

# Active Learning from Demonstration for Robust Autonomous Navigation

David Silver, J. Andrew Bagnell and Anthony Stentz  
Carnegie Mellon University

**Abstract**—Building robust and reliable autonomous navigation systems that generalize across environments and operating scenarios remains a core challenge in robotics. Machine learning has proven a significant aid in this task; in recent years learning from demonstration has become especially popular, leading to improved systems while requiring less expert tuning and interaction. However, these approaches still place a burden on the expert, specifically to choose the best demonstrations to provide. This work proposes two approaches for active learning from demonstration, in which the learning system requests specific demonstrations from the expert. The approaches identify examples for which expert demonstration is predicted to provide useful information on concepts which are either novel or uncertain to the current system. Experimental results demonstrate both improved generalization performance and reduced expert interaction when using these approaches.

## I. INTRODUCTION

Building dependable autonomous navigation systems remains a difficult challenge to the robotics community. One of the major obstacles to developing truly robust and reliable systems is the necessity to determine proper values of the numerous parameters and preferences that reside in every system. Manually tuning these settings is tedious and time consuming, and does not necessarily produce high performing systems. As a result supervised learning based approaches have gained acceptance in recent years for automating the task of selecting system parameters. A recent popular paradigm is the *Learning from Demonstration* approach, where parameters and preferences are automatically tuned to reproduce expert demonstrated behavior. This approach has proven to both improve the performance of autonomous navigation systems while reducing the time to their deployment [1], [2], [3], [4].

In general, such approaches depend on the expert to provide not only the actual demonstrations used to teach the system, but also to choose exactly what to demonstrate. This puts additional burden on the expert, and raises the bar for who can qualify to train the system; the expert must now have sufficient knowledge of the actual system internals to know which examples are most useful. In addition, the expert must understand the structure of the example to fully comprehend how useful or not it is. Having the expert choose what to demonstrate can also affect the quality of the final training set. It has been observed empirically that a good deal of expert demonstration proves to be redundant (see Section V) simply repeating the same basic concepts. While this does help to potentially fine tune such concepts, it can leave important concepts undemonstrated. If the amount of expert interaction is held constant, such inefficient use of time will

lower the overall effectiveness of the training set.

Many of these issues are common to all forms of supervised learning. As a result, the field of *Active Learning*, where the learner is at least partially responsible for choosing which training data to obtain, has received focus in order to aid in the efficient selection of a good training set. Learning from demonstration could potentially be aided by the incorporation of active learning techniques that requested demonstration in specific scenarios (e.g. how a robot should drive in a specific type of terrain). Such techniques could reduce the cognitive burden on an expert, as well as increasing the pool of potential experts. In addition, active learning could help to ensure that each additional demonstration was maximally useful and not redundant.

This work investigates the application of active learning to learning from demonstration, specifically in the context of autonomous navigation. The next section describes related work in both fields. Sections III and IV then describe two basic approaches to active learning in the context of autonomous navigation. Experimental results are presented in Section V with conclusions and in Section VI.

## II. RELATED WORK

Autonomous navigation is the task of a robot traversing from some start location to a goal location. Along the way, it must maintain knowledge of its own position, observe the surrounding environment, plan actions to achieve its goal, and finally execute these actions. Some of these subtasks, most notably perception and planning, are highly parameterized in that they make use of numerous heuristics, thresholds, and other settings that determine the actual behavior of the robot (e.g. the determination of traversability in a perception system, or specification of navigation rules for a planning system). Achieving truly autonomous behavior from a mobile robot therefore requires not just robust and accurate perception and planning algorithms, but also parameter settings that will produce the desired behavior. Towards this end, machine learning approaches are often applied, to both ease the burden on system engineers, as well as ensure a good set of parameters are chosen. Learning from demonstration has gained recent popularity as the expert generated input is exactly the desired behavior, as opposed to a surrogate (e.g. performing supervised classification to identify dangerous terrain).

Learning from demonstration can be applied to robot control and navigation in one of two basic ways. Using a model free approach, expert demonstration can be used as the training input to learn a specific policy for the robot to

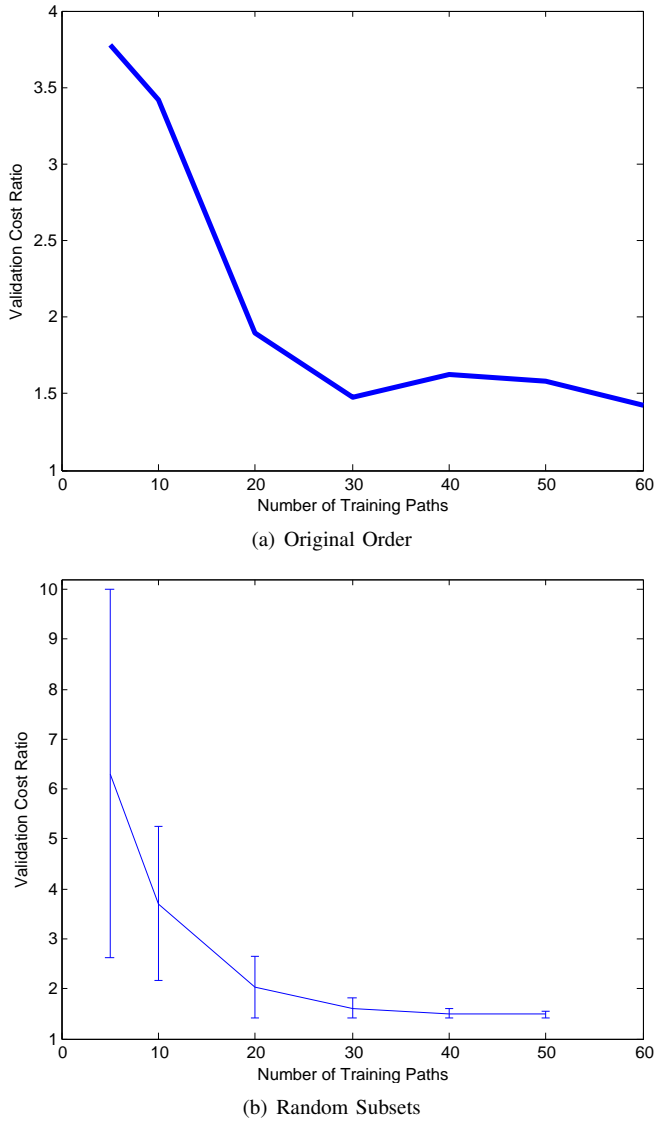


Fig. 1. Validation performance for increasing numbers of example paths, in (a) their original order and (b) random permutations (error bars indicate one standard deviation). Over time, there are clear diminishing returns to adding additional (expert chosen) examples.

follow (that is, a direct mapping from robot percepts and state to control outputs). For certain problems, this approach is highly effective. However, this approach requires all information relevant to decision making to be encoded in the current robot state. As a result, this approach performs poorly when long range decision making is required, or generalizing to previously unseen states. An alternate approach to learning from demonstration is model based. This approach uses expert demonstration to learn a cost or reward function over robot percepts and state, which can then be applied in novel scenarios to find the optimal policy for that scenario.

This latter technique mirrors one of the most popular approaches to robot navigation: assign various decisions (actions and or locations) with some sort of cost, and then use an optimal motion or path planner to identify the next sequence of actions. In order to allow for generalization, the cost func-

tion usually takes as input a set of features that describe the current state or have been extracted from the current percepts, rather than these raw inputs themselves. In this approach, the cost function essentially encodes the desired behavior of the robot. Therefore the proper parameterization and tuning of the cost function is of extreme importance to correct and reliable robot behavior. If the cost function is learned from demonstration as described above, it can both reduce the amount of human parameter tuning necessary to proper system function, as well as improving the performance and generalization of the cost function. This generalization occurs for two primary reasons. First, machine learning best practices can be applied (e.g. to avoid overfitting) which provide certain theoretical guarantees. Second, learning from demonstration provides a very rich and varied training input as it not only specifies what a robot *should* do, but it also implies what a robot *should not* do. As a result, learning cost or reward functions from demonstration has become popular in recent years [1], [2], [3], [4].

While these approaches have changed the responsibility of a system expert from that of tedious parameter tuning to demonstrating desired behavior, there remain barriers to efficient deployment. In this paradigm, the demonstrator is responsible for choosing exactly what to demonstrate. This allows for the possibility that potentially useful or important concepts could never be taught, or that the provided examples are redundant. Redundancy in particular is a serious problem, as it has been empirically observed that increasing amounts of demonstration produce rapidly diminishing returns (see Figure 1). It also raises the bar for who can demonstrate behaviors to the robot: the demonstrator must have at least a basic intuition of the underlying perception and planning algorithms in order to decide what exactly to demonstrate. Finally, for larger mobile systems actually demonstrating the correct behavior can be expensive, especially if large training sets are required to properly capture complex desired behavior. In such scenarios it may only be possible to provide a limited set of demonstrations, which ideally should be chosen to be as efficient as possible.

The field of Active Learning, where the learner is involved in the selection of training data, has developed in response to supervised learning in general facing this exact problem. Active learning with respect to certain forms of learning from demonstration has received some previous attention. The approaches of [5], [6] each allow a robot to request instruction whenever it is not sufficiently confident in a learned action it is about to execute, based on its similarity to previously observed actions. However, these approaches are primarily concerned with model free learning, as opposed to more model based and generalizable learning from demonstration. In contrast [7] investigates active learning in a model based context; however, it assumes that there is an a priori reward function, and demonstration is used to help discover the optimal policy.

A common thread of these approaches is their use of query filtering: the learner encounters tasks one at a time, and must decide whether to request expert aid for each task. This requires the expert demonstrator to be 'on call'

to provide demonstration during robot operation. This mode of operation could potentially be quite useful for autonomous navigation once an initial training set has been produced (and the robot was mostly functional). However, it would prove quite time consuming for the expert if used exclusively to train a system from scratch. Alternatively, pool based approaches (that request expert instruction over a small subset of a large, unlabeled data set) would minimize the initial expert interaction necessary to train a robust system. Therefore, the techniques presented in the remainder of this work are primarily pool based approaches; however they could also be applied for query filtering in an online setting when an expert is available.

### III. ACTIVE LEARNING THROUGH NOVELTY REDUCTION

In an ideal setting, a set of training demonstrations provided by an expert would come from the same distribution of examples as that which the robot would encounter while operating autonomously. This is quite unrealistic, especially since it is rarely the case that the training and operating environments are identical. A more realistic desire would be that the two distributions at least cover the same areas of the example space: that is, that for every possible scenario the robot could find itself in, it has been trained on something sufficiently similar such that it can properly generalize. Unfortunately, this is also difficult to ensure, as the example space is hard to quantify. Instead of the example space, the feature space (i.e. the input space to the cost function) can be used as a proxy. Therefore, a potential goal of active learning could be to ensure that for any type of decision the robot may encounter, it has at least seen something similar in the training set.

In order to understand what the robot has not been exposed to, it is necessary to model what the robot has been exposed to. This problem can be phrased as a form of density estimation over the feature space of states and actions. Then, examples can be chosen from regions with low density. Such regions are often referred to as anomalous or novel, in that feature vectors that have not been frequently seen during training represent something novel to the learner. Novelty detection in and of itself is a rich field of research [8], and has seen many applications to mobile robotics [9], [10]. [11], [12] specifically propose density estimation in the context of active learning for robotic perception systems. Specifically, a large dataset consisting of a robot's recorded perceptual history is analyzed to identify the most unlikely single percepts (given the entire history). These percepts are then provided to human experts to be labeled for use in training a terrain classification system. This process is then repeated to produce a full training set that properly covers the entire space of percepts.

This same general approach could also be applied to learning from demonstration. In this context, all terrain or action features observed during previous training would be used to build a density model. Then, the entire (unlabeled) set of features observed at any time would be analyzed to detect novel cases. Expert demonstration could then be requested

for novel structures or scenarios. Such a novelty based active learning procedure would proceed in 3 phases

- 1) **Initial Training:** Perform learning from demonstration on an existing training set, and build a data set of every feature vector encountered during learning.
- 2) **Density Estimation:** From the data set, build a density model, such that for any test feature vector, a measure of how similar it is to already seen data can be produced.
- 3) **Example Selection:** Once a density model has been computed, example problems are selected such that the resulting demonstration is likely to provide useful information about novel areas.

This process would then be iterated to continually add the most novel examples.

An additional step that could be inserted to this process is that of dimensionality reduction before density estimation. Performing density estimation in a high dimensional space is computationally expensive, as well as prone to potential overfitting. Dimensionality reduction can alleviate both of these problems, and is quite common as a precursor to density estimation. Dimensionality reduction is itself another vast area of research [13], with Principal Components Analysis (PCA) being perhaps the single most common approach.

As for performing density estimation in the (reduced) feature space, there are two main classes of approaches: parametric and non-parametric. While [11] uses non-parametric density estimation, parametric approaches are generally much faster at test time, allowing for easier application in an online setting (to allow for query filtering as well as a pool approach). Therefore, parametric density estimation is preferable in this context. There are many different specific algorithms that could be applied for both dimensionality reduction and density estimation, each with their own advantages and tradeoffs. In the end, any approach could be applied that results in the ability to produce some sort of novelty measure for various test points.

Example selection is the most unique part of this proposed procedure. In the context of autonomous navigation, this would consist of producing start and goal points such that the resulting example plan is likely to provide new information about novel terrain or maneuvers. The challenge is then to try and anticipate the demonstration that is likely to result from a specific test problem. The key insight to solving this problem is that there already exists a seed set of expert demonstration, and a cost function learned from the set that seeks to reproduce the example behavior. Since the cost function has been learned to try and imitate the expert, it can also be used to try and predict their future behavior [14]. Therefore, between any two test waypoints the planned trajectory under the current cost function is a reasonable prediction for the expert's future demonstration. Therefore, the aggregate novelty along the current planned path can be used as a measure of the overall novelty of the example problem. Regardless of the expert's actual demonstration, something will be learned about this novel example: if the expert does produce the predicted example, then an example of preferring previously novel terrain or actions has been

added to the training set, and if the expert produces a different behavior then an example of *not* preferring novel terrain has been added.

Therefore, the following example selection heuristic is proposed. First, select a large set of random waypoint pairs, and plan behaviors under the current cost hypothesis. Next, evaluate the aggregate novelty along each plan. Finally, choose a random subset of the waypoint pairs to present to the expert, with the likelihood of a pair's selection related to its aggregate novelty. As the resulting demonstrations are likely to provide information about previously novel regions of the feature space, they will help to ensure at least minimum coverage of said space. An additional option for this heuristic is to evaluate aggregate novelty under two possible plans: that under the current cost function, and under the initial cost hypothesis (usually simply a constant cost over all states and actions). High cost regions of the feature space are often novel simply because an expert never demonstrates behavior that contains them; also considering the initial cost hypothesis can produce waypoint pairs that continually force the expert to demonstrate their aversion to such areas, thus lowering their novelty.

In addition to providing a useful metric for aiding in offline active learning, a novelty function learned through the above procedure would also be useful in online active learning. Such a novelty function would allow for the identification of terrain features or maneuvers on which the robot has not been sufficiently trained. As demonstrated in [9], [10], such novelty functions can be useful in the autonomous operation of a mobile robot by preventing it from encountering scenarios for which it is unprepared. Additionally, it could also be used in a query filtering approach if an expert is available. That is, if the robot intended to drive through a low cost but novel section of terrain, it could first request expert 'permission' to do so before proceeding (and then learn from the resulting expert response).

#### IV. ACTIVE LEARNING THROUGH UNCERTAINTY REDUCTION

As opposed to specifically seeking expert feedback on novel examples, another approach is to request expert information on examples for which the learner already has information, but remains uncertain. There is of course a relationship between novelty and uncertainty, in that a learner is likely (but not guaranteed) to be uncertain about truly novel examples. However, novelty is not the only source of uncertainty; it can also come from similar examples with conflicting labels or demonstrated behavior.

When a learner explicitly models its own uncertainty, an active learning technique known as *Uncertainty Sampling* [15] can be applied. However, explicit uncertainty sampling in the context of learning cost or reward functions would limit the form of the functions to Gaussian Processes or Bayesian linear regression. Gaussian Processes are not a good choice for learning a cost function when real time cost production is required, as they are quite expensive to apply when trained on large datasets. Bayesian linear regression is parametric; however this in turn limits the hypothesis space

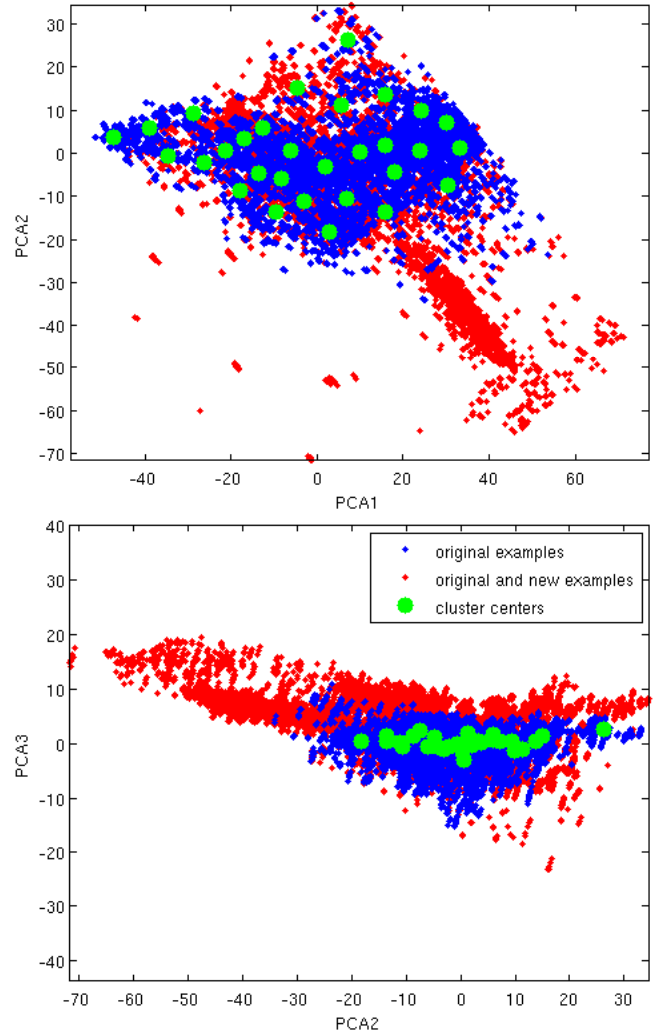


Fig. 2. Density estimation for novelty detection in Figure 4. The full feature space is first reduced to 3 dimensions through PCA. Next, K-Means is used to determine cluster centers to approximate the density of the original distribution. Active learning results in new examples that reside in previously uncovered regions of the space.

of cost functions. Therefore, explicit uncertainty sampling is not a desirable option in this context.

However, there is another approach that can work with general cost functions. The *Query by Bagging* approach [12] combines the idea of *Query by Committee* [16], [17] with the idea of *Bagging* [18] to measure the uncertainty still inherent in a training set for a particular class of learner. The idea is to train multiple learners on different random subsets (with replacement) of the available training set, and see where they disagree. In the context of cost functions, the analog would be to learn multiple functions from different subsets of the available demonstration. Then, the uncertainty of a particular example would be approximated as the variance of the different cost functions over said example.

Therefore, an uncertainty based active learning approach could be implemented that is nearly identical to the novelty based approach. First, an uncertainty model would be constructed based on existing expert demonstration. This model

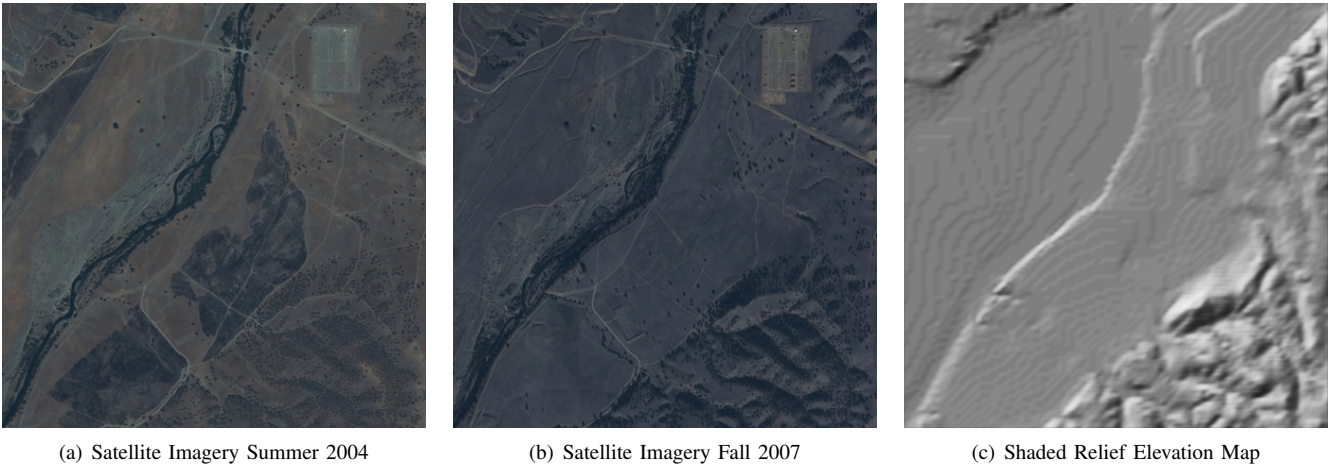


Fig. 3. The 9 km<sup>2</sup> test site used for these experiments. Quickbird imagery courtesy of Digital Globe, Inc.

would then be used to evaluate plans (under both the current hypothesis and the initial hypothesis) between random waypoints. Waypoint pairs where the demonstration is predicted to traverse high uncertainty areas would then be presented to the expert as a request for explicit demonstration, and the process repeated. Also as before, a final uncertainty model could be also used in an online setting for query filtering. As explored in [19] there are yet more additional uses for knowledge of uncertainty in a cost function.

Alternatively, another approach to example selection is possible. Unlike with novelty based active learning, the uncertainty model provides a metric in the same units as cost. Therefore, the uncertainty model can be used to estimate upper and lower bounds on the 'true' cost for any test point (the actual observed bounds could also be used). Rather than choosing waypoints that are likely to involve demonstration through regions of high uncertainty, waypoints could be chosen such that the predicted plan under the upper bound of the cost hypothesis is significantly different than the predicted plan under the lower bound hypothesis. Practically speaking, this method of example selection would be more likely to choose examples such that the cost uncertainty actually affects the current plan.

## V. EXPERIMENTAL RESULTS

In order to evaluate the effectiveness of both novelty and uncertainty based active learning, experiments were performed in the context of learning from demonstration for interpreting overhead data of an environment [1], [20]. In this context, an expert draws paths over satellite imagery or digital elevation maps that indicate how the expert would prefer a robot to navigate said terrain. A cost function (mapping features of the raw data such as color or texture to cost) is then learned over the entire environment that both reproduces the demonstrated behavior, as well as generalizing the concepts to untrained areas. As opposed to experiments performed on a live autonomous system, the primary advantage of experiments in this context is the ease of repeated experiments in a deterministic environment.

All experiments were performed on overhead data for a large (9 km<sup>2</sup>) test site. The test site consists of several distinct terrain features such as short and tall grass, bushes, sand, water, different types of trees, and rolling hills. There are also man made features such as roads, buildings and areas used for controlled burns. The overhead data consists of two sets of satellite imagery and a digital elevation map (Figure 3). The metric used for performance evaluation is the *Cost Ratio* [20], which indicates how close the current cost function is to considering all expert demonstration optimal. Candidate cost functions are evaluated using the resulting cost ratio on a large validation set of 250 example paths.

Novelty based active learning was implemented as described in Section III. First, given an existing training set a cost function is learned, and every feature vector encountered during training is recorded. The original feature space consisted of 30 dimensions, an unwieldy number for performing efficient density estimation. Therefore, PCA was used to reduce the data set to 3 dimensions. Once in this space, the K-Means algorithm [21] was used to identify clusters of points and their associated cluster centers. The number of cluster centers K-Means is tasked with finding was set very high (30-50 in these experiments) as the goal is not to actually identify distinct clusters, but simply regions of the feature space with high point density. For a test point in the reduced space, the novelty is then defined as the squared Euclidean distance to the nearest cluster center. Although a very simplistic density estimation and novelty detection approach, it proves more than sufficient for identifying regions of the feature space that have received little to no coverage. Extension to more powerful techniques (such as using the cluster centers to seed Gaussian Mixture Models) is straightforward.

Figures 2 and 4 show a single example iteration of this process. Figure 4(a) shows 5 original example paths. These paths only traverse or avoid roads, tan-colored grass, and trees. As a result, the costmap learned from this training set (Figure 4(b)) must try to generalize to other terrain features. However, it does a very poor job of generalizing to where there was a recent controlled burn of the grass (the dark grey



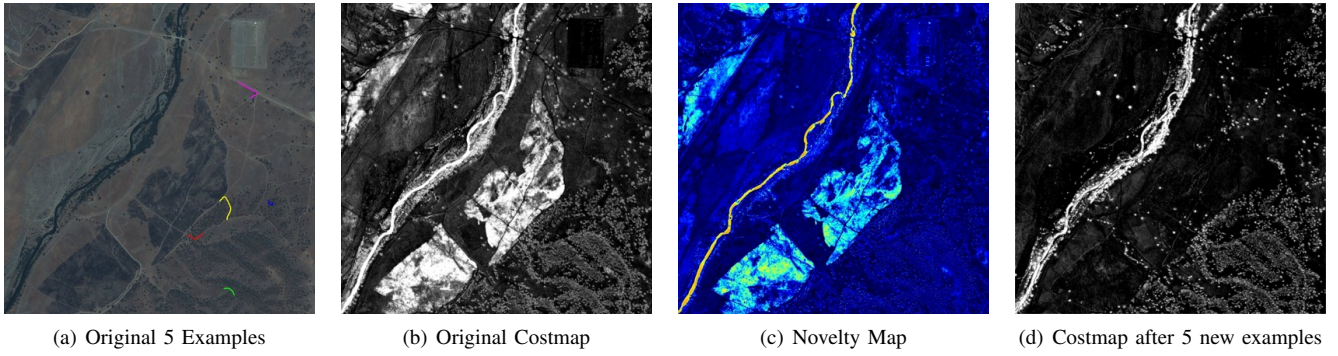


Fig. 4. An example of density based novelty detection and active learning on overhead data.

regions). These areas are given an extremely high cost, when in fact they are very preferable terrain to traverse through (due to the almost complete lack of vegetation).

After performing density estimation, the novelty map (the novelty function applied to the features of the entire test site) is shown in Figure 4(c). As would be expected given the initial training set roads, trees, and grass are low novelty while the burned areas, the river, and buildings are high novelty. 5 new examples were then requested, by picking random waypoints and evaluating the novelty of the current plan. This results in examples that traverse the burned areas, helping to demonstrate their actual preferability. The cost function learned as a result of the new training set (5 original examples and 5 new examples) is shown in Figure 4(d), with a more reasonable cost estimate over novel regions.

Figure 2 shows the distribution of observed feature vectors (in the PCA reduced space), over both the original and combined training set. The original training set is confined to one region of the feature space (due both to the homogeneous nature of the original set, as well as the reduced space being computed according to its distribution). The cluster centers identified via K-Means are able to represent the distribution of this set with orders of magnitude fewer points. In contrast, the examples selected during active learning result in features that occupy completely different regions of the space. This demonstrates the ability of novelty based active learning to ensure that new regions of the feature space (corresponding to the variety of observed terrain features) are covered via example demonstration.

Uncertainty based active learning was also implemented using Query by Bagging to produce an uncertainty estimate in the cost function. Figure 5 provides an example of this approach in action. A training set of 20 example paths were used to train the cost function shown in Figure 5(b). This cost function underestimates the cost on several types of undesirable terrain, such as slopes, trees, and water. However, since the original training set did include some examples of these terrain features, they are not considered particularly novel (Figure 5(a)). However, Query by Bagging is able to identify these areas as regions of higher cost uncertainty ((Figure 5(c)). That is, when training on some subsets of the full training set, these regions are given a high cost, and on others a lower cost. In contrast, regions that are always given a similar cost (such as roads) are given a very low uncertainty. 10 additional examples chosen via uncertainty

based active learning provide additional demonstration of the true preferences over the high uncertainty regions; the resulting cost function (Figure 5(d)) reflects this new information via increased cost on trees, water, and slopes. This not only demonstrates the effectiveness of uncertainty based active learning, but also the way in which it can complement novelty based active learning.

The novelty and uncertainty functions in Figures 4 and 5 also demonstrate that, aside from just being useful for offline (pool-based) active learning, there is potential for online query filtering and vehicle safeguarding. What is especially useful about these functions is that they have been learned with respect to the underlying cost function. That is, a perception system may be highly confident in its raw sensor data and generation of descriptive features, while the cost function is still uncertain about the desirability of the combination of such features. This makes such functions an ideal complement to novelty or uncertainty functions learned for a perception system, and could prove quite useful in the application of such techniques [10].

In addition to proof of concept experiments, larger active learning experiments were also performed to produce quantitative results. The basis for comparison is a set of 60 expert produced training paths. The performance of this training set is shown in Figure 1, both in its original order, and with repeated random permutations. In order to compare active learning based example selection to expert based example selection, 60 example paths were chosen using multiple iterations of novelty based active learning (in order to seed the novelty function when the training set is empty, a random subset of the entire training area is used). An additional 40 example paths were chosen using multiple iterations of uncertainty based active learning, with the first 20 novelty chosen paths as a seed (uncertainty based active learning requires an initial training set). The result is two (partially overlapping) sets of training examples that can be compared to the expert chosen set. Figure 6(a) shows the validation performance of these three training sets when applied in their original order. The first fact of note is that for small training sets, novelty based active learning significantly outperforms expert chosen examples, demonstrating the ability of active learning to identify important novel examples. As more examples are added to each set, the final performance of all 3 sets converges, although the active learning approaches still outperform the expert chosen examples.

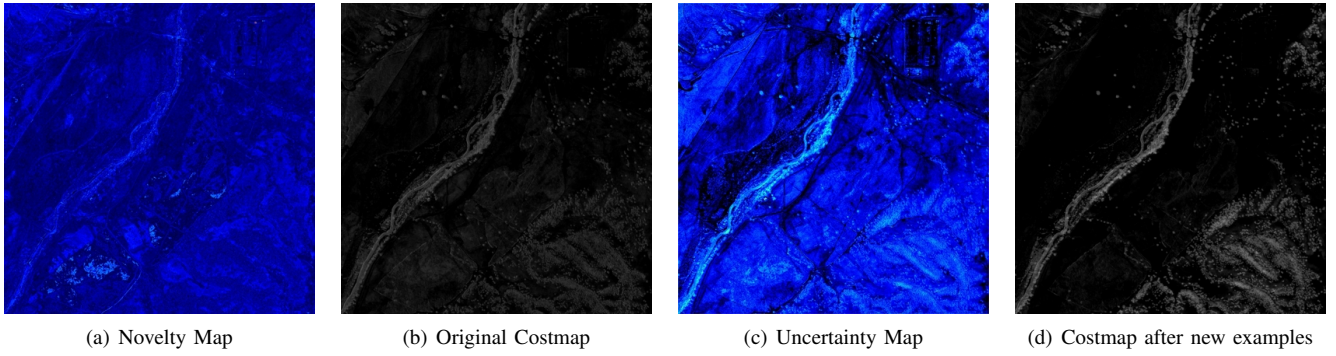


Fig. 5. An example of uncertainty based active learning, and how it can complement novelty based active learning.

Figures 6(b) and 6(c) show the performance of all 3 training sets when learning with multiple subsets in random order. Both active learning training sets consistently perform better than the expert chosen set (significant at the 1% level). Also of note is that the uncertainty active learning set performs better than the novelty based set in a consistent manner (significant at the 5% level up to 20 examples, and at the 1% level subsequently). Since these sets share one third of their examples, this demonstrates how uncertainty based active learning complements novelty based active learning. Also of note is the variance in performance of each training set. The two active learning approaches achieve a (statistically significant) lower variance than the expert chosen set at each subset size. However, the uncertainty based approach achieves a lower variance than the novelty based approach; this difference is statistically significant (at the 1% level) for subsets of size 30 and greater. This concretely demonstrates the advantage that uncertainty based active learning has over novelty based active learning: it specifically chooses examples that are likely to reduce this variance. Finally, it is informative to observe that the active learning based approaches achieve better performance with fewer examples; The novelty based set has better performance with 30 or more examples than the expert chosen set with 50 examples, and the same is true with the uncertainty based set for 20 or more examples. Overall, the take away result is that choosing a training set via active learning can achieve better and more consistent performance, with equal or fewer expert demonstrated examples.

In addition to choosing training sets from scratch, active learning can also improve the performance of previously existing training sets. To demonstrate this effect, active learning was used to add additional training data to an expert chosen set. Starting from each different subset of 30 paths of the expert chosen demonstration, two iterations of active learning was performed, each adding 10 new examples. This process was repeated for both novelty and uncertainty based active learning. The results are shown in Figure 7, along with the effect of adding additional expert chosen examples. As described previously, for expert chosen examples there are diminishing returns; the difference in performance (in a paired manner) is not even statistically significant between 40 and 50 expert chosen examples. In contrast, both active learning approaches result in validation performance that is significantly better (at the 1% level) than the corresponding

expert chosen set. This demonstrates that even when efficiency is not an issue, active learning can identify important and useful examples that improve performance.

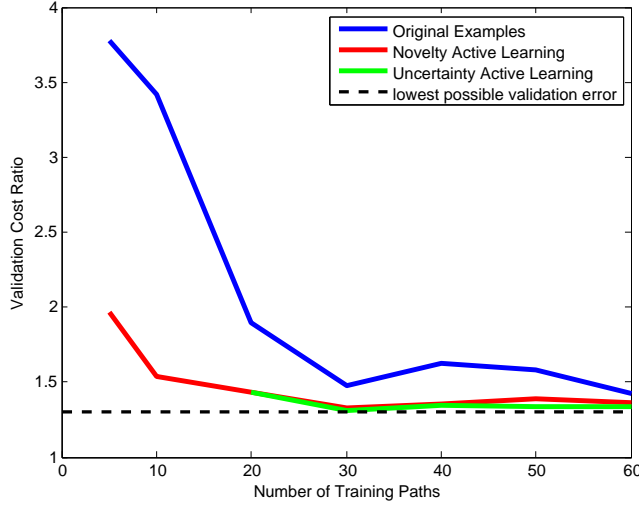
## VI. CONCLUSIONS

This work presents approaches for active learning in the context of model based learning from demonstration. While learning from demonstration can improve the performance of autonomous navigation systems while reducing the required expert interaction, it places significant responsibility on the expert to choose good examples to provide to the system. Two basic active learning heuristics are presented, one based on reducing the novelty of potential examples and another based on reducing their uncertainty. Experimental results demonstrate the overall effectiveness of both approaches, showing improved performance with equivalently sized training sets, or equivalent performance with smaller training sets.

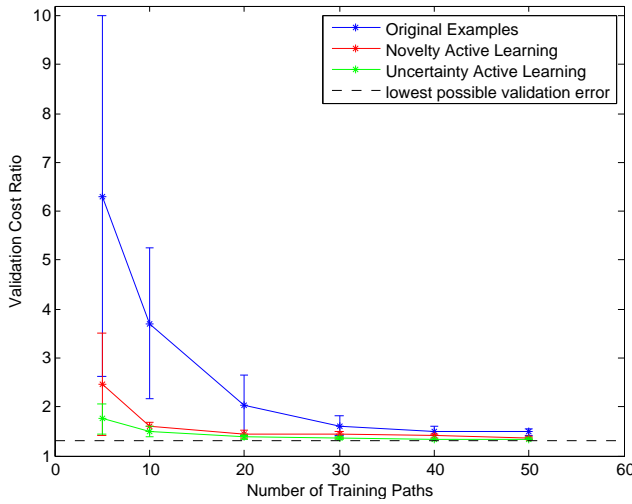
Future work will primarily be focused on applying these approaches to dynamic settings, where demonstrations involving an actual robot are necessary. We will also explore the potential applicability of these approaches in an online context for vehicle safeguarding, where a robot can request remote assistance when faced with a novel or uncertain scenario, and then learn immediately from the feedback. Finally, the use of more advanced dimensionality reduction and density estimation approaches for novelty based active learning will be explored.

## REFERENCES

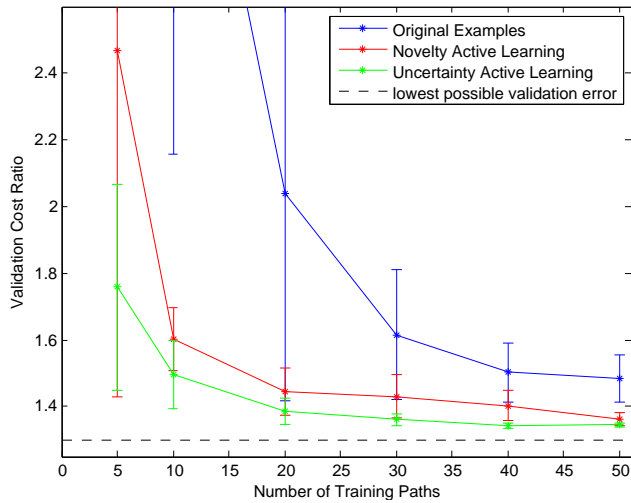
- [1] D. Silver, J. A. Bagnell, and A. Stentz, "Learning from demonstration for autonomous navigation in complex unstructured terrain," *International Journal of Robotics Research*, vol. 29, pp. 1565–1592, October 2010.
- [2] J. Z. Kolter, P. Abbeel, and A. Y. Ng, "Hierarchical apprenticeship learning with application to quadruped locomotion," in *Neural Information Processing Systems*, 2008.
- [3] M. Kalakrishnan, J. Buchli, P. Pastor, and S. Schaal, "Learning locomotion over rough terrain using terrain templates," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 167–172, 2009.
- [4] M. Zucker, N. Ratliff, M. Stolle, J. Chestnutt, J. A. Bagnell, C. G. Atkeson, and J. Kuffner, "Optimization and learning for rough terrain legged locomotion," *International Journal of Robotics Research*, vol. 30, pp. 175–191, February 2011.
- [5] S. Chernova and M. Veloso, "Confidence-based policy learning from demonstration using gaussian mixture models," in *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems*, 2007.
- [6] D. Grollman and O. Jenkins, "Dogged learning for robots," in *IEEE International Conference on Robotics and Automation*, 2007.



(a) Original Order



(b) Random Subsets



(c) Random Subsets (Zoomed)

Fig. 6. Demonstration of how training sets chosen via active learning can perform superior to training sets chosen by an expert

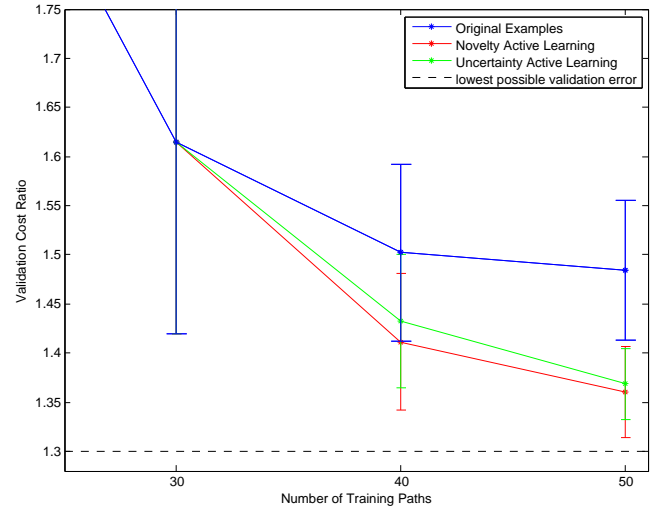


Fig. 7. Even when a reasonably sized training set already exists, active learning can identify useful and important new examples.

- [7] A. P. Shon, D. Verma, and R. P. N. Rao, "Active imitation learning," in *AAAI*, 2007.
- [8] M. Markou and S. Singh, "Novelty detection: A review - part 1: Statistical approaches," *Signal Processing*, vol. 83, no. 12, pp. 2481–2497, 2003.
- [9] C. Brooks and K. Iagnemma, "Visual detection of novel terrain via two-class classification," in *ACM symposium of Applied Computing*, pp. 1145–1150, 2009.
- [10] B. Sofman, J. A. Bagnell, and A. Stentz, "Anytime online novelty detection for vehicle safeguarding," in *IEEE International Conference on Robotics and Automation*, May 2010.
- [11] C. Dima, M. Hebert, and A. Stentz, "Enabling learning from large datasets: Applying active learning to mobile robotics," in *International Conference on Robotics and Automation*, vol. 1, pp. 108 – 114, April 2004.
- [12] C. Dima and M. Hebert, "Active learning for outdoor obstacle detection," in *Proceedings of Robotics: Science and Systems*, (Cambridge, USA), June 2005.
- [13] I. Fodor, "A survey of dimension reduction techniques," Tech. Rep. UCRL-ID-148494, Lawrence Livermore National Laboratory, May 2002.
- [14] B. Ziebart, N. Ratliff, G. Gallagher, C. Mertz, K. Peterson, J. A. Bagnell, M. Hebert, A. Dey, and S. Srinivasa, "Planning-based prediction for pedestrians," in *IROS*, 2009.
- [15] D. Lewis and W. Gale, "A sequential algorithm for training text classifiers," in *Proceedings of the International ACM-SIGIR Conference on Research and Development in Information Retrieval*, 1994.
- [16] H. Seung, M. Oppor, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual ACM Workshop on Computation Learning Theory*, pp. 287–294, 1992.
- [17] Y. Freund, H. Seung, E. Shamir, and N. Tishby, "Selective sampling using the query by committee algorithm," *Machine Learning*, vol. 28, pp. 133–168, 1997.
- [18] L. Breiman, "Bagging predictors," *Machine Learning*, vol. 24, pp. 123–140, 1996.
- [19] L. Murphy and P. Newman, "Planning most-likely paths from overhead imagery," in *International Conference on Robotics and Automation*, 2010.
- [20] D. Silver, J. A. Bagnell, and A. Stentz, "High performance outdoor navigation from overhead data using imitation learning," in *Proceedings of Robotics Science and Systems*, June 2008.
- [21] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297, 1967.