Reducing the ATHENA WFI background

Xuehao Ding

Abstract

The Wide Field Imager (WFI) on Athena X-ray observatory is a detector system which will include on-board processing algorithms designed to reduce the particle background [1]. In this project, I make use of the unsupervised learning algorithm and deep learning method to identify and eliminate the background. The algorithms in this project are tested on images generated from GEANT4 and SIXTE simulated datasets, which shows that machine learning methods are effective in distinguishing between photon-induced and proton-induced patterns on the detector.

1 Introduction:

Athena (Advanced Telescope for High Energy Astrophysics), European Space Agency's next large X-ray observatory, is scheduled to be launched in the early 2030s, which will explore the hot and energetic universe using two selectable focal plane instruments [2]. The Wide Field Imager (WFI) [1], which is one of two instruments, is a detector with $450\mu m$ depletion layer and will provide high spatial resolution imaging in the 0.2-15 keV energy band over a wide, $40 * 40$ square arcminute field-of-view. As the primary science goals of the WFI are sensitive to the total level of instrumental background, it is required to reduce the instrumental background level of the WFI. As part of the U.S. contribution to the WFI, the Science Products Module (SPM), a secondary CPU, will reduce the ATHENA WFI particle background level by improving background rejection on board, considering that full frame pixel data is produced at a rate which is much too rapid to telemeter all the pixels to the ground.

The main component of the instrumental background is generated by high energy protons (with $E > 100MeV$) that produce secondaries along their way by depositing some of their energy. The secondaries of the unfocused component detected by the WFI are hard to distinguish from X-ray events from celestial sources.

In this project, my main goal is to reduce the WFI background by using machine learning methods. The input to my algorithm is a 500 by 500 grayscale image, where the value of each pixel represents its energy. I then use DBSCAN (an unsupervised algorithm) to cluster the nonzero pixels and use neural network to identify if each cluster includes a photon. The final output is a processed 500 by 500 image where the background has been eliminated.

2 Related work

There are some existing works to understand, model, and reduce the WFI background with classical algorithms without using deep learning method. One of the approaches is to reject hits which are next to a pixel whose deposited energy is more than 15 keV [2, 3, 4]. The reasoning behind this threshold is that energy deposits above 15 keV are much more likely to originate from particles than from photons. And secondary particles produced by such a high energetic primary tend to be close to the primary which produced them. Another approach is to study spacial correlations between particle events and photon events, which is helpful to improve the rejection of particle background events [3, 5]. What's more, in Ref. [3, 6], each event is assigned a grade based on the 3 by 3 pattern of pixels above a threshold using ASCA grading scheme. Events with grades ≤ 12 are flagged to be photon events and events with grades > 12 are considered as particle events.

In summary, these existing works all tried to find features manually to distinguish between particles and X-ray photons. In contrast, my project aims to discover features using deep learning neural network, which is the main difference.

3 Dataset and Features

The data I analyze are from GEANT4 [7] simulated WFI particle background data and SIXTE [8] simulated X-ray data. GEANT4 simulates particles through the telescope and records how they interact with the detector. SIXTE simulates the X-ray detected and the interaction with the detector.

An input sample to DBSCAN records the energy levels of the pixels on the detector in one frame, which is a 500 by 500 grayscale image. Fig. 1 shows some typical patterns of photons and particles. The color of each pixel represents its energy level. In the real case, there are many such patterns on the detector in each frame and it is also possible that multiple events overlap each other. For example, the left panel of Fig. 4 is an input sample to DBSCAN.

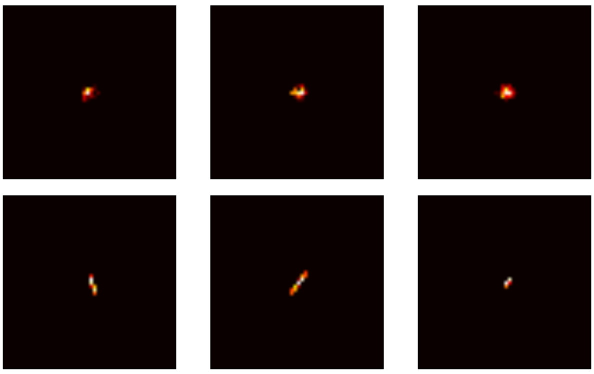

Figure 1: The upper three figures are examples of photons. The lower three figures are examples of particles.

For the neural network, each input sample is a single-cluster grayscale image, which is a 50 by 50 matrix. I create 10000 train samples and 6000 validation samples by combining events from GEANT4 and SIXTE databases randomly. Each created sample is checked by DBSCAN whether it contains only one cluster and the sample will be discarded if not. Each sample may contain $0-1$ photon and $0-2$ particles, considering the probability that more events form a cluster in one frame is neglectable in the real case. A sample will be labeled as 1 if it contains a photon, and 0 otherwise. Fig. 2 shows some examples.

Before sending the single-cluster figures into the neural network, I map every pixel value through the Heaviside step function. In other words, I assign a constant value to every non-zero pixel. This preprocessing is very useful because it turns out that the prediction accuracy becomes more than 99% in this case, while the prediction accuracy will be only about 70% if the

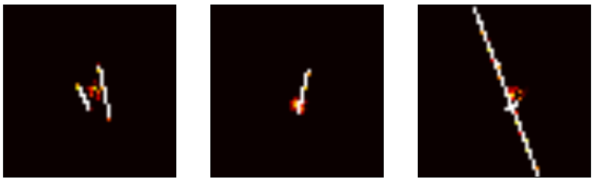

Figure 2: Three samples of the train set.

neural network runs on the single-cluster figures directly. The intuitive reason for this is that the energy of a photon is most probably much weaker than that of a particle, which results in that the neural network does not "pay much attention" to photon pixels.

The feature in the algorithm is just the original input.

4 Methods:

In this project, I use DBSCAN, an unsupervised learning algorithm, to cluster the activated pixels, and use deep learning algorithm to identify whether each cluster contains photons or not.

4.1 DBSCAN

The ultimate goal of this project is to identify whether each activated pixel belongs to a photon or a particle. But it is not practical to run a neural network on the whole detector image. The goal of this section is to cluster the activated pixels based on their connectivity, where each cluster will be processed by the neural network in the next section.

After comparing several unsupervised learning algorithms, I found that DBSCAN (Density-based spatial clustering of applications with noise) algorithm [9, 10] is best suited for our problem. The basic idea of DBSCAN is that every point in a cluster should have several nearby neighbors that also belong to this cluster. DBSCAN has two parameters: ϵ, which is the radius for each point to search for its nearby neighbors, and $minPts$, which is the minimum number of points within the radius for a point to be considered as a core point. The procedure of DBSCAN is as follows:

1. Classify each data point: **Core point:** A point that has at least $minPts$ neighbor points within its ϵ radius. **Border point:** A point within the ϵ radius of a core point but has less than $minPts$ other points within its own ϵ radius. **Noise point:** A point that is neither a core point or a border point. Fig. 3 is an example.

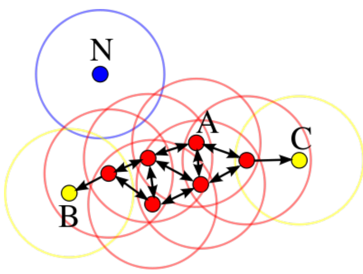

Figure 3: A is a core point. B and C are border points. N is a noise point.

2. If one data point is within the ϵ radius of a core point, these two points are considered directly density-reachable. Larger clusters are formed when directly density-reachable points are chained together.

4.2 Deep Learning

After the preprocessing mentioned above, the input (50 by 50 matrix) will be processed by the neural network [11, 12].

The first layer of this neural network transforms the images from a matrix (of 50 by 50 pixels), to a 1d-array of $50 * 50 = 2500$ elements.

After the inputs are flattened, the network consists of a sequence of two fully-connected neural layers. The first fully-connected layer is a 256-nodes Relu layer. The second (and last) layer is a single-node sigmoid layer which returns the probability that the label of this figure is 1. This process can be expressed as equations:

$$a^{[1]} = max\{0, W^{[1]T} x_{flat}\} \tag{1}$$

$$p(y = 1) = \frac{1}{1 + exp(-W^{[2]T} a^{[1]})} \tag{2}$$

The final prediction accuracy of the network on the validation set is 99.1%.

What's more, I have also tried CNN network. But the accuracy is only about 50%, which is much poorer than the fully-connected network.

5 Experiment:

In my experiment, for DBSCAN, I set $\epsilon = 3$ and $minPts = 3$, in which case the algorithm clusters activated pixels pretty well.

I created 1000 500 by 500 detector images from GEANT4 and SIXTE databases randomly for test, each of which includes 50 particle events and 50 X-ray events (left panel of Fig. 4 as an example). After processing the images using DBSCAN and the deep learning neural network, I reject the clusters whose predicted labels are smaller than 0.5. The clusters whose labels are greater than 0.5 are kept in order to ensure the X-ray information not lost.

In order to evaluate the error of my algorithm, I compare each processed image with the corresponding perfect image (generated together with the test set, but no particle included). The performance metric in this project is defined as:

$$metric = \frac{N_{output\_only} + N_{perfect\_only}}{N_{perfect}}, \tag{3}$$

where $N_{output\_only}, N_{perfect\_only}$ represent the number of non-zero pixels only in the processed image and the perfect image, respectively, $N_{perfect}$ represents the number of non-zero pixels in the perfect image. The simulation turns out that the average of the metric over 1000 test samples is equal to 97.2%.

For the purpose of giving an example, Fig. 4 shows a image before processing (the left panel) and after processing (the right panel). Compare these two images carefully, one can see that all particle tracks far away from X-ray photons are eliminated perfectly, and all photons away from particles are kept. However, if a photon overlaps with a particle, it will be possible for the algorithm to make a mistake. That is, they may be discarded together rather than kept.

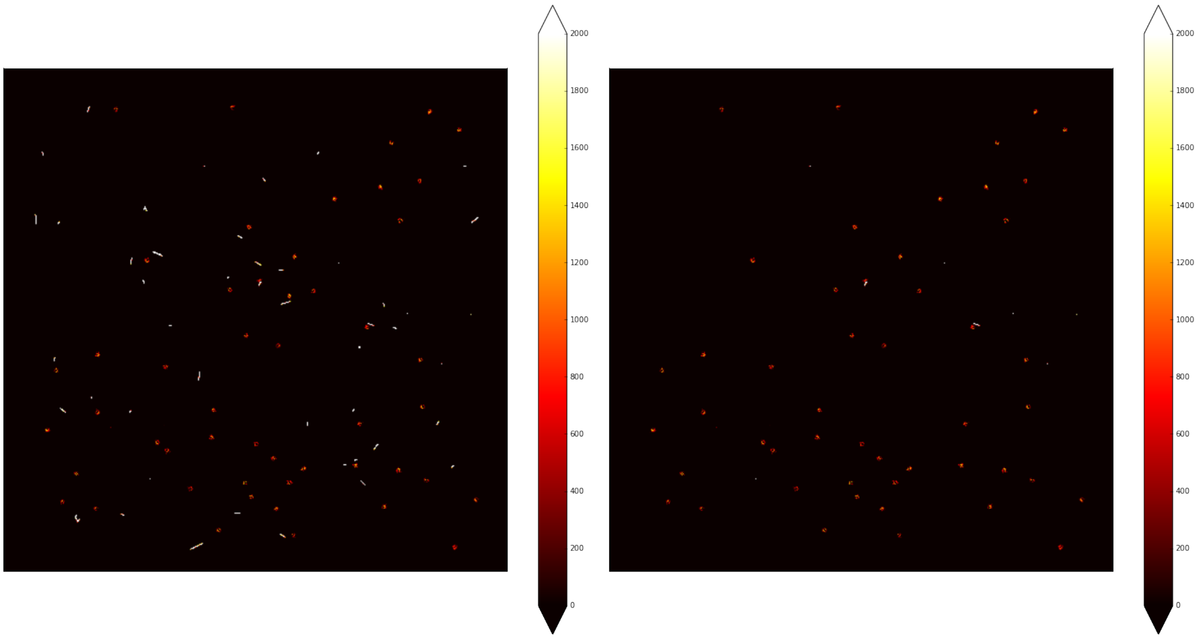

Figure 4: The left panel shows the image before processing, the right panel shows the image after processing.

6 Conclusion and Future work:

In summary, I use DBSCAN unsupervised learning algorithm to divide the raw image into clusters for the convenience of later processing. Then I use deep learning neural network to predict whether a cluster includes useful information and reject useless ones. The experiment shows that my algorithm performs very well on simulated data.

One natural future work is to run my algorithm on real data collected by other satellites. What's more, in my project, I do not modify clusters containing both particles and photons, for the sake of not losing X-ray information. Therefore, one another possible future work is to deal with those clusters very carefully to eliminate particle noises while keeping photons.

Acknowledgement

I would like to thank Prof. Steven Allen's lab for the use of the GEANT4 and SIXTE databases as well as Dr. Dan Wilkins for immensely useful discussions and advice.

Code

https://github.com/Max-Snow/Reducing-Athena-particle-background

References

[1] Meidinger, Norbert, et al. ”The wide field imager instrument for Athena.” Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray. Vol. 9144. International Society for Optics and Photonics, 2014.

[2] Burrows, David N., et al. ”The ATHENA WFI science products module.” Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray. Vol. 10699. International Society for Optics and Photonics, 2018.

[3] von Kienlin, Andreas, et al. ”Evaluation of the Athena/WFI instrumental background.” Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray. Vol. 10699. International Society for Optics and Photonics, 2018.

[4] Hall, David, et al. ”Predicting the particle-induced background for future x-ray astronomy missions: the importance of experimental validation for GEANT4 simulations.” High Energy, Optical, and Infrared Detectors for Astronomy VIII. Vol. 10709. International Society for Optics and Photonics, 2018.

[5] Bulbul, Esra, et al. ”Characterizing particle background of ATHENA WFI for the science products module: swift XRT full frame and XMM-PN small window mode observations.” Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray. Vol. 10699. International Society for Optics and Photonics, 2018.

[6] Grant, Catherine E., et al. ”Reducing the ATHENA WFI background with the science products module: lessons from Chandra ACIS.” Space Telescopes and Instrumentation 2018: Ultraviolet to Gamma Ray. Vol. 10699. International Society for Optics and Photonics, 2018.

[7] Agostinelli, Sea, et al. ”GEANT4-a simulation toolkit.” Nuclear instruments and methods in physics research section A: Accelerators, Spectrometers, Detectors and Associated Equipment 506.3 (2003): 250-303.

[8] Wilms, Jrn, et al. ”ATHENA end-to-end simulations.” Space Telescopes and Instrumentation 2014: Ultraviolet to Gamma Ray. Vol. 9144. International Society for Optics and Photonics, 2014.

[9] Ester, Martin, et al. ”A density-based algorithm for discovering clusters in large spatial databases with noise.” Kdd. Vol. 96. No. 34. 1996.

[10] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[11] Hopfield, John J. ”Neural networks and physical systems with emergent collective computational abilities.” Proceedings of the national academy of sciences 79.8 (1982): 2554-2558.

[12] Abadi, Martn, et al. ”Tensorflow: a system for large-scale machine learning.” OSDI. Vol. 16. 2016.