Modeling Human Sequential Decision-Making in the Tower of London: Incorporating Individual Differences and Timing-Based Replanning Inference

Chenyuan Zhang[1] **Yuansan Liu**[2]

Dana Kulic[1] **Pamela Carreno-Medrano**[1] **Michael Burke**[1]

[1] Monash University [2] University of Melbourne

Abstract

Modeling human sequential decision-making behavior presents a significant challenge for researchers in artificial intelligence, robotics, and cognitive science. In this paper, we introduce a human behavior model designed to predict actions in the Tower of London task, addressing two critical aspects that have been largely overlooked in existing methodologies. First, we propose a profile-based action prediction framework that extracts user and task profiles from historical data, enhancing action prediction in novel scenarios. Second, we introduce a replanning detection component that leverages thinking time as an indicator of planning processes in the human mind, enabling a more precise representation of cognitive dynamics. Our evaluations demonstrate the effectiveness of the proposed model, achieving superior performance in behavior prediction within the Tower of London task. This work lays the foundation for more robust human behavior modeling in sequential decision-making environments.

Keywords: Human Behavior Modelling; Sequential Decision-Making; Tower of London; Replanning

Introduction

In dynamic environments such as busy warehouses, robotic assistants must accurately predict human behaviors to improve efficiency and safety (Jahanmahin, Masoud, Rickli, & Djuric, 2022; Tian, Sun, Bajcsy, Tomizuka, & Dragan, 2022). Consider a scenario where a robot assists warehouse workers in picking and moving items. To provide effective support, the robot must anticipate the worker's next action. Suppose a worker needs to retrieve both a large box from the box area and a fragile glass vase from the vase area for an outgoing order. Worker A might choose to pick the glass vase first, while Worker B might prefer to start with the large box. Furthermore, if the task involves a small box instead of a large one, Worker A might now opt to pick the box first rather than the vase. This example illustrates the complexity of behavior prediction, as decision-making patterns vary between individuals and are influenced by the specific structure of the task. However, these variations are largely overlooked in cognitive modelling due to the challenges of capturing and modeling them. Learning-based methods often seek to learn latent representations to capture diverse patterns (Albrecht & Stone, 2018; Beliaev, Shih, Ermon, Sadigh, & Pedarsani, 2022; Wang et al., 2017). Nevertheless, these models typically require large amounts of training data, which is often unavailable for human-related tasks. To address this limitation, we propose a model-based approach tailored for settings with limited data. Our framework, Profile-based Action Prediction (ProAct), extracts user-specific features (e.g., tendencies for in-depth thinking) and task-specific features to predict human behaviors in novel scenarios.

While user-specific features are crucial for better behavior prediction, auxiliary information (e.g., thinking time) also provides additional insights into cognitive processes, potentially further enhancing predictive performance. Returning to the warehouse example, now a worker needs to retrieve a large box from the box area and a fragile glass vase from the vase area for an outgoing order. As the worker approaches the box area, the robot may initially infer that they intend to pick up the large box first. However, after reaching for the box, the worker suddenly pauses. If the robot overlooks this pause, it risks assisting with the box, potentially leading to accidents if the worker has changed their mind. On the other hand, if the robot recognizes the pause as a sign of the worker reconsidering their plan, it could infer that the worker might have changed the order of subtasks and chose to pick up the vase first instead. In the latter instance, the robot can decide to wait and gather more information before providing the appropriate assistance.

This scenario demonstrates how thinking time can indicate a potential shift in the original plan (i.e., replanning). While timing information has received limited attention in human behavior modeling (Jahanmahin et al., 2022; Semeraro, Griffiths, & Cangelosi, 2023), recent studies have started exploring how auxiliary data—such as thinking time (Zhang, Kemp, & Lipovetzky, 2023) and gaze (Singh et al., 2018)—can help infer mental states. However, these studies typically focus on isolated decisions, overlooking the sequential dependencies inherent in sequential decision-making tasks. To the best of our knowledge, no prior work has investigated the role of thinking time in understanding replanning behavior in sequential decision-making. In this paper, we introduce a novel replanning detection component that leverages thinking time to estimate the likelihood of planning at each step, aiming to reveal unobservable decision-making processes from observable behaviors.

In this work, we test the ProAct framework and the replanning detection component with a repeated lookahead search algorithm on the Tower of London (TOL) task and assess how effectively the proposed model captures human diverse behavior in this task.

In D. Barner, N.R. Bramley, A. Ruggeri and C.M. Walker (Eds.), *Proceedings of the 47th Annual Conference of the Cognitive Science Society*

This paper makes three interconnected contributions to the behavior prediction problem. First, we introduce the ProAct framework, which effectively leverages the identity information to extract user-specific and task-specific features, illustrating how this information can enhance action prediction. Second, we implement a repeated lookahead search algorithm to predict human behavior in the Tower of London task, showcasing the practical application of the ProAct framework. Finally, we introduce a novel replanning detection component that, combined with the repeated lookahead search, leverages timing information to improve action prediction by identifying steps where replanning is likely to occur.

Profile-based Action Prediction (ProAct)

We begin by formulating the problem. Our objective is to predict human behavior in a single-agent sequential decision-making tasks based on historical data with identity information.

Problem Formulation

Formally, we are provided with a training dataset D comprising behavior data from N users performing M different tasks. The dataset D includes only a subset of user-task pairs (i, j), indicated by $\mathbb{I}_{i,j}(D) = 1$ if user i has performed task j in the dataset D, and $\mathbb{I}_{i,j}(D) = 0$ otherwise. Each task is modelled as a uniform cost goal-directed Markov Decision Process (MDP) with varying start and goal states, defined on a state space $\mathbb{S}$ with an action space $\mathbb{A}$. For simplicity, in this work, we assume a deterministic, finite MDP, meaning the transition function F is deterministic, $F : \mathbb{S} \times \mathbb{A} \to \mathbb{S}$. The user's behavior in a task is represented as a trajectory $traj_{i,j} = (s_0, a_0, s_1, \ldots, a_{l_{i,j}-1}, s_{l_{i,j}})$, defined for pairs where $\mathbb{I}_{i,j}(D) = 1$, where $l_{i,j}$ indicates the solution length, and $s_0 = s_{0_j}$ and $s_{l_{i,j}} = s_{g_j}$ denote the start and goal states of task j, respectively.

The objective is to forecast the behavior of a specific user $n \in N$ performing a specific task $m \in M$ given trajectories $traj_{n,m}$ based on the dataset D. Given that our transition function is deterministic and known, we reduce this trajectory prediction problem to a sequence of action prediction problems, in which we predict the action a_k given the prefix $prefix_k = (s_0, a_0, s_1, \ldots, a_{k-1}, s_k)$ of the trajectory in turn, expressed as $\hat{a_k} = f(prefix_k, D)$. The complete trajectory prediction can then be achieved iteratively by appending each new predicted action $\hat{a_k}$ and the corresponding successor state $\hat{s_{k+1}} = F(s_k, a_k)$ to the prefix.

We focus on a behavior prediction scenario where the target user is part of the training dataset and leave the more challenging case—where the target user is unseen during training—for future work.

Profile-based Action Prediction (ProAct)

In this section, we introduce **ProAct**, a profile-based action prediction framework designed to address the proposed problem. This framework is compatible with both model-based and model-free approaches, provided that a compressed representation of the trajectory can be obtained; however, in this paper, we focus on the model-based approach.

Conceptually, the trajectory is influenced by both the task-specific context and the user's behavioral tendencies. Therefore, our objective is to capture both task-specific feature f_t (i.e. task profile) and user-specific information f_u (i.e. user profile) from the dataset D first. These learned profiles are then employed to predict actions in the following manner: $\hat{a_k} = M(prefix_k, f_u, f_t)$. Under this framework, we can generate explainable predictions for unseen combinations of users and tasks based on historical data.

The framework has three steps: first, obtaining a compressed representation from the trajectories (**Trajectory Encoding**); second, extracting both the user profile and task profile from the compressed representation (**Profile Extraction**); and third, using the extracted profiles to make predictions (**Action Prediction**). In the following subsections, we will provide details for each step.

Trajectory Encoding

The objective of this step is to encode the observed trajectory $traj_{i,j}$ into a compressed representation $z_{i,j}$. The representation will then be used to extract the user profile and task profile in the next step.

In a model-based approach, the compressed representation corresponds to a set of preselected model parameters, which can be estimated via Bayesian inference methods. Specifically, the compressed representation $z_{i,j}$ for the trajectory $traj_{i,j}$ is inferred using maximum a posteriori (MAP) estimation formulated as

$$\hat{z_{i,j}} = \arg\max_{z} \text{Prior}(z) \prod_{l=0}^{l_{i,j}-1} p(a_l \mid prefix_l, z)$$

.

Profile Extraction

In this step, we will extract task profiles and user profiles from the compressed representation obtained in the previous step. We assume a linear relationship where the compressed representation $z_{i,j}$ is expressed as the sum of the user profile f_{u_i}, task profile f_{t_j}, and noise $\varepsilon_{i,j}$, i.e.,

$$z_{i,j} = f_{u_i} + f_{t_j} + \varepsilon_{i,j}, \tag{1}$$

where ε represents the noise. In the Tower of London task, $z_{i,j}$ represents the search depth for trial (i, j), as this parameter predominantly influences behavior in the tree search model. Within this framework, the profiles f_{u_i} and f_{t_j} capture user- and task-specific biases in planning depth, respectively, while the noise ε accounts for inherent variability in human planning. It is important to note that the values of user and task profiles are not necessarily integers, as they reflect overall tendencies rather than the exact depth in a specific trial.

Given this formulation, the problem reduces to a linear regression problem, where the least squares estimation can be

employed to minimize the noise and obtain optimal estimates for f_u and f_t.

Action Prediction

Once we have the estimated user profile $\hat{f_{u_i}}$ for user i and task profile $\hat{f_{t_j}}$ for task j, we can predict the action a_k that maximizes the likelihood as:

$$\hat{a}_k = \arg\max_{a \in \mathbb{A}} p(a \mid prefix_k, \hat{z}_{i,j})$$

,

where the compressed representation $\hat{z}_{i,j}$ is given by $\hat{z}_{i,j} = \hat{f_{u_i}} + \hat{f_{t_j}}$.

The key consideration in applying the framework to a specific domain is determining how to estimate the likelihood function $p(a \mid prefix, z)$. In the following sections, we present how this likelihood can be approximated within the framework to predict human behavior in the Tower of London task.

Tower of London (TOL) Task

Disc-moving tasks are one of the classic domains in planning and cognitive research (Bylander, 1994). These tasks typically involve rearranging items into specific configurations, requiring explicit forward reasoning, making them an excellent environment for investigating human planning behavior. As a variant of the Tower of Hanoi, the Tower of London (TOL) introduces additional complexities due to its flexibility in start and goal configurations, and it is widely used in research on human problem-solving (Donnarumma, Maisto, & Pezzulo, 2016; Berg, Byrd, McNamara, & Case, 2010; Ayala, Zafar, & Niechwiej-Szwedo, 2022).

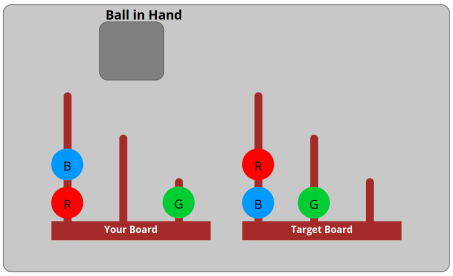

Figure 1: Tower of London (TOL) task

We use the same TOL tasks as Berg et al. (2010), which consist of 117 different tasks, each with optimal solutions ranging from 8 to 14 moves in length. Figure 1 illustrates an example of a Tower of London task used in the human experiment. The board displayed here features pegs that can hold 3, 2, and 1 ball, respectively, from left to right. Participants are presented with a board in an initial configuration (left) and are asked to either pick up a ball from a peg to their hand or put down a ball from their hand to a peg until the left board matches the given target board (right).

Our experiment was carried out with approval from Human Ethics Advisory Group at XXX. A total of 239 participants completed the experiment on Prolific, with each participant completing 39 TOL tasks randomly selected from the 117 available tasks. Consequently, the dataset includes actions and response times recorded for a total of 9,321 trajectories.

ProAct on TOL

In this section, we explain how the ProAct framework can be applied to the TOL task to better capture human behavior and improve action prediction.

Specifically, we use a **repeated lookahead search** paradigm to simulate human behavior in TOL, extending established methodologies for modelling human behavior in sequential decision-making (Zhang, Lipovetzky, & Kemp, 2023; Krusche, Schulz, Guez, & Speekenbrink, 2018; Kuperwajs, Van Opheusden, & Ma, 2019; Mattar & Lengyel, 2022). Building on their research trends, we propose that the primary factor influencing user behavior is the depth d of their lookahead search, which remains constant throughout a given trajectory, reflecting a stable cognitive strategy during task execution. Consequently, in the trajectory encoding step, d becomes the key parameter to be inferred from observed behavior, with the compressed representation expressed as $z = d$.

Repeated Lookahead Search

Lookahead search is widely used in existing research for modeling human behavior. However, most studies assume that replanning occurs at every step (Kuperwajs et al., 2019; Krusche et al., 2018), which may be valid in complex or uncertain environments. In a simple deterministic environment like TOL, replanning can still occur, as individuals may not create a complete plan before execution or may only develop ambiguous plans in their minds, realizing that the plan requires refinement during execution. Therefore, we model human sequential decision-making in TOL as follows: individuals initially perform a lookahead search with a specific depth d to generate a partial plan. At various points within this partial plan, they may choose to initiate another lookahead search with the same depth to refine their plan.

Likelihood Estimation in TOL

There are two major challenges associated with the repeated lookahead search paradigm in TOL. The first challenge lies in determining the search depth from a single action, especially given the limited action space in TOL. The second challenge is the uncertainty about whether users are following a previously generated plan or initiating replanning at each step. To tackle these challenges, we propose a novel component that leverages the thinking time recorded at each step to infer when users are engaged in replanning. This algorithm is incorporated into both the trajectory encoding and action prediction processes by augmenting the prefix used in the likelihood estimation with thinking times. Specifically, the prefix now includes the thinking time x_i for each step,

$$prefix_k = (s_0, x_0, a_0), \ldots, (s_{k-1}, x_{k-1}, a_{k-1}), (s_k, x_k).$$

The likelihood function we aim to approximate is given by:

$$P(traj_{i,j}|\hat{z_{i,j}}) = P(a_0, a_1, ..., a_{l_{i,j}} \mid d_{i,j}) \tag{2}$$

$$= \prod_{t=0}^{l_{i,j}} p(a_t \mid prefix_t, d_{i,j}). \tag{3}$$

Here, $d_{i,j}$ denotes the search depth in the specific trial that generated the trajectory $\text{traj}_{i,j}$, and $l_{i,j}$ represents the solution length for this trajectory. For simplicity, we will omit the subscripts and refer to them as d, traj, and l, respectively, in the following paragraphs where no ambiguity arises.

Estimating $p(a_t \mid prefix_t, d)$ is challenging if we do not know whether a_t originates from a replan at this step or from a plan established in previous steps. Notably, we do not need to enumerate all potential replanning steps from prior actions, as replanning effectively resets the decision-making process. Therefore, we can estimate it by marginalizing over all possibilities from the last planning step t_r, which ranges from $\max(t-d+1, 0)$ to t. Here, $t_r = t$ indicates that the user is (re)planning at the current step, while $t_r = t-d+1$ denotes the earliest planning step from which the user can still execute action a_t without replanning in between. Therefore, we can calculate $p(a_t \mid prefix_t, d)$ as follows:

$$p(a_t \mid prefix_t, d) \tag{4}$$

$$= \sum_{t_r=\max(t-d+1,0)}^{t} P(a_t, \text{Pl} = t_r \mid prefix_t, d) \tag{5}$$

$$= \sum_{t_r} P(a_t \mid \text{Pl} = t_r, prefix_t, d) P(\text{Pl} = t_r \mid prefix_t, d) \tag{6}$$

where $\text{Pl} = t_r$ means the last planning happens at step t_r.

Estimating $P(\text{Pl} = t_r \mid prefix_t, d)$ still appears challenging; however, considering the intuition that the probability of planning is higher iff a user's thinking time at a given step is longer, we can assume the thinking time x_{t_r} is influenced solely by the occurrence of planning at this step, other actions and previous thinking times are conditionally independent of the event of planning at t_r given x_{t_r}. Consequently, this allows us to simplify the estimation to $P(\text{Pl} = t_r \mid x_{t_r})$. We will use the replanning detection component to estimate this probability, which will be introduced in the next section.

For the estimation of $P(a_t \mid \text{Pl} = t_r, prefix_t, d)$, we simulate the lookahead search with depth d from s_{t_r} (i.e. the state where replanning happens), and use softmax to estimate the probability of action selection at state node s_t. This is equivalent to running a lookahead search with depth $d-t+t_r$ from state s_t. Thus, we have

$$P(a_t \mid \text{Pl} = rp, prefix_t, d) = \frac{\exp(Q_{d-t+t_r}(s_t, a_t))}{\sum_{a' \in \mathbb{A}} \exp(Q_{d-t+t_r}(s_t, a'))}, \tag{7}$$

Where $Q_d(s,a)$ is the value estimated from the simulated lookahead search with depth d from state s. For the initialization of the Q-function at the newly generated nodes, we use negative perceived distance (Donnarumma et al., 2016), defined as the number of balls not in their goal positions, as cognitive studies on TOL (Berg et al., 2010; Zhang, Lipovetzky, & Kemp, 2023) suggest this is how humans estimate progress in the Tower of London task. The values from these newly generated nodes are then backpropagated to other tree nodes using the best-child backup strategy.

Replanning Detection Component

Planning Behavior

We use thinking time as the primary indicator of planning behavior because it represents the duration between successive actions and serves as cognitive processing delay. Given a series of thinking times $\boldsymbol{x}_{1:T} = \{x_1, x_2, \cdots, x_T\}, T \in \mathbb{N}^*$, where each x_i is the thinking time at step i as aforementioned, assuming the only planning behavior happens at time step t_r, and all the other steps are normal behaviors. Then, t_r is the time step where the statistical properties, such as mean or variance, deviates significantly from normal ones.

Dynamic Real-time Replanning Detection

According to the definition of the planning behaviors, we can first recognize them as anomalies in the thinking time series data, and then apply statistical approaches to automatically detect them. This type of method usually fits a distribution model on normal data and applies a statistical inference test to determine if the new data belongs to the underlying distribution (Chandola, Banerjee, & Kumar, 2009). Inspired by this principle, and in order to enable dynamic adjustments of the distribution model, we use two distributions to model the normal and abnormal data separately.

We model both behaviors using Gaussian distributions with unknown mean and variance $\mathcal{N}(\mu, \sigma^2)$. We assume normal behaviors follow a normal distribution $\mathcal{P}_{normal} = \mathcal{N}(0,1)$, functioning as the prior belief of the normal behaviors. Then, we filter out the abnormal behaviors $\boldsymbol{x}_{abnormal}$ from a sequence based on the deviation to the sequence median value, and use their mean and variance (μ_a, σ_a^2) to form the prior belief of anomaly distribution: $\mathcal{P}_{abnormal} = \mathcal{N}(\mu_a, \sigma_a^2)$.

The standard procedure of our anomaly detection requires calculating the likelihood of a new observation being drawn from both the normal and abnormal distribution $\mathcal{L}_n, \mathcal{L}_a$, and getting the probability of anomaly $P_{abnormal}$ based on these two likelihood results, $P_{abnormal} = \frac{\mathcal{L}_a}{\mathcal{L}_a + \mathcal{L}_n}$. Here, in the absence of actual probabilities, we assume equal probabilities for normal and abnormal behaviors, without loss of generality. In addition, we use Bayesian inference to dynamically adjust the distribution models with respect to external constraints, i.e., the last planning step. To allow a closed-form expression and sequential update for posterior distribution, we apply conjugate prior in the Bayesian inference (Raiffa & Schlaifer, 2000). For our distribution model $\mathcal{N}(\mu, \sigma^2)$, the conjugate prior is a *Normal-Gamma* (Murphy, 2007). For all values of $t_r \in [t-d+1, t]$, we first update the abnormal parameters based on x_{t_r}. Then, we sequentially update the nor-

mal parameters based on the subseries $\boldsymbol{x}_{t_r+1:t-1}$, which does not contain planning behavior, and hence, should be recognized as a set of normal behaviors. Subsequently, we have two new distributions that leverage both the prior assumptions and the external constraints, allowing a better estimation of planning probability.

Model Comparison

We aim to evaluate two key hypotheses in this section. First, we want to verify whether incorporating thinking time enhances the quality of trajectory encoding, thereby improving the accuracy of action prediction. Second, we aim to test whether considering both user profiles and task profiles leads to further improvements in action prediction performance.

We randomly divide the Tower of London (TOL) dataset into training and test sets, allocating 80% for training and 20% for testing. To reduce the impact of randomness and enable robust statistical analysis, we repeat the process 10 times to obtain varied results. In each iteration, the training set is used exclusively to learn user and task profiles, while the test set evaluates the performance of each candidate model. Each TOL task is automatically encoded using PDDL with the Python package Tarski (Francés, Ramirez, & Collaborators, 2018). To capture a broad spectrum of user behaviors, we limit the trajectory length to the first 5 steps, as later steps tend to be more predictable for the model-based algorithm and often align with the optimal path to the goal.

For the first hypothesis, we use two different methods to estimate the replanning probability in equation (6) for trajectory encoding. The **Time-aware** method uses thinking time through the replanning detection algorithm as introduced earlier. The alternative **Time-agnostic** approach does not use thinking time; instead, it assumes that replanning occurs if and only if the user completes the original plan.

To test the second hypothesis, we perform an ablation study in which we compare the performance of several candidate models designed for the profile extraction step, alongside other baseline models. These models are selected based on their capacity to predict behavior under different assumptions regarding the influence of user and task profiles on user behavior. A brief description of each model is provided below:

1. **Vanilla Model (VM)**: It assumes that neither the user profile nor the task profile influences the user's planning behavior. In this case, the planning parameter d remains constant across all users and tasks, and it is chosen to maximize the likelihood of generating human behavior. The action prediction is then given by $\hat{a}_k = VM(prefix)$.

2. **Task-profile Model (TM)**: The task profile f_t is the only determinant of planning behavior, assuming that all users interact with the task in the same way. f_t is estimated by averaging the lookahead depths from all trajectories in the training set. The action prediction is then given by $\hat{a}_k = TM(prefix, f_t)$.

3. **User-profile Model (UM)**: This model assumes that each user has unique preferences and behavioral patterns that significantly influence their planning behavior, independently of the task they are completeing. The user profile f_u is estimated by averaging the lookahead depths from the user's trajectories in the training set. The action prediction is expressed as $\hat{a}_k = UM(prefix, f_u)$.

4. **User&Task-profile Model (ProAct)**: This model assumes that user profiles f_u and task profiles f_t jointly influence behavior. These profiles are inferred using the least squares estimator. The predicted action is then expressed as $\hat{a}_k = ProAct(prefix, f_t, f_u)$.

We use action prediction accuracy to quantify the performance of models.

Trajectory Encoding Results

To assess whether our framework can extract useful information from trajectory data, we first examine the distribution of search depths obtained from the training trajectories.

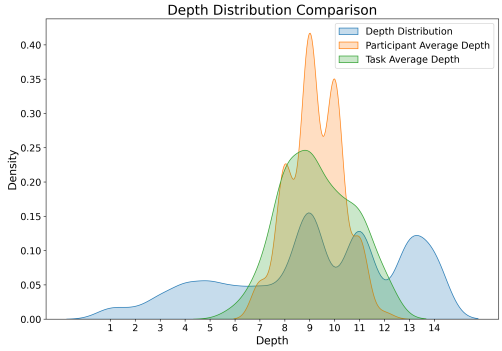

Figure 2: Depth Distribution from Time-Aware Trajectory Encoding

As shown in Fig 2, the overall distribution (blue) spans depths from 1 to 14, indicating substantial variability across all trajectories. The user average (orange), which represents the mean depth for each participant across their trajectories in the training set peaks around depths 8 to 10. Similarly, the task average (green), calculated as the average depth for each task, is also centered around 8 to 10. The narrower spread of the user average depth distribution compared to the task distribution suggests that user factors may have a smaller impact on search depth compared to task factors.

Action Prediction Performance

We evaluate the performance of each combination of trajectory encoding and profile extraction encoding approaches on the test trajectories. The results are presented in Table 1. Time-aware ProAct achieves the highest average accuracy of 70.14%.

We implemented a model-free baseline (behavior cloning, BC) and a model-based baseline (optimal agent) to validate

Table 1: Action Prediction Accuracy Represented as Mean(Variance) in Percentage (%)

Traj Encoding	Profile Extraction				baselines	
	VM	UM	TM	ProAct	Optimal	BC
Time-aware	66.86(0.50)	68.34(0.33)	68.61(0.77)	**70.14(0.89)**	54.03(0.22)	63.32(0.95)
Time-agnostic	62.44(0.36)	62.56(0.56)	65.02(1.30)	65.48(0.36)		52.77(0.90)

the effectiveness of our proposed approach. For statistical comparison, we used VM to establish a conservative benchmark. The optimal agent assumes that humans always select the optimal move toward the goal; however, it achieved an average accuracy of only 54.37%, which is significantly lower than the time-agnostic VM ($t(9) = 30.41, p < .001$). This result illustrate the complexity of predicting human behavior, as individuals exhibit suboptimality even in this simple task.

The model-free baseline employed behavior cloning using default settings from the Python imitation learning package (Gleave et al., 2022). This simple baseline assumes that neither user nor task profiles influence behavior, with both action selection and replanning solely dependent on the current state. Specifically, a two-layer multilayer perceptron (MLP) was used to learn an action prediction function, $\hat{a}_k = BC(s_k)$. The BC model achieved an average action prediction accuracy of 63.32% with timing information and 52.77% without timing information. All model-based models discussed in this study significantly outperformed this baseline ($t(9) = 9.39, p < .001$ vs. time-aware VM; $t(9) = 22.63, p < .001$ vs. time-agnostic VM), highlighting the advantages of model-based approaches in our domain.

A two-way ANOVA was conducted to examine the effects of model type and timing information on action prediction accuracy of four main models. There was a significant main effect of model type, $F(3,72) = 62.41, p < 0.001, \eta^2 = 0.72$, with ProAct achieving the highest accuracy ($M = 67.81, SD = 6.30$). A significant main effect of timing information was also observed, $F(1,72) = 669.37, p < 0.001, \eta^2 = 0.90$, indicating that models with timing information ($M = 68.48, SD = 1.97$) led to higher accuracy than those without using timing information ($M = 63.88, SD = 2.56$). We can see both hypotheses are confirmed: timing information can enhance action prediction performance, and utilizing both task and user profiles further benefits the performance.

We also observe that the task profile has greater predictive power than the user profile, consistent with our findings from the trajectory encoding results. Nevertheless, combining both task and user profiles provides additional benefits by further improving the likelihood and prediction accuracy, indicating that user-specific nuances, though secondary, contribute to a more comprehensive predictive model. However, we should be cautious to generalize this observation to other domains. In the TOL task, its simplicity limits the scope for users to demonstrate individual strategies; thus, user preferences and behavioral patterns, as captured in the user profile, may have less influence. In more open-ended environments, user profiles could play a more significant role.

Related Work

Having presented our results, we now situate our work within the context of related research. Tree search-based models are widely used to model human behavior in problem-solving tasks (Zhang, Lipovetzky, & Kemp, 2023; Callaway et al., 2022; Krusche et al., 2018; Kuperwajs et al., 2019; Mattar & Lengyel, 2022). However, their reliance on predefined heuristics and assumptions about human rationality may limit their flexibility in handling diverse behaviors, and they lack a systematic method to parameterize individual differences (Callaway et al., 2022; Mattar & Lengyel, 2022). Our work leverages the data efficiency of tree search-based methods while introducing mechanisms to capture individual user preferences and task-specific nuances, addressing some of the limitations associated with rigid planning assumptions.

Replanning behavior has been explored in robotics (Hou & Srinivasa, 2022; Garrett, Paxton, Lozano-Pérez, Kaelbling, & Fox, 2020) and planning (Yoon, Fern, Givan, & Kambhampati, 2008). These studies often rely on action sequences and overlook auxiliary information. In contrast, the cognitive science community has long used reaction times to infer mental processes in simpler decision-making tasks (Solway & Botvinick, 2015; Gates, Callaway, Ho, & Griffiths, 2021). Recently, researchers have also begun investigating how thinking time can reveal additional insights beyond observable actions in sequential decision-making tasks (Zhang, Kemp, & Lipovetzky, 2023; Berke, Tenenbaum, Sterling, & Jara-Ettinger, 2023). To our knowledge, none of the studies have integrated timing information within a comprehensive sequential decision-making context for action prediction.

Conclusion

In this paper, we investigate two largely overlooked aspects of human behavior modeling and propose an effective model to enhance human action prediction in the Tower of London (TOL) task. First, we introduce the ProAct framework, which extracts user and task profiles from historical data and augment action prediction by integrating both. Next, we design a novel replanning detection component, complemented by a repeated lookahead search mechanism that incorporates thinking time into the prediction process.

Experimental results demonstrate that the proposed algorithms leads to more accurate predictions of user behavior in the TOL task. This work lays the foundation for more robust human behavior modeling across diverse sequential decision-making environments. In future studies, we will apply ProAct and the replanning detection component to different domains to demonstrate the broad applicability of these algorithms.

References

Albrecht, S. V., & Stone, P. (2018). Autonomous agents modelling other agents: A comprehensive survey and open problems. *Artificial Intelligence*, *258*, 66–95.

Ayala, N., Zafar, A., & Niechwiej-Szwedo, E. (2022). Gaze behaviour: A window into distinct cognitive processes revealed by the tower of london test. *Vision Research*, *199*, 108072.

Beliaev, M., Shih, A., Ermon, S., Sadigh, D., & Pedarsani, R. (2022). Imitation learning by estimating expertise of demonstrators. In *International conference on machine learning* (pp. 1732–1748).

Berg, W. K., Byrd, D. L., McNamara, J. P., & Case, K. (2010). Deconstructing the tower: Parameters and predictors of problem difficulty on the Tower of London task. *Brain and Cognition*, *72*(3), 472–482.

Berke, M., Tenenbaum, A., Sterling, B., & Jara-Ettinger, J. (2023). Thinking about thinking as rational computation. In *Proceedings of the annual conference of the cognitive science society.*

Bylander, T. (1994). The computational complexity of propositional strips planning. *Artificial Intelligence*, *69*(1-2), 165–204.

Callaway, F., van Opheusden, B., Gul, S., Das, P., Krueger, P. M., Griffiths, T. L., & Lieder, F. (2022). Rational use of cognitive resources in human planning. *Nature Human Behaviour*, *6*(8), 1112–1125.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM computing surveys (CSUR)*, *41*(3), 1–58.

Donnarumma, F., Maisto, D., & Pezzulo, G. (2016). Problem solving as probabilistic inference with subgoaling: explaining human successes and pitfalls in the tower of Hanoi. *PLoS computational biology*, *12*(4), e1004864.

Francés, G., Ramirez, M., & Collaborators. (2018). *Tarski: An AI planning modeling framework.* `https://github.com/aig-upf/tarski`. GitHub.

Garrett, C. R., Paxton, C., Lozano-Pérez, T., Kaelbling, L. P., & Fox, D. (2020). Online replanning in belief space for partially observable task and motion problems. In *2020 international conference on robotics and automation* (pp. 5678–5684).

Gates, V., Callaway, F., Ho, M. K., & Griffiths, T. L. (2021). A rational model of people's inferences about others' preferences based on response times. *Cognition*, *217*, 104885.

Gleave, A., Taufeeque, M., Rocamonde, J., Jenner, E., Wang, S. H., Toyer, S., ... Russell, S. (2022). *imitation: Clean imitation learning implementations.* arXiv:2211.11972v1 [cs.LG]. Retrieved from `https://arxiv.org/abs/2211.11972`

Hou, B., & Srinivasa, S. S. (2022). Dynamic replanning with posterior sampling. In *International conference on intelligent robots and systems* (pp. 2938–2945).

Jahanmahin, R., Masoud, S., Rickli, J., & Djuric, A. (2022). Human-robot interactions in manufacturing: A survey of human behavior modeling. *Robotics and Computer-Integrated Manufacturing*, *78*, 102404.

Krusche, M. J., Schulz, E., Guez, A., & Speekenbrink, M. (2018). Adaptive planning in human search. *BioRxiv*, 268938.

Kuperwajs, I., Van Opheusden, B., & Ma, W. J. (2019). Prospective planning and retrospective learning in a large-scale combinatorial game. In *2019 conference on cognitive computational neuroscience* (pp. 13–16).

Mattar, M. G., & Lengyel, M. (2022). Planning in the brain. *Neuron*, *110*(6), 914–934.

Murphy, K. P. (2007). Conjugate bayesian analysis of the gaussian distribution. *def*, *1*, 16.

Raiffa, H., & Schlaifer, R. (2000). *Applied statistical decision theory* (Vol. 78). John Wiley & Sons.

Semeraro, F., Griffiths, A., & Cangelosi, A. (2023). Human–robot collaboration and machine learning: A systematic review of recent research. *Robotics and Computer-Integrated Manufacturing*, *79*, 102432.

Singh, R., Miller, T., Newn, J., Sonenberg, L., Velloso, E., & Vetere, F. (2018). Combining planning with gaze for online human intention recognition. In *Proceedings of the 17th international conference on autonomous agents and multiagent systems* (pp. 488–496).

Solway, A., & Botvinick, M. M. (2015). Evidence integration in model-based tree search. *Proceedings of the National Academy of Sciences*, *112*(37), 11708–11713.

Tian, R., Sun, L., Bajcsy, A., Tomizuka, M., & Dragan, A. D. (2022). Safety assurances for human-robot interaction via confidence-aware game-theoretic human models. In *2022 international conference on robotics and automation* (pp. 11229–11235).

Wang, Z., Merel, J. S., Reed, S. E., de Freitas, N., Wayne, G., & Heess, N. (2017). Robust imitation of diverse behaviors. *Advances in Neural Information Processing Systems*, *30*.

Yoon, S. W., Fern, A., Givan, R., & Kambhampati, S. (2008). Probabilistic planning via determinization in hindsight. In *Aaai* (pp. 1010–1016).

Zhang, C., Kemp, C., & Lipovetzky, N. (2023). Goal recognition with timing information. In *Proceedings of the international conference on automated planning and scheduling* (Vol. 33, pp. 443–451).

Zhang, C., Lipovetzky, N., & Kemp, C. (2023). Comparing ai planning algorithms with humans on the tower of london task. In *Proceedings of the annual meeting of the cognitive science society* (Vol. 45).