

Guided Learning from Demonstration for Robust Transferability

Fouad Sukkar*, Victor Hernandez Moreno*, Teresa Vidal-Calleja and Jochen Deuse

Abstract— Learning from demonstration (LfD) has the potential to greatly increase the applicability of robotic manipulators in modern industrial applications. Recent progress in LfD methods have put more emphasis in learning robustness than in guiding the demonstration itself in order to improve robustness. The latter is particularly important to consider when the target system reproducing the motion is structurally different to the demonstration system, as some demonstrated motions may not be reproducible. In light of this, this paper introduces a new guided learning from demonstration paradigm where an interactive graphical user interface (GUI) guides the user during demonstration, preventing them from demonstrating non-reproducible motions. The key aspect of our approach is determining the space of reproducible motions based on a motion planning framework which finds regions in the task space where trajectories are guaranteed to be of bounded length. We evaluate our method on two different setups with a six-degree-of-freedom (DOF) UR5 as the target system. First our method is validated using a seven-DOF Sawyer as the demonstration system. Then an extensive user study is carried out where several participants are asked to demonstrate, with and without guidance, a mock weld task using a hand held tool tracked by a VICON system. With guidance users were able to always carry out the task successfully in comparison to only 44% of the time without guidance.

I. INTRODUCTION

In modern industrial robotic manipulator applications there is a desire for greater autonomy through adaptability to novel tasks without requiring time consuming and costly reprogramming. Unlocking this potential would present the factories of tomorrow with the opportunity to shift from mass production towards mass customisation [1]. Learning from demonstration (LfD) methods are a promising direction for achieving this autonomy [2]. LfD enables non-robot experts to intuitively program robots that reproduce motions with high precision. However, often LfD methods limit this potential by restricting the demonstration to occur on the system reproducing the motion, for example via kinaesthetic teaching or teleoperation in order to guarantee faithful motion reproduction [3], [4], [5], [6], [7], [8], [9], [10].

We are interested in removing this restriction and allowing the demonstration and target systems to be different.

*These authors are co-first authors and contributed equally to this work. This research is partially supported by the University of Technology Sydney's President and International Research Scholarships and the Industrial Transformation Training Centre (ITTC) for Collaborative Robotics in Advanced Manufacturing (also known as the Australian Cobotics Centre) funded by ARC (Project ID: IC200100001).

Authors are with the UTS Robotics Institute and the Centre for Advanced Manufacturing, School for Mechanical and Mechatronic Engineering, University of Technology Sydney, 2007, Ultimo, NSW, Australia and the Australian Cobotics Centre (ITTC for Collaborative Robotics in Advanced Manufacturing) {Fouad.Sukkar@, Victor.HernandezMoreno@student., Teresa.VidalCalleja@, Jochen.Deuse@}uts.edu.au.

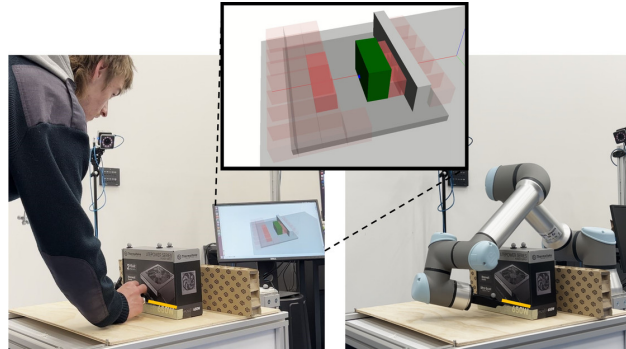


Fig. 1: Our proposed guided learning from demonstration approach with interactive GUI (inset) showing regions of reproducible motions to the demonstrator based on the kinematics of the target robot system. The demonstrator shown here is a non-expert and is carrying out the demonstration without the physical robot present.

Motivating use-cases include the following: demonstrating tasks naturally by tracking human motion directly or with an unconstrained tool detached from the robot, carrying out demonstrations in parallel while the target robot executes other tasks, and demonstrating on a smaller/safer robot in order to carry out dangerous or labour intensive tasks with a more powerful robot.

The challenge in this scenario is that there is no longer a direct mapping between the demonstration and target systems due to their differing kinematic structures. A common strategy is to encode the demonstration as the robot's end-effector trajectory. Doing so provides a common representation of the motion that can be interpreted by the target system [11], [12]. However, in these settings the kinematic discrepancy is often ignored or worked around by utilising a redundant robot for motion reproduction [13], [14], [15], [16]. These robots have more degrees of freedom (DOF) than the dimension of the task space [17] and can utilise null-space control methods in order to satisfy kinematic constraints, such as avoiding obstacles, singularities and joint limits, whilst maintaining faithful reproduction of the demonstrated end-effector motion [2].

In many applications a non-redundant arm may be preferable over a redundant arm, for example a six-DOF over a seven-DOF, due to being relatively lower cost, exhibiting higher payload-to-weight ratio and having simpler kinematics for control and planning. However, in certain cases these arms cannot exploit null-space control [17] due to their kinematic structure and as a consequence may require large joint changes between intermediate points of a demonstrated trajectory in order to satisfy kinematic constraints.

Hence the only option is to artificially restrict the motions of the demonstrator. While there exists work along this line through the form of graphical user interfaces (GUI) [18] and augmented reality [19], [20], [21], these methods usually consider only verifying, adjusting or adding loosely defined workspace and task constraints which do not directly consider the kinematic constraints of the target system.

In this paper we formulate a new guided learning from demonstration problem which explicitly considers these kinematic constraints and prevents non-reproducible motions from being demonstrated. To compute the space of reproducible motions we leverage an existing motion planning framework called Hausdorff approximation planner (HAP) [22] which finds regions in the task space where trajectories are guaranteed to be of bounded length. A novel interactive graphical user interface (GUI) then visualises this space to the user at demonstration time. Our GUI is intuitive and requires no expert knowledge about the target system from the demonstrator (see Fig. 1). For encoding and reproducing demonstrated motions, we use Cartesian space Dynamic Movement Primitives (CDMPs) [23] which facilitate a common representation space.

We evaluate our method in a workbench environment on two sets of experiments with a six-DOF UR5 as the target system. First our method is validated using a seven-DOF Sawyer as the demonstration system. Then an extensive user study is carried out where several participants are asked to demonstrate a mock weld task using a hand held tool tracked by a VICON system. Users carry out the task with and without the interactive GUI and results show that with guidance from the GUI users had 100% success rate in carrying out the task versus 44% without and were on average 30% more confident in carrying out the task.

The significance of our framework is that it enables non-expert demonstrators to confidently carry out feasible demonstrations without the presence of the physical target system and enables a greater range of possible demonstration and target system setups. The key contributions in this paper are: (1) A new guided learning from demonstration problem formulation for ensuring robust motion transfer onto a target system. (2) A novel approach to tackle this problem via a method for computing regions of reproducible motions and an interactive GUI for displaying these regions to the demonstrator.

II. PROBLEM FORMULATION

In this section, we formulate the problem of ensuring robust motion transfer from a demonstration system to a target system with differing kinematic structures. First we describe the problem setup including assumptions about the demonstration system, target system, environment and operational scenario. Then we formulate and state the problem of finding a region of reproducible motions for ensuring robust transferability.

A. Problem Setup

We consider a non-trivial LfD scenario where we wish to reproduce a demonstrated motion from one system onto another target system with different kinematic structure which may, for example, have less degrees of freedom (DOF). We assume access to the target system's kinematic model and some approximate knowledge of the operational space of the demonstrator. The environment consists of static elements such as fixtures and equipment, known a-priori, and dynamic objects that can be added and removed online.

We define the demonstration space as \mathcal{C}_{dem} and the demonstration system as \mathcal{S}_{dem} . The desired motion is captured during the demonstration as a set of discrete states over time $\lambda_{dem} = \{t_i, x_i\}_{i=0}^T$ with $x_i \in \mathcal{C}_{dem}$. A *task model* [12] is trained on λ_{dem} with generalisation capabilities, such as temporal scaling, during motion reproduction. The trained task model generates a trajectory λ_{tar} which is used for reproduction on the target system \mathcal{S}_{tar} within its reachable space \mathcal{C}_{tar} . We assume \mathcal{C}_{dem} and \mathcal{C}_{tar} are in a *common representation space* which we choose to be $SE(3)$ since motions within this space can be interpreted by both systems. We further assume that $\mathcal{C}_{dem} \cap \mathcal{C}_{tar} \neq \emptyset$. In order to realise the reproduced motion on the target system a *motion generator* interprets λ_{tar} and generates executable controls, $\Pi : \lambda_{tar} \mapsto \pi_{tar}$, where π_{tar} is a sequence of control actions.

B. Region of Reproducible Motions

In general, LfD methods impose little to no assumptions on the structure of \mathcal{C}_{tar} . However, for \mathcal{S}_{dem} and \mathcal{S}_{tar} with differing kinematic structures this is problematic since \mathcal{C}_{tar} may not entirely cover \mathcal{C}_{dem} , that is $(\mathcal{C}_{dem} \cap \mathcal{C}_{tar}) \subset \mathcal{C}_{dem}$. Furthermore, not all λ_{dem} are reproducible on \mathcal{S}_{tar} .

Thus to improve the robustness of the reproduction process we wish to constrain \mathcal{C}_{dem} such that it is contained within \mathcal{C}_{tar} and any λ_{dem} within this constrained space is reproducible on \mathcal{S}_{tar} . More concretely, we define a reproducible motion to be a λ_{dem} such that the resulting λ_{tar} when mapped through Π results in a short, smooth path in \mathcal{C}_{tar} and is collision-free.

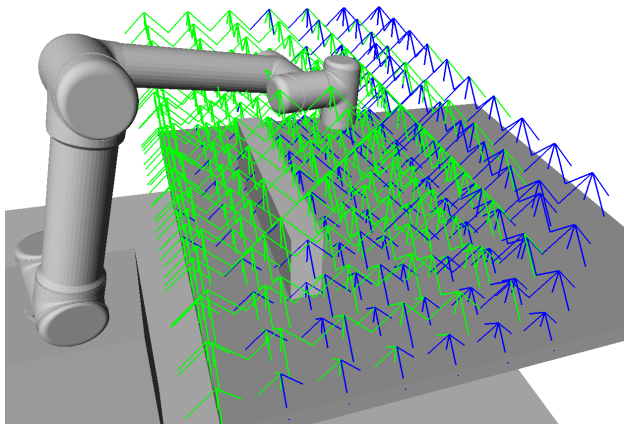
Problem 1 (Region of reproducible motions): Find a region $\mathcal{R} \subset \mathcal{C}_{dem}$ such that it is contained within \mathcal{C}_{tar} and any λ_{dem} through this region is a reproducible motion. In the case that multiple distinct regions exist, find \mathcal{R}^* that maximises coverage of the anticipated operational space.

III. GUIDED LEARNING FROM DEMONSTRATION

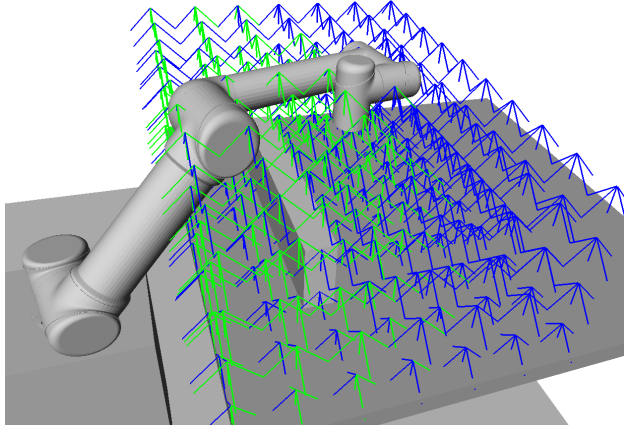
In this section we describe our approach to finding \mathcal{R}^* and then how we display this visually in order to guide the demonstrator. Additionally we outline the task model for encoding and reproducing the demonstrated motions. Finally, we explain the motion generator which is a non-local control strategy where trajectories generated from the task model are converted into appropriate controls for the target system.

A. Finding Regions of Reproducible Motions

Here we describe the method for solving Problem 1. In order to compute \mathcal{R}^* we utilise an existing robotic manipulator planning framework called Hausdorff approximation



(a) A region of reproducible motions found by HAP with 66% coverage of the task space and mapped “wrist-out” configuration.



(b) A region of reproducible motions found by HAP with 35% coverage of the task space and mapped “wrist-in” configuration.

Fig. 2: Two distinct regions of reproducible motions found by HAP for a workbench environment with wall obstacle and six-DOF UR5 robot model. (a) and (b) show a discrete task space design covering the anticipated operational space of the demonstrator where green poses indicate they are within a region of reproducible motions found by HAP and blue poses are outside the region. The former has a larger coverage of the task space due to the assigned configuration mapping.

planner (HAP) [22]. HAP operates on a user defined task space, in our case $\mathcal{C}_{tar} \subset SE(3)$, and divides it into one or more subspaces such that the path between any two points close in a particular subspace maps to a short, smooth and collision-free configuration space trajectory.

The key idea is that continuous task-space trajectories can be constructed through these subspaces and the resulting configuration space trajectory will have a bounding relation in terms of trajectory distance. Such a mapping between metric spaces is called an ϵ -Gromov-Hausdorff approximation (ϵ -GHA). Thus, any λ_{dem} that is contained within such a subspace in \mathcal{C}_{tar} will satisfy our condition for being reproducible, assuming that the generated λ_{tar} is spatially identical to λ_{dem} .

HAP generates these maps in a pre-processing step. The

procedure for generating these maps is as follows. First, HAP is given as input a discretised task space which approximately models the operational space, a robot model and an environment model. An example of each are shown in Fig. 2. An undirected graph is then constructed over the discrete task space where poses within a ball radius of each other are connected by an edge. An optimisation routine then iteratively maps unique configurations to each pose while ensuring that neighbouring poses, i.e. those connected by an edge, are within some specified bounded distance. Kinematic constraints are additionally considered, such as avoiding collision and remaining within joint limits.

The ϵ -GHA that minimises the sum of all path costs through the graph is kept. This process can be repeated multiple times with a penalty added to paths that pass through previously mapped poses in order to find multiple distinct subspaces (see Fig. 2). We choose the largest covering subspace found by HAP to be \mathcal{R}^* .

Extending on HAP, we account for dynamic objects in the scene by performing an update to \mathcal{R}^* rather than completely recomputing \mathcal{R}^* , which is computationally expensive. We perform this update by iterating over all the mapped configurations and check that they are still collision-free and if not remove them from \mathcal{R}^* .

B. Interactive Visual Guidance

Given the found \mathcal{R}^* we wish to utilise this representation to guide demonstrations such that $\lambda_{dem} \subset \mathcal{R}^*$. In order to do so, we design a graphical user interface (GUI) which displays to the user where the pose of \mathcal{S}_{dem} is within \mathcal{R}^* . Note that this GUI should be intuitive enough for a non-expert user to interpret and make effective use of without any knowledge about the underlying kinematics of the target system.

The main idea of the GUI is to block the demonstrator from moving \mathcal{S}_{dem} into regions of the task space that are outside of \mathcal{R}^* , which we will refer to as $\bar{\mathcal{R}}^*$. The representation of \mathcal{R}^* output by HAP is a discrete set of poses in $SE(3)$. One option is to display a dense grid of arrows at each of these poses, similar to Fig. 2; however, this would be too cluttered and confusing for the user to follow. Instead we compact six

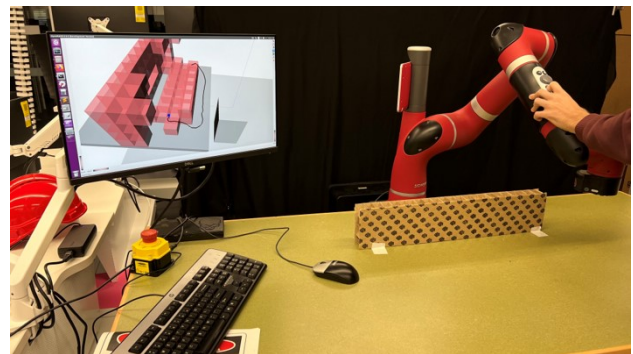


Fig. 3: Demonstration system setup for validation experiments consisting of a seven-DOF Sawyer robot, workbench environment with a wall obstacle and interactive GUI shown on a monitor.

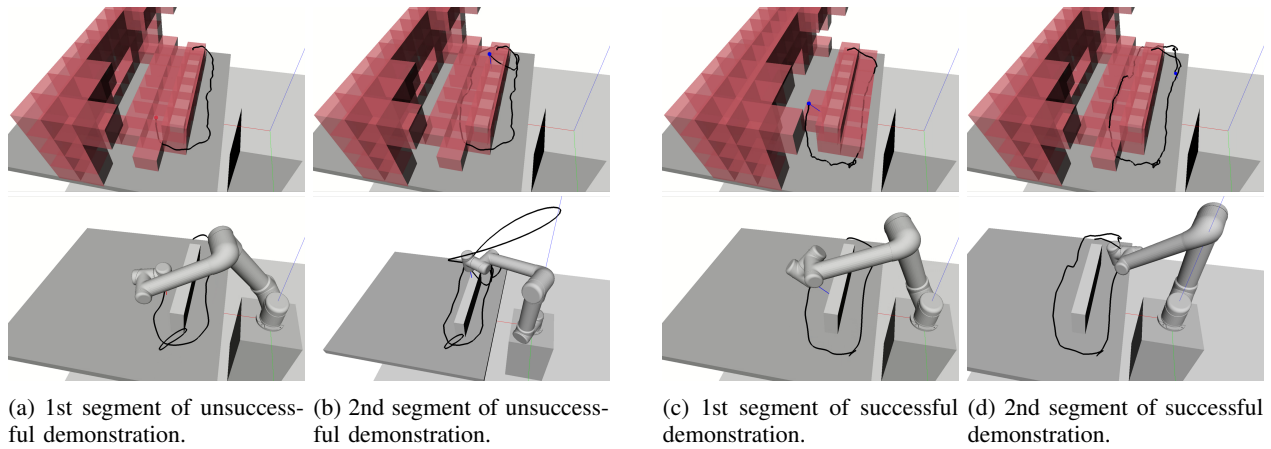


Fig. 4: Two example demonstration sequences with GUI overlaid (top row) and their corresponding reproduced motion (bottom row). The sequence in (a)-(b) attempts a downward facing orientation of the end effector on the far side of the wall and fails due to running into the red voxels in the GUI. The sequence in (c)-(d) succeeds due to pitching back the orientation of the end effector.

dimensional space - translation and orientation - into three dimensions by blocking the regions in \mathcal{R}^* dynamically when necessary. We achieve this by only displaying red voxels at poses that are within a similar orientation to the current pose of \mathcal{S}_{dem} . We compute this similarity by taking the dot product between the forward pose axes of \mathcal{S}_{dem} and \mathcal{R}^* and only display voxels corresponding to poses within a given similarity threshold. In order to track where \mathcal{S}_{dem} is in the environment, we display an arrow showing its position and forward pose axis during demonstration.

C. Task Model

As $SE(3)$ is the common representation space, λ_{dem} and λ_{tar} are a sequence of poses represented as position, \mathbf{p} , and quaternion, \mathbf{q} . Thus we choose Cartesian space Dynamic Movement Primitives (CDMPs) as our robot-independent task model. However, the general idea can be extended to other trajectory-based LfD task models.

A single demonstration of a desired motion is recorded as a trajectory $\lambda_{dem} = \{t_i, \mathbf{p}_i, \mathbf{q}_i\}_{i=0}^T$. The task model consists of two parameterised dynamical second-order systems with an additional term for external forces which shapes the trajectory to match the demonstration. The nonlinear shape of the demonstrated trajectory is modelled via weighted kernel functions, e.g., radial basis functions. For producing λ_{tar} , the dynamical systems including external wrenches with learned weights are numerically solved, resulting in a continuous model representation in $SE(3)$. In our work, we applied the revised bio-inspired formulation by Koutras et al. [23], [24]. For an in-depth explanation of DMPs see [25].

D. Motion Reproduction

For reproducing the demonstrated motion on \mathcal{S}_{tar} we utilise the ϵ -GHA mapping found by HAP to map λ_{tar} to a sequence of arm configurations, $\pi_{tar} = \{\theta_1, \theta_2, \dots, \theta_T\}$. However, poses in λ_{tar} are in continuous space where as the ϵ -GHA mapping is defined over a discrete task space.

Hence, in order to remain within \mathcal{R}^* a configuration is assigned to each $\theta \in \pi_{tar}$ that is close to the ϵ -GHA mapping. We achieve this by computing all the IK solutions for each $x \in \lambda_{tar}$ and then choosing the IK solution that minimises the euclidean distance to any of the k closest mapped configuration in \mathcal{R}^* .

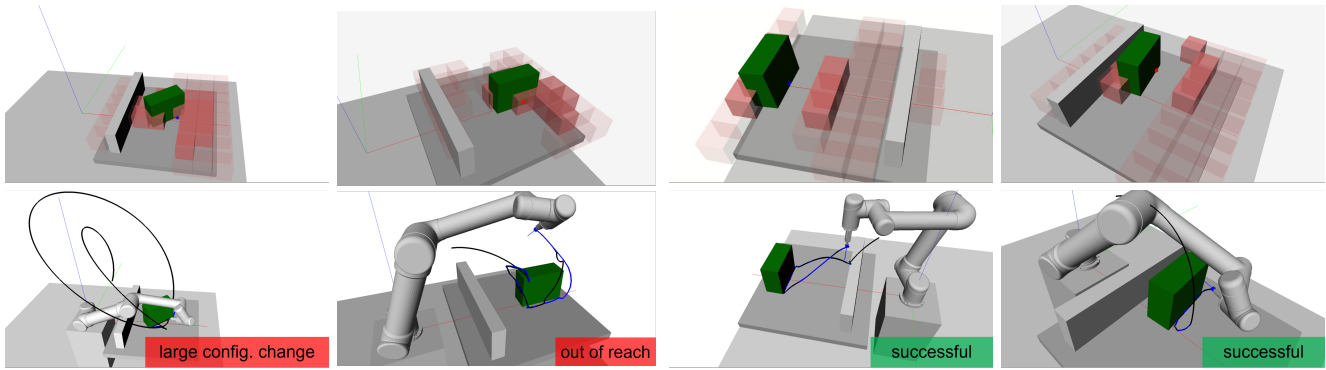
IV. EXPERIMENTS

This section presents the experimental evaluation of the proposed guided learning from demonstration approach. A validation experiment and a comprehensive user study were conducted to showcase the efficacy and usability of our guided learning from demonstration method. In both experiments the interactive GUI is displayed to the demonstrator on a monitor during demonstration. The environment and task space discretisation strategy used in the experiments is the same as shown in Fig. 2. Additional footage from the experiments is included in the attached supplementary video.

A. Validation

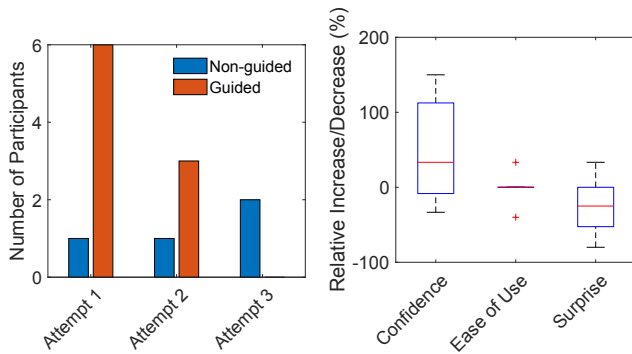
In order to validate our approach we utilise a redundant seven-DOF Sawyer arm as the demonstration system, which is moved kinaesthetically (see Fig. 3). The target system is a simulated six-DOF UR5. In this experiment we demonstrate motions around the wall obstacle and aim to show that avoiding red voxel regions displayed by the GUI leads to successful reproduction and vice versa when not.

Example demonstration sequences and their reproduced motions are shown in Fig. 4. The sequence in Figs. 4a and 4b attempts to move around the wall in an orientation normal to the workbench for the entire trajectory. However, as can be seen in the corresponding GUI snapshots this leads to moving through the red voxels. The explanation for this is that the arm must perform a large joint angle change in order to avoid colliding its wrist with the wall. This results in a poorly reproduced end effector trajectory, shown as a black line in the bottom row of Fig. 4b.



(a) Failure due to large configuration change (non-guided). (b) Failure due to unreachable weld path (non-guided). (c) Successful object placement 1 (guided). (d) Successful object placement 2 (guided).

Fig. 5: Representative user study results showing diverse success and failure cases. Reproduced motions (bottom row) and corresponding GUI output (top row) are shown. (a) and (b) are two non-guided demonstrations that failed (GUI shown here for convenience - not shown to users). (c) and (d) are two successfully demonstrated tasks using our guided method.



(a) Distribution of number of trials required for success. (b) Percentage increase/decrease in subjective metrics.

Fig. 6: Quantitative results from the user study. (a) shows the distribution of number of trials required to successfully demonstrate the weld task and (b) shows the percentage increase/ decrease in the collected subjective metrics.

In contrast, the demonstrated sequence in Figs. 4c and 4d avoids this problem by pitching back the orientation of the end effector. In this orientation the red voxels on the far side of the wall disappear and the trajectory is able to be reproduced successfully. This observed behaviour validates our method for this particular scenario and confirms our claim that reproduction can fail if the kinematics of the target system are not considered when guiding demonstrations, even in a relatively simple environment.

B. User Study

A subsequent user study was conducted with nine participants to evaluate the usability of our system. Users were tasked with carrying out a mock weld demonstration which consisted of placing an object to be welded on the workbench, shown in Fig. 1, and then performing a weld along a marked edge of the object using a hand-held 3D printed tool tracked by a VICON system. Nine users participated in the study consisting of three women and six men, five of which

had little to no experience with robot arms.

For the user study we additionally enhanced the GUI to assist users with depth perception by dynamically increasing the opacity of the voxels within some distance threshold of the weld tool position. Furthermore, red boxes above the position of the tool are temporarily removed to prevent obstructing the view of the demonstrator. An example of this is shown in Fig. 1 for a particular position and orientation of the weld tool.

Regarding evaluation against other methods, existing work that deals with transferring motion between two different systems utilise null-space control. Since the target system is non-redundant, the former is not possible.

Thus, first each user was asked to perform the task without the guidance of the interactive GUI. They were given a maximum of three attempts which involved carrying out the demonstration and then verifying that it was reproducible on a simulated UR5 in the same environment. If successful the reproduction was carried out on the real UR5 robot.

Afterwards, the user was asked to carry out the same task with the guidance of the GUI, again with three attempts. If the user was successful in the unguided attempt they were asked to find another valid placement of the weld object. The main advantage of having the GUI was that it helped users make an informed decision on where to place the weld object. Furthermore, users were able to place the object and then update the region of feasible motion before carrying out the demonstration.

Representative trials from the user study are shown in Fig. 5. Quantitative results are summarised in Fig. 6. Fig. 6a shows the distribution of the number of trials required to successfully demonstrate the weld task. Note that five out of the nine participants were unsuccessful in all three attempts for the non-guided experiment. As can be seen, all users were able to successfully demonstrate the task within two attempts with guidance, whereas a majority of users - 56% - failed to do so at all without guidance.

To assess usability we chose subjective metrics based on

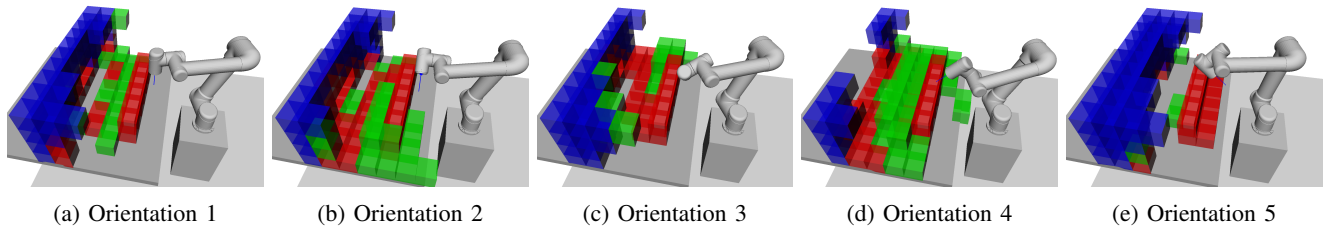


Fig. 7: Different regions of the subspace being blocked due to: All IK solutions in collision (red), requiring large change in configuration in order to reach (green) and no IK solution due being outside the arm’s reachability (blue).

a previous usability study on socially assistive robots [26]. Their study categorised their evaluation metrics into effectiveness, efficiency and satisfaction with which users were able to achieve given tasks [27]. In a similar fashion we asked users to rate on a 5-point scale how confident they were and how easy it was to carry out the task with and without the GUI. In addition, we asked how surprised they were at the reproduced trajectory. The percentage increase and decrease in each of these metrics is shown in Fig. 6b.

As can be seen, almost all participants were more confident when carrying out the task when utilising guidance from the GUI. Ease of use was comparable which is a positive result considering the addition of the GUI during demonstration. And users were less surprised by the reproduced motions when using the GUI which can be explained by the absence of large, spontaneous joint changes.

We additionally asked participants the following questions on a 5-point scale specifically about the effectiveness and design of the GUI:

- How effective was the GUI at improving your decision of where to place the weld object?
- How intuitive was the GUI in assisting with the task?
- How effective was the GUI in providing spatial awareness, i.e. avoid red voxels?
- Is the extra effort to use the GUI worth it over trial and error (without any guidance)?

On average users gave a rating of 4 and above which indicates that our GUI design was overall effective and intuitive enough for assisting with the given task. Furthermore, all participants rated the last question a 5 which reinforces the value of our method over a trial and error approach.

In terms of qualitative feedback, some participants desired more suggestive guidance. For example providing recommended object placements to the user. Other suggested improvements included receiving more information on the source of failure when entering a red voxel and the option to display the simulated robot in the GUI.

V. DISCUSSION

A. Analysing $\bar{\mathcal{R}}^*$

To explain the poses in $\bar{\mathcal{R}}^*$, color coded voxels are shown in Fig. 7 for varying end-effector orientations. The green voxels behind the wall are blocked due to requiring a large change about the wrist to avoid collision with the wall or table which would violate our region of reproducible motions

condition. The red voxels indicate regions where no IK solution is possible without environment collision. The blue voxels indicate regions outside the reachability of \mathcal{S}_{tar} .

B. Practical Considerations

We designed the discretised task space for the experiments, shown in Fig. 2, such that all poses were approximately oriented in a nominal direction, facing down into the workbench. Allowing any greater deviations from this nominal direction would result in an ambiguous guidance in the GUI. For example, it may be possible to rotate the end effector about a point in the workspace in one direction but rotating in the other direction may require a large joint angle change to avoid self-collision or joint limits. However, this is a limitation of the GUI rather than our method for finding \mathcal{R}^* . A work around is to define multiple task spaces with the required nominal directions and then carry out demonstrations independently for each of these.

Furthermore, the \mathcal{R}^* found is approximated by this discretisation. Hence if more granularity is required for more fine motor tasks one could increase the resolution. To combat the increase in computation, one could potentially employ a multi-resolution strategy.

VI. CONCLUSIONS

We presented a new guided learning from demonstration formulation for ensuring reproducibility when transferring a demonstrated motion to a system with differing kinematic structure. To address this problem we leveraged an existing robotic manipulator planner to compute regions of reproducible motions and then utilised this to provide intuitive guidance to the demonstrator via a novel interactive GUI. Results from our user study showcased the significance of being able to demonstrate reproducible motions reliably without the presence of the physical target system.

Future work includes improving the GUI design to allow for more flexible operational space definition and exploring other guidance mediums such as augmented reality and haptic feedback, which would be beneficial in more challenging cluttered environments. Presenting suggestions to the user, such as where to place objects in order to carry out a task successfully, is another important avenue for improving usability. Furthermore, HAP has the ability to find subspaces that satisfy additional target system constraints, such as minimum manipulability and joint torques, which could facilitate more interesting setups.

REFERENCES

- [1] M. R. Pedersen, L. Nalpantidis, R. S. Andersen, C. Schou, S. Bøgh, V. Krüger, and O. Madsen, “Robot skills for manufacturing: From concept to industrial deployment,” *Robotics and Computer-Integrated Manufacturing*, vol. 37, pp. 282–291, 2016.
- [2] A. G. Billard, S. Calinon, and R. Dillmann, “Springer Handbook of Robotics,” in *Springer Handbook of Robotics*, 2nd ed. Springer, Berlin, Heidelberg, 2016, ch. Learning f, pp. 1995–2014.
- [3] L. Rozo, P. Jiménez, and C. Torras, “A robot learning from demonstration framework to perform force-based manipulation tasks,” *Intelligent Service Robotics*, vol. 6, no. 1, pp. 33–51, 2013.
- [4] M. Rana, M. Mukadam, S. R. Ahmadzadeh, S. Chernova, and B. Boots, “Towards robust skill generalization: Unifying learning from demonstration and motion planning,” in *Intelligent robots and systems*, 2018.
- [5] Z. Zhu, H. Hu, and D. Gu, “Robot Performing Peg-in-Hole Operations by Learning from Human Demonstration,” in *2018 10th Computer Science and Electronic Engineering Conference, CEEC 2018 - Proceedings*. IEEE, 2019, pp. 30–35.
- [6] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Robmann, D. Martínez, G. Alenyà, C. Torras, A. Ude, B. Nemeč, A. Kramberger, F. Wörgötter, E. E. Aksoy, J. Papon, S. Haller, J. Piater, and N. Krüger, “Teaching a Robot the Semantics of Assembly Tasks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 670–692, 2018.
- [7] X. Gao, J. Ling, X. Xiao, and M. Li, “Learning Force-Relevant Skills from Human Demonstration,” *Complexity*, vol. 2019, 2019.
- [8] M. Kyrarini, M. A. Haseeb, D. Ristić-Durrant, and A. Gräser, “Robot learning of industrial assembly task via human demonstrations,” *Autonomous Robots*, vol. 43, no. 1, pp. 239–257, 2019.
- [9] N. Liu, X. Zhou, Z. Liu, H. Wang, and L. Cui, “Learning peg-in-hole assembly using Cartesian DMPs with feedback mechanism,” *Assembly Automation*, vol. 6, no. September, pp. 895–904, 2020.
- [10] H. Su, A. Mariani, S. E. Ovur, A. Menciassi, G. Ferrigno, and E. De Momi, “Toward Teaching by Demonstration for Robot-Assisted Minimally Invasive Surgery,” *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 2, pp. 484–494, 2021.
- [11] H. Ravichandar, A. S. Polydoros, S. Chernova, and A. Billard, “Recent Advances in Robot Learning from Demonstration,” *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, pp. 297–330, 2020.
- [12] A. Vakanski and F. Janabi-Sharifi, *Robot Learning by Visual Observation*. John Wiley & Sons, 2017.
- [13] R. Lioutikov, G. Neumann, G. Maeda, and J. Peters, “Probabilistic segmentation applied to an assembly task,” in *IEEE-RAS International Conference on Humanoid Robots*, vol. 2015-December. IEEE, 2015, pp. 533–540.
- [14] C. Lauretti, F. Cordella, and L. Zollo, “A Hybrid Joint/Cartesian DMP-Based Approach for Obstacle Avoidance of Anthropomorphic Assistive Robots,” *International Journal of Social Robotics*, vol. 11, no. 5, pp. 783–796, 2019. [Online]. Available: <https://doi.org/10.1007/s12369-019-00597-w>
- [15] N. Jaquier, L. Rozo, D. G. Caldwell, and S. Calinon, “Geometry-aware manipulability learning, tracking, and transfer,” *International Journal of Robotics Research*, vol. 40, no. 2-3, pp. 624–650, 2021.
- [16] F. J. Abu-Dakka and V. Kyrki, “Geometry-aware dynamic movement primitives,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 4421–4426.
- [17] K. M. Lynch and F. C. Park, *Modern robotics*. Cambridge University Press, 2017.
- [18] Y. Mollard, T. Munzer, A. Baisero, M. Toussaint, and M. Lopes, “Robot programming from demonstration, feedback and transfer,” in *IEEE International Conference on Intelligent Robots and Systems*, vol. 2015-December, 2015, pp. 1825–1831.
- [19] M. B. Luebbbers, C. Brooks, M. J. Kim, D. Szafir, and B. Hayes, “Augmented reality interface for constrained learning from demonstration,” in *Proceedings of the 2nd International Workshop on Virtual, Augmented and Mixed Reality for HRI (VAM-HRI)*, 2019.
- [20] M. Diehl, A. Plopski, H. Kato, and K. Ramirez-Amaro, “Augmented Reality interface to verify Robot Learning,” *29th IEEE International Conference on Robot and Human Interactive Communication, RO-MAN 2020*, pp. 378–383, 2020.
- [21] M. B. Luebbbers, C. Brooks, C. L. Mueller, D. Szafir, and B. Hayes, “ARC-LfD: Using Augmented Reality for Interactive Long-Term Robot Skill Maintenance via Constrained Learning from Demonstration,” *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2021-May, no. Icara, pp. 909–916, 2021.
- [22] F. Sukkar, J. Wakulicz, K. M. B. Lee, and R. Fitch, “Motion planning in task space with gromov-hausdorff approximations,” *arXiv preprint arXiv:2209.04800*, 2022.
- [23] L. Koutras and Z. Doulgeri, “A correct formulation for the orientation dynamic movement primitives for robot control in the cartesian space,” in *Conference on robot learning*. PMLR, 2020, pp. 293–302.
- [24] —, “A novel dmp formulation for global and frame independent spatial scaling in the task space,” in *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*. IEEE, 2020, pp. 727–732.
- [25] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, “Dynamic Movement Primitives in Robotics: A Tutorial Survey,” *arXiv preprint arXiv:2102.03861*, no. February, 2021.
- [26] R. A. Olde Keizer, L. van Velsen, M. Moncharmont, B. Riche, N. Ammour, S. Del Signore, G. Zia, H. Hermens, and A. N’Dja, “Using socially assistive robots for monitoring and preventing frailty among older adults: a study on usability and user experience challenges,” *Health and technology*, vol. 9, no. 4, pp. 595–605, 2019.
- [27] I. Iso, “9241-11: 2018 ergonomics of human-system interaction—part 11: Usability: Definitions and concepts,” *International Organization for Standardization*. <https://www.iso.org/obp/ui/#iso:std:iso:9241:11:2018>.