2022 31st IEEE International Conference on Robot and
Human Interactive Communication (RO-MAN)
August 29 - September 2, 2022. Naples, Italy.

Towards Transferring Human Preferences from Canonical to Actual Assembly Tasks

Heramb Nemlekar, Runyu Guan, Guanyang Luo, Satyandra K. Gupta and Stefanos Nikolaidis

Abstract— To assist human users according to their individual preference in assembly tasks, robots typically require user demonstrations in the given task. However, providing demonstrations in actual assembly tasks can be tedious and time-consuming. Our thesis is that we can learn the preference of users in actual assembly tasks from their demonstrations in a representative canonical task. Inspired by prior work in economy of human movement, we propose to represent user preferences as a linear reward function over abstract task-agnostic features, such as movement and physical and mental effort required by the user. For each user, we learn the weights of the reward function from their demonstrations in a canonical task and use the learned weights to anticipate their actions in the actual assembly task; without any user demonstrations in the actual task. We evaluate our proposed method in a model-airplane assembly study and show that preferences can be effectively transferred from canonical to actual assembly tasks, enabling robots to anticipate user actions.

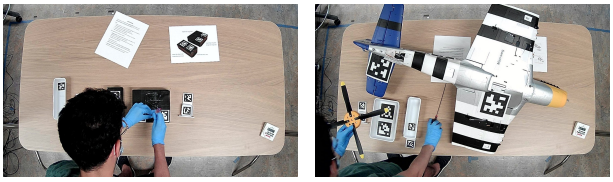
I. Introduction

The advent of human-safe robots has enabled deployment of human-robot teams on manual assembly tasks where robots can carry out supporting actions, e.g., bringing parts and tools or clearing out the assembly area, while humans focus on the high value actions, e.g., using the tool to assemble the parts. To effectively assist humans, robots need to predict actions that are likely to be performed by humans [1]–[4]. For example, if a human is expected to perform assembly of a part that requires a screwdriver, the robot can proactively fetch a screwdriver from the tool shelf and deliver it to the human to improve the task efficiency.

However, in many assembly tasks, while a lot of aspects of the task are prespecified and constrained, workers still have their individualized preferences on how to execute a task [5]–[7]. For example, one worker may prefer to do all the difficult actions first and easy actions at the end, while another worker may prefer the opposite. Thus, robotic assistants would need to adapt to the individualized preferences of a human operator, e.g., deliver parts in the worker's preferred sequence, to execute the task efficiently and fluently [8].

Prior work in learning user preferences for task execution relies on demonstrations (e.g., state-action pairs) of the user in the actual task to learn a policy [9], [10] or an underlying reward function [11]–[15] that captures the user's preferred sequence of actions. However, providing demonstrations for each assembly task that a given user must perform is especially tedious and time-consuming.

Heramb Nemlekar, Runyu Guan, Guanyang Luo, Satyandra K. Gupta and Stefanos Nikolaidis are with the University of Southern California, USA. fnemlekar, guanyu, luog, guptask, nikolaidg@usc.edu



(a) Canonical assembly task (b) Actual assembly task

Fig. 1: Example of a user that prefers to perform high-effort actions at the end of assembly tasks. (a) Last action of the user in the canonical task is to screw the long bolt which requires the most physical effort. (b) Second last action of user in the actual task is to screw the intricate propeller which also requires the most physical effort (as rated by the user).

Instead, to reduce the cost of obtaining demonstrations, we posit that we can transfer the user's preference from a representative canonical task (source task) to a new yet related assembly task (target task). We wish our canonical task to be short so that users can easily provide demonstrations, but also expressive enough so that users can demonstrate preferences that enable anticipation of their actions in the actual task. In this work, we empirically design a canonical task for a given assembly task, and focus on investigating whether human preferences can be effectively transferred from canonical to actual assembly tasks. Our problem is especially challenging and distinct from prior work in transfer learning [16]–[18], since we focus on transferring preferences of real users – as opposed to agent policies – across tasks.

Inspired from prior work in economy of human movement [19]–[22] and task ordering [23], our key insight is that user preference across different related assembly tasks can be represented with a common set of abstract, task-agnostic features, such as the movement and physical and mental effort required by the user to perform each action in the assembly. Thus, we model the user's internal reward function as linear in the task-agnostic features, where the feature weights represent the user's preference.

For a given user, we hypothesize that their preferences over these features will be similar in both the canonical and actual tasks. For example, if a worker prefers to perform the high-effort actions at the end of the canonical task, they will likely prefer the same in the actual task (see Fig. 1). We use the maximum-entropy inverse reinforcement learning (IRL) [11] approach to learn the weights for each user from their demonstration in a canonical task and then use the same weights to model their rewards and compute their policy in the actual task.

Our main contribution is to show how preferences of real users can be transferred from a canonical to an actual assembly task. We validate our proposed method in a user

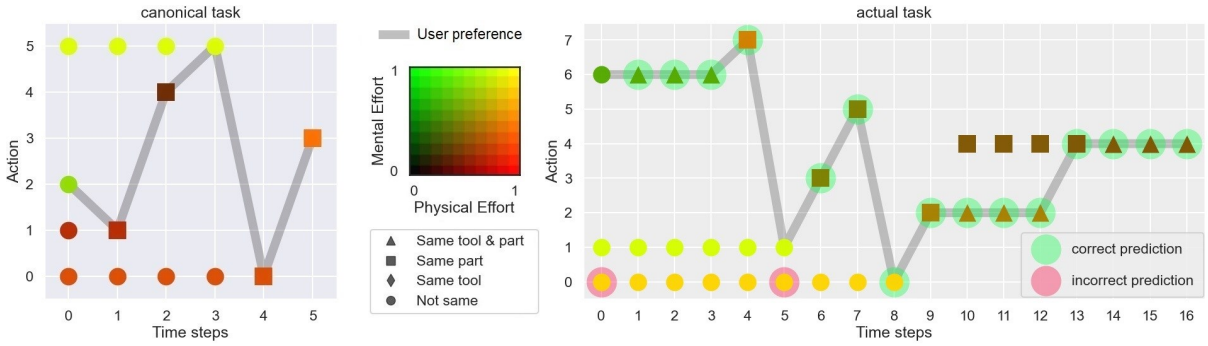


Fig. 2: Example of a user that prefers to keep working on the same part (2), and perform actions with high physical effort (red) at the end of the task. In the above plots, the x-axis represents the steps (progress) in the task, and y-axis represents all the different actions in the task. At each time step, the actions that can be executed are marked with a shape that indicates whether that action requires the same tool and part as the previous action (refer to legend). The color of the shape indicates the physical and mental effort required to perform that action (refer to color map). In the canonical task, the user performs action 2 at the start of the task (time step 0) because it requires less physical effort (least red) compared to the other choices (actions 0, 1, and 5). At the next time step, the user performs action 1 because it requires the same part as the previous action 2. In the remaining steps, the user performs actions on the same part as before, and if no actions use the same part, perform the least physical effort action. Interestingly, the user follows the same preference in the actual task by performing the least physical effort (least red) action 6 at the start. The predictions made by our proposed approach at each time step are shown with a larger green (or red) circle. For further discussion, refer to Section VI.

study where we anticipate user actions in a model-airplane assembly task based on their demonstration in a canonical task. Our results show that transferring user preferences from a canonical task can enable accurate anticipation of the user actions in the actual task by modeling user preferences in both the tasks over the same set of task-agnostic features.

II. Related Work

Similar to prior work [5], [12], [24], we consider that the preferences of a user are captured by their internal reward function. Therefore, our goal is to transfer the user's reward function from a canonical task to an actual assembly task.

A. Transferring human preference from source to target task

The problem of transferring the preferences of real users is distinct from the problem of transfer learning [17], [18], [25] that focuses on using the policies of robotic or simulated agents in a source task as priors to speed-up learning in the target task. Instead, our work focuses on learning the user's preference as a function of abstract features for anticipating their actions in a different assembly task. Moreover, unlike prior work in transfer learning [16], [26]–[28], we attempt to transfer user preferences without access to the user's rewards or demonstrations in the target task.

The prior work most similar to our problem transfers the preference of a simulated human from a (source) block stacking task to a target task with an extra red block [6] by modeling artificial preferences based on the color of the blocks. However, to our knowledge, no prior work has studied transferring preferences of real users from one assembly task to another.

B. Factors affecting human preferences in physical tasks

User preferences for sequencing the actions (or sub-tasks) in assembly and manufacturing tasks can be affected by several factors that are task-specific. For example, in a part stacking assembly [29], user preferences depended on the size and color of the parts, and were modelled as a linear reward function of the two features. While in a Lego model assembly [30], user preferences for task allocation depended on the type of sub-task - fetching or building.

In order to transfer user preferences from one assembly task to another, we want to model the preferences based on features that are task-agnostic. Previous studies [19], [21], [22] have shown that users prefer to minimize movements during task execution. We expect the same for users in an assembly task, i.e., users would prefer to minimize movements required to perform the assembly. Similarly, users may also look to minimize their effort spent in the task. For example, in an object pick-up task [23] some users preferred to pick up the closest object first because it reduced their cognitive effort, even if it increased their physical effort in the long run. In a study on human jump landing [20], users optimized a combination of active and passive efforts, while also accounting for other factors like safety.

In this work, we presume that users will prefer to minimize some combination of the movement cost and the physical and mental efforts, with different users having different combinations (i.e. preferences).

III. Methodology

We want to transfer user preferences from a canonical task C to an actual assembly task X for anticipating user actions. We model each task as a Markov Decision Process (MDP) defined by the tuple $(S; A; T; R)$, where S is the set of states in the assembly, A is the set of actions that must be performed to complete the assembly, $T(s_{t+1} | s_t; a_t)$ is the probability of transitioning to state $s_{t+1} \in S$ from state $s_t \in S$ by taking action $a_t \in A$, and $R(s_{t+1})$ is the reward received by the user in s_{t+1}.

We assume that S_X, A_X and T_X are known for the actual assembly task X. While T_X models the ordering constraints, because each worker can have their own preferred sequence $\chi = [a_1; \ldots; a_t; \ldots; a_N]$ of performing the actions $a_t \in A_X$, R_X will be specific to each worker. Therefore, to anticipate the actions of a user i in the actual assembly task, we must learn their individual reward function $R_{X;i}$.

Goal. We want to learn the user's reward function R_X from demonstrations $_C$ provided by the user in a canonical task C (which has its own set of S_C, A_C and T_C).

Intuition. Our key insight is that preferences of users in actual assembly tasks can be represented with abstract features from a task-agnostic feature space . Given : $fS_C; S_X g$! , we can map any state in the canonical and actual tasks to a d-dimensional feature vector in .

As the rewards received by a user depend on the state of the task, using , we can model the reward function of a given user i as a function of the task-agnostic features.

$$R_{X;i}(s) = f_{X;i}((s)) \quad 8\, s\, 2\, S_X$$

The function $f_{X;i}$ is specific to the user i, and captures their individual preference. Our hypothesis is that users will have similar preferences over the abstract features in both the canonical and actual tasks, i.e., $f_{X;i}$ ' $f_{C;i}$, given that: (i) the feature space fully captures the preferences of all users, and (ii) the canonical task C is expressive enough to capture preferences over a diverse range of feature values.

Knowing , we can learn each user's $f_{C;i}$ from their demonstrations $_C$ in the canonical task, and use the same function, i.e., $f_{X;i} = f_{C;i}$, to calculate the user's rewards $R_{X;i}$ for states in the actual assembly task.

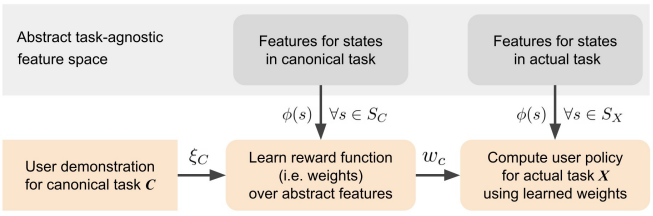


Fig. 3: Flowchart of our proposed method for transferring preferences

Approach. Following prior work [11], we model the reward function $R_{X;i}(s) = w^T_{X;i}$ (s) as linear in the features (s). Here, w is a d-dimensional weight vector where each weight in the vector represents the user's preference for a particular dimension of the feature space.

Given a demonstration sequence $_C = [a_1; : : : ; a_M]$ of actions a 2 A_C, we use maximum-entropy IRL [11] to learn the weights w_C for the user. In this approach, we iteratively update the weights to maximize entropy (as there can be multiple solutions) such that our learner visits the features in the canonical task with the same frequency as observed in $_C$. We choose the maximum entropy approach because we wish the learned weights to explain the demonstrations without adding any additional constraints to the resulting policy.

Based on our key insight, we assume that user i would have the same weights ($w_{X;i} = w_{C;i}$) for the features in the actual task. Thus, we can calculate the transferred rewards $\tilde{R}_X$ by using the weights w_C:

$$\tilde{R}_{X;i}(s) = w^T_{C;i}(s) \qquad 8s\, 2\, S_X \qquad (1)$$

Fig. 3 summarizes our proposed approach for transferring preferences from canonical to actual tasks. We conduct a user study to evaluate whether $\tilde{R}_X$ can be used to effectively anticipate user actions in the actual task.

To anticipate user actions, we assume that the user will try to maximize their long-term reward in the actual assembly task. Thus, we use the learned $\tilde{R}_X$ to perform value iteration [31] for all states s 2 S_X in the actual task, and select the action with the highest value as our prediction $\hat{a}_t$ in a given state. In our study, we calculate the value of taking an action in a given state without discounting the future rewards, since users plan for the entire assembly before they demonstrate their preference.[1]

IV. Task-Agnostic Feature Space

Inspired from prior work in economy of human movement [19], [21], [22], we presume that users would try to minimize their movement throughout the task. Because the set of actions is fixed in our assembly tasks, users can minimize their movement when they switch from one action to the next. For example, a worker may prefer to consecutively perform all the actions on one side of the assembly to avoid having to shift sides. In our user study, movement for switching between actions is performed when the user changes the tool or the part they are working on. Thus, we consider the following features to capture user preferences for minimizing movement:

TABLE I: Movement-Based Features

Feature	Weight	Value	Preference
P	w_P	f0; 1g	Keep same part
T	w_T	f0; 1g	Keep same tool

Here, w_P and w_T are the weights for the respective movement-based features in the user's reward function. For a state s_{t+1}, $_P(s_{t+1}) = 1$ if the latest action a_t uses the same part as the previous action $a_{t\ 1}$. Similarly, $_T(s_{t+1}) = 1$ when a_t requires the same tool as $a_{t\ 1}$. Because we want our features to be state dependent, we augment the current state with the previous two actions: s_{t+1} $[s_{t+1}; a_t; a_{t\ 1}]$. At the start of the task, the previous two actions in the start state are set to None.

Previous work [20] also states that users may choose actions trying to minimize the effort they spend in the task. For each action, we define the effort " as a weighted combination of the nominal physical ($"_p$) and mental ($"_m$) efforts required to perform that action.

$$"(a) = w_p"_p(a) + w_m"_m(a)$$

Here, the weights w_p and w_m are specific to the user and depend on their individual preference towards minimizing their physical and mental effort. For example, users may prefer to perform specific actions first to reduce cognitive load, even if it results in requiring more time and physical effort [23]. Specifically in our pilot studies, we observed that some users preferred performing the high-effort (mental or physical) actions at the start of the assembly, while others preferred to start with low-effort actions.

[1] For very long assembly tasks users might minimize movement or effort over a shorter horizon, in which case we would adapt the time horizon accordingly.

If a worker prefers to perform high-effort actions at the end, they must be receiving a higher internal reward for performing the high-effort action at the end instead of at the start. We consider this as the perceived effort ϕ_b of an action based on the user's preference to backload the high-effort actions. To model the time-dependency, we introduce a variable - phase $\quad : s \to [0;1]$ which represents the percentage of the task that has been completed. We use phase instead of the actual time steps for generality, since the actual task is typically much longer than the canonical task. Using this feature, we model the perceived effort as a linear function of the phase: $\phi_b(a; \quad) = \quad \phi(a)$.

We can see that at the start, i.e., $\quad' 0$, the perceived effort will be very small compared to at the end ($\quad' 1$). Thus to maximize the accumulated reward, users will backload the high-effort actions. We can also have the opposite scenario, where a workers prefers to perform the high effort actions at the start. In this case, the reward that the user receives for performing the high-effort actions at the start must be higher than at the end. Hence, we model the perceived effort for frontloading high-effort action as: $\phi_f(a; \quad) = (1 \quad) \phi(a)$.

As our state contains the information of the latest action, we can model the features for frontloading and backloading actions with high physical effort as:

$$f;p(s) = \quad f(s)\phi p(s) \tag{2}$$

$$b;p(s) = \quad b(s)\phi p(s) \tag{3}$$

Where, $f = 1 \quad$ and $b = \quad$. Similarly, we can calculate the features for sequencing actions based on their mental effort to obtain the following list of features:

TABLE II: Effort-Based Features

Feature	Weight	Equation	Preference
$f;p$	$w_{f;p}$	$f \phi p$	Frontloading of high ϕ_p actions
$f;m$	$w_{f;m}$	$\phi m f$	Frontloading of high ϕ_m actions
$b;p$	$w_{b;p}$	$\phi p b$	Backloading of high ϕ_p actions
$b;m$	$w_{b;m}$	$\phi m b$	Backloading of high ϕ_m actions

We assume that the effort values for actions in both the canonical and actual assembly tasks can be measured prior to task execution e.g., through user surveys. Therefore, we use the six features from Tables I, II to create our feature function (s), that maps each state s in the canonical and actual tasks to a 6-dimensional feature space.

V. User Study

We want to show that human preferences learned from demonstrations in a canonical task can be used to anticipate their actions in an actual assembly task. Thus, we conduct a user study where participants demonstrate their preferred sequence of actions in a canonical task and an actual model-airplane assembly task. We use the demonstrations in the actual task as ground truth to measure the accuracy of anticipating actions based on the weights (i.e. preference) learned from demonstrations in the canonical task.

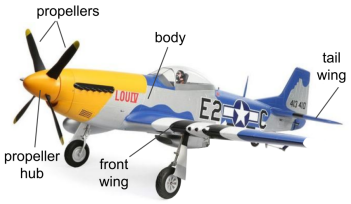


Fig. 4: Actual assembly task: Airplane model and task actions.

A. Actual assembly task

We choose an RC model-airplane assembly (see Fig. 4) as our actual task. The actions in this assembly task can be sequenced in multiple ways, with few constraints, e.g., action 2 must precede action 4 since a bolt must be inserted before it is screwed. The actions can also require different physical and mental efforts. For example, the user shown in Fig. 2 rates the action 0 of inserting the (large) main wing as having higher physical effort than the action 6 of screwing a (small) propeller. As some actions need to be repeated, e.g., action 6 must be performed for each of the 4 propellers, the length of a demonstration in the actual task is 17 time steps. On average, users required 8:81 minutes to complete this task.

B. Canonical assembly task

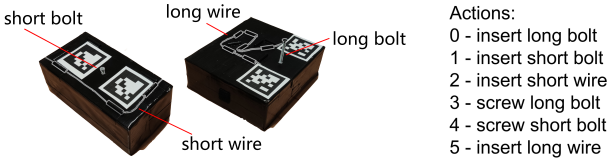


Fig. 5: Canonical assembly task: (Left) Model, (Right) Task actions.

A key challenge in designing the canonical assembly task is to create a subset of diverse states and actions, such that we can capture the user preference over a wide range of feature values, while keeping the task significantly shorter than the actual assembly task.

To capture user preferences for consecutively performing actions related to the same part, we design our canonical assembly with two different parts (see Fig. 5), each of which is required by at least two different actions. Next, to capture user preferences for consecutively performing actions that need the same tool, we design a pair of actions that require the same tool, i.e., a screwdriver. Finally, to capture the user preference for sequencing actions based on their physical and mental effort, we design an action for each combination of high and low values for physical and mental effort:

TABLE III: Actions with distinct physical and mental efforts.

	Low ϕ_p	High ϕ_p
Low ϕ_m	Action 4	Action 3
High ϕ_m	Action 2	Action 5

To induce the corresponding physical and mental efforts, we design the actions as follows:

Action 4, where the user screws a short bolt with a screwdriver. Because the bolt is short, it requires less physical effort to screw it in. Also, since we expect our participants to be familiar with using a screwdriver, we expect this action to require less mental effort.

Action 3, where the user screws a long bolt using a screwdriver. Because the bolt is long, it requires high physical effort to completely screw it in.

Action 2, where the user inserts a wire through three small spacers. Because it requires more focus than the other actions, we expect it to require high mental effort.

Action 5, where the user inserts a long wire through six spacers. Since there are more spacers, it requires more physical and mental effort to maneuver the long wire.

We conducted pilot studies to fine-tune the design of the canonical task and verified that participants perceived the physical and mental efforts of the actions as intended. Our final canonical task has 6 time steps, and on average, users required only 3:83 minutes to complete the task. In future, we wish to formalize the process of designing such a canonical task for a given actual assembly task.

C. Study protocol

We recruited 19 (M = 12, F = 7) participants from the graduate student population at the University of Southern California (USC) and compensated each user with 20 USD.

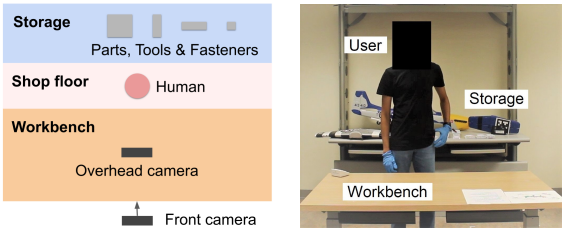


Fig. 6: (Left) Experimental setup, and (Right) setup at start of actual task

1) Experimental Setup: We divide the space into: (i) a storage area where the parts and tools and fasteners are initially placed, (ii) a workbench upon which the user performs all the actions required to complete the assembly, and (iii) the shop floor where the user can stand and move (see Fig. 6). April Tags [32] are used to track the parts during the assembly with the help of an overhead camera.

2) Study Procedure: We asked each user to perform both the canonical and the actual assembly task. We counterbalance the order of the tasks to guard against any sequencing effect. For each task, we have a (i) training round - where users learn the assembly task, and an (ii) execution round - where users plan, demonstrate, and explain their preference.

In the training round, we provide each user with a labeled image of the assembled model, as shown in Fig. 4 and 5, and we describe all the parts and actions in the assembly. We show how to perform each action in a random order and provide no instruction about the sequence of actions. We also allow users to try each action once for practice, after which they fill in a post-training questionnaire to rate the physical and mental effort that they required for performing each action. We obtain the user's values for $"_p$ and $"_m$ for each action from these ratings.

In the execution round, we first give users 5 minutes to plan their preferred sequence of actions for assembling the model as fast as possible. Finally, they demonstrate their preferred sequence and then fill in a post-execution questionnaire to report and explain the features that informed their preference in that task.

Note: To avoid influencing the users' preference, we do not inform them that their preference in one task will be used to infer their preference in another and we label the tasks as A and B (not canonical and actual). We also do not tell the users to consider effort or movement while planning their preferred sequence. We simply ask them to demonstrate their preferred sequence of actions in each assembly task.

VI. Experimental Evaluation

We wish to show that user preferences transferred from the canonical task can be used for action anticipation in the actual assembly task. Our hypothesis is that the accuracy of predicting the users' next action in the actual task based on weights learned in the canonical task would be higher than: (i) randomly picking the next action (H1) and (ii) randomly setting the weights for the features (H2).

We calculate the accuracy of anticipating the users' actions by comparing the action a_t taken by each user in the actual task with the action $\hat{a}_t$ predicted by our approach at each time step t. The accuracy is 1 when $a_t = \hat{a}_t$, and 0 otherwise.

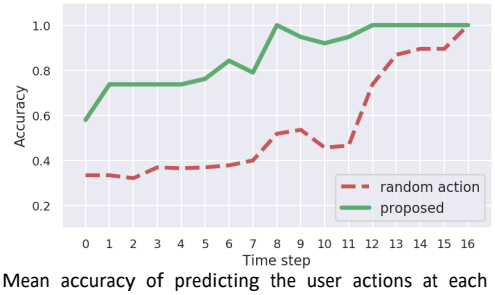


Fig. 7: Mean accuracy of predicting the user actions at each time step (averaged over all users) in the actual assembly task.

For baseline (i), we randomly select an action from the remaining actions that can be executed at each time step. For each user, we run the baseline for 100 trials and compute the average accuracy at each time step. We then compare the mean accuracy over all time steps of the proposed method and to randomly selecting actions. A paired t-test showed a statistically significant difference (t(18) = 13:82, p < 0:001) between the mean accuracy for our proposed approach (M = 0:866, S E = 0:032) and the mean accuracy for random actions (M = 0:543, S E = 0:057). This supports H1.

In baseline (ii), we calculate $\tilde{R}_X$ by uniformly sampling random weights for the actual task features and predict actions based on the computed rewards as in the proposed method. Similar to (i), we run 100 trials and compute the average accuracy. A paired t-test showed a statistically significant difference (t(18) = 2:93, p = 0:008) between the mean accuracy for our proposed approach and the mean accuracy for random weights (M = 0:828, S E = 0:042). This supports H2.

Fig. 7 shows the mean accuracy of predicting the action at each time step. We can see that the prediction accuracy increases as we reach the final time step, as there are fewer

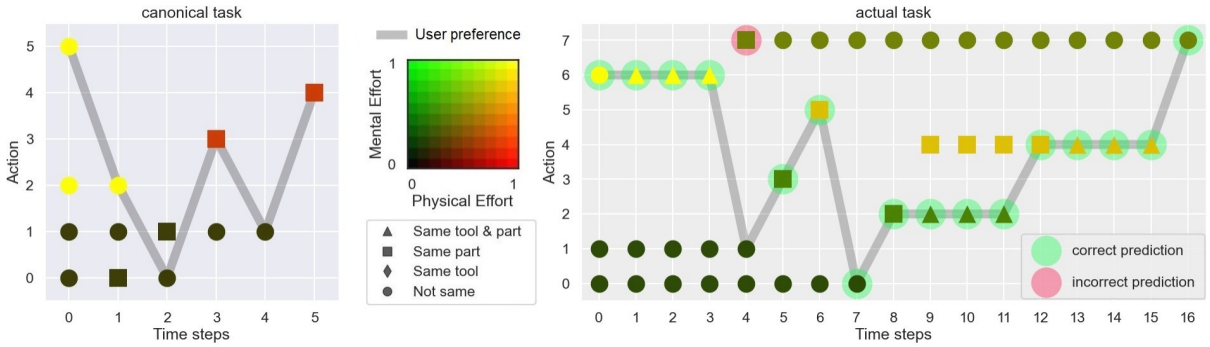


Fig. 8: Example of a user that prefers to perform actions with the high mental effort at the start of the task. In the canonical task, the user performs actions 5 and 2 that have high mental and physical effort (yellow) at the start (time steps 1 and 2). While the user leaves actions 3 and 4 that have low mental effort and high physical effort (red) for the end (time steps 3 and 5). Thus, from the canonical task demonstration, we learn that the user prefers to frontload actions with high "$_m$. In the actual task, the user starts with action 6 that has the highest mental effort. Based on the canonical task, our proposed method correctly predicts (green circle) that the user will perform action 6 at time step 0. Similarly, at time step 5 the user performs action 3, which is also what we predict, since it has higher mental effort than actions 0 and 7.

actions to choose at the end. The accuracy for random actions (and random weights) is 0:33 at the time step 0 as there are 3 actions that can be performed at the start. The accuracy for our proposed method is significantly higher at the start as we correctly anticipate the first action for 11 out of 19 users based on their transferred weights.

VII. Discussion

We consider two user examples to demonstrate how preferences transfer from our canonical to actual task.

A. Learning user preferences in the canonical task

Consider the demonstrations shown in Fig. 2, where the user prefers to pick the actions with the same part as the previous action whenever possible (time steps 1, 2 and 4). In the other steps, the user picks the action with the least physical effort (time steps 0 and 3). Accordingly, the weights we learn from the canonical task demonstration are higher for the feature of part similarity ($_P$), followed by the feature of backloading high physical effort actions ($_{b;p}$).

On the other hand, consider the user in Fig. 8 who prefers to perform actions with high mental effort (actions 5 and 2) at the start of the task (time steps 1 and 2). Based on their demonstration, we learn a high weight for the feature of frontloading high mental effort actions ($_{f;m}$).

We note that none of the users had the exact same weights w_C even if some users had the same demonstration sequence $_C$, since they gave different ratings for the physical and mental effort of the canonical task actions.

B. Transferring learned preferences to the actual task

For many users, weights learned for the task-agnostic features in the canonical task enable accurate anticipation of their actions in the actual task. For the user in Fig. 2, we learn high weights for minimizing part change ($_P$) and backloading high-effort actions ($_{b;p}$). Using these weights to calculate rewards in the actual task, we can accurately anticipate the actions at 15 out of the 17 time steps. For example, we correctly anticipate action 7 at time step 4 as it uses the same part as the action 6 at the previous time step. Similarly, for the user in Fig. 8, we are able to transfer their preference for frontloading actions with high mental effort. For example, at time step 0, we accurately predict the user's action as 6, since it has the highest mental effort.

However, in few cases, preferences (weights) over a specific feature did not transfer to the actual task due to human variability. For example, we incorrectly predict at time steps 0 and 5 for the user in Fig. 2 since we also learn a high weight for backloading actions with high mental effort ($_{b;m}$) in the canonical task. This is because the last three actions in the user's demonstration have a higher mental effort than the first three actions. Therefore, in the actual task, we predict action 0 at time step 0 even if it has high physical effort (low immediate reward), since it would allow the user to perform subsequent low mental effort actions (not shown in the figure) at the start of the task. However, since the user's true preference is to only backload the high physical effort actions, the user performs action 6 instead. Thus, we see that while the user preference for backloading actions with high physical effort transfers to the actual task, their preference for backloading actions with high mental effort does not.

Finally, we discuss a limitation where the user preference in the actual task was affected by a new feature that wasn't modeled in the canonical task. For example, in Fig. 8, we expect the user to perform action 7 at time step 4 based on their preference for frontloading actions with high mental effort. However, we see that the user instead performs action 1 which has lower mental effort and leaves action 7 for the end. Based on open-ended responses provided by users at the end of the study, we suspect that the user preferred to leave the action of screwing the propeller hub (action 7) for the end as they thought that the propellers might break (when they perform the remaining actions) if they were attached at the start. In such cases, we would like the robotic assistant to identify the states where the new feature affects user actions and query the user to actively learn the weights for the new feature. We plan to explore this in future work.

Overall, we see that preferences can be transferred when the features that determine the user's preferred sequence in the canonical task overlap with the features considered by the user to determine their preference in the actual task.

VIII. Conclusion

Our work demonstrates the potential of anticipating user actions in actual assembly tasks based on preferences learned over task-agnostic features in abstract, shorter canonical tasks. Our results show that predictions in the actual assembly task based on transferred preferences are significantly better than the predictions for random actions and random weights. Moreover by learning from user demonstrations in a shorter canonical task we can reduce the time and human effort required to obtain demonstrations in the actual assembly tasks.

In future, we want to evaluate the benefit of transferring human preferences from canonical to actual assembly tasks by having a robotic assistant perform the anticipated actions. We would also like to consider longer tasks, where users plan for only a portion of the task and may change their preference over time. While the current setup assumes full observability of the workspace by both the robot and the user, future work will consider sensor placement [33] and user viewpoint [34] in robot decision making. Finally, automatically inferring the effort for different actions, instead of using questionnaire responses, is an important area for future work.

Acknowledgements

This work was partially supported by the National Science Foundation NRI (# 2024936) and the Alpha Foundation (# AFC820-68).

References

[1] G. Hoffman and C. Breazeal, "Effects of anticipatory action on human-robot teamwork efficiency, fluency, and perception of team," in Proceedings of the ACM/IEEE international conference on Human-robot interaction, 2007, pp. 1–8.

[2] P. A. Lasota and J. A. Shah, "Analyzing the effects of human-aware motion planning on close-proximity human–robot collaboration," Human factors, vol. 57, no. 1, pp. 21–33, 2015.

[3] C.-M. Huang and B. Mutlu, "Anticipatory robot control for efficient human-robot collaboration," in 2016 11th ACM/IEEE international conference on human-robot interaction (HRI), 2016, pp. 83–90.

[4] A. M. Zanchettin, A. Casalino, L. Piroddi, and P. Rocco, "Prediction of human activity patterns for human–robot collaborative assembly tasks," IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 3934–3942, 2018.

[5] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in 2015 HRI. IEEE, 2015.

[6] T. Munzer, M. Toussaint, and M. Lopes, "Preference learning on the execution of collaborative human-robot tasks," in 2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017, pp. 879–885.

[7] H. Nemlekar, J. Modi, S. K. Gupta, and S. Nikolaidis, "Two-stage clustering of human preferences for action prediction in assembly tasks," in 2021 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2021.

[8] E. C. Grigore, A. Roncone, O. Mangin, and B. Scassellati, "Preference-based assistance prediction for human-robot collaboration tasks," in 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 4441–4448.

[9] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," Robotics and autonomous systems, vol. 57, no. 5, pp. 469–483, 2009.

[10] 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, 2020.

[11] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning." in Aaai, vol. 8. Chicago, IL, USA, 2008, pp. 1433–1438.

[12] M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions by integrating human demonstrations and preferences," 2019.

[13] E. Bıyık, D. P. Losey, M. Palan, N. C. Landolfi, G. Shevchuk, and D. Sadigh, "Learning reward functions from diverse sources of human feedback: Optimally integrating demonstrations and preferences," arXiv preprint arXiv:2006.14091, 2020.

[14] A. G. Puranic, J. V. Deshmukh, and S. Nikolaidis, "Learning from demonstrations using signal temporal logic," arXiv preprint arXiv:2102.07730, 2021.

[15] ——, "Learning from demonstrations using signal temporal logic in stochastic and continuous domains," IEEE Robotics and Automation Letters, vol. 6, no. 4, pp. 6250–6257, 2021.

[16] M. E. Taylor and P. Stone, "Behavior transfer for value-function-based reinforcement learning," in Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems, 2005, pp. 53–59.

[17] T. Brys, A. Harutyunyan, M. E. Taylor, and A. Nowé, "Policy transfer using reward shaping." in AAMAS, 2015, pp. 181–188.

[18] I. Clavera, D. Held, and P. Abbeel, "Policy transfer via modularity and reward guiding," in 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 1537–1544.

[19] J. Maxwell Donelan, R. Kram, and K. Arthur D, "Mechanical and metabolic determinants of the preferred step width in human walking," Proceedings of the Royal Society of London. Series B: Biological Sciences, vol. 268, no. 1480, pp. 1985–1992, 2001.

[20] K. E. Zelik and A. D. Kuo, "Mechanical work as an indirect measure of subjective costs influencing human movement," PloS one, vol. 7, no. 2, p. e31143, 2012.

[21] R. Ranganathan, A. Adewuyi, and F. A. Mussa-Ivaldi, "Learning to be lazy: exploiting redundancy in a novel task to minimize movement-related effort," Journal of Neuroscience, vol. 33, no. 7, 2013.

[22] C. Hesse, K. Kangur, and A. R. Hunt, "Decision making in slow and rapid reaching: Sacrificing success to minimize effort," Cognition, vol. 205, p. 104426, 2020.

[23] L. R. Fournier, E. Coder, C. Kogan, N. Raghunath, E. Taddese, and D. A. Rosenbaum, "Which task will we choose first? precrastination and cognitive load in task ordering," Attention, Perception, & Psychophysics, vol. 81, no. 2, pp. 489–503, 2019.

[24] D. Sadigh, A. D. Dragan, S. Sastry, and S. A. Seshia, "Active preference-based learning of reward functions." in Robotics: Science and Systems, 2017.

[25] M. E. Taylor and P. Stone, "Transfer learning for reinforcement learning domains: A survey." Journal of Machine Learning Research, vol. 10, no. 7, 2009.

[26] O. Day and T. M. Khoshgoftaar, "A survey on heterogeneous transfer learning," Journal of Big Data, vol. 4, no. 1, pp. 1–42, 2017.

[27] M. Jing, X. Ma, W. Huang, F. Sun, and H. Liu, "Task transfer by preference-based cost learning," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 01, 2019, pp. 2471–2478.

[28] Z. Cao, M. Kwon, and D. Sadigh, "Transfer reinforcement learning across homotopy classes," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 2706–2713, 2021.

[29] W. Wang, R. Li, Y. Chen, Z. M. Diekel, and Y. Jia, "Facilitating human–robot collaborative tasks by teaching-learning-collaboration from human demonstrations," IEEE Transactions on Automation Science and Engineering, vol. 16, no. 2, pp. 640–653, 2018.

[30] M. C. Gombolay, C. Huang, and J. Shah, "Coordination of human-robot teaming with human task preferences," in 2015 AAAI Fall Symposium Series, 2015.

[31] R. Bellman, "A markovian decision process," Journal of mathematics and mechanics, vol. 6, no. 5, pp. 679–684, 1957.

[32] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in 2011 IEEE International Conference on Robotics and Automation. IEEE, 2011, pp. 3400–3407.

[33] S. Nikolaidis, R. Ueda, A. Hayashi, and T. Arai, "Optimal camera placement considering mobile robot trajectory," in 2008 IEEE International Conference on Robotics and Biomimetics. IEEE, 2009, pp. 1393–1396.

[34] S. Nikolaidis, A. Dragan, and S. Srinivasa, "Viewpoint-based legibility optimization," in 2016 11th ACM/IEEE International Conference on Human-Robot Interaction (HRI). IEEE, 2016, pp. 271–278.