Fleet Supervisor Allocation: A Submodular Maximization Approach

Anonymous Author(s)
Affiliation
Address
email

Abstract: In real world scenarios, the data collected by robots in diverse and unpredictable environments is crucial for enhancing their models and policies. This data is predominantly collected under human supervision, particularly through imitation learning (IL), where robots learn complex tasks by observing human supervisors. However, the deployment of multiple robots and supervisors to accelerate the learning process often leads to data redundancy and inefficiencies, especially as the scale of robot fleets increases. Moreover, the reliance on teleoperation for supervision introduces additional challenges due to potential network connectivity issues. To address these inefficiencies and the reliability concerns of network-dependent supervision, we introduce an adaptive submodular maximization-based policy designed for efficient human supervision allocation within multi-robot systems under uncertain connectivity. Our approach significantly reduces data redundancy by balancing the informativeness and diversity of data collection, and is capable of accommodating connectivity variances. We evaluated the effectiveness of ASA in a simulation environment with 100 robots across four different environments and various network settings, including a real-world teleoperation scenario over a 5G network. We trained and tested both our and the state-of-the-art policies utilizing NVIDIA's Isaac Gym, and our results show that ASA enhances the return on human effort by up to $5.95\times$, outperforming current baselines in all simulated scenarios and providing robustness against connectivity disruptions.

Keywords: Imitation Learning, Submodular Maximization, Fleet Learning

1 Introduction

Today, diverse industries deploy robotic fleets for tasks ranging from autonomous driving [1, 2] to healthcare [3] and package delivery [4]. These robots are often deployed with policies trained on a dataset that is primarily based on simulations, along with a small amount of data collected through real-world interactions. While effective within their training contexts, these models often fail to adapt to new or evolving real-world scenarios [5], making data collection critical for the success of the robotics applications [6, 7].

A popular approach to collecting such data is through human supervision, where humans directly guide the robots to perform the tasks. These data are then used to train the robots via Imitation Learning (IL), where the robots are trained to perform tasks by observing the human demonstrations [8]. Imitation Learning (IL) has been effective in many robotics applications, ranging from autonomous driving [9] to robotic manipulation [10, 11]. However, the breadth of scenarios necessary for effective IL emphasizes the need for continual data collection [12], commonly done with numerous robots in parallel. Usually, the number of humans is less than that of deployed robots. For instance, a recent autonomous delivery company, Starship Technologies, operates 1700 autonomous robots while teleoperating only 1% of this robotic fleet [13, 14]. The scarcity of human supervisors necessitates the selection of informative robots for supervision [15, 16].

Human supervision is often provided through real-time teleoperation over a network, especially when supervising fleets of robots distributed across the globe. For example, various companies, including Cruise, utilize human supervisors located in their control centers to teleoperate autonomous

Submitted to the 8th Conference on Robot Learning (CoRL 2024). Do not distribute.

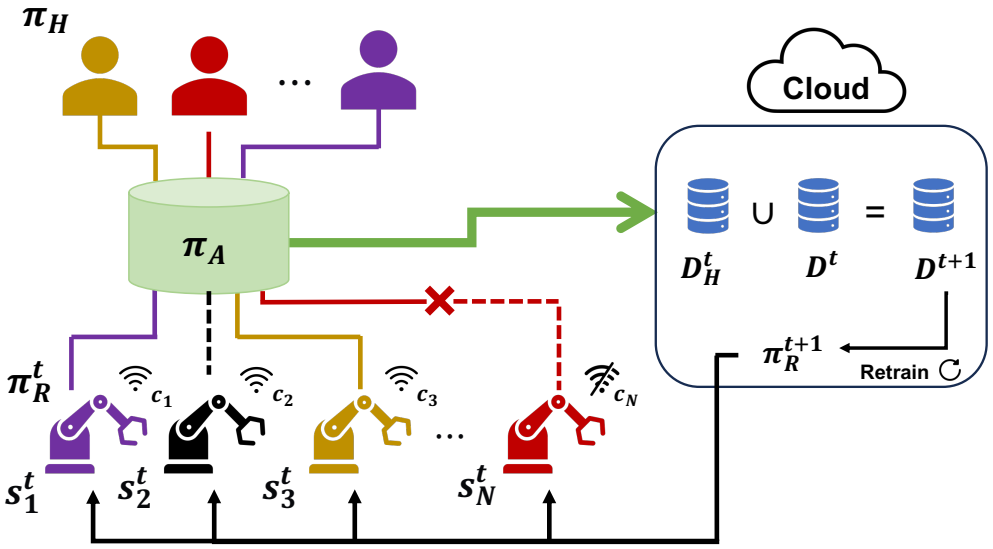

Figure 1: **Supervisor Allocation Problem:** In each time step t, the human supervisors with policy π_{H} can be allocated to the robots with policy π_{R}^t based on the allocation policy π_{A}. Each robot i has been operating in different states s_i^t, and the human supervisors are allocated to the robots based on the uncertainty of the robots and the similarity between the robots. Additionally, the supervision is provided through teleoperation with probability c_i; meaning there is a chance that the connection to robot i might fail. At the end of each time step t, the data collected by human supervisors $\mathbf{D}_H^t$ is added to the dataset $\mathbf{D}^t$ to create an updated dataset $\mathbf{D}^{t+1}$ which is then used to train the robot policy π_{R}^{t+1}.

vehicles deployed across the world [17, 18]. However, these networks might be susceptible to connection failures [19], and it is important to be robust against network uncertainties. Combining these challenges with selecting informative robots, we formulate the Supervisor Allocation Problem (Fig. 1), which involves managing limited human resources to maximize data diversity and quality under uncertain network connectivity. Our problem is an extension of the Interactive Fleet Learning (IFL) setting introduced by [15]. We extend the IFL setting to account for network elements that play an important role in real-world teleoperation scenarios [20].

We then introduce a novel human supervisor allocation policy called adaptive submodular allocation (ASA). ASA distributes human supervisory capacity across a fleet, ensuring a balance of data informativeness and diversity to minimize redundancy in data collection. Our allocation policy is shown to be robust against network instabilities and is able to adapt to the dynamic nature of data collection, which we demonstrate through extensive simulations in diverse network environments, including real-world 5G scenarios. We show that ASA improves human supervision efficacy metric Return on Human Effort (RoHE) [15] by up to $5.95\times$ compared to existing benchmarks.

2 Related Work

Data collection is a critical problem in robotics and machine learning that is essential for continually improving the performance of robots [21–25]. It is closely related to active learning [26–30], where the goal is to select the most informative samples to label. Although the goal of data collection is similar to active learning, the focus is on collecting data samples that are the most informative for training the models. In our case, however, the aim is to select the robots that provide the most informative data for human supervisors.

IL is a popular approach in robotic learning, where robots learn policies from human demonstrations [31–34]. Despite its potential, the reliance on purely offline data introduces several challenges such as distribution shifts [35], which occurs when robots encounter states that were not previously experienced by humans. These issues can be alleviated through online data collection methods, including Dataset Aggregation (DAgger) [35] and various forms of interactive IL [36, 37]. Most interactive IL methods rely on human supervision to decide on when to intervene in the robot's learning process. This presents scalability challenges, especially when applied to extensive robot networks [38] or during prolonged learning phases [39]. Robot-initiated interactive IL strategies like EnsembleDAg-

ger [40] and ThriftyDAgger [41] have been proposed to mitigate these constraints, enabling robots to request human input under specific conditions. However, these methods are designed for single-robot task allocation scenarios and do not consider multi-robot scenarios. Closest to our work, Fleet-DAgger [42] has been proposed to address the supervisor allocation problem in a multi-robot scenario. However, Fleet-DAgger does not consider operational constraints that might limit the allocation of human supervisors, such as network connectivity and the potential redundancy from employing multiple human supervisors in similar environments. Our work, on the other hand, focuses on learning an allocation policy that is adaptable to the operational constraints while minimizing the redundancy in the data collection, which is crucial for the system's scalability [27].

One popular approach to mitigate redundancy in data collection is using submodular maximization. Submodularity refers to the property of the marginal gain of adding an item to a small set being higher than adding the same item to a large set. As submodularity is a common trend in data collection, it has been widely used in machine learning tasks such as sensor placement [43], active learning [27, 30, 44], and summarization [45]. Submodular maximization has also been extended to stochastic settings [46, 47], where the goal is to select a subset of items to maximize the expected value of a submodular function. Despite its wide use in machine learning, stochastic submodular maximization has not been used in the context of IL and multi-robot data collection scenarios. Our work is the first to use stochastic submodular maximization in the context of human supervision and multi-robot scenarios to address the supervisor allocation problem.

3 Problem Formulation

Consider a geo-distributed system of N_{robot} robots, $\mathbf{I} = \{1, \cdots, N_{\text{robot}}\}$. Each robot i operates in parallel within an independent Markov Decision Process (MDP) with a different initial state. However, all robots operate within the same state and action spaces $\mathbf{S}$ and $\mathbf{A}$, respectively. Each robot i observes the state of the environment $s_i^t \in \mathbf{S}$ at time t and selects an action $a_i^t \in \mathbf{A}$ based on a policy $\pi_{\text{R}}^t : \mathbf{S} \to \mathbf{A}$. The robots share the same policy π_{R}^t that has been trained using the collective data $\mathbf{D}^t$ accumulated up to time step t. We define the collection of states and actions for all robots as $\mathbf{s}^t = (s_1^t, \ldots, s_{N_{\text{robot}}}^t) \in \mathbf{S}^{N_{\text{robot}}}$ and $\mathbf{a}^t = (a_1^t, \ldots, a_{N_{\text{robot}}}^t) \in \mathbf{A}^{N_{\text{robot}}}$. These robots can be supervised by N_{human} human supervisors with an oracle policy $\pi_{\text{H}} : \mathbf{S} \to \mathbf{A}_H$, respectively. In addition to the robot action space $\mathbf{A}$, the human action space $\mathbf{A}_H$ includes a reset action R, which can return the robot to a safe state.

Supervisor Allocation and Connectivity: In each time step t, N_{human} human supervisors can be assigned to the robots for assistance. However, the connections to the robots are unreliable, with $C = \{c_1, \cdots, c_{N_{\text{robot}}}\} \in \mathbb{R}^{N_{\text{robot}}}$ denoting independent random variables associated with the connection reliability of the robots. $c_i \in \{0, 1\}$ indicates whether a successful connection with robot i can be established ($c_i = 1$) or not ($c_i = 0$). Under this uncertain connectivity, we are interested in finding an allocation policy $\pi_{\text{A}} : \mathbb{R}^{N_{\text{robot}}} \times \mathbf{S}^{N_{\text{robot}}} \times \mathbf{A}^{N_{\text{robot}}} \times \mathbf{I} \to X$ that selects robots to be supervised $X \subseteq \mathbf{I}$ based on connection reliability C, collection of states $\mathbf{s}^t$ and action spaces $\mathbf{a}^t$.

Data Collection and Policy Retraining: Upon allocation, human supervisors contribute data only from successful connections, forming the human supervision data $\mathbf{D}_H^t$. The robot policy π_{R}^t is then updated by integrating this new data into the current dataset and retraining:

$$\mathbf{D}^{t+1} = \mathbf{D}^t \cup \mathbf{D}_H^t, \quad \mathbf{D}_H^t = \{(s_i, \pi_{\text{H}}(s_i)) : i \in X \text{ and } c_i = 1\}, \tag{1}$$

$$\pi_{\text{R}}^{t+1} = g(\pi_{\text{R}}^t, \mathbf{D}^{t+1}). \tag{2}$$

Objective: Our objective is to develop an allocation policy π_{A} that maximizes the expected Return on Human Effort (RoHE) over the connectivity C. RoHE metric was introduced along with Interactive Fleet Learning setup [15] to set a benchmark in Fleet Learning settings. It is a ratio of the total reward obtained by the fleet to the total number of human actions:

$$\max_{\pi_{\text{A}} \in \Omega} \mathbb{E}_C \left[\frac{N_{\text{human}}}{N_{\text{robot}}} \frac{\sum_{i \in \mathbf{I}} \sum_{t=0}^{T} r(s_i^t, a_i^t)}{1 + \sum_{t=0}^{T} |\pi_{\text{A}}(C, \mathbf{s}^t, \mathbf{a}^t, \mathbf{I})|_F^2} \right]. \tag{3}$$

Here T is the time horizon covering all time steps, $r : \mathbf{S} \times \mathbf{A} \to \mathbb{R}$ is the reward function, $|\cdot|_F$ denotes the Frobenius norm, and Ω refers to a set of all allocation policies. Intuitively, RoHE measures the total performance of the robotic fleet normalized by the total number of human interventions.

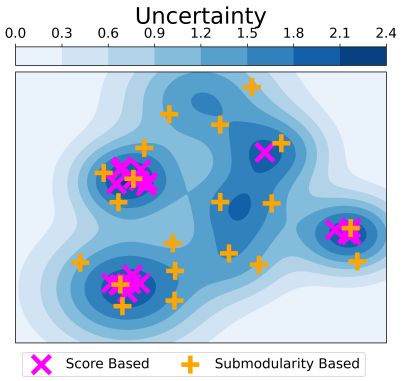

Figure 2: **Submodular Maximization Balances Uncertainty and Diversity:** This figure illustrates a toy example of our allocation problem in a 2D state space. The blue contours indicate the uncertainty levels, with darker shades representing higher uncertainty. Purple crosses (traditional score-based allocations) and yellow plus signs (our submodularity-based allocations) mark the positions of selected robots. Unlike score-based methods that often select highly uncertain but potentially overlapping states, our approach strategically picks a more diverse set of states, effectively balancing between high uncertainty and coverage, thereby reducing data redundancy and enhancing the training data's representatives.

4 A Stochastic Submodular Maximization Approach

We now present our novel policy, adaptive submodular allocation (ASA), for the problem outlined in Eq. 3 and its components. First, we define the stochastic submodular maximization problem, which represents the value of robot supervision, and then we define the greedy algorithm, which is used to pick the robots to supervise.

4.1 Submodular Maximization Problem

To address the optimization problem presented in Eq. 3, we use stochastic submodular maximization. Stochastic submodular maximization is particularly suited to our scenario because it leverages the diminishing returns property that naturally reflects the decrease in the marginal gain of supervising additional robots. This aspect is vital for assessing the efficiency of human supervision. Furthermore, the method inherently discourages the selection of similar robots, thereby avoiding the assignment of humans to robots that offer overlapping information, which decreases the return on human effort. Finally, the stochastic submodular maximization accounts for the inherent uncertainty of our problem, acknowledging the non-deterministic connections to the robots, which is crucial for developing a robust solution across different connectivity patterns.

We first define a submodular objective function $f : 2^{N_{\text{robot}}} \times \{0,1\}^{N_{\text{robot}}} \rightarrow \mathbb{R}$ that quantifies the value of supervising a selected set of robots X, considering the allocation reliability outcomes for these robots C. We define our objective function based on the facility location problem, a classical example of a submodular maximization objective [48], as follows:

$$f(X, C) = \sum_{i \in \mathbf{I}} \max_{j \in X} c_j M_{j,i}. \tag{4}$$

Here X is the set of robots selected for human supervision, and C indicates whether the connection to the robot is successful or not. $M_{j,i}$ represents the value of supervision of the robot j on the robot i, and we consider two factors: the informativeness of the robot i and the similarity between the robot j and the robot i. Additionally, our formulation is modular and can be extended to include other factors, such as prioritizing the robots that have violated the safety constraints or the robots that are in critical states. With all factors combined, we define the value of supervision $M_{j,i}$ as:

$$M_{j,i} = \mathcal{S}(j, i) * \mathcal{U}(i) + \mathcal{C}(i). \tag{5}$$

Here, $\mathcal{S}(j, i)$ defines the similarity between the robots i and j, and $\mathcal{U}(i)$ is the informativeness of the robot i, while $\mathcal{C}(i)$ is an indicator of whether the robot i violates the safety constraints or is in a critical state. Our definition of $M_{j,i}$ is modular, and each factor can be defined based on specific requirements. For example, the similarity function $\mathcal{S}$ can be defined as the cosine similarity, the Euclidean distance, or any other similarity metric. The informativeness of the robot $\mathcal{U}(i)$ can be defined as the entropy of the robot's policy or the uncertainty of the robot's state, while the constraint function $\mathcal{C}(i)$ can be defined based on the safety constraints or the critical states for the robots. With the objective function defined, we pose the following maximization problem to optimize our allocation policy:

$$\max_{X \subseteq \mathbf{I}} \quad \mathbb{E}_C[f(X, C)] \tag{6}$$
$$\text{subject to: } |X| \leq N_{\text{human}},$$

where the goal is to identify the subset of robots X that maximizes the expected value of f, constrained by the number of available human supervisors N_{human}.

4.2 Adaptive Supervisor Allocation (ASA) Policy

Now, we can present our allocation policy ASA based on a greedy algorithm given in Algorithm 1. Starting from an empty solution set X (line 1), ASA iteratively selects the robot with the highest marginal gain in expectation over probabilities of connection to the robots C (line 3). Then, ASA computes the expected marginal gain of selecting the robot x^* (line 4), and if the expected marginal gain is below a certain threshold, the algorithm stops the selection process (line 5). This threshold ensures that the algorithm avoids using unnecessary human effort by stopping when the marginal gain of selecting an additional robot is low. Otherwise, the chosen robot x^* is added to the solution set X (line 8). Finally, based on the availability of the observation on whether the connection to the robot was successful or not, the connection probabilities are updated (line 9).

Algorithm 1 ASA Policy

Input: connectivities of robots C, set of all robots $\mathbf{I}$
Output: robots selected for supervision X

1: Initialize $X \leftarrow \emptyset$
2: **for** $k = 1$ to N_{human} **do**
3: $\quad x^* \leftarrow \text{argmax}_{x \in \mathbf{I} \setminus X} \mathbb{E}_C[f(X, C)]$
4: $\quad$ Compute $\Delta \leftarrow \mathbb{E}_C[f(X \cup \{x^*\}, C) - f(X, C)]$
5: $\quad$ **if** $\Delta <$ threshold **then**
6: $\quad\quad$ **break**
7: $\quad$ **end if**
8: $\quad X \leftarrow X \cup x^*$
9: $\quad$ If possible, observe whether connection to robot x^* is successful and update connectivities C
10: **end for**

Based on the availability of the observations of the connection probabilities, we define two variants of our policy: non-adaptive submodular allocation (n-ASA) and adaptive submodular allocation (ASA). In n-ASA, we are not able to observe the connection probabilities, and thus, the allocations are done beforehand. In ASA, on the other hand, the robots are selected iteratively; based on the success of the allocations, the connection probabilities are updated. To visualize the differences between n-ASA and ASA, consider the following: in both cases, robot 1 is selected for supervision in the first iteration. While selecting the robot to supervise, n-ASA considers both possibilities (successful and unsuccessful connection to robot 1) and selects the second robot that maximizes the expected marginal gain over both cases. However, in ASA, after selecting robot 1 for supervision, we observe whether the connection was successful or not and select the next robot that maximizes the expected marginal gain based on the observation.

When the marginal threshold parameter is set to zero, ASA is equivalent to the greedy algorithm for submodular maximization, which is proven to approximate the optimal solution for the submodular maximization problem in Eq. 6 with a factor of $1 - 1/e$ [46]. Additionally, n-ASA approximates the optimal adaptive policy with a factor of $(1 - 1/e)^2$ [46]. To compute selected robots faster, we use the lazy greedy algorithm [47]; this has the same time complexity as Algorithm 1, but has a better empirical performance.

5 Experiments

We consider a fleet of $N_{\text{robot}} = 100$ robots that can be supervised by $N_{\text{human}} = 5$ human supervisors. The human supervisors are implemented as reinforcement learning agents using the Proximal Policy Optimization (PPO) algorithm [49]. We utilize the behavior cloning algorithm to initialize the robot policies based on an offline dataset of 5000 state-action pairs and use our allocation policy to collect data from the robots and update the models. In all of our experiments, when the robots violate the constraints, we perform a hard reset to bring the robots back to a safe state. We set the hard reset time $t_R = 5$ timesteps, the minimum intervention time $t_T = 5$ timesteps, and the fleet operation time $T = 10,000$ timesteps. We average the results over 3 random seeds for each task and network configuration. We have chosen these parameters to align with the settings used in the environments of the benchmark algorithms [42] for a fair comparison.

Environments: We consider four different environments in our experiments: (1) Humanoid, where the robots focus on bipedal locomotion; (2) ANYmal, where the robots focus on quadruped locomo-

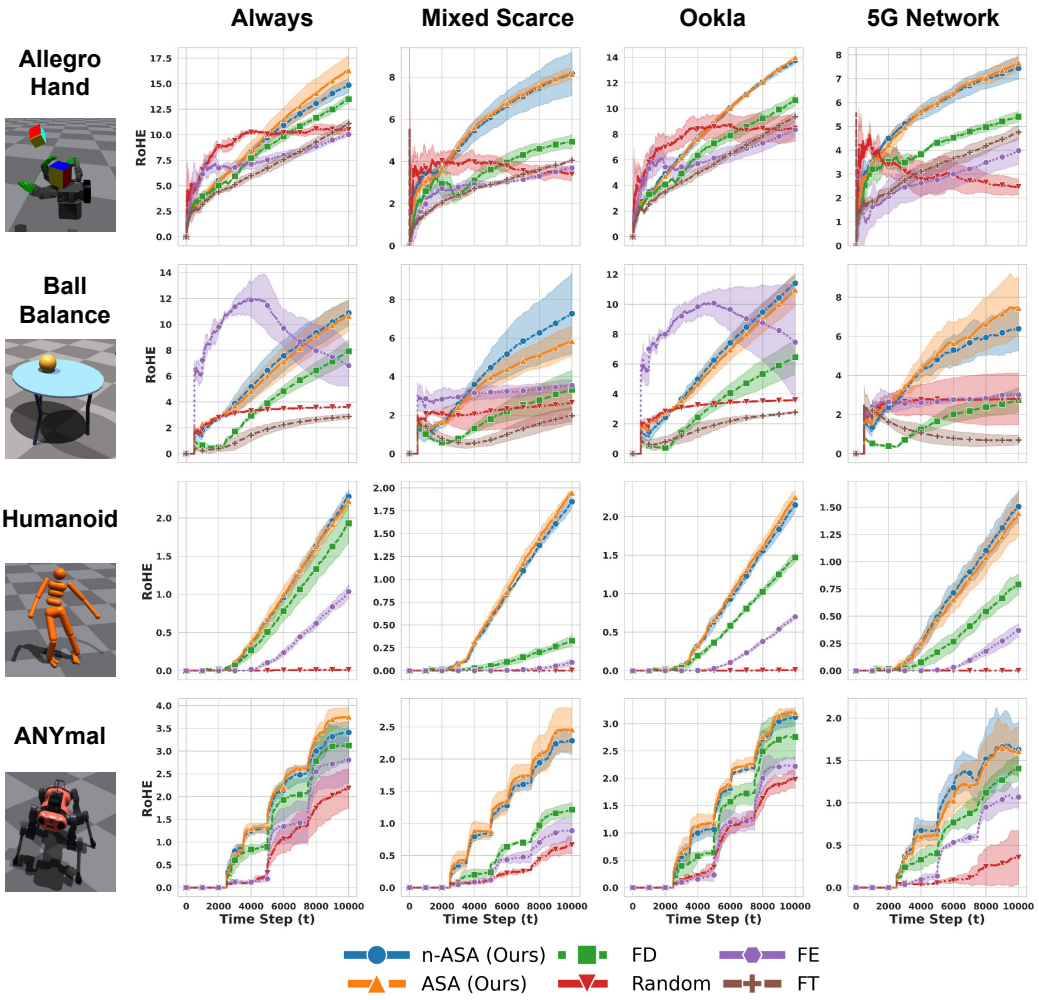

Figure 3: **Our ASA and n-ASA policies outperform other benchmarks across all environments and network combinations.** Here, each row represents a different environment, and each column corresponds to a different network configuration. ASA and n-ASA performance is affected least by changes in the network configurations because of their stochastic submodular maximization-based policies that can incorporate network uncertainties. Additionally, the submodular maximization objective improves the performance when there are no network uncertainties (column 1) due to its ability to cover diverse and informative scenarios.

tion with the ANYmal robot; (3) Allegro Hand, where the robots focus on dexterous manipulation tasks; and (4) Ball Balance where the robots focus on balancing a ball on a plate. Each environment defines constraint violations specifically similar to [50] requiring human intervention to reset the robots to a safe state.

Network Configurations: We use 4 different network configurations based on various connectivity probabilities: (1) **Always**, where the robots can always be supervised by the human supervisors; (2) **Mixed-Scarce**, where some of the robots have a high probability of connection while others have a low; (3) **Ookla**, where the robots have a varying probability of connection based on cellular network performance metrics [51] (4) **5G**, which is connectivity data that we collected over a real-world 5G network in a university robotics lab.

Metrics: We evaluate the performance of the allocation policies based on the following metrics: (1) Return on Human Effort (RoHE), which was given in Eq. 3 and (2) the cumulative number of successfully completed tasks by the entire fleet, which we will refer to as cumulative success. To simplify, RoHE measures the fleet performance per human intervention, while cumulative success only considers the total successful task completion without considering the number of interventions. For example, simply allocating all human supervisors would improve cumulative success but decrease RoHE. An ideal allocation policy should balance the two, as an ideal system would require a high total success while using humans as efficiently as possible.

Baselines: We compare the following baselines: (1) **Random**, which randomly selects the robots to be supervised by the human supervisors at each time step; (2) **Fleet-EnsembleDAgger (FE)**, which utilizes variance for uncertainty estimation, combining it with constraint-based prioritization [40]; (3) **Fleet-ThriftyDAgger (FT)**, which merges uncertainty and goal-oriented prioritization, adapting ThriftyDAgger for fleet setting [41], for environments with a defined goal; (4) **Fleet-DAgger (FD)**, which prioritizes the robots violating constraints and selects the robots with the highest uncertainty and risk of failure for fleet supervision [42] (5) **Non-Adaptive Submodular Allocation (n-ASA)** and (6) **Adaptive Submodular Allocation (ASA)**, which are the two variants of our proposed method based on submodular maximization in the absence and presence of the observation of the connection to the robots described in Section 4; please see the Appendix for the exact similarity, uncertainty and constraint functions we have used in the submodular maximization objective given in Eq. 4.

How do ASA and n-ASA perform under different network configurations?

We evaluate the performance of our policies, ASA and n-ASA, under different network configurations for each environment. The RoHE metric for each time step has been shown in Fig. 3, and cumulative success values at the final time step have been presented in Fig. 4. In both metrics, we can see that our ASA and n-ASA allocation policies are able to outperform other benchmarks. This is because our policies can incorporate network uncertainty information into their allocation policy through stochastic submodularity. On the other hand, other benchmark policies are not designed to incorporate such network uncertainty. We can see that our allocation policy outperforms other baselines in terms of the RoHE metric by up to $5.95\times$, $2.03\times$, $1.65\times$, and $2.47\times$ in Humanoid, ANYmal, Allegro Hand, and Ball Balance environments, respectively.

How do ASA and n-ASA compare when the network connectivity is stable?

To test whether our RoHE gains are only due to adaptability to different network configurations, we have also simulated a network where all robots are always reachable. In column 1 of Fig. 3, we can see that our ASA and n-ASA policies still outperform other allocation benchmarks thanks to their ability to diversify the selected robots to cover more states. As we have shown in the 2D toy example in Fig. 2, rather than only focusing on the states with high uncertainty, ASA and n-ASA consider the whole state space to collect combined data that is more informative.

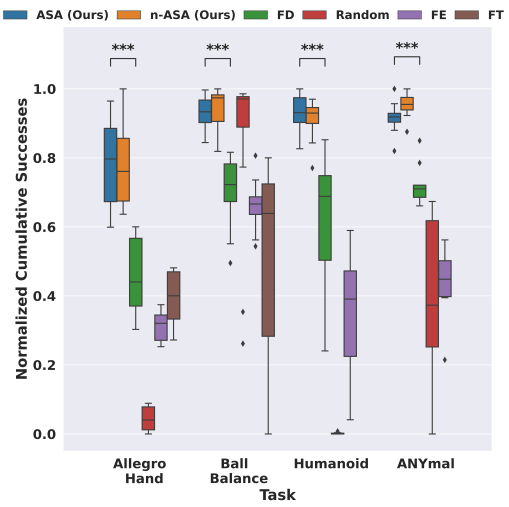

Figure 4: **ASA and n-ASA outperform all benchmarks in cumulative successes metric.** We present a box plot of the normalized cumulative success metric at the final time step in each environment and allocation policy. ASA and n-ASA can achieve higher cumulative success values with less standard deviation thanks to their robustness against network uncertainties and superior robot selection. Interestingly, Random policy achieves comparable results in the Ball Balance environment simply by allocating more humans to the process, sacrificing the return on human metric 3. On the other hand, ASA and n-ASA achieve higher performance by not allocating more humans but through a selection of better representative robots.

How does the availability of observation affect our allocation policies?

Although we know that ASA approximates the optimal solution with a stricter bound than n-ASA, in practice, these allocation policies perform very similarly. Both policies are superior to other benchmarks in all configurations and have similar RoHE and cumulative success metrics. This performance similarity between ASA and n-ASA further proves that our performance gains are mainly a result of our stochastic submodular maximization approach rather than the observation of whether the connections with the previous robot are successful or not. This flexibility enables our policies to be applied in various real-world scenarios where the observations might be impossible.

Can ASA and n-ASA improve cumulative success and RoHE metrics at the same time?

Allocation Policy	AllegroHand		AnyMAL		BallBalance		Humanoid	
	RoHE	Cumulative Success	RoHE	Cumulative Success	RoHE	Cumulative Success	RoHE	Cumulative Success
Random	2.47	228.67	0.36	47.33	2.81	1401.33	0	0
FT	4.75	2131.66	-	-	0.69	344.66	-	-
FE	3.98	1789.67	1.07	137	3.01	1315	0.37	169.33
FD	5.4	2213	1.40	200	2.75	1209	0.79	319.33
n-ASSA (Ours)	**7.66**	**3705.33**	1.61	**246.33**	**7.45**	**1758.33**	1.44	444.33
ASSA (Ours)	7.43	3658	**1.63**	239.33	6.39	1703.33	**1.51**	**470**

Table 1: Our proposed n-ASA and ASA policies outperform other baselines in all environments (columns) in real-world 5G network data in terms of cumulative success and return on human effort (RoHE). The results are consistent across all tasks, showing the adaptability of our method to different environments. Additionally, ASA and n-ASA outperform other baselines in both cumulative success and RoHE, meaning our allocation policy is both efficient in human effort and achieves higher cumulative success.

We can clearly see in Fig. 3 and Fig. 4 that our ASA and n-ASA policies achieve both the highest RoHE and cumulative success in all network configurations. Our policies are able to balance these two metrics thanks to their threshold criteria, preventing the allocation of humans to uninformative robots. For example, we can see that in the Ball Balance environment, the Random allocation policy is able to achieve comparable cumulative success in Fig. 4, but it fails to reach comparable RoHE values (see row 2 in Fig. 3). This suggests that Random policy achieved high cumulative success values by simply allocating more humans but failed to optimize human efficiency.

5.1 Physical 5G Network Connectivity Data

In addition to the simulated network connectivity data, we also evaluate our allocation policies on real-world 5G network connectivity data collected in the field. To create such a dataset, we utilize a local 5G network dedicated to testing the real-time teleoperation of the robots over a period of 24 hours. Then, we divide the geographic area into 100 different regions with the same number of users (robots) in each region and calculate the average latency and throughput of the network for each region. We use this data to create network connectivity where the robots with higher latency and lower throughput have a lower probability of establishing a successful connection with the human supervisors. Please refer to the Appendix for further details on 5G network data collection and the exact setup we used.

Results: We present our results on the real-world 5G network data in Table 1. The results show that our proposed method outperforms other baselines in terms of the RoHE and cumulative success metrics by up to $2.47\times$ and $1.67\times$, respectively under the 5G network configuration.

Limitations: Our work has several limitations. First, it uses only real-world data collected from 5G field trials without hardware robotics experiments. Additionally, it assumes that network connectivity and robot states and policies are independent across robots, while in real-world scenarios, robots might share the same network or physical location, meaning their policies might affect each other.

6 Conclusion and Future Work

We present novel supervisor allocation policies, ASA and n-ASA, for assigning human supervisors to the robotic fleet for data collection. ASA and n-ASA are based on stochastic submodular maximization, providing a modular approach to incorporate different allocation objectives, informativeness metrics, as well as re-training methods. Our allocation policies beat current benchmarks in terms of performance metrics such as RoHE and cumulative success in all environments and network configurations. These performance gains are thanks to its stochastic submodular maximization objective, which incorporates network connectivity in the allocation process while balancing the diversity and informativeness of selected robots. Finally, we collect real-world 5G network data from a field dedicated to teleoperated robots and show the applicability of our allocation policy in real-world scenarios as well.

In a future project, we plan to extend our work to include hardware robotics experiments, including teleoperation over a 5G network possible in an application such as autonomous driving. We also plan to investigate the impact of different imitation learning methods to test the generalizability of our allocation policies.

References

[1] Kara Carlson. You now can ride in a driverless car in austin, as gm-owned cruise expands rideshare services, Dec 2022. URL https://www.statesman.com/story/business/technology/2022/12/21/cruise-car-company-launches-austin-driverless-rideshare-service/69743913007/.

[2] Anthony James. Waymo expands us public operations with deployments in los angeles and austin, Mar 2024. URL https://www.autonomousvehicleinternational.com/news/robotaxis/waymo-expands-us-operations-with-deployments-in-los-angeles-and-austin.html.

[3] Robert Valner, Houman Masnavi, Igor Rybalskii, Rauno Põlluäär, Erik Kõiv, Alvo Aabloo, Karl Kruusamäe, and Arun Singh. Scalable and heterogenous mobile robot fleet-based task automation in crowded hospital environments—a field test. Frontiers in Robotics and AI, 9, 08 2022. doi:10.3389/frobt.2022.922835.

[4] Joseph Quinlivan. How amazon deploys collaborative robots in its operations to benefit employees and customers, Jun 2023. URL https://www.aboutamazon.com/news/operations/how-amazon-deploys-robots-in-its-operations-facilities.

[5] Cosmin Paduraru, Daniel Jaymin Mankowitz, Gabriel Dulac-Arnold, Jerry Li, Nir Levine, Sven Gowal, and Todd Hester. Challenges of real-world reinforcement learning: definitions, benchmarks and analysis. Machine Learning, 110:2419 – 2468, 2021. URL https://api.semanticscholar.org/CorpusID:234868359.

[6] Sergey Levine, Aviral Kumar, G. Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. ArXiv, abs/2005.01643, 2020. URL https://api.semanticscholar.org/CorpusID:218486979.

[7] Eric Jang, Alex Irpan, Mohi Khansari, Daniel Kappler, Frederik Ebert, Corey Lynch, Sergey Levine, and Chelsea Finn. Bc-z: Zero-shot task generalization with robotic imitation learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 991–1002. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/jang22a.html.

[8] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. Foundations and Trends® in Robotics, 7(1-2):1–179, 2018. ISSN 1935-8253. doi:10.1561/2300000053. URL http://dx.doi.org/10.1561/2300000053.

[9] Eli Bronstein, Mark Palatucci, Dominik Notz, Brandyn Allen White, Alex Kuefler, Yiren Lu, Supratik Paul, Payam Nikdel, Paul Mougin, Hongge Chen, Justin Fu, Austin Abrams, Punit Shah, Evan Racah, Benjamin Frenkel, Shimon Whiteson, and Drago Anguelov. Hierarchical model-based imitation learning for planning in autonomous driving. 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 8652–8659, 2022. URL https://api.semanticscholar.org/CorpusID:252968066.

[10] Bin Fang, Shi-Dong Jia, Di Guo, Muhua Xu, Shuhuan Wen, and Fuchun Sun. Survey of imitation learning for robotic manipulation. International Journal of Intelligent Robotics and Applications, 3:362 – 369, 2019. URL https://api.semanticscholar.org/CorpusID:202733441.

[11] Zipeng Fu, Tony Zhao, and Chelsea Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. ArXiv, abs/2401.02117, 2024. URL https://api.semanticscholar.org/CorpusID:266755740.

[12] Carlos Celemin, Rodrigo Pérez-Dattari, Eugenio Chisari, Giovanni Franzese, Leandro de Souza Rosa, Ravi Prakash, Zlatan Ajanović, Marta Ferraz, Abhinav Valada, and Jens Kober. Interactive imitation learning in robotics: A survey. Foundations and Trends® in Robotics, 10(1-2):1–197, 2022. ISSN 1935-8253. doi:10.1561/2300000072. URL http://dx.doi.org/10.1561/2300000072.

[13] Kevin Jost. Starship goes deep on college food deliveries, Dec 2022. URL https://insideautonomousvehicles.com/starship-goes-deep-on-college-food-deliveries/.

[14] Mark DiPietro. Starship technologies paving way for robotics in food delivery, Oct 2022. URL https://blogs.shu.edu/stillmanexchange/2022/10/15/starship-technologies-paving-way-for-robotics-in-food-delivery/#:~:text=Founded%20by%20two%20Skype%20co,over%2010%2C000%20orders%20per%20day.

[15] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, K Dharmarajan, Brijen Thananjeyan, P. Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In Conference on Robot Learning, 2022.

[16] Shivin Dass, Karl Pertsch, Hejia Zhang, Youngwoon Lee, Joseph J. Lim, and Stefanos Nikolaidis. Pato: Policy assisted teleoperation for scalable robot data collection, 2023.

[17] View All Posts by Mario Herger. First documented intervention by a cruise teleoperator, February 2023. URL https://thelastdriverlicenseholder.com/2023/02/23/first-documented-intervention-of-a-cruise-teleoperator/.

[18] View All Posts by Mario Herger. Remote-controlled driving as demonstrated by phantom auto, February 2024. URL https://thelastdriverlicenseholder.com/2024/02/01/remote-controlled-driving-as-demonstrated-by-phantom-car/.

[19] Abhinav Dahiya. Route Planning and Operator Allocation in Robot Fleets. PhD thesis, University of Waterloo, 2023.

[20] Adriana Noguera Cundar, Reza Fotouhi, Zachary Ochitwa, and Haron Obaid. Quantifying the effects of network latency for a teleoperated robot. Sensors, 23:8438, 10 2023. doi:10.3390/s23208438.

[21] Sandeep Chinchali, E. Pergament, M. Nakanoya, E. Cidon, E. Zhang, D. Bharadia, M. Pavone, and S. Katti. Harvestnet: Mining valuable training data from high-volume robot sensory streams. In 2020 International Symposium on Experimental Robotics (ISER), Valetta, Malta, 2020.

[22] Sandeep Chinchali, E. Pergament, M. Nakanoya, E. Cidon, E. Zhang, D. Bharadia, M. Pavone, and S. Katti. Sampling training data for continual learning between robots and the cloud. In 2020 International Symposium on Experimental Robotics (ISER), Valetta, Malta, 2020.

[23] Yuchong Geng, Dongyue Zhang, Po-han Li, Oguzhan Akcin, Ao Tang, and Sandeep P Chinchali. Decentralized sharing and valuation of fleet robotic data. In 5th Annual Conference on Robot Learning, Blue Sky Submission Track, 2021.

[24] Oguzhan Akcin, Po-han Li, Shubhankar Agarwal, and Sandeep P. Chinchali. Decentralized data collection for robotic fleet learning: A game-theoretic approach. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, Proceedings of The 6th Conference on Robot Learning, volume 205 of Proceedings of Machine Learning Research, pages 978–988. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/akcin23a.html.

[25] Crystal Chao, Maya Cakmak, and Andrea L. Thomaz. Transparent active learning for robots. In 2010 5th ACM/IEEE International Conference on Human-Robot Interaction (HRI), pages 317–324, 2010. doi:10.1109/HRI.2010.5453178.

[26] Burr Settles. Active learning literature survey. Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[27] Oguzhan Akcin, Orhan Unuvar, Onat Ure, and Sandeep P. Chinchali. Fleet active learning: A submodular maximization approach. In Jie Tan, Marc Toussaint, and Kourosh Darvish, editors, Proceedings of The 7th Conference on Robot Learning, volume 229 of Proceedings of Machine Learning Research, pages 1378–1399. PMLR, 06–09 Nov 2023. URL https://proceedings.mlr.press/v229/akcin23a.html.

[28] Yarin Gal, Riashat Islam, and Zoubin Ghahramani. Deep Bayesian active learning with image data. In Doina Precup and Yee Whye Teh, editors, Proceedings of the 34th International Conference on Machine Learning, volume 70 of Proceedings of Machine Learning Research, pages 1183–1192. PMLR, 06–11 Aug 2017. URL http://proceedings.mlr.press/v70/gal17a.html.

[29] David A. Cohn, Zoubin Ghahramani, and Michael I. Jordan. Active learning with statistical models. JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH, 4:129–145, 1996.

[30] Andreas Kirsch, Joost van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/95323660ed2124450caaac2c46b5ed90-Paper.pdf.

[31] Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 4950–4957. International Joint Conferences on Artificial Intelligence Organization, 7 2018. doi:10.24963/ijcai.2018/687. URL https://doi.org/10.24963/ijcai.2018/687.

[32] Felipe Codevilla, Eder Santana, Antonio M. López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pages 9328–9337, 2019. URL https://api.semanticscholar.org/CorpusID:125953399.

[33] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, page 4572–4580, Red Hook, NY, USA, 2016. Curran Associates Inc. ISBN 9781510838819.

[34] Brenna Argall, Sonia Chernova, Manuela Veloso, and Brett Browning. A survey of robot learning from demonstration. Robotics and Autonomous Systems, 57:469–483, 05 2009. doi:10.1016/j.robot.2008.10.024.

[35] Stephane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, volume 15 of Proceedings of Machine Learning Research, pages 627–635, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR. URL https://proceedings.mlr.press/v15/ross11a.html.

[36] Sonia Chernova and Manuela Veloso. Interactive policy learning through confidence-based autonomy. J. Artif. Int. Res., 34(1):1–25, jan 2009. ISSN 1076-9757.

[37] Snehal Jauhri, Carlos Celemin, and Jens Kober. Interactive imitation learning in state-space. In Jens Kober, Fabio Ramos, and Claire Tomlin, editors, Proceedings of the 2020 Conference on Robot Learning, volume 155 of Proceedings of Machine Learning Research, pages 682–692. PMLR, 16–18 Nov 2021. URL https://proceedings.mlr.press/v155/jauhri21a.html.

[38] Shih-Yi Chien, Yi-Ling Lin, Pei-Ju Lee, Shuguang Han, Michael Lewis, and Katia Sycara. Attention allocation for human multi-robot control: Cognitive analysis based on behavior data and hidden states. International Journal of Human-Computer Studies, 117:30–44, 2018. ISSN 1071-5819. doi:https://doi.org/10.1016/j.ijhcs.2018.03.005. URL https://www.sciencedirect.com/science/article/pii/S107158191830096X. Cognitive Assistants.

[39] Robin R. Murphy and Erika Rogers. Cooperative Assistance for Remote Robot Supervision. Presence: Teleoperators and Virtual Environments, 5(2):224–240, 08 1996. doi:10.1162/pres.1996.5.2.224. URL https://doi.org/10.1162/pres.1996.5.2.224.

[40] Kunal Menda, Katherine Rose Driggs-Campbell, and Mykel J. Kochenderfer. Ensembledagger: A bayesian approach to safe imitation learning. In 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2019, Macau, SAR, China, November 3-8, 2019, pages 5041–5048. IEEE, 2019. doi:10.1109/IROS40897.2019.8968287. URL https://doi.org/10.1109/IROS40897.2019.8968287.

[41] Ryan Hoque, Ashwin Balakrishna, Ellen Novoseller, Albert Wilcox, Daniel S. Brown, and Ken Goldberg. Thriftydagger: Budget-aware novelty and risk gating for interactive imitation learning. In Aleksandra Faust, David Hsu, and Gerhard Neumann, editors, Proceedings of the 5th Conference on Robot Learning, volume 164 of Proceedings of Machine Learning Research, pages 598–608. PMLR, 08–11 Nov 2022. URL https://proceedings.mlr.press/v164/hoque22a.html.

[42] Ryan Hoque, Lawrence Yunliang Chen, Satvik Sharma, Karthik Dharmarajan, Brijen Thananjeyan, Pieter Abbeel, and Ken Goldberg. Fleet-dagger: Interactive robot fleet learning with scalable human supervision. In Karen Liu, Dana Kulic, and Jeff Ichnowski, editors, Proceedings of The 6th Conference on Robot Learning, volume 205 of Proceedings of Machine Learning Research, pages 368–380. PMLR, 14–18 Dec 2023. URL https://proceedings.mlr.press/v205/hoque23a.html.

[43] Andreas Krause and Daniel Golovin. Submodular function maximization. In Tractability, 2014.

[44] Baharan Mirzasoleiman, Amin Karbasi, Rik Sarkar, and Andreas Krause. Distributed submodular maximization. Journal of Machine Learning Research, 17(235):1–44, 2016. URL http://jmlr.org/papers/v17/mirzasoleiman16a.html.

[45] Hui Lin and Jeff Bilmes. A class of submodular functions for document summarization. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1, HLT '11, pages 510–520, USA, 2011. Association for Computational Linguistics. ISBN 9781932432879.

[46] Arash Asadpour, Hamid Nazerzadeh, and Amin Saberi. Stochastic submodular maximization. In Christos Papadimitriou and Shuzhong Zhang, editors, Internet and Network Economics, pages 477–489, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[47] Daniel Golovin and Andreas Krause. Adaptive submodularity: Theory and applications in active learning and stochastic optimization. Journal of Artificial Intelligence Research, 42, 03 2010. doi:10.1613/jair.3278.

[48] A. M. Frieze. A cost function property for plant location problems. Math. Program., 7(1): 245–248, dec 1974. ISSN 0025-5610. doi:10.1007/BF01585521. URL https://doi.org/10.1007/BF01585521.

[49] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[50] Viktor Makoviychuk, Lukasz Wawrzyniak, Yunrong Guo, Michelle Lu, Kier Storey, Miles Macklin, David Hoeller, Nikita Rudin, Arthur Allshire, Ankur Handa, and Gavriel State. Isaac gym: High performance gpu-based physics simulation for robot learning, 2021.

[51] Ookla. Internet speed dataset. https://www.kaggle.com/datasets/dhruvildave/ookla-internet-speed-dataset, 2022. [Online; accessed 15-February-2024].