2020 IEEE International Conference on Robotics and Automation (ICRA)
31 May - 31 August, 2020. Paris, France

Reinforcement Learning for Adaptive Illumination with X-rays

Jean-Raymond Betterton[1], Daniel Ratner[2], Samuel Webb[2], and Mykel Kochenderfer[1]

***Abstract*— We propose a learning algorithm for automating image sampling in scientific applications. We consider settings where images are sampled by controlling a probe beam's scanning trajectory over the image surface. We explore alternatives to obtaining images by the standard rastering method. We formulate the scanner control problem as a reinforcement learning (RL) problem and train a policy to adaptively sample only the highest value regions of the image, choosing the acquisition time and resolution for each sample position based on an observation of previous readings. We use convolutional neural network (CNN) policies to control the scanner as a way to generalize our approach to larger samples. We show simulation results for a simple policy on both synthetic data and real world data from an archaeological application.**

I. Introduction

In image sampling applications, we take measurements of an unknown image with the goal of collecting a set of measurements sufficient to reconstruct the image while minimizing the cost of measurement collection. Typically, images are formed by 'raster' scanning an analysis beam across a sample pixel-by-pixel. There are a wide range of different raster-based imaging techniques that are applied across a variety of scientific applications for these purposes. Examples include x-ray fluorescence imaging, laser ablation mass-spectrometry, secondary ion mass-spectrometry, and electron probe microanalysis to name a few. Due to the extreme sparsity encountered in scientific samples, the raster approach is often suboptimal. For example, for particle detection, a large fraction of the pixels may be uninformative. Instead, by adaptively changing the resolution during the scan, it is possible to focus the costly high resolution measurements on only the highest value portions of the sample. While it is possible to manually adapt experimental parameters during acquisition, human intervention is typically slow and hand written algorithms are costly to develop and often must be rewritten for each new application. Therefore, general algorithms that can automate the measurement collection process are of value.

In this paper, we use reinforcement learning to control a probe beam using acceleration and aperture controls to adaptively probe a sample. We must balance two competing objectives. The first objective is to collect a high quality set of measurements that can be used to create a high resolution estimate of the sample scanned. The second objective is to minimize the time required to collect measurements. Since the time elapsed increases with the number and overall quality of the measurements, we seek to learn a policy that trades off image reconstruction quality with elapsed time. Furthermore, we desire for our method to be compatible with black box image quality metrics, image reconstruction algorithms, and path planning algorithms that take a set of points to visit for a specified time as input. The first two compatibilities are to make the approach general while the latter compatability is to abstract away the complexity of path planning across large images.

We focus on x-ray fluorescence imaging (XRF) to evaluate our methodology. For XRF, the duration of a raster scan scales as the 4th power of pixel size when using an aperture to govern resolution. Several XRF imaging beam lines are employed to perform a variety of raster-based imaging techniques. Many of these beam lines cover a large range of analysis pixel sizes, from microns to hundreds of microns [1]. The experiment rasters the sample across the incident x-ray beam and collects the XRF spectrum at each pixel in the image. Different elements in the sample display different characteristic fluorescence lines that can be used to quantify the concentration of elements in the pixel. The XRF spectrum can be integrated over regions of interest or fit directly using first principles [2] to extract this information. While the raster method has existed for some time [3], [4], and has been extended to cover more extensive analyses (*e.g.* imaging at a series of selected excitation energies to extract chemical information [5], [6], [7], [8]), the data collection process is slow, and much of that time is spent in areas where there is limited signal, or signal of limited scientific interest.

Several analytical approaches have been implemented to try to overcome some of the drawbacks of raster imaging. These have included measuring the data on continuous straight trajectories [9], [10], [11], [1], arbitrary or circular trajectories [12], [13], [14], as well as Lissajous scanning [15]. While these approaches improve scan speeds by minimizing the changes in raster accelerations, the overall dwell time spent on any given pixel is typically the same, no matter the level of signal or interest.

Our work follows most directly from previous works that use reinforcement learning to do active learning [16], [17], [18], [19]. The main difference in our work is that we adapt the methodology to imaging applications.

Our approach is also related to previous works on adaptive sampling methods for images. Within the compressive sensing literature, algorithms exist that can recover sparse images using fewer measurements than suggested by the Nyquist-Shannon sampling theorem [20], [21], [22]. Amongst these, the adaptive algorithms are often referred to as active learn-

[1] Authors are with the Department of Computer Science, Stanford University. [jbetterton, mykel]@stanford.edu.

[2] Authors are with the SLAC National Accelerator Laboratory. [dratner, samwebb]@slac.stanford.edu.

This work was partially supported by the SLAC National Accelerator Laboratory.

ing or adaptive sampling algorithms [23], [24], [25], [26], [27], [28], [29], [30], [31]. These works are similar in that they seek to identify the most valuable measurements to take for the purposes of reconstructing an image. However, the problem we investigate here differs in the following three ways. First, instead of starting with a data selection criterion, we seek to learn a heuristic criterion for choosing data points, which in our case, depends on a training image data set and the choices of image reconstruction algorithm and image quality metric. Second, while prior adaptive sampling algorithms optimize the number of measurements taken, in this work, we optimize the cumulative time spent collecting measurements, a function of individual measurement exposure times and the path that the XRF beam takes across the sample surface. Third, we desire a method that requires minimal computation between measurements, particularly for large images.

Adaptive acquisition is gaining traction in x-ray applications. One recent example uses deep learning [31]. An adaptive sampling mask generating network and image inpainting network are trained end to end and tested in XRF applications. The mask generating network takes an accompanying RGB image as input. Our approach differs in that we assume no initial measurements; the only input to our algorithm at runtime is the history of previous measurements. Secondly, our algorithm collects measurements over multiple timesteps with multiple resolutions, and we vary the exposure time for each measurement. A second recent work uses Kriging, a form of Gaussian process regression, to guide sampling at a synchrotron [30]. Here, Kriging both reconstructs the image from measurements and quantifies uncertainty over pixel values, guiding future measurements. Our approach differs in that we directly train the RL algorithm on simulated and archived data, allowing the solution to learn arbitrarily complex priors.

Another related research area is Informative Path Planning (IPP) [32]. Past research has focused on problems where an agent must plan a path in order to optimally collect measurements while minimizing travel time or another related cost. Within this literature, our approach is similar to approaches that choose which points to measure first, and then determine the path that will visit those points [33]. However, our approaches differ both in application and in methodology for selecting which points to measure.

Our main contributions are 1) a general formulation of the measurement collection problem as a reinforcement learning problem that abstracts away the low level controls of the scanner itself, 2) a convolutional neural network policy that enables our policies to generalize to images of arbitrary shape and scale, and 3) a set of implementation techniques that can be used for solving the reinforcement learning problem, achieving performance above a raster baseline while preserving computational efficiency.

II. Background

A. Reinforcement Learning

Reinforcement learning considers the problem of training an agent to make decisions under uncertainty that maximizes an objective [34]. In reinforcement learning, the agent learns a function, π, that takes as input the current observation of the environment, s_t, and outputs an action a_t. At every timestep, the agent receives a reward signal r_t. The policy π is trained to maximize the expected, weighted sum of rewards over an episode, $E_\pi[\sum_{t=0}^{N} \gamma^t r_t]$, where γ is the discount rate.

B. X-ray Imaging

The general setting is to scan an X-ray beam across a sample, taking measurements pixel by pixel. Measurements can be as simple as transmission intensity, or as complex as measuring a full 2-D spectrum (energy-in, energy-out) at each pixel. Here, we consider a simple example of a single scalar measured at each pixel. In this setting, the aperture controls resolution. Using aperture rather than focusing means X-ray intensity scales as the area of the aperture, requiring longer exposure for the same signal to noise ratio.

1) Image Measurement Definitions: Consider a 2D image, v, with dimension $n_h \times n_w$. Let n be the number of pixels in image v where $n = n_h n_w$. We assume that we can take a measurement at one of the n pixel locations with one of m different apertures. Each aperture takes measurements at a different resolution as explained below. Let r_k be the radius of the kth aperture with r_k decreasing with k. We define a neighborhood function $\text{neigh}_k(i,j) = \{(i',j') | \|(i,j) - (i',j')\| \leq r_k\}$, for some norm (*e.g.* L_2 or L_∞). The noiseless reading for aperture k at pixel coordinate (i,j) is $z_{ijk} = \sum_{(i',j') \in \text{neigh}_k(i,j)} v_{i'j'}$.

When taking measurements, our readings are perturbed by noise. Let x be a 3D tensor where x_{ijk} is the exposure time at pixel coordinate (i,j) with aperture k. Let y be a 3D tensor where y_{ijk} is the observed measurement at coordinate (i,j) with aperture k. We have that $y_{ijk} \sim \text{Poisson}(x_{ijk}(z_{ijk}+c_k))$ for some noise constant c_k as a function of k.

2) Generating Measurements from Scanner Controls: Since the position of the beam is controlled with acceleration inputs, for some desired set of exposure times x, we must generate a trajectory, τ, with the beam such that the beam visits the set of locations of the image for the times specified by x. From here, we let the set of measurements generated by trajectory τ be $g(\tau)$. If we desire to execute exposure times x, we must select a trajectory, τ, such that $g(\tau) = x$.

In addition, switching between apertures causes a short delay during which the beam cannot collect measurements which we must account for when planning a trajectory. We denote the total time cost of a set of measurements, as $L_C(\tau)$.

3) Imaging Objectives and Problem Formulation: We collect measurements to optimize two competing objectives. Let f be an image reconstruction function such that $f(x,y) = \hat{v}$ is an estimate of the image, v. For our first objective, let $L_Q(v,\hat{v}) = L_Q(v, f(x,y))$ be a loss function that decreases as the quality of $\hat{v}$ increases, that is, as $\|v - \hat{v}\|$ approaches 0.

Our second objective is $L_C(\tau)$, the time cost for following a trajectory to generate exposure times $x = g(\tau)$.

If we consider a distribution over images v, and recall $y_{ijk} \sim \text{Poisson}(g(\tau)_{ijk}(z_{ijk} + c))$, a nonadaptive version of the problem is

$$\min_{\tau} \quad E[L_Q(v, f(g(\tau), y)) + \lambda L_C(\tau)] \tag{1}$$

for some hyperparameter $\lambda > 0$ that controls how much we favor scan speed over image reconstruction quality.

III. Methodology

A. Reinforcement Learning Formulation

In this section, we present a reinforcement learning problem formulation that abstracts away the low level acceleration and aperture switching controls of the scanner.

1) Sequential Decision Formulation: To solve the problem from the previous section adaptively, we can reformulate it as a sequential decision problem where at every timestep t, we observe all previous measurements and then choose a new set of measurements to take at the next timestep.

At each timestep t, let x_t be the cumulative acquisition time tensor up to timestep t, and y_t the cumulative sensor reading tensor up to timestep t. Then, let τ_t be the trajectory to execute at timestep t. Let $\tilde{x}_t = g(\tau_t)$ be the set of exposure times at time t, defined such that $x_{t+1} = x_t + \tilde{x}_t$, and similarly, let $\tilde{y}_t$ be the read values from measurements taken at time t, such that $y_{t+1} = y_t + \tilde{y}_t$. Note that since $(\tilde{y}_t)_{ijk} \sim \text{Poisson}((\tilde{x}_t)_{ijk}(z_{ijk} + c))$, we have that $(y_{t+1})_{ijk} \sim \text{Poisson}((x_{t+1})_{ijk}(z_{ijk} + c))$

We define the problem state to be $s_t = (x_t, y_t)$. At each timestep, we must select τ_t. Let π be a policy function and let the action $a_t = \pi(s_t)$. Rather than having π directly output τ_t, we instead compute it as $\tau_t = q(h(a_t, t))$. Here, h and q are both functions designed with the purpose of abstracting away low level aperture and acceleration controls as explained in the following two subsections.

2) Managing Acceleration Constraints: Let $\tilde{x}_t^*$ be some set of exposure times that we desire to collect. We introduce a function q that takes as input some desired set of exposure times $\tilde{x}_t^*$ and produces a trajectory τ_t.

In general, we assume that q is a search algorithm that searches over feasible trajectories for a trajectory such that the resulting set of measurement times $\tilde{x}_t = g(q(\tilde{x}_t^*))$ approximately minimizes $\|\tilde{x}_t^* - \tilde{x}_t\|$. By letting $\tau_t = q(\tilde{x}_t^*)$, we abstract away the low level acceleration controls and focus on outputting desirable sets of measurement times $\tilde{x}_t^*$.

3) Managing Aperture Switches: Even if we output $\tilde{x}_t^*$, there is still the complication of determining when to switch apertures. Discussed previously, switching apertures causes a delay where measurements cannot be collected. We reduce the complexity of the problem by constraining our actions in the sequential problem through $\tilde{x}_t^*$. First, we restrict $\tilde{x}_t^*$ to only contain nonzero exposure times for at most one aperture at a time. Second, we constrain the policy to progress from low resolution to high. Hence, $(\tilde{x}_t^*)_{ijk}$ can only be nonzero for a single value of k and if $(\tilde{x}_t^*)_{ijk}$ is nonzero, then, for $t' > t$, $(\tilde{x}_{t'}^*)_{i'j'k'}$ can only be nonzero for $k' \geq k$.

We perform at most $m - 1$ aperture switching operations per episode, as opposed to arbitrarily many switches per action. Although this makes our solution less general, progressing from low resolution to high seems to be a reasonable restriction in practice and reduces the complexity of the solution space. Since exposure times can only be output for at most one aperture at a time, the policy output, a_t, need only be two dimensional to specify a desired $\tilde{x}_t^*$. Also, note that with these changes, we must still have some mechanism for switching apertures.

We define h such that $\tilde{x}_t^* = h(a_t, t)$ where a_t is of shape $n_h \times n_w$. Let $\tilde{h}$ be a function defined such that $\tilde{h}(t)$ is the aperture to use at timestep t. Then, we specify that $h(a_t, t)_{ijk} = (\tilde{x}_t^*)_{ijk} = (a_t)_{ij}$ for $k = \tilde{h}(t)$ and $h(a_t, t)_{ijk} = (\tilde{x}_t^*)_{ijk} = 0$ for $k \neq \tilde{h}(t)$. In this work, apertures are switched after each timestep, that is $\tilde{h}(t) = t$. We leave the exploration of defining or learning more complex h to future work.

4) RL Problem Formulation: Now, suppose that we have an imaging problem, where we have chosen a distribution over images v, image reconstruction function f, aperture switching schedule h, trajectory planner q, image quality metric L_Q, trajectory cost metric L_C, and cost tradeoff hyperparameter λ. Then, our original imaging problem can be rewritten as a sequential decision problem

$$\min_{\pi} \quad E_{v,y}[L_Q(v, f(x_T, y_T)) + \lambda \sum_{t=1}^{T} L_C(q(h(\pi(s_t), t))] \tag{2}$$

for some number of timesteps $T > 0$.

In principle, we can solve the sequential decision problem with a reinforcement learning algorithm. Let $L(x_t, y_t) = L_Q(v, f(x_t, y_t)) + \lambda L_C(q(h(a_t, t)))$. Then, let $r_1 = -L(x_1, y_1)$, and $r_t = -(L(x_t, y_t) - L(x_{t-1}, y_{t-1}))$, $t \in \{2, \ldots, T\}$ where T is the total number of steps in the episode. We can express Equation 2 explicitly as a reinforcement learning problem with states, actions, and rewards defined respectively as s_t, a_t, and r_t. Note that, if we set the RL discount rate, γ, equal to 1, then the sum of rewards we seek to maximize, $\sum_{t=1}^{T} r_t$, is equal to $-L(x_T, y_T)$, the negative of the loss function we are attempting to minimize.

B. Convolutional Policy Design

Since a_t is of shape $n_h \times n_w$, there are a few issues. First, the shape of a_t varies with v. Since we sample a v every episode, each episode may require a different shaped output and in general, we would like our policy π to generalize to v of arbitrary shape. A second issue is that, even for relatively small images, a_t is very high dimensional which may lead to difficulty or infeasiblity when it comes to doing reinforcement learning.

In order to address these potential issues, we let the target exposure time for a pixel be a function of the local observation around that pixel. This addresses the scaling issue as the policy output for each pixel is a function of a local region of fixed size. This also addresses some of the issues with the dimensionality of a_t as now the complexity

of π scales with the size of the local observation that we consider. To do this, we elect to use convolutional neural network [35], [36] policies where the size of the receptive field dictates the size of the local observation that we use.

We only consider CNN architectures that preserve the dimensionality of the input in the first two dimensions. That is, our policy function takes an input of shape $n_h \times n_w \times n_f$ where n_f is the number of input features for each pixel location derived from s_t. The policy function outputs an action of shape $n_h \times n_w$, an approximate exposure time for each pixel location at the current aperture, as desired.

C. Efficient Implementation

Depending on f, the cost of reconstructing an image and calculating the reward function can bottleneck our training algorithms. One way to reduce the computational cost of image reconstruction is to limit the size of the images processed. We train with smaller images to reduce the computation time required for each individual episode. We found this to be very beneficial for computational efficiency during our experiments.

A second way to reduce the cost of image reconstruction is through approximation. We may use a fast approximation of f, call it $\hat{f}$ when estimating our reward values. We note that although we do not demonstrate this in the current work, it is an interesting area for future work.

A third way we can improve computational efficiency is by evaluating the reward signal less frequently as each reward signal requires an image reconstruction. This leads to a tradeoff between density of rewards and faster runtimes for simulations. In our experiments, we only evaluate the cumulative reward over the entire episode at the end.

IV. Experiments

A. Simulation Environment

We reconstruct all images using L-BFGS [37] to minimize the weighted sum of the negative log likelihood of the reconstructed image given the collected measurements and a TV regularizer. We evaluate image quality using the Mean Squared Error (MSE) between the reconstructed and ground truth images. That is, $L_Q(v, \hat{v}) = \frac{1}{n}\sum_{i=1}^{n_h}\sum_{j=1}^{n_w}(v_{ij} - \hat{v}_{ij})^2$.

All environments impose constraints on the maximum velocity ($200\,\mathrm{mm/s}$) and maximum acceleration ($500\,\mathrm{mm/s^2}$) of the XRF beam. Every pixel is square shaped with a side length of $0.02\,\mathrm{mm}$. We assume that, in practice, the images we scan will be large such that the maximum cost of switching apertures at most $m-1$ times is negligible, therefore we do not account for aperture switching costs during these experiments and we set L_C equal to the total elapsed time, where $L_C(x) = \sum_{i=1}^{n_h}\sum_{j=1}^{n_w}\sum_{k=1}^{m} x_{ijk}$.

All environments make use of the same q function for transforming arbitrary sets of exposure times specified by $h(a,t)$ to real trajectories $q(h(a,t))$. For this, we first fix the order in which each measurement is taken. Recall that the rows of the image are indexed with i for $i \in \{1, 2, \ldots, n_h\}$. We proceed one row at a time from $i = 1$ to $i = n_h$. For row i, if i is odd, we take the measurements in that row from left to right, otherwise from right to left. With the order of measurements to acquire fixed, we then seek to find a feasible trajectory that approximately minimizes the total required time to visit the pixels in that order for at least as long as is specified by a. For this, we can plan for each row independently of the others. For each row, we solve this problem approximately with an iterative method. On the first iteration, we consider the pixel with the longest specified exposure time, requiring the slowest speed to measure. We calculate the maximum speed that we can achieve over the two adjacent pixels that still enables the scanner to slow down enough to fulfill the original point and the original specified exposure time. We adjust the exposure times of the left and right neighbors to reflect this maximum speed if their current speed is faster. On each subsequent iteration, we consider the point with the next longest specified exposure time and repeat. In this way, we can construct a feasible trajectory and generate approximate exposure times for that trajectory.

B. Training Image Data and Aperture Pairings

We refer to the first environment as the sparse environment. Each image is 20×20. Three random pixels are chosen to have a value of 1 while all the remaining pixels are assigned a value of 0. The background noise is Poisson with a level of 0.1. In this environment, our scanner has access to three square shaped apertures corresponding to resolutions of 1×1, 3×3, and 5×5 pixels. We train with λ values of 1×10^{-4}, 1×10^{-3}, 1×10^{-2}, 1×10^{-1}, and 1.

Our second environment is called the checkerboard texture environment. Each image is 20×20 and filled with a random number of randomly generated shapes. The number of random shapes is drawn from a Poisson distribution with mean equal to 1. Each shape is assigned a random base signal that is uniformly sampled between 0 and 1. Then, a checkerboard mask of the image is taken and all pixels in the mask have their value reduced by a factor of 2. The background noise is Poisson with a level of 0.1. In this environment, our scanner has access to three square shaped apertures corresponding to resolutions of 1×1, 3×3, and 5×5 pixels. We train with λ values of 1×10^{-6}, 1×10^{-5}, 1×10^{-4}, 1×10^{-3}, and 1×10^{-2}.

In our third environment, the realistic environment, each image is a 50×50 patch sampled from a small dataset of 17 real XRF images. The image signal levels range between 0 and 26 with an average value of approximately 0.5 and a background Poisson noise level of 1.5. In this environment, our scanner has access to five circular apertures with diameters of 1, 2.5, 7.5, 20, and 50 pixels respectively. We train with λ values of 10, 100, and 1000.

C. Methods

1) Rastering Baseline: We compare our learned policies to a baseline based on rastering, a technique where we use one fixed aperture and every pixel is measured for the same amount of time with the aperture of choice. We considered other heuristic methods, such as randomized scans, however

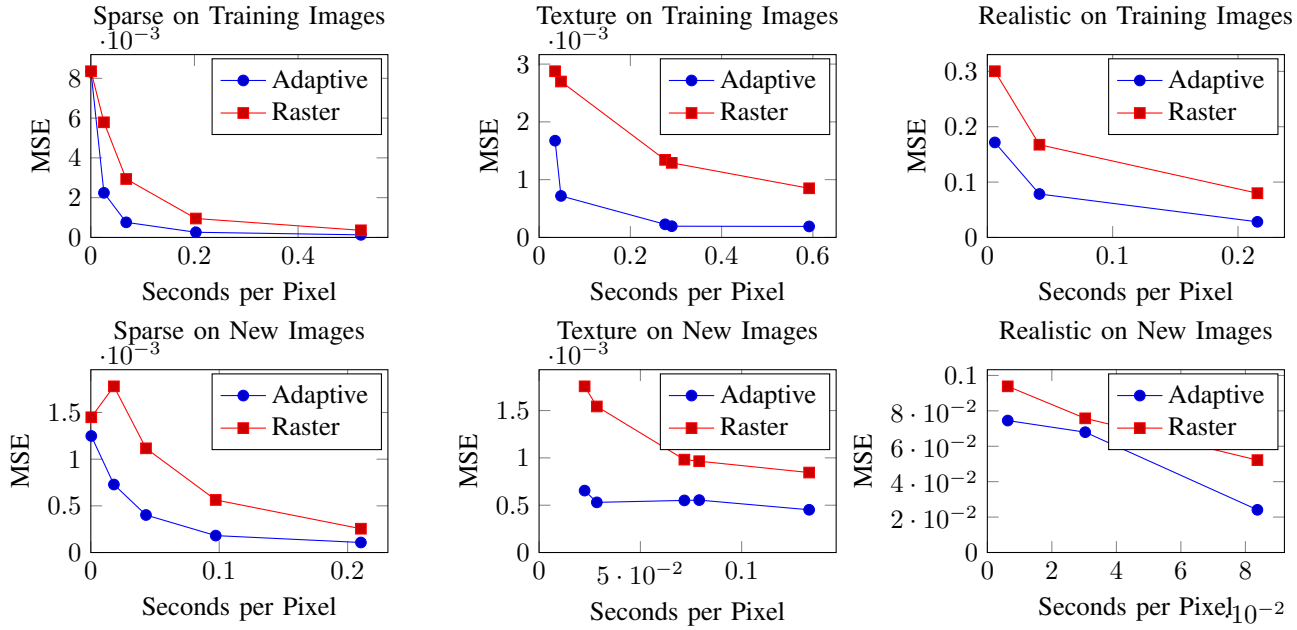

Fig. 1: This plot shows each sampling method's mean squared error versus time curve. For the adaptive policies, the top row shows performance within the training environment, the bottom row shows performance outside of the training environment, and each column corresponds to a different training environment for the adaptive policy. Also, each point in the plot corresponds to a model trained with a different λ value. We compare the adaptive policies to raster scans at all available resolutions that scan for the same average time per pixel. Only the best performing raster baseline is depicted.

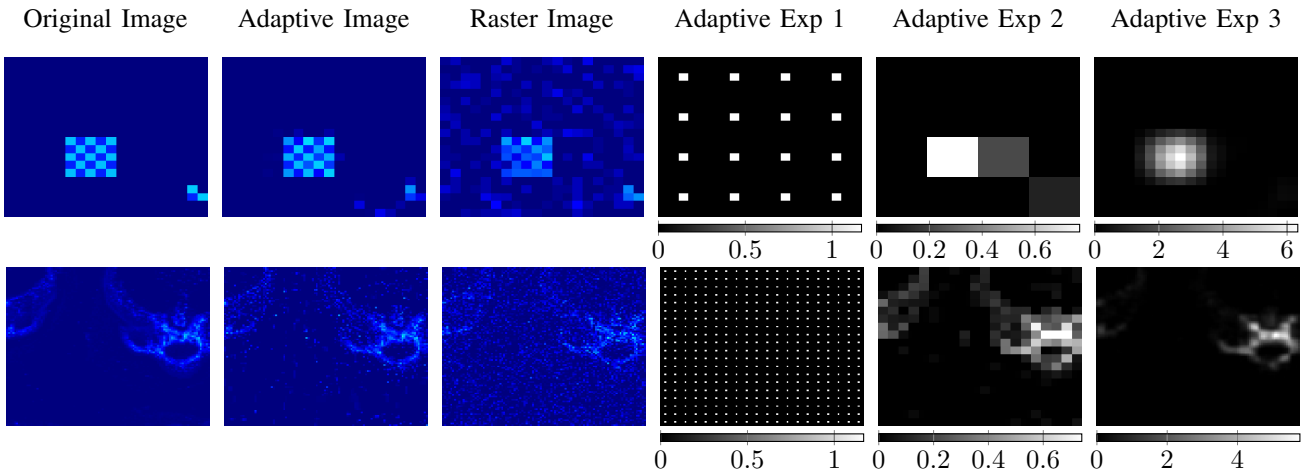

Fig. 2: This plot shows the reconstructed images for the learned adaptive policy trained in the texture environment with $\lambda = 1 \times 10^{-4}$ side by side with the highest resolution raster baseline set to scan for the same amount of time as the adaptive method. The first row shows the performance on the texture environment data. The second row shows the performance of the same policy applied to a real image. Progressing from left to right, the columns depict the original image, the adaptive reconstruction, the highest resolution raster reconstruction, and the exposure times with aperture 1, aperture 2, and aperture 3 for the adaptive method. The colorbars for the exposure time plots indicate seconds per pixel spent in each region.

these were either on par with or dominated by the raster baseline. Techniques based on compressive sensing and prior learning based approaches may be applicable, but will require dedicated research in the future to adapt to this problem formulation.

2) Proposed Method: In this work, we use a simplified observation $s_t = y_t$ with shape $n_h^{(v)} \times n_w^{(v)} \times m$. Although information is lost and learned behavior is likely to be suboptimal in general, we found the reduction in the size of s_t helpful and noted no significant drop in performance when compared to $s_t = (x_t, y_t)$, likely due to simplicity in our policy architectures.

For our policy architecture, we use a combination of m CNNs, referred to as π_k. Each π_k controls one of the m aper-

tures. Since we switch apertures after each timestep, we use a different π_k each timestep. More formally $\pi(s_t) = \pi_t(s_t)$. For these experiments, we design the π_k CNN architectures to be minimalistic while still maintaining a receptive field at least as large as the aperture being controlled. We found it helpful to manually convolve $(s_t)_{:,:,k}$ with a binary mask of 1s in the shape of the kth aperture, the aperture that collected the measurements. Although a similar operation can also be learned by a more complex CNN, we found this to be a helpful preprocessing step for our minimalistic architectures. With this preprocessing expanding the effective receptive field, each CNN, π_k, is comprised of a single convolutional layer with a receptive field of just 1×1, a ReLu nonlinearity, and a customized sparsity enforcing layer. The sparsity layer is controlled by two additional weights that control the spacing of measurements horizontally and vertically. Conceptually, the weights control the length and width of a rectangular window with shape $w_h \times w_w$. The image is then divided into nonoverlapping tiles of shape $w_h \times w_w$. Finally, all measurements values within a tile are concentrated into the center of the tile. In this way, larger windows lead to sparser measurements. Window size is limited to be no larger than $2r_k \times 2r_k$ for square apertures and $\sqrt{2}r_k \times \sqrt{2}r_k$ for circular apertures.

Since we only use one aperture per step and proceed in order from low resolution to high, by the time we use π_{k+1} only a subset of the input of shape $n_h^{(v)} \times n_w^{(v)} \times k$ can be nonzero. Therefore, the π_k policy functions progressing in order from low resolution to high, have $k+3$ weights each. For the sparse and texture environment with 3 apertures, this corresponds to a total of 15 weights while the policies in the realistic environment train a total of 30 weights.

We use Evolution Strategies [38] as our policy learning algorithm but with rewards replaced by fitness shaping values [39]. For both environments, we set step size $\alpha = 1$ and population variance $\sigma = 1$ and ran for 500 iterations. Per iteration, we sample 20 sets of policy weights and evaluate each one on 20 episodes. Since Evolution Strategies is unaffected by sparse rewards, we only evaluate the reward at the end of each episode to preserve computational efficiency as image reconstructions are relatively costly operations.

Lastly, we found that a policy of collecting no measurements in some cases behaved as a local optima that prevented exploration to better policies. To counter this effect, we augment the reward signal with a relatively large constant penalty for each pixel that is not measured by the scanner.

D. Results

1) Performance within Training Environments: After training, we test each model within the environment it was trained on. Each model is tested on 1000 new images from its training environment.

In all three environments, we found that the learned adaptive policy was able to outperform all baseline rastering policies across all lambda values. A plot showing the average MSE and time spent scanning is shown in Figure 1. Figure 2 shows an example image reconstruction for a policy trained in the texture environment with $\lambda = 1 \times 10^{-4}$ in addition to a visualization of where the learned policy spent its time scanning. These results show that we can learn to perform well on new images drawn from the same distribution that we trained on.

2) Performance outside Training Environments: We also test the performance of policies on data that is from a different distribution than the training data. In this setting all images are larger at size 100×100. Each policy is tested on 1000 new images.

For the policies trained in the simple environment, the large images are generated such that a random number of randomly selected pixels have value 1 while the rest have value 0. The number of pixels is Poisson with mean 10.

For the policies trained in the checkerboard texture environment, the large images are selected as random 100×100 patches from a hand-selected set of 15 real images from XRF applications (Figure 2). The images were selected to be similar to the texture environment data in terms of sparsity.

For the policies trained in the realistic environment, we again sample 100×100 patches from a set of 4 sparse, real images from XRF applications. These images were not included in the training set.

In the results we show here, we found that the learned adaptive policy was able to outperform all baseline rastering policies across all λ values, showing the potential for policies to generalize to new image distributions. A plot showing the average MSE and time spent scanning is shown in Figure 1. Figure 2 shows an example image reconstruction and exposure time visualizations for a policy trained in the texture environment with $\lambda = 1 \times 10^{-4}$.

It should be noted that in some preliminary trials not shown here, we did not always observe the adaptive method outperforming baselines on new image distributions. We found that how well the learned policy generalized seemed to depend on the degree of similarity between the sparsity levels of the two image distributions. We believe that the extent to which policies can generalize to new distributions of images is highly domain specific and could be an interesting topic for future work.

V. Conclusion and Future Work

These results demonstrate that reinforcement learning is a viable framework for generating adaptive sampling policies for images in XRF applications. Our experiments show that the method can outperform simple raster policies on both synthetic and real data using policy architectures containing relatively few weights. Our approach is compatible with black box image reconstruction functions, image quality metrics, and trajectory generating functions.

For future work, we believe that several of the modular components of this framework can be further developed. Some examples include exploring more complex image quality metrics, image reconstruction processes, and path planning algorithms. Further, with more complex policy architectures and richer training data, future work may produce policies that exploit more complex patterns in the image data.

REFERENCES

[1] N. P. Edwards, S. M. Webb, C. M. Krest, D. van Campen, P. L. Manning, R. A. Wogelius, and U. Bergmann, "A new synchrotron rapid-scanning x-ray fluorescence (SRS-XRF) imaging station at SSRL beamline 6-2," *Journal of Synchrotron Radiation*, vol. 25, no. 5, pp. 1565–1573, 2018.

[2] V. Solé, E. Papillon, M. Cotte, P. Walter, and J. Susini, "A multiplatform code for the analysis of energy-dispersive x-ray fluorescence spectra," *Spectrochimica Acta Part B: Atomic Spectroscopy*, vol. 62, no. 1, pp. 63–68, 2007.

[3] K. Jones, W. Berry, D. Borsay, H. Cline, W. Conner Jr, and C. Fullmer, "Applications of synchrotron radiation-induced x-ray emission (SRIXE)," *X-Ray Spectrometry: An International Journal*, vol. 26, no. 6, pp. 350–358, 1997.

[4] Z. Cai, B. Lai, W. Yun, I. McNulty, A. Khounsary, J. Maser, P. Ilinski, D. Legnini, E. Trakhtenberg, S. Xu, *et al.*, "Performance of a high-resolution x-ray microprobe at the advanced photon source," in *American Institute of Physics*, vol. 521, no. 1. AIP, 2000, pp. 31–34.

[5] J. Kinney, Q. Johnson, M. C. Nichols, U. Bonse, and R. Nusshardt, "Elemental and chemical-state imaging using synchrotron radiation," *Applied Optics*, vol. 25, no. 24, pp. 4583–4585, 1986.

[6] S. Sutton, S. Bajt, J. Delaney, D. Schulze, and T. Tokunaga, "Synchrotron x-ray fluorescence microprobe: Quantification and mapping of mixed valence state samples using micro-xanes," *Review of Scientific Instruments*, vol. 66, no. 2, pp. 1464–1467, 1995.

[7] I. J. Pickering, R. C. Prince, D. E. Salt, and G. N. George, "Quantitative, chemically specific imaging of selenium transformation in plants," *Proceedings of the National Academy of Sciences*, vol. 97, no. 20, pp. 10 717–10 722, 2000.

[8] L. Mayhew, S. Webb, and A. Templeton, "Microscale imaging and identification of fe speciation and distribution during fluid–mineral reactions under highly reducing conditions," *Environmental Science & Technology*, vol. 45, no. 10, pp. 4468–4474, 2011.

[9] A. E. Morishige, H. S. Laine, M. A. Jensen, P. X. Yen, E. E. Looney, S. Vogt, B. Lai, H. Savin, and T. Buonassisi, "Accelerating synchrotron-based characterization of solar materials: Development of flyscan capability," in *IEEE Photovoltaic Specialists Conference (PVSC)*, 2016.

[10] K. Medjoubi, N. Leclercq, F. Langlois, A. Buteau, S. Lé, S. Poirier, P. Mercere, M. C. Sforna, C. M. Kewish, and A. Somogyi, "Development of fast, simultaneous and multi-technique scanning hard x-ray microscopy at synchrotron soleil," *Journal of Synchrotron Radiation*, vol. 20, no. 2, pp. 293–299, 2013.

[11] M. W. Jones, N. W. Phillips, G. A. Van Riessen, B. Abbey, D. J. Vine, Y. S. Nashed, S. T. Mudie, N. Afshar, R. Kirkham, B. Chen, *et al.*, "Simultaneous x-ray fluorescence and scanning x-ray diffraction microscopy at the australian synchrotron XFM beamline," *Journal of Synchrotron Radiation*, vol. 23, no. 5, pp. 1151–1157, 2016.

[12] D. J. Ching, M. Hidayetoğlu, T. Biçer, and D. Gürsoy, "Rotation-as-fast-axis scanning-probe x-ray tomography: the importance of angular diversity for fly-scan modes," *Applied Optics*, vol. 57, no. 30, pp. 8780–8789, 2018.

[13] J. Deng, C. Preissner, J. A. Klug, S. Mashrafi, C. Roehrig, Y. Jiang, Y. Yao, M. Wojcik, M. D. Wyman, D. Vine, *et al.*, "The velociprobe: An ultrafast hard x-ray nanoprobe for high-resolution ptychographic imaging," *Review of Scientific Instruments*, vol. 90, no. 8, 2019.

[14] M. Odstrčil, M. Holler, and M. Guizar-Sicairos, "Arbitrary-path fly-scan ptychography," *Optics Express*, vol. 26, no. 10, pp. 12 585–12 593, 2018.

[15] K. Hwang, Y.-H. Seo, J. Ahn, P. Kim, and K.-H. Jeong, "Frequency selection rule for high definition and high frame rate lissajous scanning," *Scientific Reports*, vol. 7, no. 1, p. 14075, 2017.

[16] M. Fang, Y. Li, and T. Cohn, "Learning how to active learn: A deep reinforcement learning approach," in *Empirical Methods in Natural Language Processing*, 2017.

[17] K. Pang, M. Dong, Y. Wu, and T. M. Hospedales, "Meta-learning transferable active learning policies by deep reinforcement

[19] M. Woodward and C. Finn, "Active one-shot learning," *CoRR*, vol. abs/1702.06559, 2017. [Online]. Available: http://arxiv.org/abs/1702.06559

learning," *CoRR*, vol. abs/1806.04798, 2018. [Online]. Available: http://arxiv.org/abs/1806.04798

[18] P. Bachman, A. Sordoni, and A. Trischler, "Learning algorithms for active learning," in *International Conference on Machine Learning*, 06–11 Aug 2017, pp. 301–310.

[20] E. J. Candes, J. Romberg, and T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[21] E. J. Candes and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 21–30, 2008.

[22] D. L. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.

[23] D. A. Cohn, Z. Ghahramani, and M. I. Jordan, "Active learning with statistical models," in *Journal of Artificial Intelligence Research*, vol. 4, 1994, pp. 129–145.

[24] Y. Eldar, M. Lindenbaum, M. Porat, and Y. Y. Zeevi, "The farthest point strategy for progressive image sampling," *IEEE Transactions on Image Processing*, vol. 6, no. 9, pp. 1305–1315, 1997.

[25] D. M. Malioutov, S. R. Sanghavi, and A. S. Willsky, "Sequential compressed sensing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 2, pp. 435–444, April 2010.

[26] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Asilomar Conference on Signals, Systems and Computers*, 1993.

[27] B. Settles, "Active learning literature survey," University of Wisconsin, Madison, Tech. Rep. 1648, 2010.

[28] A. Taimori and F. Marvasti, "Adaptive sparse image sampling and recovery," *IEEE Transactions on Computational Imaging*, vol. 4, no. 3, pp. 311–325, Sep. 2018.

[29] Z. Devir and M. Lindenbaum, "Blind adaptive sampling of images," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1478–1487, April 2012.

[30] M. M. Noack, K. G. Yager, M. Fukuto, G. S. Doerk, R. Li, and J. A. Sethian, "A kriging-based approach to autonomous experimentation with applications to x-ray scattering," *Scientific Reports*, vol. 9, no. 1, p. 11809, 2019.

[31] Q. Dai, H. Chopp, E. Pouyet, O. Cossairt, M. Walton, and A. Katsaggelos, "Adaptive image sampling using deep learning and its application on x-ray fluorescence image reconstruction," *ArXiv*, vol. abs/1812.10836, 12 2018.

[32] A. Singh, A. Krause, C. Guestrin, and W. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[33] G. A. Hollinger, B. Englot, F. S. Hover, U. Mitra, and G. S. Sukhatme, "Active planning for underwater inspection and the benefit of adaptivity," *International Journal of Robotics Research*, vol. 32, no. 1, pp. 3–18, 2013.

[34] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[35] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological Cybernetics*, vol. 36, pp. 193–202, 1980.

[36] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989.

[37] D. C. Liu and J. Nocedal, "On the limited memory bfgs method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1, pp. 503–528, Aug 1989.

[38] T. Salimans, J. Ho, X. Chen, and I. Sutskever, "Evolution strategies as a scalable alternative to reinforcement learning," *ArXiv*, vol. abs/1703.03864, 03 2017.

[39] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 949–980, Jan. 2014.