Original Article

The International Journal of Robotics Research
2025, Vol. 0(0) 1–18

DOI: 10.1177/02783649241312699
journals.sagepub.com/home/ijr

Shared autonomy policy fine-tuning and alignment for robotic tasks

Ehsan Yousefi[1], Mo Chen[2] and Inna Sharf[1]

Abstract

In this paper, we present a comprehensive shared autonomy framework for human-in-the-loop policy fine-tuning and alignment. Our framework integrates policy adapting algorithms on a multi-agent system foundation tailored for human-robot interaction and decision-making arbitration. This strategy is intended for complex, task-oriented robotic tasks that require cognitive-level human-robot interactions. We design short- and long-horizon fine-tuning algorithms to adapt a policy to different operating conditions and human agents. This is accomplished using Bayesian analysis and custom deep reinforcement learning techniques, through various interaction channels strategically placed at different operational points of the system. To showcase the effectiveness of our algorithms, as well as the strength of our framework, we conduct a human user study involving operation of a laboratory robot in a sequence of high-level pick-and-place tasks. The experiments of the study are designed to demonstrate the interplay between different design elements of our framework, such as, interaction channels and multi-horizon fine-tuning algorithms. By laying out careful hypotheses, we employ objective and subjective metrics to measure the effects of shared autonomy design elements on both the system performance and human user satisfaction. Our human user study reveals significant results related to the complex interplay between shared autonomy design elements, the behavior of the algorithms, and core decision-making and arbitration formulation.

Keywords

Shared autonomy, human-robot interaction, policy adapting, fine-tuning, MDP, deep RL, robotic arms

Received 18 April 2024; Revised 1 December 2024; Accepted 2 December 2024

1. Introduction

As the application domains and capabilities of robotic intelligent systems are rapidly expanding, a growing co-existence of robots and humans is inevitable. The problem of *shared autonomy*, that is, how to allocate or share decision-making authority between artificial/machine and biological/human intelligence, specifically on a cognitive level, is thus becoming more relevant. This necessitates not only elaborate schemes of interaction between humans and robots but also an explicit framework for ongoing, continual (co-)operation and (co-)learning. A shared autonomy framework facilitates the strategic design of possible schemes of interaction between the agents in a system with different degrees of autonomy. Shared autonomy, as an umbrella strategy, already finds applications in a wide variety of challenging domains, including autonomous driving (Kiran et al., 2021) and assistive robots (Losey et al., 2022). Our interest in shared autonomy is motivated by applications involving operation of large-scale mobile robotic arms, such as the machines employed in construction, mining and timber-harvesting industries.

Motivated by the intended complex robotics applications of shared autonomy, the notion of inherent hierarchy in the task at hand is of significant importance. This is so for two reasons: 1) from a cognitive perspective, decision-making, and subsequent interaction, happen in a hierarchy of levels because of the spatiotemporal nature of their associated underlying processes (Annaswamy et al., 2024); 2) robotic tasks, also, can be decomposed into a hierarchy of tasks and sub-tasks (Yousefi et al., 2022, 2023). These hierarchies can be architecturally co-designed for efficient shared autonomy.

[1]Department of Mechanical Engineering, McGill University, Montreal, QC, Canada
[2]School of Computing Science, Simon Fraser University (SFU), Burnaby, BC, Canada

Corresponding author:
Ehsan Yousefi, Department of Mechanical Engineering, McGill University, 817 Sherbrooke Street, Montreal, QC H3A 0C3, Canada.
Email: ehsan.yousefi@mail.mcgill.ca

In setting up a shared autonomy architecture for an intuitive, yet responsive policy adapting, it is crucial to design the interaction model to leverage offline priors and online adaptations (Annaswamy et al., 2024). The challenge in our intended applications and many others is that, in practice, we have access to a limited number of human trials. At the same time, transferring a model pre-trained in a simulated world to the real world is a challenge for two main reasons: 1) the fact that the real human, environment, robot, and task settings might differ from those in the simulation environment; 2) the high-level and complex nature of the cognitive planning processes. This brings us to the central issue addressed in this paper, that of *fine-tuning* a pre-trained model to the real world, as well as, its *alignment* to a human agent. Moreover, it is imperative that a shared autonomy continually learns from the human, as well as the operations, in order to progressively accommodate a wider range of edge cases, while maintaining safety, thus, ultimately taming the long tail problem of AI.

1.1. State of the art

In this study, we focus our attention on shared autonomy between human and autonomous agents in the loop, both with decision-making capability. In the literature, the mechanism for sharing the control authority is discussed under the umbrella term of *arbitration*, which is architecturally categorized into two groups: 1) policy blending (Dragan and Srinivasa, 2013), in which the agents are parallel to each other and their actions are blended through an arbitration function. Despite its success in certain domains (Losey et al., 2022) and its conceptual closeness to the original view of shared autonomy, policy blending generally suffers from the problem of convolving two downstream signals that might differ in nature, which may result in unfavorable outcomes, especially at cognitive levels; 2) policy adapting, in which the agents are in series and the autonomous action is based on human action (Yousefi et al., 2023). The drawback of the second option is that the agent generally feels less in control, despite better task execution performance (Javdani et al., 2018). In this work, we bridge the gap between these two schools of thought and provide a unified framework for shared autonomy with different settings and protocols to accommodate a sliding level of autonomy, better alignment with human agent, minimum assumptions about the human agent and even allowing for non-persistent (non-)collaborative human input. As will be demonstrated in this paper, this requires short- and long-term autonomous policy fine-tuning and alignment.

From the policy fine-tuning point of view, there have been many successes in the Large Language Model (LLM) realm, especially with human feedback, notably Ziegler et al. (2020). While we incorporate certain ideas from the LLM fine-tuning literature, the need for shared autonomy policy fine-tuning for robotic tasks requires further analysis. From the perspective of learning with prioritized samples, specifically, prioritizing certain experiences, methods such as Prioritized experience replay (Schaul et al., 2016) and Prioritized Trajectory Replay (Liang et al., 2021) are available in the literature.

In robotic applications, pre-training of the policy in a safe, realistic simulation environment is an important first step, as it is not practical to deploy a costly robot in the field to learn a policy, while starting with a clean slate. As an example, in Smith et al. (2022), the authors use an imitation-based strategy to train RL policies in simulation for locomotion skills of a quadrupedal robot including reset/recovery using randomized ensembled double Q-learning (REDQ) (Chen et al., 2021). Then, they fine-tune those skills in real-world tests by resetting the replay buffer and leveraging the reset/recovery skills to perform multiple training loops with minimal intervention. In Julian et al. (2021), the authors suggest an offline fine-tuning methodology by using the data collected in a new environment condition and improving the pre-trained policy with it. It is demonstrated that extending this methodology to a continual learning setup is also possible by repeating the model adaptation as the environment changes. However, as we noted earlier, even after a successful pre-training, we have access to limited trials, especially for complex and safety-compromised human-in-the-loop robotic tasks involving large robotic arms. Fine-tuning and alignment of a pre-trained policy for such cases is still an open problem.

1.2. Contributions

In the following, we concisely list the contributions of this paper:

1. Building on the foundations of our earlier work (Yousefi et al., 2023), we introduce a general end-to-end policy adapting shared autonomy architecture for robotic task-based hierarchical planning. Moreover, due to the inherent differences between the biological intelligence/human and the artificial intelligence/autonomy, we design multiple interaction channels placed strategically at different points of the end-to-end loop to facilitate a seamless interaction between the agents in the system.
2. We next provide short- and long-horizon algorithms to fine-tune a shared autonomy policy with human interactions to new operational settings, thereby producing a complete shared autonomy architecture design.
3. We next lay out the implementation details accompanied with our setup for experiments. By conducting a human user study, we discuss the effects of shared autonomy design elements such as interaction channels and fine-tuning algorithms on system performance, human satisfaction, and their comfort.

To the best of our knowledge, this is the first time in literature that this type of architecture with fine-tuning for shared control of a robotic arm for functional shared autonomy is presented.

1.3. Paper organization

This paper is organized as follows: in §2, we introduce the complete architecture of our comprehensive shared autonomy design as well as the planning and interaction model. In §3, we tackle the problem of designing a multi-horizon fine-tuning algorithm for our shared autonomy framework. In §4, we discuss our implementation setup. For the case study application described in §5, we design a detailed user study in §6 to shed light on how different architecture design components interplay during co-operation between human and autonomy. We present the results and discuss our findings in §7. Finally, §8 concludes the paper together with suggested future work.

2. Problem formulation

2.1. Problem statement

The overarching problem we are concerned with in this paper is how to design a comprehensive shared autonomy framework for task-oriented robotic applications, in order to facilitate smooth co-operation and co-learning between a human and autonomous agents. To this end, we begin with: a) a complete architecture, represented with a block diagram, with accompanying discussion of each block in §2.2, and b) a planning strategy and a model of interaction (i.e., *policy adapting*) in §2.3. With this structure in place, we tackle the problem of designing two *fine-tuning* strategies—short- and long-horizon—which are at the heart of our architecture. Our fundamental hypothesis is that through these fine-tuning algorithms, the co-operation between human and autonomy can improve and co-learning can take place. Since the elements of architectural design are elaborately intertwined and tied to physical and intellectual human-robot interactions, the study of their interplay is crucial, both from a system design point of view, but also from a practical perspective. This interplay in interaction and understanding it through a carefully designed human user study is another important problem we address in this work.

2.2. Shared autonomy design

As discussed in §1, the move from simulation and offline approaches to real world, particularly in robotic applications, is challenging. We believe that the concept of shared autonomy, beyond its relevance and utility for modern human-robot interactions, is a way forward to address those challenging scenarios. We present our vision for a well-rounded shared autonomy architecture in Figure 1: it represents the next step in the evolution of our previous shared autonomy framework of Yousefi et al. (2023), while building on the foundational understanding in that earlier work. With that proviso, we introduce the blocks in Figure 1 in more detail:

I) Human: The characteristic of a shared autonomy platform is its human agent-in-the-loop nature, that by definition includes a human in various stages of the interaction.

II) Environment Encoder: In designing a successful interaction between the agents of the system, it is crucial to feed in a proper representation of the environment. In this work, we encode the environment representative of a specific, yet challenging, case study of a timber-harvesting machine. Since the machine's task is to carry out a sequence of pick-and-place operations with known goal space, the formulation can be extended to other applications with this type of tasks. Our study also illustrates how this step is intertwined with the design of the other shared autonomy elements.

III) Human encoder: This block serves as the interface between the raw measurements of different interactions collected from/on the human and the subsequent algorithms that co-process the inputs for a context-aware decision-making. The specific type of signal required from the human depends on the field of application. In our setup, which is concerned with the behavioral decision-making analysis, direct high-level human input, as well as, the inferred signals from their interaction with the robot are considered, in light of existing literature (Losey et al., 2022; Reddy et al., 2018; Yousefi et al., 2022, 2023).

IV) Interaction channels: The interaction signals are feature-rich sources of information that act as messengers between the agents as well as different levels in the hierarchy. In general, regardless of the modality of the interaction signals, we envision the following categories of interaction channels:

- From human to autonomy:
  1. Human intervention: these signals are action-level and include input actions and overrides that are intended to influence the autonomy and its policy directly or even bypass it.
  2. Satisfaction/evaluative feedback: these are signals intended to influence the policy gradually by scoring its performance.

- From autonomy to human:
  These include the purposefully designed task and the related sequence that can include different modalities such as audio.

V) Shared encoder layer 1 (L1): In this layer, human and robot data augmentation and pre-processing for the shared autonomy stage are carried out. If the robot operations occur in a continuous domain, then this layer is also tasked with the transition between higher-level discrete decision-making and low-level continuous execution.

VI) Policy: this block is dedicated to the policy of the shared autonomy setting. It is shown separately in order to highlight how other elements interact with it and affect it. This highlights the importance of the

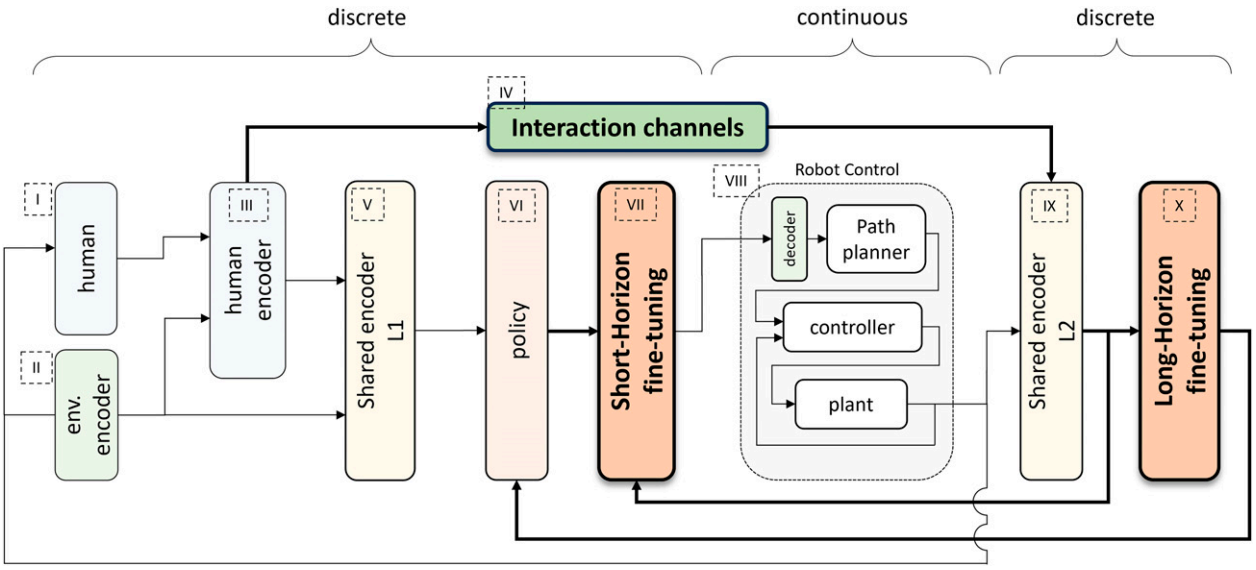

Figure 1. The proposed comprehensive shared autonomy architecture (introduced in §2.2). The focus of this paper is mainly on the 3 blocks: the short and long fine-tuning blocks, and the interaction channels (discussed in §3 and §4). Special care is given to the flow of signals (the highlighted arrows) between interaction channels, how we encode and process them jointly with the fine-tuning algorithms and how they affect the policy. We discuss these thoroughly in §2.2 and §6. The results of the interplay between these design elements are discussed in §7. We further discuss block VIII in §4 within our implementation details and blocks II, III, V, VI, IX through a case study in §5.

decision-making process and how the policy structure is related to the system as a whole and co-designed with the other blocks of the system.

VII) Short-horizon fine-tuning: Fast alignment of the policy to the spatiotemporal human signal is of significant importance to avoid humans experiencing a disconnect from the autonomy in a human-conditioned policy architecture (Javdani et al., 2018). This block is designed for this purpose.

VIII) Robot control block: This block is where the decisions are materialized. As needed, a low-level trajectory planner and a controller are deployed here.

IX) Shared encoder layer 2 (L2): This layer is designed to facilitate multi-channel interactions between the agents. This block is designed to post-process the signals from block IV.

X) Long-horizon fine-tuning: Finally, this block is designed to facilitate long-horizon (co-)learning and policy fine-tuning, while leveraging the shared nature of the setup.

2.3. Hierarchical planning and policy adapting

As previously mentioned, we approach the challenge of robot planning by framing it as a hierarchy of tasks and functions. This approach offers benefits when dealing with various agents, such as humans, in scenarios involving shared autonomy and variable degrees of autonomy (Jorda et al., 2022). This is at the core of the designed layers and blocks in §2.2 and Figure 1. Furthermore, adopting this task-centered perspective allows us to seamlessly integrate the natural task hierarchy, leading to a hierarchical structure in human-robot interactions (Guo et al., 2019). Moreover, task-oriented planning can be viewed as a sequential decision-making problem aimed at optimizing specific metrics tailored to the task at hand. We draw an analogy between an *option* within the Options framework (Pateria et al., 2021; Sutton et al., 1999) and a task within our robot planning context. Much like a task can encompass various sub-tasks, an option typically involves multiple actions.

Leveraging the above analogy, we employ the Markov decision process (MDP) framework, which offers a model for making sequential decisions, accounting for the actions of agents while considering the inherent randomness in the process. Notably, the Options framework is constructed upon semi-Markov decision processes (semi-MDPs), which extend the concept of MDPs to incorporate a temporal dimension. This extension allows our shared autonomy approach to accommodate robotic tasks of varying duration and hierarchical structures. Furthermore, it has been demonstrated that the behavior of a human operator controlling a robotic arm can be described as a structured sequence of repetitive sub-tasks (Westerberg, 2014). Consequently, our shared autonomy framework is designed to consider the hierarchical, spatiotemporal aspects of a shared policy.

Figure 2 shows the graphical probabilistic model of a typical task/option-level interaction in our shared autonomy platform. This is a human-in-the-loop setup with human action/option a^H/o^H included by design. More specifically, in designing a shared autonomy policy, we categorize the variables of our shared autonomy framework into three categories:

1. $z_0^{T,A}$: task/option related variables of the shared autonomy, representing encoding of the task(s). This goes beyond the standard state s^T defined without considering any task-related variables. The augmented state is, therefore, $\widehat{s}^T = (s^T, z_0^{T,A})$. For simplicity of the notation, we will use s.

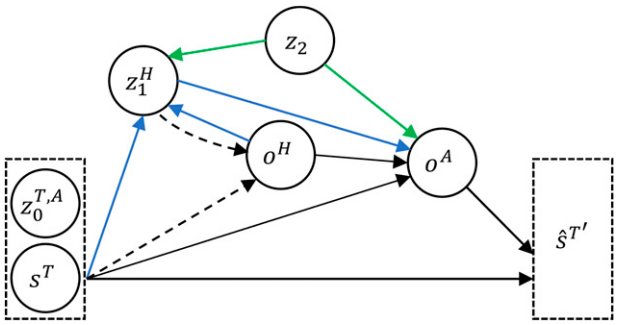

Figure 2. Graphical probabilistic model of interaction for shared autonomy. The dashed arrows are related to Bayesian goal inference and human analysis discussed in §3.1. The green arrows are related to including the pre-trained base model in the shared autonomy operation. The blue arrows designate the flow of signals to human.

2. z_1^H: human-related variables,
3. z_2: pre-training related variables.

The above three types of variables are introduced in analogy to how humans learn to perform a task. In particular, we incorporate our prior knowledge of the task (category 3), refine it to align with the specific task requirements (category 1), and customize our approach to executing actions in the present task execution (category 2).

3. Policy fine-tuning

As discussed in §1, we tackle the challenge of policy fine-tuning for shared autonomy settings. Given a base policy π_0^A, we introduce two fine-tuning strategies shown in Figure 1 (blocks VII and X), namely, short- and long-horizon policy fine-tuning. The overall fine-tuning algorithm is shown in Algorithm 1, and we will refer to its various parts as we introduce our methodologies, in §3.1 and 3.2.

The reason why we need two separate short- and long-horizon fine-tuning algorithms is routed in the practical needs of a comprehensive shared autonomy framework. Firstly, the framework has to react fast to human inputs or intentions by altering the policy in real-time. The reason is that a human can change their intended goal in less than 5 seconds, as revealed by our previous experiments Yousefi et al. (2022); this is even faster than a typical < 10-second short-horizon step used in our setup. We will discuss the span of short- and long-horizon steps further in §3.3. Real-time policy alterations cannot be realized in a deep-learning paradigm because of the nature of the algorithms involved, requiring large amounts of data, in addition to their deployment complexities and risks. Instead, we use a fast, short-horizon fine-tuning by deploying Bayesian analysis, as discussed in §3.1.

Secondly, with the ultimate goal of tuning to a human over a longer horizon, as well as, transitioning to higher levels of autonomy, the policy itself has to evolve to accommodate human preferences, nuanced task requirements and subtleties, and configuration changes. This can be realized in a deep RL paradigm by retraining the baseline policy using online human-in-the-loop data while the robot operates over longer horizons, as discussed in §3.2.

Finally, we note that these fine-tuning algorithms are designed jointly and are tied to the co-planning, shared encoding, and interaction channels, which, in turn, provides a streamlined strategy for the multi-horizon fine-tuning algorithms. From a control system's perspective, the multi-horizon fine-tuning design is partly inspired by the commonly used Cascade Control (Luyben, 1973; Yu, 1988), with the inner loop to control/respond to fast system dynamics and the outer loop to control the slow dynamics, with corresponding disturbance rejection.

3.1. Short-horizon policy fine-tuning through Bayesian Inference

We now present an online methodology to align the autonomous policy with the human agent. Following the Bayes rule, we write:

$$\pi_{sh}^A(a^A|g_H^*, a^H, s) = \frac{p(g_H^*|a^H, s)\pi_0^A(a^A|s)}{\sum_{a^A\in\mathcal{A}} p(g_H^*|a^A, s)\pi_0^A(a^A|s)}, \quad (1)$$

where π_{sh}^A is the posterior probability distribution, that is, the aligned shared autonomy policy; π_0^A is the prior, obtained from the original (pre-trained) autonomous policy. To obtain the likelihoods, we incorporate available information about the human-in-the-loop behavior as *evidence*. In our scenario, the evidence is the intended goal of the human, $g_H^* \in \mathcal{G}$. A short horizon of fine-tuning can be done as long as we have access to the evidence (in this formulation, human's inferred goal).

In general, a human is optimizing for a reward function $r^H(\boldsymbol{x})$, where $\boldsymbol{x} \in \mathcal{X}$ is a set of variables that a human might consider. Following our earlier work (Yousefi et al., 2022) and inspired by Luce's axiom of choice (Luce, 1977), we convert the reward into a probability over $\mathcal{X}$ by using Boltzmann model of noisily rational behavior (Bobu et al., 2020; Chris et al., 2007; Ziebart et al., 2008), where $p(\boldsymbol{x}) \propto \exp(\beta r^H(\boldsymbol{x}))$, with $\beta \in \mathbb{R}^-$ as the noise factor. As mentioned above, the candidate $\boldsymbol{x}$ for evidence in our scenario is the intended goal $g_H^* \in \mathcal{G}$. We obtain the probability distribution over the goal space as follows:

$$p(g_H|a^H, s) \propto \exp(\beta M|_{a^H}(s, \mathcal{G})), \quad (2)$$

where $M|_{a^H}(s, \mathcal{G})$ is the Manhattan distance between the current state s and goal space $\mathcal{G}$, given a particular action a^H. Considering $g_H^* = max(p(g_H|.))$ as the evidence, the likelihoods are obtained, as follows:

$$p(g_H^*|a^H, s) \propto \exp(\beta M|_{a^A}(s, g_H^*)), \quad (3)$$

where $a^A \in \mathcal{A}$ is any action permissible within the action space. Algorithm 1, lines 4-10 show the fine-tuning steps, followed by the shared action taken in lines 11-14.

3.2. Long-horizon policy fine-tuning through shared-RL

In the long-horizon fine-tuning, we start with a baseline[1] model trained using deep RL, in particular, the Proximal Policy Optimization (PPO) algorithm (Schulman et al., 2017). The reason we opted for this methodology is multifaceted, as explained below.

Firstly, for a system to deliver a sliding level of autonomy, as required from a shared autonomy framework for complex robotic tasks, it has to be equipped with an internal policy structure and a model of decision-making with the capacity to deliver a fully autonomous policy, which is also extendable to human-in-the-loop operations. This requires minimum assumptions about the dynamics of the tasks and the human. This is the motivation behind the careful design of the graphical probabilistic model (i.e., policy adapting) discussed in §2.3 and shown in Figure 2 with modular state variables. We will discuss the specific design of our setup in §4 and §5.

Secondly, as we will see in §7, the baseline, pre-trained policy will be fine-tuned during operation, as it interacts with different human and operational settings. Therefore, a similar policy structure and training methodology for the pre-trained model and the model to be fine-tuned is chosen to complete the end-to-end loop of Figure 1, from human-inspired planning (see §5 and further in Yousefi et al. (2022)) to the multi-horizon policy fine-tuning algorithms (see §2.2).

Finally, in choosing the specific model training algorithm, we opted for a model-free deep RL (i.e., learning-based) because of our minimum assumptions requirement about the dynamics of the tasks and the human. Moreover, since we will be interacting with the reward function elements directly, we opted to work directly with the policy to prevent degenerate policies, as this class of methods is more robust to reward re-shaping (Ho and Ermon, 2016). Therefore, we chose PPO, however, we adapted the algorithm with a new understanding for shared autonomy in our framework.

Algorithm 1: Algorithm for Policy Fine-Tuning.

Data: base policy π_0^A, current state s, human action a^H, goal space $\mathcal{G}$

sPPO (π_0^A: *base policy, sRB: shared rollout buffer*) :

// **Long-Horizon Fine-Tuning**

sRB ← initialize shared rollout buffer

for *episode* **to** N **do**

  Bayesian Inference (a^H, $\mathcal{G}$, s) :

    $p(g_H|a^H, s) \propto \exp(\beta M|_{a^H}(s, \mathcal{G}))$ // infer the intended goal

    return $g_H^* = max(p(g_H|.))$ // evidence

  Bayesian Fine-Tuning (g_H^*, $\mathcal{G}$, s) :

// **Short-Horizon Fine-Tuning**

    $p(g_H^*|a, s)$ from (2) // get the likelihoods

    $\pi_{sh}^A(a|g_H^*, s)$ from (1) // get the Bayesian posterior

    return π_{sh}^A

  Interact (π_{sh}^A, s) :

    perform action $\pi_{sh}^A(a^A|s)$ // taking action interactively

    follow Algorithm 2 to update $\boldsymbol{R}$ and s'

    return s' // next state

  sRB ← collect-shared-rollout-buffer() // collect the HITL samples

  if *episode* == *done* **then**

    Train (*sRB*) :

      while *not converged* **do**

        use shared rollout buffer sRB

        compute advantage estimates A_t using V_ϕ and $\boldsymbol{R}$

        for $t \leftarrow 1$ **to** T **do**

          compute surrogate objective: $L(\theta) = \min\left(\frac{\pi_{sh}^A(a_t|s_t)}{\pi_0^A(a_t|s_t)} A_t, \text{clip}\left(\frac{\pi_{sh}^A(a_t|s_t)}{\pi_0^A(a_t|s_t)}, 1-\epsilon, 1+\epsilon\right) A_t\right)$

        end

        update policy π_0^A using gradient ascent: $\theta \leftarrow \theta + \alpha \nabla_\theta \frac{1}{T} \sum_{t=1}^{T} L(\theta)$

        fit value function using regression: $\min \frac{1}{T} \sum_{t=1}^{T} (V_\phi(s_t) - V_{\text{target}}(s_t))^2$

      end

      return π_0^A // new base policy

  end

end

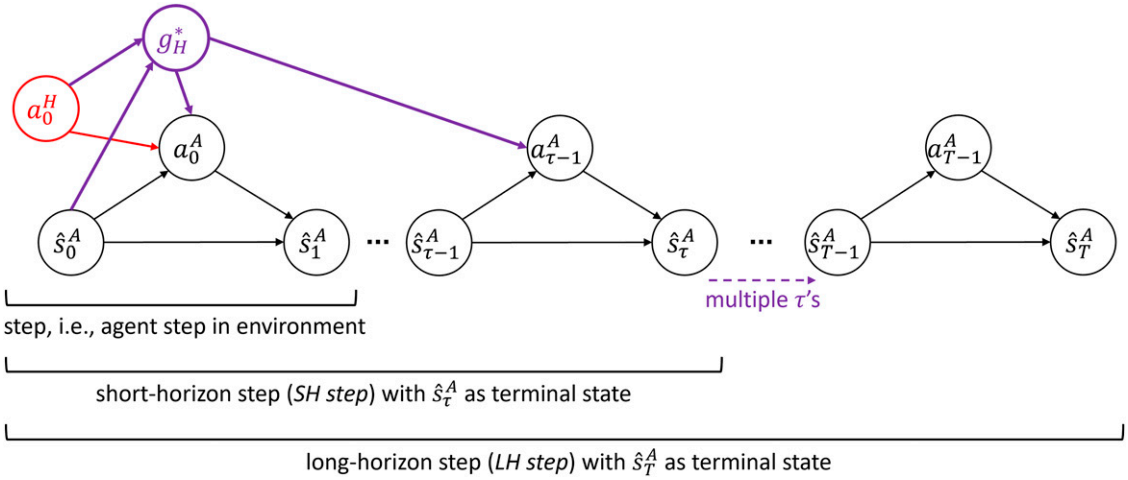

Figure 3. Comparison of different step sizes in our policy adapting and fine-tuning algorithms.

To this end, inspired by the successful implementations of this algorithm reported in literature, we defined the algorithm parameters according to the commonly used settings (see, e.g., the original paper (Raffin et al., 2021; Schulman et al., 2017)[2]). Notably, we have chosen a batch size of 5, learning rate of 1×10^{-3}, clip range of 0.2, and γ of 0.99 for a multilayer-perceptron (MLP) policy network with 2 layers of 64 nodes. We have used softmax in the last layer of the policy with a temperature of 5. Next, we introduce the reward function and how it is tied to the interaction channels by design.

It is worth noting that for the online human-in-the-loop training phase, we assume to have access to the *expert* human signal, whose policy is an optimal one. This realization is very important, as it enables us to keep the trajectories with human-in-the-loop in our priority *shared rollout buffer (sRB)* even in on-policy methods. The reward function for the training process also follows our hierarchical task/option-oriented shared autonomy blue-print and takes the following form:

$$R = c_1 R_1 + c_2 R_2 + c_3 R_3 + c_4 R_4 = \boldsymbol{c}^T \boldsymbol{R}, \qquad (4)$$

where the reward elements are:

- R_1: Robotic *task-related* rewards,
- R_2: *Deviation* of autonomous input from human input,
- R_3: Human's *explicit satisfaction* reward to an autonomous action; assigned by ± 1,
- R_4: Negative reward related to human *overriding* autonomous action.

Moreover, $\boldsymbol{c}$ contains the shared autonomy effectiveness coefficients. In other words, the distribution of those coefficients enforces the level of autonomy. The variable $\boldsymbol{c}$ is a design choice and depends on the task/application needs, data quality, human expertise, and mode of operation. We will discuss this further in §7.

In Algorithm 1, the outer loop starting in lines 1-2 and continuing in lines 16-29 show the implementation of the long-horizon fine-tuning.

3.3. Steps, horizons and episode choices

Based on §3.1-3.2, our policy adapting foundation and multi-horizon fine-tuning algorithms are designed to operate in three step sizes representing different time scales, and these are an important aspect of the proposed algorithms. We illustrate these steps and their inter-relation in Figure 3 and define them precisely as follows.

Step: this is an agent's step in its environment and represents the finest level of granularity in the system operation. For example, in our environment setup (shown in Figure 6(a)), a step is to move from one cell to another.

Short Horizon Step (*SH step*): it is delineated by the moment when the human gives a direct input to the system, ending with the terminal state $\widehat{s}_\tau^A$, right before the next direct human input. This step defines the duration of effect of the short-horizon fine-tuning (see §3.1). Multiple SH steps might exist in an episode of the overall task. In our user study in §6, the short-horizon will be defined by the experiment design choice.

Long Horizon Step (*LH step*): in our framework, this step represents the largest granularity duration for the long-horizon fine-tuning to update the agent's base policy and is another shared autonomy design choice. In our current implementation and considering the application task and the time constraints of a user study in a lab. environment, we choose the LH step to be equal to an episode with a terminal state of $\widehat{s}_T^A$. Note that the LH step is equivalent to a training step in the context of a long-horizon fine-tuning algorithm.

4. Shared autonomy implementation setup

The complete shared autonomy architecture implemented to conduct an experimental evaluation of the policy fine-tuning is shown in Figure 4: it is a comprehensive framework that implements hierarchical layers of reasoning, planning, execution, and control following the general architecture shown in Figure 1. The framework is designed such that the sequential multi-agent decision-making happens in the higher-level, discrete-time layers (Blocks V and VI in Figure 4). For these purposes, we created our

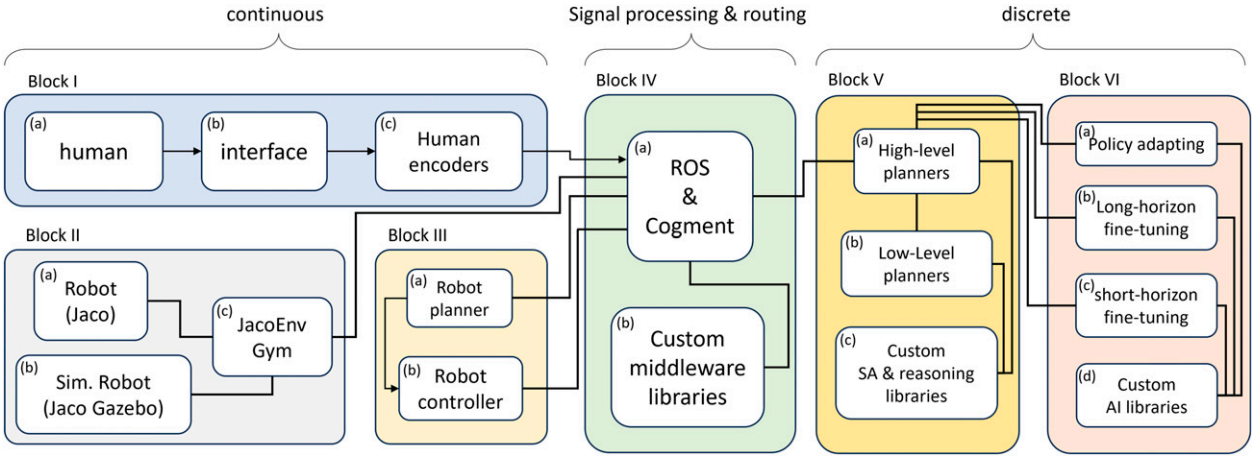

Figure 4. Shared autonomy setup implementation details.

custom hierarchical models and interfaces for shared autonomy high- and low-level planning, arbitration, and task and environment modeling. We also implemented custom libraries for various processes involved in a shared autonomy platform. The action space is tuned to the application, which will be discussed in §5.

Algorithm 2: Algorithm for Human Interaction.

Data: Autonomous base policy π_0^A, Controller button configuration $\boldsymbol{b}$

1 Human action $a^H \leftarrow \boldsymbol{b}_3$
2 $probs \leftarrow \pi_0^A(a^A|s)$
3 $a^A = argmax(probs)$
4 **if** $a^H == \varnothing$ **then**
5 $\quad R_2 \leftarrow 0$
6 **else**
7 $\quad e \leftarrow$ normalized deviation from a^H
8 $\quad R_2 \leftarrow -|e|$
9 **end**
10 **if** $\mathbf{b}_2 \neq \varnothing$ **then**
11 $\quad$ **if** $b_{21} \neq \varnothing$ **then**
12 $\quad\quad R_3 \leftarrow 1$
13 $\quad$ **if** $b_{20} \neq \varnothing$ **then**
14 $\quad\quad R_3 \leftarrow -1$
15 **if** $b_1 \neq \varnothing$ **then**
16 $\quad$ `Override Protocol(`a^H_{t+1}, *probs*`)`**:**
17 $\quad\quad R_4 \leftarrow 1$
18 $\quad\quad$ return to s
19 $\quad\quad$ **if** a^H_{t+1} *is not None* **then**
20 $\quad\quad\quad \pi^A_{sh} = \mathbf{1}(a^H_{t+1})$
21 $\quad\quad\quad$ proceed with Algorithm 1
22 $\quad\quad$ **else**
23 $\quad\quad\quad$ sorted actions = argsort(-probs)
24 $\quad\quad\quad$ pick next best action
25 $\quad\quad$ **end**
26 **end**

Moreover, as the hallmark of shared autonomy, Block I is dedicated to handling the human agent-in-the-loop. The human and the robot are interfaced using custom libraries. The execution of the high-level commands happens in the lower-level continuous-time layers (Blocks II and III in Figure 4). The latter includes trajectory planning and reference tracking control. The robotic platform employed to conduct the experiments is the Kinova Jaco-2 assistive arm,[3] shown in Figure 5(b). As noted in §2.2, the seamless transition between continuous and discrete time is crucial for the success of the shared autonomy platform, and is enabled by the "Shared encoder" L1 and L2 layers in Figure 1. These processes are implemented in Block IV and VIII, respectively, in Figure 4. The middle-ware to enable signal routing is ROS or Cogment (Gottipati et al., 2023) depending on the mode of the setup, as enabled by our custom libraries. Cogment platform is an open-source framework built on a micro-service architecture for running different kinds of RL, multi-agent RL and human-in-the-loop learning applications. Table 1 lists the correspondence between the implementation blocks in Figure 4 to architecture blocks in Figure 1.

The control interface is shown in Figure 5(c), which is used by the human to interact with the system through the following interaction channels:

- Two active input channels: channel ch_1 for direct action buttons $\boldsymbol{b}_3 = \{b_{30}, \ldots, b_{33}\}$ corresponding to actions 0, 1, 2, 3 of our action space, and channel ch_2 for the *override* command by pressing button b_1;
- Passive channel ch_3 by pressing buttons $\boldsymbol{b}_2 = \{b_{20}, b_{21}\}$ to relay the negative or positive satisfaction feedback to an autonomous action.

Details of the online interaction process are shown in Algorithm 2. We provide a high-level description of the algorithm here. Specifically, the reward term R_2 depends on whether or not the human action is given, and its value is assigned accordingly (lines 1-8). The value of R_3 is updated based on $\boldsymbol{b}_2$ (lines 9-10).

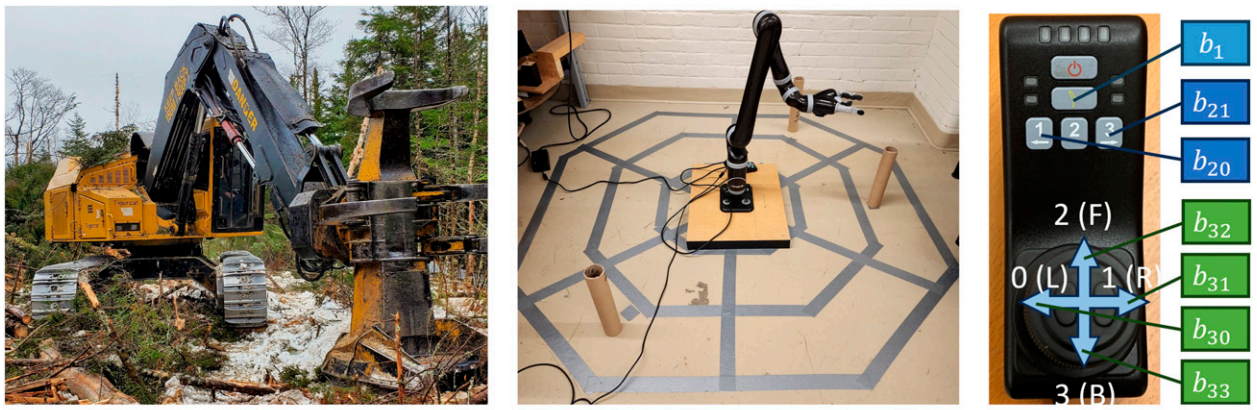

Figure 5. (a) Example of robot/machine: a feller-buncher machine; (b) Kinova Jaco-2 assistive arm in our grid world; (c) Human feedback and override implementation on Jaco-2 assistive arm controller with button labels for different interaction channels.

In case of *override* (non-zero b_1), the algorithm looks at whether or not an explicit human action is given: if so, it follows the human action, otherwise, takes the next best action (lines 11-21).

5. Case study application

Since the motivating application for our work is the operation of timber-harvesting machines, our experiments are set up to showcase the shared autonomy framework for an emulation of planning the operations of a feller-buncher. This machine, shown in Figure 5(a), is equipped with a robotic arm (the crane) with a specialized end-effector. A typical maneuver in the operation at a particular location involves felling and grabbing the tree(s), followed by manipulating and placing the felled trees in a storage location in the working area of the machine. This maneuver sequence is repeated until all trees within the crane reach have been bunched at the storage location, thus, completing a full operation cycle at a fixed base location. The overarching objective for our work is to increase the level of autonomy of machines, such as the feller-buncher, in a safe manner, with the ultimate goal of full autonomy. There has been recent progress towards this goal for different types of timber-harvesting machines (Ayoub et al., 2023; Jebellat and Sharf, 2023; Löfgren, 2009; Song and Sharf, 2022; Westerberg, 2014), considering the autonomy at the task level of operation. In line with our earlier contributions in Yousefi et al. (2022, 2023), our focus is on the planning problem at the cognitive level in a multi-agent human-in-the-loop shared autonomy framework.

5.1. Environment encoding

We describe how we realize Block II in Figure 1 for our case study application. Our planning algorithm is designed for the environment model illustrated in Figure 6(a): it shows the decomposition of the environment and the objects, that is, trees, around the machine/robot into a *region* (bounded by the minimum and maximum reach of the crane) and *cell*s which discretely encode the location of the machine end-effector p_{EE}. The planning algorithm utilizes a concept we termed the "Envelope of Manipulation," denoted as E^M, which comprises a curve connecting "key points" represented as E_i (refer to Figure 6(a)), these associated with clusters of trees. These key points are organized within a set called $\mathcal{E}$. Drawing from our field observations and adopting a human operator's perspective, we identified two primary high-level options within the option space, denoted as $\mathcal{O}$: 1) $O_1 \in \mathcal{O}$ encompasses the motion along the envelope between two specific cells, while 2) $O_2 \in \mathcal{O}$ encodes the operations occurring within each cell. The shape of the envelope E^M may vary, but it can be effectively approximated as a circular arc. Consequently, it becomes feasible to represent the complete operating cycle of a feller-buncher as a sequence of the two aforementioned options. Thus, the problem of robot task planning boils down to optimizing this sequence of options. It is worthwhile to draw the analogy between the large-scale robot/machine in Figure 5(a) and Jaco-2 arm in Figure 5(b) (introduced earlier) for the purpose of evaluating our shared autonomy algorithms situated in a similarly defined environment characteristics and task definitions.

Table 1. Block to Block Mapping of Implementation Blocks in Figure 4 to Architecture Blocks in Figure 1.

Block in Figure 1	Block in Figure 4
I, III	I
II	II
V	V
VI	VI
VII, IX, X	VI
VIII	III, IV, V
IV	I-c

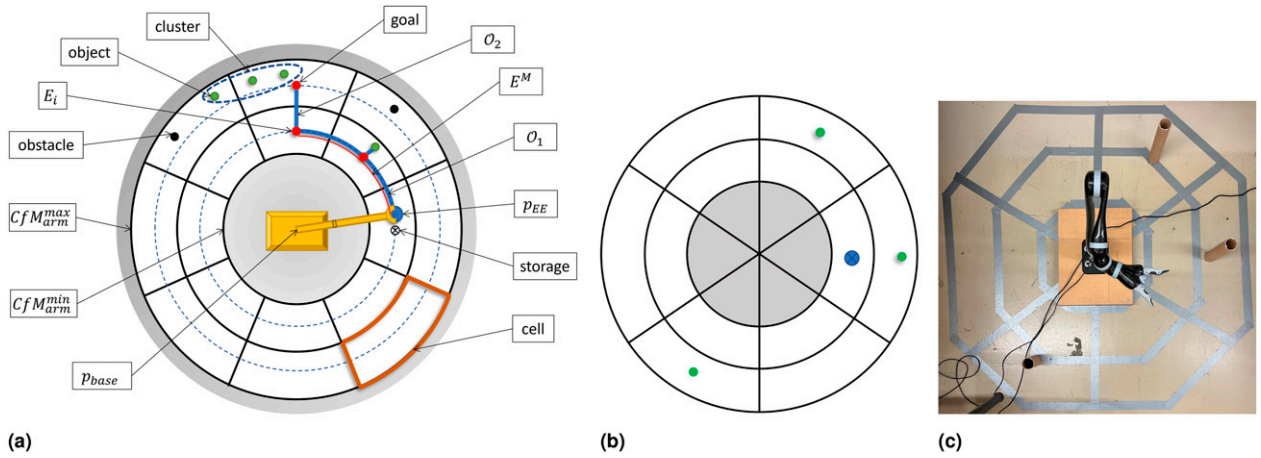

Figure 6. (a) Setup for robot task planning with details; (b) schematic of starting environment configuration for the fine-tuning examples; (c) environment implementation in our setup with Jaco-2 arm.

5.2. Hierarchical structure of planning

Here, we discuss how to approach Blocks III, V, and VI in Figure 1. It is noted that in Yousefi et al. (2023), we laid out a detailed hierarchical MDP formulation of the planning of tasks for a feller-buncher robot. The reader is referred to our previous work for the details of the models, which we summarize here briefly for completeness. In particular, we have designed a three-layer, hierarchical planning architecture, as follows:

5.2.1. Interaction layer. In this layer, the interaction between the agents of the system happens. Following the model shown in Figure 2, the elements of the MDP formulation are:

- State: the "state" is comprised of: 1) the human action or option (a^H or o^H) and 2) the state defined for the task(s) layer (z_0),
- Action: the actions correspond to the left-wise, right-wise, forward and backward motions of the end-effector, encoded discretely with 0 (L), 1 (R), 2 (F), 3 (B), respectively.

5.2.2. Task layer(s). In this layer, we focus on the robotic task and task-related design variables:

- State: defined for options O_1 and O_2, the state is comprised of the 3 elements: 1) discrete position of the end-effector: this basically assigns a discrete integer to a particular cell in the radial and angular direction, 2) the number of the objects carried in the end-effector, and 3) discrete angular distance to the goal space.
- Action: defined identically to that of the Interaction layer.

Although in the present application, we focus on options/tasks O_1 and O_2, this layer can accommodate other tasks, depending on the application.

5.2.3. Manipulation layer. In this layer, the high-level planning is executed on the robotic arm using standard robotic trajectory planning and control tools. This layer is implemented in Block III in Figure 4.

6. User study design

We present a series of results to demonstrate fine-tuning and alignment of the pre-trained policy for the environment shown in Figure 6. We have employed a deep RL methodology discussed in §3.2.

6.1. Experiment design objectives

Our human-in-the-loop experiments are purposefully designed to showcase the interplay and effect on performance of the following aspects of the proposed shared autonomy framework: a) different design elements in our policy adapting algorithm, b) long- and short-horizon tuning strategies, and c) interaction channels, as schematically shown in Figure 7. These experiments constitute a practical demonstration of our first and second contributions, and particularly provide the necessary material for our third contribution (see §1.2). The specific objectives are:

- To analyze the short-horizon fine-tuning and how the posterior probabilities of shared autonomy actions π_{sh}^A are affected by a human action (see Algorithm 1 line 9).
- To investigate the interplay between short-horizon and long-horizon fine-tuning. Since this aspect is closely tied to human's intentions and comfort, we are curious to see how this interplay affects the behavior of our shared policy.
- To analyze the effects of the interaction channels, progressively over multiple training steps, that is, task episodes, while being subject to multi-horizon fine-tuning. We carry out an ablation study to identify how each of these elements help to shape the policy and

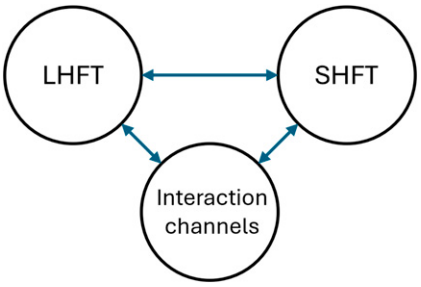

Figure 7. Schematic of the high-level objectives in our user study, the "interplay" between long- and short-horizon fine-tuning (i.e., LHFT and SHFT) and shared autonomy interaction channels.

what purpose they serve. Moreover, since the reward elements are designed to correspond to the interaction channels, this study also provides insights into the effects of interaction channels on the system behavior. In other words, the efficacy of reward shaping, which is by design tied to the interaction channels, is addressed in these experiments.

6.2. Multi-horizon fine-tuning user study and ablation design

Following the objectives laid out in §6.1, for this study, we conduct a total of five different experiments. Depending on the details of each experiment, the human can interact with the robot in this episodic task through different interaction channels. In experiments 1, 2, and 3 (which will referred to as EXP-1, EXP-2, and EXP-3), the human has a passive role, unless in case of an override, which will be described. In experiments 4 and 5 (EXP-4 and EXP-5), the human can give direct inputs as per specific instructions. Note that while each successive experiment adds a new condition, *all* conditions prior to that experiment hold.

The overall task of the human-robot system is to pick up objects in an environment shown in Figure 6(b) and (c) (cardboard tubes) and unload them in a designated storage cell. The hand controller in use by the human agent to give feedback and override is shown in Figure 5(c). In all experiments, we always start with pre-trained model and fine-tune it in accordance with the settings of each experiment. The following description provides the details of each experiment.

Experiment 1 (EXP-1): In this experiment, we demonstrate the importance of the override input from the human operator. In fact, without this minimal input during the policy fine-tuning, the system may fall into infinite loops and fail to complete the task. Infinite loops are always a possibility with any online policy fine-tuning. This problem gets exacerbated when we have access to human user for a rather limited time. We therefore allow the human to use the override channel, after any action from the autonomy, thus activating the R_4 term in the reward function. The human is not to give any direct input after the override, so that the autonomy continues to its next best action(s), following Algorithm 2. Since there is no direct human input (after the override or otherwise), there is no short-horizon fine-tuning. Therefore, this experiment includes only R_1 and R_4 reward terms in total.

Experiment 2 (EXP-2): In this experiment, we augment the previous experiment by adding the human satisfaction feedback, R_3, to the reward function, where the human can provide positive and negative feedback for each action taken by the autonomy by pressing appropriate buttons on the controller. The human is still not to give any direct input, and hence, no short-horizon fine-tuning. Also, the human is not to give any direct input after the override. Therefore, this experiment includes R_1, R_3, and R_4 reward terms in total.

Experiment 3 (EXP-3): We allow the human to give direct inputs, but only right after overriding the autonomy. There is still no short-horizon fine-tuning. In this case, the posterior probability of the direct action is updated according to Algorithm 2. Therefore, this experiment includes all the designed reward terms R_1, ..., R_4.

For the next two experiments, we allow additional human input in the loop, beyond the override option, but limited to direct input only right after an unloading (i.e., beginning of a new plan to pick another object), which marks the beginning of a *SH step* (see §3.1 and §3.3). The reason for this experiment design choice is twofold: 1) based on our setup, a single human input is sufficient for the inference algorithms to work, and 2) we wanted to avoid overloading the human operator with too many inputs through different interaction channels. The only difference between the two experiments is the activation of the short-horizon fine-tuning. Thus:

Experiment 4 (EXP-4): In addition to the previous channels, the human is allowed direct input as described above, but we do not enable the short-term fine-tuning.

Experiment 5 (EXP-5): The same conditions as in EXP-4, but we enable short-horizon fine-tuning (see §3.1 and Algorithm 1). Therefore, here we have both short- and long-horizon fine-tuning.

The key features of the above five experiment designs are summarized in Table 2, where the red box around a checkmark highlights the new variable/change introduced in the experiment relative to the previous one. The nature of the experiments are within-subjects, where we progressively (see Table 2) add one more variable/change and look at its effects through the metrics introduced next. For each user, we collect user data for $N_s = 10$ episodes per experiment and therefore, we have access to the evolution of the policy per user/per experiment/per LH step.

6.3. Metrics of study

The overarching metric of this study is the algorithm(s) efficiency/performance evaluated through the quality

Table 2. Details of design of experiment for Ablation/baseline study. "R_4 w/ a^H" ("R_4 w/o a^H") means that the user is (not) allowed to enter direct input after override. "a^H input" means that the user is allowed to give direct input right after each unloading. "SHFT" and "LHFT" stand for short- and long-horizon fine-tuning, respectively. The red box around a check-mark indicates introducing that variable/change anew to each experiment after its previous one.

Experiment No.	R_1	R_2	R_3	R_4 w/o a^H	R_4 w/ a^H	a^H input	SHFT	LHFT
EXP-1	✓	-	-	✓	-	-	-	✓
EXP-2	✓	-	✓	✓	-	-	-	✓
EXP-3	✓	✓	✓	-	✓	-	-	✓
EXP-4	✓	✓	✓	-	✓	✓	-	✓
EXP-5	✓	✓	✓	-	✓	✓	✓	✓

of human-robot interactions and robotic task execution. For this purpose, we define the following *objective* metrics:

1. User Satisfaction: Arguably, one of the most important metrics to assess the success of a shared autonomy platform is user satisfaction. It is noted that during our experiments, a user is asked to give explicit feedback after *all* actions, regardless of the experiment type. Thus, we have access to human satisfaction directly through R_3 and we use this value as the user satisfaction metric. This metric also indicates how well the decisions made by the autonomous agent align to the preferences of the human, that is, preference matching. To evaluate the metric, we used the proportion of human's positive evaluations (i.e., satisfactions) per episode length.
2. User Overrides: Another important measure is how many times per episode length a user decided to override the autonomy, regardless of the subsequent action. We have access to this metric through R_4.
3. Task Return: The ultimate goal of the system is to get the task done and this metric measures the task performance in terms of its overall return.
4. Episode Length: Similar to the previous metric, we are also interested in seeing how the task as well as interaction performance are reflected in the episode length. Episode length is obtained by counting the actions to success (the end of the episode).

Recalling $N_s = 10$ of collected episodes, note that first episode (i.e., LH step 0) is primarily for the human to adjust to the new settings. We isolate the data for the i'th step of experiments for all users ($i \in \{0, ..., N_s - 1\}$), and compare them in pairs, as will be discussed in § 7.2.

Note that we *evaluate* the metrics for a complete episode, or in other words, a long-horizon step.

In our *subjective* studies, following best-practices recommendation (Schrum et al., 2020), we designed an original seven-point Likert scale with five items, summarized in Table 3, that each user completed after each experiment. We report Cronbach's alpha as a measure of reliability of the responses for each experiment (Collins, 2007) with a threshold of 0.7 to justify composite score from multiple items. In reporting the effect sizes, we use Cohen's d as our metric, which is the difference between two means divided by a measure of standard deviation for the data (Cohen, 1992).

6.4. Recruitment and procedure

To conduct our experiments, we have recruited 10 participants from McGill community, with mixed familiarity with our setup.[4] Prior to the experiments, we gave the users the opportunity to interact with the system under non-experiment related conditions. Subsequently, each participant conducted each experiment once for a total of 10 episodes per experiment, that is 50 episodes in total. The users started with EXP-1 onwards, in the same order for all participants. The progressive, not randomized, nature of the experiments in terms of added human-robot interaction channels helps the users to become progressively more comfortable with each added channel, as the experiments become cognitively more challenging.

6.5. Hypotheses

Figure 8 shows how intertwined the short- and long-horizon fine-tuning and interaction channels are. The interplay between these factors (in line with the objectives in §6.1 and Figure 7) is where we hypothesize and evaluate. In this light and in conjunction with the experiment design, we list our hypotheses to be tested:

Hypothesis 1. *Regarding the effects of short-horizon fine-tuning and its interplay with the long-horizon fine-tuning, we hypothesize that:*

*H1-a) in the early long-horizon step (*i.e., *LH step 1), short-horizon fine-tuning (EXP-5) is beneficial over its counterpart without it (EXP-4) as well as no-tuning with any other interaction channel combination (EXP-1,2,3).*
H1-b) The use of short-horizon fine-tuning does not degrade the performance of long-horizon fine-tuning.

Hypothesis 2. *Regarding the interplay between interaction channels and multi-horizon fine-tuning, we hypothesize that there is a correlation between long-horizon policy fine-tuning (and thus, policy updates) and interaction channels, namely:*

H2-a) human's explicit satisfaction per action;
H2-b) override without subsequent action;
H2-c) override with subsequent direct action;
H2-d) human's direct action per design choice;

7. Results and discussion

In our shared autonomy platform with multiple interaction channels as well as two stages of policy fine-tuning acting in two horizons (see §3), it is crucial to look at the policy evolution in each experiment and how it is linked to interaction quality between the agents of the system towards completing the task. It is with this perspective that we approached the post-processing of raw data which necessitated the *intra*-user as well as *inter*-user analyses.

We begin by illustrating the short-horizon fine-tuning with a sample single action-step result. Then, we move to examine the results to (in)-validate our hypotheses, first by assessing the objective metrics and then the subjective metrics.

Table 3. User experience survey items.

Item	Statement
1	I had a sense of control
2	The system assisted me in achieving my tasks
3	The communication between me and the system was effective
4	The system respected my preferences and decision-making choices
5	I was satisfied with the user interface and interaction design of the system

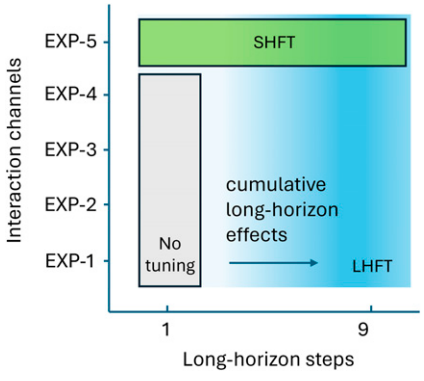

Figure 8. Interplay of no-tuning, short- and long-horizon fine-tuning, and interaction channels (through different experiments).

7.1. Human inference results and short-horizon fine-tuning

We first zoom in and discuss the short-horizon fine-tuning and alignment mechanism using the methodology discussed in §3.1 and shown in Algorithm 1.

We start with a pre-trained, intentionally sub-optimal model, to showcase the fine-tuning results. We look at a specific scenario. At the initial end-effector position, $s_0 = [0, 0]$, the base policy returns the following priors:

$$\pi_0(a_0^A|s_0) = [0.142, 0.673, 0.087, 0.098],$$

which means that the autonomous action is 1 (i.e., turn right). At this instance, the human, however, gives $a^H = 0$ (i.e., turn left). Given that the goal space is $\mathcal{G} = \{[1,1],[1,0],[1,4]\}$, the output of Bayesian goal inference is $g_H^* = [1, 1]$. Given this evidence, we obtain the likelihoods in the Bayesian sense, as follows:

$$p(g_H^*|a, s) = [0.495, 0.009, 0.495, 0],$$

which leads to the following posteriors:

$$p(a|g_H^*, s) = [0.589, 0.051, 0.36, 0],$$

which indicates that the aligned autonomous action is $a_{sh}^A = 0$. Thus, the short-horizon fine-tuning strategy produced a perfect alignment with human's intention. Figure 9 shows a summary of the short-horizon fine-tuning process and how the posterior probabilities of the actions are fine-tuned to the human preference.

Next, we present the results of the multi-horizon fine-tuning with multiple users.

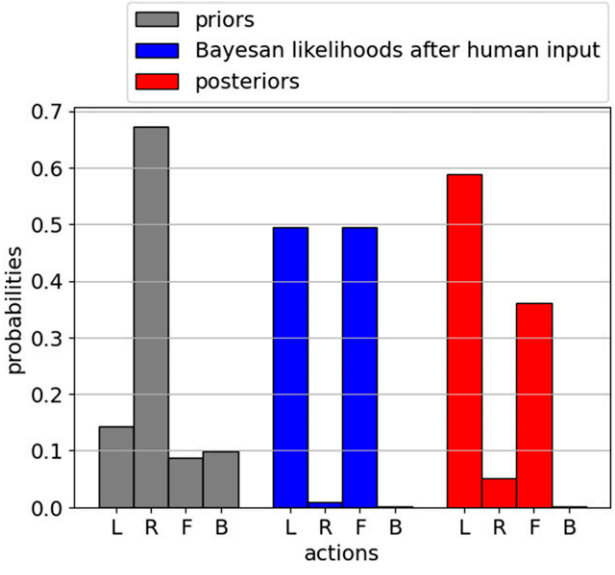

Figure 9. Bar-plots to visually present the evolution of probability distribution over actions during short-horizon fine-tuning process. The actions are encoded discretely with 0 (L), 1 (R), 2 (F), 3 (B), respectively.

7.2. Objective results

To report our objective results, we highlight the fact that our experiments, by design and in tandem with our algorithms, demonstrate two types of evolution: 1) long-horizon fine-tuning, or in other words, policy model updates in each training LH step of each experiment, 2) progressive addition of interaction channels from one experiment to another. With this in mind, we aggregate and compare the objective metrics recorded at LH step 1 and LH step 9, the last LH step, of each experiment and compare each to its respective counterpart in any other experiment. The goal is to deduce the evolution of policy over the training steps (i.e., LH steps) and how it was affected by the interaction channels. Then, we use these data to assess the hypotheses objectively. We use $\boldsymbol{c} = [1, 4, 8, 16]$ as the reward function weights. For both LH steps, we first report Repeated Measures ANOVA results, and in case of statistical significance, we then report post-hoc analysis with Holm corrections for the important pairs of our experiments. We use 0.05 as the statistical significance threshold. As will be discussed next, Tables 4 and 5 contain the reported statistical results. In these tables, each row is dedicated to a metric of our study, for which we report post-hoc *p*-value and the means for each of the designated pairs.

7.2.1. For LH step 1. In this LH step, the Repeated Measures ANOVA shows 1.55×10^{-6}, 0.0005, 1.48×10^{-6}, and 0.007 for each of the metrics listed in §6.3, respectively. Hence, we proceed with post-hoc analysis, the results of which are shown in Table 4. We observe that at this early stage of long-horizon fine-tuning (LHFT), the short-horizon fine-tuning (SHFT) shows a significant effect across all metrics and all experiments, except for EXP-1 and EXP-2 for the "episode length" metric. This signals the success of our short-horizon fine-tuning algorithm at the early stage of long-horizon training: as can be seen from the mean values reported in Table 4, SHFT contributes to the higher user satisfaction, fewer overrides, as well as higher task return. As expected at this early stage of long-horizon fine-tuning, we do not observe any significant effect of additional interaction channels. Based on the above observations, we next assess our hypotheses objectively.

Regarding Hypothesis 1, we found full support for H1a, where at the early stage of long-horizon fine-tuning, EXP-5 outperforms all other experiments at LH step 1.

7.2.2. For LH step 9. In this LH step, the Repeated Measures ANOVA shows 0.009, 0.2299, 0.0034, and 0.0464 for each of the metrics listed in §6.3, respectively, which indicates that the measures "user satisfaction," "task return," and "episode length" are of statistical significance. Results of post-hoc analysis for this LH step are shown in Table 5.

Regarding Hypothesis 2, we found support for this hypothesis and, rather surprising dynamics between interaction channels and fine-tunings. Here we shed light on the results from two perspectives: 1) from the "user satisfaction" point of view in Table 5, we observe that long-horizon fine-tuning *catches up* in EXP-1 and EXP-4, where we have minimal and maximal interaction channels. 2) from the "task return" of view in Table 5, we observe that additional interaction channels after EXP-1 seem to contribute positively. The other measures do not show any significance. Therefore, we have found support for H2. We also found partial support for Hypothesis 1b, and it depends on the interaction channels. This result also demonstrates how intertwined the short- and long-horizon fine-tuning and interaction channels are. Additionally, in response to the experiments, the user-related objective metrics (i.e., user satisfaction and overrides) and task-related objective metrics (i.e., task return and episode length) do not exhibit aligned, similar behavior, in other words, improving one group does not inherently imply improvements in the other; hence, there does not exist an alignment between different objective metrics.

7.3. Subjective results

To evaluate our hypotheses subjectively, we first report that Cronbach's alpha values for the five experiments are 0.912, 0.898, 0.864, 0.9, and 0.801, respectively, which indicate

Table 4. Post-hoc p-values for Pairs of Experiments at **LH step 1** of the Long-Horizon Fine-Tuning (LHFT) of the Policy Vs. the Experiments With Short-Horizon Fine-Tuning (SHFT) Activated. See Table 2 for the List of Experiments. "NS" stands for Not-significant.

metric	value	LHFT (EXP-1) - LHFT + SHFT (EXP-5)	LHFT (EXP-2) - LHFT + SHFT (EXP-5)	LHFT (EXP-3) - LHFT + SHFT (EXP-5)	LHFT (EXP-4) - LHFT + SHFT (EXP-5)
User satisfaction	*p*-value	$<$**0.001**	$<$**0.001**	**0.001**	**0.004**
	Mean	(0.423, 0.778)	(0.530, 0.778)	(0.492, 0.778)	(0.538, 0.778)
User overrides	*p*-value	**0.018**	$<$**0.018**	$<$**0.001**	**0.013**
	Mean	(0.073, 0.007)	(0.104, 0.007)	(0.133, 0.007)	(0.120, 0.007)
Task return	*p*-value	$<$**0.001**	**0.032**	$<$**0.001**	**0.012**
	Mean	(0.047, 0.091)	(0.068, 0.091)	(0.067, 0.091)	(0.067, 0.091)
Episode length	*p*-value	NS	NS	**0.010**	**0.020**
	Mean			(16.9, 12.4)	(17.4, 12.4)

Table 5. Post-hoc p-value for the Pairs of our Study at **LH step 10** of the Long-Horizon Fine-Tuning (LHFT) of the Policy Compared Compared to the Case With Short-Horizon Fine-Tuning (SHFT) Activated. See Table 2 for the List of Experiments.

metric	value	LHFT (EXP-1) - LHFT + SHFT (EXP-5)	LHFT (EXP-2) - LHFT + SHFT (EXP-5)	LHFT (EXP-3) - LHFT + SHFT (EXP-5)	LHFT (EXP-4) - LHFT + SHFT (EXP-5)
User satisfaction	*p*-value Mean	NS	**0.046** (0.547, 0.736)	**0.011** (0.538, 0.736)	NS
User overrides	*p*-value	-	-	-	-
Task return	*p*-value Mean	**0.046** (0.053, 0.078)	NS	NS	NS
Episode length	*p*-value	NS	NS	NS	NS

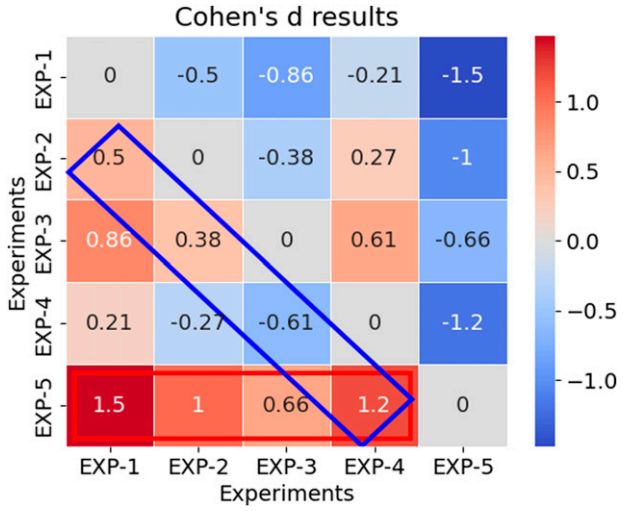

Figure 10. Cohen's d results on overall Likert scale scores. See Figure 11 for extended user responses to Likert questionnaire items.

strong internal consistency. The detailed results for each of the five Likert questionnaire items can be found in Appendix 9. Therefore, we proceed to evaluate our hypotheses. On the overall Likert scale scores (i.e., aggregated and averaged), we report Cohen's d values to compare the responses in Figure 10.

We consider 0.2, 0.5, and 0.8 as the thresholds for small, medium and large effect size, respectively. Based on Figure 10, we particularly focus on interpreting two sets of results which are bordered by red and blue rectangles, as follows:

(i) The red rectangle in Figure 10 shows the comparison of EXP-5 to all other experiments. We report effect sizes of 1.5 (large), 1 (large), 0.66 (medium), and 1.2 (large) for EXP-5 compared to EXP-1, EXP-2, EXP-3, and EXP-4, respectively, corresponding to the effects of interaction channels progressively added (see Table 2. Therefore, in subjective assessment of our hypotheses, regarding Hypothesis 1, while a step-wise comparison in long-horizon sense is not feasible with the current questionnaire, we report a large effect size supporting positive contribution of short-horizon fine-tuning to the overall subjective evaluation of the performance of the shared autonomy framework. Therefore, we found subjective evidence in partial support of Hypothesis 1a only.

(ii) The blue rectangle in Figure 10 shows the comparison of each experiment to its previous one. We report 0.5 (medium), 0.38 (small), −0.61 (medium), and 1.2 (large) effect size for pairs of experiments (EXP-2, EXP-1), (EXP-3, EXP-2), (EXP-4, EXP-3), (EXP-5, EXP-4). We particularly observe a noticeable decrease with addition of direct human input channel (EXP-4). This is in line with our observations during the tests when one participant commented that "if there is a direct input channel but the robot violates, then it is not helpful." This signals the fact that humans may not be patient enough with a sub-optimal policy in the early stages of fine-tuning if their direct actions are violated. Therefore, we have also found evidence for Hypothesis 2 with various effect sizes.

8. Conclusions and future work

In this paper, we introduced a comprehensive shared autonomy architecture with multi-horizon fine-tuning and modular design elements, starting from theoretical decision-making algorithms to a user study on a physical robot platform. Our approach to the decision-making problem between human and autonomy, as the agents of the system, is built on the foundation of multi-agent system design for human-robot interaction and decision-making arbitration. Moreover, it is designed as a hierarchical shared autonomy policy adapting based on gamified human-robot interactions and careful design of decision-making elements. This approach is particularly relevant to task-based, high-level robotic operations, where the human-robot interaction is cognitively challenging. Moreover, due to the inherent differences between the biological (human) and artificial intelligence, we strategically incorporate multiple interaction channels at different points of the end-to-end loop to enable a seamless interaction between the two agents of the system.

Next, we designed short- and long-horizon policy fine-tuning and alignment algorithms to adapt a pre-trained

shared policy to new operating conditions and to the human agent. The short-horizon fine-tuning acts fast using Bayesian analysis adapting to human's spatiotemporal goals. The long-horizon fine-tuning updates the baseline policy using our custom deep RL algorithm. We also discussed the implementation details of our physical setup which constitutes our test-bed for human-in-the-loop user studies.

To showcase the strength of our framework and study the effects of different design elements, we conducted a human user study with our test-bed. The human-in-the-loop experiments were purposefully designed to showcase the interplay and effect on performance of the design elements in our policy adapting algorithm, long- and short-horizon tuning strategies and interaction channels.

We observed complex intertwined effects of short-and long-horizon fine-tuning and interaction channels, demonstrating the importance of a comprehensive design approach to shared autonomy and our advances on the theoretical and algorithmic side as well as implementation and physical side. Our results indicate that a careful design of policy adapting structure on the foundation of multi-agent system and decision-making arbitration is crucial to fine-tune and align the autonomy with the preferences of the human. As we observed, this goal is achievable through: a) careful design of interaction channels placed at strategically chosen points of the end-to-end loop, and b) hierarchical multi-horizon fine-tuning algorithms designed in-tune with the other design elements for an efficient interplay between them.

The future directions are numerous, given the potential of this novel framework. Specifically, a number of improvements can be considered to the formulation of design elements of the shared autonomy. Starting with the environment encoding, it would be helpful to have generalizable guidelines (beyond the type of tasks discussed) on efficient design of the environment representations and their encoding process for shared autonomy. Moreover, the decision-making model could be improved to encompass more adversarial scenarios, as well as multiple human-robot systems interacting with each other. Fine-tuning algorithms can then be extended to the listed scenarios. In terms of implementation, a more realistic robot environment and a more diverse set of tasks can be explored. It is worth noting that the length of time required for each user to complete all of the designed experiments is long (∼ 3 hours), pointing to the need for consideration of the longitudinal aspects of the study. Finally, although the application we considered for our shared autonomy framework is the operation of robotic arms, such as timber-harvesting machines, we suggest that its application to autonomous vehicle driving may offer an alternative pathway to full autonomy, this through interaction with the human driver, co-learning, and shared autonomy.

Acknowledgments

We would like to thank Professor Dylan P. Losey for his early contributions to this work. This work was supported by the National Sciences and Engineering Research Council (NSERC) Canadian Robotics Network (NCRN). The authors also acknowledge the valuable contributions of Rafid Saif to the development of the shared autonomy setup.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This work was supported by the NSERC Canadian Robotics Network (NCRN); 249169 NCRN/Ind. NETGP 508451 X-249640.

ORCID iDs

Ehsan Yousefi https://orcid.org/0000-0002-8791-2998
Mo Chen https://orcid.org/0000-0001-8506-3665

Notes

1. Further details regarding the pre-training process with human data can be found in our previous work (Yousefi et al., 2023).
2. https://stable-baselines3.readthedocs.io/en/master/modules/ppo.html\#parameters. Last accessed on 2024.08.27.
3. https://www.kinovarobotics.com/product/gen2-robots. Last accessed on 2024.04.05.
4. Ethics approval obtained from Research Ethics Office, McGill University, under REB# 21-07-071.

References

Annaswamy AM, Johansson KH and Pappas G (2024) *Control for Societal-Scale Challenges: Road Map 2030," in IEEE Control Systems Magazine* 44(3): 30–32. doi: 10.1109/MCS.2024.3382376.

Ayoub E, Levesque P and Sharf I (2023) Grasp planning with cnn for log-loading forestry machine 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May 2023 - 02 June 2023, 11802–11808. DOI: 10.1109/ICRA48891.2023.10161562.

Bobu A, Scobee DRR, Fisac JF, et al. (2020) *LESS Is More: Rethinking Probabilistic Models of Human Behavior*. New York, NY: Association for Computing Machinery, 429–437.

Chen X, Wang C, Zhou Z, et al. (2021, May) Randomized ensembled double q-learning: learning fast without a model International Conference on Learning Representations, Vienna, Austria.

Chris L, Rebecca R, Baker CL, et al. (2007, August), Goal inference as inverse planning Proceedings of the Annual Meeting of the Cognitive Science, Nashville, Tennessee, USA.

Cohen J (1992) Statistical power analysis. *Current Directions in Psychological Science* 1(3): 98–101. DOI: 10.1111/1467-8721.ep10768783.

Collins L (2007) Research design and methods. In: JE Birren (ed.) *Encyclopedia of Gerontology* (2nd edition). New York:

Elsevier. 433–442. DOI:DOI: 10.1016/B0-12-370870-2/00162-1.

Dragan AD and Srinivasa SS (2013) A policy-blending formalism for shared control. *The International Journal of Robotics Research* 32(7): 790–805.

Gottipati SK, Nguyen L-H, Mars C and Taylor ME (2023) Hiking up that HILL with Cogment-Verse: Train & Operate Multi-agent Systems Learning from Humans. *In Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems (AAMAS '23)*. Richland, SC: International Foundation for Autonomous Agents and Multiagent Systems, 3065–3067.

Guo C, Sentouh C, Popieul JC, et al. (2019) Predictive shared steering control for driver override in automated driving: a simulator study. *Transportation Research Part F: Traffic Psychology and Behaviour* 61: 326–336. DOI: 10.1016/j.trf.2017.12.005.

Ho J and Ermon S (2016) Generative adversarial imitation learning. In: D Lee, M Sugiyama, U Luxburg, et al. (eds) *Advances in Neural Information Processing Systems*. Newry: Curran Associates, Inc, Vol. 29.

Javdani S, Admoni H, Pellegrinelli S, et al. (2018) Shared autonomy via hindsight optimization for teleoperation and teaming. *The International Journal of Robotics Research* 37(7): 717–742.

Jebellat I and Sharf I (2023) Trajectory generation with dynamic programming for end-effector sway damping of forestry machine 2023 IEEE International Conference on Robotics and Automation (ICRA), London, UK, 29 May 2023 - 02 June 2023, 8134–8140. DOI: 10.1109/ICRA48891.2023.10161232.

Jorda M, Vulliez M and Khatib O (2022) Local autonomy-based haptic-robot interaction with dual-proxy model. *IEEE Transactions on Robotics* 38(5): 2943–2961. DOI: 10.1109/TRO.2022.3160053.

Julian R, Swanson B, Sukhatme G, et al. (2021) Never stop learning: the effectiveness of fine-tuning in robotic reinforcement learning. In: J Kober, F Ramos and C Tomlin (eds) Proceedings of the 2020 Conference on Robot Learning, Proceedings of Machine Learning Research, 2120–2136.

Kiran BR, Sobh I, Talpaert V, et al. (2021) Deep reinforcement learning for autonomous driving: a Survey. *IEEE Transactions on Intelligent Transportation Systems* 23: 4909–4926. DOI: 10.1109/TITS.2021.3054625.

Liang X, Ma Y, Feng Y, et al. (2021) Ptr-ppo: Proximal policy optimization with prioritized trajectory replay.

Löfgren B (2009) *Kinematic Control of Redundant Knuckle Booms with Automatic Path Following Functions*. PhD Thesis, Stockholm: KTH, Mechatronics. QC 20100729.

Losey DP, Jeon HJ, Li M, et al. (2022) Learning latent actions to control assistive robots. *Autonomous Robots* 46: 115–147. DOI: 10.1007/s10514-021-10005-w.

Luce RD (1977) The choice axiom after twenty years. *Journal of Mathematical Psychology* 15(3): 215–233. DOI: 10.1016/0022-2496(77)90032-3.

Luyben WL (1973) Parallel cascade control. *Industrial & Engineering Chemistry Fundamentals* 12(4): 463–467.

Pateria S, Subagdja B, Tan AH, et al. (2021) Hierarchical reinforcement learning: a comprehensive Survey. *ACM Computing Surveys* 54(5): 1–35.

Raffin A, Hill A, Gleave A, et al. (2021) Stable-baselines3: reliable reinforcement learning implementations. *Journal of Machine Learning Research* 22(268): 1–8.

Reddy S, Dragan AD and Levine S (2018) Shared autonomy via deep reinforcement learning. arXiv:1802.01744 DOI:10.48550/ARXIV.1802.01744.

Schaul T, Quan J, Antonoglou I, et al. (2016) Prioritized experience replay. arXiv:1511.05952.

Schrum ML, Johnson M, Ghuy M, et al. (2020) Four years in review: statistical practices of likert scales in human-robot interaction studies. Companion of the 2020 ACM/IEEE International Conference on Human-Robot Interaction, HRI '20, 43–52. DOI: 10.1145/3371382.3380739.

Schulman J, Wolski F, Dhariwal P, et al. (2017) Proximal policy optimization algorithms. arXiv:1707.06347.

Smith L, Kew JC, Bin Peng X, et al. (2022) Legged robots that keep on learning: fine-tuning locomotion policies in the real world 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23-27 May 2022, 1593–1599. DOI: 10.1109/ICRA46639.2022.9812166.

Song J and Sharf I (2022) Stability-constrained mobile manipulation planning on rough terrain. *Robotica* 40(11): 4090–4119. DOI: 10.1017/S0263574722000777.

Sutton RS, Precup D and Singh S (1999) Between mdps and semi-mdps: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence* 112: 181–211. DOI: 10.1016/S0004-3702(99)00052-1.

Westerberg S (2014) *Semi-Automating Forestry Machines*. Sweden: Umeå University. PhD Thesis.

Yousefi E, Losey DP and Sharf I (2022) Assisting operators of articulated machinery with optimal planning and goal inference 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23-27 May 2022, 2832–2838. DOI: 10.1109/ICRA46639.2022.9811864.

Yousefi E, Chen M and Sharf I (2023) Hierarchical planning and policy shaping shared autonomy for articulated robots.

Yu CC (1988) Design of parallel cascade control for disturbance-rejection. *AIChE Journal* 34(11): 1833–1838. DOI:DOI: 10.1002/aic.690341109.

Ziebart BD, Maas AL, Bagnell JA, et al. (2008) Maximum entropy inverse reinforcement learning. *AAAI Conference on Artificial Intelligence* 8: 1433–1438.

Ziegler DM, Stiennon N, Wu J, et al. (2020) Fine-tuning language models from human preferences. arXiv:1909.08593.

Appendix

A Extended results for subjective tests

Related to §7.3 and Figure 10, Figure 11 shows the visualization of user responses to Likert questionnaire items in Figures (a)-(e) corresponding to items 1-5 (see Table 3). In this figure, we report mean (red annotations), median (green annotations), and standard deviation (black annotations) of user responses for each experiment.

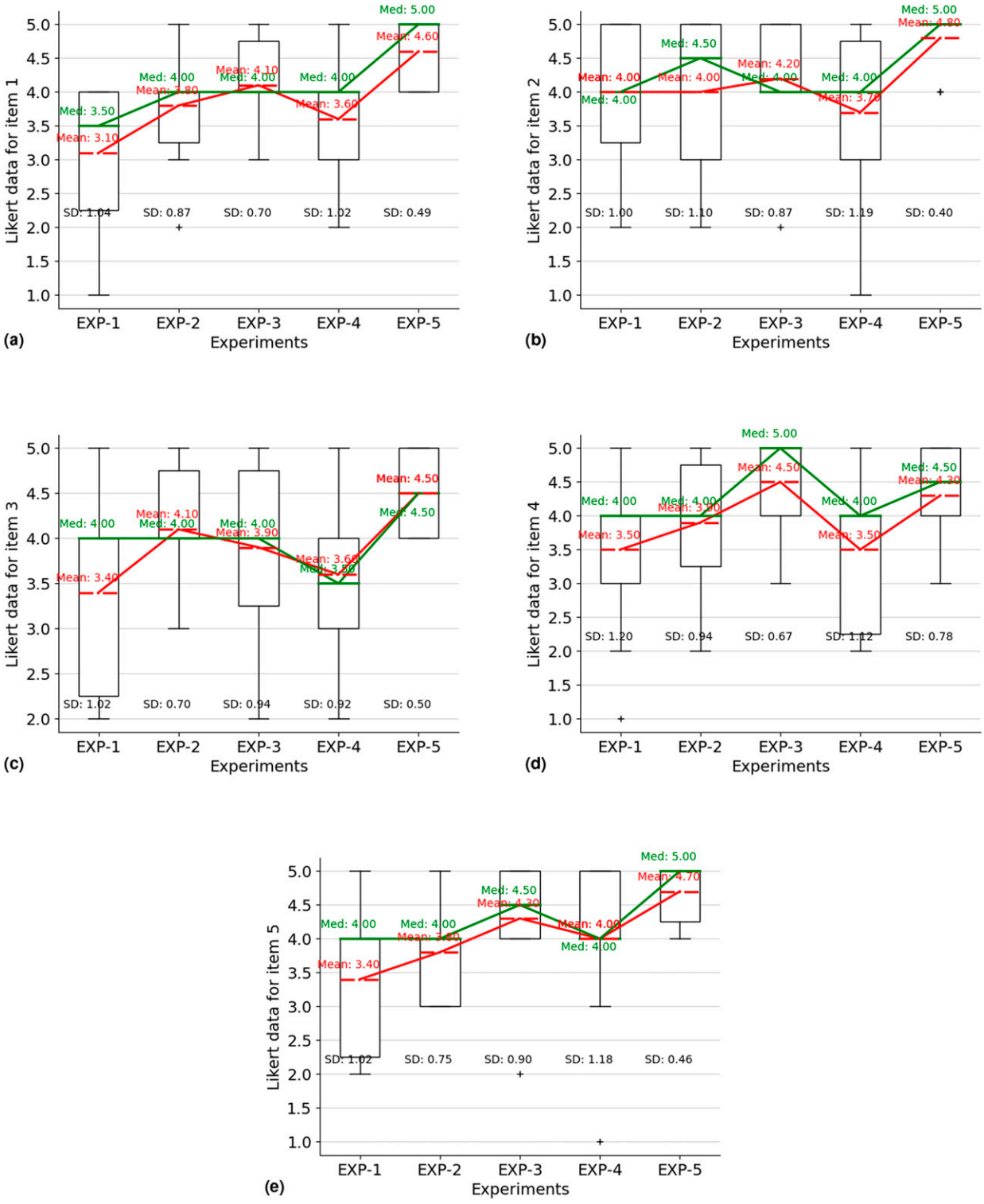

Figure 11. Visualization of user responses to Likert questionnaire items corresponding to items 1-5 (see Table 3).