Human-In-The-Loop Task and Motion Planning for Imitation Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract: Imitation learning from human demonstrations can teach robots complex manipulation skills, but is time-consuming and labor intensive. In contrast, Task and Motion Planning (TAMP) systems are automated and excel at solving long-horizon tasks, but they are difficult to apply to contact-rich tasks. In this paper, we present Human-in-the-Loop Task and Motion Planning (HITL-TAMP), a novel system that leverages the benefits of both approaches. The system employs a TAMP-gated control mechanism, which selectively gives and takes control to and from a human teleoperator. This enables the human teleoperator to manage a fleet of robots, maximizing data collection efficiency. The collected human data is then combined with an imitation learning framework to train a TAMP-gated policy, leading to superior performance compared to training on full task demonstrations. We compared HITL-TAMP to conventional teleoperation system — users gathered more than 3x the number of demos given the same time budget. Furthermore, proficient agents (75%+ success) could be trained from just 10 minutes of non-expert teleoperation data. Finally, we collected 2.1K demos with HITL-TAMP across 12 contact-rich, long-horizon tasks and show that the system often produces near-perfect agents. Videos and additional results at https://sites.google.com/view/corl-2023-hitl-tamp.

Keywords: Imitation Learning, Task and Motion Planning, Teleoperation

1 Introduction

Learning from human demonstrations has emerged as a promising way to teach robots complex manipulation skills [1, 2]. However, scaling up this paradigm to real-world long-horizon tasks has been difficult — providing long manipulation demonstrations is time-consuming and labor intensive [3]. At the same time, not all parts of a task are equally challenging. For example, significant portions of complex manipulation tasks such as part assembly or making a cup of coffee are free-space motion and object transportation, which can be readily automated by non-learning approaches such as motion planning. However, planning methods generally require accurate dynamics models [4] and precise perception, which are often unavailable, limiting their effectiveness at contact-rich and low-tolerance manipulation. In this context, our work aims at solving real-world long-horizon manipulation tasks by combining the benefits of learning and planning approaches.

Our method focuses on augmenting Task and Motion Planning (TAMP) systems, which have been shown to be remarkable at solving long-horizon problems [5]. TAMP methods can plan behavior for a wide range of multi-step manipulation tasks by searching over valid combinations of a small number of primitive skills. Traditionally, each skill is hand-engineered; however, certain skills, such as closing a spring-loaded lid or inserting a rod into a hole, are prohibitively difficult to model in a productive manner. Instead, we use a combination of human teleoperation and closed-loop learning to implement just these select skills, keeping the rest automated. These skills use human teleoperation at data collection time and a policy trained from the data at deployment time. Integrating TAMP

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

systems and human teleoperation poses key technical challenges — special care must be taken to enable seamless handoff between them to ensure efficient use of human time.

To address these challenges, we introduce Human-in-the-Loop Task and Motion Planning (HITL-TAMP), a system that symbiotically combines TAMP with teleoperation. The system collects demonstrations by employing a TAMP-gated control mechanism — it trades off control between a TAMP system and a human teleoperator, who takes over to fill in gaps that TAMP delegates. Critically, human operators only need to engage at selected steps of a task plan when prompted by the TAMP system, meaning that they can manage a fleet of robots by asynchronously engaging with one demonstration session at a time while a TAMP system controls the rest of the fleet.

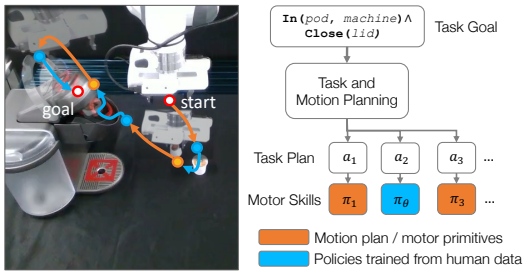


Figure 1: **Overview.** HITL-TAMP decomposes a task (making coffee) into planning-based (TAMP) and learning-based (human imitation) segments.

By soliciting human demonstrations only when needed, and allowing for a human to participate in multiple parallel sessions, our system greatly increases the throughput of data collection while lowering the effort needed to collect large datasets on long-horizon, contact-rich tasks. We combine our data collection system with an imitation learning framework that trains a TAMP-gated policy (as illustrated in Fig. 1) on the collected human data. We show that this leads to superior performance compared to collecting human demonstrations on the entire task, in terms of the amount of data and time needed for a human to teach a task to the robot, and the success rate of learned policies.

The main contributions of this paper are:

• We develop HITL-TAMP, an efficient data collection system for long-horizon manipulation tasks that synergistically combines and trades off control between a TAMP system and a human operator.

• HITL-TAMP contains novel components including (1) a mechanism that allows TAMP to learn planning conditions from a small number of demonstrations and (2) a queuing system that allows a demonstrator to manage a fleet of parallel data collection sessions.

• We conduct a study (15 users) to compare HITL-TAMP with a conventional teleoperation system. Users collected over 3x more demos with our system given the same time budget. Proficient agents (over 75% success) could be trained from just 10 minutes of non-expert teleoperation data.

• We collected 2.1K demos with HITL-TAMP across 12 contact-rich and long-horizon tasks, including real-world coffee preparation, and show that HITL-TAMP often produces near-perfect agents.

2 Preliminaries

Summary of Related Work. Several works have shown the value in learning robot manipulation with human demonstrations [6, 1, 7, 8, 9, 10, 11, 12, 9], in developing automatic control hand-offs between a human supervisor and an automated system for more effective data collection [13, 14, 15, 16, 17], and in combining learned and predefined skills [18, 19, 20, 21, 22]. Prior TAMP [5, 23, 4, 24] works have also integrated learning-based components [25, 26, 27, 28, 29, 30, 31, 32, 33] to make less assumptions on prior knowledge. **See Appendix D for full related work.**

Problem Statement. We consider a robot acting in a discrete-time Markov Decision Process (MDP) $\langle \mathcal{X}, \mathcal{U}, \mathcal{T}(x' \mid x, u), \mathcal{R}(x), \mathcal{P}_0 \rangle$ defined by state space $\mathcal{X}$, action space $\mathcal{U}$, transition distribution $\mathcal{T}$, reward function $\mathcal{R}$, and initial state distribution $\mathcal{P}_0$. We assume we are given an offline dataset of N partial demonstration trajectories (collected via our HITL-TAMP system, see Sec. 3.3) $\mathcal{D} = \{\langle x_0^i, u_0^i \rangle, \langle x_1^i, u_1^i \rangle, ..., x_{T^i}^i\}_{i=1}^N$. We train policies π with Behavioral Cloning [34] using the objective $\arg\min_\theta \mathbb{E}_{(x,u)\in\mathcal{D}} ||\pi_\theta(x) - u||^2$ (details in Appendix J).

We consider a TAMP policy $\pi_t(u \mid x)$ for controlling the robot. It plans a sequence of actions that will be tracked using a high-frequency feedback controller. We use the PDDLStream [24] planning

framework, a logic-based action language that supports planning with continuous values, to model our TAMP domain. States and actions are described using *predicates*, Boolean functions, which can have both discrete and continuous parameters. A predicate paired with values for its parameters is called a *literal*. Our TAMP domain uses the following parameters: o is an object, g is a 6-DoF object grasp pose, p is an object placement pose, q is a robot configuration with d DoFs, and τ is a robot trajectory comprised of a sequence of robot configurations.

The planning state s is a set of true literals for *fluent* predicates, predicates who's truth value can change over time. We define the following fluent predicates: $\texttt{AtPose}(o, p)$ is true when object o is placed at placement p; $\texttt{AtGrasp}(o, p)$ is true when object o is grasped using grasp g; $\texttt{AtConf}(q)$ is true when the robot is at configuration q; $\texttt{Empty}()$ is true when the robot's end effector is empty; $\texttt{Attached}(o, o')$ is true when object o is attached to object o';

We use the **Tool Hang** task as a running example (see Fig. 5), where the robot must insert a *frame* into a *stand* and then hang a *tool* on the *frame*. The set of goal system states $\mathcal{X}_*$ is expressed as a *logical formula* over literals. Let s_0 be the initial state s_0 and G be the goal formula:

$$s_0 = \{\texttt{AtPose}(\mathit{frame}, \boldsymbol{p_0^f}), \texttt{AtPose}(\mathit{tool}, \boldsymbol{p_0^t}), \texttt{AtPose}(\mathit{stand}, \boldsymbol{p_0^s}), \texttt{AtConf}(\boldsymbol{q_0}), \texttt{Empty}()\}. \qquad G = \texttt{Attached}(\mathit{frame}, \mathit{stand}) \wedge \texttt{Attached}(\mathit{tool}, \mathit{frame}) \wedge \texttt{Empty}().$$

Planning actions a are represented using action schemata. An action schema is defined by a 1) name, 2) list of parameters, 3) list of *static* (non-fluent) literal *constraints* (**con**) that valid parameter values satisfy, 4) list of fluent literal *preconditions* (**pre**) that must hold to correctly execute the action, and 4) list of fluent literal *effects* (**eff**) that specify changes to state. The `move` action advances the robot from configuration q_1 to q_2 via trajectory τ. The constraint $\texttt{Motion}(q_1, \tau, q_2)$ is satisfied if q_1 and q_2 are the start and end of τ. In the `pick` action, the constraint $\texttt{Grasp}(o, g)$ holds if g is a valid grasp for object o, and the constraint $\texttt{Pose}(o, p)$ holds if p is a valid placement for object o. The explicit constraint $f(q) * g = p$ represents kinematics, namely that forward kinematics f from configuration q multiplied with grasp p produces pose p.

$\texttt{move}(q_1, \tau, q_2)$
- **con:** $[\texttt{Motion}(q_1, \tau, q_2)]$
- **pre:** $[\texttt{AtConf}(q_1),\ \texttt{Safe}(\tau)]$
- **eff:** $[\texttt{AtConf}(q_2),\ \neg\texttt{AtConf}(q_1)]$

$\texttt{pick}(o, g, p, q)$
- **con:** $[\texttt{Grasp}(o, g),\ \texttt{Pose}(o, p),\ [f(q) * g = p]]$
- **pre:** $[\texttt{AtPose}(o, p),\ \texttt{Empty}(),\ \texttt{AtConf}(q)]$
- **eff:** $[\texttt{AtGrasp}(o, g),\ \neg\texttt{AtPose}(o, p),\ \neg\texttt{Empty}()]$

The limitations of the TAMP system are that, although it can readily observe the robot state, it does not have the ability to precisely estimate the environment and productively react to changes in it in real-time. Thus, it's advantageous to teleoperate skills that require 1) contact-rich interaction that is difficult to accurately model and 2) precision greater than that which the perception system can deliver. An example of 1) is the insertion phase of **Tool Hang**, which typically requires contacting the walls of the hole to align the *frame*, and an example of 2) is the hanging phase of **Tool Hang**, which requires precisely aligning the hole of the tool with the resting *frame*.

3 Integrating Human Teleoperation and TAMP

To make TAMP and conventional human teleoperation systems compatible, we describe crucial components that allow for seamless handoff between TAMP and a human operator. These include 1) a novel constraint learning mechanism that allows TAMP to plan to states that enable subsequent human teleoperation (Sec. 3.2) and 2) the core TAMP-gated teleoperation algorithm (Sec. 3.3).

3.1 Teleoperation Action Modeling

To account for human teleoperation during planning, we need an approximate model of the teleoperation process. We build on the strategy of Wang *et al.* [26], which addresses constraint learning for TAMP, by specifying an action schema for each skill identifying which constraints can be modeled using classical techniques. Then, we extract the remaining constraints from a handful of teleoperation trajectories. We teleoperate the *frame* insertion and *tool* hang in the **Tool Hang** task.

The `attach` action models any skill that involves attaching one movable object to another object, for example, by placing, inserting, or hanging. Its parameters are a held object o, the current grasp g for o, the corresponding current pose p of o, the current robot configuration q, the subsequent pose $\widehat{p}$ of o, the subsequent robot configuration $\widehat{q}$, and the object to be attached to o'. This action is stochastic as the human teleoperator "chooses" the resulting pose $\widehat{p}$ and configuration $\widehat{q}$ (indicated by $\widehat{\square}$), which modeled by the constraint $\texttt{HumanAttach}(o, \widehat{p}, \widehat{q}, o')$. Rather than explicitly model this constraint, we take an *optimistic* determinization of the outcome by assuming that the human produces a satisficing $\widehat{p}, \widehat{q}$ pair, without committing to specific numeric values.

$\texttt{attach}(o, g, p, q, \widehat{p}, \widehat{q}, o')$
  con: $[\underline{\texttt{AttachGrasp}(o, g)}, \underline{\texttt{PreAttach}(o, p, o')}, [f(q) * g = p],$
    $\underline{\texttt{GoodAttach}(o, \widehat{p}, o'), \texttt{HumanAttach}(o, \widehat{p}, \widehat{q}, o')}]$
  pre: $[\texttt{AtGrasp}(o, g), \texttt{AtConf}(q)]$
  eff: $[\texttt{AtPose}(o, \widehat{p}), \texttt{Empty}(), \texttt{Attached}(o, o'), \texttt{AtConf}(\widehat{q}), \neg\texttt{AtGrasp}(o, g), \neg\texttt{AtConf}(q)]$

The key constraint is $\texttt{GoodAttach}(o, \widehat{p}, o')$, which is true if object o at pose p satisfies the ground-truth goal attachment condition in $\mathcal{G}$ with object o'. The human teleoperator is tasked with reaching a pose $\widehat{p}$ that satisfies this constraint, which is a postcondition of the action. The goal of model learning is to represent the preconditions (Sec. 3.2) that facilitate this.

3.2 Constraint Learning

To complete the action model, we learn the `AttachGrasp` and `PreAttach` constraints, which involve parameters in `attach`'s preconditions. We can bootstrap these constraint models from a few ($\sim$ 3) human demonstrations. These demonstrations only need to showcase the involved action, which is only a small component of a task. But through compositionality, these actions can be deployed in many new tasks without the need for retraining. In this work, because the set of objects is fixed, we learn a distribution over poses per task and objects. In settings where there are novel objects at test time, we could instead estimate these affordances across objects directly from observations [35, 36].

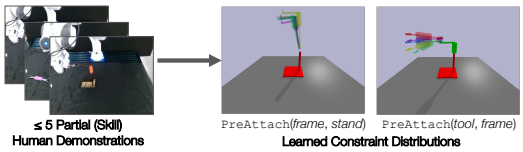


Figure 2: **Constraint learning.** Example of learned attach conditions for the *frame* (*left*) and *tool* from a handful of demonstrations for the **Tool Hang** task.

We define $\texttt{PreAttach}(o, p, o')$ to be true if p is a pose for object o immediately prior to the human achieving $\texttt{GoodAttach}(o, \widehat{p}, o')$. For each human demonstration, we start at the first state where `GoodAttach` is satisfied and then search backward in time for the first state where (1) the robot is holding object o and (2) objects o and o' are at least δ centimeters apart. This minimum distance constraint ensures that o and o' are not in contact in a manner that is spatially consistent and robust to perception and control error. We log the relative pose p between o and o' as a data point and continue iterating over human demonstrations to populate a dataset $P^o_{o'} = \{p \mid \texttt{PreAttach}(o, p, o')\}$.

Similarly, we define $\texttt{AttachGrasp}(o, g)$ to be true if g is a grasp for object o allows for the human achieving $\texttt{GoodAttach}(o, \widehat{p}, o')$. Not all object grasps enable the human to satisfy the target condition, for example, a *frame* grasp on the tip that needs to be inserted. Similar to `PreAttach`, for each demonstration we log the relative pose between the robot end effector and object o at the first pre-contact state before satisfying `GoodAttach`, producing dataset $G^o = \{g \mid \texttt{AttachGrasp}(o, g)\}$.

3.3 TAMP-Gated Teleoperation

We now describe TAMP-gated teleoperation, where a TAMP system decides when to execute portions of a task, and when a human operator should complete a portion (full details in Appendix I). Each teleoperation episode consists of one or more *handoffs* where the TAMP system prompts a human operator to control a portion of a task, or where the TAMP system takes control back after it determines that the human has completed their segment.

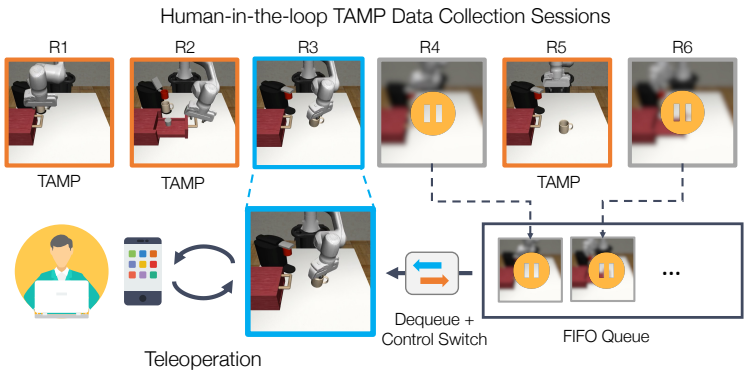


Figure 3: **Queueing system.** HITL-TAMP's queueing system allows a human teleoperator (bottom left) to manage a fleet of asynchronously-running data collection sessions (R1-R6).

Every task is defined by a goal formula G. On each TAMP iteration, it observes the current state s. If it satisfies G the episode terminates, otherwise, the TAMP system solves for a plan $\vec{a}$ from current state s to the goal G. TAMP subsequently issues joint position commands to carry out planned motions until reaching an action a requiring the human. Next, control switches into teleoperation mode, where the human has full 6-DoF control of the end effector. We use a smartphone interface similar to prior teleoperation systems [37, 38, 11]. The robot end effector is controlled using an Operational Space Controller [39]. The TAMP system monitors whether the state satisfies the planned action postconditions $a.effects$. Once satisfied, control switches back to the TAMP system, which replans.

4 Scaling Data Collection for Learning

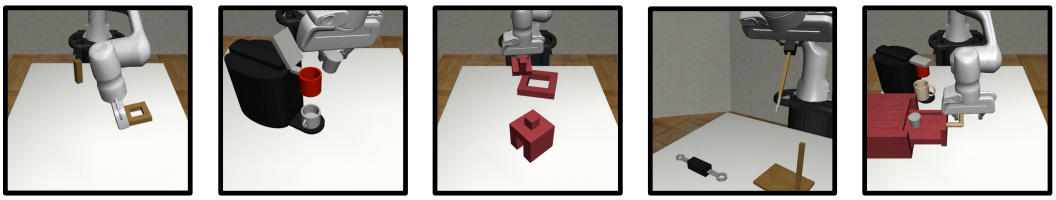
Increasing Data Throughput with a Queueing System. Since the TAMP system only requires human assistance in small parts of an episode, a human operator has the opportunity to manage multiple robots and data collection sessions simultaneously. To this end, we propose a novel queueing system (Fig. 3) allowing each operator to interact with a fleet of robots. We implement this by using several (N_{robot}) **robot processes**, a single **human process**, and a **queue** (more analysis in Appendix H). Each **robot process** runs asynchronously, and spends its time in 1 of 3 modes — (1) being controlled by the TAMP system, (2) waiting for human control, or (3) being controlled by the human. This allows the TAMP system to operate multiple robots in parallel. When the TAMP system wants to prompt the human for control, it enqueues the environment into the shared queue. The **human process** communicates with the human teleoperation device and sends control commands to one robot process at a time. When the human completes a segment, TAMP resumes control of the robot, and the human process dequeues the next session from the queue.

TAMP-Gated Policy Deployment. HITL-TAMP results in demonstrations that consist of TAMP-controlled parts and human-controlled parts — we train a policy with Behavioral Cloning [34] on the human portions (details in Appendix J). To deploy the learned agent, we use a TAMP-gated control loop that is identical to the handoff logic in Sec. 3.3, using the policy instead of the human.

5 Experiment Setup

Tasks. We chose evaluation tasks that are *contact-rich* and *long-horizon*, to validate that HITL-TAMP indeed combines the benefits of the two paradigms (see Fig. 4 and Fig. 5). We further evaluated HITL-TAMP on variants of tasks where objects are initialized in **broad** regions of the workspace, a difficult setting for imitation learning systems in the past. Full details in Appendix E.

Pilot User Study. We conducted a pilot user study with 15 participants to compare our system (HITL-TAMP) to a conventional teleoperation system [37], where task demonstrations were collected without TAMP involvement. Each participant performed task demonstrations on 3 tasks (**Coffee**, **Square (Broad)**, and **Three Piece Assembly (Broad)**) for 10 minutes on each system,



(a) Square (b) Coffee (c) 3 Pc. Assembly (d) Tool Hang (e) Coffee Prep.

Figure 4: **Tasks.** We use HITL-TAMP to collect demonstrations for contact-rich, long-horizon tasks.

Task	Demos (avg-user)	Demos (avg-novice)	Demos (all)	SR (avg-user)	SR (avg-novice)	SR (all)
Coffee (C)	11.2	7.2	168.0	24.4	15.0	76.0
Coffee (HT)	**28.7**	**25.2**	**431.0**	**90.7**	**90.0**	**100.0**
Square Broad (C)	11.1	5.2	166.0	1.2	0.0	20.0
Square Broad (HT)	**49.8**	**41.8**	**747.0**	**80.0**	**77.5**	**98.0**
Three Piece Assembly Broad (C)	7.8	**7.0**	117.0	0.0	0.0	0.0
Three Piece Assembly Broad (HT)	**15.1**	**8.0**	**227.0**	**27.7**	**17.5**	**66.0**

Table 1: **User Study Data Collection and Policy Learning Results.** We report the number of demos collected averaged across users (avg-user), averaged across novice users (avg-novice), and summed across all users (all). We also report the success rate of policies trained on per-user data (avg-user: averaged across all users, and avg-novice: averaged across novice users), and trained on all user data (all). Users collected more demonstrations using HITL-TAMP (HT) than the conventional system (C), and policy performance was vastly greater as well.

totaling 60 minutes of data collection across the 3 tasks and 2 systems. Participants filled out a post-study survey to rank their experience with both systems. Each participant's number of successful demonstrations was recorded to evaluate the data throughput of each system, and agents were trained on each participant's demonstrations and across all participants' demonstrations (Sec. 6.1).

6 Experiment Results

We (1) present user study results to highlight HITL-TAMP's data collection efficiency (Sec. 6.1), (2) compare trained HITL-TAMP agents to policies trained from full task demonstrations (Sec. 6.2), and (3) deploy HITL-TAMP in the real world without precise perception (Sec. 6.3).

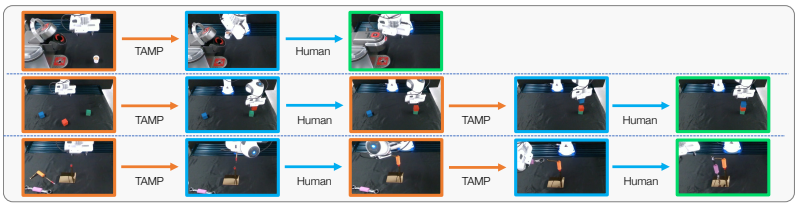
6.1 System Evaluation: User Study

We show that (1) HITL-TAMP allows participants to collect demonstrations much faster than conventional teleoperation, (2) we can train performant policies using data collected from users with varying system proficiency, (3) HITL-TAMP enables novice operators to collect high-quality demonstration data, and (4) HITL-TAMP requires less user effort than conventional teleoperation.

HITL-TAMP enables users to collect task demonstrations at a much higher rate than a conventional teleoperation system. As Table 1 shows, collectively, our 15 users gathered 2.5x more demonstrations with HITL-TAMP when compared to the conventional system on the Coffee task (431 vs. 168), 4.5x more on Square Broad (747 vs. 166), and nearly 2x more on Three Piece Assembly Broad (227 vs. 117). The high collection efficacy of HITL-TAMP was also reflected on a per-user basis — users averaged 28.7 demos on Coffee (vs. 11.2), 49.8 demos on Square Broad (vs. 11.1), and 15.1 demos on Three Piece Assembly Broad (vs. 7.8), during their 10-minute sessions.

HITL-TAMP enables performant policies to be trained from minutes of data. We used each person's 10-minute demonstrations to train a policy for each (user-task) pair with behavioral cloning. Agents trained on HITL-TAMP data vastly outperformed those trained from the conventional teleoperation data (Table 1) — agents achieved an average success rate of 90.7% on Coffee (vs. 24.4%), 80.0% on Square Broad (vs. 1.2%), and 27.7% on Three Piece Assembly Broad (vs. 0.0%).

HITL-TAMP enables training proficient agents from multi-user data. Prior work [40, 1] noted that imitation learning from multi-user demonstrations can be difficult. However, we found agents trained on the full set of multi-user HITL-TAMP data achieve high success rates (100.0%, 98.0%, and 66.0% on Coffee, Square Broad, and Three Piece Assembly Broad, respectively) compared to



Task	Success Rate
Stack Three	62.0
Coffee	74.0
Coffee Broad	66.0
Tool Hang	64.0

Figure 5: (left) **Real Tasks.** Coffee (*top*), a version where the machine can be on either side (Coffee Broad, not shown), Stack Three (*middle*), and **Tool Hang** (*bottom*). We show TAMP in orange and the human in blue. (right) **Real World Policy Performance.** We collected 100 demonstrations on Stack Three, Coffee, and Coffee Broad and 50 demonstrations on Tool Hang with HITL-TAMP and report the policy performance in this table.

Task	Time (min)	SR (low-dim)	SR (image)
Square	13.5	100.0 ± 0.0	100.0 ± 0.0
Square Broad	14.0	100.0 ± 0.0	100.0 ± 0.0
Coffee	22.6	100.0 ± 0.0	100.0 ± 0.0
Coffee Broad	28.8	99.3 ± 0.9	96.7 ± 0.9
Tool Hang	48.0	80.7 ± 1.9	78.7 ± 0.9
Tool Hang Broad	51.5	49.3 ± 1.9	40.7 ± 0.9
Three Piece Assembly	30.0	100.0 ± 0.0	100.0 ± 0.0
Three Piece Assembly Broad	34.9	84.7 ± 4.1	82.0 ± 1.6
Coffee Preparation	78.4	96.0 ± 3.3	100.0 ± 0.0

Task	Time (min)	SR (im)	TAMP-gated SR (im)
Square (C) [1]	25.0	82.0 ± 0.0	$\mathbf{100.0 \pm 0.0}$
Square (HT)	**13.5**	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
Square Broad (C)	48.0	15.3 ± 0.0	94.7 ± 0.9
Square Broad (HT)	**14.0**	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
Three Piece Assembly (C)	60.0	75.3 ± 0.0	77.3 ± 7.7
Three Piece Assembly (HT)	**30.0**	$\mathbf{100.0 \pm 0.0}$	$\mathbf{100.0 \pm 0.0}$
Tool Hang (C) [1]	80.0	67.3 ± 0.0	$\mathbf{82.0 \pm 2.8}$
Tool Hang (HT)	**48.0**	$\mathbf{78.7 \pm 0.9}$	$\mathbf{78.7 \pm 0.9}$

Figure 6: (left) **Results on HITL-TAMP datasets.** We collected 200 demonstrations on each task with HITL-TAMP and trained low-dim and visuomotor agents TAMP-gated agents on each dataset. (right) **Comparison to conventional teleoperation datasets.** We trained both normal and TAMP-gated policies using conventional teleoperation (C) and compared them to HITL-TAMP (HT). Surprisingly, TAMP-gating makes policies trained on the data comparable to HITL-TAMP data, but data collection still involves significantly higher operator time.

those trained on the full set of conventional teleoperation data (76.0%, 20.0%, 0.0%) (see Table 1). In fact, the worst per-user HITL-TAMP policy (10-minutes of data) outperformed the policy trained on the full set of conventional teleoperation data (150 minutes) on both Square Broad (56.0% vs. 20.0%) and Three Piece Assembly Broad (14.0% vs. 0.0%).

HITL-TAMP enables non-experts to demonstrate tasks efficiently. 4 of the 15 users in our study had no experience with teleoperation. Table 1 shows we found that they were able to collect far more data on average with HITL-TAMP (more than 3x on Coffee, more than 8x on Square Broad) and policies trained on their HITL-TAMP data achieved significantly higher success over the conventional system — 90.0% (vs. 15.0%) on Coffee, 77.5% (vs. 0.0%) on Square Broad, and 17.5% (vs. 0.0%) on Three Piece Assembly Broad.

HITL-TAMP results in a lower perceived workload compared to the conventional teleoperation system. Each participant completed a NASA-TLX survey [41] to rank their perceived workload for each system across 6-categories (100-point scale, increments of 5). Users found HITL-TAMP to require less mental demand (36% vs. 74%), less physical demand (29.7% vs. 63.7%), and less temporal demand (28.3% vs. 53.7%), while enabling higher overall performance (83.7% vs. 59.7%), with lower effort (29.3% vs. 75.7%) and lower frustration (30.0% vs. 65.0%).

6.2 Learning Results

We collect datasets with HITL-TAMP across 9 tasks (see Sec. 5) and show that highly capable policies can be trained from this data. The results compare favorably to training on equal amounts of demonstrations from a conventional teleoperation system.

HITL-TAMP is broadly applicable to a wide range of contact-rich and long-horizon tasks. Using HITL-TAMP, we had a single human operator collect 200 demonstrations on each of our tasks. We then trained agents from this data on two observation spaces — *low-dim* observations, where agents directly observe the poses of relevant objects, and *image* observations, where agents observe a front-view RGB image and wrist RGB image (as in [1]). Table 6 shows that across both observation spaces, HITL-TAMP trains near-perfect agents on several tasks (Square, Coffee, Three Piece Assembly), including **broad** tasks with a wide distribution of object initialization (Square Broad, Coffee Broad, Three Piece Assembly Broad). HITL-TAMP also achieves high performance on the Tool Hang task (80.7% low-dim, 78.7% image), which is the hardest task in the robomimic

benchmark [1]. It is also able to train performant agents (49.3% low-dim, 40.7% image) on a **broad** version of the task (Tool Hang Broad). Finally, HITL-TAMP trains near-perfect agents (96%) on the Coffee Preparation task, which consists of several stages (4 TAMP segments and 4 policy segments) involving low-tolerance mug placement, drawer grasping and opening, lid opening, and pod insertion and lid closing.

HITL-TAMP compares favorably to conventional teleoperation systems in terms of operator time and policy learning. Even when an equal number of task demonstrations are used, learned policies from HITL-TAMP still outperform those from conventional teleoperation. We run our comparison on 4 tasks — Square, Square Broad, Three Piece Assembly, and Tool Hang, where each task has 200 HITL-TAMP demos collected and 200 conventional system demos. As Table 6 shows, HITL-TAMP enabled collecting 200 demonstrations on each task in much shorter periods of time (additional analysis in Appendix F). Furthermore, agents trained on HITL-TAMP data outperform agents trained on conventional data (with the largest gap being 100.0% vs. 15.3% on Square Broad).

TAMP-gated control is a crucial component to train proficient policies. We took the 200 demonstration datasets collected via conventional teleoperation, trained the agents as normal, but deployed them with TAMP-gated control during policy evaluation. This dramatically increases their success rates and gives comparable results to HITL-TAMP data (see Table 6). This shows that datasets consisting of entire human demonstration trajectories are compatible with TAMP-gated control. However, they remain time-consuming to collect, and HITL-TAMP greatly reduces the time needed.

6.3 Real Robot Validation

We apply HITL-TAMP to a physical robot setup with a robotic arm, a front-view camera, and a wrist-mounted camera. The only significant change from simulation is the need for perception to obtain pose estimates of the objects to populate the TAMP state. We do not assume any capability to track object poses in real-time. Instead, we allow the human to demonstrate (and the policy to imitate) behaviors from partial observations (RGB cameras). We collected 100 demonstrations for each of 3 tasks — Stack Three, Coffee, and Coffee Broad, and 50 demonstrations on Tool Hang, and report policy learning results across 50 evaluations for each task (25 for Tool Hang) (see Fig. 5). Our TAMP-gated agent achieves 62% on Stack Three, 74% on Coffee, 66% on Coffee Broad (72% with the machine on the right side of the table, and 60% with the machine on the left side), and **64% on Tool Hang (as opposed to the 3% from 200 human demonstrations in prior work [1])**.

7 Limitations

See Appendix C for full limitations. We assume tasks can be described in PDDLStream and that human teleoperators can demonstrate them. The tasks in this work focus on tabletop domains with limited object variety — future work could scale HITL-TAMP to more diverse settings. Currently, HITL-TAMP requires prior information (at a high-level) on which task portions will be difficult for TAMP. We also assume access to coarse object models and approximate pose estimation to conduct TAMP segments in the real world. Future work could relax these assumptions by integrating perception uncertainty estimates, and extending TAMP to not require object models [36].

8 Conclusion

We presented a new approach to teach robots complex manipulation skills through a hybrid strategy of automated planning and human control. Our system, HITL-TAMP, collects human demonstrations using a TAMP-gated control mechanism and learns preimage models of human skills. This allows for a human to efficiently supervise a team of worker robots asynchronously. The combination of TAMP and teleoperation in HITL-TAMP results in improved data collection and policy learning efficiency compared to collecting human demonstrations on the entire task.

References

[1] A. Mandlekar, D. Xu, J. Wong, S. Nasiriany, C. Wang, R. Kulkarni, L. Fei-Fei, S. Savarese, Y. Zhu, and R. Martín-Martín. What matters in learning from offline human demonstrations for robot manipulation. In *Conference on Robot Learning (CoRL)*, 2021.

[2] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[3] 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*, 3: 297–330, 2020.

[4] M. A. Toussaint, K. R. Allen, K. A. Smith, and J. B. Tenenbaum. Differentiable physics and stable modes for tool-use and manipulation planning. 2018.

[5] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez. Integrated task and motion planning. *Annual review of control, robotics, and autonomous systems*, 4:265–293, 2021.

[6] T. Zhang, Z. McCarthy, O. Jow, D. Lee, K. Goldberg, and P. Abbeel. Deep imitation learning for complex manipulation tasks from virtual reality teleoperation. *arXiv preprint arXiv:1710.04615*, 2017.

[7] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, J. Ibarz, B. Ichter, A. Irpan, T. Jackson, S. Jesmonth, N. Joshi, R. Julian, D. Kalashnikov, Y. Kuang, I. Leal, K.-H. Lee, S. Levine, Y. Lu, U. Malla, D. Manjunath, I. Mordatch, O. Nachum, C. Parada, J. Peralta, E. Perez, K. Pertsch, J. Quiambao, K. Rao, M. Ryoo, G. Salazar, P. Sanketi, K. Sayed, J. Singh, S. Sontakke, A. Stone, C. Tan, H. Tran, V. Vanhoucke, S. Vega, Q. Vuong, F. Xia, T. Xiao, P. Xu, S. Xu, T. Yu, and B. Zitkovich. Rt-1: Robotics transformer for real-world control at scale. In *arXiv preprint arXiv:2212.06817*, 2022.

[8] E. Jang, A. Irpan, M. Khansari, D. Kappler, F. Ebert, C. Lynch, S. Levine, and C. Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In *Conference on Robot Learning*, pages 991–1002. PMLR, 2022.

[9] C. Lynch, M. Khansari, T. Xiao, V. Kumar, J. Tompson, S. Levine, and P. Sermanet. Learning latent plans from play. In *Conference on robot learning*, pages 1113–1132. PMLR, 2020.

[10] C. Lynch, A. Wahid, J. Tompson, T. Ding, J. Betker, R. Baruch, T. Armstrong, and P. Florence. Interactive language: Talking to robots in real time. *arXiv preprint arXiv:2210.06407*, 2022.

[11] A. Mandlekar, D. Xu, R. Martín-Martín, S. Savarese, and L. Fei-Fei. Learning to generalize across long-horizon tasks from human demonstrations. *arXiv preprint arXiv:2003.06085*, 2020.

[12] M. Ahn, A. Brohan, N. Brown, Y. Chebotar, O. Cortes, B. David, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, et al. Do as i can, not as i say: Grounding language in robotic affordances. *arXiv preprint arXiv:2204.01691*, 2022.

[13] R. Hoque, A. Balakrishna, C. Putterman, M. Luo, D. S. Brown, D. Seita, B. Thananjeyan, E. Novoseller, and K. Goldberg. Lazydagger: Reducing context switching in interactive imitation learning. In *2021 IEEE 17th International Conference on Automation Science and Engineering (CASE)*, pages 502–509. IEEE, 2021.

[14] R. Hoque, L. Y. Chen, S. Sharma, K. Dharmarajan, B. Thananjeyan, P. Abbeel, and K. Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In *6th Annual Conference on Robot Learning*.

[15] J. Zhang and K. Cho. Query-efficient imitation learning for end-to-end autonomous driving. *arXiv preprint arXiv:1605.06450*, 2016.

[16] R. Hoque, A. Balakrishna, E. Novoseller, A. Wilcox, D. S. Brown, and K. Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. *arXiv preprint arXiv:2109.08273*, 2021.

[17] S. Dass, K. Pertsch, H. Zhang, Y. Lee, J. J. Lim, and S. Nikolaidis. Pato: Policy assisted teleoperation for scalable robot data collection. *arXiv preprint arXiv:2212.04708*, 2022.

[18] T. Silver, K. Allen, J. Tenenbaum, and L. Kaelbling. Residual policy learning. *arXiv preprint arXiv:1812.06298*, 2018.

[19] T. Johannink, S. Bahl, A. Nair, J. Luo, A. Kumar, M. Loskyll, J. A. Ojea, E. Solowjow, and S. Levine. Residual reinforcement learning for robot control. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6023–6029. IEEE, 2019.

[20] A. Kurenkov, A. Mandlekar, R. Martin-Martin, S. Savarese, and A. Garg. Ac-teach: A bayesian actor-critic method for policy learning with an ensemble of suboptimal teachers. *arXiv preprint arXiv:1909.04121*, 2019.

[21] O. Mees, J. Borja-Diaz, and W. Burgard. Grounding language with visual affordances over unstructured data. *arXiv preprint arXiv:2210.01911*, 2022.

[22] E. Valassakis, N. Di Palo, and E. Johns. Coarse-to-fine for sim-to-real: Sub-millimetre precision across wide task spaces. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5989–5996. IEEE, 2021.

[23] L. P. Kaelbling and T. Lozano-Pérez. Hierarchical task and motion planning in the now. In *ICRA*, 2011.

[24] C. R. Garrett, T. Lozano-Pérez, and L. P. Kaelbling. Pddlstream: Integrating symbolic planners and blackbox samplers via optimistic adaptive planning. In *Proceedings of the International Conference on Automated Planning and Scheduling*, volume 30, pages 440–448, 2020.

[25] G. Konidaris, L. P. Kaelbling, and T. Lozano-Perez. From skills to symbols: Learning symbolic representations for abstract high-level planning. *Journal of Artificial Intelligence Research*, 61: 215–289, 2018.

[26] Z. Wang, C. R. Garrett, L. P. Kaelbling, and T. Lozano-Pérez. Learning compositional models of robot skills for task and motion planning. *The International Journal of Robotics Research*, 40(6-7):866–894, 2021.

[27] J. Liang, M. Sharma, A. LaGrassa, S. Vats, S. Saxena, and O. Kroemer. Search-based task planning with learned skill effect models for lifelong robotic manipulation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6351–6357. IEEE, 2022.

[28] H. M. Pasula, L. S. Zettlemoyer, and L. P. Kaelbling. Learning symbolic models of stochastic domains. *Journal of Artificial Intelligence Research*, 29:309–352, 2007.

[29] T. Silver, R. Chitnis, J. Tenenbaum, L. P. Kaelbling, and T. Lozano-Pérez. Learning symbolic operators for task and motion planning. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3182–3189. IEEE, 2021.

[30] R. Chitnis, D. Hadfield-Menell, A. Gupta, S. Srivastava, E. Groshev, C. Lin, and P. Abbeel. Guided search for task and motion plans using learned heuristics. In *ICRA*. IEEE, 2016.

[31] B. Kim, L. Shimanuki, L. P. Kaelbling, and T. Lozano-Pérez. Representation, learning, and planning algorithms for geometric task and motion planning. *IJRR*, 41(2), 2022.

[32] S. Cheng and D. Xu. Guided skill learning and abstraction for long-horizon manipulation. *arXiv preprint arXiv:2210.12631*, 2022.

[33] T. Silver, A. Athalye, J. B. Tenenbaum, T. Lozano-Pérez, and L. P. Kaelbling. Learning neuro-symbolic skills for bilevel planning. In *6th Annual Conference on Robot Learning*.

[34] D. A. Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In *Advances in neural information processing systems*, pages 305–313, 1989.

[35] M. Sundermeyer, A. Mousavian, R. Triebel, and D. Fox. Contact-graspnet: Efficient 6-dof grasp generation in cluttered scenes. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13438–13444. IEEE, 2021.

[36] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett. Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 1940–1946. IEEE, 2022.

[37] A. Mandlekar, Y. Zhu, A. Garg, J. Booher, M. Spero, A. Tung, J. Gao, J. Emmons, A. Gupta, E. Orbay, S. Savarese, and L. Fei-Fei. RoboTurk: A Crowdsourcing Platform for Robotic Skill Learning through Imitation. In *Conference on Robot Learning*, 2018.

[38] A. Mandlekar, J. Booher, M. Spero, A. Tung, A. Gupta, Y. Zhu, A. Garg, S. Savarese, and L. Fei-Fei. Scaling robot supervision to hundreds of hours with roboturk: Robotic manipulation dataset through human reasoning and dexterity. *arXiv preprint arXiv:1911.04052*, 2019.

[39] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal on Robotics and Automation*, 3(1):43–53, 1987.

[40] A. Mandlekar, F. Ramos, B. Boots, S. Savarese, L. Fei-Fei, A. Garg, and D. Fox. Iris: Implicit reinforcement without interaction at scale for learning control from offline robot manipulation data. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4414–4420. IEEE, 2020.

[41] S. G. Hart and L. E. Staveland. Development of nasa-tlx (task load index): Results of empirical and theoretical research. In *Advances in psychology*, volume 52, pages 139–183. Elsevier, 1988.

[42] O. Mees, L. Hermann, E. Rosete-Beas, and W. Burgard. Calvin: A benchmark for language-conditioned policy learning for long-horizon robot manipulation tasks. *IEEE Robotics and Automation Letters*, 7(3):7327–7334, 2022.

[43] A. Mandlekar, D. Xu, R. Martín-Martín, Y. Zhu, L. Fei-Fei, and S. Savarese. Human-in-the-loop imitation learning using remote teleoperation. *arXiv preprint arXiv:2012.06733*, 2020.

[44] K. Van Wyk, M. Culleton, J. Falco, and K. Kelly. Comparative peg-in-hole testing of a force-based manipulation controlled robotic hand. *IEEE Transactions on Robotics*, 34(2):542–549, 2018.

[45] H. Park, J. Park, D.-H. Lee, J.-H. Park, and J.-H. Bae. Compliant peg-in-hole assembly using partial spiral force trajectory with tilted peg posture. *IEEE Robotics and Automation Letters*, 5(3):4447–4454, 2020.

[46] M. J. McDonald and D. Hadfield-Menell. Guided imitation of task and motion planning. In *Conference on Robot Learning*, pages 630–640. PMLR, 2022.

[47] M. Kelly, C. Sidrane, K. Driggs-Campbell, and M. J. Kochenderfer. Hg-dagger: Interactive imitation learning with human experts. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8077–8083. IEEE, 2019.

[48] J. S. Warm, R. Parasuraman, and G. Matthews. Vigilance requires hard mental work and is stressful. *Human factors*, 50(3):433–441, 2008.

[49] R. Tedrake, M. Fallon, S. Karumanchi, S. Kuindersma, M. Antone, T. Schneider, T. Howard, M. Walter, H. Dai, R. Deits, et al. A summary of team mit's approach to the virtual robotics challenge. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2087–2087. IEEE, 2014.

[50] R. Luo, C. Wang, E. Schwarm, C. Keil, E. Mendoza, P. Kaveti, S. Alt, H. Singh, T. Padir, and J. P. Whitney. Towards robot avatars: Systems and methods for teleinteraction at avatar xprize semi-finals. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 7726–7733. IEEE, 2022.

[51] J. M. Marques, N. Patrick, Y. Zhu, N. Malhotra, and K. Hauser. Commodity telepresence with the avatrina nursebot in the ana avatar xprize semifinals. In *RSS 2022 Workshop on "Towards Robot Avatars: Perspectives on the ANA Avatar XPRIZE Competition*, 2022.

[52] H. Le, N. Jiang, A. Agarwal, M. Dudík, Y. Yue, and H. Daumé III. Hierarchical imitation and reinforcement learning. In *International conference on machine learning*, pages 2917–2926. PMLR, 2018.

[53] K. Shiarlis, M. Wulfmeier, S. Salter, S. Whiteson, and I. Posner. Taco: Learning task decomposition via temporal alignment for control. In *International Conference on Machine Learning*, pages 4654–4663. PMLR, 2018.

[54] Y. Zhu, J. Wong, A. Mandlekar, and R. Martín-Martín. robosuite: A modular simulation framework and benchmark for robot learning. In *arXiv preprint arXiv:2009.12293*, 2020.

[55] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, 1996.