Efficient Touch Based Localization through Submodularity

Shervin Javdani, Matt Klingensmith, Drew Bagnell, Nancy Pollard, Siddhartha Srinivasa

***Abstract*— We explore the problem of selecting a sequence of information gathering actions to localize an object quickly. We present two approaches to this problem, applied to touch based localization with a robotic end effector. In the first, we greedily select actions at each step of the sequence that minimize the Shannon entropy of our current belief. In the second, we consider many possible hypotheses of the object's pose, and greedily select actions expected to disprove the most hypotheses. We show that this formulation is adaptive submodular [1], and thus derive guarantees compared to the optimal sequence of actions. This enables us to derive guarantees compared to the *optimal* sequence. We evaluate these approaches in simulation by comparing accuracy and computation time for localizing and grasping known objects.**

I. Introduction

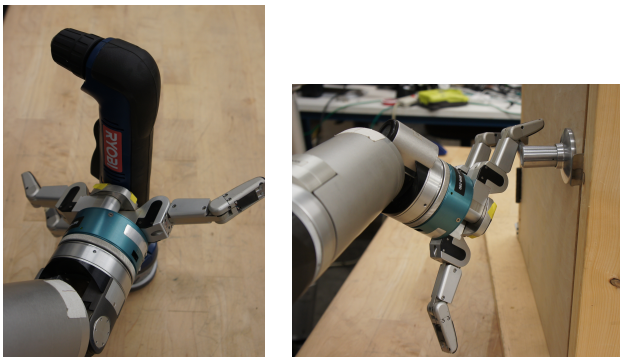
Dealing with noisy sensors, inaccurate kinematics models, and calibration error is a fundamental problem in robotics. This is particularly relevant for tasks of fine manipulation, such as precise grasping, pushing a button, or inserting a key into a keyhole. Due to the accuracy required for these tasks, small errors often lead to failure.

One approach to dealing with these inaccuracies is to perform a sequence of uncertainty reducing actions prior to attempting the task. We would like to find the optimal sequence, taking the minimal amount of time to execute while providing enough information to accomplish the task. In general, this can be formulated as a Partially Observable Markov Decision Process (POMDP). However, finding optimal solutions to POMDPs has been shown to be PSPACE complete [2]. Instead, approximate methods have been developed for offline policy computation, such as finding low-dimensional projections of the belief space [3], or point-based methods which only sample a small set of beliefs [4].

While these can achieve good performance, they may still be intractable for large problems where there are too many states to enumerate. For these problems, online planning may be used within the POMDP framework, looking at the locally reachable states during each decision step [5]. Even these can be computationally intractable, in particular if the problem has a large branching factor or search horizon.

To circumvent these issues, some approaches limit the search to a low horizon [6], often using the greedy strategy of selecting actions with the highest expected benefit in one step [7]. This is generally necessary to solve these problems, since the branching factor is very large. Though these algorithms can perform sufficiently well, there are no guarantees.

*The authors are with The Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA - 15213, USA. Contact Email: sjavdani@cmu.edu



(a) Triggering a Drill (b) Opening a Door

Fig. 1. Examples of tasks from the DARPA ARM-S Project requiring precise pose estimation.

One class of problems which performs well with a simple greedy algorithm is *submodular maximization*. A metric is *submodular* if it exhibits the diminishing returns property, which we define rigorously in Section III-A. A striking feature of submodular maximizations is that a simple greedy selection scheme is provably near-optimal compared to the optimal sequence. Furthermore, no polynomial time algorithm can guarantee optimality (unless $P = NP$) [8], [9]. If we can structure our problem in this way, we know a greedy algorithm will perform well.

One natural metric for uncertainty reduction is the expected decrease in Shannon entropy. This is referred to as the information gain metric, and has been shown to be submodular in the *non-adaptive* setting [10]. That is, if we were to select a sequence of actions offline, and perform the same sequence regardless of which observations we received online, greedy action selection would be near-optimal.

We could consider performing this selection offline - however, it has been shown that this can perform exponentially worse than an adaptive algorithm [11]. On the other hand, no optimality bounds have been shown for information gain in the adaptive setting. Thus, we cannot provide guarantees for this algorithm in the adaptive setting, though we may hope for good performance due to the connection with submodularity.

Recent notions of *adaptive submodularity* [1] extend the guarantees of submodularity to the adaptive setting, allowing us to think of reactive policies, rather than a offline plans. The set of requirements for adaptive submodular functions are different, and information gain does not meet that criteria. Instead, we design a similar metric that does. In addition to providing guarantees with respect to that metric, we can use

a lazy-greedy algorithm [1], [12] which does not reevaluate every action at every step, giving us a further computational speedup.

In this work, we draw a connection between touch based localization with a robotic end effector and submodularity. Understanding this connection enables us to restructure the problem to fit into this framework, enabling the use of an efficient algorithm that provides near-optimal solutions. Section III discusses how we structure our problem, some consequences, and what type of problems we believe work well under this scenario.

We present two approaches for uncertainty reducing action selection. The first approach optimizes the information gain by fitting a Gaussian to the remaining particles, and evaluating the expected entropy that will result from each action. The second approach maximizes the expected number of hypotheses it will disprove. We show that our formulation of this metric is adaptive submodular. We apply both algorithms to selecting touch based sensing actions prior to performing an object grasp. We present results in Section V both in accuracy and computation time of each.

II. Related work

Hsiao et al. [6] formulate the problem of producing uncertainty reducing tactile actions with a POMDP. Since it is intractable to solve fully, they perform a forward search in the same space online. Potential actions are specified by a small set (typically 5) of world-relative trajectories [13], making for a low branching factor. By searching at a limited horizon and performing aggressive pruning and clustering of observations, they circumvent the computational issues. However, this removes guarantees of optimality, and can still take many seconds to compute each action and update.

Others forgo the ability to plan with the entire belief space altogether, projecting onto a low-dimensional space before generating a plan to the goal. During execution, it is likely that this plan will fail, since the true state was not known. Erez and Smart use local controllers to adjust the trajectory [14]. Platt et al. note when the belief space diverges from what the plan expected, and re-plan from the new belief. They prove their approach will eventually converge to the true hypothesis. While these methods can plan significantly faster due to their low-dimensional projection, they may pick actions suboptimally. Furthermore, by ignoring part of the belief space, they sacrifice the ability to avoid potential failures. For example, you cannot guarantee that your trajectories won't knock an object over, since you are ignoring part of the belief space during planning.

Petrovskaya et al. [15] consider the problem of full 6DOF pose estimation of objects through tactile feedback. Their primary contribution is an algorithm capable of running in the full 6DOF space quickly. In their experiments, action selection was done randomly, as they do not attempt to select optimal actions. To achieve an error of $\sim$5*mm*, they needed an average of 29 actions for objects with complicated meshes. While this does show that even random actions converge eventually, we note that this is significantly more actions than used by other approaches [6].

In the DARPA Autonomous Robotic Manipulation Software (ARM-S) competition, teams were required to localize, grasp, and manipulate various objects within a time limit. Many teams first took uncertainty reducing actions before attempting to accomplish many of these tasks [16]. Similar strategies were used to enable a robot to prepare a meal with a microwave [17], where touch-based actions are used prior to pushing buttons. To accomplish these tasks quickly, some of these works rely on predefined motions and policies, specified for a particular object and environment. While this does enable very fast localization with high accuracy, a sequence must be made manually for each task and environment. Furthermore, these sequences aren't entirely adaptive, in that they don't deal well with every situation.

Hebert and collaborators also approached the problem of action selection for touch based localization [18], noting the use of these actions in the ARM-S competition. They utilize a greedy information gain metric, similar to one we develop. However, they do not make a connection to submodularity, and provide no guarantees with their approach.

Dogar and Srinivasa use the natural interaction of an end effector and an object to handle uncertainty with a push-grasp. By utilizing offline simulation, they reduce the online problem to enclosing the object's uncertainty in a pre-computed capture region, where a push funnels the object into the end effector. Online, they plan a grasp which encloses the uncertainty inside the capture region. This work is complimentary to ours - the push-grasp works well on objects which slide easily, while we assume objects do not move. We believe these are applicable in different scenarios.

Our contribution is drawing a connection between submodularity and end-effector localization. By understanding this link, we can formulate the problem in such a way that a greedy algorithm is guaranteed to perform near-optimally. In addition, we can achieve a speedup compared to a greedy algorithm by utilizing lazy evaluation [1], [12]. We show good performance with all of our schemes as compared to randomly selecting actions

III. Problem Formulation

This section covers the basic formulation for adaptive submodular maximization. For a more detailed explanation, see [1], [19].

Let $\mathbb{A}$ be the set of all actions available to us (end-effector trajectories), and O the set of all possible observations (the distance along the trajectory where the hand will make contact). We represent the state (object pose) with ϕ, called the *realization*. Let Φ be a random variable over all realizations. Thus, the probability of a certain pose is given by $p(\phi) = \mathbb{P}[\Phi = \phi]$.

At each iteration, we select an action $a \in \mathbb{A}$, with cost $c(a)$, and receive an observation $o \in O$. Let $A \subseteq \mathbb{A}$ be all the actions selected so far. During execution, we maintain a *partial realization* ψ_A, a sequence of observations received

indexed by A. This essentially encodes the "belief state" used in POMDPs, which we denote by $p(\phi|\psi_A) = \mathbb{P}[\Phi = \phi|\psi_A]$.

Our goal is to find an adaptive policy for selecting tests, based on the outcome of previous tests performed. Formally, a policy π is a mapping from a partial realization ψ_A to an action item a. Let $A(\pi,\phi)$ be the set of actions selected by policy π if the true state is ϕ. We define two cost functions for a policy - the expected cost and the worst case cost. These are:

$$c_{avg} = \mathbb{E}_{\Phi}[c(A(\pi,\Phi))]$$
$$c_{wc} = \max_{\phi} c(A(\pi,\phi))$$

Define some utility function (amount of uncertainty reduced) $f : 2^{\mathbb{A}} \times O^{\mathbb{A}} \to \mathbb{R}_{\geq 0}$. We would like to find a policy which, for all realizations, will reach some utility threshhold Q while minimizing one of our cost functions. Formally:

$$\min c_{\{avg,wc\}}(A(\pi,\Phi))$$
$$s.t. f(A(\pi,\phi),\phi) \geq Q, \forall \phi$$

This is often referred to as the *Minimum Cost Cover* problem, where we would like to achieve some coverage Q while minimizing the cost to do so. We can consider optimal policies π^*_{avg} and π^*_{wc} for the above, which optimize their respective cost functions. Unfortunately, obtaining even approximate solutions for this is difficult [1], [8]. However, we find near-optimal performance with a simple greedy algorithm if our objective function f satisfies properties of adaptive submodularity and monotonicty. We now briefly review these properties.

A. Submodularity

First, let us consider the case where $p(\phi)$ is deterministic. We call a function f submodular if whenever $X \subseteq Y \subseteq \mathbb{A}$, $a \in \mathbb{A} \backslash Y$:

Submodularity (diminishing returns):

$$f(X \cup \{a\}) - f(X) \geq f(Y \cup \{a\}) - f(Y)$$

That is, the marginal benefit of adding a to a smaller set X is at least as much as adding it to the superset Y. We also require monotonicty, or that adding more elements never hurts:

Monotonicity (more never hurts):

$$f(X \cup \{a\}) \geq 0$$

In this case, the greedy algorithm maximize $\frac{f(A \cup \{a\}) - f(A)}{c(a)}$ at each step. Since we do not gain observations, this is essentially a plan generated offline. This has been shown to have a $(1 + \log \max_a f(a))$ approximation for integer values f [9].

B. Adaptive Submodularity

Now we consider the case where $p(\phi)$ is non-deterministic, and we gain information as we make observations [1]. In this case, we will consider the expected marginal benefit of performing an action:

$$\Delta(a|\psi_A) = \mathbb{E}[f(A \cup \{a\}, \Phi) - f(A, \Phi)|\, \psi_A]$$

We call a function f adaptive submodular if whenever $X \subseteq Y \subseteq \mathbb{A}$, $a \in \mathbb{A} \backslash Y$:

Adaptive Submodularity:

$$\Delta(a|X) \geq \Delta(a|Y)$$

That is, the expected benefit of adding a to a smaller set X is at least as much as adding it to the superset Y, for any set of observations received for actions $Y \backslash X$. We also require strong adaptive monotonicity, or more items never hurts. For any $a \notin Y$, and any possible outcome o, this requires:

Strong Adaptive Monotonicity:

$$\mathbb{E}[f(X,\Phi)|\psi_X] \leq \mathbb{E}[f(X \cup \{a\}, \Phi)|\psi_X, \psi_a = o]$$

In this case, the greedy algorithm maximize $\frac{\Delta(a|\psi_A)}{c(a)}$ at each step. This encodes an online policy, since at each ψ_A incorporates the new observations. Interestingly, we can bound the performance of the same algorithm with respect to both the optimal average case policy π^*_{avg} and worst case policy π^*_{wc}. This has been shown to have a $(1 + \ln(Q))$ approximation for π^*_{avg}, and a $(1 + \ln(\frac{Q}{\min_\phi p(\phi)}))$ approximation for π^*_{wc} approximation for integer valued f, for self-certifying instances (see [1] for a more detailed explanation).

C. Submodularity Assumptions for Touch Localization

In order to create objectives which fit into this framework, we must make certain assumptions. First, all actions must be available for evaluation at every step. Thus, we generate a large set of information gathering trajectories at the start, and evaluate these motions at each step. Second, we cannot alter the underlying realization ϕ, so actions are not allowed to change the state of the environment or move objects.

When applied to object localization for grasping, these assumptions lend this framework towards heave objects which will remain stationary when touched. For future work, we hope to explore the use of near-touch sensors as well [20], [21].

IV. Application to Touch Localization

We would like to appeal to the above algorithms and guarantees for touch localization, while still maintaining generality for different objects and motions. Given an object mesh, we model the random realization Φ with a particle filter. We can think of each particle $\phi \in \Phi$ representing some hypothesis of the true object pose. We take the Bayesian approach and assume there is a known prior $p(\phi)$.

Each action $a \in \mathbb{A}$ corresponds to a trajectory of the end-effector. The cost of the action $c(a)$ is the total time it would take to run this trajectory. An observation $o \in \mathbb{R}$ corresponds

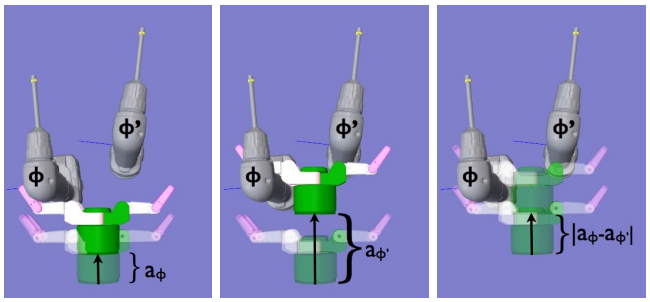


Fig. 2. The observations for action a and realizations ϕ and ϕ'. Each observation a_ϕ and a'_ϕ corresponds to the time along the straight line trajectory when contact first occurs with the object. We use the difference of times $|a_\phi - a_{\phi'}|$ when measuring how far apart observations are.

to the time from the start when the end-effector first makes contact with the object. We define a_ϕ as the time during trajectory a where contact would occur if the true state were ϕ. See Figure 2 for an example. If the swept path of a does not ever hit the object ϕ, then $a_\phi = \infty$. Note that all the methods presented below use this as one of the possible observations, and thus handle the case of no touch.

Under this formulation, we now present different utility functions f for our original objective, which capture the idea of reducing the uncertainty in Φ. In general, our original objective will look like "achieve a certain amount of uncertainty reduction while minimizing the time to do so".

A. Information Gain

Following Krause and Guestrin [10], we define the information gain as the reduction in Shannon entropy from performing actions. Let Ψ be the random variable over ψ. Then we have

$$IG(\Phi;\Psi_A) = H(\Phi) - H(\Phi|\Psi_A)$$

As they show, this function is monotone submodular if the observations $\Psi_\mathbb{A}$ are conditionally independent given the state ϕ. Thus, if we are evaluating this offline, we would be near-optimal compared to the optimal offline solution.

Unfortunately, this does not hold in the adaptive setting, so we cannot provide guarantees with respect to the optimal policy. Even so, this is one of the most popular heuristics for localization using a greedy maximization [18]. Hsiao points out the connection to non-adaptive submodularity as one reason this heuristic works well even with a shallow search [6].

We use the greedy information gain as one metric for touch localization. To compute the expected reduction in entropy, we need the probability of each observation given the true state. We consider a "blurred" measurement model where the probability of stopping at time o conditioned on a realization ϕ is weighted based on the difference in time between o and the time if ϕ was the state and the sensor was perfect:

$$p(a_\Phi = o|\phi) \propto e^{-\frac{|o-a_\phi|}{2\sigma^2}}$$

Where σ is a parameter determining how noisy we believe the measurements to be. We use Bayes rule to compute the probabilities $p(\phi|a_\Phi = o)$. For any observation o, we can now consider reweighting all hypotheses and computing the resulting entropy. As it is difficult to compute the entropy of the total distribution underlying these hypotheses, we instead fit a Gaussian to the weighted hypotheses and compute the entropy of that distribution. Let Σ_o be the covariance over the weighted set of hypotheses, and N the number of parameters (typically x, y, z, θ). We use the approximated entropy:

$$H(\Phi|o) \approx \frac{1}{2}\ln((2\pi e)^N|\Sigma_o|)$$

Finally, for selecting actions, we take an expectation over observations. Thus, the function used in our minimization becomes:

$$\Delta_{IG}(a) = H(\Phi) - \mathbb{E}_o\left[H(\Phi|o)\right]$$

After performing the selected action, we update the belief by reweighting hypotheses as described above and resampling. We repeat the action selection process, with setting Φ to be the updated distribution, until we reach some desired entropy reduction.

B. Hypothesis Pruning

Intuitively, information gain is reducing the probability mass of the distribution at each step by minimizing the expected variance. Here, we formulate a method with the same underlying idea, which we show to be adaptive submodular and strongly adaptive monotone. We refer to this metric as Hypothesis Pruning, since the idea is to prune away hypotheses which do not agree with the observations so far. This is similar to Active Learning, which involves performing tests to confirm or disprove hypotheses, except with a different probability distribution for test outcomes. Golovin et al. describe the connection between Active Learning and adaptive submodularity [1].

Similar to before, we will consider a blurred version of the probability. We consider two different observation models. In the first, we define a cutoff threshold d_T. If a hypothesis is within the threshold, we keep it entirely. Otherwise, it is removed. We call this metric Hypothesis Pruning (HP). In the second, we downweight the hypotheses with a (non-normalized) Gaussian kernel, and thus remove a portion of the hypothesis. We call this metric Weighted Hypothesis Pruning (WHP). The weighting functions are:

$$w_o^{HP}(a_\phi) = \begin{cases} 1 & \text{if } |o - a_\phi| \leq d_T \\ 0 & \text{else} \end{cases}$$

$$w_o^{WHP}(a_\phi) = e^{-\frac{|o-a_\phi|^2}{2\sigma^2}}$$

For a partial realization ψ, we take the product of weightings:

$$p_\psi(\phi) = \left(\prod_{\{a,o\}\in\psi} w_o(a_\phi)\right) p(\phi)$$

Note that this can never increase the probability - for any actions and observations, $p_\psi(\phi) \leq p(\phi)$.

To calculate how much probability mass m remains with partial realization ψ, and after taking action a and receiving observation o, we use:

$$M_\psi = \sum_{\phi' \in \Phi} p_\psi(\phi')$$
$$m_{\psi,a,o} = \sum_{\phi' \in \Phi} p_\psi(\phi') w_o(a_{\phi'})$$

We can now define the utility of a set of actions if ϕ is the true state. Let A be the sequence of actions taken, and A_ϕ be the sequence of observations received for those actions:

$$f(A, \phi) = 1 - M_{\{A, A_\phi\}}$$

To calculate the expected marginal gain, we also need to define the probability of receiving any observation. We present it here, and show the derivation in the Appendix. Intuitively, this will be proportional to how much probability mass agrees with the observation. Let O be the set of all possible observations:

$$p(a_\Phi = o | \psi) = \frac{m_{\psi,a,o}}{\sum_{o' \in O} m_{\psi,a,o'}}$$

Finally, we define the marginal utility as the additional probability mass removed. For an observation o this is $f_{\psi,a,o} = M_\psi - m_{\psi,a,o}$. Thus, the expected marginal gain is:

$$\begin{aligned}\Delta(a|\psi) &= \mathbb{E}_o \left[f_{\psi,a,o} \right] \\ &= \sum_{o \in O} \frac{m_{\psi,a,o}}{\sum_{o' \in O} m_{\psi,a,o'}} \left[M_\psi - m_{\psi,a,o} \right]\end{aligned}$$

In practice, we need to discretize the infinite observation set O. For an action a, we do so by considering observations exactly at each hypothesis, or $O = \{a_\phi : \phi \in \Phi\}$.

Thus, the greedy algorithm will simply attempt to maximize the expected probability mass removed at each step. After selecting an action and receiving an observation, the hypotheses are downweighted or removed as described above, and the action selection is iterated. Our proofs that this metric is adaptive submodular are in the Appendix.

Relative to this metric, we can therefore guarantee that a greedy algorithm will perform near-optimally. In addition, we can achieve a speedup from a lazy algorithm which does not reevaluate every action at every step [1], [12].

V. Experiments

We implemented a greedy action selection scheme with each of the three metrics described above (IG, HP, WHP). In addition, we compare against two other schemes - random action selection, and a simple human-designed scheme which approaches the object orthogonally along the X, Y and Z axes. The object pose is modeled as a 4-tuple $(x, y, z, \theta) \in \mathbb{R}^4$, where (x, y, z) are the 3-dimensional coordinates of the object's center, and θ is the rotation of the object around the z axis.

We implement our algorithms for the DARPA ARM robot, which has a 7-dof Barret arm with an attached 3-dof Barret hand. The algorithms were applied to localizing two objects: a large drill upright on a table, and a door. We simulate the object of interest being at some true location $X_t \in \mathbb{R}^4$, with an initial sensed location $X_s \in \mathbb{R}^4$. To generate the initial Φ, we sample a Gaussian distribution $N(\mu, \Sigma)$, where $\mu = X_s$, and Σ is the prior covariance of the sensor's noise.

A. Action Generation

In our experiments, we generate straight line actions for the end effector, which consist of a starting 6-dof pose and a movement direction. Each action starts outside of all hypotheses, and moves as far as necessary to hit all hypotheses along that direction. Note that using straight-line trajectories is not a requirement for our algorithm, and simply a design choice for these experiments. We generate actions via three main techniques: randomly sampling a sphere around the object, randomly sampling surface normals, and randomly sampling rays on a plane.

1) Sphere Sampling: To generate starting positions, we sample points on the sphere around sensed position X_s. For each starting position, the end-effector is oriented to face the object, and the movement direction to pass through the object's center. Then, a random rotation is applied to the end-effector about the movement direction, and a random translation to the starting position along the plane orthogonal to the sphere. With this, we get highly randomized actions with different orientations.

2) Normal Sampling: We also generate actions where the end-effector is oriented to move normal to the surface of the object. In this case, we start by sampling a point on the object's triangle mesh. The movement direction is the outward-pointing surface normal of the triangle the point was located on, and the initial orientation of the hand is setup to face the object as in the sphere sampling from section V-A.1. This allows us to generate actions which touch the object orthogonal to the surface, which work well in practice for grasping.

3) Plane Sampling: In addition, we also sample actions starting from arbitrary planes offset from the object - for instance, to touch the table on which a drill rests. The pose of the end effector can be arbitrary, but we to use a pose so that the hand's palm pointed orthogonal to the direction of motion.

B. Experiment Setup

In these tests, we simulate an initial sensor estimation of the object position given as $X_s = (0.975, -0.135, 1.12, 0.00)$ (in meters and radians), with the ground truth position given as $X_t = (0.99, -0.15, 1.11, 0.05)$. The initial probability distribution of the particles from sensor noise, $N(\mu, \Sigma)$ is given with $\mu = X_s$, and the covariance matrix Σ is a diagonal matrix with $\Sigma_{xx} = 0.02$, $\Sigma_{yy} = 0.03$, $\Sigma_{zz} = 0.02$, $\Sigma_{\theta\theta} = 0.08$, and $\Sigma_{ij} = 0$ for all other i, j. We generate an initial set of 1200 hypothesis from this prior distribution, which is our random realization Φ.

We then generate an identical action set A for each metric. The set consists of the 3 human designed trajectories, 50 randomly generated trajectories using the method described in Section V-A.1, 100 randomly generated trajectories using

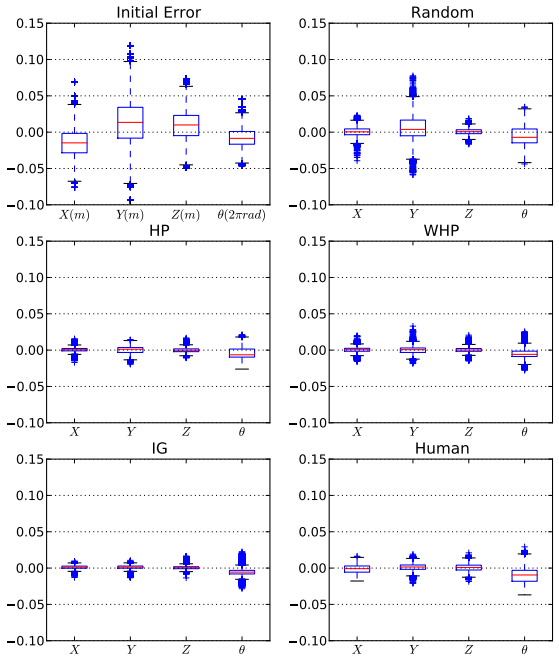


Fig. 3. Resulting realization ϕ of the drill as it compares to the true pose X_t for each metric after all actions have been performed. Here, red lines represent the mean of the particle distribution, the blue boxes represent the ± 25 percentile, the blue horizontal lines the ± 75 percentile, and the blue '+' signs indicate outliers of greater than ± 75 percent. Data is aggregated over all 10 experiments. Values are given in meters for X, Y and Z, and normalized to 2π radians for θ.

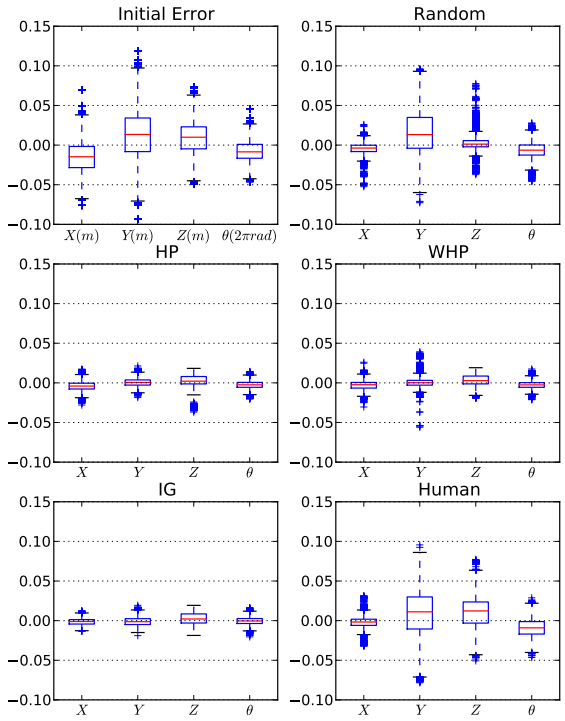


Fig. 4. Resulting realization ϕ of the door as it compares to the true pose X_t for each metric after all actions have been performed, as in Figure 3.

the method described in Section V-A.2, and 10 randomly generated trajectories using the method described in Section V-A.3, giving 163 actions in total to choose from.

Finally, we run 10 experiments using a different random seed for each experiment, generating a different set Φ and A. Each metric chooses a sequence of five actions (except the human designed sequence which is only three actions long).

C. Results

All of the metrics significantly reduced the mean error of the particles on average – confirming our speculation in Section II that even random actions would cause the distribution to converge toward the true pose of the object over time. However, the overall variance of the particle distribution, as well as the speed at which the distribution converges toward the true pose of the object, is greatly affected by the choice of metric.

We find all of our metrics to perform fairly well. In the case of the drill, we found that Information Gain (IG) most reduced the variance along each dimension, even though it was the only metric without strict guarantees in this adaptive setting. This is not surprising, Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP), on the other hand, simply reduce the probability mass, and thus may split the hypotheses into separated groups, having a higher overall variance. As shown by Figure 3, after all actions have been performed, all of the metrics obtain sub-centimeter accuracy on the mean of the distribution. Note that while the human designed trajectories were very good at localizing the position, it performed poorly compared to all three metrics along the θ dimension.

We see similar results for the door (Figure 4), albeit the differences between metrics are even more dramatic. Unlike the drill, the door is not radially symmetric, and its flat surface and protruding handle offer many geometric landmarks that our action selection metrics can exploit. Here, our human-designed sequence of touching the principal axes of the door handle were not sufficient to localize the door.

We also examine the covariance of the total distribution after each action, as in Figure 5. We see that Information Gain reduces the covariance of the distribution faster than the other metrics, even though it is the only method without strict guarantees. Again, this is not surprising, as this metric most directly reduces the variance while Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP) are both content splitting the distribution up. We also see that in terms of the covariance of the distribution, all three action selection metrics (IG, HP, and WHP), perform significantly better than random, and at least as well as the human-designed sequence of actions up to the third action. Notice that even after the human-designed sequence of actions has ended, our

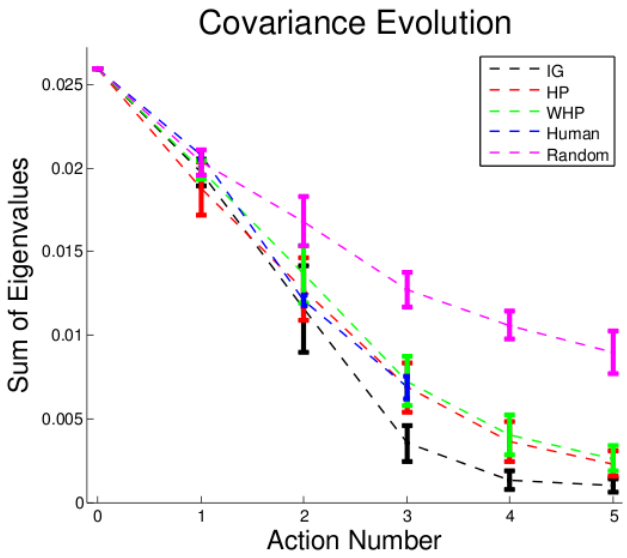


Fig. 5. Noise over the hypotheses for the drill experiments. After each action, the covariance matrix of the current hypotheses are calculated, and the sum of eigenvalues are plotted. Bars show the mean and 95% confidence interval over all 10 experiments.

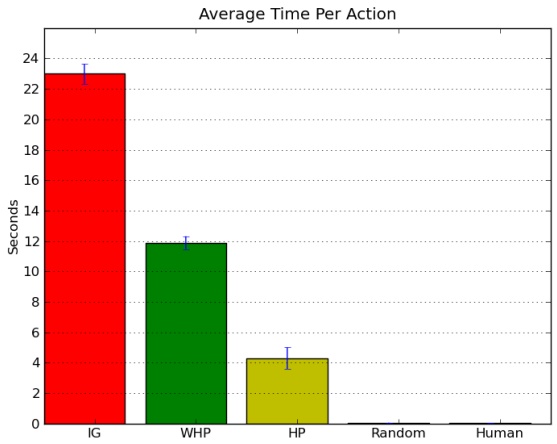


Fig. 6. Time to select the next best action for each of the metrics, averaged over each of the experiments in section V-B.

metrics continue to compute new actions which reduce the uncertainty of the distribution further.

Plotting the actual distributions of the hypotheses for one particular experiment, as in Figure 7, reveals further patterns. For this particular experiment with the drill and a single random seed, Greedy Information Gain (IG) was the clear winner in terms of converging the particle distribution to the true pose of the drill, whereas Hypothesis Pruning (HP), and Weighted Hypothesis Pruning (WHP) performed only marginally better after five actions as the human-designed sequence did after three actions.

Finally, we also display the distribution of drill hypotheses and the selected actions for one experiment in Tables I and II, which display two views of the same set of experiment. We note that the metrics choose fairly different actions, with some similarities between Information Gain (IG) and Weighted Hypothesis Pruning (WHP).

1) Performance: Using the experiments described in Section V-B, we provide timing benchmarks for the action selection phase of the algorithm (after actions have been generated, but before an action has been taken and resampling is performed). Tests were performed on a computer with an Intel Core i7 processor, running on Ubuntu Linux 10.04. As shown in 6, the hypothesis pruning methods select the next best action much faster than the Greedy Information Gain metric, due to the comparative simplicity of their heuristic.

Other phases which are the same for each metric (action generation, and action performance with resampling), also take a significant amount of time , but these could potentially be computed offline and cached for immediate use. In addition, assuming the prior distribution is the same, the very first action could be computed offline and stored.

VI. Conclusion

In this work, we drew a connection between submodularity and touch based localization of objects for grasping. One of the key challenges for action selection under uncertainty is search horizon, making it take many minutes to localize an object of interest. By utilizing submodular functions, good performance can be achieved with a simple greedy algorithm. We present three different metrics based on submodular maximization. The first, Information Gain (IG), does not provide any guarantees in the adaptive setting. However, we observe good performance for this metric. The final two, Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP) are adaptive submodular, and thus have strict guarantees compared to the optimal sequence. In addition, these metrics are much faster to compute, both due to their simplicity and a more efficient lazy-greedy algorithm [1], [12]. We show good performance over all metrics as compared to randomly selecting actions, as well as a human generated sequence.

Currently, our results are all in simulation. We hope to extend this work to use full arm planning with kinematics, and obtain results with a robot.

References

[1] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research (JAIR)*, vol. 42, pp. 427–486, 2011.

[2] C. Papadimitriou and J. N. Tsitsiklis, "The complexity of markov decision processes," *Math. Oper. Res.*, vol. 12, no. 3, pp. 441–450, Aug. 1987.

[3] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *Journal of Artificial Intelligence Research*, vol. 23, p. 2005, 2005.

[4] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based POMDP solvers," *Autonomous Agents and Multi-Agent Systems (AAMAS 2012)*, pp. 1–51, June 2012. [Online]. Available: http://dx.doi.org/10.1007/s10458-012-9200-2

[5] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for pomdps," *J. Artif. Int. Res.*, vol. 32, no. 1, pp. 663–704, July 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622673.1622690

[6] K. Hsiao, "Relatively robust grasping," Ph.D. dissertation, Massachusetts Institute of Technology, 2009.

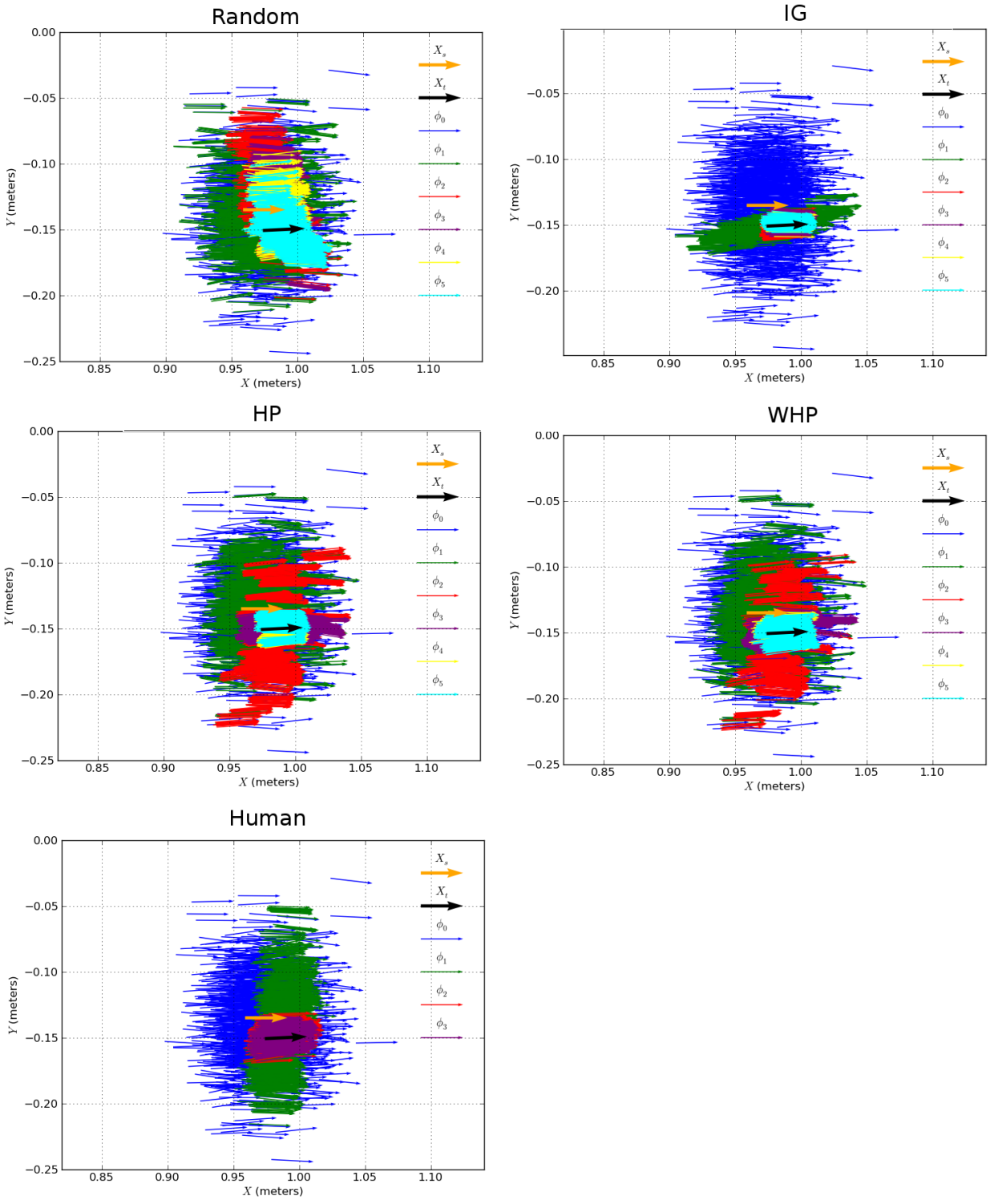


Fig. 7. Plots of the realizations of the drill probability distribution after each action given a single random seed. The positions of the arrows correspond to the (X, Y) coordinates of the particle on the table, and the orientations of the arrows correspond to θ. The length of an arrow is approximately the same as the base of the drill. Arrows are colored according to the action that produced their realization. Both the sensed position X_s and the ground truth position X_t are shown, as well as the initial realization, ϕ_0.

[7] J. Fu, S. Srinivasa, N. Pollard, and B. Nabbe, "Planar batting under shape, pose, and impact uncertainty," in *IEEE International Conference on Robotics and Automation (ICRA)*, April 2007.

[8] U. Feige, "A threshold of ln n for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, July 1998.

[9] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, pp. 385–393, 1982.

[10] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *UAI*, 2005, pp. 324–331.

[11] G. A. H. U. Mitra and G. S. Sukhatme, "Active classication: Theory and application to underwater inspection," in *International Symposium on Robotics Research (ISRR)*, August 2011.

[12] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, ser. Lecture Notes in Control and Information Sciences, J. Stoer, Ed. Springer Berlin / Heidelberg, 1978, vol. 7, pp. 234–243, 10.1007/BFb0006528. [Online]. Available: http://dx.doi.org/10.1007/BFb0006528

[13] *Robust Belief-Based Execution of Manipulation Programs*, 2008.

[14] T. Erez and W. D. Smart, "A scalable method for solving high-dimensional continuous pomdps using local approximation," in *UAI*, 2010, pp. 160–167.

[15] A. Petrovskaya and O. Khatib, "Global localization of objects via touch," *IEEE Trans. on Robotics*, vol. 27, no. 3, pp. 569–585, June 2011.

[16] DARPAtv, "Darpa autonomous robotic manipulation (arm) - phase 1," http://www.youtube.com/watch?v=jeABMoYJGEU.

[17] A. Collet, C. Dellin, M. Dogar, A. Dragan, S. Javdani, K. Strabala, M. V. Weghe, and S. Srinivasa, "Herb prepares a meal," http://www.youtube.com/watch?v=9Oav3JajR7Q.

[18] P. Hebert, J. W. Burdick, T. Howard, N. Hudson, and J. Ma, "Action inference: The next best touch," in *Robotics: Science and Systems Workshop on Mobile Manipulation*, July 2012.

[19] D. Golovin, A. Krause, and D. Ray, "Near-optimal bayesian active learning with noisy observations," in *NIPS*, 2010, pp. 766–774.

[20] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," *Proceedings of the IEEE International Conference on Robotics and Automation (2009)*, vol. 113, pp. 2098–2105, 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5152849

[21] L.-T. Jiang and J. R. Smith, "Seashell effect pretouch sensing for robotic grasping," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2012.

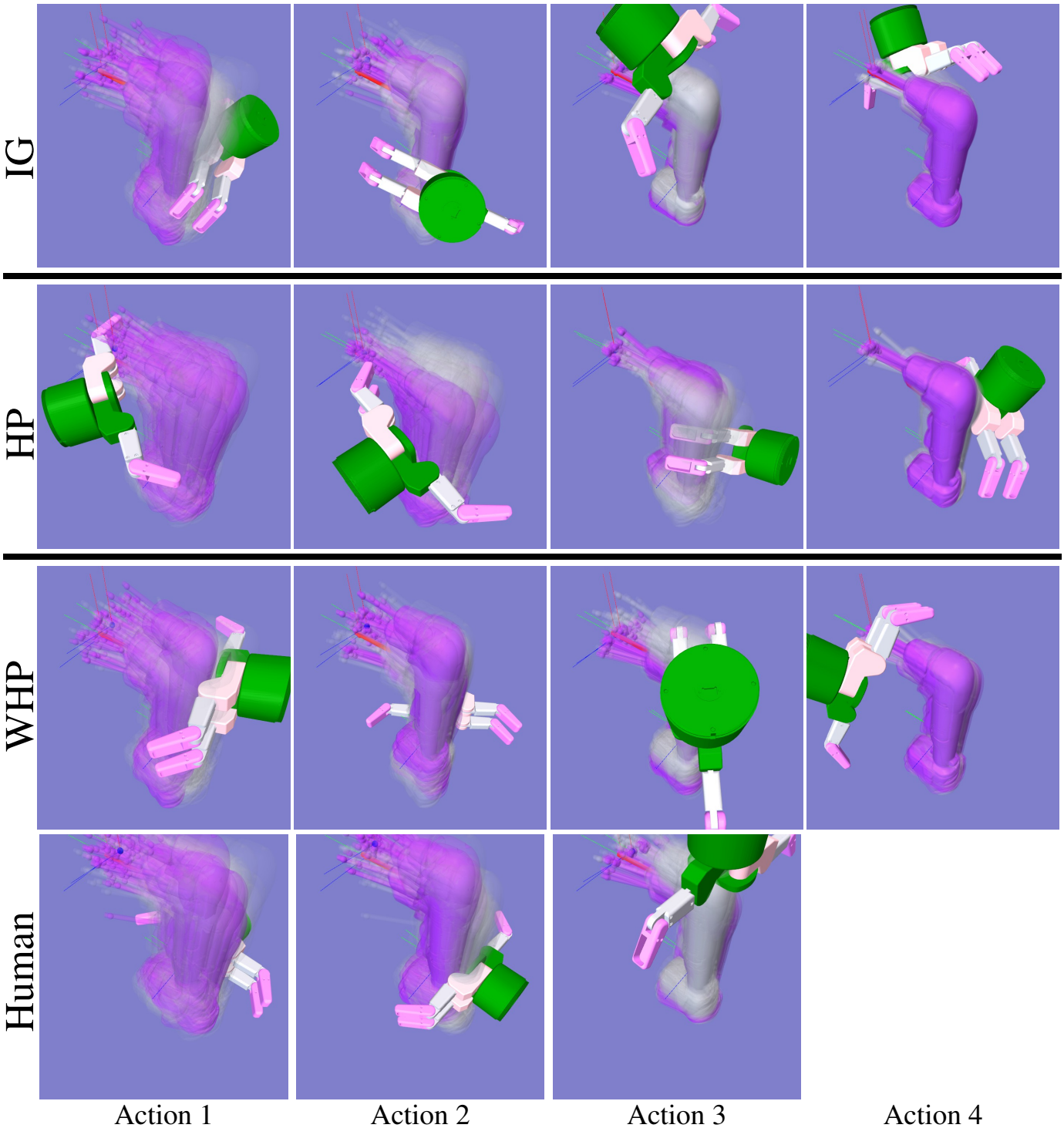


TABLE I

ACTIONS SELECTED FOR DIFFERENT METRICS. THE INITIAL DISTRIBUTION WAS GENERATED USING A NORMAL DISTRIBUTION WITH $\sigma_x = 0.02$, $\sigma_y = 0.02$, $\sigma_z = 0.02$, $\sigma_\theta = 0.2$. RED DISPLAYS THE TRUE STATE, BLUE IS THE CURRENT MEAN OF THE DISTRIBUTION, AND THE CURRENT REALIZATION ϕ IS SHOWN IN PURPLE, AND THE REALIZATION PRIOR TO THE ACTION AND SENSOR FEEDBACK IS SHOWN IN GREY.

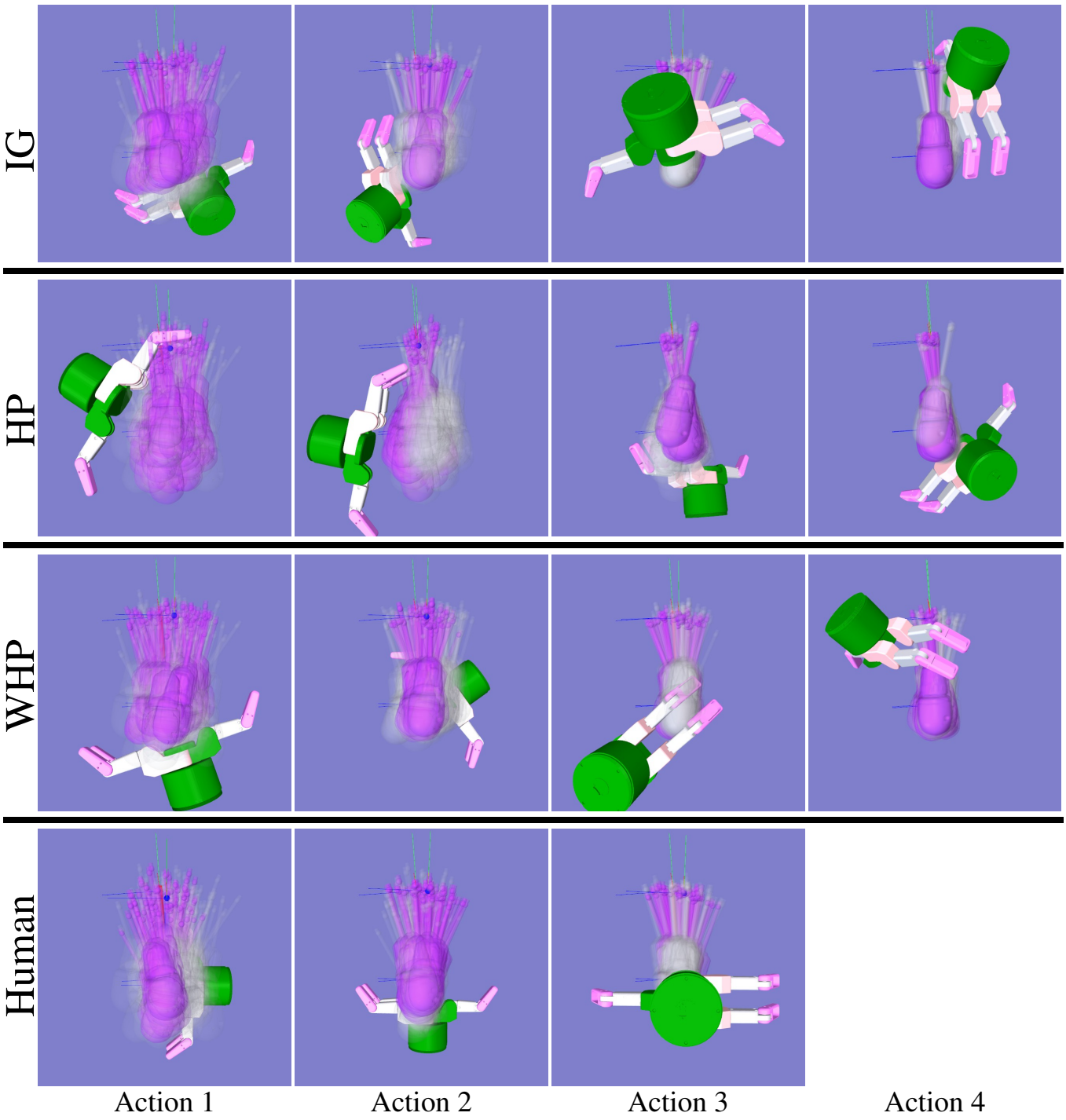


TABLE II

TOP DOWN VIEW. ACTIONS SELECTED FOR DIFFERENT METRICS. THE INITIAL DISTRIBUTION WAS GENERATED USING A NORMAL DISTRIBUTION WITH $\sigma_x = 0.02$, $\sigma_y = 0.02$, $\sigma_z = 0.02$, $\sigma_\theta = 0.2$. RED DISPLAYS THE TRUE STATE, BLUE IS THE CURRENT MEAN OF THE DISTRIBUTION, AND THE CURRENT REALIZATION ϕ IS SHOWN IN PURPLE, AND THE REALIZATION PRIOR TO THE ACTION AND SENSOR FEEDBACK IS SHOWN IN GREY.

VII. Appendix

Here we present the theorems and proofs showing the Hypothesis Pruning metrics are near-optimal. To do so, we prove our metrics are adaptive submodular, strongly adaptive monotone, and self-certifying. We define a function for calculating the total probability mass removed from the original Φ: $\hat{f}(A,\phi) = 1 - M_{\{A,A_\phi\}}$. This function can utilize either of the two reweighting functions w^{HP} or w^{WHP} defined in Section IV-B. Our objective is a truncated version of this: $f(A,\phi) = \min\left\{Q, \hat{f}(A,\phi)\right\}$, where Q is the target value for how much probability mass we wish to remove. We assume that the set of all actions $\mathbb{A}$ is sufficient such that $f(\mathbb{A},\phi) = Q, \forall \phi \in \Phi$. Note that adaptive monotone submodularity is preserved by truncation, so showing these properties for $\hat{f}$ implies them for f.

First, we show how we derive $p(a_\Phi = o|\psi) = \frac{m_{\psi,a,o}}{\sum_{o' \in O} m_{\psi,a,o'}}$. We note that each observation can be accounted for independently when computing $p_\psi(\phi)$ by just taking the product of the weighting functions.

Ideally, we compute as:

$$\begin{aligned} p(a_\Phi = o|\psi) &= \sum_{\phi \in \Phi} p(o|\phi,\psi) p(\phi|\psi) \\ &= \sum_{\phi \in \Phi} p(o|\phi) p(\phi|\psi) \end{aligned}$$

We can think of our weighting function as an unnormalized version of $p(o|\phi)$, and $p_\psi(\phi)$ as an unnormalized version of $p(\phi|\psi)$. Thus, we define an unnormalized version $\hat{p}(a_\Phi = o|\phi)$:

$$\begin{aligned} \hat{p}(a_\Phi = o|\phi) &= \sum_{\phi \in \Phi} w_o(a_\phi) p_\psi(\phi) \\ &= m_{\psi,a,o} \end{aligned}$$

Finally, we need to normalize all observations, so we get:

$$p(a_\Phi = o|\psi) = \frac{m_{\psi,a,o}}{\sum_{o' \in O} m_{\psi,a,o'}}$$

Now we can compute the expected marginal utility:

$$\begin{aligned} \Delta(a|\psi_A) &= \mathbb{E}\left[\hat{f}(A \cup \{a\}, \Phi) - \hat{f}(A,\Phi) \middle| \psi_A\right] \\ &= \sum_{\phi \in \Phi} \left(\sum_{o \in O} p(\phi|o,\psi_A) p(o|\psi_A) \right) \left[(1 - m_{\psi_A,a,o}) - (1 - M_{\psi_A}) \right] \\ &= \sum_{\phi \in \Phi} \sum_{o \in O} p(o|\phi,\psi_A) p(\phi|\psi_A) \left[(1 - m_{\psi_A,a,o}) - (1 - M_{\psi_A}) \right] \\ &= \sum_{o \in O} \sum_{\phi \in \Phi} p(o|\phi,\psi_A) p(\phi|\psi_A) \left[(1 - m_{\psi_A,a,o}) - (1 - M_{\psi_A}) \right] \\ &= \sum_{o \in O} p(o|\psi_A) \left[(1 - m_{\psi_A,a,o}) - (1 - M_{\psi_A}) \right] \\ &= \sum_{o \in O} \frac{m_{\psi_A,a,o}}{\sum_{o' \in O} m_{\psi_A,a,o'}} \left[M_{\psi_A} - m_{\psi_A,a,o} \right] \end{aligned} \tag{1}$$

With this, we are ready to present our guarantees for the Hypothesis Pruning metrics:

Theorem 1: Let $\delta = \min_\phi p(\phi)$. Let our utility function for Hypotheses Pruning be f as defined above, utilizing either of the weighting functions w^{HP} or w^{WHP} defined in section IV-B. Let π^*_{avg} and π^*_{wc} be the optimal policies minimizing the expected and worst-case number of items selected, respectively, to guarantee every realization is covered. The greedy policy π^{greedy} on average costs at most $\left(\ln\left(\frac{Q}{\delta}\right) + 1\right)$ times the average cost of the best policy obtaining reward Q, and $\left(\ln\left(\frac{Q}{\delta^2}\right) + 1\right)$ times the worst case cost of the best policy obtaining reward Q. More formally:

$$\begin{aligned} c_{avg}(\pi^{greedy}) &\leq c_{avg}(\pi^*_{avg}) \left(\ln\left(\frac{Q}{\delta}\right) + 1\right) \\ c_{wc}(\pi^{greedy}) &\leq c_{wc}(\pi^*_{wc}) \left(\ln\left(\frac{Q}{\delta^2}\right) + 1\right) \end{aligned}$$

In order to prove Theorem 1, we will need to show that our objective is adaptive submodular, strongly adaptive monotone, and self-certifying. Then our theorem follows from [1].

Lemma 1: Let $A \subseteq \mathbb{A}$, which result in partial realizations ψ_A. Our objective function defined above is strongly adaptive monotone.

Proof: We need to show that for any action and observation, our objective function will not decrease in value. Intuitively, our objective is strongly adaptive monotone, since we only remove probability mass and never add hypotheses. More formally:

$$\begin{aligned}
\mathbb{E}\left[\hat{f}(A,\Phi)|\psi_A\right] &\leq \mathbb{E}\left[\hat{f}(A\cup\{a\},\Phi)|\psi_A,\psi_a=o\right] \\
\Leftrightarrow 1-M_{\psi_A} &\leq 1-M_{\{\psi_A\cup\{a,o\}\}} \\
\Leftrightarrow 1-M_{\psi_A} &\leq 1-m_{\psi,a,o} \\
\Leftrightarrow m_{\psi,a,o} &\leq M_{\psi_A} \\
\Leftrightarrow \sum_{\phi\in\Phi} p_\psi(\phi)w_o(a_{\phi'}) &\leq \sum_{\phi\in\Phi} p_\psi(\phi)
\end{aligned}$$

As noted before, both of the weighting functions defined in Section IV-B never have a value greater than one. Thus each term in the sum from the LHS is smaller than the equivalent term in the RHS.

■

Lemma 2: Let $X\subseteq Y\subseteq\mathbb{A}$, which result in partial realizations $\psi_X\subseteq\psi_Y$. Our objective function defined above is adaptive submodular.

Proof: For the utility function f to be adaptive submodular, it is required that the following holds over expected marginal utilities:

$$\begin{aligned}
\Delta(a|\psi_Y) &\leq \Delta(a|\psi_X) \\
\sum_{o\in O}\frac{m_{\psi_Y,a,o}}{\sum_{o'\in O}m_{\psi_Y,a,o'}}\left[M_{\psi_Y}-m_{\psi_Y,a,o}\right] &\leq \sum_{o\in O}\frac{m_{\psi_X,a,o}}{\sum_{o'\in O}m_{\psi_X,a,o'}}\left[M_{\psi_X}-m_{\psi_X,a,o}\right]
\end{aligned}$$

We simplify notation a bit for the purposes of this proof. For a fixed partial realization ψ_X and action a, let $m_{\psi_X,a,o}=m_o$. Additionally, we note that for any action a and observation o, it is always true that $m_{\psi_Y,a,o}\leq m_{\psi_X,a,o}$ when $X\subseteq Y$. As noted before, the weighting functions can only remove probability mass. Let $k_o=m_{\psi_X,a,o}-m_{\psi_Y,a,o}$, which represents the difference of probability mass remaining between partial realizations ψ_Y and ψ_X if we performed action a and received observation o. We note that $k_o\geq 0,\forall o$, which follows from the strong adaptive monotonicity, and $k_o\leq m_{\psi_X,a,o}$, which follows from $m_{\psi_Y,a,o}\geq 0$. Rewriting the equation above:

$$\begin{aligned}
\sum_{o\in O}\frac{m_o-k_o}{\sum_{o'\in O}m_{o'}-k_{o'}}\left[M_{\psi_Y}-m_o+k_o\right] &\leq \sum_{o\in O}\frac{m_o}{\sum_{o'\in O}m_{o'}}\left[M_{\psi_X}-m_o\right] \\
\Leftrightarrow \left(\sum_{o\in O}M_{\psi_Y}m_o-m_o^2+m_ok_o-M_{\psi_Y}k_o+m_ok_o-k_o^2\right)\left(\sum_{o'\in O}m_{o'}\right) &\leq \left(\sum_{o\in O}M_{\psi_X}m_o-m_o^2\right)\left(\sum_{o'\in O}m_{o'}-k_{o'}\right) \\
\Leftrightarrow \sum_{o\in O}\sum_{o'\in O}M_{\psi_Y}m_om_{o'}-m_o^2m_{o'}+m_om_{o'}k_o-M_{\psi_Y}m_{o'}k_o+m_om_{o'}k_o-m_{o'}k_o^2 &\leq \sum_{o\in O}\sum_{o'\in O}M_{\psi_X}m_om_{o'}-M_{\psi_X}m_ok_{o'}-m_o^2m_{o'}+m_o^2k_{o'} \\
\Leftrightarrow \sum_{o\in O}\sum_{o'\in O}M_{\psi_Y}(m_om_{o'}-m_{o'}k_o)+2m_om_{o'}k_o-m_{o'}k_o^2 &\leq \sum_{o\in O}\sum_{o'\in O}M_{\psi_X}(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}
\end{aligned}$$

We also note that $M_{\psi_X}-M_{\psi_Y}\geq\max_{\hat{o}\in O}(k_{\hat{o}})$. That is, the total difference in probability mass is greater than or equal to the difference of probability mass remaining if we received any single observation, for any observation.

$$\begin{aligned}
\Leftrightarrow \sum_{o\in O}\sum_{o'\in O}2m_om_{o'}k_o-m_{o'}k_o^2 &\leq \sum_{o\in O}\sum_{o'\in O}(M_{\psi_X}-M_{\psi_Y})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'} \\
\Leftarrow \sum_{o\in O}\sum_{o'\in O}2m_om_{o'}k_o-m_{o'}k_o^2 &\leq \sum_{o\in O}\sum_{o'\in O}\max_{\hat{o}\in O}(k_{\hat{o}})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'} \\
\Leftarrow \sum_{o\in O}\sum_{o'\in O}2m_om_{o'}k_o-m_{o'}k_o^2 &\leq \sum_{o\in O}\sum_{o'\in O}\max(k_o,k_{o'})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}
\end{aligned}$$

In order to show the inequality for the sum, we will show it holds for any pair o,o'. First, if $o=o'$, than we have an equality and it holds trivially. For the case when $o\neq o'$, we assume that $k_o>k_{o'}$ WLOG, and show the inequality for the

sum:

$$\begin{aligned}
2m_o m_{o'}(k_o + k_{o'}) - m_{o'}k_o^2 - m_o k_{o'}^2 &\leq 2m_o m_{o'}k_o - m_o k_{o'}k_o - m_{o'}k_o^2 + m_o^2 k_{o'} + m_{o'}^2 k_o \\
\Leftrightarrow 2m_o m_{o'}k_{o'} - m_o k_{o'}^2 &\leq m_o^2 k_{o'} + m_{o'}^2 k_o - m_o k_o k_{o'} \\
\Leftrightarrow 0 &\leq k_{o'}(m_o - m_{o'})^2 - (k_o - k_{o'})k_{o'}(m_o - m_{o'}) + (k_o - k_{o'})m_{o'}(m_{o'} - k_{o'}) \\
\Leftarrow 0 &\leq k_{o'}(m_o - m_{o'})^2 - (k_o - k_{o'})k_{o'}(m_o - m_{o'}) + (k_o - k_{o'})k_{o'}(m_{o'} - k_{o'})
\end{aligned}$$

We split into 3 cases:

A. $k_{o'} = 0$

This holds trivially, since the RHS is zero

B. $k_{o'} \neq 0, m_o \leq 2m_{o'} - k_{o'}$

Since $k_{o'} \neq 0$, we can rewrite:

$$\begin{aligned}
0 &\leq (m_o - m_{o'})^2 - (k_o - k_{o'})(m_o - m_{o'}) + (k_o - k_{o'})(m_{o'} - k_{o'}) \\
\Leftarrow 0 &\leq -(k_o - k_{o'})(m_o - m_{o'}) + (k_o - k_{o'})(m_{o'} - k_{o'}) \\
\Leftarrow (m_o - m_{o'}) &\leq (m_{o'} - k_{o'})
\end{aligned}$$

Which follows from the assumption for this case.

C. $m_o \geq 2m_{o'} - k_{o'}$

We show this step by induction. Let $m_o = 2m_{o'} - k_{o'} + x, x \geq 0$

Base Case: $x = 0$, which we showed in the previous case.

Induction Assume this inequality holds for $m_o = 2m_{o'} - k_{o'} + x$. Let $\widehat{m_o} = m_o + 1$. We now show that this holds for $\widehat{m_o}$:

$$\begin{aligned}
&0 \leq (\widehat{m_o} - m_{o'})^2 - (k_o - k_{o'})(\widehat{m_o} - m_{o'}) + (k_o - k_{o'})(m_{o'} - k_{o'}) \\
\Leftrightarrow &0 \leq (m_o - m_{o'} + 1)^2 - (k_o - k_{o'})(m_o - m_{o'} + 1) + (k_o - k_{o'})(m_{o'} - k_{o'}) \\
\Leftrightarrow &0 \leq (m_o - m_{o'})^2 - (k_o - k_{o'})(m_o - m_{o'}) + (k_o - k_{o'})(m_{o'} - k_{o'}) + 2m_o - 2m_{o'} + 1 + k_o - k_{o'} \\
\Leftarrow &0 \leq 2m_o - 2m_{o'} + 1 - k_o + k_{o'} && \text{by inductive hypothesis} \\
\Leftarrow &0 \leq m_o + 1 - k_o && \text{by assumption from case} \\
\Leftarrow &0 \leq 1
\end{aligned}$$

And thus, we have shown the inequality holds for any pair o, o'. ■

Finally, it is easy to show that the sum can be decomposed into pairs of o, o'. Therefore, we can see the inequality over the sum also holds. ■

Lemma 3: Let $A \subseteq \mathbb{A}$, which result in partial realizations ψ_A. The utility function f defined above is self-certifying.

Proof: An instance is self-certifying if whenever the maximum value is achieved for the utility function f, it is achieved for all realizations consistent with the observation. See [1] for a more rigorous definition. Golovin and Krause point out that any instance which only depends on the state of items in A is automatically self-certifying (Proposition 5.6 in [1].) That is the case here, since the objective function $f = \min\left\{Q, 1 - M_{\psi_A}\right\}$ only depends on the outcome of actions in A. Therefore, our instance is self-certifying. ■

As we have shown our objective is adaptive submodular, strongly adaptive monotone, and self-certifying, Theorem 1 follows from Theorems 5.8 and 5.9 from [1]. Following their notation, we note that $\eta = \min_\phi p(\phi)$, since it is always true that $f(S, \phi) > Q - \min_\phi p(\phi)$ implies $f(S, \phi) = Q$.