LAIP: Learned Adaptive Inspection Paths Using Offline Reinforcement Learning

Samuel Matloob*, Ayan Dutta†, O. Patrick Kreidl†, Swapnonel Roy† and Ladislau Bölöni*
*Dept. of Computer Science, University of Central Florida, Orlando, FL, USA
†School of Computing, University of North Florida, Jacksonville, FL, USA

***Abstract*—In many scenarios for informative path planning done by ground robots or drones, certain types of information are significantly more valuable than others. For example, in the precision agriculture context, detecting plant disease outbreaks can prevent costly crop losses. Quite often, there is a limit on the exploration budget, which does not allow for a detailed investigation of every location. In this paper, we propose Learned Adaptive Inspection Paths (LAIP), a methodology to learn policies that handle such scenarios by combining uniform sampling with close inspection of areas where high-value information is likely to be found. LAIP combines Q-learning in an offline reinforcement learning setting, careful engineering of the state representation and reward system, and a training regime inspired by the teacher-student curriculum learning model. We found that a policy learned with LAIP outperforms traditional approaches in low-budget scenarios.**

***Index Terms*—informative path planning, reinforcement learning**

I. Introduction

Exploring a geographical area with sensors or cameras carried by a mobile robot (such as a drone) is a technology with many practical applications, ranging from building inspections to precision agriculture. A frequent expectation for the robot path is to achieve coverage. For instance, a coverage-guaranteeing path might follow a back-and-forth lawnmower or boustrophedon ("as the ox goes") pattern [4], or a closely related spiral movement. In practice, many deployments operate within a budget that doesn't allow a detailed inspection of every location; thus instead of perfect coverage, the user might need to settle for uniform density sampling. This can be achieved, for instance, with a looser lawnmower or spiral pattern that does not "cover" every point in the area but will at least reach the vicinity of every point. From this set of observations, an estimator can then create a full map of the measured quantity. If no a priori knowledge exists and the customer is agnostic about the direction in which the estimated model differs from the ground truth, uniform sampling provides the best set of observations for the estimator.

There are, however, examples when certain types of information are significantly more valuable to the customer. For instance, in an agricultural application, the outbreak of a disease such as the tomato yellow leaf curl virus (TYLCV) [10] can lead to a total crop loss. Thus, learning about a TYLCV outbreak is significantly more important than learning about over-watered patches. Similar considerations apply to roof damage or dangerous gas leaks in pipelines. In these situations, uniform sampling does not align with the interest of the customer, who would want the affected zones to be *inspected in detail* – possibly even accepting the possibility that the rest of the area will be covered at a lower resolution.

The desired trajectory of the robot can be characterized through several attributes. First, the trajectory has to be *adaptive*: the path of the robot cannot be planned ahead of time because it depends on the observations. If the type of observations can be classified into two classes of normal and high value (or negative and positive observations), we can conjecture that the robot's behavior will also alternate between a normal, coverage-optimizing behavior and a behavior that focuses on close inspection of the affected area.

The uniform sampling behavior is a well-understood path planning technology. As the optimal coverage path only depends on the geometry of the area, it can be planned offline (for instance, with a lawnmower or spiral coverage), with the task of the robot remaining to enact the pre-calculated path. The close inspection behavior, however, is significantly more complex. We do not know the size or shape of the area that will need to be inspected. Decisions need to be made about what kind of observations should trigger close inspection, what path should be followed, and when and how to return to the uniform sampling behavior.

In contrast to coverage path planning, which can be decided from just geographical information, the optimal close inspection behavior depends in a complex and probabilistic

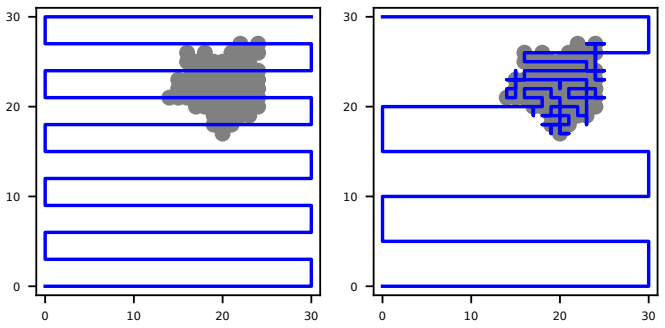

Fig. 1: Possible exploratory paths for a disease outbreak in an agricultural area. Left: a lawn-mower type uniform sampling path. Right: an adaptive path that switches to close inspection of the detected disease patches (LAIP, this work).

This work is supported in part by NSF CPS Grants #1932300 and #1931767.

way on the likely shape of the affected area. In general, we do not have a formal model (even probabilistic) of the shape of the area. For instance, in the case of a disease outbreak, oil spill, or gas leak, the shape of the area might be affected by the time since the initiation, the propagation laws, wind, and other factors. Even when a theoretical model exists, finding an optimal path to perform the inspection remains a complex challenge, arguably much harder than the coverage path planning problem, as the latter does not need to deal with uncertainty and can often assume areas of regular shape.

Due to the manyfold, difficult-to-formalize uncertainties of the problem, a learning-based algorithm promises to be a better fit than hand-crafted solutions. In particular, the problem can be naturally modeled with a reinforcement learning setup, where the high-value observations are directly translated into rewards. Unfortunately, the naive online application of such a model is not feasible. Practical considerations make it impossible for a robot to learn the path through the rewards obtained while exploring the area, as every RL rollout would either bring no reward or would risk the total crop loss of an agricultural field. Furthermore, while some ground truth maps of well-investigated outbreaks might be available, in most cases the number of these is much smaller than the number of training runs typically required by reinforcement learning algorithms.

In this paper, we develop Learned Adaptive Inspection Paths (LAIP), an offline reinforcement learning methodology to solve the informative path planning problem in the presence of high-value information. Our running example will be the exploration of a costly disease (such as TYLCV) in an agricultural field. The main contributions of the paper are:

- Developed a state representation and reward system for RL-based informative path planning that induces a behavior of uniform sampling when encountering low-value (negative) samples and detailed inspection when encountering high-value (positive) samples.
- Developed an offline reinforcement learning workflow, inspired by the teacher-student curriculum learning model that allows the efficient learning of a robot behavior starting from a small number (even possibly a single) example ground truths.
- Qualitatively and quantitatively investigated the learned behavior and compared it with several baseline informative path planning algorithms. We found that the learned behavior promotes a more thorough exploration of the diseased area.

II. Related work

The problem of informative path planning (IPP), which aims to find the optimal trajectory for a sensor-equipped robot, has an extensive literature. The optimal path depends on the overall goal of the exploration, the capabilities of the robot, prior knowledge about the environment and its dynamics, as well as the performance of the estimator that transforms observations into a model of the environment.

Probably the easiest setting for IPP is one in which we assume that there is no variation in the information content of the various locations and no geographical correlation between the measured information. In this case, the problem becomes one of coverage of the areas of interest [3]. Even under these simplifying assumptions, the problem is NP-hard.

If the observations at various locations are not independent, coverage is not the best proxy metric for the quality of the model. Models based on spatial statistics, such as Gaussian processes [13] can use techniques that model the correlations between values measured at various locations. Under these assumptions, path planning might be seen as a process to optimize the collected information [2], [14], [15].

In practice, the information maximization problem often needs to be solved in the context of constraints on the movement and communication abilities of the robot. For instance, Binney et al. [1] solves the informative path planning problem in the case of an underwater glider, which needs to avoid the shipping lanes at high traffic hours. The path planner also needs to consider that while the glider can collect data while moving, it cannot communicate it until it resurfaces. Another complication is that the value of information can vary depending on the area or (as in the case considered in this paper) on the content of the information.

This variability in the capabilities of the robots and the optimization criteria makes various informative path planning algorithms very difficult to compare against each other. Things are more manageable if we restrict ourselves to a specific application area, where the domain enforces a certain model of the environment, the commonly used robots or drones provide a limitation on the sensor suite, and the economical necessities of the application frequently dictate the optimization criteria. Thus, benchmark suites for path planning often focus on a particular application – for instance, the Waterberry Farms benchmark specifically considers a strawberry and tomato farm and measures pairs of path planners and estimators [7].

In the remainder of this section, we consider several examples in which the informative path planning problem is considered either in settings related to ours (agriculture) or with technologies similar to the ones we are considering.

Popović et al. [11] study the IPP problem in the context of weed detection in precision agriculture. Unusually for IPP projects, the work allows the UAV to follow a 3D trajectory, which is limited by the feasibility model of the robot and the observation quality limited by the height. The quality of the solution is measured by the entropy of the map built by the observations. The proposed approach was shown to create a map with a lower entropy compared to a lawn-mower-style path. Both [11] and our paper solve the similar problem of abnormalities in the an agricultural field (weeds in the case of the [11], tomato yellow leaf curl virus in our case). The primary difference in the problem setup is that TYLCV is a disease spreading from plant to plant which justifies our exploration model switching between uniform sampling and detailed exploration starting from a point. The mathematical techniques deployed for optimization are also

different, evolutionary computing for [11] and reinforcement learning for our model.

Mishra et al. [9] consider the estimation of quantities such as chlorophyll concentration or temperature in a marine area using an underwater robot. A sparse Gaussian Process estimator is used to obtain an approximation of the scalar field from the observations. An adaptive algorithm is used to plan the next observation, taking into consideration both the variance in the current prediction and the constraint of the remaining mission time.

Zhao et al. [17] consider the problem of coverage path planning using multiple agents. The approach decomposes the problem into a higher-level multi-agent path planning problem and a lower-level single-agent coverage path planning for a certain sub-region. The latter problem is solved with a lawn-mower type of algorithm. The multi-agent path planning problem is solved by formulating it as a centralized-training / distributed-execution multi-agent reinforcement learning problem, which also takes into consideration the remaining energy of the robots.

An interesting insight can be gained from taking a wider perspective on the problem of IPP. At a given point in exploration, IPP encourages the robot to *visit* certain areas, whereas collision avoidance algorithms aim to find paths that *avoid* certain areas. These can be seen as complementary problems that might allow for similar algorithms. For instance, Du et al. [5] proposed a deep RL algorithm for simultaneously avoiding multiple obstacles.

Said et al. [12] proposed mean-field deep reinforcement learning to find an optimal exploration path for multiple robots, with each robot using a recurrent neural network-based model and the reward being based on a mutual information objective.

Matloob et al. [8] propose an exploratory path based on splitting the area of interest into a grid, choosing a number of random waypoints from the grid cells, and visiting them in an order defined either by an approximated TSP or various heuristics. Experiments show that this approach, positioned in the design space between a systematic lawn-mower algorithm and a random waypoint approach, can exhibit desirable properties that make it more suitable for certain applications compared to both endpoints.

III. Algorithm

A. Formalizing the IPP problem in environments with high-value information

Let us consider a geographical area of size $x_{max} \times y_{max}$ where every location is denoted by its integer coordinates (x, y). We assume that the environment describes a scalar field, where the individual grid cells are described by a matrix $E(x, y) \in \mathbb{R}$, where the individual values represent the ground truth which ranges from 0.0 to 1.0. In our running example, this formalism maps to a tomato field, where 1.0 represents a location infected with the Tomato Yellow Leaf Curl Virus (TYLCV) and 0.0 represents a healthy location.

A robot is exploring this area of interest, its state being described by its current location $s = (x, y)$. The observation made by the robot is the current value of the cell. After every timestep, the robot will take a movement action $a \in \{a_N, a_S, a_W, a_E\}$ which can move it north, south, west, or east, as long as the robot does not leave the area of interest. Through its movement in the area, the robot will generate a series of observations $(x_0, y_0, E[x_0, y_0]), (x_1, y_1, E[x_1, y_1]) \ldots$.

An estimator takes the observations as input and generates an approximation of the overall map of disease outbreaks. Let us now consider what type of information an estimator can infer from these observations and what this means for finding the optimal path.

The most trivial estimator would simply remember the observed values and mark the other values as unknown. For such an estimator, the only criterion of a good path would be that whenever the robot revisits an already visited location, the new observation would not add any new information.

However, a more performant estimator might extract additional information based on knowledge of the dynamics of the disease. For example, it might be possible to infer the likelihood that a location is infected from the observation of nearby cells. For such an estimator, the best quality estimation can be obtained by spreading the limited number of observations in such a way that they form a relatively uniform sample of the area of interest. For example, a lawn-mower pattern that covers the area uniformly is a good path to be used in conjunction with such an estimator.

What makes our problem specific, however, is that in the case of disease detection, our goal is not to obtain the most exact approximation model, but the most valuable one. In particular, information about diseased locations is significantly more valuable than information about healthy locations. The information asymmetry is so large that for the robot it is justifiable to switch to a dense, close inspection when detecting a patch of diseased area, as shown in Figure 1.

To translate this insight into actual robot behavior, we need operational answers to several questions. What triggers the transition into the close inspection mode? What makes the robot leave this mode and return to uniform sampling? Which direction should the robot inspect first from the current location, and how long should it pursue a given direction? An important fact is that the answers to these questions depend on the size and distribution of the disease patches, which in turn depend on the dynamics of disease propagation, wind, and insects that transmit the disease, with a significant probabilistic component. For instance, if the typical distribution of the disease outbreaks is in randomly distributed independent cells, the inspection of the area around the cell is not necessary. In general, it is not cost-effective for stakeholders to develop a detailed environment model to guide a robot on an IPP path. However, a limited number of ground truth samples from different or previous outbreaks are usually available.

Thus, our proposed approach is to develop a learning-based technique for the development of robot behavior. We call this approach Learned Adaptive Inspection Paths (LAIP). We do not have previous robot path examples, thus supervised or imitation learning is not possible. Due to the high cost

of running a suboptimal path in an agricultural field, online reinforcement learning is also not a realistic proposition. On the other hand, the problem fits well with offline reinforcement learning because, with the right state representation, the requirements of the problem can be conveniently expressed in the form of rewards.

At the core of our approach is a standard tabular Q-learning algorithm [16]. However, what distinguishes our representation is the set of careful decisions regarding the state representation and rewards. Furthermore, because our algorithm will need to operate in an offline setting, with a very small number of samples (possibly only one), the training regime needs to specifically take this into account. Our approach takes inspiration from the teacher-student curriculum learning model [6]. In the remainder, we describe these choices in detail.

B. State representation

The objective of the state representation in reinforcement learning is to capture sufficient information to inform the behavior policy $\pi(s) : s \rightarrow a$. Note that the full state of the robot includes its current location, previous observations, and the previous trajectory. Note that the MDP underlying the RL algorithm follows the Markov property by generating a policy that does not depend on the history of the states. On the other hand, it is standard practice for the state representation to capture knowledge that depends on the history. However, folding the complete history of the robot into the state would make the state space impractically large. Thus, we need to find ways to reduce the state space.

In LAIP, we will accomplish this by making the state representation local: it will only contain information about the immediate neighborhood of the robot. More precisely, the chosen state representation will contain six pieces of information: about the current location, about the four neighboring locations that can be reached through one action, and about the previous action taken by the robot. For ease of reference, we will assign letter codes to the state components. Information about the current location can be H - healthy, first time visiting; HH - healthy previously visited; D - diseased, first time visiting; and DD - diseased, previously visited. The state representation also includes the number of steps up to follow the lawn mower up directions.

The information about the neighboring locations will be prefixed with the direction code (N, S, E, W). The state of these locations can be "?" - not explored; H - healthy; D - diseased; and "-" - out of the environment. Finally, the previous action will be prefixed with A, and it can be any of the directions or "-" if the agent did not move or it was at the beginning of the episode. Figure 2 shows three examples of such state representation.

C. Designing the reward function

The objective of the reward function $r(s)$ is to capture our intuition about the desirable aspects of the movement policy of the robot. The primary objective of LAIP is to discover as many diseased locations as possible; the positive reward

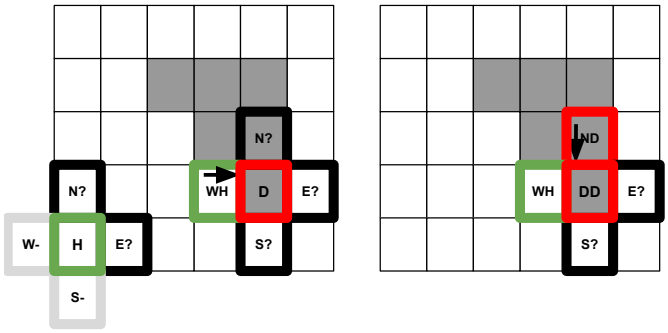

Fig. 2: Illustrating the state representation with three example states of the robot from an exploration episode. The first state {H, N?, S-, W-, E?, A-} represents the agent at the beginning of the episode. The second state, shows the agent the first time visiting a cell, coming from the direction of west. The third state shows the agent revisiting the same cell, coming from the north direction.

associated with this is the most important component of the reward system. The general consensus of the reinforcement learning community is that it is best to minimize the hand-engineered part of the rewards and allow the learning process to discover an appropriate behavior. We found, however, that relying only on rewarding discovered diseased locations does not lead the agent to acquire the desired behaviors, making it necessary to add auxiliary rewards to achieve the goals. Nevertheless, we still aim to limit the type of hand-engineered information: we found that we don't need to explicitly specify the duality of the uniform sampling vs. close inspection behavior, nor the properties of the path taken by the robot during close inspection. However, we found that the robot will not evolve by itself a uniform sampling path covering the area of interest. Therefore, in our reward system, we have explicitly encouraged the robot to take a lawn-mower-type path while not encountering diseased areas while discouraging it from repeatedly revisiting the same area. These values had been chosen empirically to convey to the learning system the relative importance of various behaviors. While the exact values are less important, their sign and relative scale are used to convey the overall desired behavior. The components of the LAIP reward system can be summarized as follows:

- $r = +1$ whenever the agent takes a movement that coincides with the lawn-mower path. To discourage incorrect learning about the previous state, this reward is not applied if the current location is diseased, but the previous one was not.
- $r = -15$ if the agent revisited a previously visited area. To encourage the exploration of the area from south to north, this reward is $r = -30$ if the agent was moving to the south.
- $r = -30$ if the agent is moving south while the previous location was not diseased.
- $r = +30 \cdot y\_steps$ if the agent changes from going to the north into a west or east location at the appropriate

position of the lawn-mower path.

- $r = +10$ if both the current and the previous locations were diseased.
- $r = -30$ if the current location is diseased and the agent is not on the lawnmower path.
- $r = -20$ if the current location is not diseased and the agent is not on the lawnmower path.

D. Training regime

The overarching goal of the training is to develop a policy that achieves the desired exploration behavior in *typical scenarios*. In our running example, we only have a single ground truth map. Offline RL training, using this single map as training data, raises the risk of the policy overfitting to the scenario, obtaining a good result, but only if the disease outbreak is at a specific location and has the exact expected shape. In our experiments, the policy trained in this way performed very poorly when deployed on other maps.

To improve the generality of the learned behavior, we deploy a more complex training regime inspired by the teacher-student curriculum learning model [6]. In the inner loop of this policy, the student agent uses repeated runs of Q-learning with ϵ-greedy exploration and a specific learning rate α to improve the policy on a given map. Throughout the inner loop runs, ϵ and α follow a standard decay schedule.

In the outer loop of the process, the teacher determines the tasks the student will train on, provides the tasks for the student in the inner loop, evaluates the progress of the learning, and decides on the termination of the training. Over the iterations of the inner loop, the student agent starts the new training process with the Q-table obtained from the previous student-run. The overall output of the process is the final Q-table obtained by the student.

In our case, the various tasks differ by the ground truth map of the disease. As we have only a small number of such maps, to promote more generalizable student behavior, the teacher process creates a larger number of tasks through *augmentation*. Starting from one sample with a recorded disease outbreak, the teacher process generates several tasks by a) shifting the location of the outbreak over the x and y dimensions, b) adding multiple outbreaks with different shifts, and c) making small modifications in the shape of the outbreak.

Algorithm 1: LAIP training regime

Input :
  e_{gt}; /* ground truth env. */
  α; /* initial training rate */
  ϵ; /* par. of epsilon-greedy */
Output:
  $Q(S, A)$; /* Q-table */
$E \leftarrow$ teacher_generate_tasks(e_{gt})
$Q \leftarrow$ initialize_randomly()
repeat
  repeat
    task $\leftarrow$ teacher_select_task(E)
    student_init(Q, α, ϵ)
    $Q_{new} \leftarrow$ student_run_episodes(*task*)
    $Q \leftarrow Q_{new}$
  until *all tasks considered*;
  $r_{total} \leftarrow$ teacher_evaluate_total_reward(E)
until r_{total} *is satisfactory*;
return $Q(S, A)$

IV. Experiments

A. Experimental setup

To study the properties of the proposed approach, we trained LAIP policy using the representation, reward system, and training regime described in Section III. As a ground truth sample, we used a single sample outbreak of the TYLCV disease in tomato fields, as shown in Fig. 1, generated by the Waterberry Farms benchmark [7]. We used a grid of 30 x 30 cells, initial values of $\alpha = 0.1$, $\epsilon = 0.6$ and $y\_steps = 5$. The number of synthetic environments generated for training was 16. The trained policy (which is fully determined by the Q-table) was saved and used in all subsequent experiments in this section. For the sake of conciseness, in the remainder of this section, we will use the term LAIP to refer to this particular policy. Whenever not specified otherwise, the budget was set to 400 steps throughout the experiments.

B. A qualitative evaluation of the generalizability of LAIP

Our first series of experiments involved a qualitative evaluation of the learned policy, especially with regard to its ability to generalize to environments that it had not seen during training. The sixteen examples (A) through (P) in Fig. 3 show the behavior exhibited by the learned policy, illustrating both the strengths and the limitations of the learned behavior.

Our first observation is that indeed all the examples show the expected behavior of switching between uniform sampling and close inspection when encountering a diseased area. This behavior switch is correctly exhibited for both one diseased area (B to J) and two diseased areas (K to P). As expected, when there is no diseased area, the resulting policy correctly follows a uniform sampling path as in sample (A).

We find that the policy, in most cases, performs a thorough inspection of the diseased area, discovering a large fraction of the diseased cells. However, in certain scenarios, such as (E), (F), and (G), it returns to the uniform sampling behavior prematurely, after inspecting about 80% of the diseased area. If there is more than one diseased area, this sometimes occurs in one of the patches, as shown by examples (N) and (M).

Another challenge is that when the robot is focusing on the inspection of a diseased area, it sometimes does not entirely cover the remaining area with a lawn-mower span. In some cases, such as (J), the missed areas can be significant.

Another phenomenon we can observe is that the robot might exhaust its exploration budget through the close inspection process, and thus it needs to skip the remainder of the field. This situation is most likely to happen if the diseased areas are large, for instance in examples (L) and (P). Note that there is no easy answer to whether this behavior is desirable or

undesirable. We will further explore this tradeoff in the next subsection.

C. Efficient use of the budget

One of the critical challenges of exploratory path planning is the limited budget of the robots. This might appear in the form of fuel or battery charge limitations or available daylight. In other situations, there is simply a limited amount of time the robot can devote to a particular task.

If the exploration budget is unlimited and the observed phenomena do not change, a coverage path will be able to observe every location. In the case of a limited budget, the situation is more complex, as the algorithms must consider the budget in the path planning process. This budget adaptation can be very simple: for instance, a random waypoint algorithm might move to random waypoints until it runs out of the budget and then returns to base. For a lawn-mower algorithm, a more complex optimization is needed. The fixed-budget lawnmower (FBLM) algorithm shown in the upper row of Fig. 4 finds the densest lawnmower pattern that can be accommodated by the budget. We notice that with a budget of 960, this algorithm can achieve complete coverage of the grid of interest. Thus, FBLM uses a predefined path that adapts to the communicated budget, but it is not adaptive with regard to the observations.

In contrast, LAIP adapts to the observed data, but it does not take into account the remaining budget in its movement decisions – in our experiments, we assume the simple model of the robot returning to the base when the exploration budget is exhausted. The results for several budgets are shown in the lower row of Fig. 4. We see that with a budget of 240 steps, the agent runs out of budget during the close inspection of the diseased patch. With a budget of 360 steps or higher, LAIP is able to finish the exploration of the patch; however, it is not able to take advantage of the higher budget to increase the density of the sampling.

Comparing FBLM and LAIP with different budgets, we find that LAIP performs much better with a limited budget: even with a budget of 240, LAIP found more diseased cells compared to FBLM with a budget of 510. However, with a large budget, systematic methods like FBLM can achieve complete coverage, making adaptive methods unnecessary. The comparison also motivates a direction for future research on learned policies that adapt not only to observations but also take into consideration the remaining budget.

D. Model quality

The ultimate objective of exploratory path planning is to develop a model of the environment. Thus, an end-to-end evaluation of the path planning policy should not consider the path or the list of the observations made along it, relying instead on the quality of the model that the estimator can create from these observations.

Fig. 5 compares LAIP against several recent algorithms from [8]. These algorithms are variations of the Grid Limited Randomness models (GLR-EOP, GLR-SD, and GLR-CA). GLR algorithms are not adaptive, but for limited budgets were found to exceed both random waypoint and FBLM in terms of the accuracy of the model. For all algorithms, the adaptive disk (AR) estimator was used.

We find that LAIP obtains the best approximation of the shape of the disease outbreak as shown in Fig. 5 (middle row). However, the cost of this precision was that some parts of the area were left with relatively large uncertainty values, a problem shared with the GLR-EOP algorithm. The higher uncertainty portions are shown with yellow color in Fig. 5 (bottom row).

V. Conclusion

In this paper, we introduced LAIP, a methodology to learn policies for robots exploring an area characterized by the fact that certain types of information have a very high value, justifying the close inspection of certain areas. LAIP combines Q-learning in an offline reinforcement learning setting, careful engineering of the state representation and reward system, and a training regime inspired by the teacher-student curriculum learning model. We evaluated an LAIP-trained policy for a precision agriculture application of investigating lethal plant disease outbreaks. We found that the learned policy outperforms traditional approaches in low-budget scenarios. However, the learned policy shows some weaknesses that suggest directions for future work: the policy does not dynamically take into account the remaining exploration budget and in some scenarios, the close inspection of disease patches leads to the skipping of other areas. Another direction is the development of more general reward functions that do not require fine-tuning to the specifics of the scenario.

References

[1] J. Binney, A. Krause, and G. S. Sukhatme. Informative path planning for an autonomous underwater vehicle. In *IEEE Int. Conf. on Robotics and Automation (ICRA-2010)*, pages 4791–4796, 2010.

[2] C. Chekuri and M. P'al. A recursive greedy algorithm for walks in directed graphs. In *Proc. of IEEE Symposium on Foundations of Computer Science (FOCS-2005)*, pages 245–253, 2005.

[3] H. Choset. Coverage for robotics – A survey of recent results. *Annals of Mathematics and Artificial Intelligence*, 31(1-4):113–126, 2001.

[4] H. Choset and P. Pignon. Coverage path planning: The boustrophedon cellular decomposition. In *Field and Service Robotics*, pages 203–209. Springer, 1998.

[5] Y. Du, J. Zhang, J. Xu, X. Cheng, and S. Cui. Global map assisted multi-agent collision avoidance via deep reinforcement learning around complex obstacles. In *Proc. 2023 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 298–305, 2023.

[6] T. Matiisen, A. Oliver, T. Cohen, and J. Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020.

[7] S. Matloob, P. P. Datta, O. P. Kreidl, A. Dutta, S. Roy, and L. Bölöni. Waterberry farms: A novel benchmark for informative path planning. *arXiv preprint arXiv:2305.06243*, 2023.

[8] S. Matloob, A. Dutta, P. Kreidl, D. Turgut, and L. Bölöni. Exploring the tradeoffs between systematic and random exploration in mobile sensors. In *Proc. of the Int. ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWIM-2023)*, pages 209–216, 2023.

[9] R. Mishra, M. Chitre, and S. Swarup. Online informative path planning using sparse Gaussian processes. In *Proc. of 2018 IEEE OCEANS Conference*, pages 1–5, 2018.

[10] E. Moriones and J. Navas-Castillo. Tomato yellow leaf curl virus, an emerging virus complex causing epidemics worldwide. *Virus Research*, 71(1-2):123–134, 2000.

Fig. 3: Investigating the ability of LAIP to generalize to scenarios not seen during training.

[11] M. Popović, G. Hitz, J. Nieto, I. Sa, R. Siegwart, and E. Galceran. Online informative path planning for active classification using UAVs. In *Proc. of IEEE International Conference on Robotics and Automation (ICRA-2017)*, pages 5753–5758, 2017.

[12] T. Said, J. Wolbert, S. Khodadadeh, A. Dutta, O. P. Kreidl, L. Bölöni, and S. Roy. Multi-robot information sampling using deep mean field reinforcement learning. In *Proc. of IEEE Conference on Systems, Man and Cybernetics (SMC 2021)*, pages 1215–1220, October 2021.

[13] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *J. Artif. Intell. Res. (JAIR)*, 34:707–755, 2009.

[14] A. Singh, A. Krause, C. Guestrin, W. J. Kaiser, and M. A. Batalin. Efficient planning of informative paths for multiple robots. In *Proc. of Int. Joint Conference on Artificial Intelligence (IJCAI-2007)*, volume 7, pages 2204–2211, 2007.

[15] A. Singh, A. Krause, and W. J. Kaiser. Nonmyopic adaptive informative path planning for multiple robots. In *Proc. of Int. Joint Conference on Artificial Intelligence (IJCAI-2009)*, pages 1843–1850, 2009.

[16] R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[17] C. Zhao, J. Liu, S.-U. Yoon, X. Li, H. Li, and Z. Zhang. Energy constrained multi-agent reinforcement learning for coverage path planning. In *Proc. of 2023 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS-2023)*, pages 5590–5597, 2023.

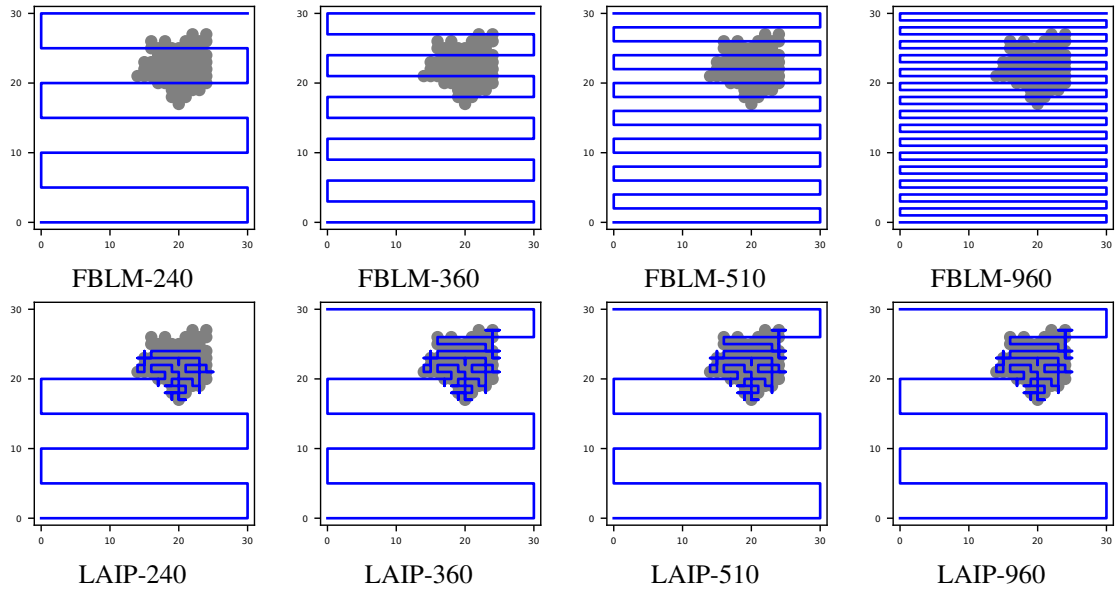

Fig. 4: Exploratory path planning with a limited budget of 240, 360, 510, and 960 steps. Top row: fixed-budget lawnmower (FBLM) algorithm. Bottom row: the LAIP learned policy.

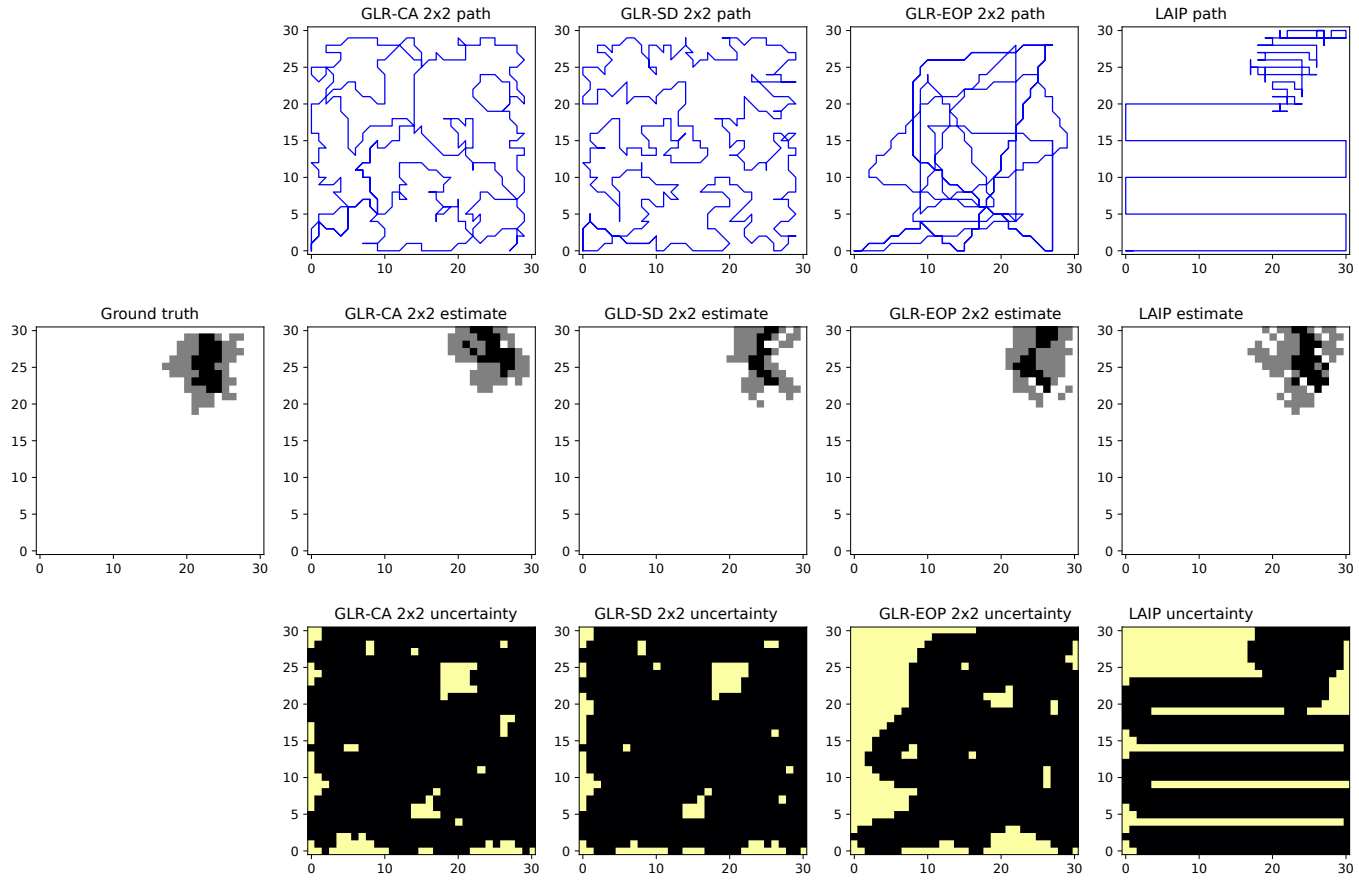

Fig. 5: Comparing the model building of LAIP to the GLR family of models. The first row shows the path created by the various algorithms. The second row shows the ground truth and the model outputs of the various algorithms, with a white color showing a healthy area, while grey and black show various levels of disease. The third row shows the uncertainty of the estimator, with black indicating lower and yellow indicating higher uncertainty.