 doi:10.20944/preprints202108.0018.v1

*Article*

# Proximal Policy Optimization for Radiation Source Search

**Philippe Proctor [1,†,*], Christof Teuscher [1,†], Adam Hecht [2] and Marek Osiński [3]**

[1] Maseeh College of Engineering and Computer Science, Portland State University; {pproctor,teuscher}@pdx.edu
[2] Department of Nuclear Engineering, University of New Mexico; hecht@unm.edu
[3] Center for High Technology Materials, University of New Mexico; osinski@chtm.unm.edu
* Correspondence: pproctor@pdx.edu;
† Current address: 1900 SW 4th Ave, Portland, OR 97201

**Abstract:** Rapid search and localization for nuclear sources can be an important aspect in preventing human harm from illicit material in dirty bombs or from contamination. In the case of a single mobile radiation detector, there are numerous challenges to overcome such as weak source intensity, multiple sources, background radiation, and the presence of obstructions, i.e., a non-convex environment. In this work, we investigate the sequential decision making capability of deep reinforcement learning in the nuclear source search context. A novel neural network architecture (RAD-A2C) based on the *actor critic* (A2C) framework and a particle filter gated recurrent unit for localization is proposed. Performance is studied in a randomized $20 \times 20$ m convex and non-convex environment across a range of *signal-to-noise ratio* (SNR)s for a single detector and single source. RAD-A2C performance is compared to both an information-driven controller that uses a bootstrap particle filter and to a *gradient search* (GS) algorithm. We find that the RAD-A2C has comparable performance to the information-driven controller across SNR in a convex environment and at lower computational complexity per action. The RAD-A2C far outperforms the GS algorithm in the non-convex environment with greater than 95% median completion rate for up to seven obstructions.

**Keywords:** deep reinforcement learning; source search and localization; active search; gamma radiation; source parameter estimation; sequential decision making; non-convex environment

## 1. Introduction

The advancement of nuclear technology has brought the benefits of energy production and medical applications, but also the risks associated with exposure to radiation [1]. Radioactive materials can be used for dirty bombs, or might be diverted from its intended use. Effective detection when these types of materials are present in the environment is of the utmost importance and measures need to be in place to rapidly locate a source of radiation in an exposure event to limit human harm [2].

Detection, localization, and identification are based upon the measured gamma-ray spectrum from a radiation detector. Radioactive sources decay at a certain rate which, with the amount of material, gives an activity, often measured in disintegrations per second or Becquerels [bq]. Most decays leave the resulting nucleus in an excited state, which may lose energy by emitting specific gamma rays. Localization methods in the current work rely upon the intensity [cps] of the gamma photon radiation measured by scintillation detectors composed of materials such as sodium iodide (NaI) [3]. The number of counts per second recorded by a detector is related to the total photons emitted per second through a scaling factor determined by detector characteristics. It is common to approximate each detector measurement as being drawn from a Poisson distribution because the success probability of each count is small and constant [3]. The inverse square relationship, $\frac{1}{r^2}$, is a useful approximation to describe the measured intensity of the radiation as a function of the distance between the detector and and source, r. The

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

size of the detector also affects count rates, with a larger detector having a larger solid angle. This nonlinear relationship paired with the probabilistic nature of gamma-ray emission and background radiation from the environment leads to ambiguity in the estimation of a source's location.

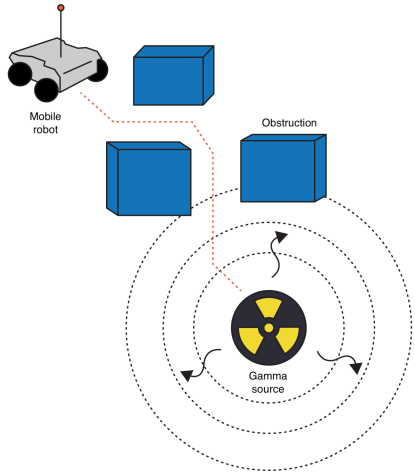



**Figure 1.** An autonomous mobile robot operating in a non-convex environment. The unshielded gamma source emits gamma radiation isotropically. Obstructions (blue cubes) attenuate the gamma radiation signal and block the robot's path.

In the case of a single mobile detector, there are numerous challenges to overcome. Detectors deployed to smaller autonomous systems such as drones or robots have a smaller surface area and volume resulting in poorer counting statistics per dwell time. Common terrestrial materials such as soil and granite contain *naturally occurring radioactive materials* (NORM) that can contribute to a spatially varying background rate [3]. Far distances, shielding with materials such as lead, and the presence of obstructions, i.e., a non-convex environment, can significantly attenuate or block the signal from a radioactive source. Further challenges arise with multiple or weak sources. Given the high variation in these variables, the development of a generalizable algorithm with minimal priors becomes quite difficult. Additionally, algorithms for localization and search need to be computationally efficient due to energy and time constraints. Figure 1 shows an example illustration of a mobile robot performing active nuclear source search in a non-convex environment.

### *1.1. Machine Learning (ML)*

ML is broadly concerned with the paradigm of computers learning how to complete tasks from data. *Reinforcement learning* (RL) is a subset of ML focused on developing a control policy that maximizes cumulative reward in an environment. *Deep learning* (DL) is another subset of ML with an emphasis on learning a function of interest using data. A key difference between RL and other subsets of ML is that learning is dependent on the data that is gathered by the policy thereby directly impacting future learning. The intersection of RL and DL has resulted in a framework called *Deep reinforcement learning* (DRL). DRL uses deep neural networks to learn a control policy and approximate state values through trial and error in an environment. While training of these networks is computationally intensive, once the weights are learned, inference (the application of a trained ML model) can be performed at lower computational cost. In this paper, we investigate a branch of DRL known as stochastic, model-free, on-policy gradients and assess its performance in the task of control in the radiation source search domain.

DRL has far surpassed human expertise in a myriad of other tasks, for example, the board game Go, which has a state space of $10^{174}$ [4]. Since these algorithms learn strictly through environmental interaction, they can discover and develop heuristics and

 doi:10.20944/preprints202108.0018.v1

action trajectories that humans might never have considered in their algorithm design. Radiation source search is a well studied problem, however, data-driven approaches have received less attention, in part because of the high variability mentioned above. This paper demonstrates that DRL can learn an effective policy that generalizes across a range of scenarios where background rate, source strength and location, and the number of obstructions are varied.

### *1.2. Related Work*

Many solutions have been proposed for nuclear source search and localization across a broad range of scenarios and radiation sensor modalities. These methods are generally limited to the assumptions made about the problem such as the background rate, mobility of the source, shielding presence, and knowledge of obstruction layout and composition. Morelande et al. present a maximum likelihood estimation approach and a Bayesian approach to multi-source localization using multiple fixed detectors in an unobstructed environment [5]. Hite et al. also use a Bayesian approach with Markov chain Monte Carlo to localize a single point source in a cluttered urban environment by modeling the radiation attenuation properties of different materials [6]. Hellfeld et al. focused on a single detector in 3D space moving along a pre-defined path for single and multiple weak sources [7]. They utilized an optimization framework with sparsity regularization to estimate the source activity and coordinates.

There is great interest in autonomous search capabilities for source search to limit human exposure to harmful radiation. Cortez et al. proposed and experimentally tested a robot that used variable velocity uniform search in a single source scenario [8]. Ristic et al. proposed three different formulations of information-driven search with Bayesian estimation. An information-driven search algorithm selects actions that maximize the available information for its estimates of user-specified quantities at each timestep. The first method utilized the Fisher information matrix and a particle filter for a single source and single detector in an open area with constant background [9]. The second and third method both used the Renyi information divergence metric and particle filter to control a detector/detectors in open/cluttered environments with multiple sources, respectively [10],[11]. In the cluttered environment, the layout was considered to be known before the start of the search. Anderson et al. considered a single mobile detector used for locating multiple sources in a cluttered environment through an optimization based on the Fisher information and travel costs [12]. The obstruction attenuation and nuclear decay models were specified by hand.

RL and DRL have also been applied to the control of single robots. Landgren used a multi-armed bandit approach to control nuclear source search in an indoor environment [13]. This was implemented on a Turtlebot3 and used to find multiple radioactive sources in a lab through radiation field sampling. Liu et al. used double Q-learning to control a single detector search for a single radioactive source with a varying sized wall in simulation [14]. The model performed well when the test environment matched its training set but did not generalize when new geometries were introduced and had to be retrained. This approach is the most similar to the one used in this research.

In contrast to the majority of the methods mentioned above, our algorithm does not directly rely on any hard-coded modeling assumptions for decision making. This gives greater flexibility to our approach and allows the opportunity for generalization to a greater variety of situations. For example, our approach was only trained on up to five obstructions in an environment at any one time but can easily operate when greater than five obstructions are present. Additionally, it would be relatively simple to retrain the agent to account for a moving source or novel obstruction types and layouts, among other things. This comes with the caveat that there is a heavy reliance upon the assumptions made in modeling an environment that are likely to fail in capturing the intricacies of reality (reality gap). This is an area of intense interest in the DRL research space [15].

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

### 1.3. Contributions

The main contributions of this paper are an on-policy, model-free DRL approach to radiation source search, a novel neural network architecture, the RAD-A2C, and an open-source radiation simulation for convex and non-convex environments. Our approach will be evaluated in the context of single detector search for a single radiation source in a simulated 2D environment with variable background radiation, variable source intensity and location, variable detector starting position, and variable number of obstructions. The RAD-A2C will be compared against a modified information-driven search algorithm previously proposed in the nuclear source search literature and a gradient search algorithm in a convex environment across *signal-to-noise ratio* (SNR)s. We will examine the effect of obstructions on the RAD-A2C performance in a non-convex environment with comparison to a gradient search algorithm across SNRs.

## 2. Materials and Methods

### 2.1. Radiation Source Search Environment

The radiation source search environment was fundamental to the training of the policy. The development of the environment required many careful design decisions in an attempt to provide a useful proof of concept for the efficacy of DRL in practical radiation source search contexts. In the remainder of the paper, we assume that a gamma radiation source has already been detected through some other means and the objective is to now locate it. We also assume an isotropic detector and a constant background rate per search. An episode is defined to be a finite sequence of observations, actions, and rewards in an environment.

#### 2.1.1. Gamma Radiation Model

Gamma radiation measured by a detector typically comes in two configurations, the total gamma-ray counts or the gamma-ray counts in specific peaks. The full spectrum is more information rich as radiation sources have identifiable photo-peaks but is more complex and computationally expensive to simulate. Thus, our localization and search approach uses the gross counts across the energy bins. Cesium-137 was selected as the source of interest since it is commonly used in industry applications and is fairly monoenergetic [16]. We denote the parameter vector of interest as $\mathbf{x} = [\mathcal{I}_s, x_s, y_s]$, where $x_s, y_s$ are the source coordinates in [m] and $\mathcal{I}_s$ is the source intensity in *counts per second* [cps] at a source-detector distance of 1 m. These quantities are assumed to be fixed for the duration of an episode. An observation at each timestep, $n$, is denoted as $\mathbf{o}_n$, and consists of the measured counts, $z_n$, detector position denoted $[x_n, y_n]$ [m], and 8 obstruction range sensor measurements.

The background radiation rate is a constant $\lambda_b$ [cps]. The following model is used to approximate the mean rate of radiation counts measurements in an unobstructed environment (convex),

$$\lambda_n(\mathbf{x}) = \frac{\mathcal{I}_s \epsilon A \Delta t}{4\pi[(x_s - x_n)^2 + (y_s - y_n)^2]} + \lambda_b, \tag{1}$$

where $A$, $\epsilon$, and $\Delta t$, are the detector surface area [m$^2$], the detector intrinsic efficiency, and the dwell time [s], respectively. The detector intrinsic efficiency is assumed to be one and we consider a unit dwell time. The detector is assumed to be a cylinder with surface area equal to $4\pi$ and isotropic for ease of computation. This results in the following binary attenuation model when the detector does and does not have *line-of-sight* (LOS):

$$\lambda_n(\mathbf{x}) = \begin{cases} \frac{\mathcal{I}_s}{(x_s - x_n)^2 + (y_s - y_n)^2} + \lambda_b & \text{LOS}, \\ \lambda_b & \text{NLOS}. \end{cases} \tag{2}$$

Thus, the measurement likelihood function when the detector has LOS is defined as

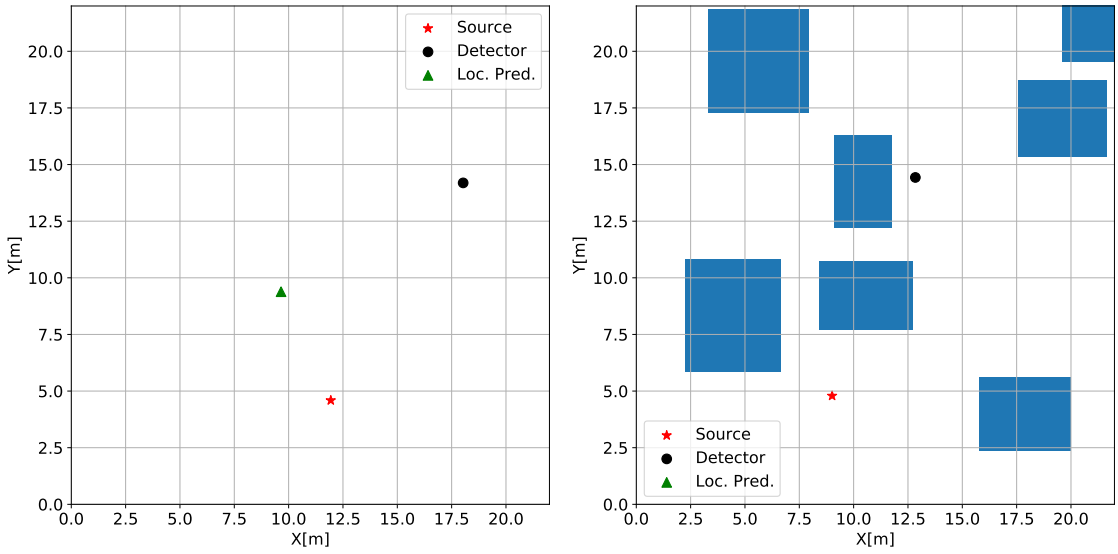
Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

$$p(z_n|\theta) = \mathcal{P}(z_n; \lambda_n(\mathbf{x})) = \frac{e^{-\lambda_n(\mathbf{x})}\lambda_n(\mathbf{x})^{z_n}}{z!}. \tag{3}$$

We define the *signal-to-noise ratio* (SNR) as,

$$\text{SNR} = \frac{\mathcal{I}_s / D_{init}^2 + \lambda_b}{\lambda_b}, \tag{4}$$

where $D_{init}$ is the initial Euclidean distance between the source and detector positions. Equation (4) was also used for the non-convex environments to maintain consistency even though it is not strictly true. Figure 2 shows a randomly generated episode for convex (2*a*) and non-convex (2*b*) environments. The environment was implemented using the open-source Gym interface developed by OpenAI [17].




**(a)** Convex environment. **(b)** Non-convex environment.

**Figure 2.** A sample of the starting conditions for (**a**) convex and (**b**) non-convex environment. In both environment types, the red star is the source position, the black circle is the detector position, and the green triangle is the agent's prediction of source position. In the non-convex environment, the blue rectangles are obstructions that block line of sight between the source and detector. The initial source prediction is in the obstruction as the agent does not have any prior information about the environment.

### 2.1.2. Partial Observability

In the context of the radiation search scenario where measurements are noisy and uncertain, it is more useful to describe the *partially observable Markov decision process* (POMDP). The finite POMDP is defined by the 6-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R}, \Omega, \mathcal{O} \rangle$ at each time step, $n$. $\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{O}$ are the states, actions, rewards, and observation, respectively. The probability distributions for observation, $\Omega$, and transition, $\mathcal{T}$, are considered fixed and unknown. An observation is a function of the true state but is not necessarily representative of the true state due to the stochastic nature of the environment.

A history is a sequence of observations up to timestep n is defined as $H_n = (o_0, o_1, ..., o_{n-1}, o_n)$. A successful policy needs to consider $H_n$ to inform its decisions since a single observation does not necessarily uniquely identify the current state. This can be implemented directly by concatenation of all previous observations with the current observation input or through the use of the hidden state, $h_n$, of recurrent neural networks. The function $M(H_n)$ provides a sufficient statistic of the past history and serves as the basis for the agent's decision making [18]. This allows the policy to be

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

reformulated as $\pi(a_{n+1}|h_n) = p(a_{n+1}, M(H_n); \theta)$ where $\theta$ is some parameterization and $a_{n+1}$ is the next action.

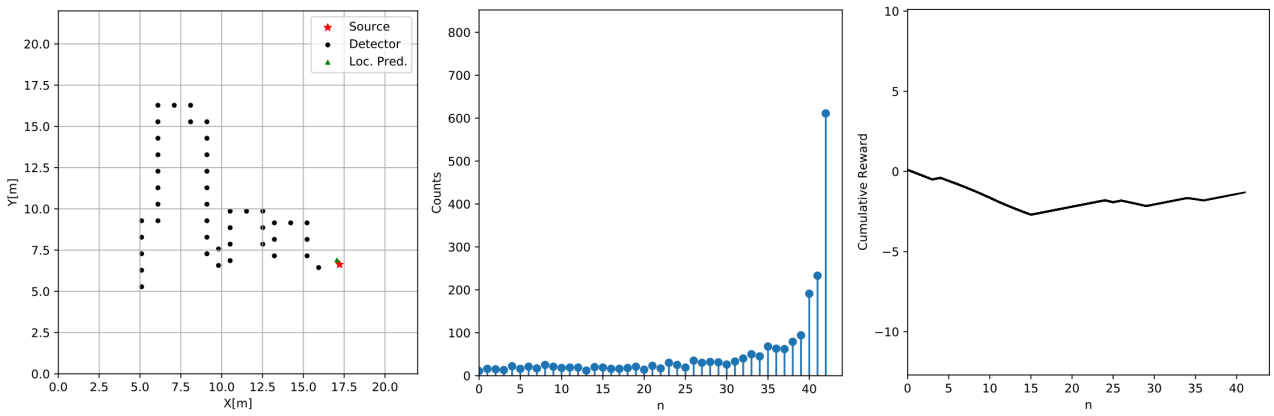
#### 2.1.3. Reward Function

The reward function defines the objective of the DRL algorithm and completely determines what will be learned from the environment. Reward is only utilized for the update of the weights during the optimization phase and does not directly factor into the DRL agent's decision making during an episode. The reward function for the convex and non-convex environment is as follows,

$$r_{n+1} = \begin{cases} 0.1 & \text{if } \psi_{n+1} < \min \psi_n, \\ -0.5 * \frac{\psi_{n+1}}{D_{\text{search}}} & \text{otherwise.} \end{cases} \tag{5}$$

Here, the source-detector shortest path distance is defined as $\psi$, and $D_{\text{search}}$ defines the largest Euclidean distance between vertices of the search area. The shortest path distance is essential for the non-convex environment and becomes the Euclidean distance when there is LOS. The normalization factor, $D_{\text{search}}$, in the negative reward provides an implicit boundary to the search area. This reward scheme incentivizes the DRL agent to find the source in the fewest actions possible as the negative reward is weighted more heavily. The reward magnitudes were selected so that standardization was not necessary during the training process as mean shifting of the reward can adversely affect training [19].

The reward function was designed to provide greater feedback for the quality of an action selected by the DRL agent in contrast to only constant rewards. For example, in the negative reward case, if the DRL agent initially takes actions that increase $D$ above the previous closest distance for several timesteps and then starts taking actions that reduce $D$, the negative reward will be reduced as it has started taking more productive actions. This distance-based reward function gives the DRL agent a more informative reward signal per episode during the learning process. Figure 3 shows an episode of the DRL agent operating within the environment, the radiation measurements it observes, and the reward signal it receives.




**(a)** Detector path. **(b)** Radiation measurements. **(c)** Cumulative reward.

**Figure 3.** Key data streams used by the DRL agent in training and inference. (**a**) shows the detector position at each timestep as it moves closer to the source. (**b**) shows the radiation counts measurements at each timestep corresponding with the detector position. (**c**) shows the cumulative reward signal that the DRL agent uses during training. The reward signal is only used for weight updates after all episodes in an epoch have been completed.

#### 2.1.4. Configuration

Detector step size was fixed at 1 m/sample and the movement direction in radians was limited to the set, $\mathcal{U} = \{i * \frac{\pi}{4} : i \in [0, 7]\}$. The DRL implementation can easily be

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

| Parameter | Value |
|---|---|
| Area Dimensions | $20 \times 20$ m |
| Src., det. initial positions | [-20, 20] m |
| Src. rate | $[1 \times 10^2, 1 \times 10^3]$ cps |
| Background rate | [10, 50] cps |
| State space | 11 |
| Action space | 8 |
| Max. search time | 120 samples |
| Velocity | 1 m/sample |
| Termination dist. | 1.1 m |
| Min. src.-det. initial dist. | 10 m |
| Number of obstructions | [1,5] |
| Obstruction dim. | [2,5] m |

**Table 1.** Radiation source simulation for convex and non-convex environment parameters. The brackets indicate an interval that was uniformly sampled on a per episode basis. Src. and det. are abbreviations for source and detector, respectively.

adapted to handle more discrete directions and variable step sizes or even continuous versions of these quantities. These two constraints were made to limit the computational requirements for the comparison algorithm. Maximum episode length was set at 120 samples to ensure ample opportunity for the policy to explore the environment, especially in the non-convex case. Episodes were considered completed if the detector came within 1.1 m of the source or a failure if the number of samples reached the maximum episode length. The termination distance was selected to cover a range of closest approaches as the detector movement directions and step size are fixed.

The state space has eleven dimensions that include eight detector-obstruction range measurements for each movement direction. This modeled some range sensing modality such as an ultrasonic or optical sensor. The maximum range was selected to be 1.1 m to allow the controller to sense obstructions within its movement step size. The range measurements were normalized to the interval $[0, 1]$, where 0 corresponds to no obstruction within range of the detector. If the policy selected an action that moved the detector within the boundaries of an obstruction, then the detector location was unchanged for that sample.

### *2.2. Proximal Policy Optimization (PPO)*

On policy, model-free DRL methods require that the agent learns a policy from its episodic experiences throughout training, whereas model-based methods focus on using a learned or given model to plan action selection. On policy methods are worse in terms of sample efficiency than Q-learning because learning takes place in an episodic fashion, i.e., the policy is updated on a per-episode basis. The benefit being that the agent directly optimizes policy parameters through the maximization of the reward signal. The decision to use model-free policy gradients was motivated by the stability and ease of hyperparameter tuning during training. Specifically, we used a variant of the *actor-critic* (A2C) framework called PPO. The actor, $\pi_\theta$, and critic, $V_\phi$, are the two main components of the A2C where $\theta, \phi$ denote neural network parameterizations. This will be covered in more detail in Section 2.3.

Schulman et al. propose the following *generalized advantage estimator* (GAE) with parameters $\gamma, \lambda$ to control the bias-variance tradeoff,

$$\hat{A}_n^{GAE(\gamma,\lambda)} = \sum_{n'=0}^{N-1} (\lambda\gamma)^{n'} \delta_{n+n'}, \tag{6}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

where $\delta$ is the temporal difference error as defined in [20].This is an exponentially-weighted average of the temporal differences error where $\gamma$ determines the scaling of the value function that adds bias when $\gamma < 1$ and $\lambda$ that adds bias when $\lambda < 1$ if the value function is inaccurate [21]. This leaves the final A2C gradient used in our algorithm as,

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_H[\sum_{n=0}^{N-1} \nabla_\theta \log \pi_\theta(a_{n+1}|h_n)\hat{A}_n^{GAE(\gamma,\lambda)}]. \tag{7}$$

The value function parameters are updated with stochastic gradient descent on the *mean square error* (MSE) loss between the value function estimate and the empirical returns,

$$\phi_k = \arg\min_\phi \mathbb{E}_{h_n,\hat{R}_n}[(V_\phi(h_n) - \hat{R}_n)^2]. \tag{8}$$

A common issue in policy gradient methods is the divergence or collapse of policy performance after a parameter update step. This can prevent the policy from ever converging to the desired behavior or result in high sample inefficiency as the policy rectifies the performance decrease. Schulman et al. proposed the PPO algorithm as a principled optimization procedure to ensure that each parameter update stays within a trust-region of the previous parameter iterate [22]. We chose to use the PPO-Clip implementation of the trust-region because of the strong performance across a variety of tasks, stability, and ease of hyperparameter tuning as shown in [22] and [23].

The PPO-Clip objective is formulated as,

$$\mathcal{L}(\theta_{k+1},\theta_k) = \mathbb{E}_H[\mathbb{E}_n[\min(r_n(\theta_{k+1},\theta_k)\hat{A}_n, \text{clip}(r_n(\theta_{k+1},\theta_k), 1-\epsilon, 1+\epsilon)\hat{A}_n)]]. \tag{9}$$

Here, $r_n(\theta_{k+1},\theta_k) = \frac{\pi_{\theta_{k+1}}(a_{n+1}|h_n)}{\pi_{\theta_k}(a_{n+1}|h_n)}$, denotes the probability ratio of the previous policy iterate to the proposed policy iterate and $\epsilon$ is the clipping parameter that enforces a hard bound on how much the latest policy iterate can change in probability space reducing the chance of a detrimental policy update. A further regularization trick is early-stopping based on the *approximate Kullback-Leibler divergence* (AKLD). The AKLD is a measure of the difference between two probability distributions and the approximation is the inverse of $r_n(\theta_{k+1},\theta_k)$ in log space. If the AKLD between the current and previous iterate over a batch of histories exceeds a user-defined threshold, then the parameter updates over that batch of histories are skipped.

### *2.3. RAD-A2C*

#### 2.3.1. Gated Recurrent Unit (GRU)

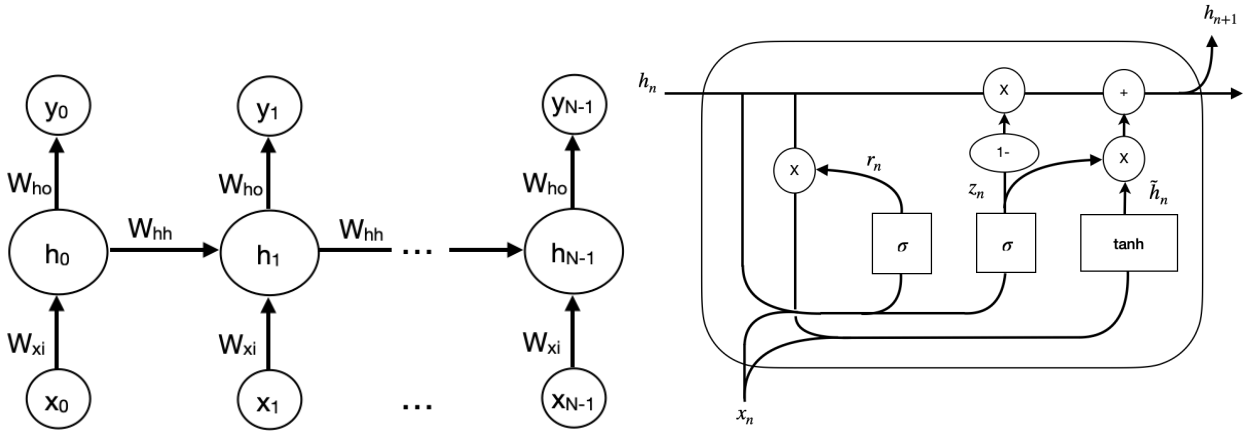
The GRU architecture proposed by Cho et al. is a subset of the *recurrent neural network* (RNN)s family that use gates to address the vanishing and exploding gradients encountered when using backpropagation-through-time and increase the network's ability to establish dependencies across long temporal gaps [24]. The following set of equations describe the GRU operations,

$$\begin{aligned} z_{n+1} &= \sigma(W_{xr}^T x_{n+1} + W_{hr}^T h_n + b_h), \\ r_{n+1} &= \sigma(W_{xz}^T x_{n+1} + W_{hz}^T h_n + b_h), \\ \tilde{h}_{n+1} &= \tanh(W_{xh}^T x_{n+1} + W_{hh}^T(r_{n+1} \odot h_n) + b_h), \\ h_{n+1} &= (1 - z_{n+1}) \odot h_n + z_{n+1}\tilde{h}_{n+1}, \end{aligned} \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid activation function and $\tanh(\cdot)$ is the hyperbolic tangent activation function. The GRU has more parameters than the standard RNN but the huge gain is in training stability and the increased range for sequence relationships.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

Figure 4 shows the design of a single GRU cell taken from Olah [25]. Each box represents a weight matrix and activation function and the circles represent mathematical operations. The conjoining lines represent the concatenation of the quantity and diverging lines represent the copying. The crux of the reset ($r_n$) and update ($z_n$) gates are to modify the candidate hidden state ($\tilde{h}_n$), which then becomes the output hidden state ($h_n$). The reset gate determines how much of the previous hidden state to factor into the new hidden state and the update gate determines the convex combination of the previous hidden state and the candidate hidden state. This cell is a drop-in replacement for the hidden state $h_n$ found in Figure 4a.




**(a)** RNN sequential inputs. **(b)** GRU architecture.

**Figure 4.** (**a**) shows the input flow for an RNN where each $x_n$ is fed to the network sequentially. The learned weight matrices $W_{ho}, W_{xi}, W_{hh}$ are the same across all sequence steps so the only changes are the input, output and hidden state. The $h_n$ represents the hidden state which is passed between sequence steps and is combined with the input to carry information across time. The output, $y_n$, is mapped from the hidden state. (**b**) shows the GRU architecture, a variation on the $h_n$ in (**a**). Each box represents a weight matrix and activation function and the circles represent mathematical operations. The conjoining lines represent concatenation of the quantity and diverging lines represent the copying. The crux of the reset ($r_n$) and update ($z_n$) gates are to modify the candidate hidden state ($\tilde{h}_n$) which then becomes the output hidden state ($h_n$) [25].

#### 2.3.2. Architecture

The RAD-A2C is composed of a *particle filter gated recurrent unit* (PFGRU) proposed by Ma et. al [26] (Appendix A.1), one GRU module to encode the inputs over time for action selection, and three linear layers. At each timestep, the observation is propagated to both the PFGRU and the A2C modules. The PFGRU uses a linear layer to regress its mean "particles" onto a source location, which is concatenated with the observation and fed into the A2C. The Actor layer regresses the GRU hidden state onto a multinomial distribution over actions using a softmax function. The Critic layer regresses the hidden state onto a value prediction. This value prediction is only necessary for the training phase and has no direct impact during inference. Figure 5 shows the RAD-A2C architecture and the flow of information through the system. The dotted lines indicate the path of the error gradients for backpropagation during training. Appendix A.2 covers implementation and training details and Table 1 shows the selected hyperparameters. The code is available at https://github.com/peproctor/radiation_ppo.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

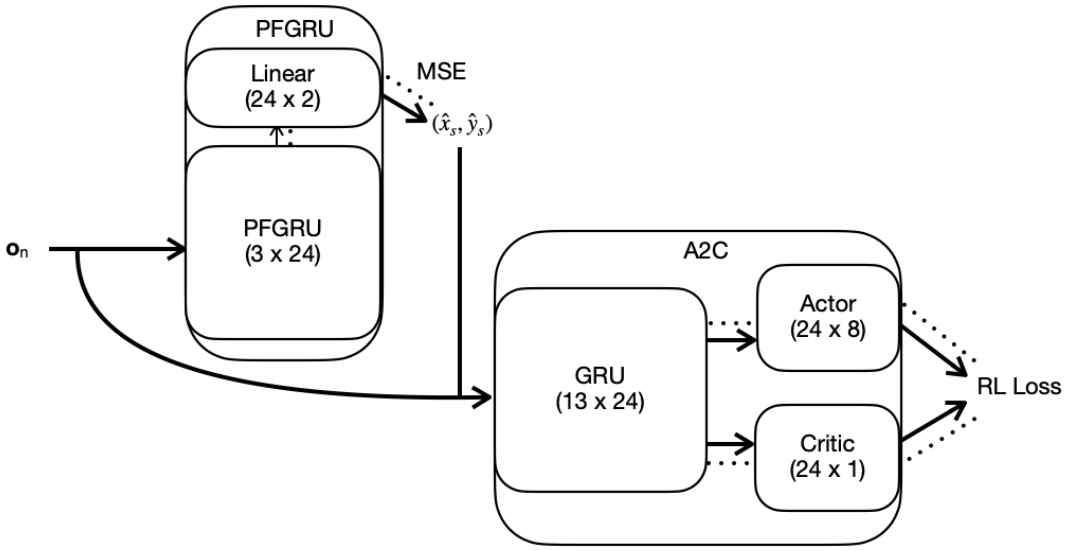



**Figure 5.** RAD-A2C source search architecture where quantities in the parenthesis denote the dimensions. The PFGRU provides a location prediction, denoted $(\hat{x}_s, \hat{y}_s)$, at each timestep, which is concatenated with the observation and fed into the A2C. The GRU module encodes the inputs over time in its hidden state and the Actor layer selects an action from this hidden state. The Critic layer predicts the expected return from the hidden state and is only needed during training. The dotted lines indicate the gradient flow during backpropagation.

The RAD-A2C is easily extendible to other source search scenarios such as a 3D environment, moving sources, using more advanced radiation transport simulators, and selection of detector step size and dwell time. These variations would only require a change in the dimensions of the input and output of the model, a potential increase in the hidden state size, and an appropriate update of the simulation environment/reward function. This is a major advantage of DRL as compared to human-specified algorithms. The downside of DRL is the long and computationally intense training costs and sensitivity to hyperparameters. A weakness of our RAD-A2C implementation is that the source intensity is not predicted by the PFGRU as this would require prior knowledge about the upper limit of the intensity. We opted for scenario generalization by performing search without a source intensity estimate. While source intensity is of interest in radiation source localization scenarios, an additional estimator such as least squares fitting could be used in conjunction with our model for this end.

### *2.4. Evaluation*

Appendix B details the information-driven control method (RID-FIM) and the *gradient search* (GS) algorithms used as comparison against our method. All search methods were evaluated across a range of SNRs in the convex environment. Only the RAD-A2C and GS were compared in the non-convex environment as the *bootstrap particle filter* (BPF) measurement and process model do not account for obstructions. The SNRs were broadly grouped into "low" $(1.0 - 1.2)$, "medium" $(1.2 - 1.6)$, and "high" $(1.6 - 2.0)$ intervals. For each SNR, $1,000$ different environments were uniformly randomly sampled to create a fixed test. Monte Carlo simulations were performed for all experiments to determine the average performance of the algorithms. Each algorithm was run for 100 episodes per environment.

#### 2.4.1. Metrics

Weighted median completed episode length and median percent of completed episodes served as the main performance metrics. The weighted median was used for the completed episode length with a weighting factor between $1 - 100$, determined by the number of Monte Carlo simulations that were completed by the agent per environment. The completed episode length corresponds to the number of radiation

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

measurements required to come within the episode termination distance of the source before the maximum episode length is reached. This quantifies the agent's effectiveness in incorporating the measurements to inform exploration of the search area. Percent of episodes completed is the more important metric as the priority in radiation source search is mission completion and this works in tandem with the completed episode length to characterize the agent's performance. An ideal agent would have a low median episode length and a high median percent of episodes completed.

#### 2.4.2. Experiments

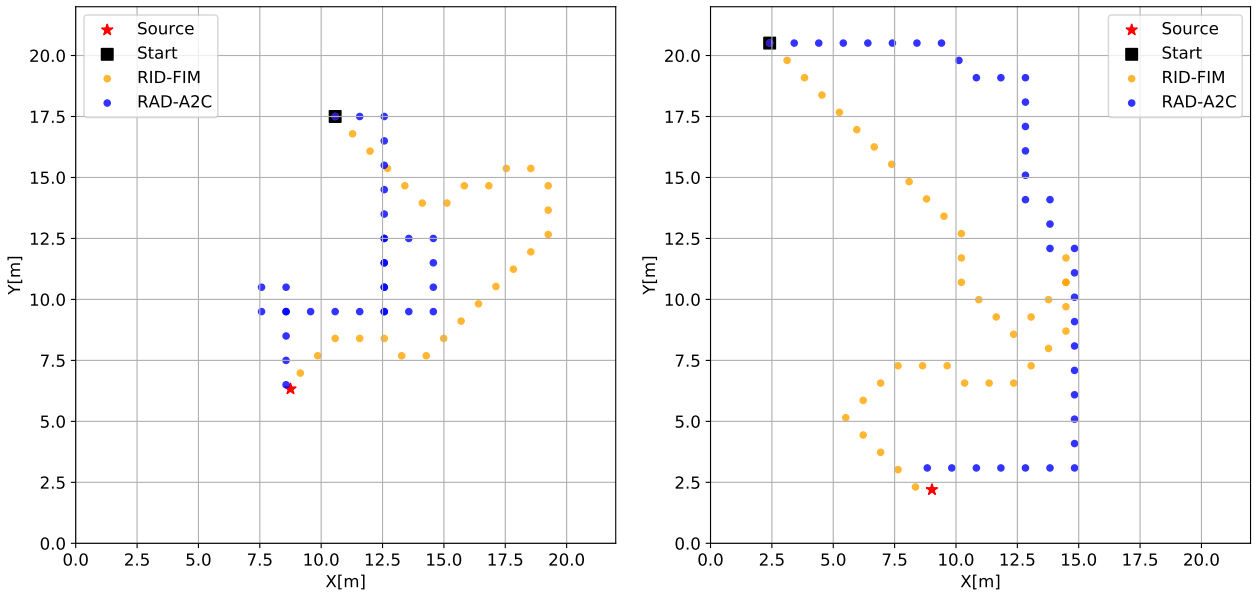
Three sets of experiments were run in the radiation source search environment to assess the performance characteristics of our proposed RAD-A2C architecture. The first experiment focused on the comparison of all of the search algorithms. The second experiment assessed the RID-FIM and A2C action selection quality with BPF performance as a proxy. The final experiment looked at the performance of the GS and RAD-A2C in a non-convex environment where the number of obstructions was varied.

## 3. Results

### *3.1. Convex Environment*

#### 3.1.1. Detector Path Examples

Two detector paths for the RAD-A2C and the RID-FIM in two different SNR configurations of the convex environment are shown in Figures 6a, 6b. The source prediction marker was omitted to reduce clutter. Both algorithms must explore the area as they search for radiation signal above the noise floor. In the high SNR configuration, both algorithms make sub-optimal decisions that move the detector away from the source, a result of the probabilistic nature of the measurement process. However, they both quickly adjust and successfully find the radiation source. The detector starts much further from the source in the low SNR configuration and the controllers select many more actions before picking up any signal. In both scenarios, the RID-FIM makes more diagonal movements relative to the RAD-A2C.



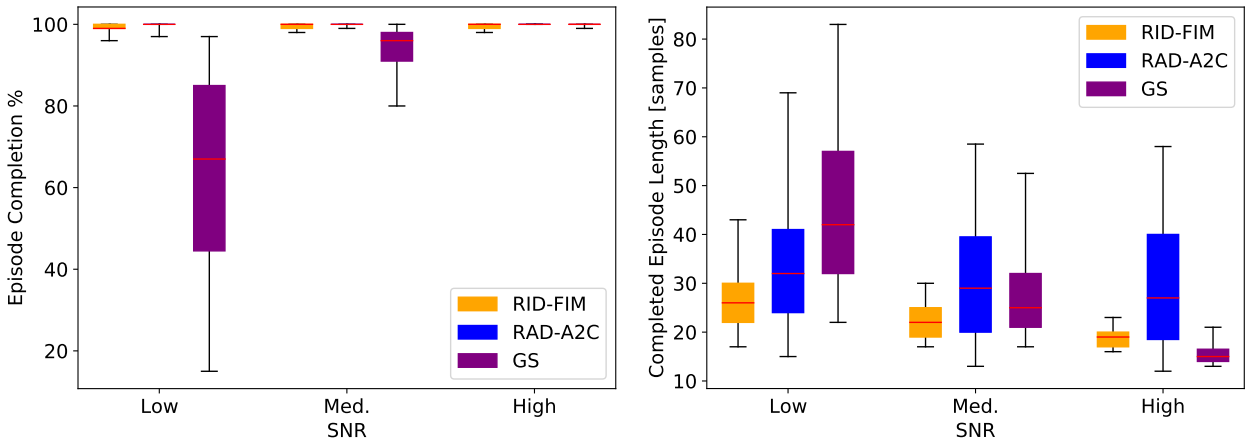

**(a)** High SNR configuration. **(b)** Low SNR configuration.

**Figure 6.** Two detector paths for the RAD-A2C and the RID-FIM in high and low SNR configurations of the convex environment overlayed on a single plot. The black square denotes the detector starting position and the red star represents the radiation source. (**a**) shows a low SNR configuration and (**b**) shows a high SNR configuration. In both cases, the stochastic nature of gamma radiation measurement results in the control algorithms taking sub-optimal actions before the source could be located.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

### 3.2. Performance

Box plots for the completed episode percentage and completed episode length for all methods in the convex environment are found in Figures 7a and 7b, respectively. The median is denoted in red, the boxes range from the first to the third quartile and the whiskers extend to the 2.5$^{th}$ and 97.5$^{th}$ percentiles. GS achieved the shortest episode completion length for all experiments at high SNR but performance decreased swiftly at the lower SNR levels. The RID-FIM had a consistent performance with tight boxes for both metrics at all SNRs. The RAD-A2C was the only algorithm to maintain 100% completion for all SNRs with the tradeoff being the longest median episode length for all but one of the SNRs. Figure 8 shows the relationship between median episode length to median episode completion. Top-performing search algorithms are located on the far right of the plot and ideally near the bottom.




**(a)** Completed episode percentage. **(b)** Completed episode length.

**Figure 7.** Box plots for the completed episode percentage and completed episode length against SNR in the convex environment. The median is denoted in red, the boxes range from the first to the third quartile and the whiskers extend to the 2.5$^{th}$ and 97.5$^{th}$ percentiles. Figure 7b shows the RID-FIM consistently found the source in a short amount of time even as SNR decreased. Figure 7a shows the RAD-A2C was the only method that completed 100% of the episodes. GS performance sharply declined for lower SNRs.

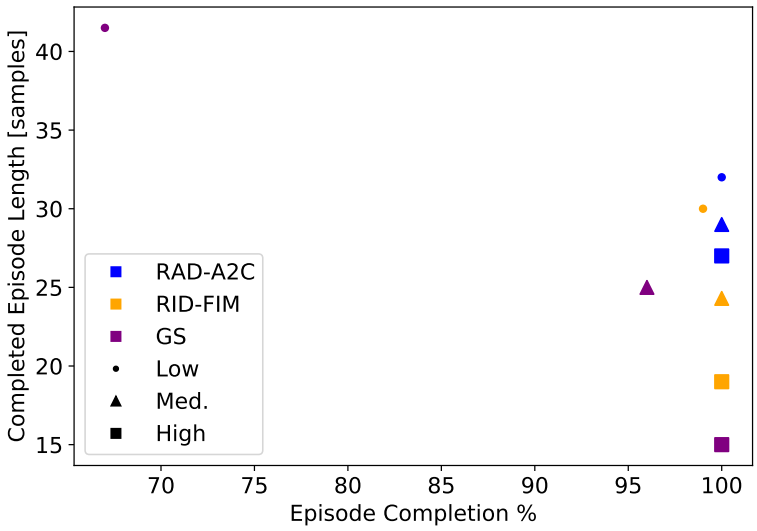



**Figure 8.** Median completed episode length against median completion rate. The marker shapes denote the SNR level and the color denotes the search method. An ideal search algorithm would be located in the bottom right of the plot for all the SNRs. GS has the best performance at high SNR due to the strong radiation field but also uses seven more measurements per action selection than the other methods.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

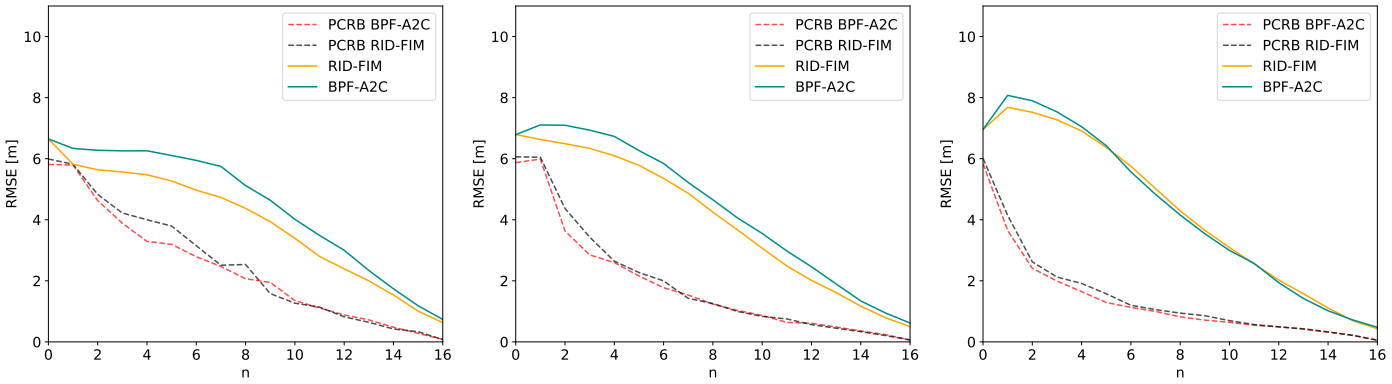
### 3.3. BPF Comparison

The RID-FIM and A2C controller are compared directly by replacing the PFGRU in the RAD-A2C with the BPF. This new system will be denoted as BPF-A2C in the following plots. Swapping in the BPF for the PFGRU facilitates in-depth analysis of the controllers through the lens of the BPF performance. The estimator performance depends entirely on the quality of action selection throughout an episode as this determines what information the estimates will be based on. Thus, we compare the RMSE for the Euclidean distance between the actual and predicted source location at each timestep for three different episode completion lengths across SNR.

Figures 9, 10, and 11, show the RMSE and *posterior Cramér-Rao lower bound* (PCRB) for the RID-FIM and the BPF-A2C for three different completed episode lengths across SNRs. The PCRB serves as a proxy for the sub-optimality of the controllers because of the use of the same estimator (see Appendix B.5). Each plot is averaged over at least 200 different episodes and at least 700 total runs. An episode was only considered for this analysis if the completed episode length was the same for both algorithms in the set of the Monte Carlo runs for that episode. This ensured that RMSEs and PCRBs were only averaged over the same set of episodes.

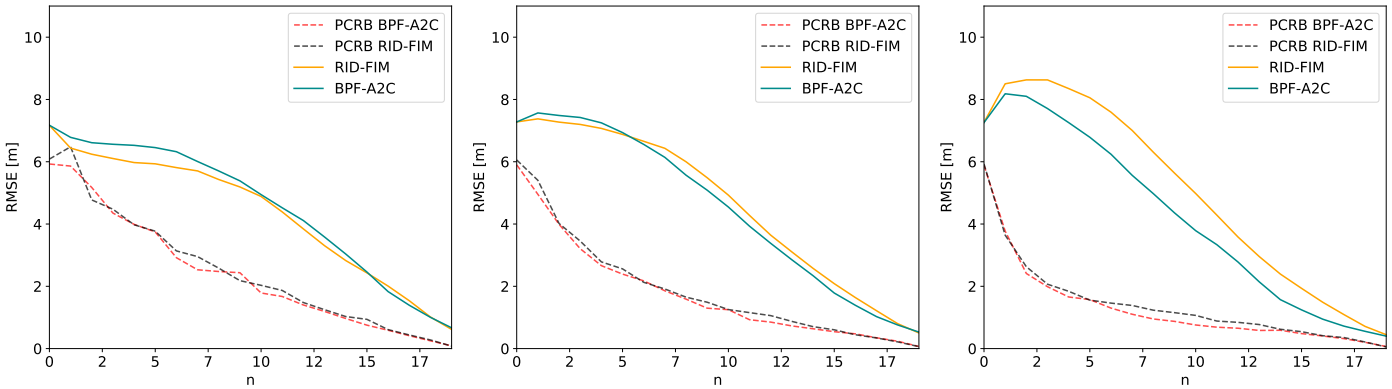
These specific completed episode lengths were chosen to highlight a variation in estimator performance that was observed across completed episode lengths ranging from $10 - 60$ samples and SNR levels. The RMSE for the RID-FIM is lower or equal to the BPF-A2C at a completed episode length of 17 across SNR. This changes for a completed episode length of 20 where the RID-FIM RMSE is only lower than the BPF-A2C at the lowest SNR. For the completed episode length of 28, the BPF-A2C now has a lower RMSE than the RID-FIM for all SNRs. In all of the plots, the PCRB for the BPF-A2C is slightly lower or equal to the PCRB for the RID-FIM. The PCRB decreases at a faster rate for the high SNR compared to the low SNR. Estimator RMSE consistently approaches the PCRB by the end of an episode. The RMSE initially increased for the high SNR in direct relation with the completed episode length in all the RMSE plots shown.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1



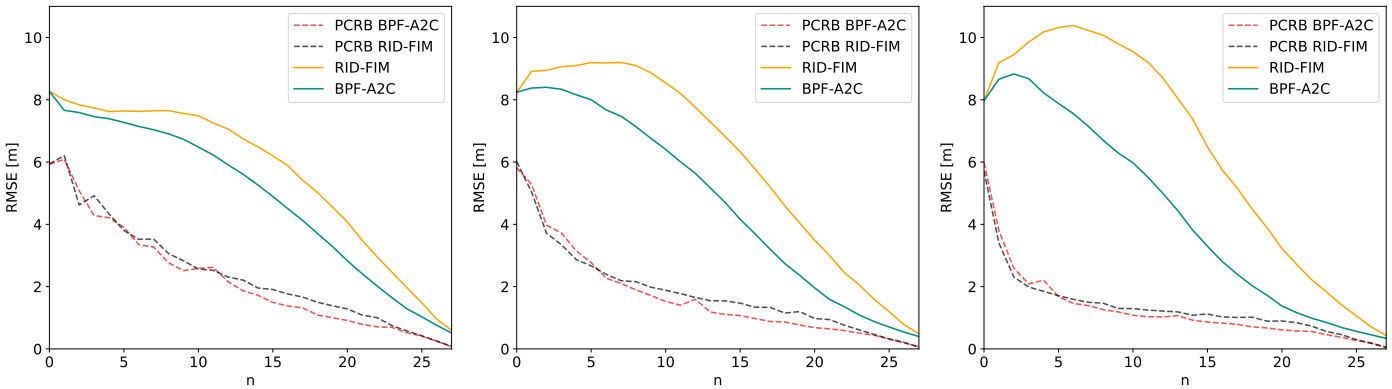

**(a)** Low. **(b)** Medium. **(c)** High.

**Figure 9.** Comparison of the Monte Carlo RMSE for BPF estimation of the source location at each timestep for a completed episode length of 17. Each plot contains the BPF PCRB and RMSE for the RID-FIM and BPF-A2C controllers averaged over at least 200 different episodes. (**a**) is at low SNR, (**b**) is at medium SNR, and (**c**) is at high SNR. The RID-FIM has a lower RMSE than the BPF-A2C for the low and medium SNR but the RID-FIM's action selection was solely dependent on potentially spurious BPF state estimates, which caused the BPF-A2C to match the RID-FIM performance at the high SNR.




**(a)** Low. **(b)** Medium. **(c)** High.

**Figure 10.** Comparison of the Monte Carlo RMSE for BPF estimation of the source location at each timestep for a completed episode length of 20. Each plot contains the BPF PCRB and RMSE for the RID-FIM and BPF-A2C controllers averaged over at least 400 different episodes. (**a**) is at low SNR, (**b**) is at medium SNR, and (**c**) is at high SNR. The RID-FIM has a lower RMSE than the BPF-A2C for the low SNR but the RID-FIM's action selection was solely dependent on potentially spurious BPF state estimates, which caused the BPF-A2C to outperform the RID-FIM at medium and high.




**(a)** Low. **(b)** Medium. **(c)** High.

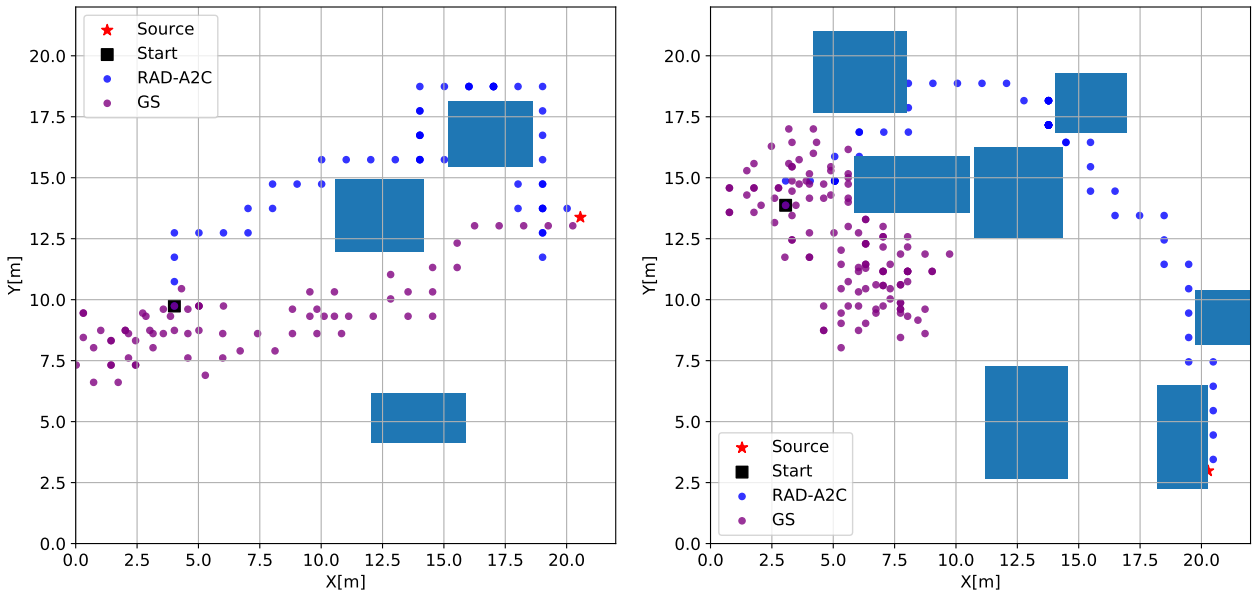
**Figure 11.** Comparison of the Monte Carlo RMSE for BPF estimation of the source location at each timestep for a completed episode length of 28. Each plot contains the BPF PCRB and RMSE for the RID-FIM and BPF-A2C controllers averaged over at least 650 different episodes. (**a**) is at low SNR, (**b**) is at medium SNR, and (**c**) is at high SNR. The BPF-A2C has a lower RMSE than then RID-FIM when the completed episode length was longer due to the RID-FIM's action selection dependence on potentially spurious BPF state estimates.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

### 3.4. Non-convex Environment

#### 3.4.1. Detector Path Examples

Two detector paths for the RAD-A2C and the GS in two non-convex environments with three and seven obstructions are shown in Figures 12a and 12b, respectively. The source prediction marker was omitted to reduce clutter. The GS takes many more samples to find a radiation gradient in the three obstruction environment but eventually finds the source. Gradient information is extremely sparse in the seven obstruction environment and thus the GS only moves randomly. The RAD-A2C can avoid the obstructions and find the source in both situations, even moving diagonally between two obstructions in Figure 12b. As in the convex environment, the majority of the RAD-A2C movements are in the cardinal directions.




**(a)** Three obstructions. **(b)** Seven obstructions.

**Figure 12.** Two detector paths for the RAD-A2C and the GS overlayed on a single plot for the non-convex environment. (**a**) shows the three obstruction environment and (**b**) shows the seven obstruction environment. The black square denotes the detector starting position, the blue rectangles represent obstructions that block radiation propagation, and the red star is the radiation source. Both algorithms must explore the area as they search for radiation signal above the noise floor.

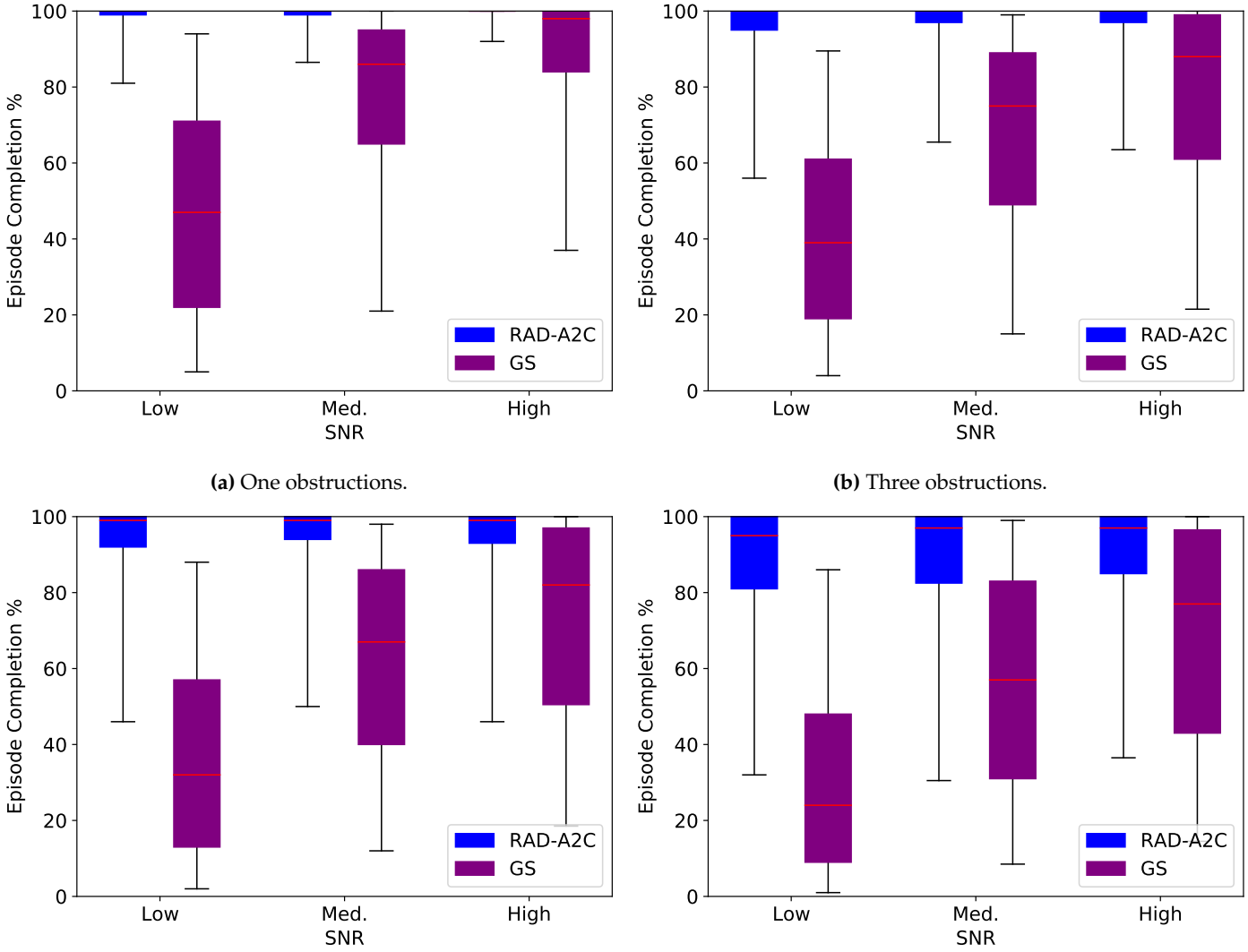
### 3.5. Performance

Box plots for the episode completion percentage and completed episode length against SNR for both methods in the non-convex environment are found in Figures 13 and 14, respectively. Figures 13a and 14a are results with one obstruction, Figures 13b and 14b are results with three obstructions, Figures 13c and 14c are results with five obstructions, and Figures 13d and 14d are results with seven obstructions. The median is denoted in red, the boxes range from the first to the third quartile and the whiskers extend to the $2.5^{\text{th}}$ and $97.5^{\text{th}}$ percentiles.

Across obstruction number, the RAD-A2C maintains above 95% episode completion even at low SNR. The distribution of the RAD-A2C episode completion gets larger as the number of obstructions increases. GS has $>$ 85% episode completion when there are less than 7 obstructions at high SNR but sees a sharp decrease in performance as the SNR level decreases. Even at high SNR, GS only completes 77% of episodes when 7 obstructions are present. GS also has significant spread in the first and third quartile for most of the completed episode non-convex experiments. The RAD-A2C median for completed episode length increases by approximately 10 samples from a single

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

obstruction to seven obstructions. The first and third quartiles for completed episode length also increase as the number of obstructions increase.




**(a)** One obstructions.

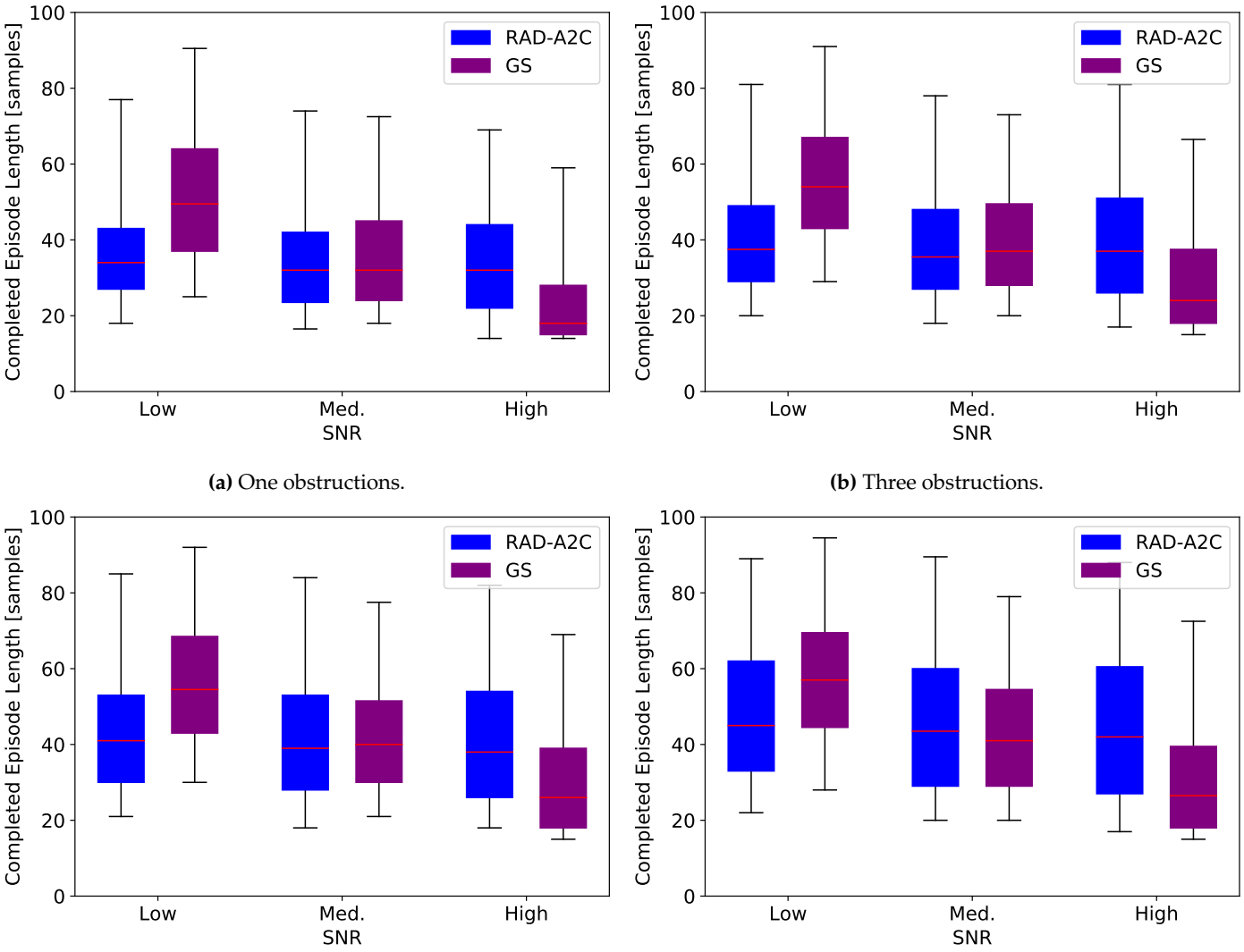
**(b)** Three obstructions.

**(c)** Five obstructions.

**(d)** Seven obstructions.

**Figure 13.** Box plots for the completed episode percentage against SNR in the non-convex environment, where each plot corresponds to a different number of obstructions in the environment. The median is denoted in red, the boxes range from the first to the third quartile and the whiskers extend to the $2.5^{th}$ and $97.5^{th}$ percentiles. (**a**) was for a single obstruction, (**b**) was for three obstructions, (**c**) was for five obstructions, and (**d** ) was for seven obstructions. GS episode completion deteriorates with increasing number of obstructions while the RAD-A2C maintains greater than 95% median episode completion.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1




**(a)** One obstructions.

**(b)** Three obstructions.

**(c)** Five obstructions.

**(d)** Seven obstructions.

**Figure 14.** Box plots for the completed episode length against SNR in the non-convex environment, where each plot corresponds to a different number of obstructions in the environment. The median is denoted in red, the boxes range from the first to the third quartile and the whiskers extend to the 2.5$^{th}$ and 97.5$^{th}$ percentiles. (**a**) was for a single obstruction, (**b**) was for three obstructions, (**c**) was for five obstructions, and (**d**) was for seven obstructions. The RAD-A2C maintains a low completed episode length across the varying number of obstructions and SNR while GS performance deteriorates.

## 4. Discussion

### *4.1. Convex Environment*

The results indicate close search performance between the RID-FIM and RAD-A2C algorithms in the convex environment. GS had the shortest episode completion length at high SNR but this required 7 more measurements per action selection. The RAD-A2C showed the best reliability in completing all of the episodes with a minimal spread in the distribution of results but had a greater spread in the completed episode length even at the highest SNR. The longer completed episode length of the RAD-A2C could be due to learned behavior that is advantageous in non-convex environments as the training environment always had obstructions present. The RID-FIM had a tighter and lower distribution of completed episode lengths across the SNRs.

Completion of episodes is the priority in practice as this will eliminate the threat of human harm from nuclear materials. Both algorithms get the job done effectively, however, the RID-FIM has a slightly greater chance of failing when SNR conditions

 doi:10.20944/preprints202108.0018.v1

are poor compared with the RAD-A2C. The RID-FIM utilized perfect knowledge of the background rate, which is a reasonable assumption in this particular source search context, however, its performance is likely to be degraded to some degree when it must also estimate an unknown background rate. The RAD-A2C did not receive the true episode background rate directly but did have prior exposure to the interval of background rates through training. Additionally, the RAD-A2C input standardization filters the radiation measurement inputs (see Appendix A.2).

### *4.2. BPF Comparison*

The BPF serves as an interesting comparison point between the A2C and RID-FIM controllers. When the completed episode length was short ($< 16$ samples), the RID-FIM location prediction RMSE was lower than the BPF-A2C and closer to the PCRB at all SNRs. This evidences the effectiveness of information-driven search schemes and the near-optimal performance of the RID-FIM when the BPF does not make spurious estimates. However, the occurrence of the intersection point of the RMSE curves highlights the disadvantage of the RID-FIM's reliance on the estimator for action selection. If early state estimates are incorrect, this leads the RID-FIM to take more sub-optimal actions until the estimate is corrected. This is evidenced by the longer completed episode lengths $(20, 28)$ that have a greater initial increase in the RMSE as seen in Figures 10c and 11c. Interestingly, the higher SNR contributes a sharper increase, likely due to the strong radiation measurements being interpreted by the BPF as evidence for the incorrect estimate.

In contrast, the A2C module of the BPF-A2C selects its actions from the location prediction and the measurement directly. Thus, when the SNR is high, the RMSE intersection point occurs at an earlier completed episode length (17 samples) because the A2C factors in measurement information at each timestep, rather than strictly following the possibly incorrect location prediction as the RID-FIM must do. This also explains why the BPF-A2C has lower RMSE at longer completed episode lengths as seen in Figure 11. The intersection point occurred at longer completed episode lengths for lower SNR because it takes the A2C longer to come across informative measurements that can correct the spurious BPF state estimates.

### *4.3. Non-convex Environment*

The results showcase the strong performance of the RAD-A2C in the non-convex environment. Surprisingly, the episode completion percentage did not decrease substantially in the seven obstruction configuration and the median completed episode length did not increase drastically. This demonstrates the algorithm's ability to generalize as it was only trained on up to five obstructions per environment. The RAD-A2C is not simply a gradient search algorithm as the non-convex environment has many areas with no gradient information as evidenced by the ineffectiveness of the GS. Overall, these results support our hypothesis that the RAD-A2C is an effective search algorithm for both convex and non-convex environments.

## 5. Conclusions and Future Work

This paper investigated the efficacy of PPO and our proposed DRL architecture, the RAD-A2C, for a convex and non-convex radiation source search through comparison against the RID-FIM and GS across SNR. The GS had strong performance when the SNR was high but quickly lost efficacy with decreasing SNR. The RID-FIM typically required fewer measurements to complete episodes but had a slightly greater chance of not completing all of the episodes at lower SNRs. The RAD-A2C consistently completed all episodes albeit at the cost of taking more measurements. Guaranteed episode completion is arguably the most important criteria for radiation source search applications.

Estimator performance served as another lens to compare the controller performance directly. The same BPF was used for both controllers (RID-FIM, A2C) so that

the RMSE and PCRB for the location prediction could be compared. We found that on average, the BPF RMSE was lower for the longer episode lengths when the A2C was the controller as it was able to factor in measurements to its action selection, as opposed to the RID-FIM which selected actions solely on the BPF location prediction. The RID-FIM's action selection scheme is well-motivated but is susceptible to incorrect state estimates from the estimator.

In the non-convex environment, the RAD-A2C completed greater than 95% of episodes over a range of obstructions and SNRs. There was very little gradient information available in the environments with more obstructions and thus the GS algorithm completed a much lower percentage of episodes. The RAD-A2C demonstrated generalizability as it was able to maintain a high completion percentage in a seven obstruction environment that it had never been trained on.

As mentioned in Section 2.3, the RAD-A2C formulation has the potential to be applied to other variations of the radiation source search. These include moving and/or shielded nuclear sources, spatially varying background rates, utilizing an attenuation model for different environment materials, locating an unknown number of multiple sources, and a larger, more complex urban environment such as the one used by Hite et al. [6]. A classification layer could also be added to the A2C module that is trained on detecting whether a source is present or not and how many sources are present. Noise could be added to the other dimensions of the observation vector such as the detector coordinates and/or the obstruction range measurements. In theory, the majority of these cases only require modification of the simulation environment, clever shaping of the reward signal, and hyperparameter sweeps to determine the model parameters.

Our proposed algorithm could be trained in a more realistic environment and gamma sensor simulation such as the one utilized for a single UAV source search by Baca et al. [27]. The authors developed a realistic gamma radiation simulation plugin for the Gazebo/ROS environment. Gazebo is a realistic open-source robotics simulator [28]. This plugin could then be easily interfaced with our DRL algorithm using the OpenAI_ROS Gym developed by Ezquerro et al. that seamlessly connects Gazebo and OpenAI Gym interfaces [29].

**Author Contributions:** "Conceptualization, P.P., C.T., A.H. and M.O.; methodology, P.P.; software, P.P.; validation, P.P.; formal analysis, P.P.; investigation, P.P.; resources, C.T. and M.O.; data curation, P.P.; writing—original draft preparation, P.P.; writing—review and editing, P.P., C.T., A.H. and M.O.; visualization, P.P.; supervision, C.T. and A.H.; project administration, C.T.; funding acquisition, C.T. and M. O.

**Funding:** This study was supported by the Defense Threat Reduction Agency under the grant HDTRA1-18-1-0009

**Acknowledgments:** The authors would like to thank their colleague Merlin Carson for his support throughout this project. They would also like to thank Ren Cooper and Tenzing Joshi of Lawrence Berkeley National Laboratory along with Jason Hite from Oak Ridge National Laboratory for their correspondence and helpful suggestions.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

| | |
|---|---|
| A2C | Actor-critic |
| BPF | Bootstrap particle filter |
| BPF-A2C | Bootstrap particle filter and actor-critic |
| CRB | Cramér-Rao lower bound |
| DRL | Deep reinforcement learning |
| DL | Deep learning |
| FIM | Fisher information matrix |
| GRU | Gated recurrent unit |
| GS | Gradient search |
| LOS | Line-of-sight |
| NLOS | No line-of-sight |
| ML | Machine learning |
| PCRB | Posterior Cramér-Rao lower bound |
| PFGRU | Particle filter gated recurrent unit |
| PPO | Proximal policy optimization |
| RAD-A2C | Our proposed actor-critic architecture |
| RNN | Recurrent neural network |
| RID | Rényi information divergence |
| RID-FIM | Hybrid information-driven controller that uses RID and FIM |
| RL | Reinforcement learning |
| SNR | Signal-to-noise ratio |

## Appendix A. RAD-A2C

### *Appendix A.1. Particle Filter Gated Recurrent Unit (PFGRU)*

The PFGRU is an embedding of the BPF into a GRU architecture proposed by Ma et al [26]. As in the BPF, there are a set of particles and weights used for filtering and prediction of the posterior state distribution. In the case of the PFGRU, the particles are represented by the set of hidden or latent state vectors, $\{h_n^i\}_{i=1}^{N_{gp}}$. The latent states are propagated and the weights updated at each timestep by a learned transition and measurement function denoted as,

$$\begin{aligned} h_{n+1}^i &= f_{tr}(h_n^i, \zeta_{n+1}^i) \\ y_{n+1}^i &= f_{out}(h_{n+1}^i), \end{aligned} \tag{A1}$$

where $\zeta_n^i \sim p(\zeta_{n+1}^i | h_{n+1}^i)$ is a learned noise term akin to the process noise in the BPF. The weight update also relies on a learned likelihood function,

$$w_{n+1}^i = \eta f_{obs}(y_{n+1}, h_{n+1}^i) w_n^i, \tag{A2}$$

where $\eta$ is a normalization factor.

The PFGRU utilizes a soft resampling scheme to combat particle degeneracy while maintaining model differentiability. This is achieved by sampling particle indices from a multinomial distribution with probabilities determined by a convex combination of a uniform distribution and the particle weight distribution. The new weights are then determined by,

$$w_{n+1}^{'} = \frac{w_{n+1}^{a_n^j}}{\alpha w_{n+1}^{a_{n+1}^j} + (1-\alpha)(1/N_{gp})}, \tag{A3}$$

where $\alpha$ is the mixture coefficient parameter. The loss function consists of two components to capture the important facets of state space tracking. The first component is the mean squared loss between the mean particle and the predicted quantity. The second component is the *evidence lower bound* (ELBO) loss that measures the difference in distribution of the particle distribution relative to the observation likelihood, for more details see [26]. The total loss is expressed as,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

$$\mathcal{L}(\theta) = \mathcal{L}_{MSE} + \beta * \mathcal{L}_{ELBO}, \tag{A4}$$

where $\beta$ is a weighting parameter determined by the user. Figure 1 shows the PFGRU architecture.

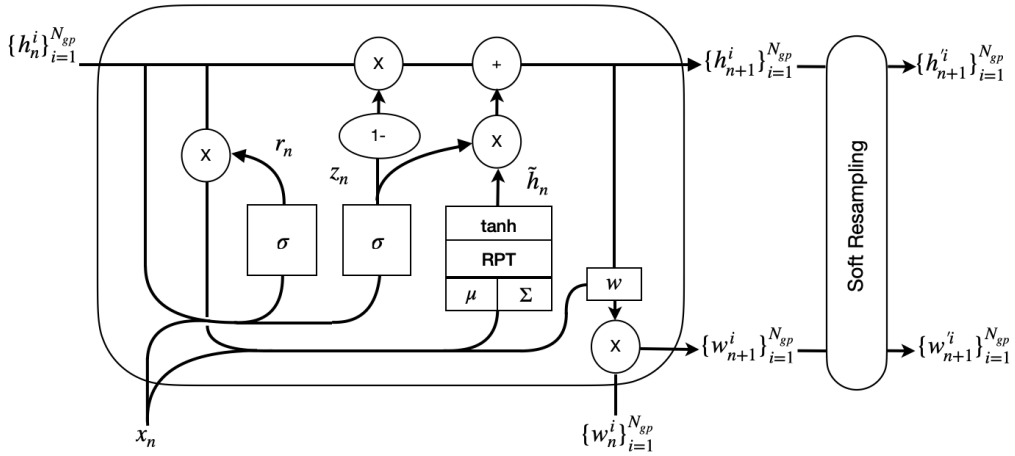



**Figure 1.** PFGRU Architecture. The hidden state $h_n^i$ and weights $w_n^i$ are elements of a set of size $N_{gp}$. Each box represents a weight matrix and activation function and the circles represent mathematical operations. The conjoining lines represent concatenation of the quantity and diverging lines represent the copying. The crux of the reset ($r_n$) and update ($z_n$) gates are to modify the candidate hidden state ($\tilde{h}_n$) which then becomes the output hidden state ($h_n$). The hidden state and weights are resampled using a soft-resampling scheme at each timestep to preserve differentiability. Recreated from [26].

### *A.2. Training*

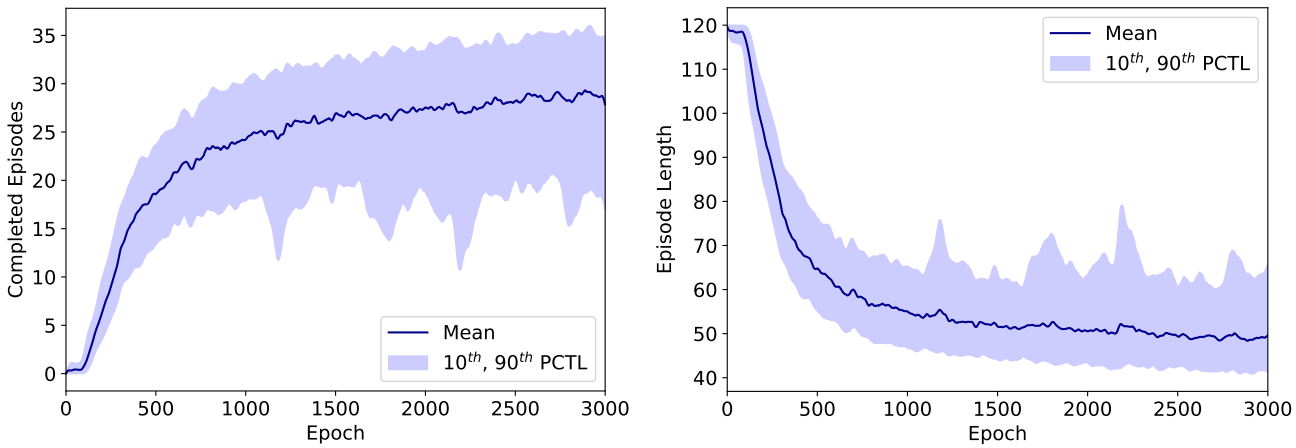
The estimate of the gradient iterate (Eq. 7) is improved by increasing the number of histories being averaged over. Schulman et al. improved training scalability by instantiating copies of the DRL agent and environment on different CPU cores to parallelize episode collection [22]. Each DRL agent computes its parameter gradients after all episodes for an epoch have been collected. The gradients are then averaged across all the cores and a weight update is performed per core. An important distinction in the implementation used here is the environment variation across the CPU cores. All of the sampled quantities were different per core and fixed per epoch resulting in a more generalized policy. This is because the averaged gradient step will be in the direction that improves performance across a diverse set of environments. Tobin et al. proposed a similar idea called domain randomization that aimed to bridge the gap between DRL simulators and reality by introducing extra variability into the simulator [15]. Table 1 shows the hyperparameters that resulted in the strongest performance for the DRL agent from the parameter sweep. The total training time for a single DRL agent running on 10 cores took approximately 26 hours.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

| Parameter | Value |
|---|---|
| Epochs | 3,000 |
| Episodes per epoch | 4 |
| Num. cores | 10 |
| Tot. weights & biases | 7,443 |
| GRU hidden size | 24 |
| PFGRU hidden size | 24 |
| PFGRU particles | 40 |
| Learning Rate A2C | $3 \times 10^{-4}$ |
| Learning Rate PFGRU | $5 \times 10^{-3}$ |
| Optimizer | Adam |
| $(\gamma,\lambda,\eta)$ | $(0.99, 0.9, 0.105)$ |

**Table 1.** Hyperparameter values with the strongest performance for the DRL agent from our parameter sweep. The parameters $\gamma$ and $\lambda$ are used in the generalized advantage estimator [21]. The parameter $\eta$ is the maximum value for the approximate Kullback-Leibler divergence before weight updates are terminated for the epoch.

The RAD-A2C was trained eight separate times with eight different random seeds to assess model stability. In seven of the eight models, the RAD-A2C achieved performance that was consistent with the model we used for the assessment in this paper. This is evidenced by the training curves in Figure 2 that show the average number of completed episodes and the average episode length over the 10 parallelized environments per epoch. The dark blue line represents the smoothed mean and the shaded region represents the smoothed $10^{th}$ and $90^{th}$ percentiles over the eight random seeds. The maximum possible number of completed episodes per epoch was 40. The one model that did not converge as well as the others showed oscillations in the performance curves indicating that a parameter update resulted in an adverse policy change. Training for more than 3000 epochs did not significantly improve performance.




**(a)** Average completed episodes. **(b)** Average episode length.

**Figure 2.** Performance curves during the training process for the RAD-A2C over eight random seeds. (**a**) shows the number of completed episodes and (**b**) shows the episode length averaged over the 10 parallelized environments per epoch. The dark blue line represents the smoothed mean and the shaded region represents the smoothed $10^{th}$ and $90^{th}$ percentiles over the eight random seeds. Episode length decreases and number of completed episodes increases as the model converges to a useful policy. Training for more than 3000 epochs did not significantly improve performance.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

A.2.1. Standardization

A common technique in DL is to standardize the input data to increase training stability and speed. This is done by subtracting the mean and dividing by the standard deviation per feature across a batch of input data. The DRL context does not have easy access to the full data statistics since it is collected and processed online. We used a technique proposed by Welford for estimating a running sample mean and variance as follows [30],

$$\begin{aligned} \mu_{n+1} &= \mu_n + \frac{(o_{n+1} - \mu_n)}{n} \\ S_{n+1} &= S_n + (o_{n+1} - \mu_n)(o_{n+1} - \mu_{n+1}) \\ \sigma^2_{n+1} &= \frac{S_{n+1}}{n}, \end{aligned} \tag{5}$$

where $\mu_0 = o_0$, $S_0 = 0$. The statistics were updated after each new observation and then standardization was performed.

**B. Information-driven Controller**

Information-driven search is an information-theoretic framework for sequential action selection. This framework endows the controller with the ability to update its path plan as new observations become available as opposed to relying only on whether the target has been detected or not [31]. Information is integrated across time by tracking the posterior probability density of states of interest. This can quickly become computationally prohibitive and so heuristic methods such as the *bootstrap particle filter* (BPF) are employed.

*B.1. Bootstrap Particle Filter (BPF)*

The BPF is typically used to track a dynamic process over time. It has been proven that an optimal estimate of the state can be recovered from the posterior state distribution, however, it is often computationally intractable to track when the state dimension is high [32]. Thus, methods such as the BPF attempt to approximate the posterior state through a set of samples, $\{x^i_n, w^i_n\}^{N_p}_{i=1}$, often referred to as particles and weights, respectively. This leads to the approximation,

$$P(x_{n+1}|y_{0:n+1}) \approx \sum_{i=1}^{N_p} w^i_{n+1} \delta(x_{n+1} - x^i_{n+1}), \tag{6}$$

where $P(x_{n+1}|y_{0:n+1})$ is the marginal posterior, $w^i_{n+1}$ is the $i^{th}$ particle weight, $x^i_{n+1}$ is the $i^{th}$ particle state, $\delta(\cdot)$ is the Dirac Delta function, and $N_p$ is the number of particles. At each timestep, the particles are propagated through the process model and a measurement prediction is generated with the measurement model. The particle weights are calculated recursively as,

$$w^i_{n+1} \propto \frac{p(y_{n+1}|x^i_{n+1}) p(x^i_{n+1}|x^i_n)}{q(x^i_{n+1}|x^i_n, y_{n+1})} w^i_n, \tag{7}$$

where $p(y_{n+1}|x^i_{n+1})$ is the measurement likelihood, $p(x^i_{n+1}|x^i_n)$ is the transition density, and $q(x^i_{n+1}|x^i_n, y_{n+1})$ is an importance density [32]. Particles are drawn from a user-specified importance density $q_x$. In our implementation, the importance density is set equal to the prior distribution to reduce the weight update step to the measurement likelihood and the previous weight:

$$w^i_{n+1} \propto p(y_{n+1}|x^i_{n+1}) w^i_n. \tag{8}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

If a particle has a low probability for a given measurement, this effectively removes the particle's contribution to the estimated posterior which can adversely affect state estimation over the trajectory and is known as the degeneracy problem. Particle degeneracy can be tracked by the following metric to characterize the number of effective particles at a given time step,

$$N_{eff,n} = \frac{1}{\sum_{i=1}^{N_p}(w_n^i)^2}. \tag{9}$$

Particle degeneracy can be alleviated by resampling the particles and reinitializing the weights when the number of effective particles becomes too low. In our context, the nuclear source intensity and coordinates are fixed throughout an episode. We adapt the BPF for parameter estimation with a random walk process model that has low variance Gaussian noise. The initial particles were sampled uniformly from fixed intervals as specified in Table 2. Equation 2 and Equation 3 are the measurement model and likelihood, respectively. The background rate, $\lambda_b$, was considered constant and known.

Sequential importance resampling is a technique to combat particle degeneracy and occurs when the number of effective particles drops below a given threshold. We selected the *Srinivasan sampling process* (SSP) resampling proposed by Gerber et al. because of asymptotic convergence of the error variance [33]. Additionally, SSP resampling requires only $\mathcal{O}(N_p)$ operations. See [33] and [34] for more details.

### *B.2. Fisher Information Matrix (FIM)*

The FIM is a measure of the information content of a measurement relative to the measurement model. It was first used in optimal observer motion for bearings-only tracking by Hammel et al. [35]. In their implementation, the controller selects the action at each timestep that maximizes the determinant of the FIM (system observability), which is equivalent to minimizing the area of the uncertainty ellipsoids around the state estimates. This arises from the connection between the FIM and the *Cramér-Rao lower bound* (CRB).

The CRB provides a lower bound on the error covariance of an unbiased estimator and is the inverse of the FIM [36]. The FIM is the Hessian of the log-likelihood and is denoted as follows,

$$J_{n+1}(\mathbf{x}) = -\mathbb{E}[\nabla_x \nabla_x^T \ln(p(z_{n+1}|\mathbf{x}))], \tag{10}$$

where $T$ denotes the transpose. Morelande et al. derived the closed form FIM for the radiation source localization problem as [5],

$$J_{n+1}(\mathbf{x}) = \frac{\nabla_{\mathbf{x}}\lambda_{n+1}(\mathbf{x})\nabla_{\mathbf{x}}^T\lambda_{n+1}(\mathbf{x})}{\lambda_{n+1}(\mathbf{x})}, \tag{11}$$

where $\lambda_n(\mathbf{x})$ is defined in Eq. 2. This results in the following gradient for each parameter,

$$\begin{aligned}
\frac{\delta\lambda_n}{\delta\mathcal{I}_s} &= \frac{1}{(x_n - x_s)^2 + (y_n - y_s)^2}, \\
\frac{\delta\lambda_n}{\delta x_s} &= \frac{2(x_n - x_s)\mathcal{I}_s}{[(x_n - x_s)^2 + (y_n - y_s)^2]^2}, \\
\frac{\delta\lambda_n}{\delta y_s} &= \frac{2(y_n - y_s)\mathcal{I}_s}{[(x_n - x_s)^2 + (y_n - y_s)^2]^2}.
\end{aligned} \tag{12}$$

Ristic et al. used the BPF particles at each time step to calculate the FIM as follows,

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

$$J_{n+1}(x_n) \approx \sum_{j=1}^{N_p} J_{n+1}(\mathbf{x}_n^j) w_n^j, \tag{13}$$

due to better performance when the posterior is multi-modal [9]. They applied this formulation to action selection in the radiation source search in the following manner,

$$a_{n+1} = \arg\max_{u_{n+1,L}} \left[ \sum_{l=n+1}^{L} \mathrm{tr}(J_l(u_l)) \right], \tag{14}$$

where $L$ is the number of lookahead steps, tr() is the matrix trace, and $u_n$ is the control vector that determines the detector's next position.

Helferty et al. proposed to use the trace of the CRB as it is a sum of squares of the axes of the uncertainty ellipsoid [37]. This is also known as A-optimality in the optimal experimental design literature [38]. Ristic et al. maximized the trace of the FIM that corresponds to maximizing the information, however, it is beyond the scope of this paper to show the relation between these two criteria. This control strategy will result in the optimal trajectory for minimizing the uncertainty of the estimated quantities given perfect source information (i.e., low or no measurement error). The source information in the nuclear source search context is not perfect due to the stochastic nature of nuclear decay and background radiation. Additionally, the FIM is not well defined for initial search conditions where the background radiation dominates the signal from the source, i.e., when the source-detector distance is large and/or the background rate is high.

### *B.3. Rényi Information Divergence (RID)*

Ristic et al. proposed another information-driven search strategy to address the shortcomings of the FIM-based approach. This approach is based upon the RID, also known as $\alpha$-divergence, a general information metric that quantifies the difference between two probability distributions. In Bayesian estimation, maximizing this difference corresponds to reducing the uncertainty around the state estimates. The use of RID was first proposed in the sensor management context by Kreucher et al. [39]. The RID is defined as,

$$D_\alpha(P||Q) = \frac{1}{\alpha - 1} \ln \left[ \int P^\alpha(x) Q^{1-\alpha}(x) dx \right], \tag{15}$$

where $\alpha$ specifies the order. In the limit as $\alpha$ approaches one, the RID approaches the Kullback-Leibler Divergence [39].

Ristic et al. adapted the RID for action selection in the nuclear source search context with a BPF [10]. The general flow of the algorithm is to apply an action from the set of actions to get the next potential detector position, calculate the expected posterior density for that action over a measurement interval, and then select the action that resulted in the greatest RID. The particle approximation of the RID is shown in the following equation,

$$\mathbb{E}[D_\alpha(p(\mathbf{x}^{u_{n+1}}|z), p(\mathbf{x}|z))] \approx \frac{1}{\alpha - 1} \sum_{z=Z_0}^{Z_1} p(z|\mathbf{x}) \ln \left[ \frac{p_\alpha(z|\mathbf{x}^{u_{n+1}})}{p(z|\mathbf{x})^\alpha} \right], \tag{16}$$

where $\mathbf{x}^{u_{n+1}}$ denotes the potential detector position after taking action $u_{n+1}$, $Z_0, Z_1$ is a measurement interval, and $z \in \mathbb{N}$. The density $p_\alpha(z|\mathbf{x}^{u_{n+1}})$ is approximated after filtering the latest measurement and particle resampling as,

$$p_\alpha(z|\mathbf{x}^{u_{n+1}}) = \sum_{j=1}^{N_p} w_n^j p(z|\mathbf{x}_n^{j,u_{n+1}})^\alpha, \tag{17}$$

and $p(z|\mathbf{x})$ results from the particle approximation of the marginal distribution of a measurement. Like the FIM, the RID can also be computed for L-step planning.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

*B.4. Hybrid RID-FIM Controller*

We propose a hybrid controller that utilizes either the RID or FIM as metrics for action selection. This was motivated in part by the empirical observation that the RID controller would often get stuck oscillating between two positions that were just above our termination criteria for source-detector distance resulting in incomplete episodes. The FIM is a poor control metric when there is little information available as is often the case at the start of a search. The RID is more computationally expensive than the FIM but provides a principled control method even in low information contexts. Thus, the RID was used for control at the beginning of each episode until the RID reached a sufficient threshold, then the metric was switched over to the FIM for the remainder as shown in Algorithm 1.

**Algorithm 1** RID-FIM Controller

**Input**: $\{\mathbf{x}_0^j, w_0^j\}_{j=1}^{N_p}$, set RID FLAG to 1, switch threshold $\eta$, effective particles threshold $\beta$, measurement interval $[Z_0, Z_1]$
Receive init. measurement, $z_0$, perform prediction and filtering of particles
**while** episode not terminated **do**
  **if** RID FLAG **then**
    Calculate RID according to 16 over $[Z_0, Z_1]$
  **else**
    Calculate FIM according to 13
  **end if**
  Select action that maximizes information metric
  Receive $z_{n+1}$, perform prediction and filtering of particles
  **if** $N_{eff} < \beta * N_p$ **then**
    Resample and reweight particles
  **end if**
**end while**

We decided on myopic (one-step lookahead) planning due to the exponential increase in computational cost inherent to both metric calculations. Additionally, many source search scenarios will have high uncertainty in the state estimates for many timesteps so planning far in advance is not advantageous. Myopic search is often sub-optimal but is a fair tradeoff when the problem dynamics are stable [39]. The parameter values for the RID-FIM, as well as the BPF, are detailed in Table 2. All parameters were selected by a parameter sweep over a set of 100 randomly sampled environments where the selection criteria was shortest average episode length and most episodes completed.

| Parameter | Value |
|---|---|
| $N_p$ | 6,000 |
| $N_{eff}$ | 6,000 |
| Process noise XY | 15 cps |
| Process noise $\mathcal{I}_s$ | 0.01 m |
| Prior XY | $[0, 22]$ m |
| Prior $\mathcal{I}$ | $[100, 1,000]$ cps |
| Resampling threshold, $\beta$ | 1.0 |
| Lookahead, $L$ | 1 |
| Order, $\alpha$ | 0.6 |
| Switch threshold, $\eta$ | 0.36 |
| Meas. interval $[Z_0, Z_1]$ | $\pm$75 cps |

**Table 2.** Parameter values for the BPF and RID-FIM.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 August 2021 doi:10.20944/preprints202108.0018.v1

*B.5. Posterior Cramér-Rao Lower Bound (PCRB)*

The BPF is a biased estimator as it only uses a finite number of particles. The PCRB provides a lower bound on the *root-mean-square error* (RMSE) performance for a biased estimator. Tichavsky proposed the PCRB for discrete-time nonlinear filtering [40], however, we follow a similar formulation found in Bergman's dissertation [41]. The PCRB is determined recursively in the following manner,

$$\begin{aligned} P_{0|0}^{-1} &= \Sigma^{-1}\Lambda^{-1}\int_x \nabla_x \lambda_0(\mathbf{x})\nabla_x^T \lambda_0(\mathbf{x})d\mathbf{x}, \\ P_{n+1|n+1}^{-1} &= Q_n + R_{n+1} - S_n^T(P_{n|n}^{-1} + V_n)^{-1}S_n, \end{aligned} \tag{18}$$

where the terms are $S_n$, $V_n$, and $Q_n$ are all the same inverse process noise covariance matrix, denoted as $\Sigma^{-1}$. This arises from the fact that our process model is a random walk with Gaussian noise for each state. The term $R_n$ is the FIM defined in Eq. 10. The prior, $P_{0|0}$, is a result of the uniform distribution of the particles where $\Lambda$ is a diagonal matrix of the uniform probabilities for each parameter. More details of the derivation of the PCRB and prior can be found in Bergman's dissertation in Theorem 4.5 and Section 7.3, respectively [41].

We average the RMSE and PCRB over the Monte Carlo evaluations resulting in the following formulation,

$$\sqrt{\frac{1}{K}\sum_{i=1}^{K}\left\|\hat{\mathbf{x}}_n^i - \mathbf{x}_n^i\right\|^2} \gtrsim \sqrt{\frac{1}{K}\sum_{i=1}^{K}\mathrm{tr}(P_n^i)}, \tag{19}$$

where $K$ is the total number of episodes and $\gtrsim$ denotes that the inequality only holds approximately for finite $K$ [41]. The PCRB provides an indicator of the suboptimality of an estimator and so we use it to directly compare the performance of the A2C with the RID-FIM. This is accomplished by evaluating the A2C with the exact same BPF estimator used with the RID-FIM for the source location state estimates. Not only can the estimator RMSE be compared against the PCRB, but the PCRBs resulting from both controllers can be compared as well. This will serve as a proxy for the quality of the control path generated by each controller.

### C. Gradient Search

We use the simple GS algorithm implemented by Liu et al. [14]. GS relies on sampling the gradient of the radiation field for each search direction at each timestep. This is not an efficient algorithm as the detector must make $D$ moves per action selection but serves as a useful baseline for performance comparison. The action selection is made stochastic by sampling from a multinomial distribution, denoted multi(n,p), over actions with probabilities proportional to the softmax of the gradients to avoid the trapping of local optima. GS is summarized by the following equation,

$$a_{n+1} \sim \mathrm{multi}(|\mathcal{U}|, \mathrm{softmax}([\frac{1}{q}\frac{\delta z_{n+1}}{\delta u_1}, \ldots, \frac{1}{q}\frac{\delta z_{n+1}}{\delta u_{|\mathcal{U}|}}])), \tag{20}$$

where $u$ is the detector position after action $i$, $\sigma$ is the softmax function, and $q$ is a temperature parameter. The temperature parameter was set at 0.0042 and was determined via a parameter sweep over a set of 100 randomly sampled environments where the selection criteria was shortest average episode length and most episodes completed.

## References

1. Sieminski, A.; others. International energy outlook. *Energy Information Administration (EIA)* **2014**, *18*, 2.
2. on the Effects of Atomic Radiation, U.N.S.C. *Sources and Effects of Ionizing Radiation: Sources*; Vol. 1, United Nations Publications, 2000.
3. Knoll, G.F., Radiation Detection and Measurement; John Wiley & Sons, 2010; chapter 3, pp. 73–76.

4. Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; others. Mastering the game of Go without human knowledge. *nature* **2017**, *550*, 354–359.
5. Morelande, M.; Ristic, B.; Gunatilaka, A. Detection and parameter estimation of multiple radioactive sources. 2007 10th International Conference on Information Fusion. IEEE, 2007, pp. 1–7.
6. Hite, J.; Mattingly, J. Bayesian Metropolis methods for source localization in an urban environment. *Radiation Physics and Chemistry* **2019**, *155*, 271–274.
7. Hellfeld, D.; Joshi, T.H.; Bandstra, M.S.; Cooper, R.J.; Quiter, B.J.; Vetter, K. Gamma-ray point-source localization and sparse image reconstruction using Poisson likelihood. *IEEE Transactions on Nuclear Science* **2019**, *66*, 2088–2099.
8. Cortez, R.; Papageorgiou, X.; Tanner, H.; Klimenko, A.; Borozdin, K.; Priedhorsk, W. Experimental implementation of robotic sequential nuclear search. 2007 Mediterranean Conference on Control & Automation. IEEE, 2007, pp. 1–6.
9. Ristic, B.; Gunatilaka, A. Information driven localisation of a radiological point source. *Information fusion* **2008**, *9*, 317–326.
10. Ristic, B.; Morelande, M.; Gunatilaka, A. Information driven search for point sources of gamma radiation. *Signal Processing* **2010**, *90*, 1225–1239.
11. Ristic, B.; Morelande, M.; Gunatilaka, A. A controlled search for radioactive point sources. 2008 11th International Conference on Information Fusion. IEEE, 2008, pp. 1–5.
12. Anderson, R.B.; Pryor, M.; Abeyta, A.; Landsberger, S. Mobile Robotic Radiation Surveying With Recursive Bayesian Estimation and Attenuation Modeling. *IEEE Transactions on Automation Science and Engineering* **2020**, pp. 1–15. Early Access.
13. Landgren, P.C. Distributed Multi-agent Multi-armed Bandits. PhD thesis, Princeton University, 2019.
14. Liu, Z.; Abbaszadeh, S. Double Q-learning for radiation source detection. *Sensors* **2019**, *19*, 960.
15. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2017, pp. 23–30.
16. Beigzadeh, A.M.; Vaziri, M.R.R.; Soltani, Z.; Afarideh, H. Design and improvement of a simple and easy-to-use gamma-ray densitometer for application in wood industry. *Measurement* **2019**, *138*, 157–161.
17. Brockman, G.; Cheung, V.; Pettersson, L.; Schneider, J.; Schulman, J.; Tang, J.; Zaremba, W. OpenAI gym. *arXiv preprint arXiv:1606.01540* **2016**.
18. Wierstra, D.; Förster, A.; Peters, J.; Schmidhuber, J. Recurrent policy gradients. *Logic Journal of the IGPL* **2010**, *18*, 620–634.
19. Henderson, P.; Islam, R.; Bachman, P.; Pineau, J.; Precup, D.; Meger, D. Deep reinforcement learning that matters. Proceedings of the AAAI conference on artificial intelligence, 2018, Vol. 32.
20. Tesauro, G. Temporal difference learning and TD-Gammon. *Communications of the ACM* **1995**, *38*, 58–68.
21. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438* **2015**.
22. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347* **2017**.
23. Andrychowicz, M.; Raichuk, A.; Stańczyk, P.; Orsini, M.; Girgin, S.; Marinier, R.; Hussenot, L.; Geist, M.; Pietquin, O.; Michalski, M.; others. What matters in on-policy reinforcement learning? A large-scale empirical study. *arXiv preprint arXiv:2006.05990* **2020**.
24. Cho, K.; Van Merriënboer, B.; Bahdanau, D.; Bengio, Y. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259* **2014**.
25. Olah, C. Understanding LSTM Networks. http://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.
26. Ma, X.; Karkus, P.; Hsu, D.; Lee, W.S. Particle filter recurrent neural networks. Proceedings of the AAAI Conference on Artificial Intelligence, 2020, Vol. 34, pp. 5101–5108.
27. Baca, T.; Stibinger, P.; Doubravova, D.; Turecek, D.; Solc, J.; Rusnak, J.; Saska, M.; Jakubek, J. Gamma Radiation Source Localization for Micro Aerial Vehicles with a Miniature Single-Detector Compton Event Camera. *arXiv preprint arXiv:2011.03356* **2020**.
28. Koenig, N.; Howard, A. Design and Use Paradigms for Gazebo, An Open-Source Multi-Robot Simulator. IEEE/RSJ International Conference on Intelligent Robots and Systems; , 2004; pp. 2149–2154.
29. Téllez, R.; Ezquerro, A.; Rodríguez, M.Á. *ROS Manipulation in 5 days: Entirely Practical Robot Operating System Training*; Independently published, 2017.
30. Welford, B. Note on a method for calculating corrected sums of squares and products. *Technometrics* **1962**, *4*, 419–420.
31. Stone, L.D. OR Forum—What's Happened in Search Theory Since the 1975 Lanchester Prize? *Operations Research* **1989**, *37*, 501–506.
32. Arulampalam, M.S.; Maskell, S.; Gordon, N.; Clapp, T. A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE Transactions on Signal Processing* **2002**, *50*, 174–188.
33. Gerber, M.; Chopin, N.; Whiteley, N.; others. Negative association, ordering and convergence of resampling methods. *Annals of Statistics* **2019**, *47*, 2236–2260.
34. Srinivasan, A. Distributions on level-sets with applications to approximation algorithms. Proceedings 42nd IEEE Symposium on Foundations of Computer Science. IEEE, 2001, pp. 588–597.
35. Hammel, S.; Liu, P.; Hilliard, E.; Gong, K. Optimal observer motion for localization with bearing measurements. *Computers & Mathematics with Applications* **1989**, *18*, 171–180.
36. Van Trees, H.L. *Detection, estimation, and modulation theory, part I: detection, estimation, and linear modulation theory*; John Wiley & Sons, 2004.

37. Helferty, J.P.; Mudgett, D.R. Optimal observer trajectories for bearings only tracking by minimizing the trace of the Cramér-Rao lower bound. Proceedings of 32nd IEEE Conference on Decision and Control. IEEE, 1993, pp. 936–939.
38. Pronzato, L. Optimal experimental design and some related control problems. *Automatica* **2008**, *44*, 303–325.
39. Kreucher, C.; Kastella, K.; Hero Iii, A.O. Sensor management using an active sensing approach. *Signal Processing* **2005**, *85*, 607–624.
40. Tichavsky, P.; Muravchik, C.H.; Nehorai, A. Posterior Cramér-Rao bounds for discrete-time nonlinear filtering. *IEEE Transactions on Signal Processing* **1998**, *46*, 1386–1396.
41. Bergman, N. Recursive Bayesian estimation: Navigation and tracking applications. PhD thesis, Linköping University, 1999.