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 of actions, 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 even with only one step lookahead. Our work first explains this high performance – we note that 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. Recently developed notions of *adaptive submodularity* enable guarantees for a greedy algorithm in the *online* setting. We develop new methods within this framework, enabling us to provide guarantees compared to the optimal *online policy*, as well as exploit additional computational speedups. We demonstrate the effectiveness of these methods in simulation and on a robot.**

I. INTRODUCTION

Dealing with uncertainty is a fundamental problem in robotics. Uncertainty accumulates from various sources such as noisy sensors, inaccurate models, and calibration error. This is particularly problematic for *fine manipulation tasks* [1], such as precise grasping, button pushing, or inserting a key into a keyhole. Due to the high accuracy required for these tasks, small errors often lead to catastrophic failure.

A standard approach is to perform a sequence of uncertainty reducing actions [2]–[6]. In fine manipulation, such actions are often encoded as *guarded moves* [7], where the hand moves along a path until it feels a touch.

The *optimal* sequence of actions provides enough information to accomplish the task while optimizing a performance criterion like minimum energy or time. Computing the optimal sequence can be formulated as a Partially Observable Markov Decision Process (POMDP) [8]. However, finding optimal solutions to POMDPs has been shown to be 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.

In this paper, we address the efficient automatic construction of such a touch-based localization sequence. Our primary insight is a connection to *submodularity*, allowing us to utilize efficient greedy algorithms and additional computational speedups. Furthermore, we provide guarantees that the sequence we select is near-optimal. Our experiments confirm that our methods provide accurate localization in an efficient
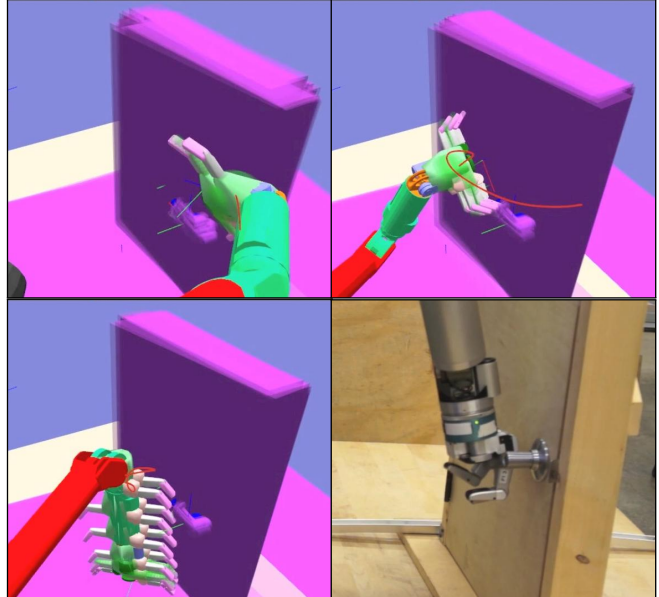

Fig. 1: We adaptively select a sequence of touch actions to reduce uncertainty. Here, we show actions selected by our Hypothesis Pruning method, enabling a successful grasp.

manner. See Fig. 1 for an example sequence which enabled a successful grasp of a noisy door handle.

Previous work on localization often utilizes online planning within the POMDP framework, looking at locally 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], [3], [5], [6]. This is generally out of necessity - computational time increases exponentially with the search depth. However, this simple greedy strategy often works surprisingly well for uncertainty reduction.

One class of problems known to perform well with a 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. Furthermore, no polynomial time algorithm can guarantee optimality (unless $P = NP$) [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], [3], [5], [6], [18], though they do not make the connection with submodularity. We note that Hsiao mentions that touch localization could be formulated as a submodular maximization problem [15]. One of our contributions is identifying the assumptions required for greedy action selection to be near-optimal, their consequences, and applicability to different problems in Section IV-A, as well as providing experimental results.

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], and thus we evaluate information gain online.

Recent notions of *adaptive submodularity* [23] extend the guarantees of submodularity to the adaptive setting. The set of requirements for adaptive submodular functions are different, and information gain does not meet those criteria. With information gain as our inspiration, we design a similar metric which does. In addition to providing guarantees with respect to that metric, we can use a lazy-greedy algorithm [23], [24] which does not reevaluate every action at each step, enabling a computational speedup.

In this work, we draw an explicit connection between touch based localization with a robotic end effector and submodularity. Understanding this connection enables us to design methods that fit into this framework and use a more efficient algorithm with near-optimal performance.

Along those lines, we present two approaches for uncertainty reducing action selection. The first approach optimizes the information gain by fitting a Gaussian distribution to the remaining particles, and evaluating the expected entropy that results 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 methods to selecting touch based sensing actions. We present results in Section V, comparing the accuracy and computation time of each metric. Finally, we show the applicability of these methods on a real robot.

II. Related work

Hsiao et al. [5], [15] formulate the problem of producing uncertainty reducing tactile actions with a POMDP. Since it is intractable to solve fully, they perform a forward search online. Potential actions are specified by a small set (typically ~5) of preset world-relative trajectories [5], making for a low branching factor. By searching at a limited horizon and performing aggressive pruning and clustering of observations, they circumvent computational issues. However, this can still take many seconds to compute each action and update, and there are no guarantees for optimality. In contrast, our work focuses on simpler actions and observations with a horizon of one, enabling us to consider significantly more actions (typically ~150) and achieve localization more quickly.

Hebert et al. [6] 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 [6], [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 object 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 identically 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 a more general observation space, allowing us to essentially model some amount of noise.

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 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, which terminate when the hand touches an obstacle. 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 $f : 2^\mathbb{A} \times O^\mathbb{A} \to \mathbb{R}_{\geq 0}$, which depends on actions selected and observations received. We would like to find a policy which that will reach some utility 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 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 $X \subseteq Y \subseteq \mathbb{A}$, $a \in \mathbb{A} \backslash Y$:

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 have a $(1 + \ln \max_a f(a))$ of optimal 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 $X \subseteq Y \subseteq \mathbb{A}$, $a \in \mathbb{A} \backslash Y$:

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 from actions $Y \backslash X$. 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

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.

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 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 [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.

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 amount 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 $a_\phi = \infty$. Note that this allows us to handle the observation corresponding to no contact.

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

In order to create objectives that fit into the framework of submodular maximization, we must make 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.
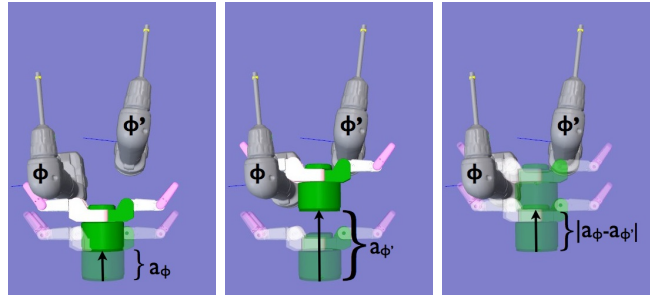

Fig. 3: The observations for action a and realizations ϕ and ϕ'. Each observation a_ϕ and a'_ϕ corresponds to the time along the straight line trajectory when contact first occurs with the object. We use the difference of times $|a_\phi - a_{\phi'}|$ when measuring how far apart observations are.

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 noise 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

Following Krause and Guestrin [21], we define the information gain as the reduction in Shannon entropy from performing actions. 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:

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

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 a 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:

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

If we were selecting with a discrete measure of entropy, locality of particles would not be taken into account. However, our particles actually represent samples from an underlying continuous distribution - we should prefer keeping two nearby particles as opposed to two faraway ones. Thus,

instead of evaluating $H(\Phi)$ directly, we instead fit a Gaussian distribution 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:

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

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 reduction.

C. Hypothesis Pruning

Intuitively, information gain is attempting to reduce uncertainty by shrinking the probability mass. 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. Golovin et al. describe the connection between this sort of query selection and adaptive submodularity by drawing a connection to Set Cover [23]. Our formulation is similar - see Fig. 2 for a visualization.

As before, we consider a blurred measurement model. 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, and thus remove a portion of the hypothesis. We call this metric Weighted Hypothesis Pruning (WHP). The weighting functions are:

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

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

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. Then our utility is:

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

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[1]. 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:

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

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 maximize the expected probability mass removed at each step, per unit cost. After selecting an action and receiving an observation, the hypotheses are downweighted or removed as described above, and action selection is iterated. We now present the main guarantee for this method:

Theorem 1: Let our utility function for Hypotheses Pruning be f as defined above, utilizing either weighting function w^{HP} or w^{WHP}. Define $\delta = \min_\phi p(\phi)$. 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, and $\left(\ln\left(\frac{Q}{\delta^2}\right) + 1\right)$ times the worst case cost of the best policy. More formally:

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

$$c_{wc}(\pi^{greedy}) \leq c_{wc}(\pi_{wc}^*)\left(\ln\left(\frac{Q}{\delta^2}\right) + 1\right)$$

Proof: In order to prove Theorem 1, we will need to show that our objective is adaptive submodular, strongly adaptive monotone, and self-certifying. We show this in the Appendix[1]. Our proof then follows directly from [23]. ∎

In addition to being a logarithmic factor of optimal, we can utilize a lazy-greedy algorithm which does not reevaluate all actions at every step, enabling a computational speedup [23], [24].

V. EXPERIMENTS

We implement a greedy action selection scheme with each of the methods 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

---

[1]Located at `http://www.cs.cmu.edu/~sjavdani/touch_loc_submodular.html`

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 predefined contact point and normal with the one randomly sampled from the object, randomly rotate the hand about the contact normal, and 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 $|\mathbb{A}| = 203$.

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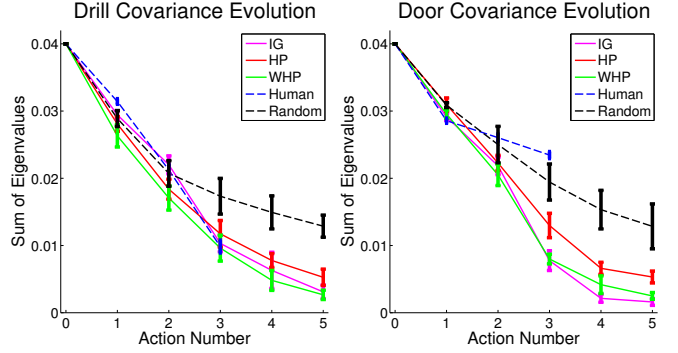

Fig. 4: Uncertainty after each action for drill and door experiments. The bars show the mean and 95% CI of the sum of eigenvalues of the covariance matrix over 10 experiments.

	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 in Section V-B

C. Results

We analyze the uncertainty reduction of each metric as the sum of eigenvalues of the covariance matrix, as in Fig. 4. All of the 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.

We note that this measure of uncertainty is good for unimodal distributions, as it assumes a single covariance matrix captures the uncertainty. Our Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP) method actually make no attempt to keep a unimodal distribution, as they naively prune hypotheses. On the other hand, our Information Gain (IG) method optimizes this measure directly, as it evaluates entropy by fitting a Gaussian. Surprisingly, even for this measure of uncertainty reduction, our HP and WHP methods have comparable performance with IG. Additionally, we find that they perform significantly faster, due to both their inherent simplicity, and a speedup from a lazy-greedy algorithm [23], [24]. See Table I.

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. Interestingly, each of our metrics selected a different sequence of actions, though they obtain similar performance.

1) 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
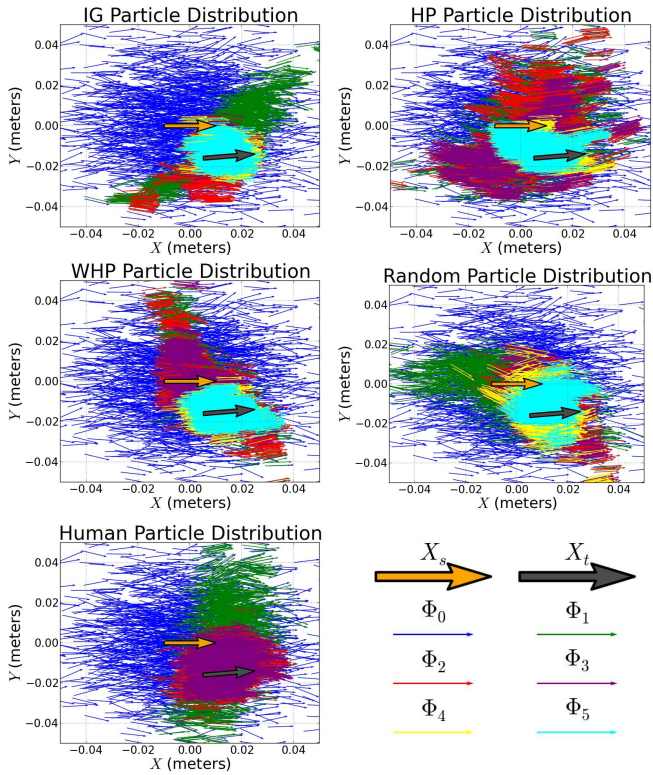
Fig. 5: The particle sets Φ from a single drill experiment after each update. Each plotted position corresponds to the x, y parameter of $\phi \in \Phi$, rotated by θ. X_s is the sensed position, X_t the true position, and Φ_i the particles after update i. Arrow lengths are approximately the length of the drill base. The initial distribution was generated from a normal distribution with $\sigma_x = 0.02$, $\sigma_y = 0.02$, $\sigma_z = 0.02$, $\sigma_\theta = 0.2$.

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 utilize a motion planner to transition to its start pose, and perform the straight line motion 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 III, and full videos are provided online[1].

VI. CONCLUSION AND DISCUSSION

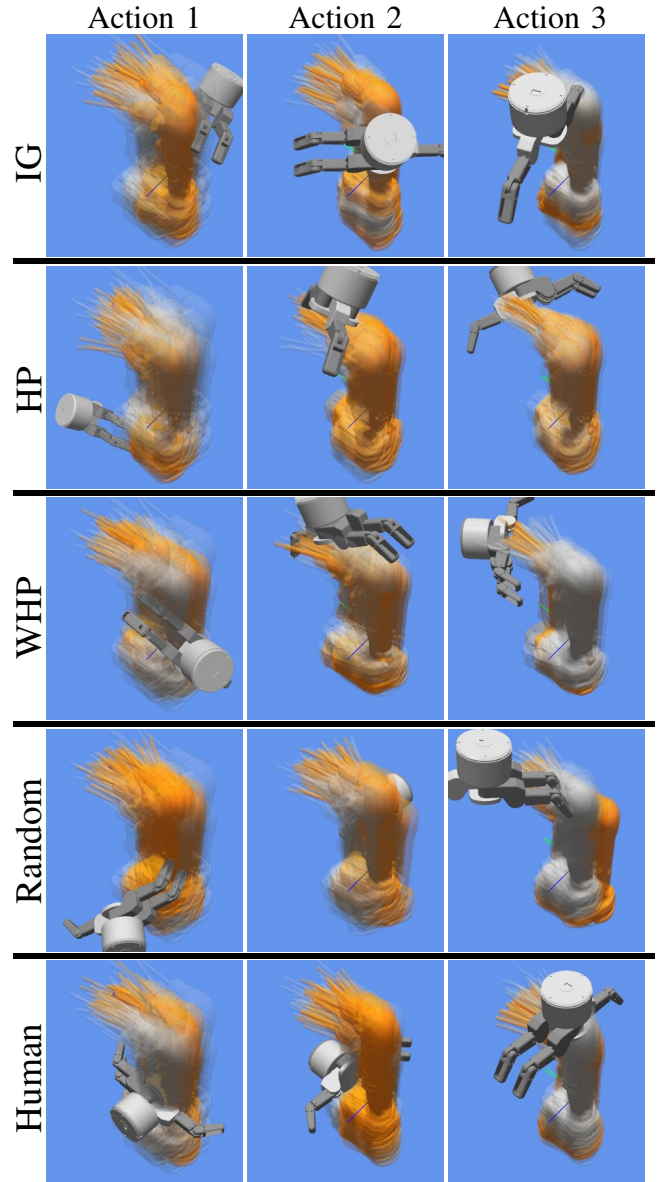In this work, we drew an explicit connection between submodularity and touch based localization. We presented

TABLE II: First 3 actions selected for each metric from the experiment in Fig. 5. The updates particles Φ are shown in yellow, with the previous in grey.

three greedy methods of selecting uncertainty reducing touch actions. The first, Information Gain (IG), has been used extensively for robot localization [2]–[6], [18], [20]. We noted the assumptions necessary for this method to be submodular, rendering the greedy algorithm near-optimal in the *offline* setting. We design our own methods, Hypothesis Pruning (HP) and Weighted Hypothesis Pruning (WHP), which we show are adaptive submodular. Thus, an efficient greedy algorithm 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]. We demonstrate good performance for all our methods, both in simulation and on a robot.

One limitation of our current work is the assumption that
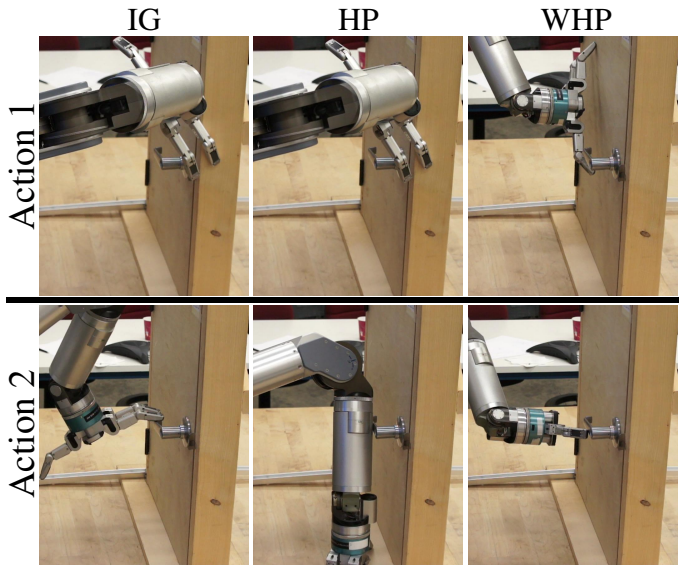
IG	HP	WHP

TABLE III: Actions selected during robot experiment. Note that both IG and HP selected the same first action. All metrics lead to a successful grasp of the door handle.

the hand and object are completely rigid, and contact is sensed with any force. Some of the actions selected may not be robust in the physical world. We hope to incorporate better action generation, as well as a more expressive hand and sensor model within our metrics to alleviate this.

Though our hypothesis pruning methods satisfy conditions of adaptive submodularity, we note that Information Gain performs comparably well. One limitation of our current hypothesis pruning formulations is that they naively remove hypotheses, with no notion of the underlying continuous distribution. Furthermore, they simply reduce uncertainty until it falls below some threshold. In actuality, we may wish to drive our uncertainty to a particular distribution, dependent on the desired task. We hope to extend the ideas developed here to formulations which do.

Acknowledgments

References

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

[2] A. R. Cassandra, L. P. Kaelbling, and J. A. Kurien, "Acting under uncertainty: Discrete bayesian models for mobile-robot navigation," in *IROS*, 1996, pp. 963–972.

[3] W. Burgard, D. Fox, and S. Thrun, "Active mobile robot localization," in *In Proceedings of IJCAI-97. IJCAI, Inc.* Morgan Kaufmann, 1997.

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

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

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

[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, Aug. 1987.

[10] N. Roy, G. Gordon, and S. Thrun, "Finding approximate pomdp solutions through belief compression," *JAIR*, vol. 23, p. 2005, 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, June 2012.

[14] S. Ross, J. Pineau, S. Paquet, and B. Chaib-draa, "Online planning algorithms for pomdps," *J. Artif. Int. Res.*, vol. 32, no. 1, July 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," *J. ACM*, vol. 45, no. 4, pp. 634–652, July 1998.

[17] L. A. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, 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 *IROS*, 2002, pp. 540–545.

[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 *ICRA*, 2008.

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

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

[23] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *JAIR*, vol. 42, pp. 427–486, 2011.

[24] M. Minoux, "Accelerated greedy algorithms for maximizing submodular set functions," in *Optimization Techniques*, ser. Lec Notes in Control and Information Sciences, J. Stoer, Ed., 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, pp. 160–167.

[26] R. Platt, L. Kaelbling, T. Lozano-Perez, 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, pp. 569–585, June 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 *IROS*, October 2010.

[31] R. Nowak, "Generalized binary search," in *Proceedings of the 46th Allerton Conference on Communications, Control, and Computing*, 2008, pp. 568–574.

[32] ——, "Noisy generalized binary search," in *NIPS*, 2009, pp. 1366–1374.

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

[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 *ICRA*, vol. 113, 2009.

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

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-C. Our objective is a truncated version of this: $f(A, \phi) = \min\{Q, \hat{f}(A, \phi)\}$, 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 $p(a_\Phi = o|\psi) = \frac{m_{\psi,a,o}}{\sum_{o' \in O} m_{\psi,a,o'}}$:

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

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 $\hat{p}(a_\Phi = o|\psi)$:

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

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:

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

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:

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

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:

$$\Delta(a|\psi_Y) \leq \Delta(a|\psi_X)$$

$$\sum_{o \in O} \frac{m_{\psi_Y,a,o}}{\sum_{o' \in O} m_{\psi_Y,a,o'}} \left[M_{\psi_Y} - m_{\psi_Y,a,o} \right] \leq \sum_{o \in O} \frac{m_{\psi_X,a,o}}{\sum_{o' \in O} m_{\psi_X,a,o'}} \left[M_{\psi_X} - m_{\psi_X,a,o} \right]$$

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:

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

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

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

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

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.

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

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

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

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

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

$$\Leftrightarrow 2 m_o m_{o'} k_{o'} - m_o k_{o'}^2 \leq m_o^2 k_{o'} + m_{o'}^2 k_o - m_o k_o k_{o'}$$

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

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

We split into 3 cases:

A. $k_{o'} = 0$

This holds trivially, since the RHS is zero

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

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

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

$$\Leftarrow 0 \leq -(k_o - k_{o'})(m_o - m_{o'}) + (k_o - k_{o'})(m_{o'} - k_{o'})$$

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

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 $\widehat{m_o}$:

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

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

Finally, it is easy to 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 [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 note that $\eta = \min_\phi p(\phi)$, since it is always true that $f(S, \phi) > Q - \min_\phi p(\phi)$ implies $f(S, \phi) = Q$.