius, Quinn A. Boser, Ognjen Kovic, Albert H. Vette, Patrick M. Pilarski, Jacqueline S. Hebert, and Craig S. Chapman. 2018. Using synchronized eye and motion tracking to determine high-precision eye-movement patterns during object-interaction tasks. *Journal of Vision* 18, 6 (2018), 18–18.
[16] Yuan Liu, Yilin Wen, Sida Peng, Cheng Lin, Xiaoxiao Long, Taku Komura, and Wenping Wang. 2022. Gen6D: Generalizable model-free 6-DoF object pose estimation from RGB images. In *Proceedings of the European Conference on Computer Vision*. Springer, 298–315.
[17] Gerard T. McKee and Paul S. Schenker. 1995. Human-robot cooperation for automated viewing during teleoperation. In *Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems. Human Robot Interaction and Cooperative Robots*, Vol. 1. IEEE, 124–129.
[18] Alexander M. Morison, Taylor Murphy, and David D. Woods. 2016. Seeing through multiple sensors into distant scenes: The essential power of viewpoint control. In *Proceedings of the International Conference on Human-Computer Interaction*. Springer, 388–399.
[19] Robin R. Murphy. 2015. Meta-analysis of autonomy at the DARPA robotics challenge trials. *Journal of Field Robotics* 32, 2 (2015), 189–191.
[20] Robin R. Murphy and Satoshi Tadokoro. 2019. User interfaces for human-robot interaction in field robotics. In *Disaster Robotics: Results from the ImPACT Tough Robotics Challenge*. Satoshi Tadokoro (Ed.), Springer, 507–528.
[21] Benjamin A. Newman, Reuben M. Aronson, Siddhartha S. Srinivasa, Kris Kitani, and Henny Admoni. 2022. HARMONIC: A multimodal dataset of assistive human–robot collaboration. *The International Journal of Robotics Research* 41, 1 (2022), 3–11.
[22] Davide Nicolis, Marco Palumbo, Andrea Maria Zanchettin, and Paolo Rocco. 2018. Occlusion-free visual servoing for the shared autonomy teleoperation of dual-arm robots. *IEEE Robotics and Automation Letters* 3, 2 (2018), 796–803.
[23] Bukeikhan Omarali, Brice Denoun, Kaspar Althoefer, Lorenzo Jamone, Maurizio Valle, and Ildar Farkhatdinov. 2020. Virtual reality based telerobotics framework with depth cameras. In *Proceedings of the 2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 1217–1222.
[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 32 (2019), 8026–8037.
[25] Pragathi Praveena, Luis Molina, Yeping Wang, Emmanuel Senft, Bilge Mutlu, and Michael Gleicher. 2022. Understanding control frames in multi-camera robot telemanipulation. In *Proceedings of the 2022 ACM/IEEE International Conference on Human-Robot Interaction*, 432–440.
[26] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2017. A motion retargeting method for effective mimicry-based teleoperation of robot arms. In *Proceedings of the 2017 ACM/IEEE International Conference on Human-Robot Interaction*, 361–370.
[27] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2018. An autonomous dynamic camera method for effective remote teleoperation. In *Proceedings of the 2018 13th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 325–333.

[28] Daniel Rakita, Bilge Mutlu, and Michael Gleicher. 2019. Remote telemanipulation with adapting viewpoints in visually complex environments. In *Proceedings of the Robotics: Science and Systems XV*, 68.
[29] Daniel Rakita, Haochen Shi, Bilge Mutlu, and Michael Gleicher. 2021. CollisionIK: A per-instant pose optimization method for generating robot motions with environment collision avoidance. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 9995–10001.
[30] Nicklas Ritola, Alberto Giaretta, and Andrey Kiselev. 2023. Operator identification in a VR-based robot teleoperation scenario using head, hands, and eyes movement data. Retrieved from https://openreview.net/forum?id=Xals4UE6ZS
[31] Eric Rosen, David Whitney, Elizabeth Phillips, Daniel Ullman, and Stefanie Tellex. 2018. Testing robot teleoperation using a virtual reality interface with ROS reality. In *Proceedings of the 1st International Workshop on Virtual, Augmented, and Mixed Reality for HRI (VAM-HRI)*, 1–4.
[32] Emmanuel Senft, Michael Hagenow, Pragathi Praveena, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2022. A method for automated drone viewpoints to support remote robot manipulation. In *Proceedings of the 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 7704–7711.
[33] Emmanuel Senft, Michael Hagenow, Kevin Welsh, Robert Radwin, Michael Zinn, Michael Gleicher, and Bilge Mutlu. 2021. Task-level authoring for remote robot teleoperation. *Frontiers in Robotics and AI* 8 (2021), 707149.
[34] Mahmood M. Shilleh, Qusai A. Amer, Jan Dufek, and Robin R. Murphy. 2021. Best and worst external viewpoints for teleoperation visual assistance. In *Proceedings of the Companion of the 2021 ACM/IEEE International Conference on Human-Robot Interaction*, 675–676.
[35] Vincent Sitzmann, Julien Martel, Alexander Bergman, David Lindell, and Gordon Wetzstein. 2020. Implicit neural representations with periodic activation functions. In *Proceedings of the Advances in Neural Information Processing Systems*, Vol. 33 (2020), 7462–7473.
[36] Mehdi Sobhani, Alex Smith, Manuel Giuliani, and Tony Pipe. 2022. Usability study of a novel triple-arm mixed-reality robotteleoperation system. In *Proceedings of the 2022 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*. IEEE, 217–223.
[37] Alexandra Valiton, Hannah Baez, Naomi Harrison, Justine Roy, and Zhi Li. 2021. Active telepresence assistance for supervisory control: A user study with a multi-camera tele-nursing robot. In *Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3722–3727.
[38] Alexandra Valiton and Zhi Li. 2020. Perception-action coupling in usage of telepresence cameras. In *Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3846–3852.
[39] Raphael Vallat. 2018. Pingouin: Statistics in Python. *Journal of Open Source Software* 3, 31 (2018), 1026.
[40] Michael E. Walker, Hooman Hedayati, and Daniel Szafir. 2019. Robot teleoperation with augmented reality virtual surrogates. In *Proceedings of the 2019 14th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 202–210.
[41] Dong Wei, Bidan Huang, and Qiang Li. 2021. Multi-view merging for robot teleoperation with virtual reality. *IEEE Robotics and Automation Letters* 6, 4 (2021), 8537–8544.
[42] Yilin Wen, Hao Pan, Lei Yang, and Wenping Wang. 2020. Edge enhanced implicit orientation learning with geometric prior for 6D pose estimation. *IEEE Robotics and Automation Letters* 5, 3 (2020), 4931–4938.
[43] Xuesu Xiao, Jan Dufek, and Robin R. Murphy. 2021. Autonomous visual assistance for robot operations using a tethered UAV. In *Proceedings of the Field and Service Robotics: Results of the 12th International Conference*. Springer, 15–29.
[44] Yiheng Xie, Towaki Takikawa, Shunsuke Saito, Or Litany, Shiqin Yan, Numair Khan, Federico Tombari, James Tompkin, Vincent Sitzmann, and Srinath Sridhar. 2022. Neural fields in visual computing and beyond. In *Proceedings of the Computer Graphics Forum*, Vol. 41. Wiley Online Library, 641–676.
[45] Dingyun Zhu, Tom Gedeon, and Ken Taylor. 2011. Exploring camera viewpoint control models for a multi-tasking setting in teleoperation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 53–62.

Received 6 February 2023; revised 26 February 2024; accepted 7 March 2024