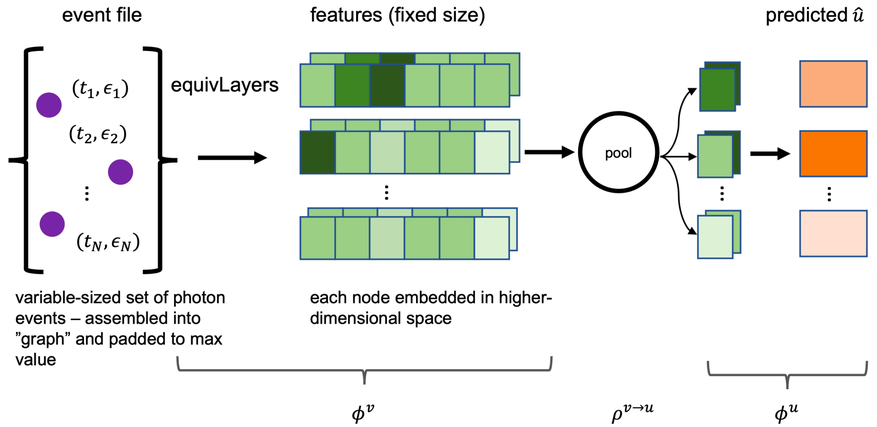
X-ray datasets:
a machine learning perspective

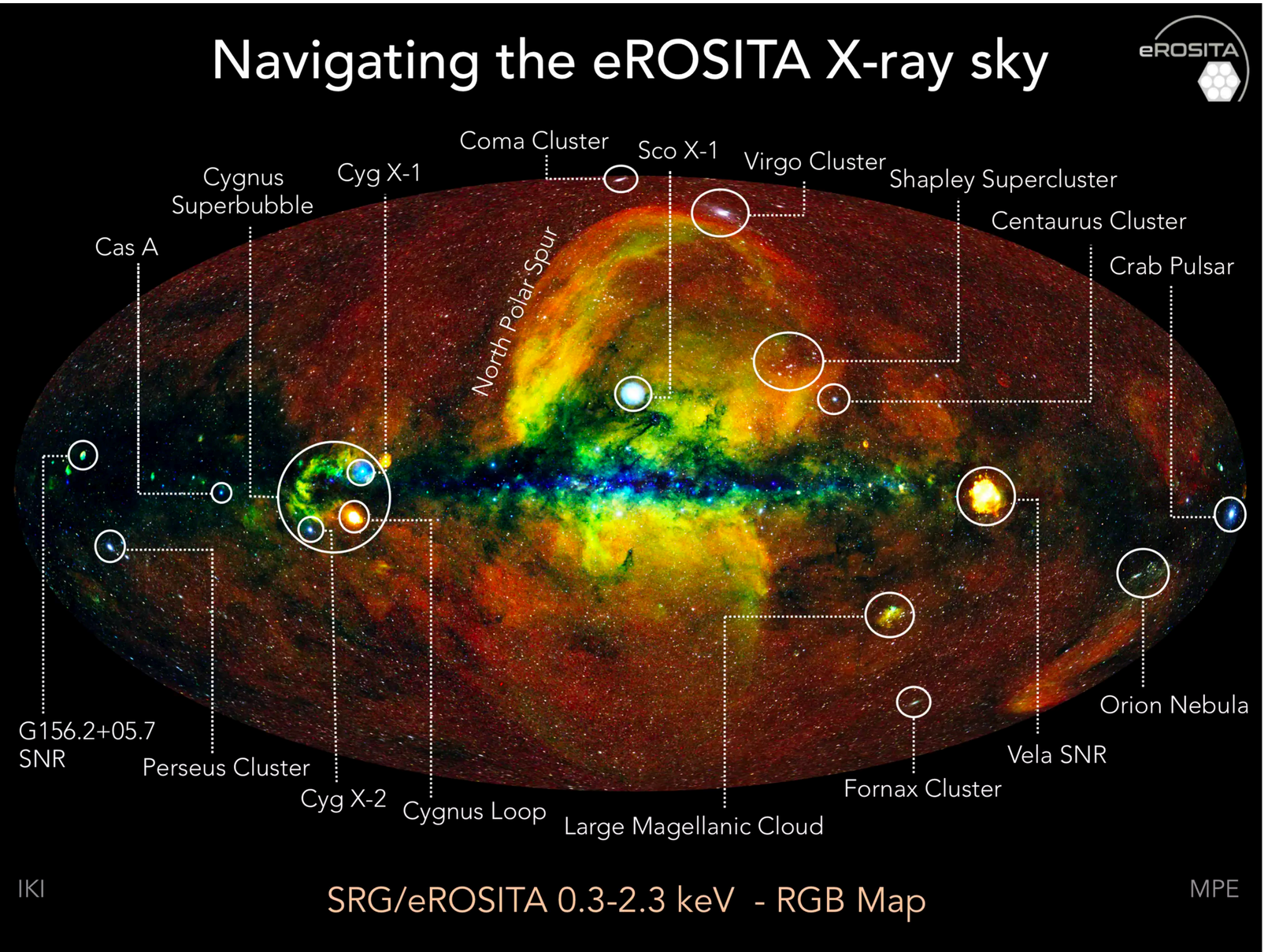
Rafael Martínez-Galarza

CENTER FOR ASTROPHYSICS
HARVARD & SMITHSONIAN



IVOA Interop Meeting - April 2022

X-ray astronomy is now also data science



- Data volume: 600/MB per day over a period of ~7yr: 1.5TB.
- Time domain aspect: about 1000 sq deg of the sky near the poles will be visited more than 30 times in the first 4 years.
- This first full sweep of the sky contains ~1 million sources, and about 165 GB of raw data, that are transformed into event files
- Early Data Release data is currently available, and includes event files, catalogs, as well as light curves and spectra.

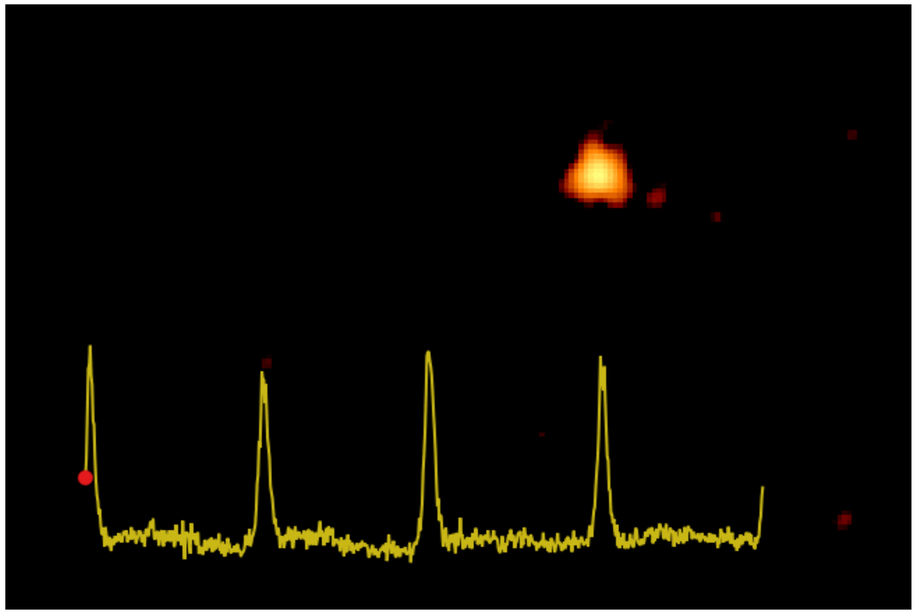
Potential applications

1. Source classification: most sources in X-ray catalogs are of unknown nature.

2. Time-domain studies: transient identification, multi-messenger astronomy.

3. Anomaly identification. Serendipitous discoveries happen often. Can we somehow make them “systematic”? (Giles & Walkowicz 2019).

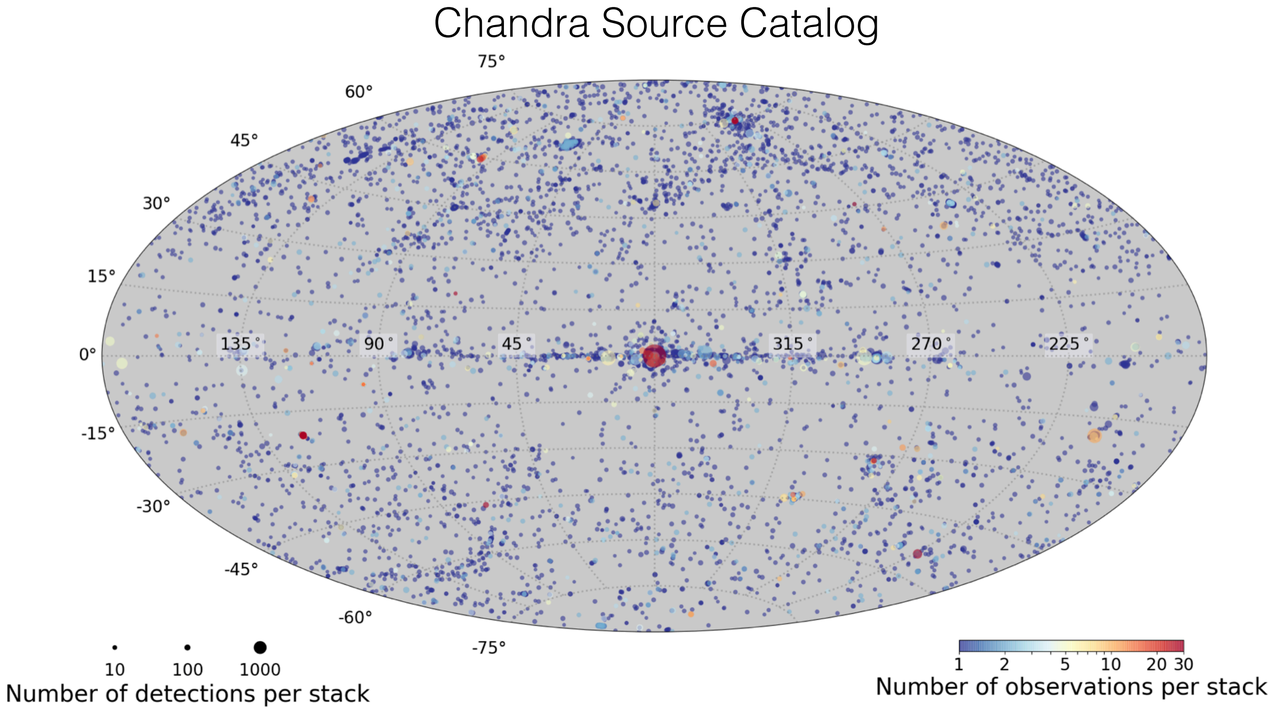
4. Statistical learning of X-ray properties. To what extent are algorithms such as neural networks useful to achieve our scientific goals?

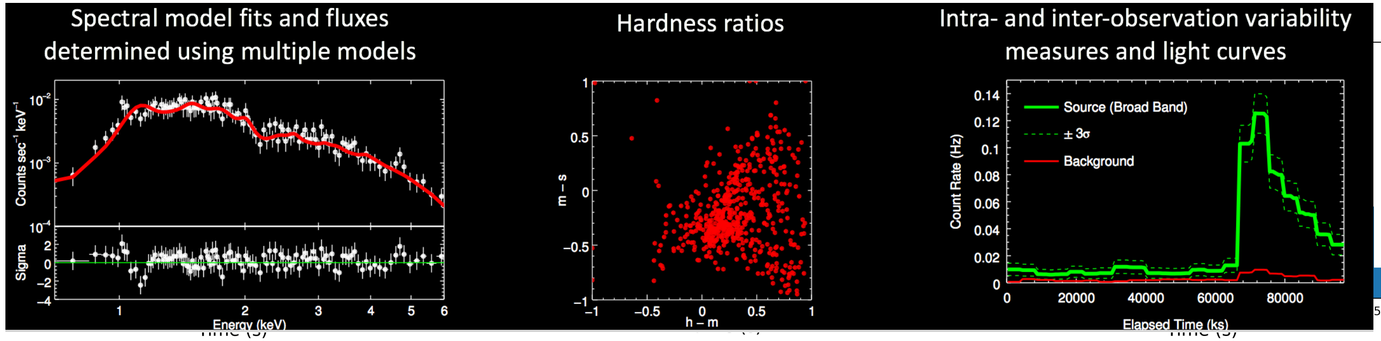


Quasi-periodic oscillations in GSN 069 likely due to a tidal disruption event “in pieces.” (Miniutti et al. 2019, Nature). This was a **serendipitous** discovery

Data format: catalogs and event files

- Both catalogs and calibrated event files can be used as input for ML algorithms.
- Catalogs compiled from event files (CSC, XMM, etc.) are in fact rich material for supervised learning.
- Yet, X-ray datasets are not like optical and IR datasets, because they record information related to INDIVIDUAL photon arrivals.
- Example: light curves.
- Most (but not all) ML methods take input examples that are of the same length. How do we deal with this in X-rays?





Example 1: source classification

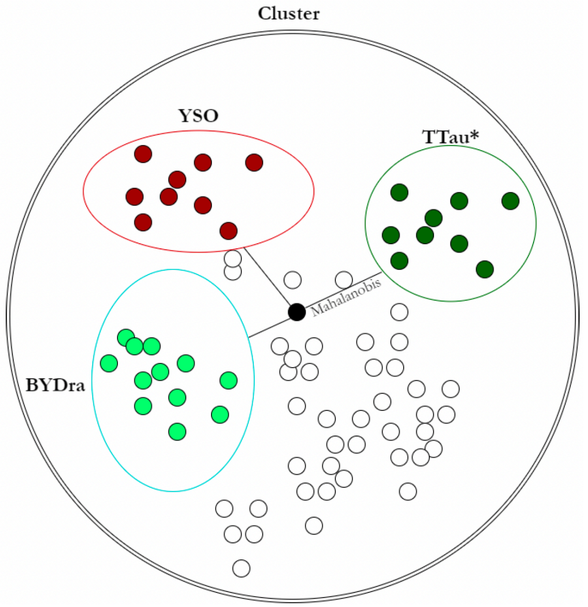
Samuel Pérez

- Input: catalog data (CSC2), together with some labels (training sets in X-rays are even less representative and even more biased).

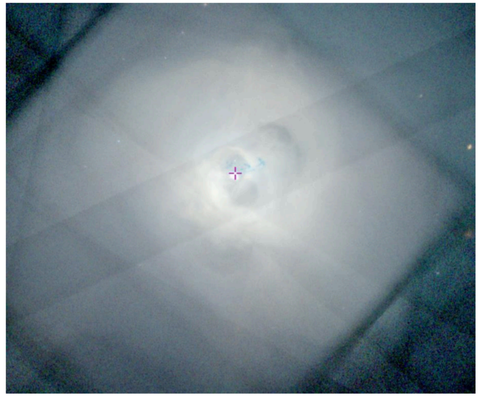
- Data access requirements: tabulated properties for each source, or for each detection (hierarchy). Ideally, accessible from TAP services, so that we can directly load them into a Python session.

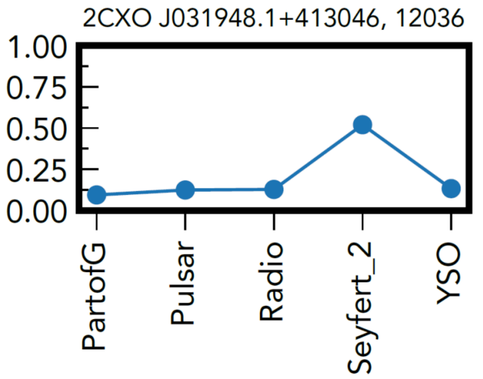
- Pre-processing: minimal. Catalog pipelines do the work for us. Limited mostly by significance of the sources. So, S/N cutoffs might be needed. Data inputation often needed

- Output: Probabilistic classification. Would be desirable to add as an extended data product to the CSC2 catalog. Does IVOA support such probabilistic approaches?



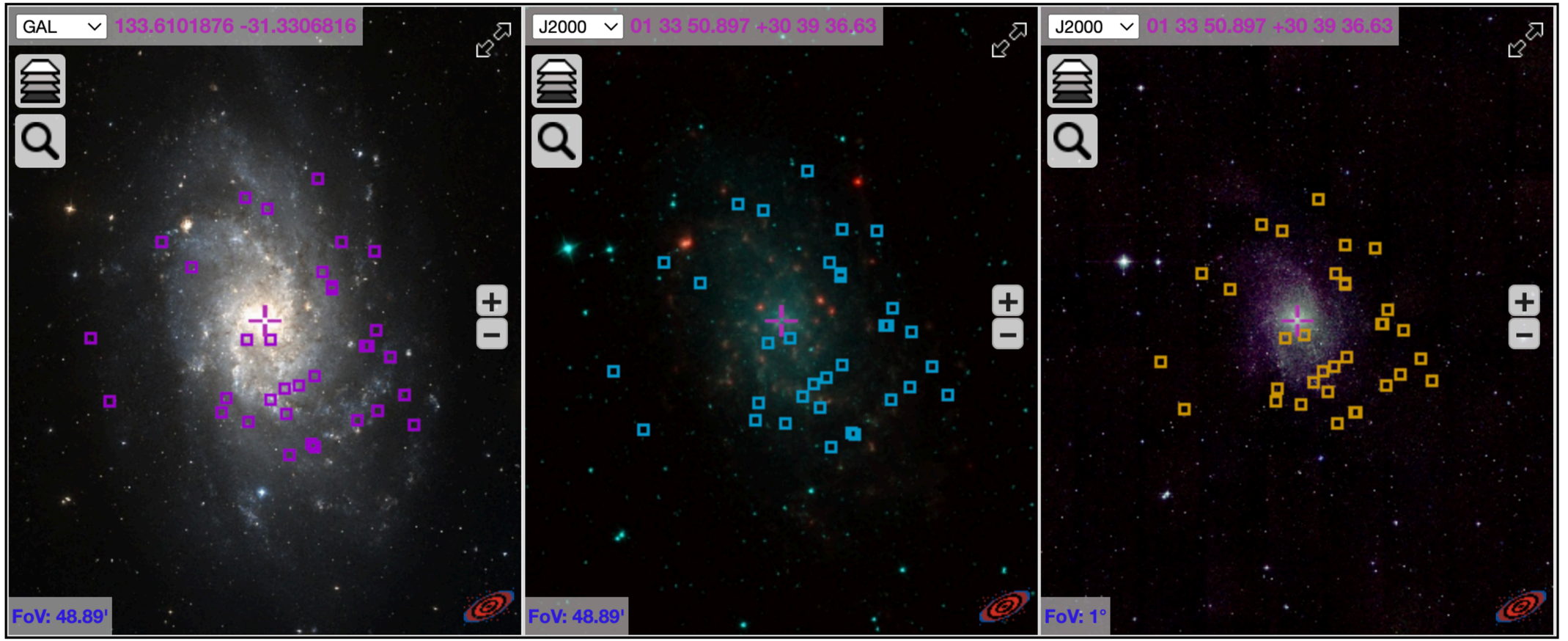
- **Validation:** our X-ray classifications are in agreement with classes of independently classified control objects (e.g., Radio galaxy 3C84)





Example 1: classification validation

Previously unclassified HMXBs in M33

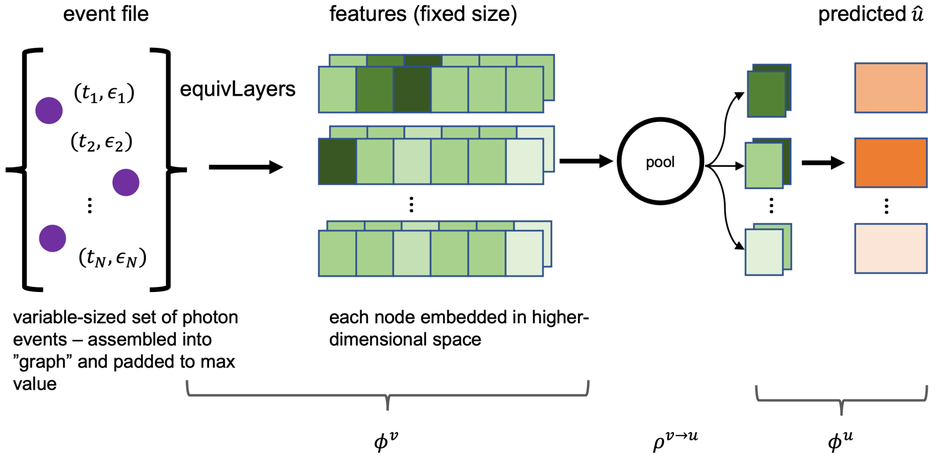


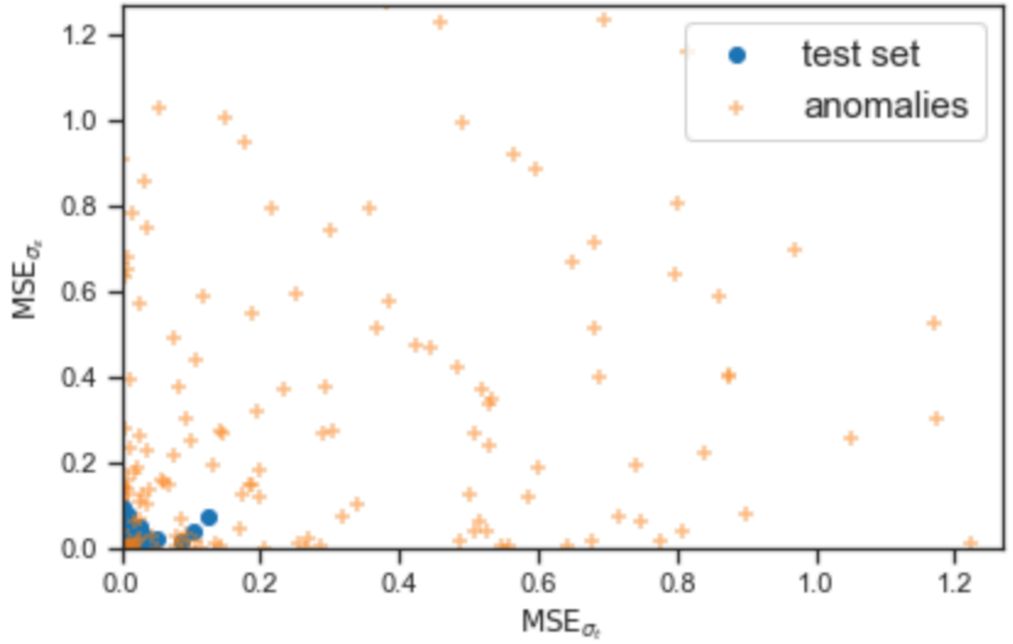
See also Yang et al. (GWU) at: https://github.com/huiyang-astro/MUWCLASS_demo_HEAD19

Example 2: anomaly detection

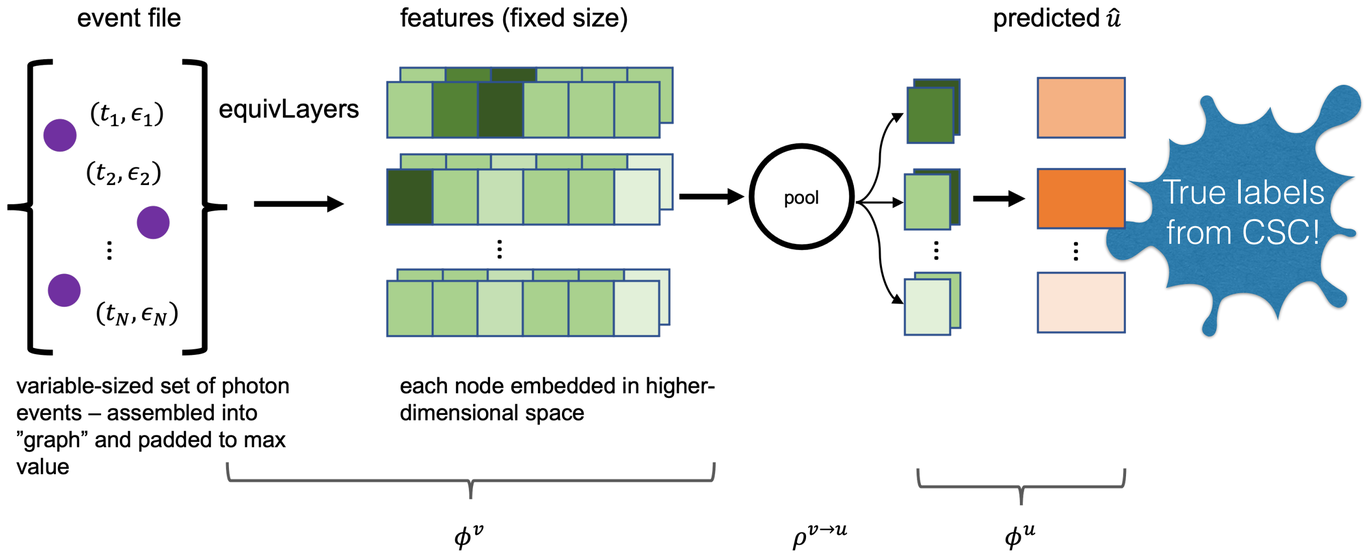
Lucas Makinen

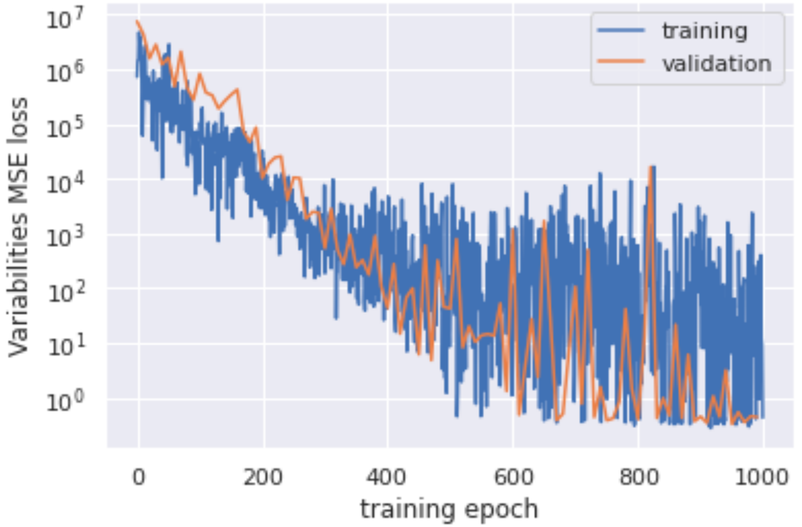
- Input: individual events (e.g.,times, energies)
- Pre-processing: event files are of different length. Possible approaches:
  - “Agregated” features: the mean, or the sum, or the total number of events, that are then passed to a neural network. These are typically permutation invariant.
  - dEdT maps: 2D histograms of the differences in times and energies between all events. Could IVOA support such representations of event files?
- Output: A prediction error that is typically larger for anomalies not seen by the network before





Learning from the events





- Some of the CSC properties are permutation-invariant with respect to the arrival times of the events (e.g. hardness ratios).
- The Deepest architecture is invariant to the ordering of the events, in a similar way as a convolutional neural network is invariant to rotation or translation of an image.
- We use a Deepsets to create a set of features from variable length objects (event files). Such features can them be used for regression or anomaly detection purposes.

Bringing it all together in one place

- When dealing with X-ray datasets, the power of ML can be maximized if there is:
  - Seamless access to the tabulated data in a hierarchical form (sources, but also their multiple detections).
  - Seamless access to the events that form the detections, and other data products, such as light curves, spectra, etc.
  - Effective data representations of event files of variable length (aggregate representations)
  - Straightforward access of analysis tools (ML packages, fitting packages) from the same platform were data are being loaded.

Accessing release 2.0 of the Chandra Source Catalog with PyVO and CIAO tools, with basic science applications using astropy and scikit-learn

Original notebook by Doug Burke, expanded by Rafael Martinez-Galarza

The PyVo package allows users to query Virtual Observatory services from Python. In this notebook we show how you can use it to query release 2.0 of the Chandra Source Catalog using the VO interfaces for tabulated properties, and CIAO tools to retrieve catalog data products. We also show how to use astropy to perform cross-matches with other catalogs, and some basic applications in machine learning with scikit-learn.

This notebook assumes you have installed CIAO 4.14 using conda, but you should also be able to do this with a CIAO installed using the `ciao-install` script, or any other Python environment as we are just going to use Astropy along with `pyvo`. The `pyyaml` poackage is needed to support writing out the file (it avoids a warning message you would get). For CIAO 4.14 installation instructions, check the CIAO pages.

Packages to install:

- `pip install astropy pyvo pyyaml matplotlib`
- `pip install pyvo`
- `pip install -U scikit-learn`
- `pip install bxa`

In [1]:

```
# Basic numeric and plotting imports

import numpy as np
import scipy

from matplotlib import pyplot as plt
import matplotlib
from matplotlib import cm
import matplotlib.colors as colors

# Astropy/Astroquery imports

import astropy
from astropy.coordinates import SkyCoord
from astropy import units as u
from astropy.io import fits
from astroquery.vizier import Vizier

# PyVO import

import pyvo as vo

# CIAO/Sherpa imports

from ciao_contrib.runtool import search_csc
from sherpa.astro import ui
```

https://cxc.cfa.harvard.edu/csc/threads/pyvoaccess/notebook.html

Thanks!