NOPA: Neurally-guided Online Probabilistic Assistance for Building Socially Intelligent Home Assistants

Xavier Puig*, Tianmin Shu*, Joshua B. Tenenbaum, Antonio Torralba

***Abstract*— In this work, we study how to build socially intelligent robots to assist people in their homes. In particular, we focus on assistance with online goal inference, where robots must simultaneously infer humans' goals and how to help them achieve those goals. Prior assistance methods either lack the adaptivity to adjust helping strategies (i.e., when and how to help) in response to uncertainty about goals or the scalability to conduct fast inference in a large goal space. Our NOPA (Neurally-guided Online Probabilistic Assistance) method addresses both of these challenges. NOPA consists of (1) an online goal inference module combining neural goal proposals with inverse planning and particle filtering for robust inference under uncertainty, and (2) a helping planner that discovers valuable subgoals to help with and is aware of the uncertainty in goal inference. We compare NOPA against multiple baselines in a new embodied AI assistance challenge: Online Watch-And-Help, in which a helper agent needs to simultaneously watch a main agent's action, infer its goal, and help perform a common household task faster in realistic virtual home environments. Experiments show that our helper agent robustly updates its goal inference and adapts its helping plans to the changing level of uncertainty.**[1]

I. Introduction

There has been growing interest in engineering socially intelligent robots that can safely and productively work with humans in the real world. Prior work on robot assistance has achieved some success in scenarios where robots are given the true human goals a priori or only need to help humans in simple environments with a small state space. However, it remains very challenging to build robot assistants that can help humans perform all the activities of daily life in more natural settings, such as in our homes, where the space of human goals is vast and a person's goal at any point in time will not generally be known with certainty.

Our goal here is to build robot assistants that are able to help people perform a wide range of tasks in complex home environments. Our robot assistants must have the ability to infer the true goals of humans based on past observations in an online fashion, plan how to help humans without disrupting them, and adapt to their behaviors by simultaneously updating goal inference and helping strategies as the task progresses (as illustrated in Fig. 1). Such ability has proven difficult for robots to date due to two main technical obstacles. On the one hand, online goal inference in realistic environments is extremely difficult due to large state, action, and goal spaces; on the other hand, inaccurate or ambiguous goal inferences often lead to ineffective or even counterproductive attempts to help in systems that are not aware of their own uncertainty.

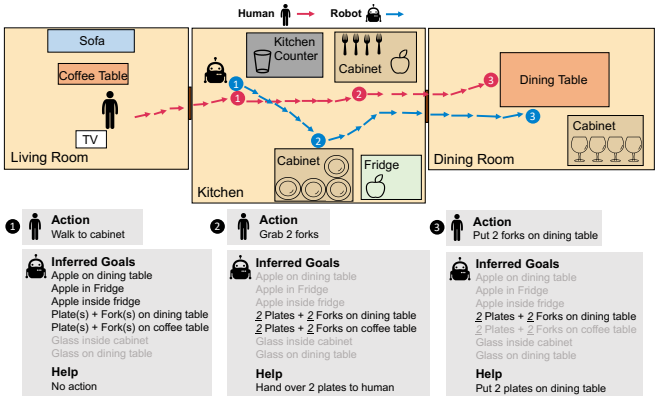

Fig. 1: Illustration of successful online assistance. The robot initially has no knowledge about the human's goal and thus would opt to observe. As it observes more human actions, it becomes more and more confident in its goal inference, so it would dynamically adjust its helping subgoal. For instance, in this figure, the robot first sees the human walking towards a cabinet and consequently infers that the goal involves objects inside the cabinet. After the human grabs 2 forks, the robot infers that the goal is to put 2 sets of dining pieces (plates and forks) on the dining table or the coffee table but is uncertain about the goal location. Thus, it hands over 2 plates to the human instead of randomly guessing a location.

To address these challenges, we propose a novel online assistance method, NOPA (Neurally-guided Online Probabilistic Assistance). As illustrated in Fig. 2**a**, NOPA consists of two main components: (1) a neurally-guided online goal inference module and (2) an uncertainty-aware helping planner. The neurally-guided online goal inference module first produces bottom-up goal proposals from a neural network and then maintains a set of predictions of goals and future trajectories consistent with the observed actions via particle filtering and inverse planning. This ensures that inferences are both fast and robust. Given the latest predictions and their certainty, the helping planner first identifies a subgoal that is most valuable to help with and then plans the corresponding helping actions using a symbolic planner. The resulting helping plan can adapt to all levels of uncertainty in the predictions. For instance, when there are multiple possible target locations for a goal object, the robot assistant will deliver the object to the human agent instead of risking misplacing the object.

For evaluation, we present a new embodied AI assis-

* Equal contribution. All authors are with MIT. {xpuig, tshu, jbt, torralba}@mit.edu

[1]Code is available at https://github.com/xavierpuigf/online_watch_and_help. Project website: https://www.tshu.io/online_watch_and_help.

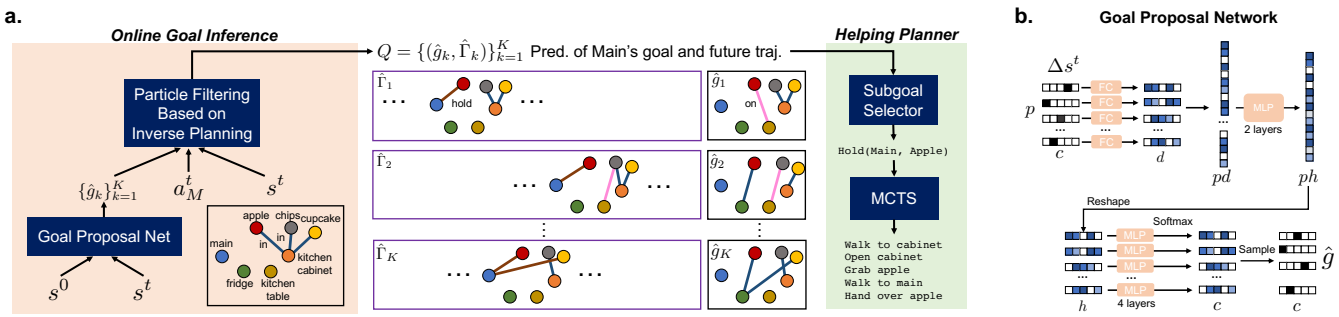

Fig. 2: (**a**) Overview of our approach, which consists of an online goal inference module and a helping planner. We represent states and goals using scene graphs. s^t is the state at time t; a^t_{M} is the main agent's action at time t; $\hat{g}_k$ is the k-th goal proposal; and $\hat{\Gamma}_k$ is the prediction of the main agent's future trajectory corresponding to $\hat{g}_k$. (**b**) Goal proposal network. Δs^t is a matrix encoding the difference between the predicate counts in the states s^t and s^0, where each row is a one-hot vector, indicating the change in the count of a specific predicate. $p = 136$ is the number of all predicate types, $c = 9$ is the maximum number of counts, and $d = 100$ and $h = 128$ are the dimensions of the intermediate layers.

tance challenge, Online Watch-And-Help (O-WAH), which builds off a recent multi-agent virtual home environment, VirtualHome-Social. Unlike most existing challenges (e.g., [1], [2]), in O-WAH, a helper agent needs to infer the goal of a main agent in an online fashion and simultaneously help achieve the inferred goal as efficiently as possible. We evaluate agents built with NOPA and several baselines in a range of household tasks, helping the main agent controlled by either a human player or a planner-based agent. The experimental results show that NOPA significantly outperforms all baselines. We are also able to observe intelligent helping strategies emerging from NOPA adapting to new observations.

In sum, our main contribution includes (1) a new online assistance method for building socially intelligent home assistants in complex settings and (2) an embodied AI assistance challenge, Online Watch-And-Help, as a testbed for training and testing embodied agents to perform online goal inference and helping in realistic virtual home environments.

II. Related Work

Online goal inference. Online goal inference approaches generally fall into two categories – (1) feedforward prediction that directly maps observed past trajectories to possible goals, typically enabled by goal prediction networks [3]–[11], or (2) generative approaches such as inverse planning [12]–[19] which conduct inference by comparing generated plans of given goal hypotheses with observed actions. Feedforward methods are fast and can perform well in simple tasks (such as destination prediction for pedestrians [5]) when trained with a large amount of data. However, in unfamiliar scenarios, inverse planning methods often outperform feedfoward prediction due to their ability to imagine rational behaviors under various conditions. One of the limitations of inverse planning methods is that they can be slow if the goal space is large and rely on manually designed heuristics to speed up the inference [17], [18]. Our work integrates these two types of approaches to achieve both speed and robustness.

Embodied AI assistance with unknown goals. There has been a rich history of research on embodied AI assistance. Many of the existing works assume a known common goal shared among human and AI/robot partners [1], [20]–[27]. However, in the real world, robots often need to infer humans' goals on the fly. There has been work on helping with inferred goals [2], [28]–[32] which shows that an accurate goal inference can improve the objective and perceived performance of embodied AI assistants. However, when the goal inference is uncertain, helping with inferred goals often leads to counterproductive behaviors such as undoing finished goals [2]. For this, some prior work devised planners under uncertain goal inference in simple environments [33]. A recent study proposed a goal-agnostic assistance framework via empowerment [34], which aims at changing the states to maximize an agent's ability to reach as many goals as possible regardless of its true intent. Despite its success in certain domains, assisting humans in real-world settings without the knowledge of their goals would often result in counterproductive behaviors. Our work investigates how to design an uncertainty-aware planner that intelligently adjusts the helping behavior ranging from goal-agnostic strategies to goal-specific plans in a complex environment.

Embodied AI assistance benchmarks. There have been benchmarks designed to evaluate AI agents' ability to collaborate with human teammates [1], [25]. However, most of them focus on simple game environments and assume a common goal given to both the AI and human agents a priori. Based on a realistic virtual home environment, [2] proposed a challenge, Watch-And-Help, in which a helper agent must infer a main agent's goal from a pre-recorded demonstration, and then help the main agent to achieve the same goal. We extend this challenge to an online assistance setting, where the demonstration is no longer available. As a result, helper agents have to pay attention to the main agent's actions and constantly update the inference and its uncertainty while working towards inferred goals, which is closer to what robots are expected to do in real homes. Such benchmark complements conventional robot assistance studies conducted in lab environments [20], [21], providing a reproducible and

Algorithm 1 NOPA

Input: $\Gamma_{\text{M}}^0 = \{(s^0, a_{\text{M}}^0)\}$, s^t, K, $T_{\max}$, T_{prop}, q, w_r, w_c, w_m, $L_{\max}$
$t \leftarrow 1$, $l \leftarrow 0$
$Q \leftarrow \emptyset$
repeat
  $Q, l \leftarrow$ **GoalInf**$(t, Q, \Gamma_{\text{M}}^{t-1}, s^t, q, K, l, T_{\text{prop}})$
  $\Gamma_{\text{H}}^t \leftarrow$ **HelpPlanner**$(Q, s^0, s^t, w_r, w_c, w_m, L_{\max})$
  Execute the first action from the helping plan a_{H}^t
  Observe a_{M}^t, s^{t+1} from the environment
  $t \leftarrow t + 1$
until $t = T_{\max}$ or the true goal has not been reached

Algorithm 2 GoalInf

Input: t, Q, $\Gamma_{\text{M}}^{t-1} = \{(s^\tau, a_{\text{M}}^\tau)\}_{\tau=0}^{t-1}$, s^t, q, K, l, T_{prop}
Output: Updated proposals Q' and steps since last proposal l'
$Q' \leftarrow \emptyset$
if $Q \neq \emptyset$ and $l < T_{\text{prop}}$ **then**
  for $k = 1, \cdots, |Q|$ **do**
    if a_{M}^{t-1} is part of the plan $\hat{\Gamma}_k$ **then**
      $Q' \leftarrow Q' \cup \{(\hat{g}_k, \hat{\Gamma}_k)\}$
    end if
  end for
end if
if $Q' = \emptyset$ **then**
  for $k = 1, \cdots, K$ **do**
    $\hat{g}_k \sim q(g|s^0, s^t)$ // Sample a goal proposal
    $\hat{\Gamma}_k \leftarrow$ **MCTS**$(s^t, \hat{g}_k, T_{\text{prop}})$ // Sample a plan
    $Q' \leftarrow Q' \cup \{(\hat{g}_k, \hat{\Gamma}_k)\}$
  end for
  $l' \leftarrow 0$
else
  $l' \leftarrow l + 1$
end if
return Q', l'

scalable way to compare different methods.

III. Method

A. Problem Setup

We define the online assistance problem as a mixed-observability Markov decision process (MOMDP) [35], where a ***helper*** agent needs to infer a ***main*** agent' goal and help the main agent achieve its goal faster. This can be formalized by $\langle \mathcal{S}, \mathcal{G}, \mathcal{A}_{\text{H}}, \mathcal{O}, \mathcal{T}_{\mathcal{S}}, \mathcal{T}_{\mathcal{G}}, Z, R_{\text{H}}, \gamma \rangle$. The overall state has two components: the world state, $s \in \mathcal{S}$, which is fully observable to both agents, and the main agent's goal, $g \in \mathcal{G}$, which is partially observable to the helper agent. $\mathcal{A}_{\text{H}}$ is the action space of the helper agent. The helper agent's observation consists of the world state and the main agent's action, i.e., $\mathcal{O} = \mathcal{S} \times \mathcal{A}_{\text{M}}$. $\mathcal{T}_{\mathcal{S}}(s, g, a_{\text{H}}, s') = p(s'|s, g, a_{\text{H}})$ is the transition function for the world state, and $\mathcal{T}_{\mathcal{G}}(s, g, a_{\text{H}}, s', g') = p(g'|s, g, a_{\text{H}}, s')$ is the transition function for the goal. $Z(s', g', a_{\text{H}}, o) = p(o = (s, a_{\text{M}})|s', g', a_{\text{H}})$ is the conditional probability function for the observation result. At step $t + 1$, the helper infers the main's goal given main's past trajectory upon t, i.e., $\Gamma^t = \{(s^\tau, a_{\text{M}}^\tau)\}_{\tau=1}^t$. The expected reward function for the helper agent is defined as $R_{\text{H}}(s, a|\Gamma^t) = E_{p(g|\Gamma^t)}[\mathbb{1}(s = g)] - c_{\text{H}}(a)$, where $c_{\text{H}}(a)$ is the cost for action a, and $\mathbb{1}(\cdot)$ checks if the goal is satisfied in the current world state s. γ is the discount factor. Note that the assumption that the world states are fully observable for both agents is common in prior work on assistance with unknown goals [29], [31], [32].

B. Method Overview

To solve the online assistance problem formalized above, we propose NOPA (Neurally-guided Online Probabilistic assistance). Fig. 2**a** provides an overview of NOPA, showing the two main components: i) Neurally-guided online goal inference, and ii) an uncertainty-aware helping planner. As sketched in Algorithm 1, NOPA updates a set of particles conditioned on observed states and the main agent's actions. Each particle corresponds to a possible final goal. Common assistance frameworks [2], [29], [31], [32] typically only consider the final goal for helping. However, when there is uncertainty in the goal inference, an intelligent assistant should seek intermediate subgoals that can be helpful with high certainty. For that, we also predict the main agent's future trajectory for each particle. We represent both intermediate states and final goals as a set of edges in a scene graph [36], [37], $\langle O, E \rangle$, as shown in Fig. 2**a**. Each node, $o \in O$, represents an entity (agent/object); each edge, $e \in E$, corresponds to a predicate (e.g., `IN(apple, kitchencabinet)`), indicating the spatial relationship between two entities. Such representations have been widely adopted in robotics and embodied AI [38]–[41]. Given the particles, the helping planner assesses the value of the edges in the intermediate states and the final goals and selects the most valuable edge as the helping subgoal. We introduce the details below.

C. Neurally-guided Online Goal Inference

Unlike prior work on online goal inference, the objective of the online goal inference in this work is to help the downstream task, i.e., assistance. This poses additional challenges: (1) the helper agent has to estimate the uncertainty in the inference instead of only predicting the most probable goal; (2) it has to ensure that the inference is resilient in a dynamic environment; and (3) the inference has to be efficient so that the helper can have a prompt reaction to offer assistance. For this, we propose a neurally-guided online goal inference algorithm as summarized in Algorithm 2, which combines inverse planning and a neural network.

We use a goal proposal network (GPN), as depicted in Fig. 2**b**, to learn a proposal distribution $q(g|s^0, s^t)$, from which we can sample K goal proposals $\{\hat{g}_k\}_{k=1}^K$ given the initial state s^0 and the current state s^t. Each goal proposal is a set of goal predicates. Here, we only consider the first state and the current state instead of a sequence of past states for the input to GPN so that the GPN trained on episodes with only the main agent performing the tasks can be robustly applied to the helping condition where the sequence of state changes could become very different from the training sequences.

We use inverse planning to evaluate the goals proposed by the GPN and reject the ones that are inconsistent with the observed main agent's actions. To model an agent's behavior given each goal $\hat{g}_k$ with bounded rationality, we use the built-in planner to predict the future trajectory of the main agent in the next T_{prop} steps $\hat{\Gamma}_k = \{(\hat{s}^{t+\tau_k}, \hat{a}^{t+\tau_k})\}_{\tau=1}^{T_{\text{prop}}}$. Specifically, the level of rationality can be adjusted by the number of

Algorithm 3 HelpPlanner

Input: Q, s^0, s^t, w_r, w_c, w_m, $L_{\max}$
Output: helping plan Γ_{H}^t
for $e \in \mathcal{E}$ **do**
  $L_{\mathrm{M}}(e) \leftarrow \infty$
  $p(e) \leftarrow 0$
  $V(e) \leftarrow -\infty$
  $\Gamma_{\mathrm{H}}(e) \leftarrow \emptyset$
  if e appears in the initial state **then**
    $L_{\mathrm{M}}(e) \leftarrow 0$
    $p(e) \leftarrow 1$
  else
    for $k = 1, \cdots, |Q|$ **do**
      if e appears in the future traj. $\hat{\Gamma}_k$ **then**
        Let $\tau_k(e)$ be the first step when e appears in $\hat{\Gamma}_k$
        $L_{\mathrm{M}}(e) \leftarrow \min(L_{\mathrm{M}}(e), \tau_k(e))$
        $p(e) \leftarrow p(e) + 1/|Q|$
      else if e appears in the predicted goal $\hat{g}_k$ **then**
        $L_{\mathrm{M}}(e) \leftarrow \min(L_{\mathrm{M}}(e), L_{\max})$
        $p(e) \leftarrow p(e) + 1/|Q|$
      end if
    end for
  end if
  if $p(e) > 0$ **then**
    $\Gamma_{\mathrm{H}}(e) \leftarrow$ **MCTS**$(s^t, \{e\}, \infty)$ // Helper's plan for subgoal e
    $L_{\mathrm{H}}(e) \leftarrow |\Gamma_{\mathrm{H}}(e)|$
    Compute $V(s^t, e)$ as Eq (1)
  end if
end for
$e^* \leftarrow \arg\max V(s^t, e)$
return $\Gamma_{\mathrm{H}}(e^*)$

simulations and the length of rollouts. We then create K particles $Q = \{\hat{g}_k, \hat{\Gamma}_k\}$. Whenever the main agent takes a new action, a_{M}^t, we check if it is part of the predicted plan for each particle. If for a particle k, the action is not included in the predicted plan, then it suggests that the rational behavior under the corresponding goal is not consistent with the observed action. Thus the goal is likely to be wrong and the particle needs to be rejected. When there is no particle left or we have reached the prediction horizon T_{prop}, we resample another K goals from the GPN based on the latest state and create new particles. Since some particles may share the same goal but have different predicted plans, our approach can consider different ways to reach a goal.

D. Uncertainty-aware Helping Planner

Finding the optimal helping plan at each step is expensive. To balance the speed and the optimality, our helping planner (Algorithm 3) focuses on finding valuable subgoals instead of updating the whole plan at each step. It considers all edges that appear in the final goal and the intermediate states in the predicted main agent's plans as candidate helping subgoals. Additionally, the helper agent may find that the objects it grabs are no longer needed when it updates goal inference or after the main agent achieves the corresponding subgoals. To allow the helper agent to return those objects to their initial locations, the helping subgoal space also includes edges in the initial state. For each edge e, we estimate how long it would take the main agent to reach that subgoal, $L_{\mathrm{M}}(e)$. If this edge appears in one of the predicted trajectories in the particles, we can conveniently estimate $L_{\mathrm{M}}(e)$ based on when it appears in the trajectories. If it only appears in the final goals, we then use a fixed length, $L_{\max}$, to anticipate how many steps it would take the main agent to reach that subgoal. We can also use MCTS to search for a plan for the helper agent, $\Gamma_{\mathrm{H}}(e)$, to reach the same subgoal. Let $L_{\mathrm{H}}(e)$ be the length of the helper's plan, we then define the benefit of helping with subgoal e as the speed up the helper agent can offer by reaching the subgoal e, i.e., $\max(L_{\mathrm{M}}(e) - L_{\mathrm{H}}(e), 0)$. To account for the uncertainty in inference, we estimate how likely e is going to be necessary, $p(e)$, by counting how many particles include e in either the intermediate states or the final goal. Finally, we define a value function for selecting the best subgoal for the helper agent:

$$\begin{aligned} V(s^t, e) = \quad & w_r p(e) |L_{\mathrm{M}}(e) - L_{\mathrm{H}}(e)|_+ - w_c L_{\mathrm{H}}(e) \\ & - w_m (\mathcal{D}(s^0, \hat{s}(e)) - \mathcal{D}(s^0, s^t)), \end{aligned} \tag{1}$$

where w_r, w_c, and w_m are constant weights; $\mathcal{D}$ measures the difference between two states; and $\hat{s}(e)$ is the state after reaching the subgoal e from the current state s^t. The three terms in Eq. (1) evaluate i) the expected benefit of helping reach the subgoal, ii) the cost of the helper agent, and iii) the additional state change (compared with the initial state) introduced by the subgoal. These three terms make sure that the helper agent selects a subgoal that i) is likely to speed up the task with high certainty, ii) is not too costly for the helper agent, and iii) could restore the initial states of objects that are not needed for the task respectively. Given the subgoal e^*, we execute the first action of the helping plan $\Gamma_{\mathrm{H}}(e^*)$.

IV. Online Watch-And-Help

To evaluate different assistance methods, we propose Online Watch-And-Help (O-WAH), an embodied AI assistance challenge, in which a helper agent has to infer a main agent's goal and help reach the goal as fast as possible. This extends an existing challenge, Watch-And-Help (WAH), to an online assistance problem. O-WAH is built in a realistic multi-agent virtual platform, VirtualHome-Social [2], simulating daily household tasks (as shown in Fig. 3). The goal for each task is defined by a set of predicates and their counts, representing the target locations of different objects in the environment. We sample each goal in the challenge from five general types of household tasks: *set table*, *put dishwasher*, *stock fridge*, *prepare meal*, and *get snacks*. Note that we define task types only to ensure that the goals are emulating real-life household tasks, but that this information is not provided to the helper agents. As summarized in Table I, different kinds of uncertainty may arise from these tasks: i) uncertainty in the number of objects, ii) uncertainty in which objects are needed, and iii) uncertainty in the target locations. Compared to prior work, the goal space in O-WAH is 1 or 2 orders of magnitude larger. We adopt the same action space in [2].

To create a training episode, we first sample a goal and an initial environment using one of the five training apartments and then use a built-in planner to control the main agent to perform the task alone. The built-in planner is the same hierarchical planner as in [2]. We create a large training set with 6,000 episodes and a small training set with 300

Fig. 3: An example setup of O-WAH in one of the simulated apartments.

TABLE I: The goal definition and the number of unique goals for each task type.

Task Name	Goal definition	#Goals
Set table	Put `N plate`, `N fork`, `N OBJ` on `LOC`, where `N` $\sim$ `U(1,3)`, `OBJ` $\sim$ `choice([waterglass, wineglass])`, `LOC` $\sim$ `choice([kitchentable, coffeetable])`	12
Put dishwasher	Put `N` objects from `OBJ_POOL` in `dishwasher`, where `N` $\sim$ `U(3,7)`, `OBJ_POOL = [fork, plates, waterglass, wineglass]`	315
Stock fridge	Put `N` objects from `OBJ_POOL` in `fridge`, where `N` $\sim$ `U(3,7)`, `OBJ_POOL = [salmon, apple, cupcake, pudding]`	315
Prepare meal	Put `N salmon`, `N apple`, `N OBJ` on `LOC`, where `N` $\sim$ `U(1,3)`, `OBJ` $\sim$ `choice([cupcake, pudding])`, `LOC` $\sim$ `choice([kitchentable, coffeetable, stove])`	18
Get snacks	Put `1 remote`, `1 condiment`, `1 chips` on `coffeetable`	1

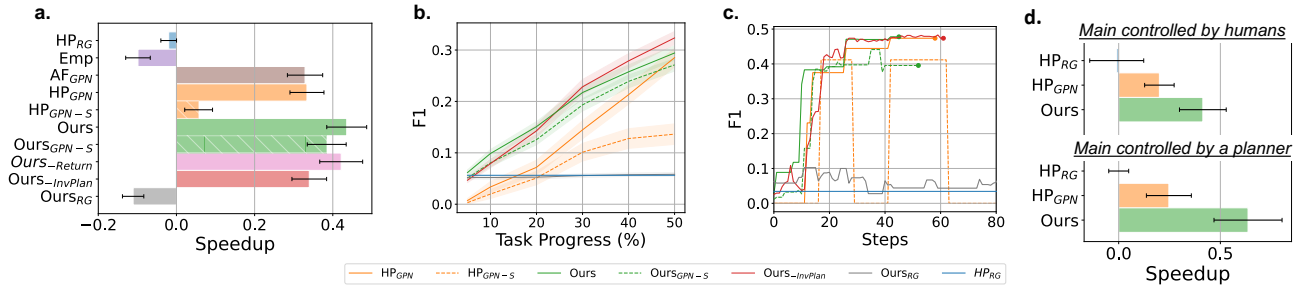

Fig. 4: (**a**) Speedup of different methods (striped bars indicate using the small training set). Errors are standard errors. (**b**) F1-scores of the predicted goal over the course of a task. The x axis is normalized in proportion to the number of steps needed for the main agent to perform each task alone. The curves show the means and the shaded regions show the standard errors. **c)** F1-scores over time for different approaches in a single test episode, a dot indicates the number of steps a given baseline took to complete the task. The dashed lines in (**b**) and (**c**) indicate using the small training set. (**d**) Results of the human experiment. Here we show the speedup of different methods when the main agent is controlled by the built-in planner or by human players (note that the results under the two conditions are based on the same 10 testing episodes).

episodes. The testing set has 100 episodes in the two testing apartments unseen during training.

We use F1-score over the goal predicates to measure the goal inference accuracy. To evaluate the helping performance, we use speedup, where we compare the episode length when the helper agent works with the main agent (L_{H}) against the episode length when the main agent works alone (L_{M}), i.e., $L_{\text{M}}/L_{\text{H}} - 1$. For each episode, set a time limit of 250 steps and report the average performance across 3 runs.

V. Experiments

A. Baselines

We compare NOPA against several baselines.

HP$_{\text{GPN}}$: We adopt the best performing approach in the original Watch-And-Help challenge [2] for this baseline, which is a hierarchical planner (HP) based on the most probable goal according to the GPN. In particular, at each step, **HP$_{\text{GPN}}$** uses the goal $\hat{g} = \arg\max_g q(g|s^0, s^t)$.

AF$_{\text{GPN}}$: We extend **HP$_{\text{GPN}}$** by using NOPA's online goal inference. We generate a plan for each predicated goal using HP and execute the most frequent first action among all plans.

Empowerment: By adopting the idea of empowerment [34], this baseline uniformly samples K goals at each step, predicts plans and intermediate states for the goals, and selects the most frequent edge in the intermediate state as the helping subgoal (i.e., the most common subgoal for *any* goal).

HP$_{\text{RG}}$: A hierarchical planner based on a randomly sampled goal at the beginning of the episode.

We consider the following ablated methods to evaluate the effect of different components of NOPA.

Ours$_{\text{RG}}$: We replace the proposal distribution q in Algorithm 2 with a uniform distribution.

Ours$_{\text{-InvPlan}}$: Ours without inverse planning.

Ours$_{\text{-Return}}$: Ours without returning irrelevant objects to their initial locations ($w_m = 0$ in Eq.(1)).

By default, the GPN is trained on the large training set. To evaluate the sample efficiency of NOPA, we also report the performance of **Ours** and **HP$_{\text{GPN}}$**, when the GPN is trained on a small training set, indicated by the subscript **GPN-S**. To measure the upper bound on the helping performance, we also implement an oracle helper **HP$_{\text{GT}}$**, which knows the ground-truth goal and is controlled by an HP.

We set $T_{\text{max}} = 250$, $T_{\text{prop}} = 15$, $w_r = 1$, $w_c = 1$, $w_d = 5$, and $L_{\text{max}} = 100$ for NOPA. For all approaches that propose multiple goals, we use $K = 20$ proposals. We train the GPN using Adam [42] with a learning rate of .0009 and a batch size of 256.

B. Results

1) Main Controlled by a Planner: We evaluate all methods with a main agent controlled by the built-in planner and report the helping speedup (average and standard error across episodes) in Fig. 4**a**. For methods that have different goal inference modules, we also report the F1-score of their goal inference results in Fig. 4**b**. The speedup of the oracle agent, operating with true knowledge about the goal, **HP$_{\text{GT}}$** is 1.29.

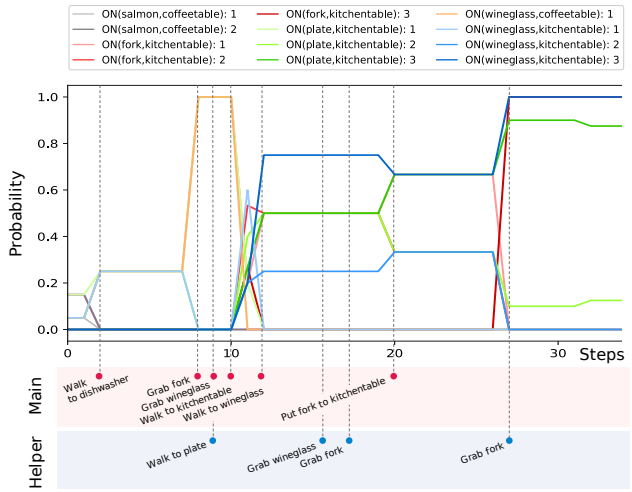

Fig. 5: Goal inference and plans by NOPA for the same task shown in Fig. 4**c**, which is setting up a kitchen table for 3 persons. We show the posterior probabilities of the top predicates and their counts based on the particles at each step, key actions of the main agent (indicated by red dots), and key helping actions (indicated by blue dots). At step 2, after watching the main agent walking towards the dishwasher, NOPA rejects proposals involving nearby objects (e.g., apples, salmons) that are not inside of the dishwasher. After the main agent grabs a fork at step 8, NOPA infers with high confidence that the goal is setting up a table for at least one person. So at the following step, the helper agent takes its very first action – walking to grab a plate. Upon seeing Main walking to the kitchen table at step 10, the goal location becomes certain. After observing more actions, the inference converges to setting up the kitchen table for 3 persons.

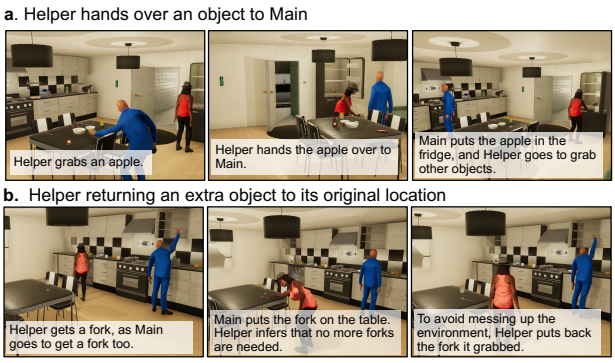

Fig. 6: Examples of helping plans that are beyond directly achieving final goals. Main is in red, and Helper is in blue.

NOPA (**Ours**) outperforms all baselines, offering the highest speedup. It also achieves the best goal inference accuracy at the early stage of the tasks, which serves as the foundation of its successful assistance. This benefit can be more clearly seen from Fig. 4**c** (the improvement margin appears to be smaller since the temporal normalization for each episode is different). The low speedup by **Empowerment** suggests that online goal inference is necessary for effective assistance, despite its success in certain domains shown in prior work [34]. Given that the predicted goal may be uncertain, using multiple goal proposals leads to a better helping performance, as seen by comparing $\textbf{Ours}_{\textbf{GPN}}$ with $\textbf{HP}_{\textbf{GPN}}$. The effect is more pronounced when the GPN is trained with fewer data and is consequently less accurate (**GPN-S**). We also find that the neurally-guided goal proposals can greatly improve the goal inference over uniform goal proposals ($\textbf{Ours}_{\textbf{RG}}$). Moreover, the results demonstrate that inverse planning is important for filtering spurious goal proposals from **GPN**, significantly improving the speedup over $\textbf{Ours}_{\textbf{-InvPlan}}$ since it allows the goal inference to reach a relatively high accuracy much earlier than $\textbf{Ours}_{\textbf{-InvPlan}}$ and other baselines do (see Fig. 4**c**). Finally, by comparing NOPA with $\textbf{Ours}_{\textbf{-Return}}$, we can see a marginal improvement in speedup by avoiding unnecessarily distorting the environment; $\textbf{Ours}_{\textbf{-Return}}$ also causes 11.2% more unnecessary state changes.

Fig. 5 shows a typical successful example by NOPA, where the task is the same as the one in Fig. 4**c**. It demonstrates that NOPA can i) achieve accurate goal inference early on by filtering out goal proposals that are inconsistent with the main agent's actions, ii) correctly update the goal inference and its uncertainty estimation based on more observation, and iii) plan for effective helping actions based on the filtered goal proposals and the uncertainty in the inference. Note that the helper remains idle but takes a useful helping action as soon as the goal inference becomes confident and is able to avoid grabbing extra objects thanks to its gradual update of the number of objects needed.

We also observe diverse helping behaviors enabled by NOPA that are not just about directly achieving the final goals as shown in Fig. 6. First, the helper agent sometimes selects a subgoal of handing over objects to the main agent. For example, as illustrated in Fig. 6**a**, the helper agent hands over the apple to the main agent who is right next to the fridge so that the task execution can be faster. Second, the helper agent can return extra objects to their initial locations once it realizes that they are not needed for reaching the goal (Fig. 6**b**). The supplementary video[2] shows more examples.

2) Main Controlled by Humans: To evaluate how effective helper agents are at assisting real humans, we conducted a human experiment where the main agent is controlled by human players. We used 10 testing episodes to run 40 trials. In each trial, a human participant was asked to either perform the task alone (to estimate the number of steps needed for completing each task alone) or work with a helper agent controlled by one of the three approaches, NOPA, $\textbf{HP}_{\textbf{GPN}}$, and $\textbf{HP}_{\textbf{RG}}$. Participants did not know which helper agent they were working with. We recruited 10 participants (mean age = 32.3; 4 female) who had no prior exposure to our system. As shown in Fig. 4(**d**), the ranking of the methods remains consistent when the main agent is controlled by human players. There is no significant difference in NOPA's performance under the two conditions ($t(9) = 0.87$, $\rho = 0.40$).

[2]The supplementary video is available at https://youtu.be/Oawo9pynPL0.

VI. Conclusion

In this work, we propose a novel method for building socially intelligent home assistants, Neurally-guided Online Probabilistic Assistance (NOPA), which integrates (1) a hybrid online goal inference algorithm combining a goal proposal network and inverse planning and (2) an uncertainty-aware helping planner that identifies valuable helping subgoals from both the final goals and intermediate states. For a systematic and scalable evaluation, we introduce a new embodied AI assistance challenge, Online Watch-And-Help, based on a realistic virtual home platform. We evaluate NOPA with several baselines in our challenge with a main agent controlled either by a built-in planner or by humans. Our experiments show that NOPA significantly outperforms baselines and achieves great sample efficiency for training. In the future, we plan to extend NOPA to a partial observability setting and apply it to robots in real homes.

ACKNOWLEDGMENT

This work was supported by the DARPA Machine Common Sense program, ONR MURI N00014-13-1-0333, and a grant from Lockheed Martin.

References

[1] M. Carroll, R. Shah, M. K. Ho, T. Griffiths, S. Seshia, P. Abbeel, and A. Dragan, "On the utility of learning about humans for human-AI coordination," *Advances in neural information processing systems*, vol. 32, 2019.

[2] X. Puig, T. Shu, S. Li, Z. Wang, Y.-H. Liao, J. B. Tenenbaum, S. Fidler, and A. Torralba, "Watch-And-Help: A challenge for social perception and human-AI collaboration," in *International Conference on Learning Representations*, 2021.

[3] N. Rabinowitz, F. Perbet, F. Song, C. Zhang, S. A. Eslami, and M. Botvinick, "Machine theory of mind," in *International conference on machine learning*. PMLR, 2018, pp. 4218–4227.

[4] Z. Cao, H. Gao, K. Mangalam, Q.-Z. Cai, M. Vo, and J. Malik, "Long-term human motion prediction with scene context," in *European Conference on Computer Vision*. Springer, 2020, pp. 387–404.

[5] B. Liu, E. Adeli, Z. Cao, K.-H. Lee, A. Shenoi, A. Gaidon, and J. C. Niebles, "Spatiotemporal relationship reasoning for pedestrian intent prediction," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3485–3492, 2020.

[6] Z. Nan, T. Shu, R. Gong, S. Wang, P. Wei, S.-C. Zhu, and N. Zheng, "Learning to infer human attention in daily activities," *Pattern Recognition*, vol. 103, p. 107314, 2020.

[7] P. Dendorfer, A. Osep, and L. Leal-Taixé, "Goal-GAN: Multimodal trajectory prediction based on goal position estimation," in *Proceedings of the Asian Conference on Computer Vision*, 2020.

[8] H. Zhao, J. Gao, T. Lan, C. Sun, B. Sapp, B. Varadarajan, Y. Shen, Y. Shen, Y. Chai, C. Schmid, *et al.*, "TNT: Target-driven trajectory prediction," *arXiv preprint arXiv:2008.08294*, 2020.

[9] Y. Yao, E. Atkins, M. Johnson-Roberson, R. Vasudevan, and X. Du, "BITRAP: Bi-directional pedestrian trajectory prediction with multi-modal goal estimation," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1463–1470, 2021.

[10] H. Tran, V. Le, and T. Tran, "Goal-driven long-term trajectory prediction," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 796–805.

[11] K. Mangalam, Y. An, H. Girase, and J. Malik, "From goals, waypoints & paths to long term human trajectory forecasting," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 233–15 242.

[12] S. Carberry, "Techniques for plan recognition," *User modeling and user-adapted interaction*, vol. 11, no. 1, pp. 31–48, 2001.

[13] M. Ramírez and H. Geffner, "Plan recognition as planning," in *Twenty-First international joint conference on artificial intelligence*, 2009.

[14] S. Sohrabi, A. V. Riabov, and O. Udrea, "Plan recognition as planning revisited." in *IJCAI*, 2016, pp. 3258–3264.

[15] C. L. Baker, J. Jara-Ettinger, R. Saxe, and J. B. Tenenbaum, "Rational quantitative attribution of beliefs, desires and percepts in human mentalizing," *Nature Human Behaviour*, vol. 1, no. 4, pp. 1–10, 2017.

[16] T. Ullman, C. Baker, O. Macindoe, O. Evans, N. Goodman, and J. Tenenbaum, "Help or hinder: Bayesian models of social goal inference," *Advances in neural information processing systems*, vol. 22, 2009.

[17] T. Zhi-Xuan, J. Mann, T. Silver, J. Tenenbaum, and V. Mansinghka, "Online bayesian goal inference for boundedly rational planning agents," *Advances in Neural Information Processing Systems*, vol. 33, pp. 19 238–19 250, 2020.

[18] A. Netanyahu, T. Shu, B. Katz, A. Barbu, and J. B. Tenenbaum, "PHASE: Physically-grounded abstract social events for machine social perception," in *35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.

[19] R. Tejwani, Y.-L. Kuo, T. Shu, B. Katz, and A. Barbu, "Social interactions as recursive mdps," in *Conference on Robot Learning*. PMLR, 2022, pp. 949–958.

[20] M. A. Goodrich and A. C. Schultz, "Human-robot interaction: a survey," *Foundations and trends in human-computer interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[21] P. A. Lasota, T. Fong, J. A. Shah, *et al.*, *A survey of methods for safe human-robot interaction*. Now Publishers Delft, The Netherlands, 2017, vol. 104.

[22] S. Nikolaidis, R. Ramakrishnan, K. Gu, and J. Shah, "Efficient model learning from joint-action demonstrations for human-robot collaborative tasks," in *2015 10th ACM/IEEE International Conference on Human-Robot Interaction (HRI)*. IEEE, 2015, pp. 189–196.

[23] L. Rozo, S. Calinon, D. G. Caldwell, P. Jimenez, and C. Torras, "Learning physical collaborative robot behaviors from human demonstrations," *IEEE Transactions on Robotics*, vol. 32, no. 3, pp. 513–527, 2016.

[24] M. Jaderberg, W. M. Czarnecki, I. Dunning, L. Marris, G. Lever, A. G. Castaneda, C. Beattie, N. C. Rabinowitz, A. S. Morcos, A. Ruderman, *et al.*, "Human-level performance in 3d multiplayer games with population-based reinforcement learning," *Science*, vol. 364, no. 6443, pp. 859–865, 2019.

[25] N. Bard, J. N. Foerster, S. Chandar, N. Burch, M. Lanctot, H. F. Song, E. Parisotto, V. Dumoulin, S. Moitra, E. Hughes, *et al.*, "The hanabi challenge: A new frontier for AI research," *Artificial Intelligence*, vol. 280, p. 103216, 2020.

[26] H. Hu, A. Lerer, A. Peysakhovich, and J. Foerster, ""other-play" for zero-shot coordination," in *International Conference on Machine Learning*. PMLR, 2020, pp. 4399–4410.

[27] S. A. Wu, R. E. Wang, J. A. Evans, J. B. Tenenbaum, D. C. Parkes, and M. Kleiman-Weiner, "Too many cooks: Bayesian inference for coordinating multi-agent collaboration," *Topics in Cognitive Science*, vol. 13, no. 2, pp. 414–432, 2021.

[28] A. Fern, S. Natarajan, K. Judah, and P. Tadepalli, "A decision-theoretic model of assistance," *Journal of Artificial Intelligence Research*, vol. 50, pp. 71–104, 2014.

[29] C. Liu, J. B. Hamrick, J. F. Fisac, A. D. Dragan, J. K. Hedrick, S. S. Sastry, and T. L. Griffiths, "Goal inference improves objective and perceived performance in human-robot collaboration," in *15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2016.

[30] S. V. Albrecht and P. Stone, "Autonomous agents modelling other agents: A comprehensive survey and open problems," *Artificial Intelligence*, vol. 258, pp. 66–95, 2018.

[31] D. Hadfield-Menell, S. J. Russell, P. Abbeel, and A. Dragan, "Cooperative inverse reinforcement learning," *Advances in neural information processing systems*, vol. 29, 2016.

[32] S. Javdani, H. Admoni, S. Pellegrinelli, S. S. Srinivasa, and J. A. Bagnell, "Shared autonomy via hindsight optimization for teleoperation and teaming," *The International Journal of Robotics Research*, vol. 37, no. 7, pp. 717–742, 2018.

[33] A. Shah, S. Li, and J. Shah, "Planning with uncertain specifications (PUnS)," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3414–3421, 2020.

[34] Y. Du, S. Tiomkin, E. Kiciman, D. Polani, P. Abbeel, and A. Dragan, "AVE: Assistance via empowerment," *Advances in Neural Information Processing Systems*, vol. 33, pp. 4560–4571, 2020.

[35] S. C. Ong, S. W. Png, D. Hsu, and W. S. Lee, "Planning under uncertainty for robotic tasks with mixed observability," *The International Journal of Robotics Research*, vol. 29, no. 8, pp. 1053–1068, 2010.

[36] J. Johnson, R. Krishna, M. Stark, L.-J. Li, D. Shamma, M. Bernstein, and L. Fei-Fei, "Image retrieval using scene graphs," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3668–3678.

[37] Y.-H. Liao, X. Puig, M. Boben, A. Torralba, and S. Fidler, "Synthesizing environment-aware activities via activity sketches," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 6291–6299.

[38] F. Yan, D. Wang, and H. He, "Robotic understanding of spatial relationships using neural-logic learning," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 8358–8365.

[39] S. Nguyen, O. S. Oguz, V. N. Hartmann, and M. Toussaint, "Self-supervised learning of scene-graph representations for robotic sequential manipulation planning." in *CoRL*, 2020, pp. 2104–2119.

[40] Y. Zhu, J. Tremblay, S. Birchfield, and Y. Zhu, "Hierarchical planning for long-horizon manipulation with geometric and symbolic scene graphs," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 6541–6548.

[41] S. Srivastava, C. Li, M. Lingelbach, R. Martín-Martín, F. Xia, K. E. Vainio, Z. Lian, C. Gokmen, S. Buch, K. Liu, *et al.*, "Behavior: Benchmark for everyday household activities in virtual, interactive, and ecological environments," in *Conference on Robot Learning*. PMLR, 2022, pp. 477–490.

[42] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.