Efficient Touch Based Localization through Submodularity

Shervin Javdani, Matthew Klingensmith, J. Andrew Bagnell, Nancy S. Pollard, Siddhartha S. Srinivasa
The Robotics Institute, Carnegie Mellon University
{sjavdani, mklingen, dbagnell, nsp, siddh}@cs.cmu.edu

Abstract— Many robotic systems deal with uncertainty by performing a sequence of information gathering actions. In this work, we focus on the problem of efficiently constructing such a sequence by drawing an explicit connection to submodularity. Ideally, we would like a method that finds the *optimal* sequence, taking the minimum amount of time while providing sufficient information. Finding this sequence, however, is generally intractable. As a result, many well-established methods select actions greedily. Surprisingly, this often performs well. Our work first explains this high performance – we note a commonly used metric, reduction of Shannon entropy, is submodular under certain assumptions, rendering the greedy solution comparable to the optimal plan in the *offline* setting. However, reacting *online* to observations can increase performance. Recently developed notions of *adaptive submodularity* provide guarantees for a greedy algorithm in this *online* setting. In this work, we develop new methods based on adaptive submodularity for selecting a sequence of information gathering actions online. In addition to providing guarantees, we can capitalize on submodularity to attain additional computational speedups. We demonstrate the effectiveness of these methods in simulation and on a robot.

I. Introduction

Uncertainty is a fundamental problem in robotics. It accumulates from various sources such as noisy sensors, inaccurate models, and poor calibration. This is particularly problematic for *fine manipulation tasks* [1], such as grasping and pushing a small button on a drill, hooking the fingers of a hand around a door handle and turning it (two running examples in our paper), or inserting a key into a keyhole. Because these tasks require high accuracy, failing to account for uncertainty often results in catastrophic failure.

To alleviate these failures, many works perform a sequence of uncertainty reducing actions prior to attempting the task [2]–[6]. In this work, we address the efficient automatic construction of such a sequence when information gaining actions are *guarded moves* [7]. See Fig. 1 for an example sequence which enabled a successful grasp of door handle with a noisy pose estimate.

Ideally, the selected actions reduce uncertainty enough to accomplish the task while optimizing a performance criterion like minimum energy or time. Computing the *optimal* such sequence can be formulated as a Partially Observable Markov Decision Process (POMDP) [8]. However, finding optimal solutions to POMDPs is PSPACE complete [9]. Although several promising approximate methods have been developed [10]–[13], they are still not well suited for many manipulation tasks due to the continuous state and observation spaces.

Previous work on uncertainty reduction utilizes online planning within the POMDP framework, looking at locally
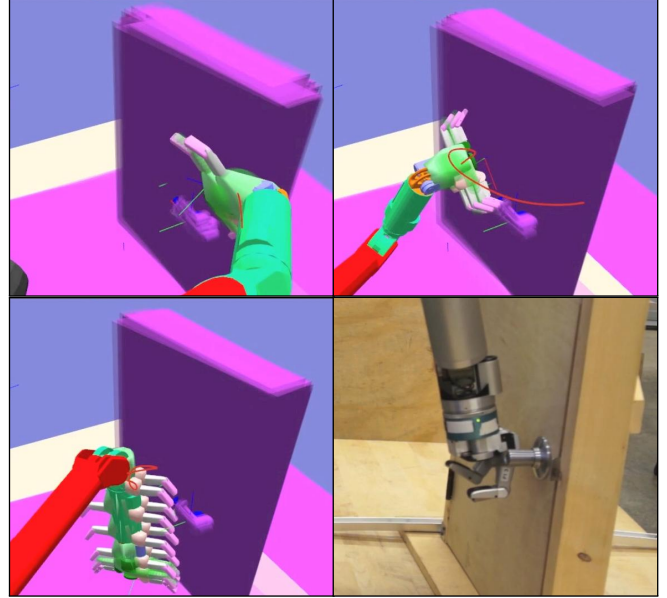

Fig. 1: We adaptively select a sequence of touch actions to reduce uncertainty. Here, we show actions selected by our Hypothesis Pruning method, enabling a successful grasp.

reachable states during each decision step [14]. In general, these methods limit the search to a low horizon [15], often using the *greedy* strategy of selecting actions with the highest expected benefit in one step [2]–[5]. This is out of necessity - computational time increases exponentially with the search depth. However, this simple greedy strategy often works surprisingly well.

One class of problems known to perform well with a greedy strategy 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 the greedy strategy is provably near-optimal. Furthermore, no polynomial time algorithm can guarantee optimality [16], [17].

One often used metric for uncertainty reduction is the expected decrease in Shannon entropy [2]–[6], [18]–[20]. This is referred to as the *information gain* metric, and has been shown to be submodular under certain assumptions [21]. Not surprisingly, many robotic systems which perform well with a low horizon use this metric [2]–[5], [18], though most do not make the connection with submodularity. We note that Hsiao mentions that touch localization could be formulated as a submodular maximization [15].

The guarantees for submodular maximization only hold

in the *non-adaptive* setting. 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. Unfortunately, it has been shown that this can perform exponentially worse than a greedy *adaptive* algorithm for information gain [22]. Thus, while there are no formal guarantees for performing a submodular maximizations online, we might hope for good performance.

Recent notions of *adaptive submodularity* [23] extend the guarantees of submodularity to the adaptive setting, requiring properties similar to those of submodular functions. Unfortunately, information gain does not have these properties. With information gain as our inspiration, we design a similar metric that does. In addition to providing guarantees with respect to that metric, formulating our problem as an adaptive submodular maximization enables a computational speedup through a lazy-greedy algorithm [23], [24] which does not reevaluate every action at each step.

We present three greedy approaches for selecting uncertainty reducing actions. The first is our variant of information gain. This approach is similar to previous works [2]–[6], [18]–[20], though we also enforce the assumptions required for submodular maximization. The latter two maximize the expected number of hypotheses they disprove. We show these metrics are adaptive submodular. We apply all methods to selecting touch based sensing actions and present results comparing the accuracy and computation time of each in Section V. Finally, we show the applicability of these methods on a real robot.

II. Related work

Hsiao et al. [4], [15] select a sequence of uncertainty reducing tactile actions through forward search in a POMDP. Possible actions consist of pre-specified world-relative trajectories [4], motions based on the current highest probability state. Actions are selected using either information gain or probability of success as a metric [15], with a forward search depth of up to three actions. Aggressive pruning and clustering of observations makes online selection tractable. While Hsiao considers a small, focused set of actions (typically ~5) at a greater depth, we consider a broad set of actions (typically ~150) at a search depth of one action.

Hebert et al. [5] independently approached the problem of action selection for touch based localization. They utilize a greedy information gain metric, similar to our own. However, they do not make a connection to submodularity, and provide no theoretical guarantees with their approach. Additionally, they model noise only in X, Y, Z, while we use X, Y, Z, θ. Furthermore, by using a particle based representation instead of a histogram (as in [5], [15]), we can model the underlying belief distribution more efficiently.

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, this plan will likely fail, because the true state was not known. Erez and Smart use local controllers to adjust the trajectory [25].

Platt et al. note when the belief space diverges from what the plan expected, and re-plan from the new belief [26]. They prove their approach will eventually converge to the true hypothesis. While these methods plan significantly faster due to their low-dimensional projection, they pick actions suboptimally. Furthermore, by ignoring part of the belief space, they sacrifice the ability to avoid potential failures. For example, these methods cannot guarantee that a trajectory will not collide and knock over an object, since the planner may ignore the part of the belief space where the object is actually located.

Petrovskaya et al. [27] 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 ~5*mm*, they needed an average of 29 actions for objects with complicated meshes. While this does show that even random actions achieve localization eventually, we note that our methods take significantly fewer actions.

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 tasks [28]. Similar strategies were used to enable a robot to prepare a meal with a microwave [29], where touch-based actions are used prior to pushing buttons. To accomplish these tasks quickly, some of these works rely on hand-tuned motions and policies, specified for a particular object and environment. While this enables very fast localization with high accuracy, a sequence must be created manually for each task and environment. Furthermore, these sequences aren't entirely adaptive.

Dogar and Srinivasa [30] 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. Online, they simply plan a push-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 each approach is applicable in different scenarios.

Outside of robotics, many have addressed the problem of query selection for identification. In the *noise-free* setting, a simple adaptive algorithm known as generalized binary search (GBS) [31] is provably near optimal. Interestingly, this algorithm selects queries identical to greedy information gain if there are only two outcomes [19]. The GBS method was extended to multiple outcomes, and shown to be adaptive submodular [23]. Our Hypothesis Pruning metric is similar to this formulation, but with different action and observation spaces that enable us to model touch actions naturally.

Recently, there have been guarantees made for the case of *noisy* observations. For binary outcomes and independent, random noise, the GBS was extended to noisy generalized

binary search [32]. For cases of persistent noise, where performing the same action results in the same noisy outcome, adaptive submodular formulations have been developed based on eliminating noisy versions of each hypothesis [33], [34]. In all of these cases, the message is the same - with the right formulation, greedy selection performs well for uncertainty reduction.

III. PROBLEM FORMULATION

We review the basic formulation for adaptive submodular maximization. For a more detailed explanation, see [23].

Let a possible object state be ϕ, called the *realization*. Let Φ be a random variable over all realizations. Thus, the probability of a certain state is given by $p(\phi) = \mathbb{P}[\Phi = \phi]$. At each decision step, we select an action a from $\mathbb{A}$, the set of all available actions, which incurs a cost $c(a)$. Each action will result in some observation o from $\mathbb{O}$, the set of all possible observations. We assume that given a realization ϕ, the outcome of an action a is deterministic. Let $A \subseteq \mathbb{A}$ be all the actions selected so far. During execution, we maintain the *partial realization* ψ_A, a sequence of observations received indexed by A. We call it a partial realization as it encodes how realizations $\phi \in \Phi$ agree with observations.

For the case of tactile localization, ϕ is the object pose. $\mathbb{A}$ corresponds to all end-effector guarded move trajectories [7], which terminate when the hand touches an obstacle. $\mathbb{O}$ encompasses any possible observation, which is the set of all distances along any trajectory within which the guarded move may terminate. The partial realization ψ_A 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 actions based on observations so far. 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 average 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 $f : 2^\mathbb{A} \times \mathbb{O}^\mathbb{A} \to \mathbb{R}_{\geq 0}$, which depends on actions selected and observations received. We would like to find a policy which that will reach some utility threshold 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 achieve some coverage Q while minimizing the cost to do so. We can consider optimal policies π^*_{avg} and π^*_{wc} for the above, optimized for their respective cost functions. Unfortunately, obtaining even approximate solutions is difficult [16], [23]. However, a simple greedy algorithm achieves near-optimal performance 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 when we do not condition on observations, optimizing an offline plan. We call a function f submodular if whenever $X \subseteq Y \subseteq \mathbb{A}$, $a \in \mathbb{A} \setminus Y$:

Submodularity (diminishing returns):

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

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):

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

The greedy algorithm maximizes $\frac{f(A \cup \{a\}) - f(A)}{c(a)}$, the marginal utility per unit cost. As observations are not incorporated, this corresponds to an offline plan. If submodularity and monotonicty are satisfied, the greedy algorithm will be within a $(1 + \ln \max_a f(a))$ factor of $c_{avg}(A(\pi^*_{avg}, \Phi))$ for integer valued f [17].

B. Adaptive Submodularity

Now we consider the case where the policy adapts to new observations [23]. In this case, the expected marginal benefit of performing an action is:

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

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

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 from actions $Y \setminus X$. We also require strong adaptive monotonicity, or more items never hurts. For any $a \notin X$, 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 $\frac{\Delta(a|\psi_X)}{c(a)}$. This encodes an online policy, since at each ψ_X incorporates the new observations. Surprisingly, we can bound the performance of the same algorithm with respect to both the optimal average case policy π^*_{avg} and optimal worst case policy π^*_{wc}. This has been shown to be within a $(1 + \ln(Q))$ factor of π^*_{avg}, and a $(1 + \ln(\frac{Q}{\min_\phi p(\phi)}))$ factor of π^*_{wc} approximation for integer valued f, for self-certifying instances (see [23] for a more detailed explanation).

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 as a set of sampled particles. We can think of each particle $\phi \in \Phi$ representing some hypothesis of the true object pose.

Fig. 2: We can think of tactile localization as a problem of set cover, which is adaptive submodular [23]. Each observation amounts to covering (green area) the hypotheses (black dots) which do not agree. Our objective is to maximize our coverage, or rule out as many hypotheses as possible.

Fig. 3: 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.

Each action $a \in \mathbb{A}$ corresponds to an end-effector trajectory which stops when the object is touched. The cost $c(a)$ is the time it would take to run this entire trajectory, plus some fixed time for moving to the start pose. An observation $o \in \mathbb{R}$ corresponds to the time it takes for the end-effector to make contact with the object. We define a_ϕ as the time during trajectory a where contact first occurs if the true state were ϕ. See Figure 3 for an example. If the swept path of a does not contact object ϕ, then $a_\phi = \infty$, which is a valid observation.

With this formulation, we first discuss some assumptions made about interactions with the world. We then present our different utility functions f, which capture the idea of reducing the uncertainty in Φ. In general, our objective will be to achieve a certain amount of uncertainty reduction while minimizing the time to do so.

A. Submodularity Assumptions for Touch Localization

Fitting into the framework of submodular maximization necessitates certain assumptions. First, all actions must be available at every step. Intuitively, this makes sense as a necessity for diminishing returns - if actions are generated at each step, then a new action may simply be better than anything so far. In some sense, non-greedy methods which generate actions based on the current belief state are optimizing both the utility of the current action, and the potential of actions that could be generated in the next step. Instead, we generate a large, fixed set of information gathering trajectories at the start.

Second, we cannot alter the underlying realization ϕ, so actions are not allowed to change the state of the environment or objects. Therefore, we cannot intentionally reposition objects, or model object movement caused by contact.

When applied to object localization, this frameworks lends itself towards heavy objects that remain stationary when touched. For such problems, we believe having an efficient algorithm with guaranteed near-optimality outweighs these limitations. To alleviate some of these limitations, we hope to explore near-touch sensors in the future [35], [36].

B. Information Gain

Information gain has been applied to touch localization before [4], [5]. In contrast to these, we utilize a large fixed set of actions, enforce the assumptions from Section IV-A, and use a particle-based model (as opposed to a histogram).

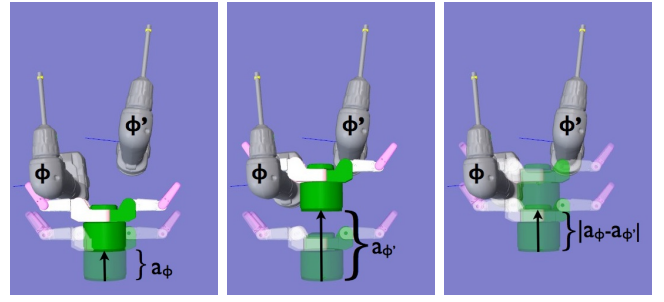Following Krause and Guestrin [21], we define the information gain as the reduction in Shannon entropy. Let Ψ_A be the random variable over ψ_A. Then we have

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

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. However, this can actually perform exponentially worse than the online solution [22]. Therefore, we greedily select actions based on the marginal utility of a single action:

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

We also need to define the probability of an observation. We consider a "blurred" measurement model where the probability of stopping at o conditioned on realization ϕ is weighted based on the time difference between o and a_ϕ (the time of contact had ϕ been the true state), with σ modelling the measurement noise:

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

We could consider evaluating $H(\Phi)$ with a discrete entropy calculation, where each $\phi \in \Phi$ is treated as an individual item. However, our particle set Φ is modeling an underlying continuous distribution, and we would like to capture that. Thus, we instead fit a Gaussian to the current set Φ and evaluate 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:

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

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

C. Hypothesis Pruning

Intuitively, information gain is attempting to reduce uncertainty by removing probability mass. Here, we formulate a method with this underlying idea that also satisfies properties of adaptive submodularity and strong adaptive monotonicity.

We refer to this as Hypothesis Pruning, since the idea is to prune away hypotheses that do not agree with observations. Golovin et al. describe the connection between this type of objective and adaptive submodular set cover [23]. Our formulation is similar - see Fig. 2 for a visualization.

We note that adaptive submodular functions [23] cannot handle noise - they require any realization ϕ be consistent with only one observation per action. However, we would like to model sensor noise. A standard method for alleviating this is to construct a non-noisy problem by generating a realization for every possible noisy observations of every $\phi \in \Phi$. Let $\Omega_A(\phi) = \{\hat{\phi}_1, \hat{\phi}_2, \dots\}$ be the function that generates the new realizations $\hat{\phi}_i$ for actions A. Underlying our formulation, we consider constructing this non-noisy problem. Luckily, we can compute our objective function on the original Φ, and do not need to explicitly perform this construction. We present this more efficient computation below, and show how to construct the equivalent non-noisy problem in Section VII.

As before, we consider a "blurred" measurement model through 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 hypotheses with a (non-normalized) Gaussian, effectively removing a portion of the hypothesis. We call this metric Weighted Hypothesis Pruning (WHP). The weighting functions are:

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

$$\omega_o^{WHP}(a_\phi) = \exp\left(-\frac{|o - a_\phi|^2}{2\sigma^2}\right)$$

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

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

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

Define M_ψ as the non-normalized probability remaining after partial realization ψ, and $m_{\psi,a,o}$ as the probability remaining after an additional action a and observation o:

$$M_\psi = \sum_{\phi' \in \Phi} p_\psi(\phi')$$

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

We can now define our objective function for any partial realization ψ, corresponding to removing probability mass:

$$f(\psi) = 1 - M_\psi$$

In practice, we need to discretize the infinite observation set $\mathbb{O}$. Formally, we require that an equal number of observations per ϕ are considered. That is, for any action a

and any realizations $\phi_i, \phi_j, |\Omega_a(\phi_i)| = |\Omega_a(\phi_j)|$[1]. In practice, we sample observations uniformly along the trajectory to approximately achieve this effect.

To calculate the expected marginal gain, we also need to define the probability of receiving an observation. We present it here, and show the derivation in Section VII. Intuitively, this will be proportional to how much probability mass agrees with the observation. Let $\mathbb{O}_a$ be the set of all possible observations for action a:

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

We note the marginal utility is the additional probability mass removed. For an action a and observation o this is $f_{\psi,a,o} = M_\psi - m_{\psi,a,o}$. The expected marginal gain is:

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

The greedy policy π^{greedy} maximizes the expected probability mass removed per unit cost, or $\frac{\Delta(a|\psi)}{c(a)}$. After selecting an action and receiving an observation, hypotheses are removed or downweighted as described above, and action selection is iterated. We now present our main guarantee:

Theorem 1: Let our utility function be f as defined above, utilizing either weighting function ω^{HP} or ω^{WHP}. Define a quality threshold Q such that $\min_{\phi, \Omega_{\mathbb{A}}(\phi)} f(\{\mathbb{A}, \mathbb{A}_{\Omega_{\mathbb{A}}(\phi)}\}) = Q$.[2] Let η be any value such that $f(\psi) > Q - \eta$ implies $f(\psi) \geq Q$ for all ψ. Let π_{avg}^* and π_{wc}^* be the optimal policies minimizing the expected and worst-case number of items selected, respectively. The greedy policy π^{greedy} satisfies:

$$c_{avg}(\pi^{greedy}) \leq c_{avg}(\pi_{avg}^*)\left(\ln\left(\frac{Q}{\eta}\right) + 1\right)$$

$$c_{wc}(\pi^{greedy}) \leq c_{wc}(\pi_{wc}^*)\left(\ln\left(\frac{Q}{\delta\eta}\right) + 1\right)$$

With δ a constant based on the underlying non-noisy problem (see Section VII).

Proof: In order to prove Theorem 1, we show that our objective f is adaptive submodular, strongly adaptive monotone, and self-certifying in an equivalent non-noisy problem. We show this in our Section VII. ∎

If we utilize ω^{HP} as our weighting function, we can use $\eta = \min_\phi p(\phi)$. If we utilize ω^{WHP}, η is related to how we discretize observations.

In addition to being within a logarithmic factor of optimal, we utilize a lazy-greedy algorithm which does not reevaluate all actions at every step, speeding up computation [23], [24].

---

[1]Note that we must be consistent between contact and no-contact observations. That is, if we believe action a will contact ϕ_i but not ϕ_j, it still must be that $|\Omega_a(\phi_i)| = |\Omega_a(\phi_j)|$. Thus, we also have multiple noisy no-contact observations. See Section VII for details.

[2]If we have a target uncertainty Q', we can define a truncated function $g(\psi) = \min(f(\psi), Q')$ to decrease our quality threshold. Truncation preserves adaptive submodularity [23], so g is adaptive submodular if f is.

V. EXPERIMENTS

We implement a greedy action selection scheme with each of the 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. Each object pose ϕ consist of a 4-tuple $(x, y, z, \theta) \in \mathbb{R}^4$, where (x, y, z) are the coordinates of the object's center, and θ is the rotation about the z axis.

We implement our algorithms using a 7-dof Barret arm with an attached 4-dof Barret hand. We localize two objects: a drill upright on a table, and a door. We define 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. For simulation experiments, we also define the ground truth pose $X_t \in \mathbb{R}^4$.

For efficiency purposes, we also use a fixed number of particles $|\Phi|$ at all steps, and resample after each selection, adding small noise to the resampled set of particles.

A. Action Generation

We generate linear motions of the end effector, consisting of a starting pose and a movement vector. Each action starts outside of all hypotheses, and moves as far as necessary to contact every hypotheses along the path. Note that using straight-line trajectories is not a requirement for our algorithm. We generate actions via three main techniques.

1) Sphere Sampling: Starting positions are generated by sampling a sphere around the sensed position X_s. For each starting position, the end-effector is oriented to face the object, and the movement direction set to X_s. A random rotation is applied about the movement direction, and a random translation along the plane orthogonal to the movement.

2) Normal Sampling: These actions are intended to have the hand's fingers contact the object orthogonally. First, we uniformly sample random contacts from the surface of the object. Then, for each fingertip, we align its pre-defined contact point and normal with our random sample, and randomly rotate the hand about the contact normal. We then set the movement direction as the contact normal.

3) Table Contacting: We generate random start points around the sensed position X_s, and orient the end effector in the $-z$ direction. These are intended to contact the table and reduce uncertainty in z.

B. Simulation Experiments Setup

We simulate an initial sensor error as $X_t - X_s = (0.015, -0.015, -0.01, 0.05)$ (in meters and radians). Our initial random realization Φ is sampled from $N(\mu, \Sigma)$ with $\mu = X_s$, and Σ a diagonal matrix with $\Sigma_{xx} = 0.03$, $\Sigma_{yy} = 0.03$, $\Sigma_{zz} = 0.03$, $\Sigma_{\theta\theta} = 0.1$. We fix $|\Phi| = 1500$ hypotheses.

We then generate an identical action set $\mathbb{A}$ for each metric. The set consists of the 3 human designed trajectories, 30 sphere sampled trajectories (Section V-A.1), 160 normal trajectories (Section V-A.2), and 10 table contact trajectories (Section V-A.3), giving $|\mathbb{A}| = 203$.
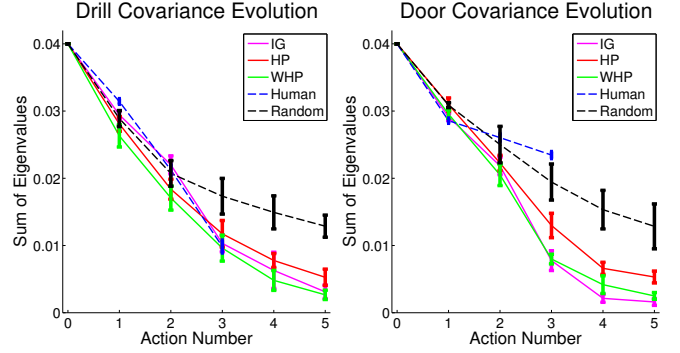

Fig. 4: Uncertainty after each action for drill and door experiments. The bars show the mean and 95% CI of the sum of eigenvalues of the covariance matrix over experiments described in in Section V-B.

	IG	HP	WHP
Time (s)	47.171 ± 0.25	$\mathbf{8.41 \pm 0.58}$	25.70 ± 0.29

TABLE I: Time to select one action for each metric, average and 95% CI over drill experiments described in Section V-B

We run 10 experiments using a different random seed for each, generating a different set $\mathbb{A}$ and Φ, but ensuring each metric has the same $\mathbb{A}$ and initial Φ for a random seed. Each method chooses a sequence of five actions, except the human designed sequence which consists of only three actions.

C. Simulation Experiments Results

We analyze the uncertainty reduction of each metric as the sum of eigenvalues of the covariance matrix, as in Fig. 4. All metrics were able to reduce the uncertainty significantly – confirming our speculation in Section II that even random actions reduce uncertainty. However, as the uncertainty is reduced, the importance of action selection increases, as evidenced by the relatively poor performance of random selection for the later actions. Additionally, we see the human designed trajectories are effective for the drill, but perform poorly on the door. Unlike the drill, the door is not radially symmetric, and its flat surface and protruding handle offer geometric landmarks that our action selection metrics can exploit, making action selection more useful.

For one drill experiment, we also display the hypothesis set after each action in Fig. 5, and the first 3 actions selected in Table II. Note that the actions selected are very different, while performance appears similar.

Observation 1: Information Gain (IG), Hypothesis Pruning (HP), and Weighted Hypothesis Pruning (WHP) all perform similarly well. On the one hand, you might expect IG to perform poorly with adaptive greedy selection, as we don't have any guarantees. On the other, Shannon entropy has many properties that make it a good measure of uncertainty. Fig. 4 displays the covariance of all particles, which is the criterion IG directly optimizes. Note that, surprisingly, HP and WHP perform comparably despite not directly optimizing this measure.

Observation 2: The HP and WHP perform faster than IG. This is due to their inherent simplicity and the more efficient
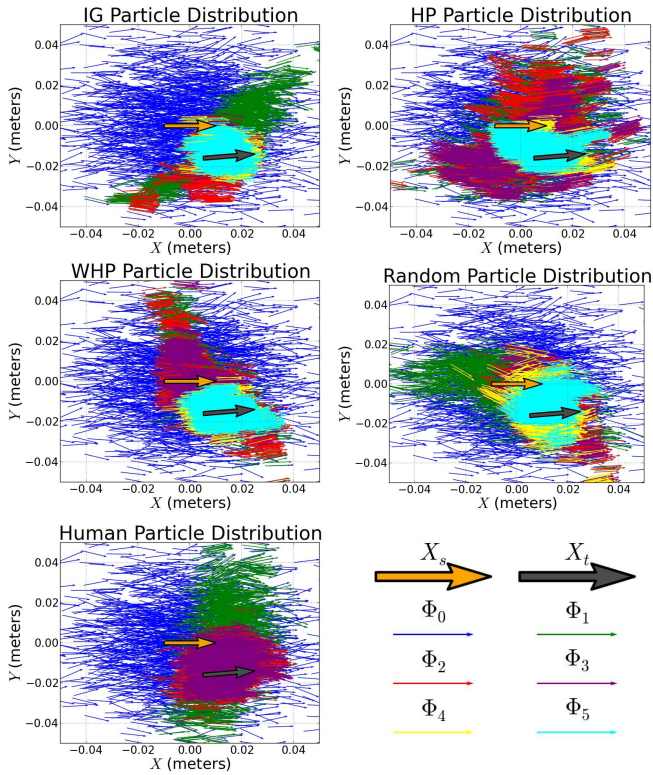
Fig. 5: The particle sets Φ from a single drill experiment after each update. Plotted positions corresponds to the x, y parameter of each $\phi \in \Phi$, rotated by θ. X_s is the sensed position, X_t the true position, and Φ_i the particles after update i. Arrow lengths are approximately the length of the drill base. The initial Φ was sampled from a normal distribution with $\sigma_x = 0.02$, $\sigma_y = 0.02$, $\sigma_z = 0.02$, $\sigma_\theta = 0.2$.

lazy-greedy algorithm [23], [24]. See Table I for times. Additionally, we lose little performance with large computational gains with the non-weighted observation model of HP.

D. Robot Experiments

We implemented each of our methods (IG, HP, WHP) on a robot with a Barret arm and hand, and attempted to open a door. X_s is initialized with a vision system corrupted with an artificial error of $0.035m$ in the y direction. Our initial random realization Φ is sampled from $N(\mu, \Sigma)$ with $\mu = X_s$, and Σ a diagonal matrix with $\Sigma_{xx} = 0.02$, $\Sigma_{yy} = 0.04$, $\Sigma_{zz} = 0.02$, $\Sigma_{\theta\theta} = 0.08$. We fix $|\Phi| = 2000$ hypotheses. We initially generate 600 normal action trajectories (Section V-A.2), though after checking for kinematic feasibility, only about 70 remain.

We utilize each of our uncertainty reducing methods prior to using an open-loop sequence to grasp the door handle. Once our algorithm selects the next action, we motion plan to its start pose and perform the straight line guarded-move using a task space controller. We sense contact by thresholding the magnitude reported by a force torque sensor in the Barret hand.

Without touch localization, the robot missed the door handle entirely. With any of our localization methods, the robot successfully opened the door, needing only two uncertainty reducing actions to do so. Selected actions are shown in
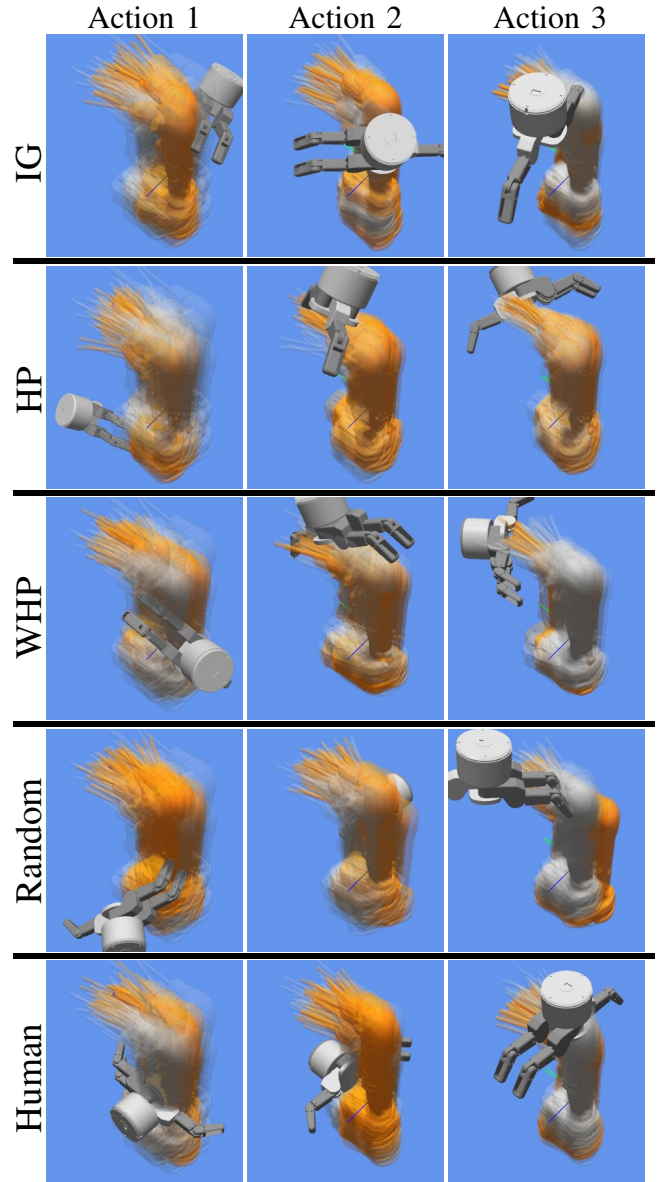

TABLE II: First three actions selected for each metric from the experiment in Fig. 5. Particles prior to the action are grey, and particles updated after observation are yellow.

Table III, and full videos are provided online[3].

Observation 3: Using our faster adaptive submodular metrics, selecting an action takes approximately as long as planning and executing it. This suggests that adaptive action selection will often outperform a non-adaptive plan generated offline that requires no planning time, but more actions.

VI. CONCLUSION AND FUTURE WORK

In this work, we aim to show that greedy selection with the proper metric works well for information gathering actions, both theoretically and practically. To do so, we draw an explicit connection between submodularity and touch based localization. We start with Information Gain (IG), which has been used extensively for uncertainty reduction [2]–[6], [18],

[3]http://www.youtube.com/watch?v=_HiyKKDStBE

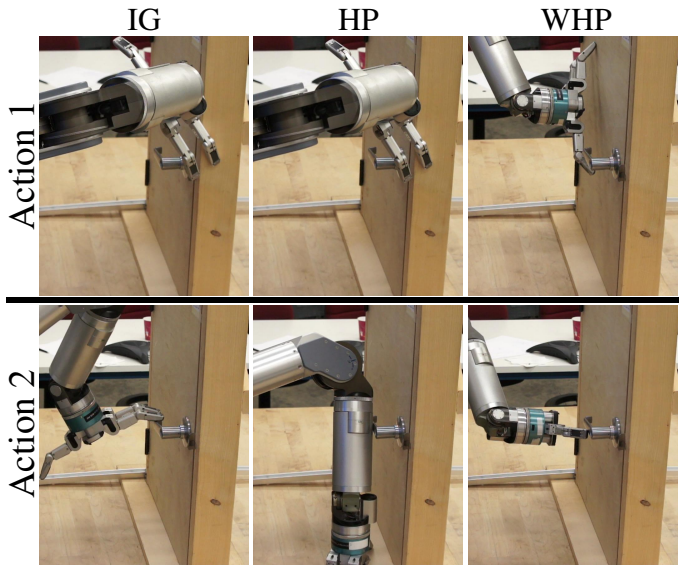| IG	HP	WHP

TABLE III: Actions selected during robot experiment. Interestingly, IG and HP select the same first action. All metrics led to a successful grasp of the door handle.

[20]. We note the assumptions necessary for this metric to be submodular, rendering the greedy algorithm near-optimal in the *offline* setting. We design our own metrics, Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP), which we show are adaptive submodular. Thus, greedy selection is guaranteed to provide near-optimal performance in the *online* setting. In addition, these metrics are much faster, both due to their simplicity and a more efficient lazy-greedy algorithm [23], [24].

The methods presented here naively reduce uncertainty without considering the underlying task. In actuality, a task and planner may not require the exact pose, but that all uncertainty lie within a particular distribution. While we can apply our naive methods and terminate when this holds, we may achieve the task-based target more quickly by optimizing for it directly.

ACKNOWLEDGMENTS

REFERENCES

[1] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," *IJRR*, vol. 3, no. 1, pp. 3–24, 1984.

[2] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: Discrete bayesian models for mobile-robot navigation," in *IEEE/RSJ IROS*, 1996.

[3] D. Fox, W. Burgard, and S. Thrun, "Active markov localization for mobile robots," *Robotics and Autonomous Systems*, vol. 25, pp. 195–207, 1998.

[4] K. Hsiao, T. Lozano-Pérez, and L. P. Kaelbling, "Robust belief-based execution of manipulation programs," in *WAFR*, 2008.

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

[6] J. Fu, S. Srinivasa, N. Pollard, and B. Nabbe, "Planar batting under shape, pose, and impact uncertainty," in *IEEE ICRA*, 2007.

[7] P. M. Will and D. D. Grossman, "An experimental system for computer controlled mechanical assembly," *IEEE Trans. Computers*, vol. 24, no. 9, pp. 879–888, 1975.

[8] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artificial Intelligence*, vol. 101, pp. 99–134, 1998.

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

[10] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *JAIR*, vol. 23, pp. 1–40, 2005.

[11] T. Smith and R. G. Simmons, "Point-based POMDP algorithms: Improved analysis and implementation," in *UAI*, 2005.

[12] H. Kurniawati, D. Hsu, and W. S. Lee, "Sarsop: Efficient point-based pomdp planning by approximating optimally reachable belief spaces," in *RSS*, 2008.

[13] G. Shani, J. Pineau, and R. Kaplow, "A survey of point-based pomdp solvers," *AAMAS*, pp. 1–51, 2012.

[14] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for pomdps," *JAIR*, vol. 32, pp. 663–704, 2008.

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

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

[17] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[18] F. Bourgault, A. A. Makarenko, S. B. Williams, B. Grocholsky, and H. F. Durrant-Whyte, "Information based adaptive robotic exploration," in *IEEE/RSJ IROS*, 2002.

[19] A. X. Zheng, I. Rish, and A. Beygelzimer, "Efficient test selection in active diagnosis via entropy approximation," in *UAI*, 2005.

[20] L. Erickson, J. Knuth, J. M. O'Kane, and S. M. LaValle, "Probabilistic localization with a blind robot," in *IEEE ICRA*, 2008.

[21] A. Krause and C. Guestrin, "Near-optimal nonmyopic value of information in graphical models," in *UAI*, 2005.

[22] G. A. Hollinger, U. Mitra, and G. S. Sukhatme, "Active classication: Theory and application to underwater inspection," in *ISRR*, 2011.

[23] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *JAIR*, vol. 42, no. 1, pp. 427–486, 2011.

[24] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, 1978, vol. 7.

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

[26] R. Platt, L. Kaelbling, T. Lozano-Pérez, and R. Tedrake, "A hypothesis-based algorithm for planning and control in non-gaussian belief spaces," Massachusetts Institute of Technology, Tech. Rep. CSAIL-TR-2011-039, 2011.

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

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

[29] 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.

[30] M. Dogar and S. Srinivasa, "Push-grasping with dexterous hands: Mechanics and a method," in *IEEE/RSJ IROS*, 2010.

[31] R. Nowak, "Generalized binary search," in *Proc. Allerton Conference on Communications, Control, and Computing*, 2008.

[32] ——, "Noisy generalized binary search," in *NIPS*, 2009.

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

[34] G. Bellala, S. K. Bhavnani, and C. Scott, "Group-based active query selection for rapid diagnosis in time-critical situations," *IEEE Transactions on Information Theory*, vol. 58, no. 1, pp. 459–478, 2012.

[35] K. Hsiao, P. Nangeroni, M. Huber, A. Saxena, and A. Y. Ng, "Reactive grasping using optical proximity sensors," in *IEEE ICRA*, 2009.

[36] L.-T. Jiang and J. R. Smith, "Seashell effect pretouch sensing for robotic grasping," in *IEEE ICRA*, 2012.

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. Note that the bounds on adaptive submodular functions require that observations are not noisy - that is, for a fixed realization ϕ, an action is consistent with exactly one observation. In our case, we would like multiple observations to be consistent with a ϕ, as our sensors are noisy. Thus, we first construct a non-noisy problem by creating many weighted "noisy" copies of each realization ϕ. We then show how to compute our objective on the original problem. Finally, we prove our bounds.

A. Constructing the Non-Noisy Problem

Similar to previous approaches in active learning, we construct a non-noisy problem by creating "noisy" copies of each realization ϕ for every possible noisy observation [33], [34]. Let $\Omega_a(\phi) = \{\hat{\phi}_1, \dots \hat{\phi}_K\}$ be the function that creates K noisy copies for action a. Here, the original probability of ϕ is distributed among all the noisy copies, $p(\phi) = \sum_{\hat{\phi} \in \Omega_a(\phi)} p(\hat{\phi})$. For convenience, we will also consider overloading Ω to take sets of actions, and sets of realizations. Let $\Omega_A(\phi)$ recursively apply Ω for each $a \in A$. That is, if we let $A = \{a_1, a_2, \dots\}$ we apply Ω_{a_1} to ϕ, then Ω_{a_2} to every output of $\Omega_{a_1}(\phi)$, and so on. Note that we still have $p(\phi) = \sum_{\hat{\phi} \in \Omega_A(\phi)} p(\hat{\phi})$. Additionally, let $\Omega_A(\Phi)$ apply Ω_A to each $\phi \in \Phi$ and combine the set.

The probability of each noisy copy comes from our weighting functions defined in Section IV-C:

$$\Omega_A(\phi) = \{\hat{\phi}_1, \dots \hat{\phi}_K\}$$

$$p(\hat{\phi}) = p(\phi) \frac{\omega_{a_{\hat{\phi}}}(a_\phi)}{\sum_{\hat{\phi}' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}'}}(a_\phi)} \qquad \text{(one action)}$$

$$p(\hat{\phi}) = p(\phi) \prod_{a \in A} \frac{\omega_{a_{\hat{\phi}}}(a_\phi)}{\sum_{\hat{\phi}' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}'}}(a_\phi)} \qquad \text{(multiple actions)}$$

For simplicity, we also assume that the maximum value of our weighting function is equal to one for any action. We note that our weighting functions in Section IV-C have this property for the non-noisy observation where $a_{\hat{\phi}} = a_\phi$:

$$\max_{\hat{\phi} \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}}}(a_\phi) = 1 \quad \forall \phi, a \qquad (1)$$

We build our set of non-noisy realizations $\widehat{\Phi} = \Omega_{\mathbb{A}}(\Phi)$. Our objective function is over $\widehat{\Phi}$, specifying the probability mass removed form the original problem. One property we desire is if our observations are consistent with one noisy copy of ϕ, then we keep some proportion of all of the noisy copy (proportional to our weighting function ω^{HP} or ω^{WHP}). In our HP algorithm for example, if any noisy copy of ϕ remains, the objective function acts as if all of the probability mass remains. Following Golovin and Krause [23], we define our utility function over a set of actions A, and the observations we would receive if $\hat{\phi}$ were the true state:

$$f(A, \hat{\phi}) = 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi} \in \Omega_{\mathbb{A}}(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

Where $\delta_{a_{\hat{\phi}} a_{\hat{\phi}'}}$ is the Kronecker delta function, equal to 1 if $a_{\hat{\phi}} = a_{\hat{\phi}'}$ and 0 otherwise, ϕ is the original realization from which $\hat{\phi}$ was produced, and $\max p(\Omega_a(\phi))$ is the highest probability of the "noisy" copies. By construction, any action will keep at most $\max p(\Omega_a(\phi))$ probability mass per action, since at most one noisy copy from $\Omega_a(\phi)$ will be consistent with the observation. Intuitively, multiplying by $\frac{p(\phi)}{\max p(\Omega_a(\phi))}$ will make it so if we kept the highest weighted noisy copy of ϕ, our

objective would be equivalent to keeping the entire realization ϕ.

$$f(A,\hat{\phi}) = 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi}' \in \Omega_{\mathbb{A}}(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} \sum_{\hat{\phi}'' \in \Omega_{\mathbb{A} \setminus A}(\hat{\phi}')} p(\hat{\phi}'') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}''}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} \left(\prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right) \sum_{\hat{\phi}'' \in \Omega_{\mathbb{A} \setminus A}(\hat{\phi}')} p(\hat{\phi}'') \right)$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

Here, we separate the recursive splitting over the realization ϕ into those split based on actions in A and those split from other actions. Since $\prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}''}}$ only depends on the response to actions in A, it only depends on noisy copies made from Ω_A. Thus, we can factor those out. Additionally, we marginalize over the copies of $\hat{\phi}'$ as $\sum_{\hat{\phi}'' \in \Omega_{\mathbb{A} \setminus A}(\hat{\phi}')} p(\hat{\phi}'') = p(\hat{\phi}')$. Overall, this simplification enables us to only consider the copies from the actions in A. We further simplify:

$$f(A,\hat{\phi}) = 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \frac{p(\phi)}{\max p(\Omega_a(\phi))} \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} p(\phi) \left(\frac{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)}{\max_{\hat{\phi}' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}'}}(a_\phi) p(\phi)} \right) \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right) \tag{2}$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi) \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} p(\hat{\phi}') \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right) \tag{3}$$

$$= 1 - \sum_{\phi \in \Phi} \left(\prod_{a \in A} \sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi) \right) \left(\sum_{\hat{\phi}' \in \Omega_A(\phi)} p(\phi) \left(\prod_{a \in A} \frac{\omega_{a_{\hat{\phi}'}}(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \right) \prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi}' \in \Omega_A(\phi)} \left(\prod_{a \in A} \sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi) \right) \left(\prod_{a \in A} \frac{\omega_{a_{\hat{\phi}'}}(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \right) \left(\prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi}' \in \Omega_A(\phi)} \left(\prod_{a \in A} \omega_{a_{\hat{\phi}'}}(a_\phi) \right) \left(\prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}} \right)$$

$$= 1 - \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi}' \in \Omega_A(\phi)} \prod_{a \in A} \omega_{a_{\hat{\phi}'}}(a_\phi) \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}}$$

Where (2) plugging in the value of $\Omega_a(\phi)$, (3) used (1) above. Now we consider how the function Ω generates noisy copies. We require that exactly one noisy copy $\hat{\phi}' \in \Omega_A(\phi)$ agree with every observation received so far, and thus only one term will have a nonzero product $\prod_{a \in A} \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}}$. We defer further specific details of Ω until the next section. We get:

$$f(A,\hat{\phi}) = 1 - \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi}' \in \Omega_A(\phi)} \prod_{a \in A} \omega_{a_{\hat{\phi}'}}(a_\phi) \delta_{a_{\hat{\phi}} a_{\hat{\phi}'}}$$

$$= 1 - \sum_{\phi \in \Phi} p(\phi) \prod_{a \in A} \omega_{a_{\hat{\phi}}}(a_\phi)$$

At this point we can see how this equals the objective function for partial realization ψ in Section IV-C. Let $\psi_{\hat{\phi}} = \{A, A_{\hat{\phi}}\}$,

where A are the actions taken and $A_{\hat{\phi}}$ are the observations. We can see that:

$$f(A,\hat{\phi}) = 1 - \sum_{\phi \in \Phi} p(\phi) \prod_{a \in A} \omega_{a_{\hat{\phi}}}(a_\phi)$$

$$= 1 - \sum_{\phi \in \Phi} p_{\psi_{\hat{\phi}}}(\phi)$$

$$= 1 - M_{\psi_{\hat{\phi}}}$$

$$= f(\psi_{\hat{\phi}})$$

B. Observation Probabilities

To compute expected marginal utilities, we will need to define our space of possible observations, and the corresponding probability for these observations. Let $\psi = \{\tilde{A}, \tilde{O}\}$ be the partial realization, and $p(a_\Phi = o | \psi)$ be the probability of receiving observation o from performing action a conditioned on the partial realization ψ. Intuitively, this will correspond to how much probability mass agrees with the observation. More formally:

$$p(a_\Phi = o | \psi) \propto \sum_{\phi \in \Phi} \sum_{\hat{\phi} \in \Omega_{\mathbb{A}}(\phi)} p(\hat{\phi}) \delta_{a_{\hat{\phi}} o} \prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \delta_{\tilde{a}_{\hat{\phi}} \tilde{o}}$$

Similar to before, we will be able to consider noisy copies made from only actions in $\tilde{A}$ and a (the derivation follows exactly as in Section VII-A). This will simplify to:

$$p(a_\Phi = o | \psi) \propto \sum_{\phi \in \Phi} \sum_{\hat{\phi} \in \Omega_{\{\tilde{A} \cup a\}}(\phi)} p(\hat{\phi}) \delta_{a_{\hat{\phi}} o} \prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \delta_{\tilde{a}_{\hat{\phi}} \tilde{o}}$$

$$= \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi} \in \Omega_{\tilde{A}}(\phi)} \sum_{\hat{\phi}' \in \Omega_a(\hat{\phi})} \left(\frac{\omega_{a_{\hat{\phi}'}}(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \delta_{a_{\hat{\phi}'} o} \right) \left(\prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \frac{\omega_{\tilde{a}_{\hat{\phi}'}}(\tilde{a}_\phi)}{\sum_{\hat{\phi}'' \in \Omega_{\tilde{a}}(\phi)} \omega_{\tilde{a}_{\hat{\phi}''}}(\tilde{a}_\phi)} \delta_{\tilde{a}_{\hat{\phi}'} \tilde{o}} \right)$$

The first term in parenthesis comes from the weighting of performing action a and receiving observation o, where we would like the only noisy copy of ϕ that agrees with that observation. The second term comes from that same operation, but for all actions and observations in ψ. Again, we know by construction that exactly one noisy copy agrees with all observations. Hence, we can write this as:

$$p(a_\Phi = o | \psi) \propto \sum_{\phi \in \Phi} p(\phi) \sum_{\hat{\phi} \in \Omega_{\tilde{A}}(\phi)} \sum_{\hat{\phi}' \in \Omega_a(\hat{\phi})} \left(\frac{\omega_{a_{\hat{\phi}'}}(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \delta_{a_{\hat{\phi}'} o} \right) \left(\prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \frac{\omega_{\tilde{a}_{\hat{\phi}'}}(\tilde{a}_\phi)}{\sum_{\hat{\phi}'' \in \Omega_{\tilde{a}}(\phi)} \omega_{\tilde{a}_{\hat{\phi}''}}(\tilde{a}_\phi)} \delta_{\tilde{a}_{\hat{\phi}'} \tilde{o}} \right)$$

$$= \sum_{\phi \in \Phi} p(\phi) \left(\frac{\omega_o(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \right) \left(\prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \frac{\omega_{\tilde{o}}(\tilde{a}_\phi)}{\sum_{\hat{\phi}'' \in \Omega_{\tilde{a}}(\phi)} \omega_{\tilde{a}_{\hat{\phi}''}}(\tilde{a}_\phi)} \right)$$

Finally, we would also like for $\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)$ to be constant for all actions a and realizations ϕ, enabling us to factor those terms out. To approximately achieve this, we generate noisy copies by discretizing the trajectory uniformly along the path, and generate a noisy copy of each realization ϕ at every discrete location. We approximate our realization to be set at one of the discrete locations, such that a_ϕ is equal to the nearest discrete location. For many locations, the weighting function will be less than some negligible ε. Let there be K discrete locations for any ϕ and a where $\omega_{a_\phi} > \varepsilon$. We say that $|\Omega_a(\phi)| = K$. Thus, we can fix the value of $\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi) = \kappa \quad \forall a, \phi$. Note that we also need to be consistent with observations corresponding to not contacting an object anywhere along the trajectory. Therefore, we also consider K noisy copies for this case. Under these assumptions, we can further simplify:

$$p(a_\Phi = o | \psi) \propto \sum_{\phi \in \Phi} p(\phi) \left(\frac{\omega_o(a_\phi)}{\sum_{\hat{\phi}'' \in \Omega_a(\phi)} \omega_{a_{\hat{\phi}''}}(a_\phi)} \right) \left(\prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \frac{\omega_{\tilde{o}}(\tilde{a}_\phi)}{\sum_{\hat{\phi}'' \in \Omega_{\tilde{a}}(\phi)} \omega_{\tilde{a}_{\hat{\phi}''}}(\tilde{a}_\phi)} \right)$$

$$\approx \sum_{\phi \in \Phi} p(\phi) \left(\frac{\omega_o(a_\phi)}{\kappa} \right) \left(\prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \frac{\omega_{\tilde{o}}(\tilde{a}_\phi)}{\kappa} \right)$$

$$\propto \sum_{\phi \in \Phi} p(\phi) \omega_o(a_\phi) \prod_{\{\tilde{a}, \tilde{o}\} \in \psi} \omega_{\tilde{o}}(\tilde{a}_\phi)$$

$$= \sum_{\phi \in \Phi} p_\psi(\phi) \omega_o(a_\phi)$$

$$= m_{\psi, a, o}$$

Finally, we need to normalize all observations to get:

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

Where $\mathbb{O}_a$ consists of all the discrete stopping points sampled, and the K observations for non-contact.

C. Proving the Bound

We showed that our utility function is equivalent to the mass removed from the original Φ: $\hat{f}(\psi) = 1 - M_\psi$. This function can utilize either of the two reweighting functions w^{HP} or w^{WHP} defined in Section IV-C. Our objective is a truncated version of this: $f(\psi) = \min\{Q, \hat{f}(\psi)\}$, 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 $f(\mathbb{A}, \hat{\phi}) = Q, \forall \hat{\phi} \in \widehat{\Phi}$. Note that adaptive monotone submodularity is preserved by truncation, so showing these properties for $\hat{f}$ implies them for f.

Using our utility function and observation probability, it is not hard to see that the expected marginal benefit of an action is given by:

$$\begin{aligned}
\Delta(a|\psi_A) &= \mathbb{E}\left[\hat{f}(A \cup \{a\}, \widehat{\Phi}) - \hat{f}(A, \widehat{\Phi}) \,\middle|\, \psi_A\right] \\
&= \sum_{o \in \mathbb{O}_a} p(o|\psi_A)\left[(1 - m_{\psi_A,a,o}) - (1 - M_{\psi_A})\right] \\
&= \sum_{o \in \mathbb{O}_a} \frac{m_{\psi_A,a,o}}{\sum_{o' \in \mathbb{O}_a} m_{\psi_A,a,o'}}\left[M_{\psi_A} - m_{\psi_A,a,o}\right]
\end{aligned} \quad (4)$$

This shows the derivation of the marginal utility, as defined in Section IV-C. We now provide the proof for Theorem 1, by showing that this utility function is adaptive submodular, strongly adaptive monotone, and self-certifying:

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:

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

As noted before, both of the weighting functions defined in Section IV-C 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:

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

We simplify notation a bit for the purposes of this proof. As the action is fixed, we will replace $\mathbb{O}_a$ with $\mathbb{O}$. For a fixed partial realization ψ_X and action a, let $m_{\psi_X,a,o} = m_o$. 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:

$$\sum_{o\in\mathbb{O}}\frac{m_o-k_o}{\sum_{o'\in\mathbb{O}}m_{o'}-k_{o'}}\left[M_{\psi_Y}-m_o+k_o\right]\le\sum_{o\in\mathbb{O}}\frac{m_o}{\sum_{o'\in\mathbb{O}}m_{o'}}\left[M_{\psi_X}-m_o\right]$$

$$\Leftrightarrow\left(\sum_{o\in\mathbb{O}}M_{\psi_Y}m_o-m_o^2+m_ok_o-M_{\psi_Y}k_o+m_ok_o-k_o^2\right)\left(\sum_{o'\in\mathbb{O}}m_{o'}\right)\le\left(\sum_{o\in\mathbb{O}}M_{\psi_X}m_o-m_o^2\right)\left(\sum_{o'\in\mathbb{O}}m_{o'}-k_{o'}\right)$$

$$\Leftrightarrow\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}M_{\psi_Y}m_om_{o'}-m_o^2m_{o'}+m_om_{o'}k_o-M_{\psi_Y}m_{o'}k_o+m_om_{o'}k_o-m_{o'}k_o^2\le\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}M_{\psi_X}m_om_{o'}-M_{\psi_X}m_ok_{o'}-m_o^2m_{o'}+m_o^2k_{o'}$$

$$\Leftrightarrow\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}M_{\psi_Y}(m_om_{o'}-m_{o'}k_o)+2m_om_{o'}k_o-m_{o'}k_o^2\le\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}M_{\psi_X}(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}$$

We also note that $M_{\psi_X}-M_{\psi_Y}\ge\max_{\hat{o}\in\mathbb{O}}(k_{\hat{o}})$. 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.

$$\Leftrightarrow\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}2m_om_{o'}k_o-m_{o'}k_o^2\le\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}(M_{\psi_X}-M_{\psi_Y})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}$$

$$\Leftarrow\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}2m_om_{o'}k_o-m_{o'}k_o^2\le\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}\max_{\hat{o}\in\mathbb{O}}(k_{\hat{o}})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}$$

$$\Leftarrow\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}2m_om_{o'}k_o-m_{o'}k_o^2\le\sum_{o\in\mathbb{O}}\sum_{o'\in\mathbb{O}}\max(k_o,k_{o'})(m_om_{o'}-m_ok_{o'})+m_o^2k_{o'}$$

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 $o\ne o'$, we assume that $k_o>k_{o'}$ WLOG, and show the inequality for the sum:

$$2m_om_{o'}(k_o+k_{o'})-m_{o'}k_o^2-m_ok_{o'}^2\le2m_om_{o'}k_o-m_ok_{o'}k_o-m_{o'}k_o^2+m_o^2k_{o'}+m_{o'}^2k_o$$

$$\Leftrightarrow2m_om_{o'}k_{o'}-m_ok_{o'}^2\le m_o^2k_{o'}+m_{o'}^2k_o-m_ok_ok_{o'}$$

$$\Leftrightarrow0\le k_{o'}(m_o-m_{o'})^2-(k_o-k_{o'})k_{o'}(m_o-m_{o'})+(k_o-k_{o'})m_{o'}(m_{o'}-k_{o'})$$

$$\Leftarrow0\le k_{o'}(m_o-m_{o'})^2-(k_o-k_{o'})k_{o'}(m_o-m_{o'})+(k_o-k_{o'})k_{o'}(m_{o'}-k_{o'})$$

We split into 3 cases:

D. $k_{o'}=0$

This holds trivially, since the RHS is zero

E. $k_{o'}\ne0,m_o\le2m_{o'}-k_{o'}$

Since $k_{o'}\ne0$, we can rewrite:

$$0\le(m_o-m_{o'})^2-(k_o-k_{o'})(m_o-m_{o'})+(k_o-k_{o'})(m_{o'}-k_{o'})$$

$$\Leftarrow0\le-(k_o-k_{o'})(m_o-m_{o'})+(k_o-k_{o'})(m_{o'}-k_{o'})$$

$$\Leftarrow(m_o-m_{o'})\le(m_{o'}-k_{o'})$$

Which follows from the assumption for this case.

F. $m_o\ge2m_{o'}-k_{o'}$

We show this step by induction. Let $m_o=2m_{o'}-k_{o'}+x,x\ge0$
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 $\widehat{m_o}$:

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

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

Finally, it is easy to see 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 [23] 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 [23].) That is the case here, since the objective function $f = \min\{Q, 1 - M_{\psi_A}\}$ 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 [23]. Following their notation, we let η be any value such that $f(\psi) > Q - \eta$ implies $f(\psi) \geq Q$ for all ψ. For Hypothesis Pruning, for example, we have $\eta = \min_\phi p(\phi)$. Additionally, the bound on the worst case cost includes $\delta = \min_{\hat{\phi}} p(\hat{\phi})\ \hat{\phi} \in \widehat{\Phi}$. The specific values of these constants are related to the weighting function and how discretization is done. Nonetheless, for either weighting function and any way we discretize, we can guarantee the greedy algorithm selects a near-optimal sequence.