# 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 preferences 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 user preferences in actual assembly tasks from demonstrations provided by the user in a representative *canonical task*. Inspired by previous work in economy of human movement, we propose to represent user preferences as a linear function of abstract task-agnostic features, such as movement and physical and mental effort required by the user. For each user, we learn their preference from demonstrations in a canonical task and use the learned preference 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 transferred from canonical to actual assembly tasks, enabling robots to anticipate user actions.**

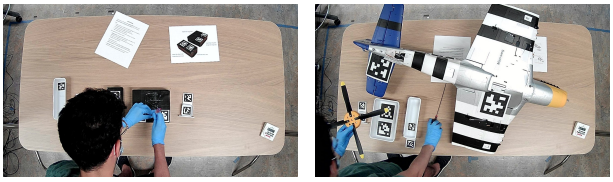
## I. Introduction

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 tools or clearing out the assembly area, while humans focus on the high value actions, e.g., using the tools for assembly. 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 order, 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]–[13] that captures the user's preferred sequence of actions. However, providing demonstrations for each assembly task that a given user must perform is tedious and time-consuming.

Heramb Nemlekar, Runyu Guan, Guanyang Luo, Satyandra K. Gupta and Stefanos Nikolaidis are with the University of Southern California, USA. {nemlekar, guanyu, luog, guptask, nikolaid}@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 [14]–[16], 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 [17]–[20] and task ordering [21], 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 physical and mental effort required by the user to perform the actions in the assembly tasks. 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 start of the canonical task, they will likely prefer the same, i.e., to start with high-effort actions, in the actual task.

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

## II. Related Work

Similar to prior work [5], [12], [22], 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* [15], [16], [23] 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. The work that is closest to our problem transfers the preferences of a simulated human from a (source) block stacking task to a target task with an extra red block [6]. However, to our knowledge, no prior work has studied transferring preferences of real users across assembly tasks.

### B. Factors affecting human preferences in physical tasks

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 [17], [19], [20] 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 in the task. For example, in an object pick-up task [21] 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 [18], 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 $X$. However, because each worker can have their own preferred sequence $\xi_X = [a_1, \ldots, a_t, \ldots, a_N]$ 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 $\xi_C$ 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 $\phi : \{S_C, S_X\} \mapsto \Phi$ from a task-agnostic feature space $\Phi$. Given $\phi$, we can map any state in the canonical and actual tasks to a $d$-dimensional feature vector in $\Phi$.

As the rewards received by a user depend on the state of the task, given $\phi$, we can model the reward function of a given user $i$ as a function of the abstract features of the states, i.e., $R_{X,i}(s) = f_{X,i}(\phi(s)) \quad \forall\, s \in S_X$.

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., $f_{X,i} \simeq f_{C,i}$, given that: (i) the feature space $\Phi$ 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 $\phi$, we can learn each user's $f_{C,i}$ from their demonstrations $\xi_C$ 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.

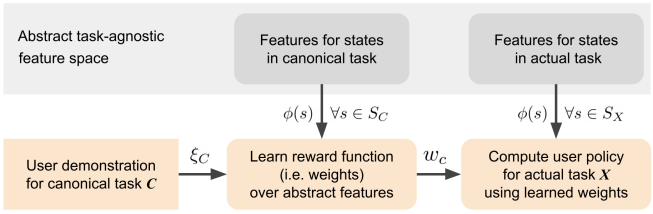



Fig. 2: Flowchart of our proposed method for transferring preferences

**Approach.** Following prior work [11], we model the reward function $R_{X,i}(s) = w_{X,i}^T \phi(s)$ as linear in the features $\phi(s)$. Here, $w$ is a $d$-dimensional vector of weights where each weight represents the user's preference for a particular dimension of the feature space.

Given a demonstration sequence $\xi_C = [a_1, \ldots, a_M]$ of actions $a \in A_C$, 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 states in the canonical task with the same frequency as that in $\xi_C$. 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 hypothesis, we assume that user $i$ would have the same weights (i.e., $w_{X,i} = w_{C,i}$) for the features $\phi$ in the actual task. Thus, we can calculate the transferred rewards $\tilde{R}_X$ by using the weights $w_C$:

$$\tilde{R}_{X,i}(s) = w_{C,i}^T \phi(s) \qquad \forall s \in S_X \tag{1}$$

Fig. 2 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

[24] for all states $s \in S_X$ 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. User Study

We want to show that user preferences learned from demonstrations in a canonical task can be used to anticipate their actions in an actual assembly task. Therefore, 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 weights (i.e. preference) learned from demonstrations in the canonical task.

### A. Actual assembly task

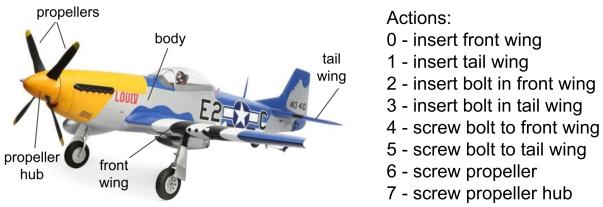



Fig. 3: Actual assembly task: Airplane model and task actions.

We choose an RC model-airplane assembly (see Fig. 3) as our actual task. The different actions in this assembly can be sequenced in multiple ways, with a few constraints, e.g., action 2 must precede action 4 since a bolt must be inserted before it is screwed. 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 the actual task.

### B. Task-agnostic feature space

Inspired from prior work in economy of human movement [17], [19], [20], we presume that users would try to minimize their movement throughout the task. 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 | Value | Preference |
|---|---|---|
| $\phi_{\mathcal{P}}$ | $\{0,1\}$ | Keep same part |
| $\phi_{\mathcal{T}}$ | $\{0,1\}$ | Keep same tool |

[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.

For a state $s_{t+1}$, $\phi_{\mathcal{P}}(s_{t+1}) = 1$ if the latest action $a_t$ uses the same part as the previous action $a_{t-1}$. Similarly, $\phi_{\mathcal{T}}(s_{t+1}) = 1$ when $a_t$ requires the same tool as $a_{t-1}$.

In addition to movement cost, users may also sequence their actions based on the mental ($\varepsilon_m$) and physical ($\varepsilon_p$) effort requried to perform them [21]. 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 and leave the harder actions for the end.

We model the features for user preferences in performing high-effort actions at the end as - $\psi\ \varepsilon_m$ and $\psi\ \varepsilon_p$; where $\psi : s \mapsto [0,1]$ 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. We can see that at the start, i.e., $\psi \simeq 0$, the feature value of an action will be smaller than at the end ($\psi \simeq 1$). Thus to maximize the accumulated reward, users will backload the high-effort actions.

We also consider the opposite scenario where workers prefer to perform the high-effort actions at the start by including the features - $(1-\psi)\ \varepsilon_m$ and $(1-\psi)\ \varepsilon_p$. Thus, we model the following features to capture preferences based on the physical and mental effort of actions:

TABLE II: Effort-Based Features

| Feature | Variable | Preference |
|---|---|---|
| $\phi_{f,p}$ | $\psi_f \varepsilon_p$ | Frontloading of high $\varepsilon_p$ actions |
| $\phi_{f,m}$ | $\psi_f \varepsilon_m$ | Frontloading of high $\varepsilon_m$ actions |
| $\phi_{b,p}$ | $\psi_b \varepsilon_p$ | Backloading of high $\varepsilon_p$ actions |
| $\phi_{b,m}$ | $\psi_b \varepsilon_m$ | Backloading of high $\varepsilon_m$ actions |

Here, $\psi_f = 1 - \psi$ and $\psi_b = \psi$. We use the six features from Tables I, II to create our feature function $\phi(s)$, that maps each state $s$ in the canonical and actual task to a 6-dimensional feature space.

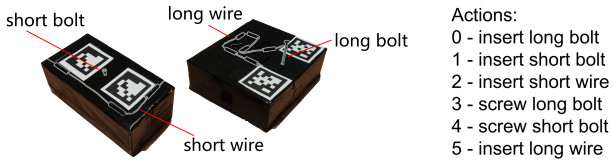



Fig. 4: Canonical assembly task: (Left) Model, (Right) Task actions.

### C. Canonical assembly task

In this work, we empirically design the canonical task 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. 4), 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 two actions that use 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 physical and mental effort.

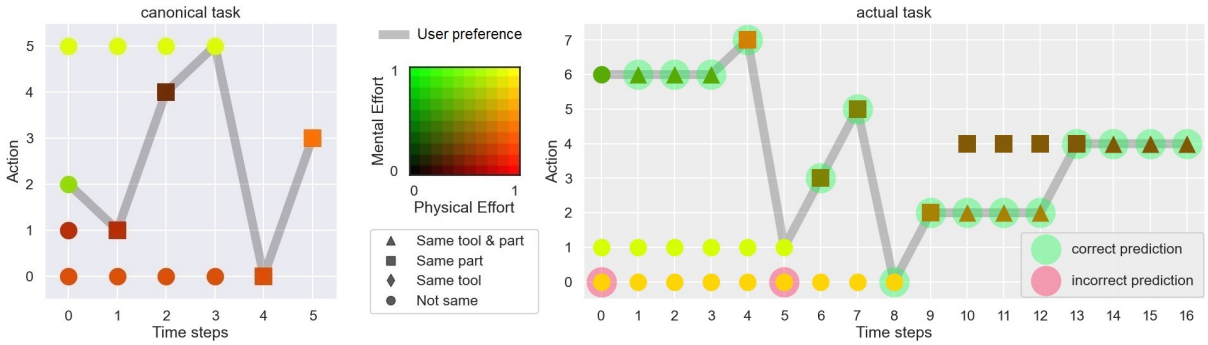



Fig. 5: Example of a user that prefers to keep working on the same part (□), and perform actions with high physical effort (red) at the end of the task. In the above plots, the x-axis represents the steps (progress) in the task, and y-axis represents all the different actions in the task. At each time 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 V.

TABLE III: Actions with distinct physical and mental efforts.

| | Low $\varepsilon_p$ | High $\varepsilon_p$ |
|---|---|---|
| Low $\varepsilon_m$ | Action 4 | Action 3 |
| High $\varepsilon_m$ | Action 2 | Action 5 |

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 an average users required only 3.83 minutes to complete the task. We leave the problem of formalizing the process of designing a canonical task for a given actual task for future work.

### D. Study protocol

We recruited 11 ($M = 10$, $F = 1$) participants from the graduate student population at the University of Southern California (USC) and compensated each user with 20 USD.

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 fill in a *post-training questionnaire* (Table IV) to rate the physical and mental effort they required for performing each action in the task, and an (ii) execution round - where users plan and demonstrate the preference, and then explain their preference in a *post-execution questionnaire*.

*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. We also do not tell them to specifically consider effort or movement cost while planning their preferred sequence.

| |
|---|
| Q1. Please rate the physical effort that was required for each action. |
| Q2. Please rate the mental effort that was required for each action. |

TABLE IV: Post-training questionnaire (ratings on 1-7 Likert scale) used to obtain the values for $\varepsilon_p$ and $\varepsilon_m$ for each action in the task.

## V. 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**).

### A. User preferences in the canonical and actual tasks

While many users (6 out of 11) prefer to minimize changing parts more than other features, some users also give a higher weightage to minimizing physical or mental effort. Consider the demonstrations shown in Fig. 5, 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 ($\phi_{\mathcal{P}}$), followed by the feature of backloading high physical effort actions ($\phi_{b,p}$). By using these weights to calculate rewards in the actual task we are able to accurately anticipate the actions at most time steps.

However, we incorrectly predict at time steps 0 and 5. This is because we also learn a high weight for performing low mental effort actions at the start ($\phi_{b,m}$), since the last three actions in the canonical task 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. Because the user's 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.

### B. Accuracy of anticipating user actions in the actual task

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.

In baseline (i), we randomly select an action from the remaining actions that can be executed at each time step. 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. For each user, we run the above baselines for 100 trials and compute the average accuracy over all trials.

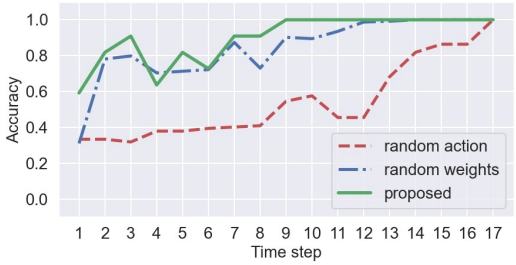



Fig. 6: Mean accuracy of predicting the user actions at each time step (averaged over all users) in the actual assembly task.

A paired t-test showed a statistically significant difference ($t(10) = 15.52$, $p < 0.001$) between the mean accuracy for our proposed approach ($M = 0.901$, $SE = 0.025$) and the mean accuracy for random actions ($M = 0.526$, $SE = 0.019$). This supports **H1**. Next, we compare our proposed approach to assigning random weights (averaged over all time steps). A paired t-test showed a statistically significant difference ($t(10) = 3.077$, $p = 0.011$) between the mean accuracy for our proposed approach and the mean accuracy for random weights ($M = 0.843$, $SE = 0.015$). This supports **H2**. Fig. 6 shows the mean accuracy of predicting the action at each time step. We can see that the accuracy for all methods increases as we reach the final time step, as there are fewer actions to choose at the end. The accuracy for our proposed method is significantly higher at the start as we correctly anticipate the first action for 6 users based on their transferred weights.

## VI. Conclusion

Our work demonstrates the potential of anticipating user actions in actual assembly tasks based on preferences learned in an abstract, shorter canonical task, thus significantly reducing the time and human effort required to provide the demonstrations. 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 and also update the weights based on new observations of user actions in the actual task.

## 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] 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.

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

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

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

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

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