Finding **IMBHs** in Star Clusters with **Machine Learning**

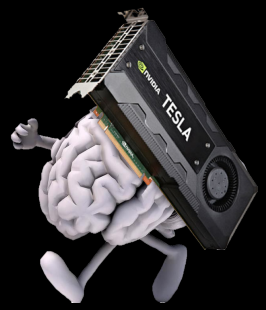
Mario Pasquato

Marie Skłodowska Curie Fellow (Astrofit 2)

Istituto Nazionale di Astrofisica

Osservatorio di Padova

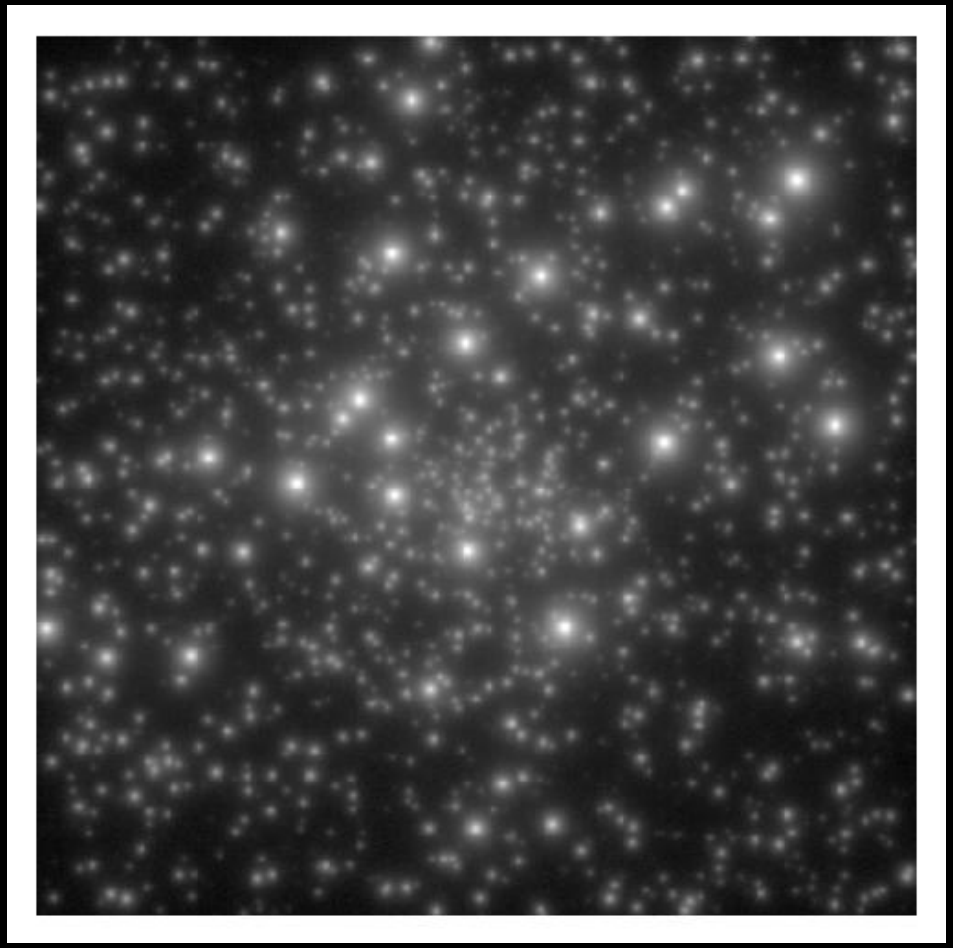
mario.pasquato@oapd.inaf.it



In collaboration with:

Mirek Giersz, Abbas Askar, Michela Mapelli, Mario Spera, Alessandro Ballone, Elisa Bortolas, Nicola Giacobbo, Ugo N. di Carlo, Maria C. Artale, Alessandra M. Battisti, Alessandro A. Trani, Federico Abbate

The problem



mario.pasquato@oapd.inaf.it

Is this the picture of an

Intermediate Mass Black Hole (IMBH) host or non-host?

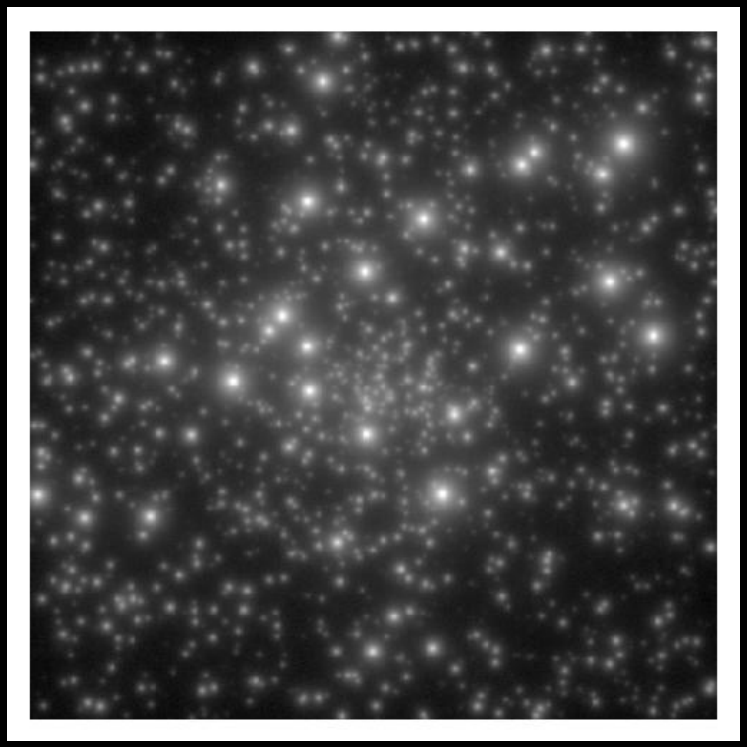
Looks like this problem

Is this the picture of a
dog or a cat?

But it's really not the same

- Dog/cat → hand-labeled data
- IMBH/no IMBH → no such data

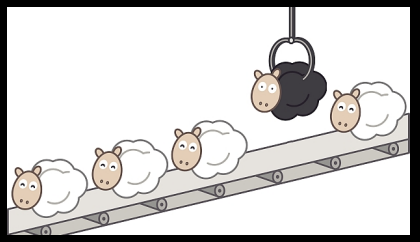
mario.pasquato@oapd.inaf.it



≠

Not the usual machine learning project

Raw data

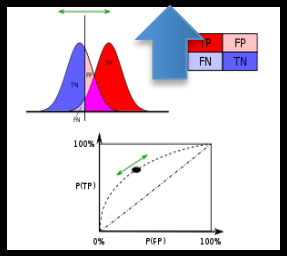


Classification

manual labeling

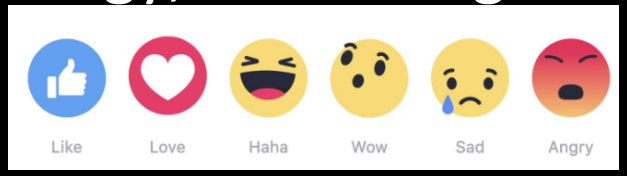
Pre-labeled data

Training



ML model

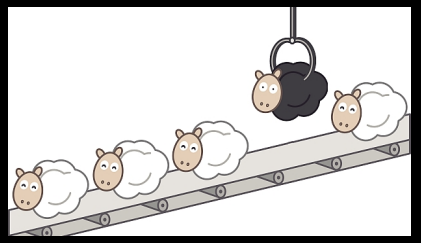
- light curve classification, galaxy morphology, face recognition, fraud detection, sentiment analysis...
- They all use hand-labeled data
- Machine learning does something that we already know how to do, just faster



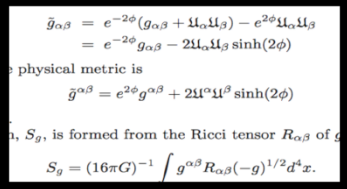
Towards automatic science

Raw data

Indistinguishable (?)



Classification

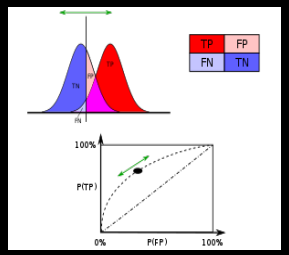


Competing theories

Simulation

Mock data

Training



ML model

- We do not have pre-labeled data
- We make mock data from simulations
- Tell dogs from cats in pictures (real data) based on drawings of cats and dogs (simulations)... better make good drawings

mario.pasquato@oapd.inaf.it

Train on this

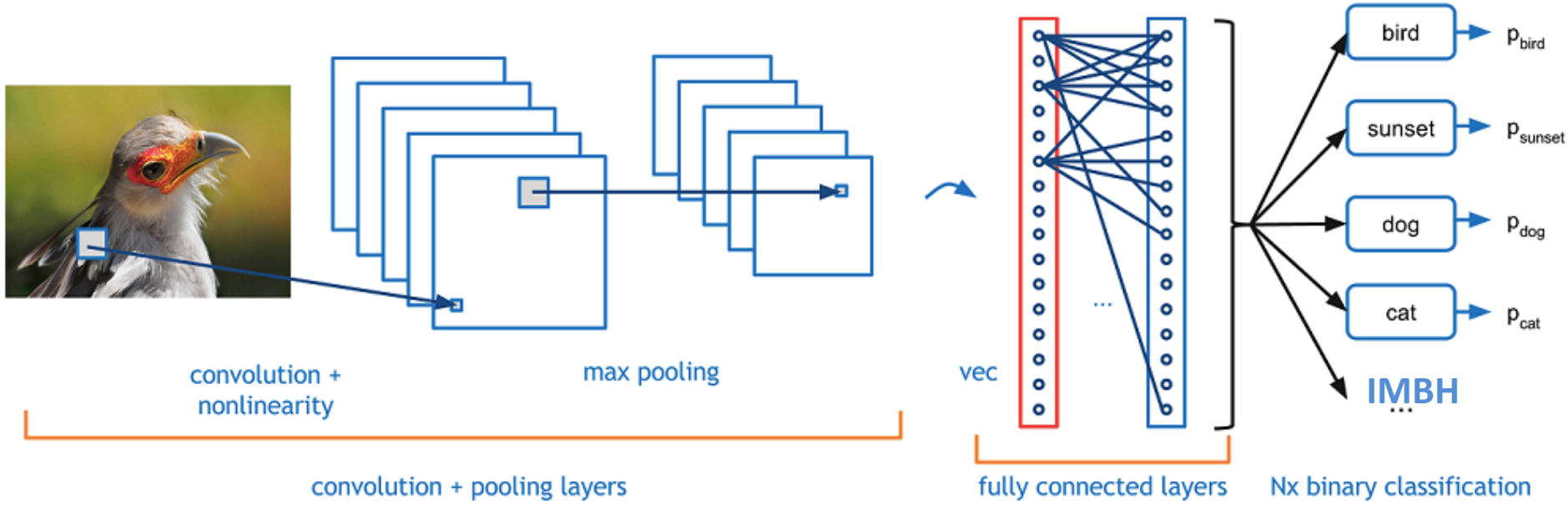
simulation

Test on this

observation

Approach I: deep Convolutional Neural Net (CNN) directly on mock images

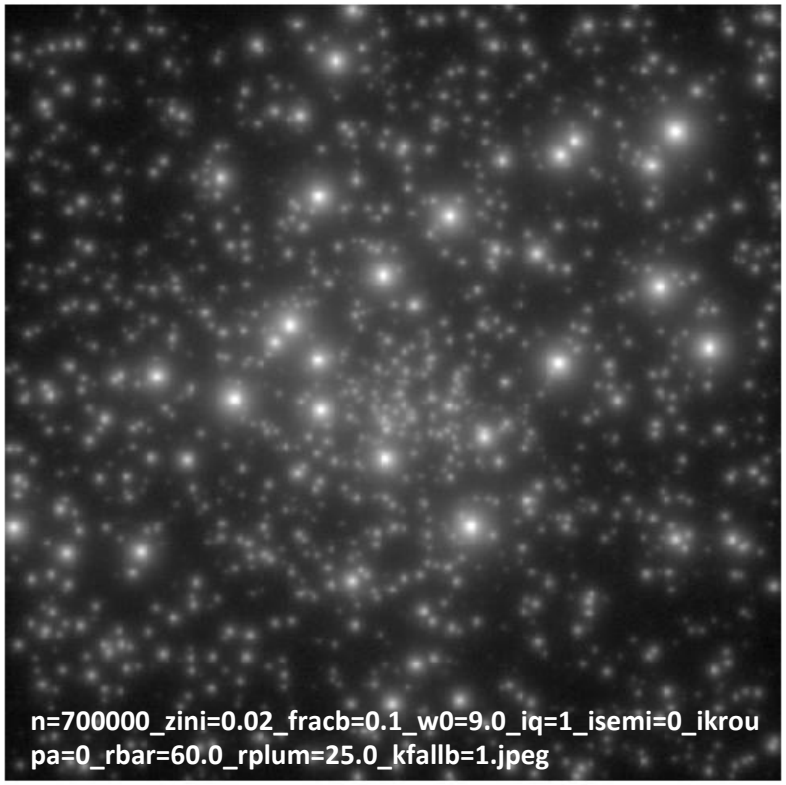
- Make mock-observations (images, velocity maps...)
- Train a convolutional neural network
- Pixel-by-pixel values are input, features are not handmade but learned explain!

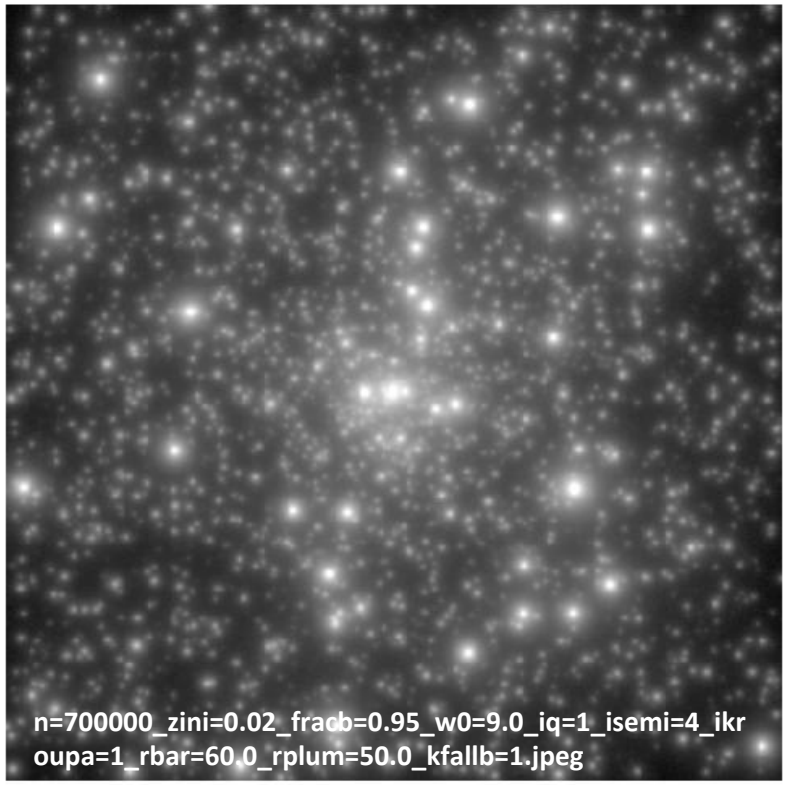


MOCCA to Mock + CNN

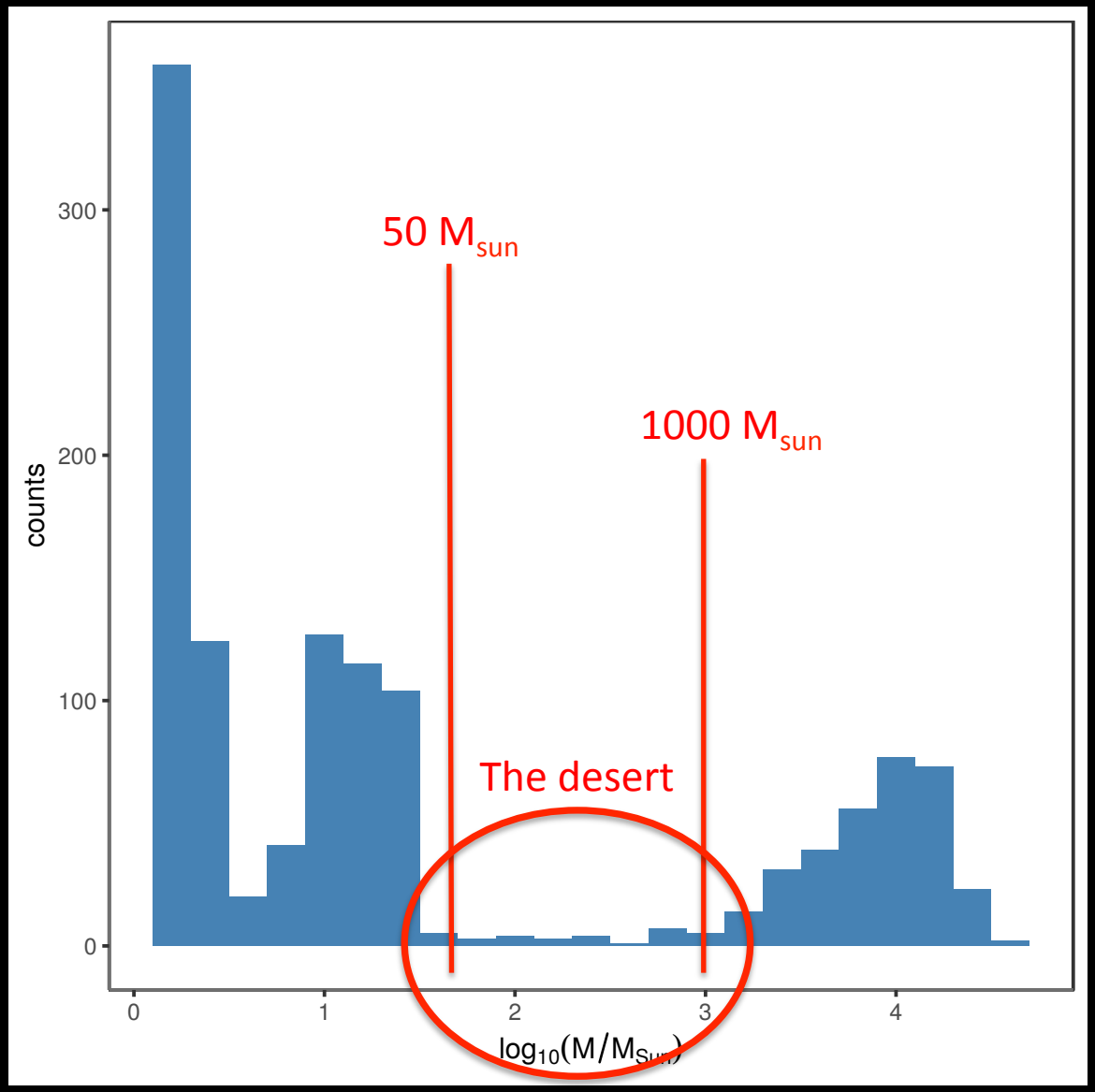
- MOCCA Survey Database simulations turned into images using COCOA (Askar et al. 2018)
- Make and train a Convolutional Neural Net (CNN) in Keras on top of Tensorflow, in python

mario.pasquato@oapd.inaf.it



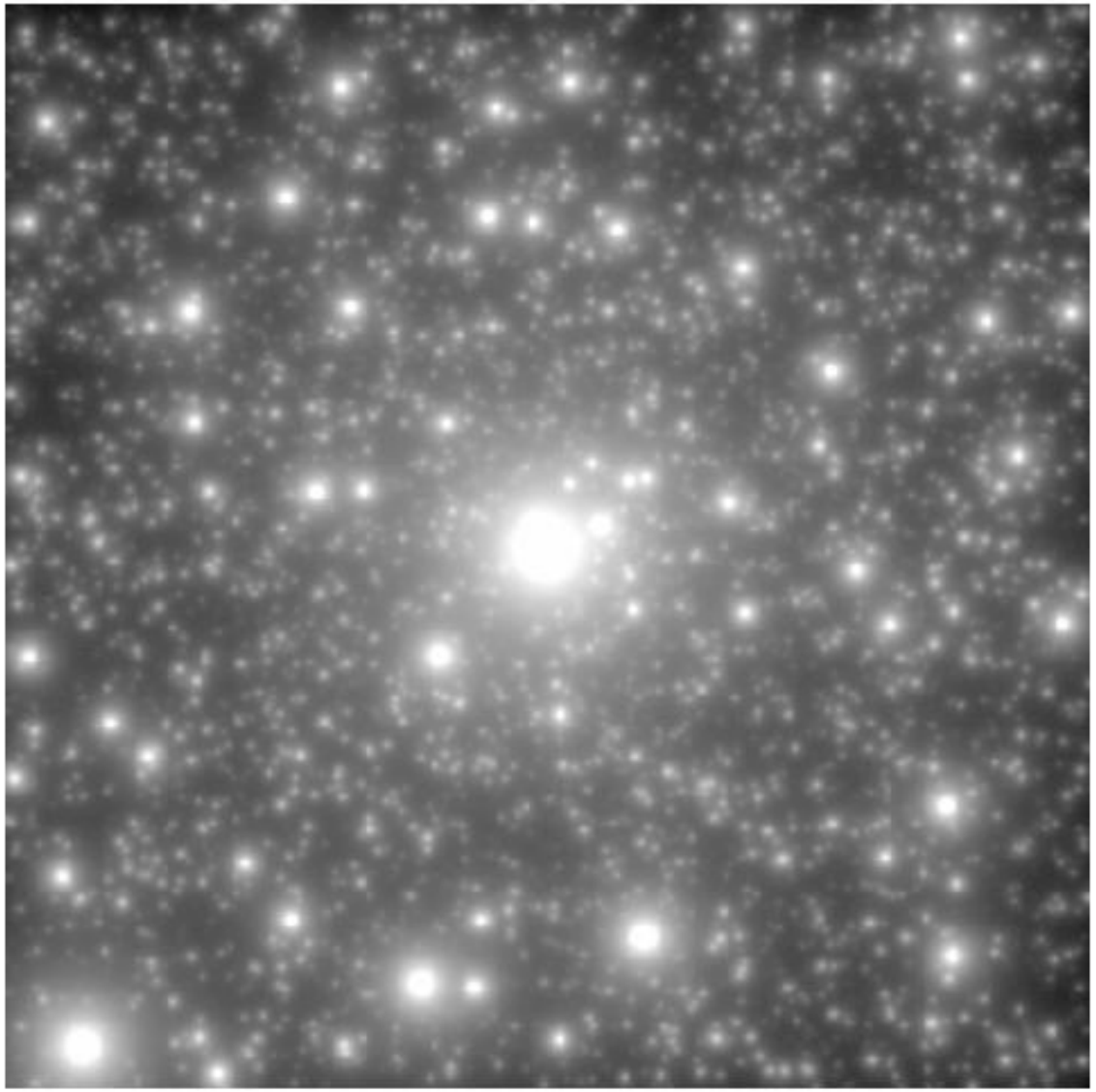


Simulations: MOCCA Survey database I



- 1237 simulations out of ~2000 survive to 12 Gyr
- 339 form an IMBH > 100 M$_{sun}$
- Clear distinction between stellar mass and IMBH → classification problem ✔

Some examples



Every snapshot at 12Gyr turned into an image using COCOA by Askar et al. 2018

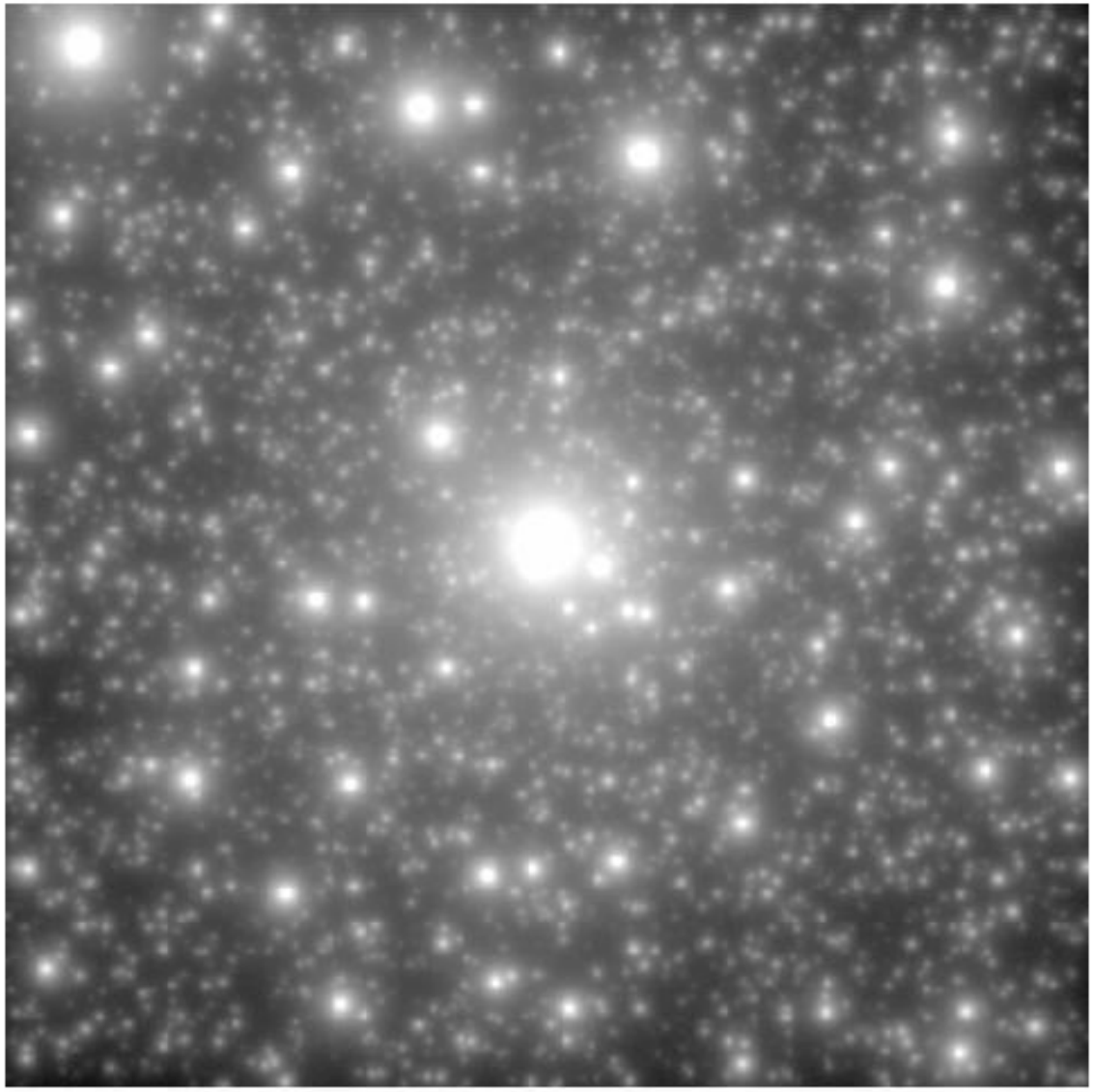
← Predicted host
(also real host)
not seen in training

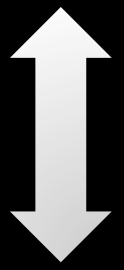
confidence*
0.811

*output of the last (softmax) layer

flip_flop_n=1200000_zini=0.001_fracb=0.95_w0=9.0_iq=1_isemi=4_ikroupa=1_rbar=60.0_rplum=25.0_kfallb=1.jpeg

Flipping/Flopping

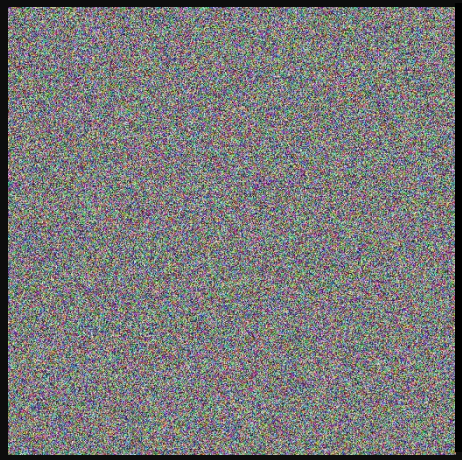




Flipped
still predicted host

confidence
0.776

Unexpected images



HOST 0.565

HOST 0.565

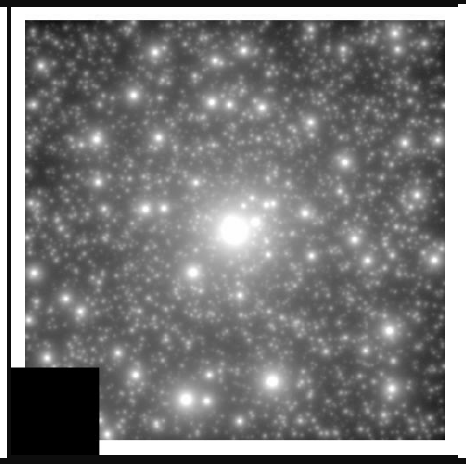
NON HOST 0.587

HOST 0.514

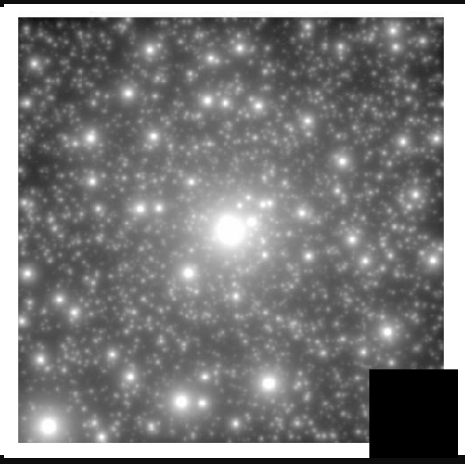
Images that we should really be unable to classify have lower confidence

Altered images

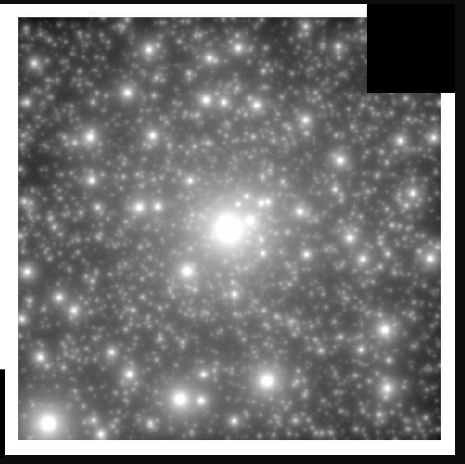
HOST 0.829



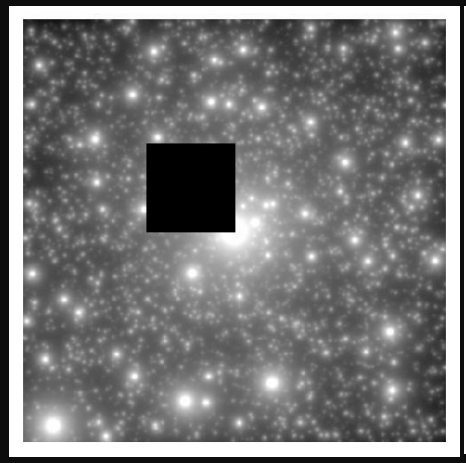
HOST 0.796



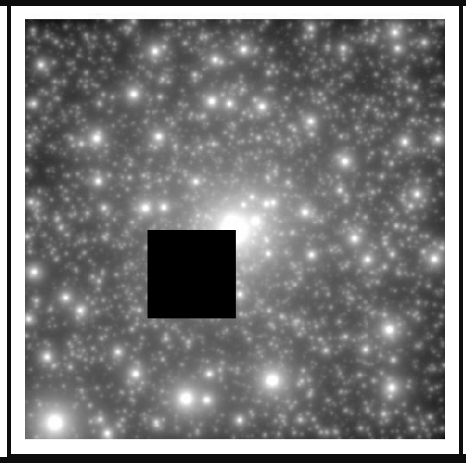
HOST 0.775



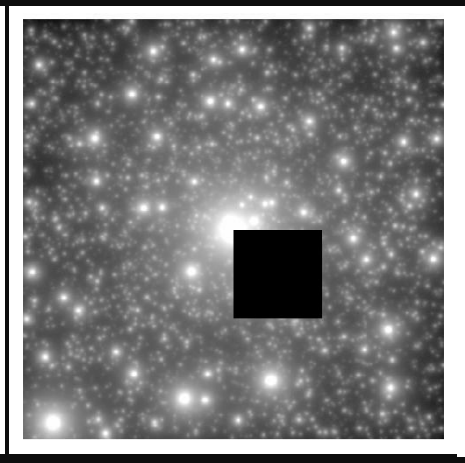
HOST 0.807



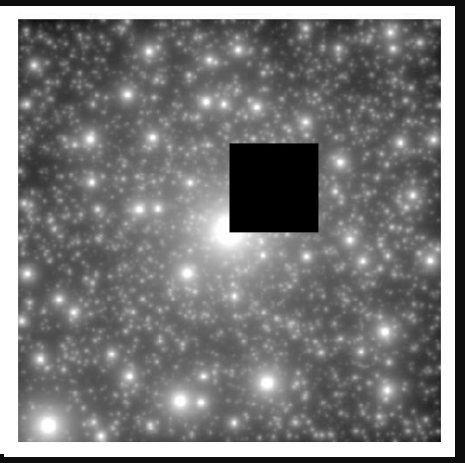
HOST 0.720



HOST 0.816



HOST 0.811

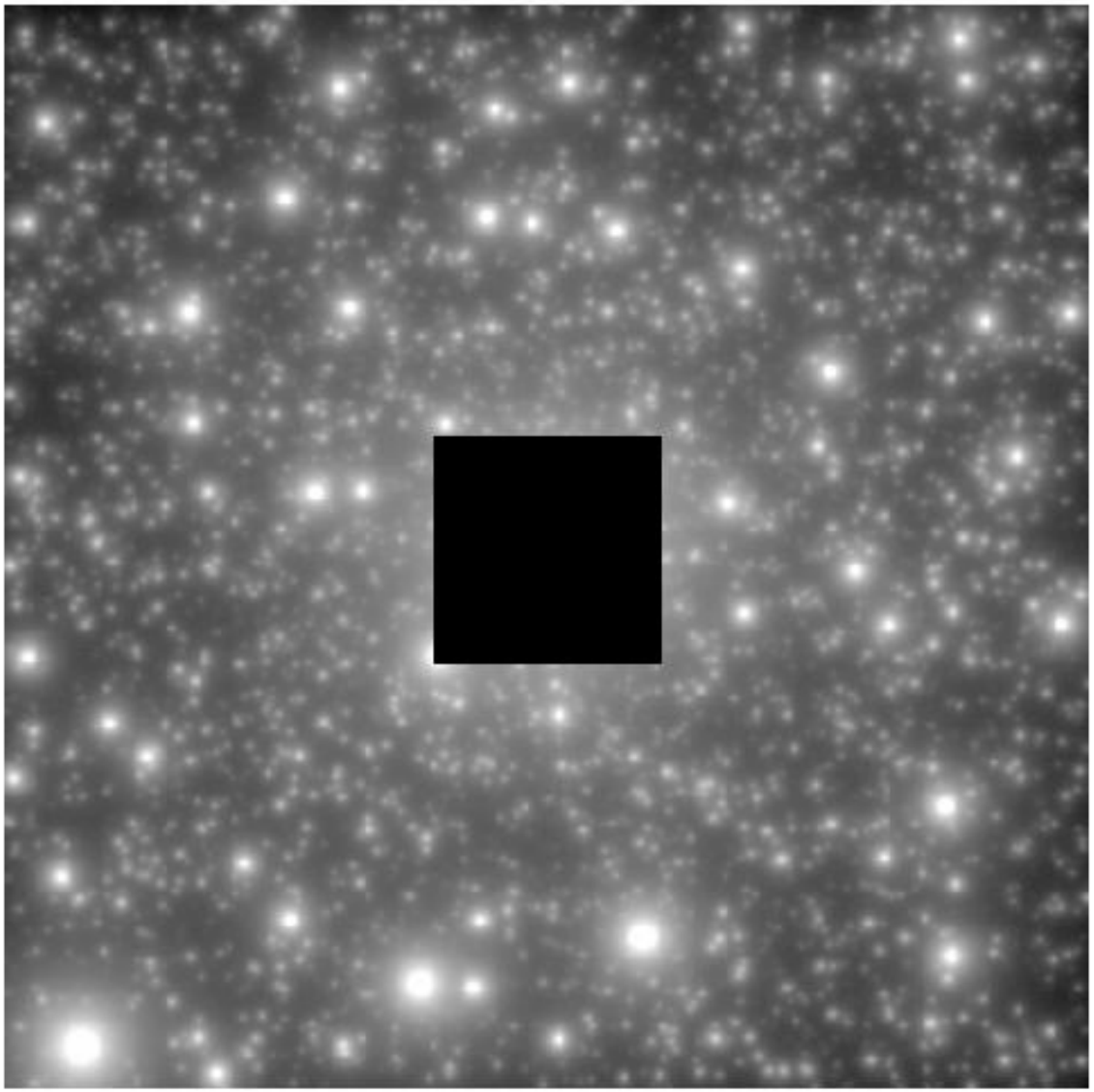


HOST 0.864

Covering parts of the original image: still pretty confident it's a host

Opening the black box

with a black square



Covering the center

NON HOST!

confidence

0.644

Important information encoded in the center

Nice pictures, but?

mario.pasquato@oapd.inaf.it

- Convolutional Neural Net approach still a work in progress
- I also have results...
- ...from a different approach
- Feature based approach

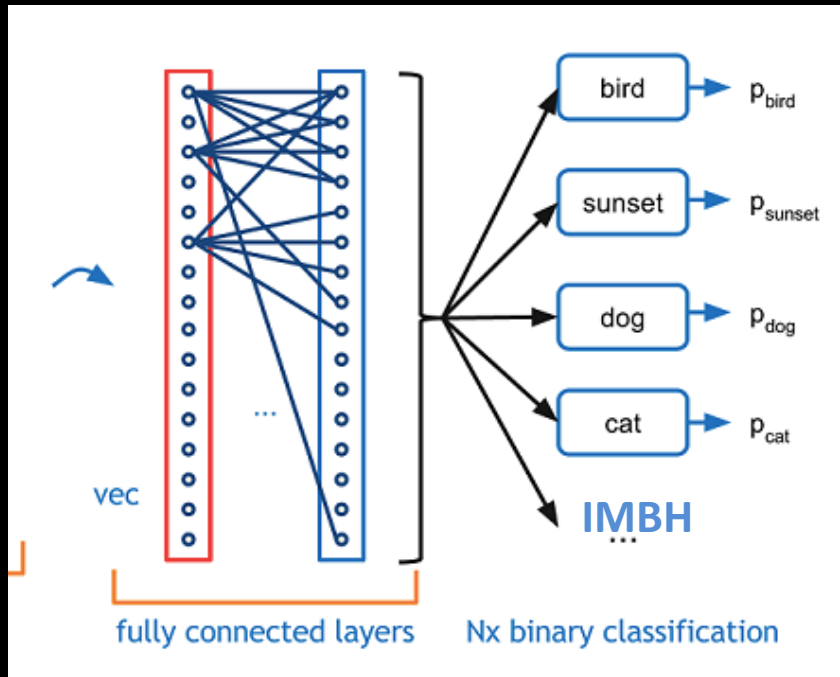
Pasquato, Askar, Giersz & Mapelli 2018, MNRAS almost submitted

classified as non-host

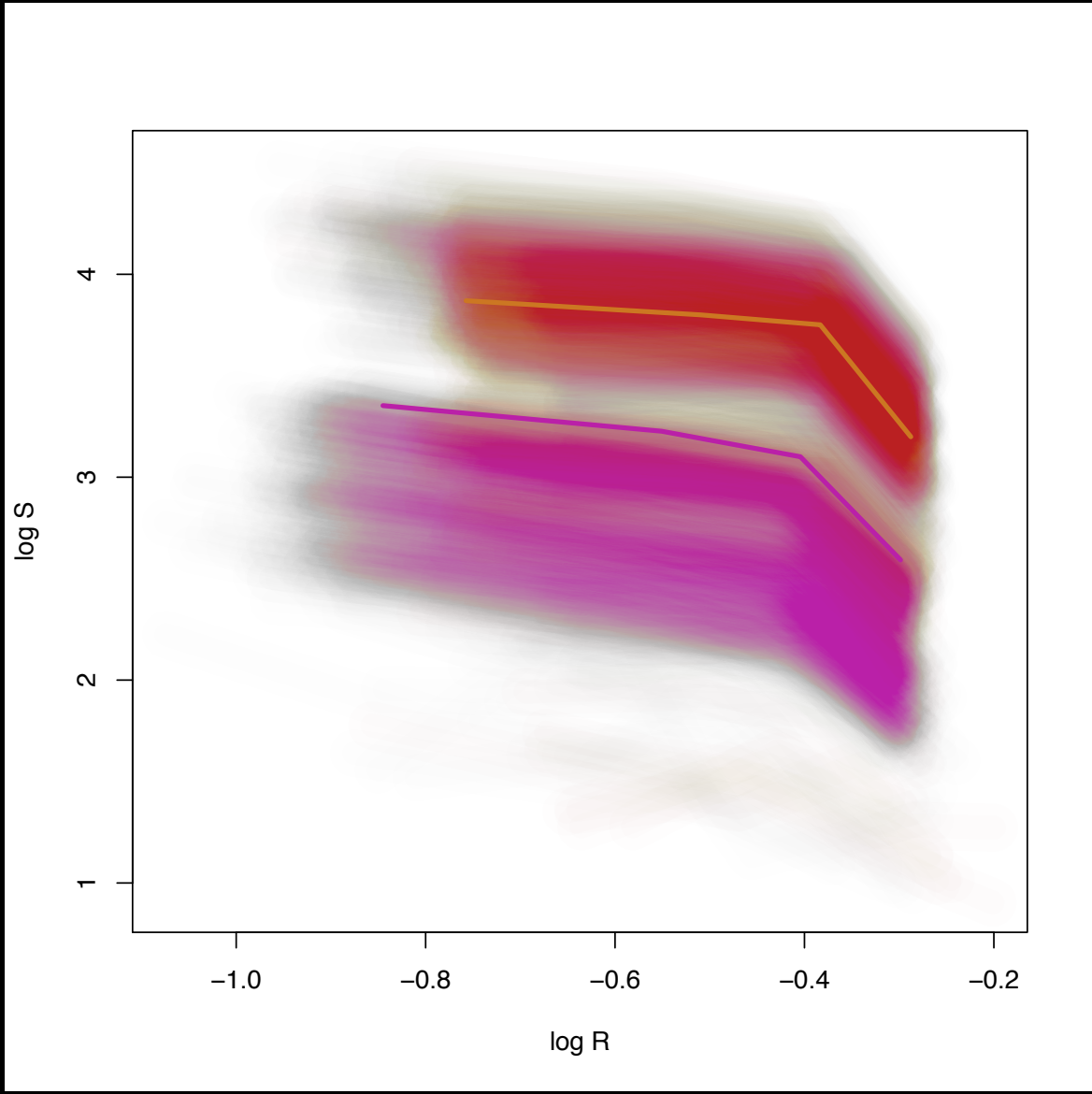
Approach II: handmade features + conventional machine learning

- Make mock-observations (images)
- Turn images into numbers (e.g. surface density profile)
- Learn on these numbers

Feature engineering



Approach II: handmade features + conventional machine learning



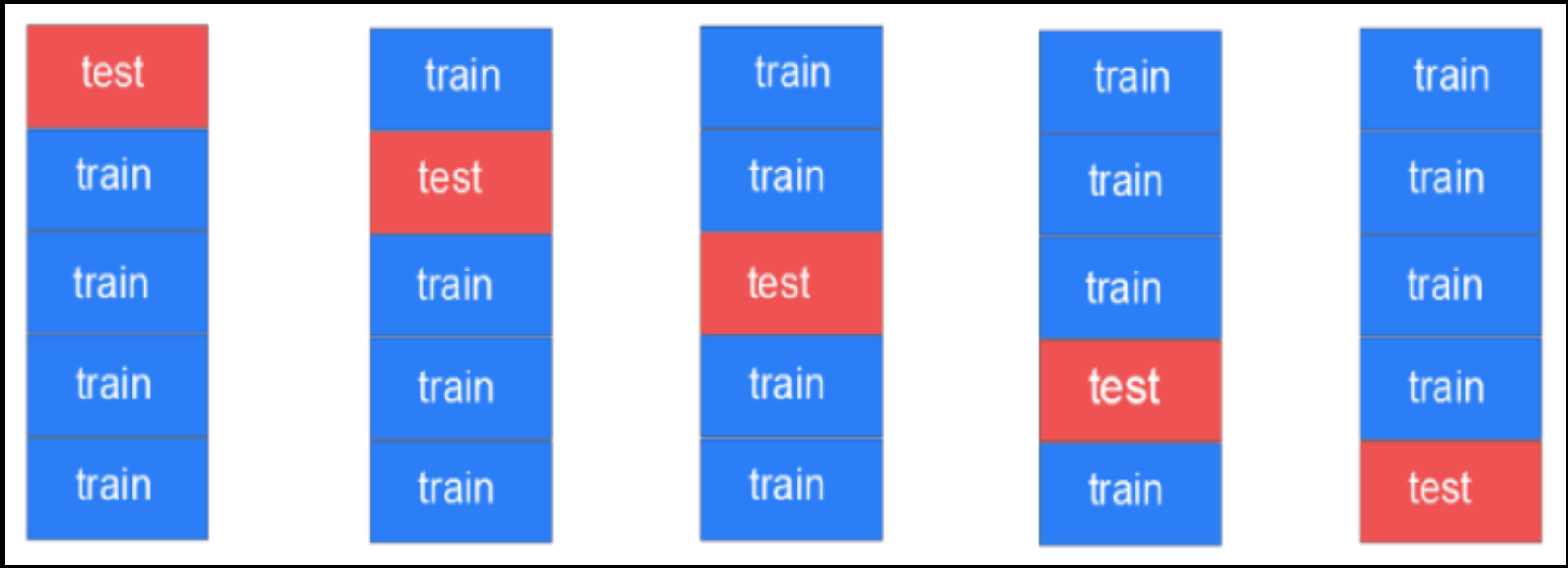
- Manually calculate features (numbers associated to each image)
- e.g. surface density profiles
- Each image → N numbers
- Train classifiers on these N numbers

Building models
(R package::function)

- k-nearest neighbor (FNN::knn)
- fully connected neural network (h2o::h2o.deeplearning)
- support-vector machines (e1071::svm)
- random forest (party::cforest)

mario.pasquato@oapd.inaf.it

Testing models: cross validation



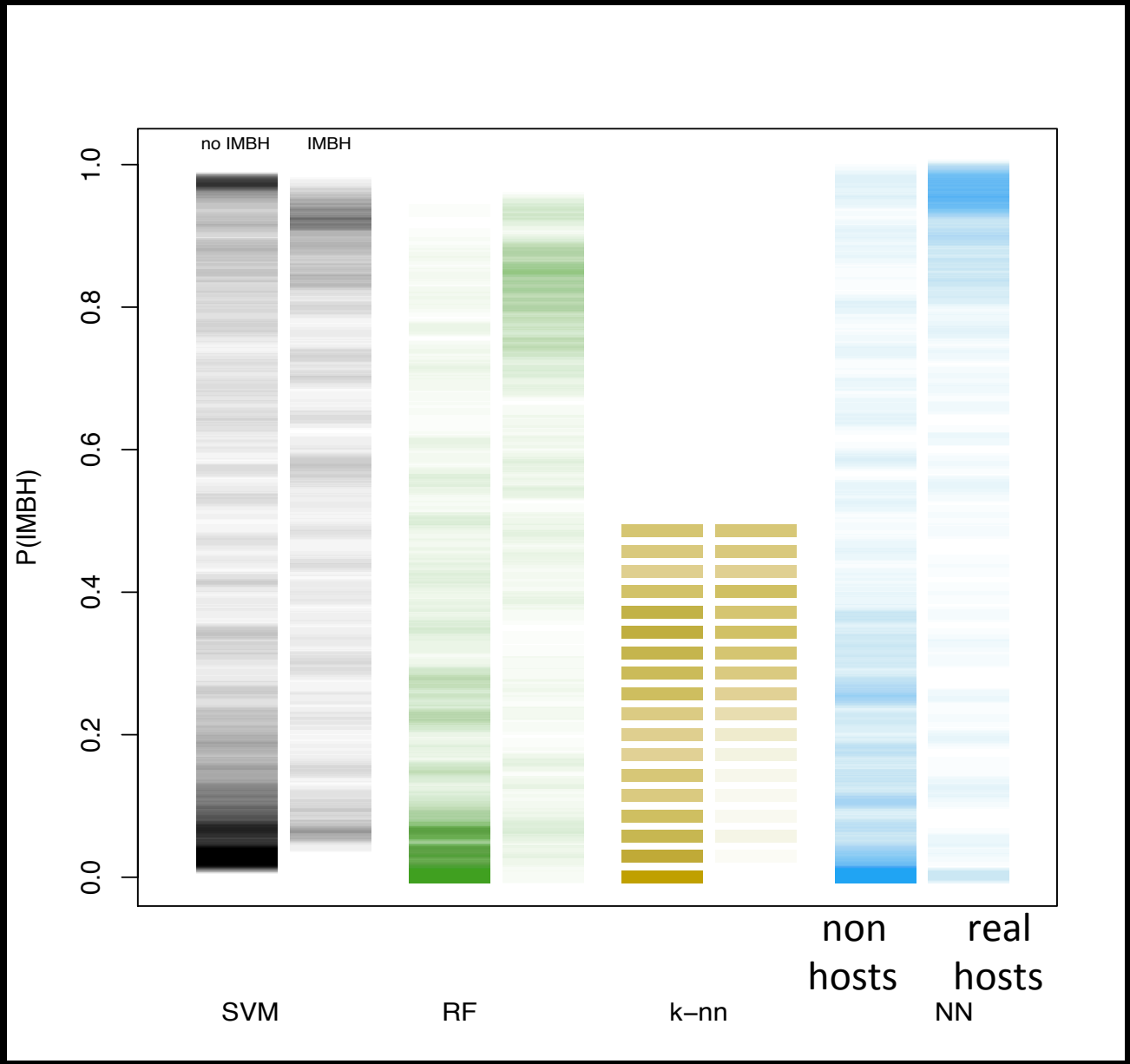
- Evaluate the trained model on unseen data, but still use all data we have (here data = mock observations)
- Split data, use subset for training and complement for testing
- Loop over data (here five times, 5-”fold” CV)

mario.pasquato@oapd.inaf.it

Predicting probabilities of hosting an IMBH

mario.pasquato@oapd.inaf.it

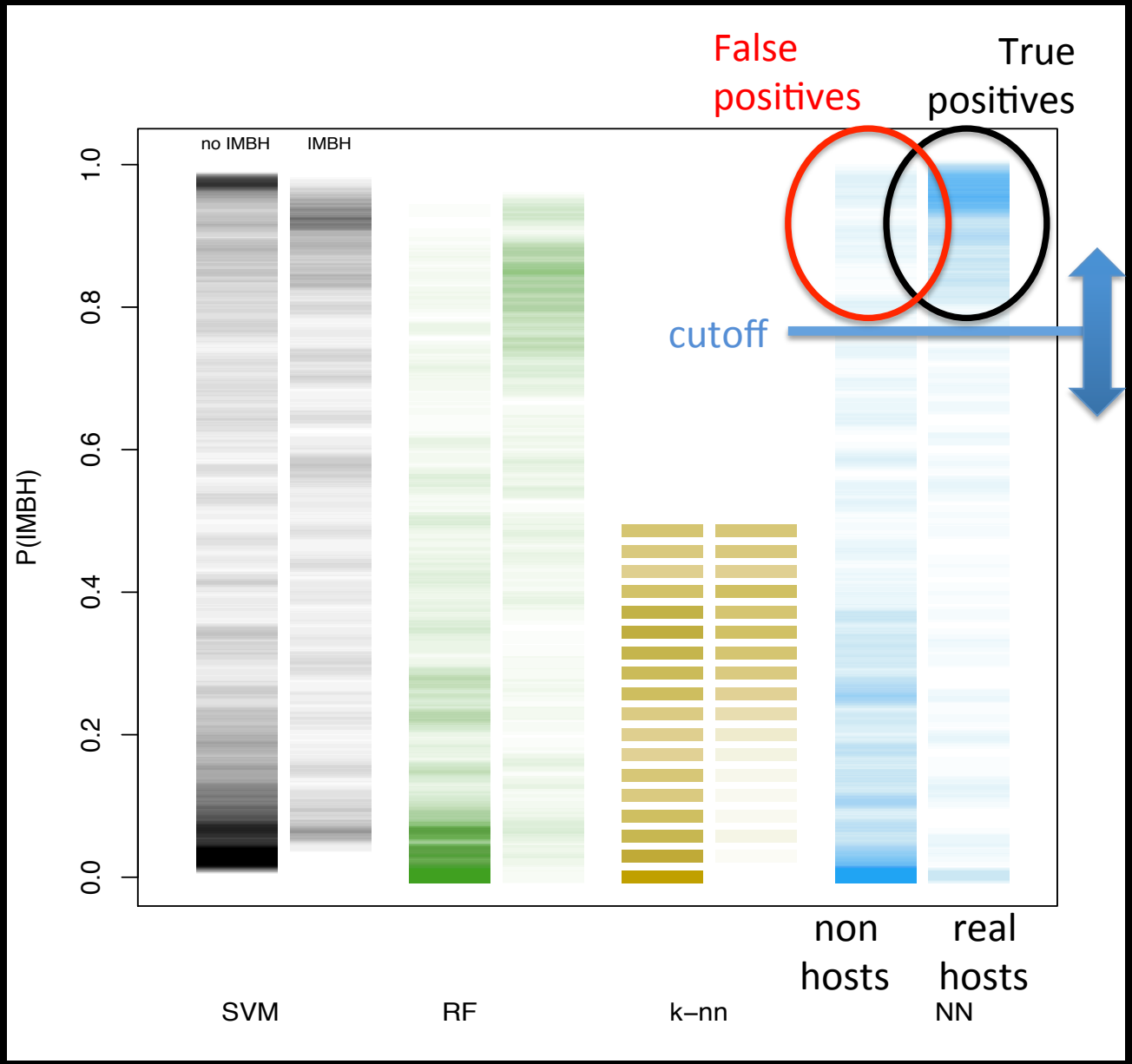
probability of being a host



different models →

Probabilities → classification

mario.pasquato@oapd.inaf.it

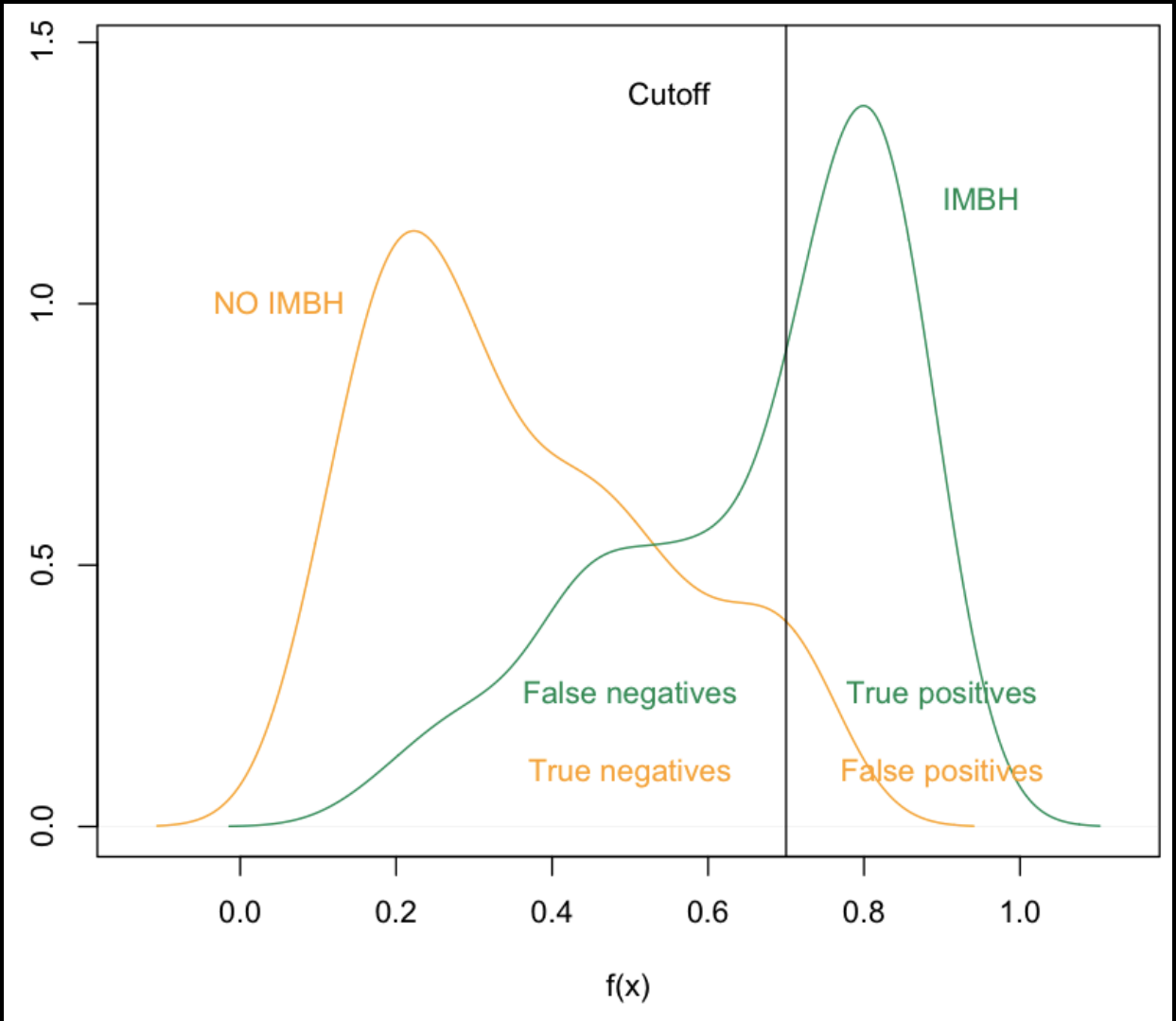


moving the cutoff up: less false positives less true positives

moving the cutoff down: more false positives more true positives

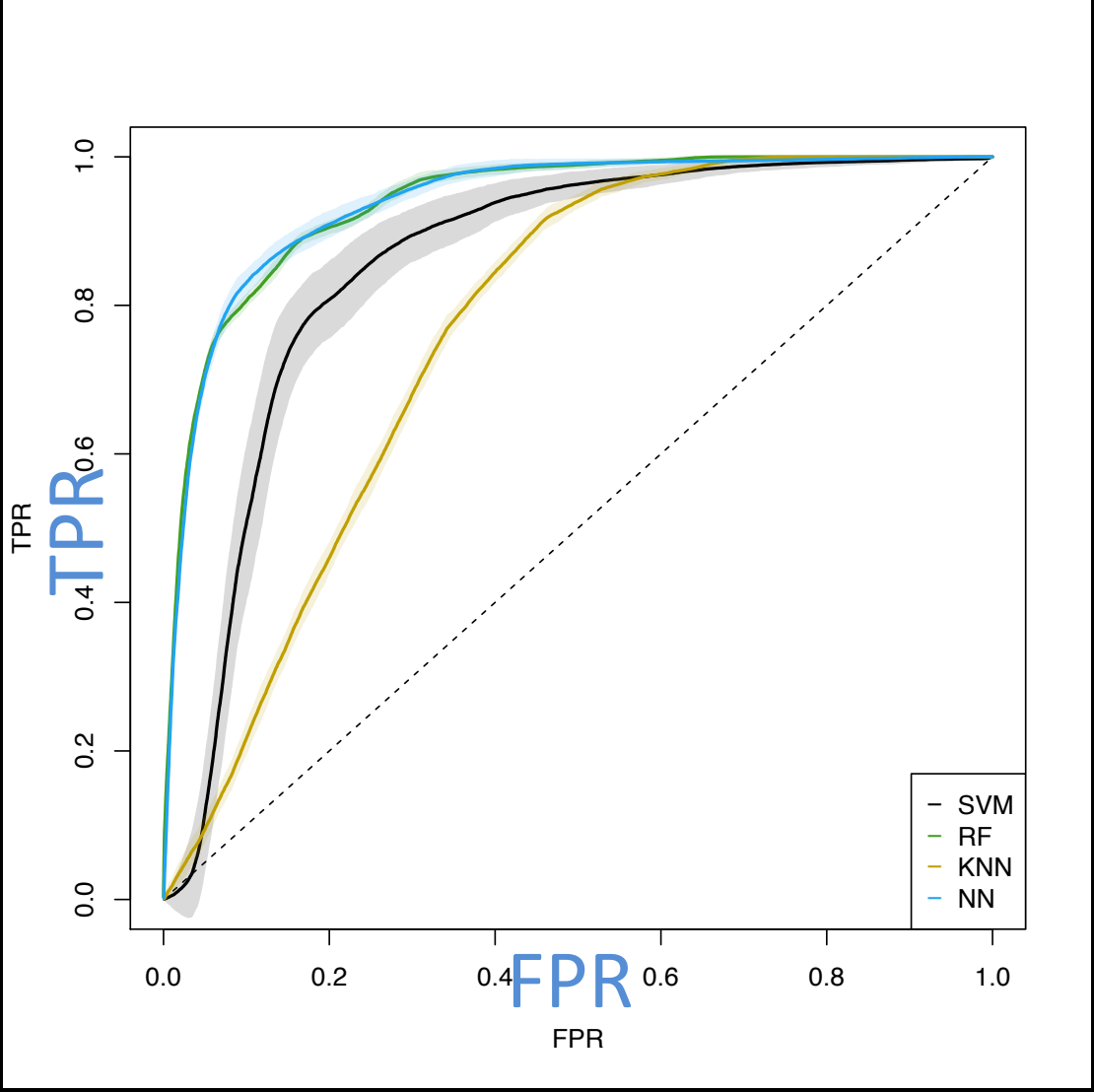
Measuring preformance: ROC curve

mario.pasquato@oapd.inaf.it



Receiver Operating Characteristic (ROC) curve:
fraction of true positives VS fraction of false positives

ROC curves for our classifieres



- How many true positives you get by accepting a given rate of false positives
- Random classifier: diagonal line
- Perfect classifier: step to 1 immediately
- Area Under the Curve (AUC) = overall performance

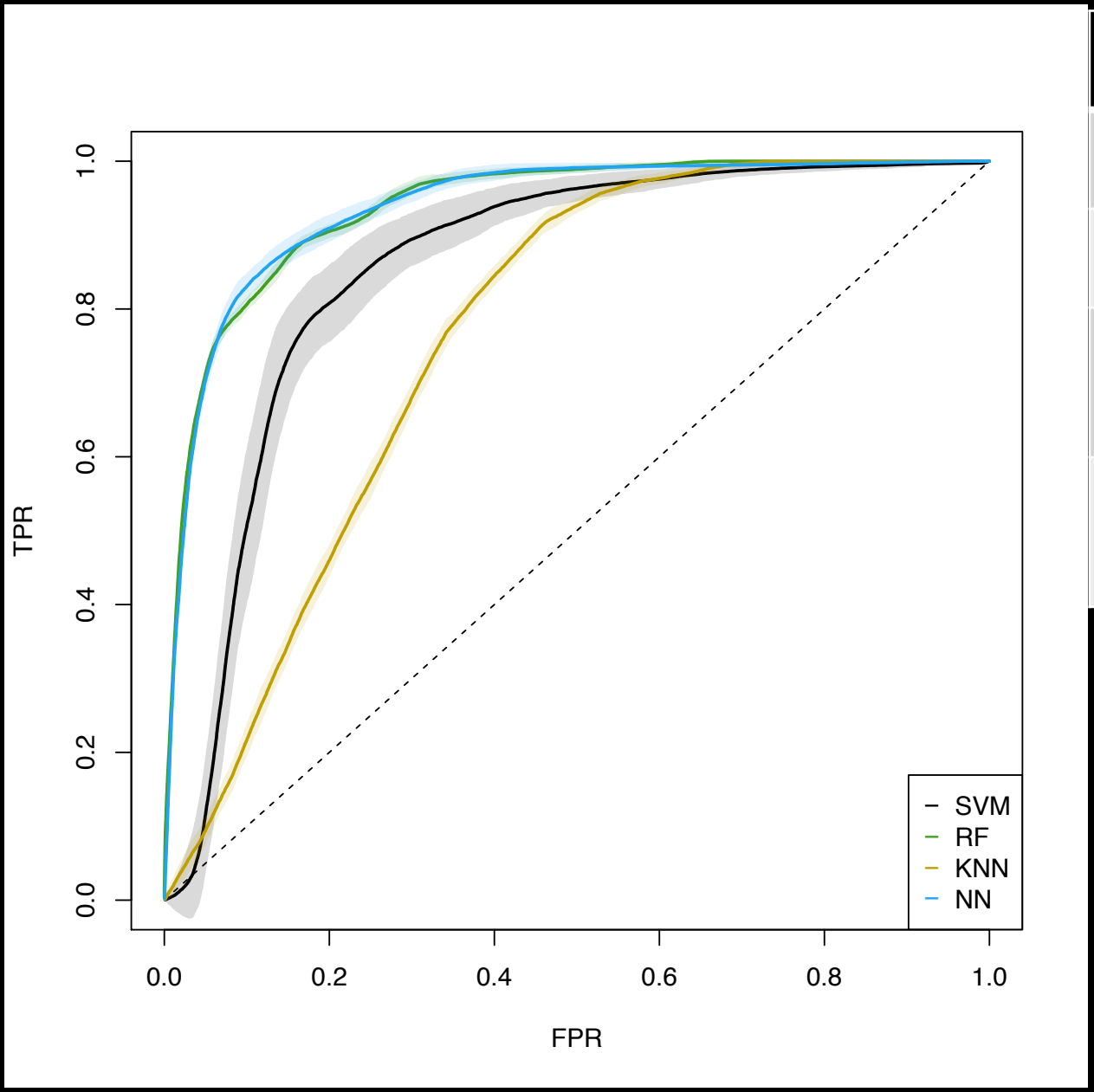
False positives / True positives

- False positive rate: fraction of non hosts that is wrongly classified as host
- True positive rate: fraction of hosts that is correctly classified as host

Real hosts	Real non-hosts	False Positive Rate	True Positive Rate	Claimed hosts	Correct claims	Correct claims / total
100	100	10%	80%	90	80	89%
100	100	50%	99%	149	99	66%
100	100	5%	50%	55	50	91%
200	0	10%	80%	160	160	100%
0	200	10%	80%	20	0	0%

Unknown Unknown Choose this Get this from ROC

ROC curves for our classifieres



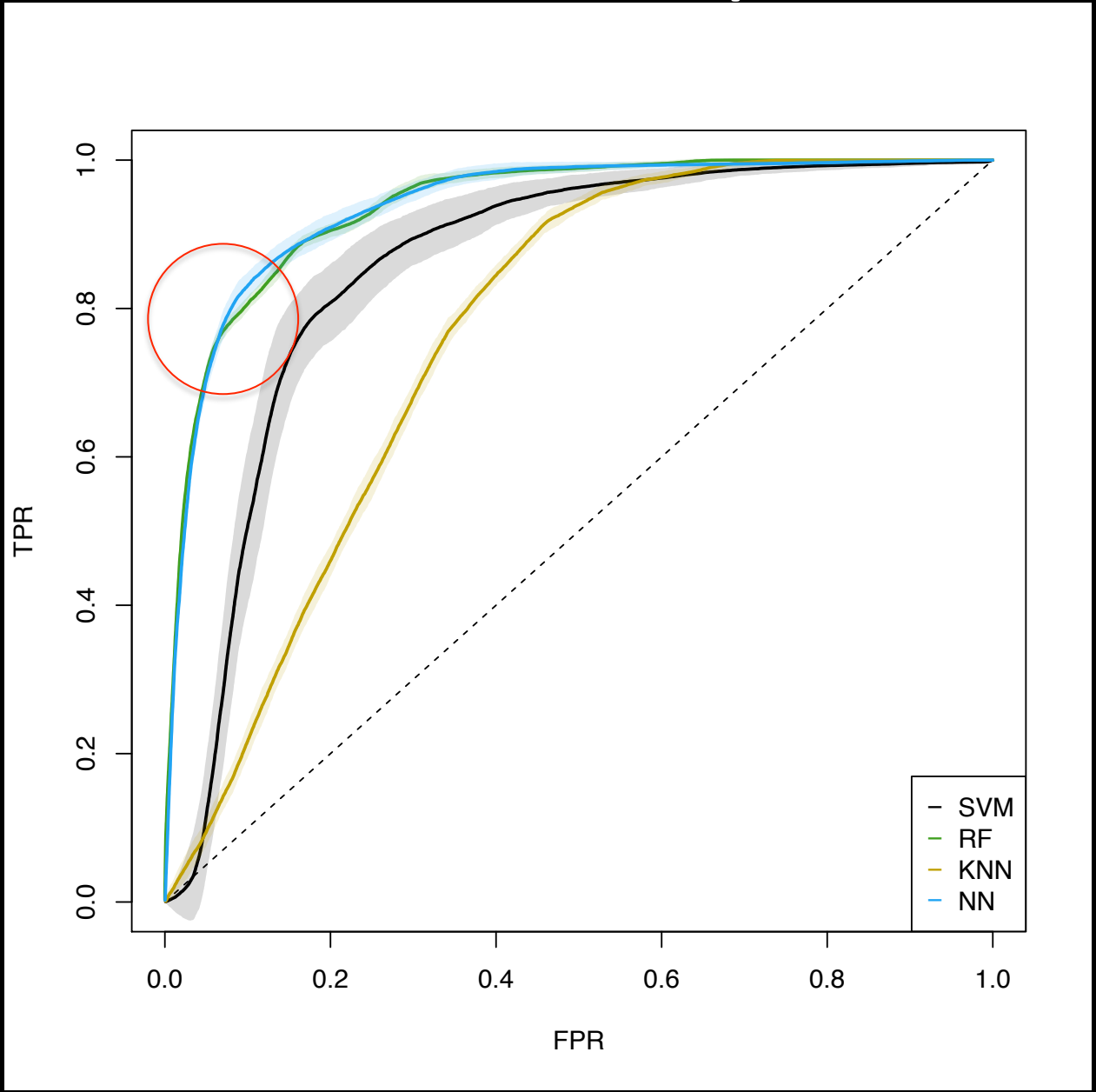
Model	AUC
Neural net	0.94
Random forest	0.94
Support vector machine	0.85
k-nearest neighbor	0.76

Perfomance of the two best classifiers (NN and RF) is very similar

No ad hoc tuning of classifier parameters

Performance measured on snapshots not seen in training

Interpretation



If we, say, accept
10% false positives

With the random forest
or the neural net

we get >80% sensitivity
i.e. 80% of the actual
IMBH hosts are actually
found

Results

- Four classifiers: the best two (neural net, random forest) have very similar ROC curves
- At 5% FPR they yield 70% TPR
- Scenarios
  - IMBH prevalence 50%
  - IMBH prevalence 10%

Real hosts	Real non-hosts	False Positive Rate	True Positive Rate	Claimed hosts	Correct claims	Correct claims / total
100	100	5%	70%	75	70	93%
20	180	5%	70%	23	14	61%

To summarize

Based only on surface density profiles,
within the MOCCA Survey database,
our classifiers
without fine tuning,
on snapshots not seen in training,
catch 70% of IMBH hosts
with a 5% false positive rate

Need more machine learning in your project?
drop me a line → mario.pasquato@oapd.inaf.it

Backup slides

Approach I: deep Convolutional Neural Net (CNN) directly on mock images

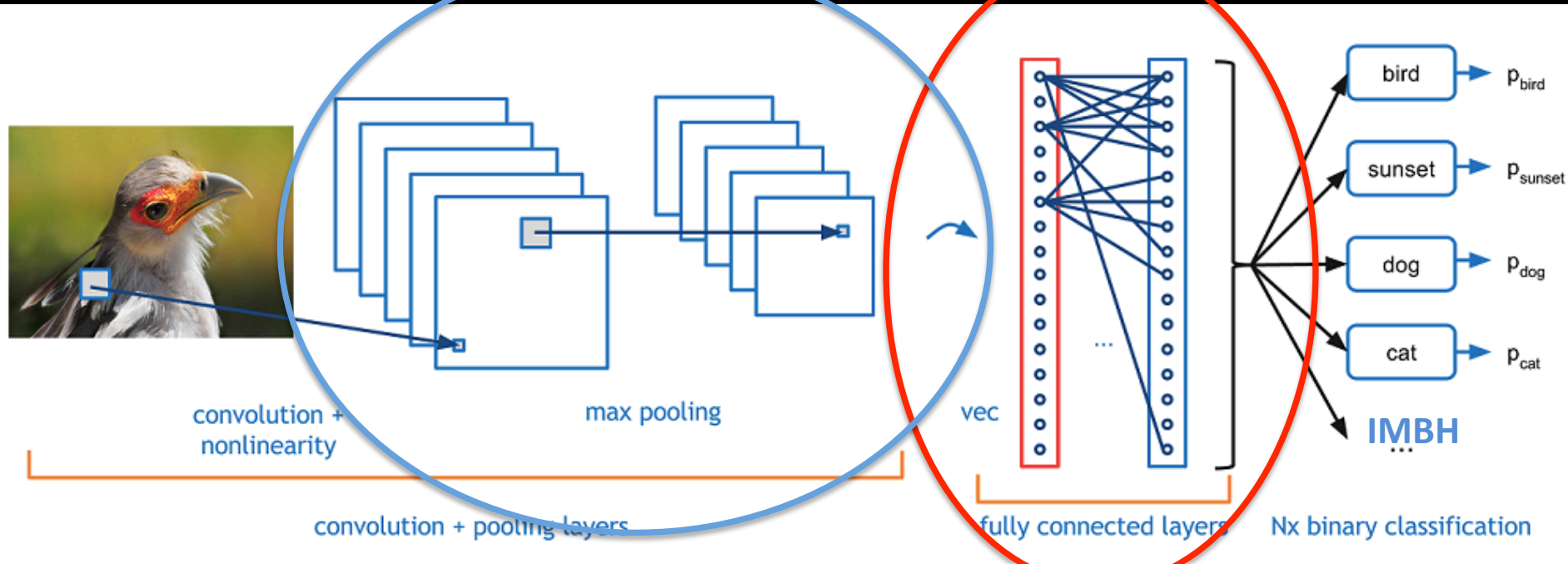
Convolution layers

Features

Fully connected layers

These layers triggers on noses, beaks, ears, stars...

These layers do the understanding



Astronomical Turing test

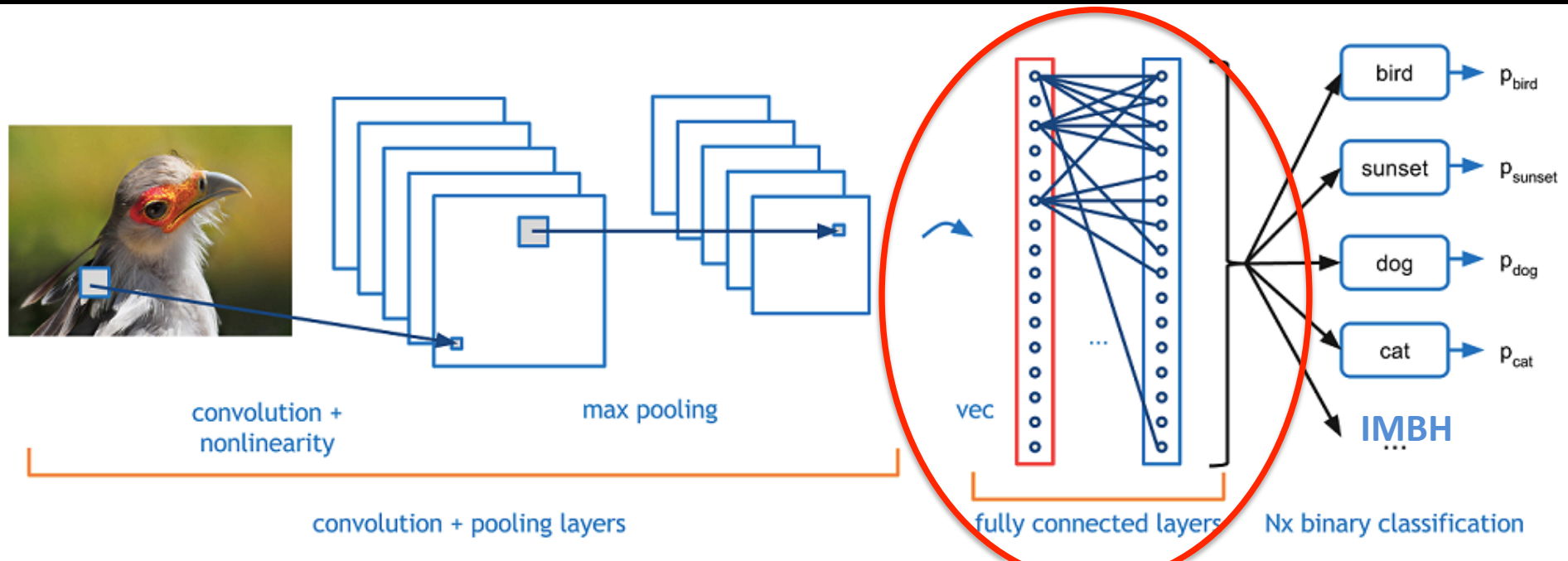
Raw data

Indistinguishable (?)

Mock data

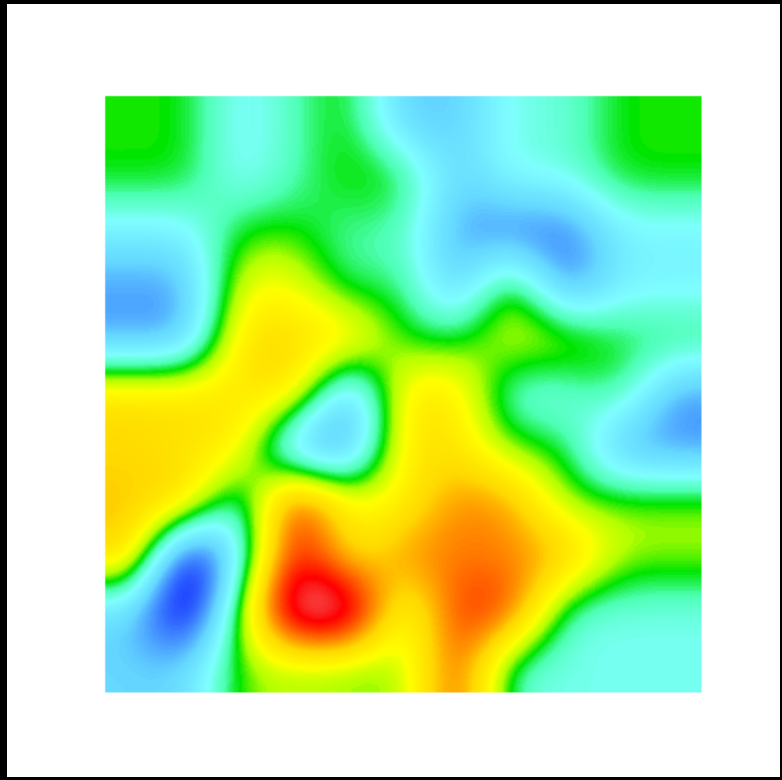
Mock data is indistinguishable from real data if we cannot build any classifier that tells it apart (including a human classifier)

It is weakly indistinguishable from real data if we cannot retrain the fully connected layers of the IMBH/no-IMBH classifier to tell it apart from real data



Pros and cons of CNN approach

- Computationally intensive
- But it parallelizes **very** well on GPUs
- Black box (relatively so...)
- But there are ways to look into the black box
- Generalizes easily to any image



Velocity map of a 16k-particle NBODY6 simulation, courtesy Dr. Mastrobuono-Battisti

We can classify this too!