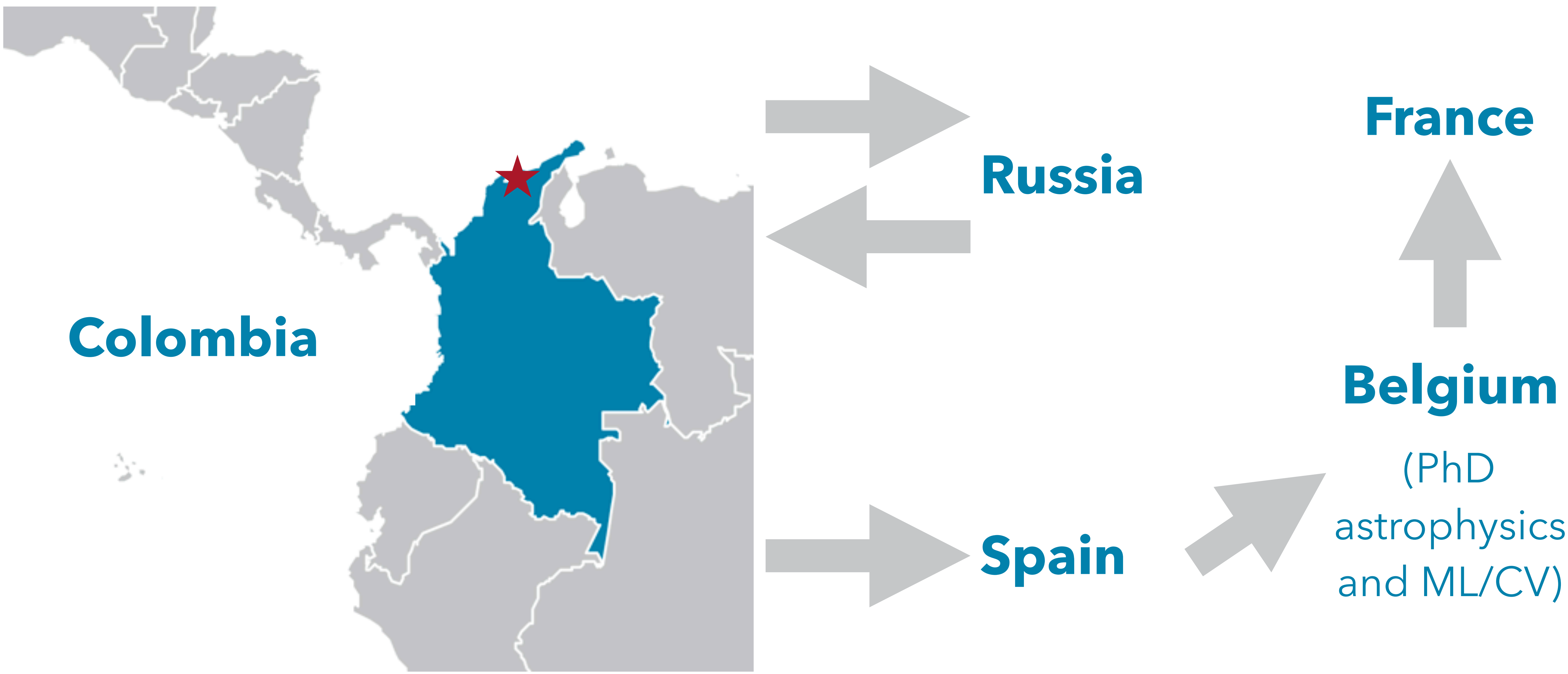
A data scientist’s guide to direct imaging of exoplanets

Carlos Alberto Gomez Gonzalez

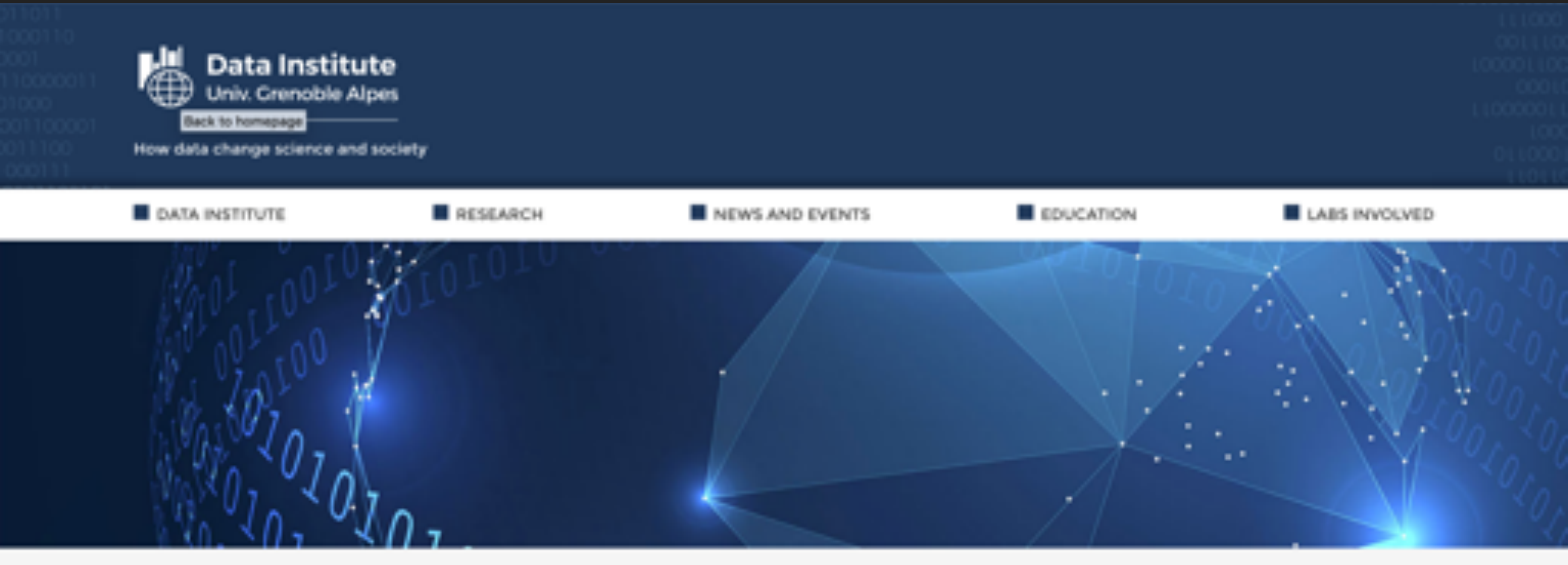
RADA big data workshop (Medellín). Feb 12, 2019

whoami


Colombia
Russia
Spain
Belgium
(PhD astrophysics and ML/CV)
France

whoami

Research data scientist for Earth, Space and environmental sciences



whoami

I have a very particular set of skills :)

A data scientist's guide to direct imaging of exoplanets

A data scientist's guide to direct imaging of exoplanets

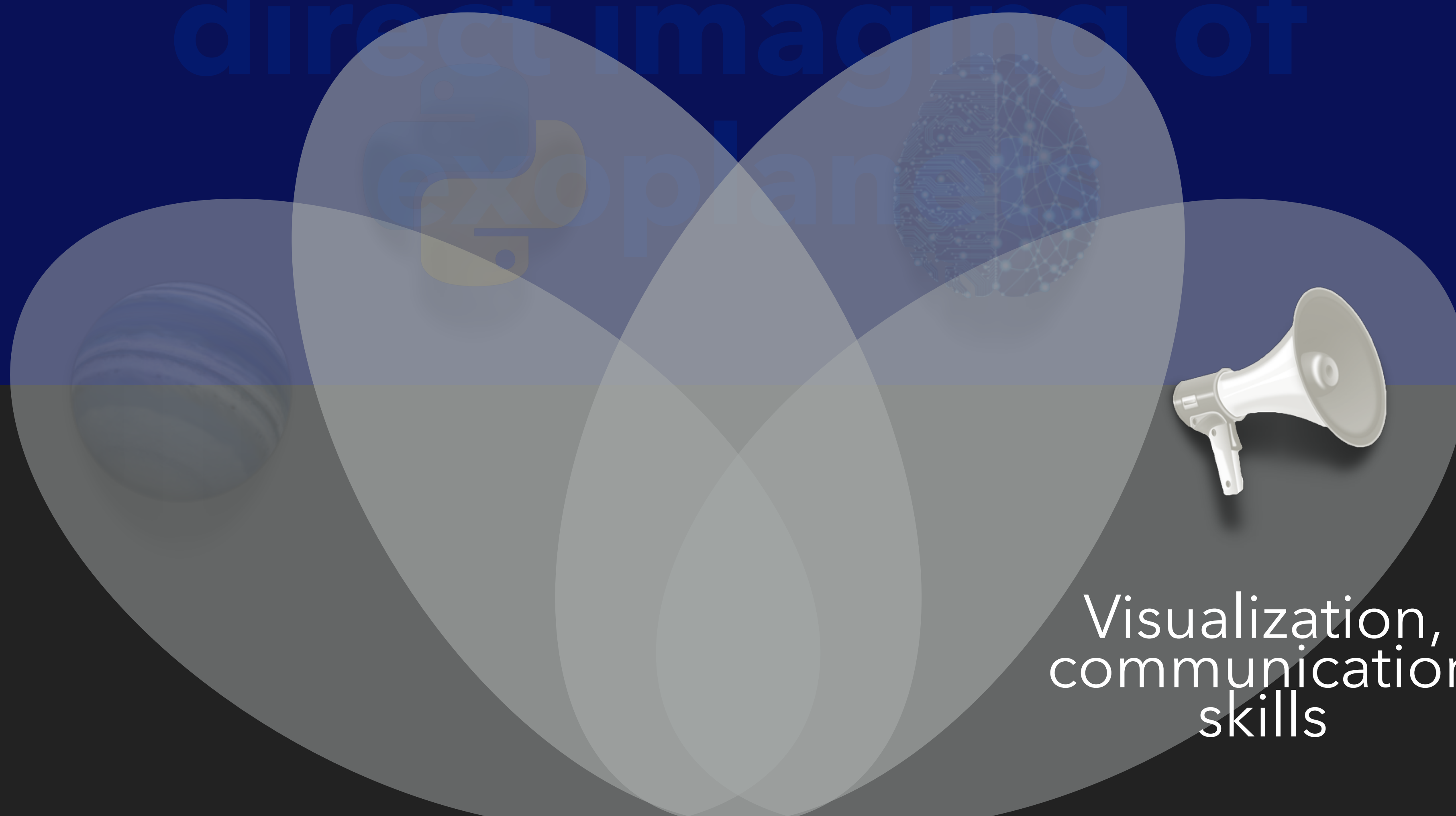
A data scientist's guide to direct imaging of

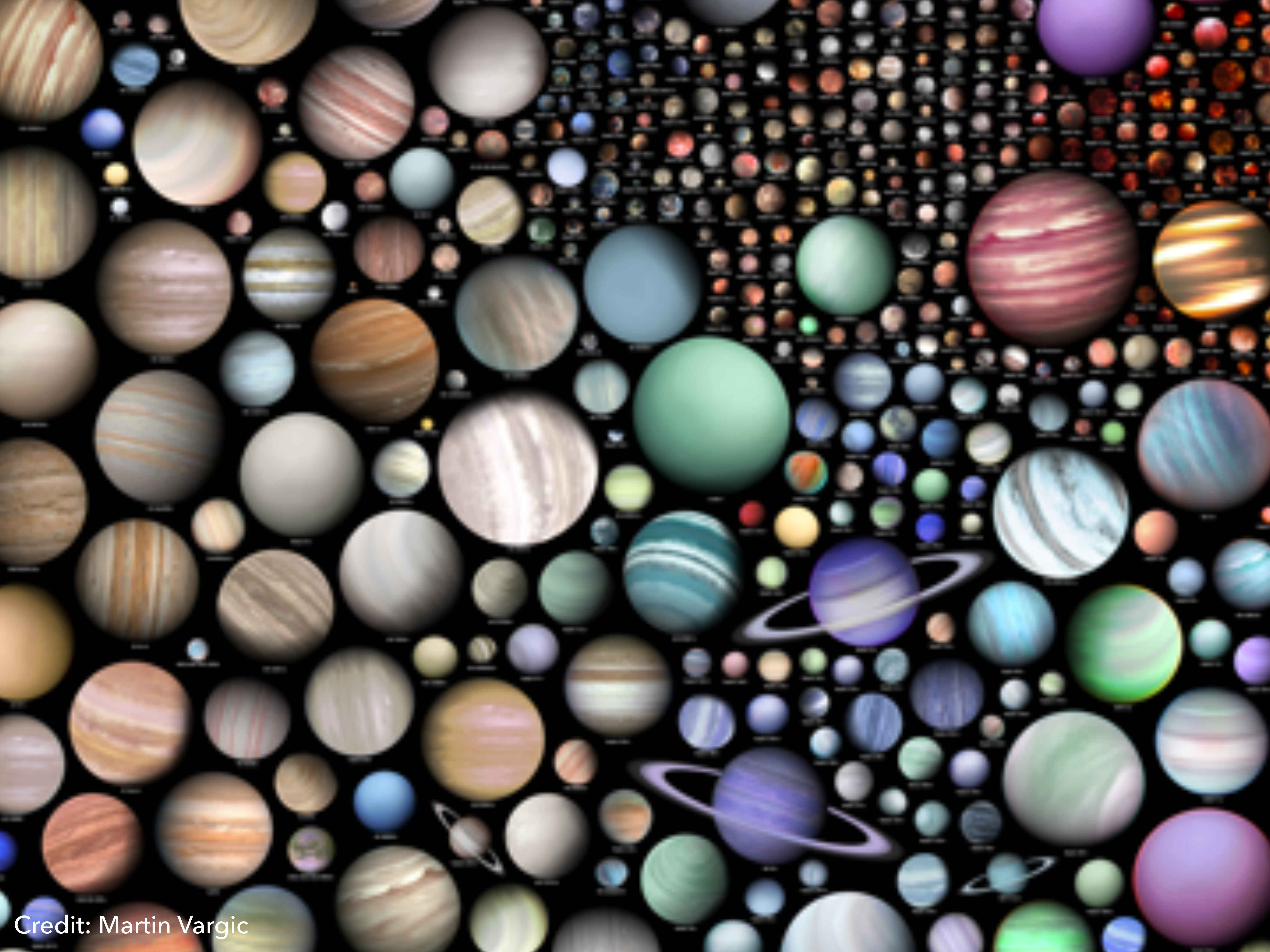
A data scientist's guide to direct imaging of

A data scientist's guide to direct imaging of exoplanets

A data scientist’s guide to direct imaging of

A data scientist's guide to direct imaging of

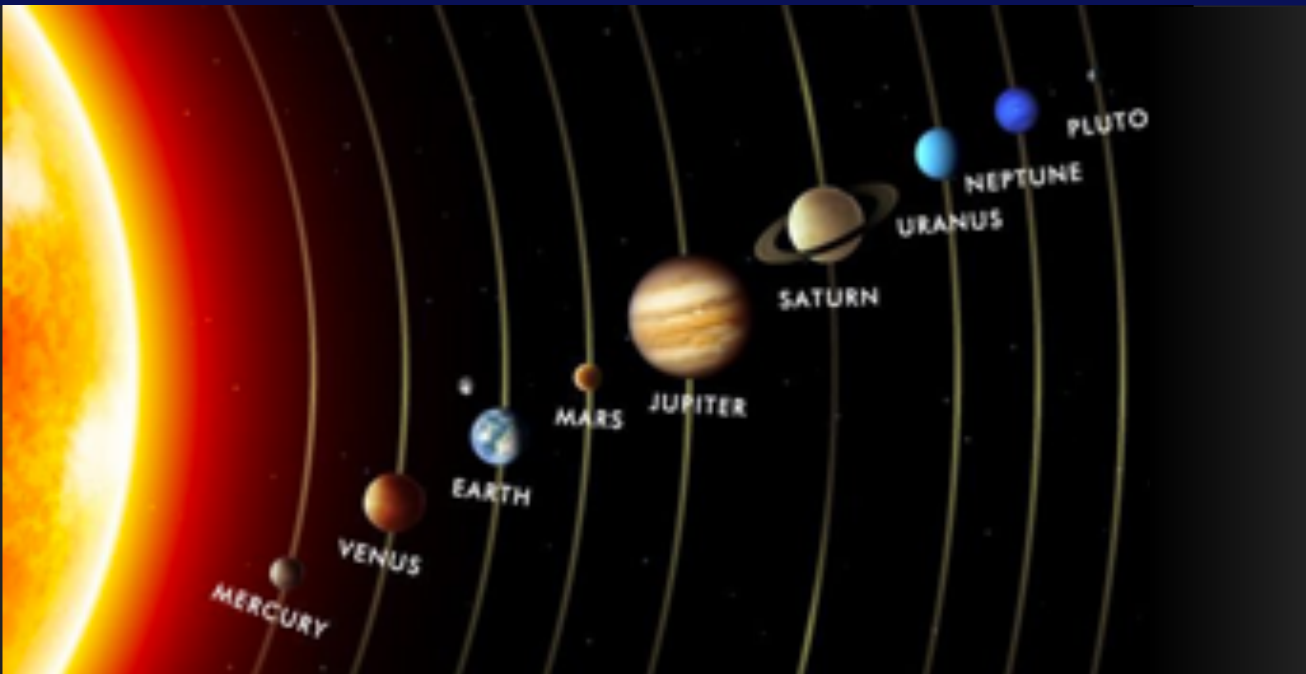



Credit: Martin Vargic

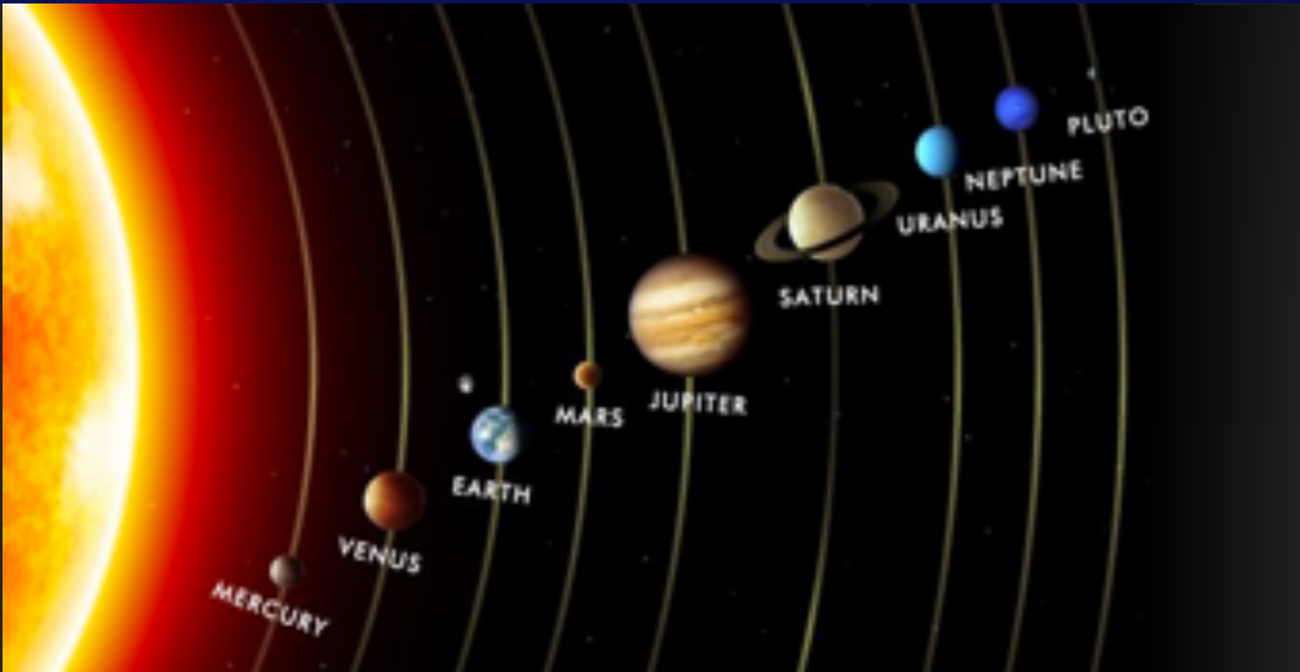
A data scientist's guide to direct imaging of exoplanets

Solar System

A data scientist's guide to
direct imaging of exoplanets

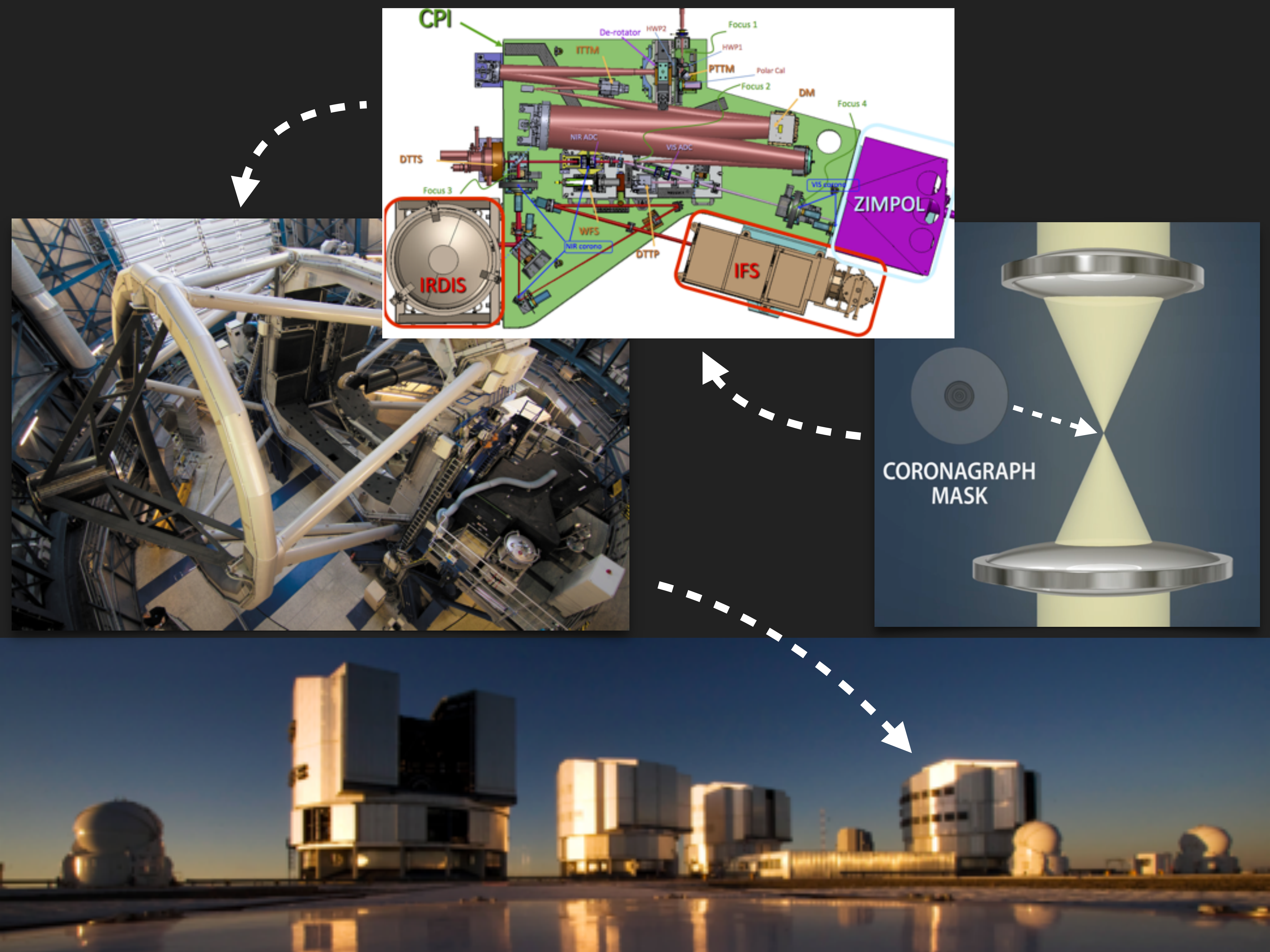


Exo from extrasolar



Exo from extrasolar (outside of the Solar System)

A data scientist’s guide to
direct imaging of exoplanets


CPI
De-rotator
HWP2
Focus 1
HWP1
ITTM
PTTM
Polar Cal
Focus 2
DM
Focus 4
NIR ADC
VIS ADC
DTTS
Focus 3
VIS corono
ZIMPOL
WFS
NIR corono
DTTP
IRDIS
IFS
CORONAGRAPH
MASK

High-contrast imaging

Lighthouse -> Star

Firefly -> exoplanet

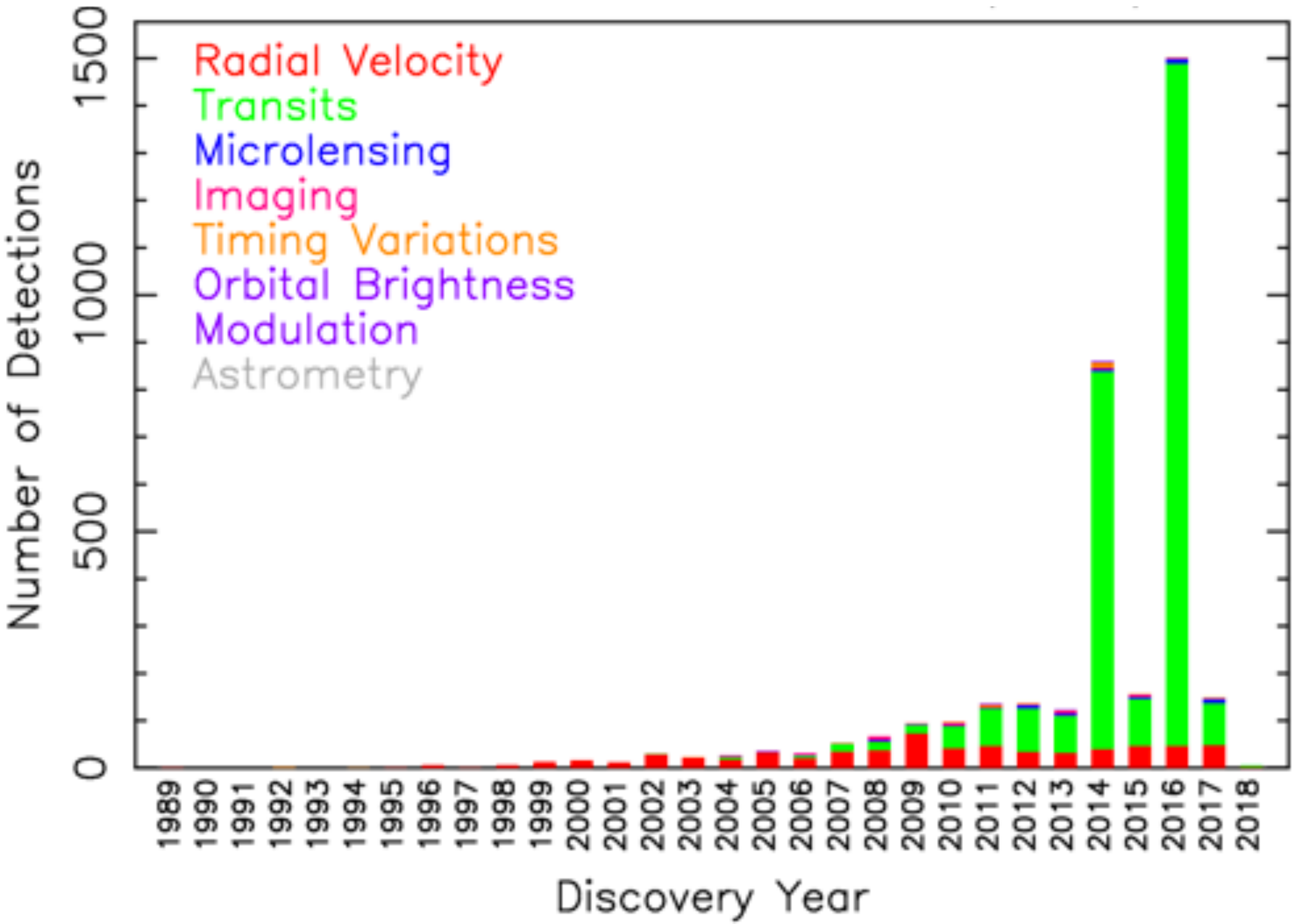
Credit: NASA, http://planetquest.jpl.nasa.gov

High-contrast imaging

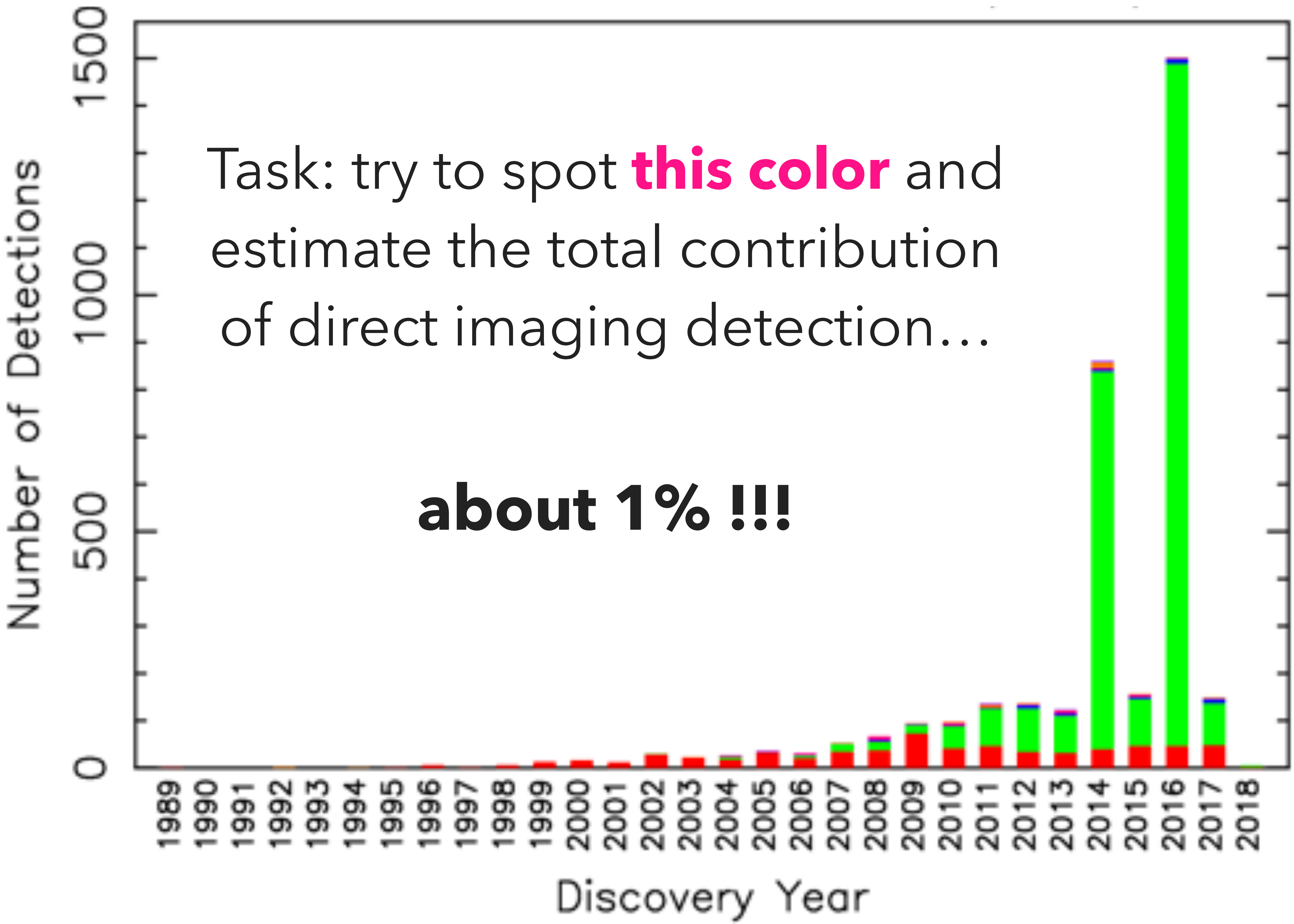
Lighthouse -> Star

Firefly -> exoplanet

Credit: NASA, http://planetquest.jpl.nasa.gov



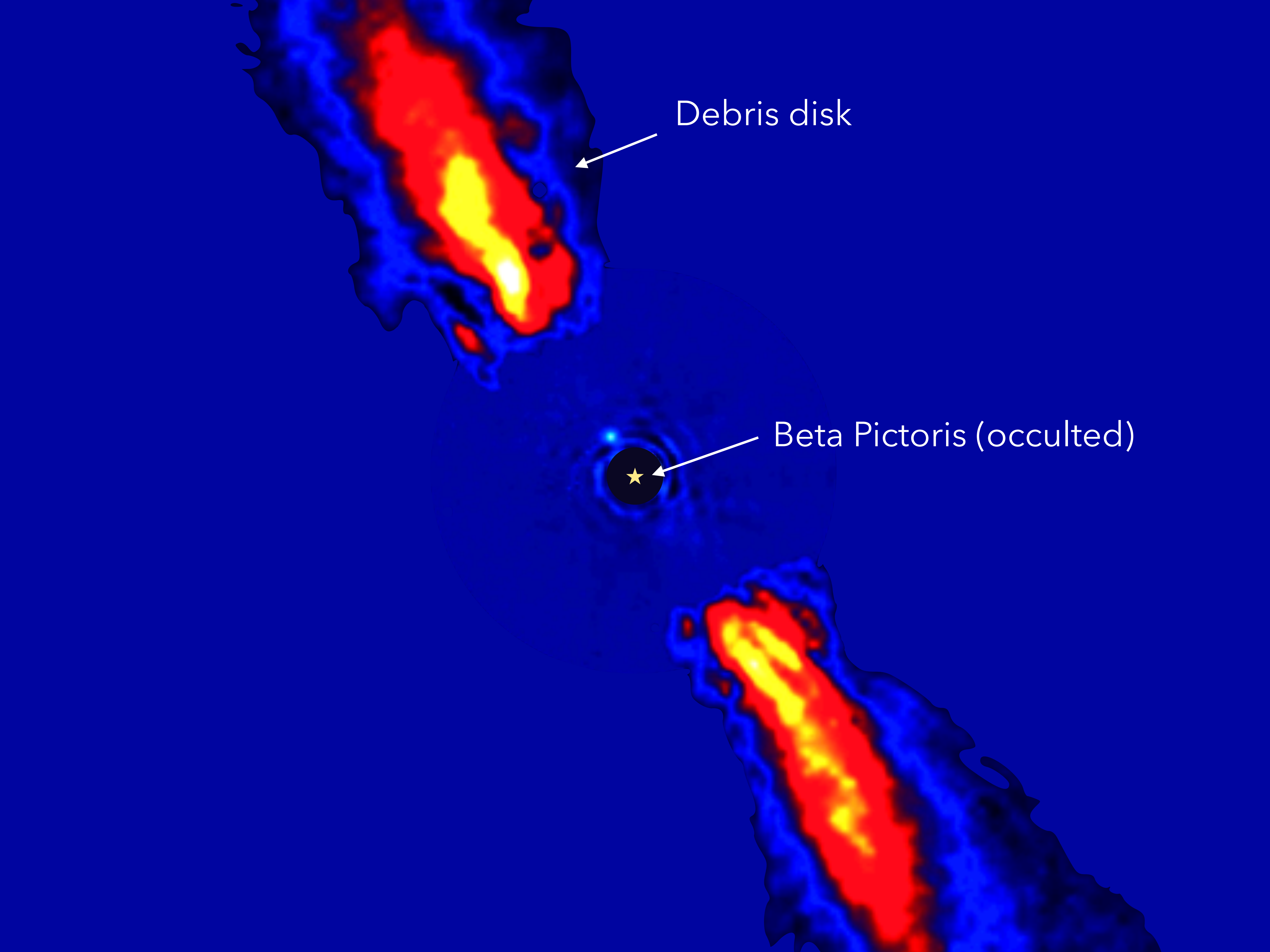
http://exoplanetarchive.ipac.caltech.edu, 25 Jan 2018

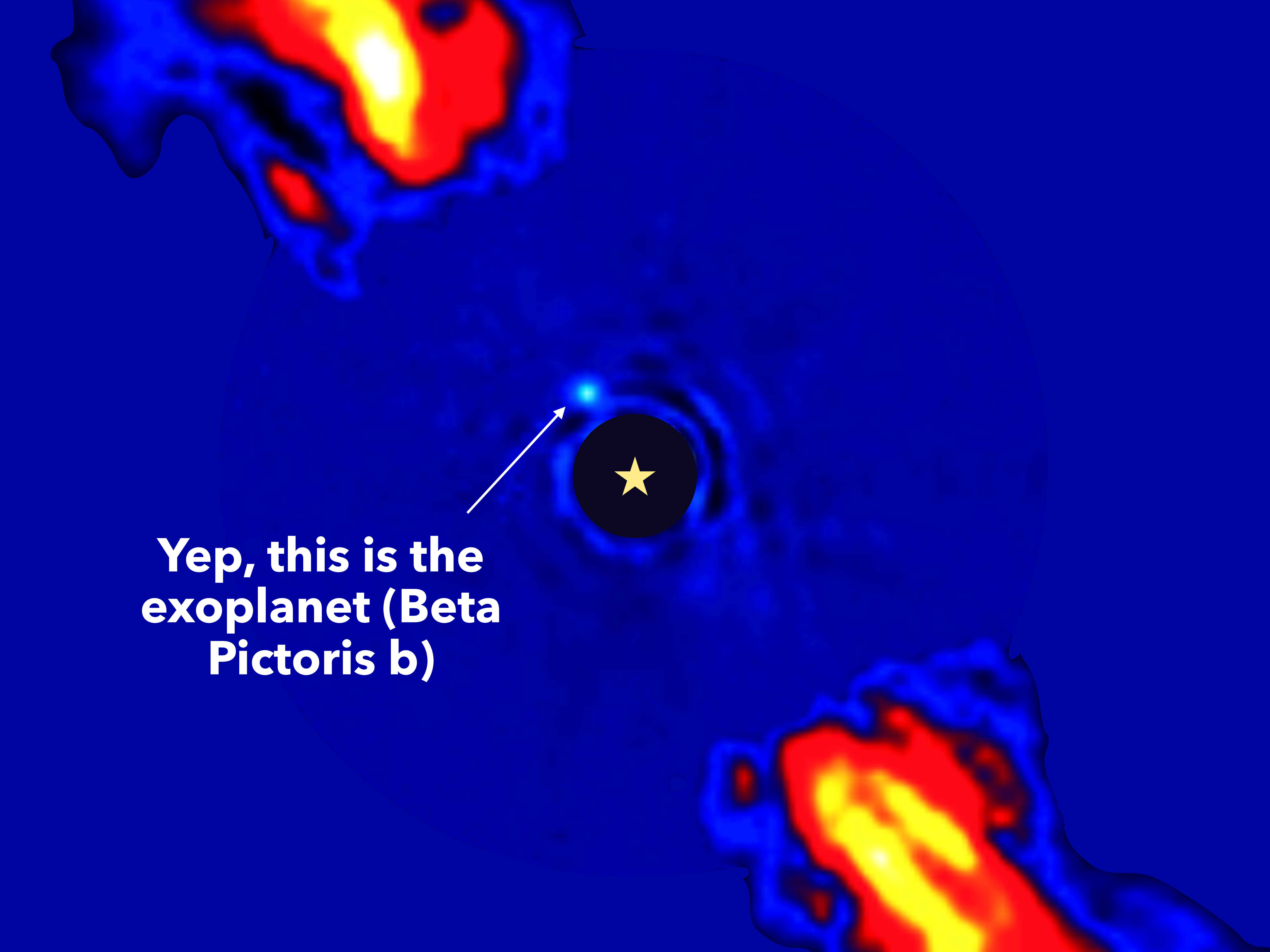


Task: try to spot **this color** and estimate the total contribution of direct imaging detection…

about 1% !!!

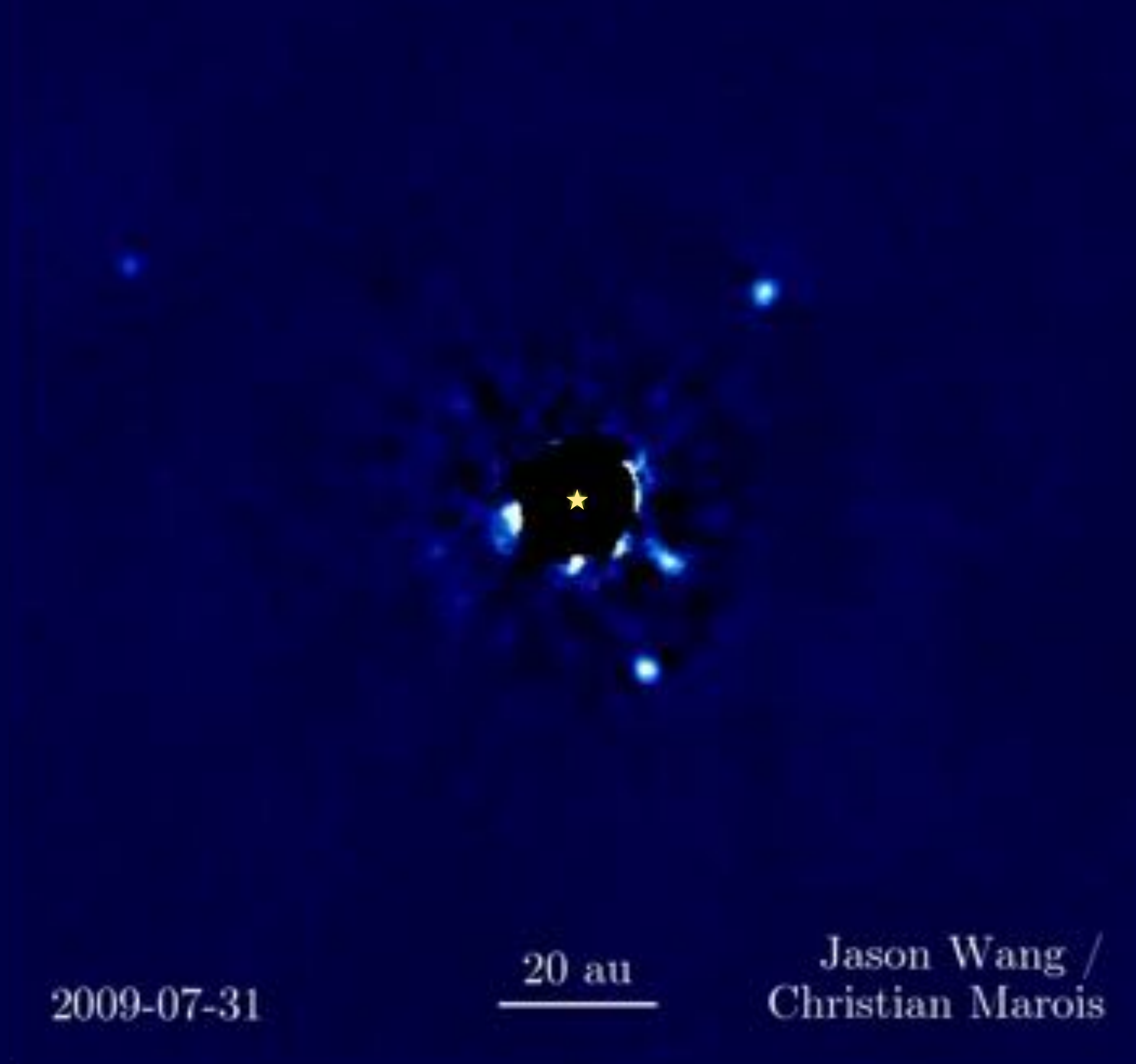
http://exoplanetarchive.ipac.caltech.edu, 25 Jan 2018


Debris disk
Beta Pictoris (occulted)

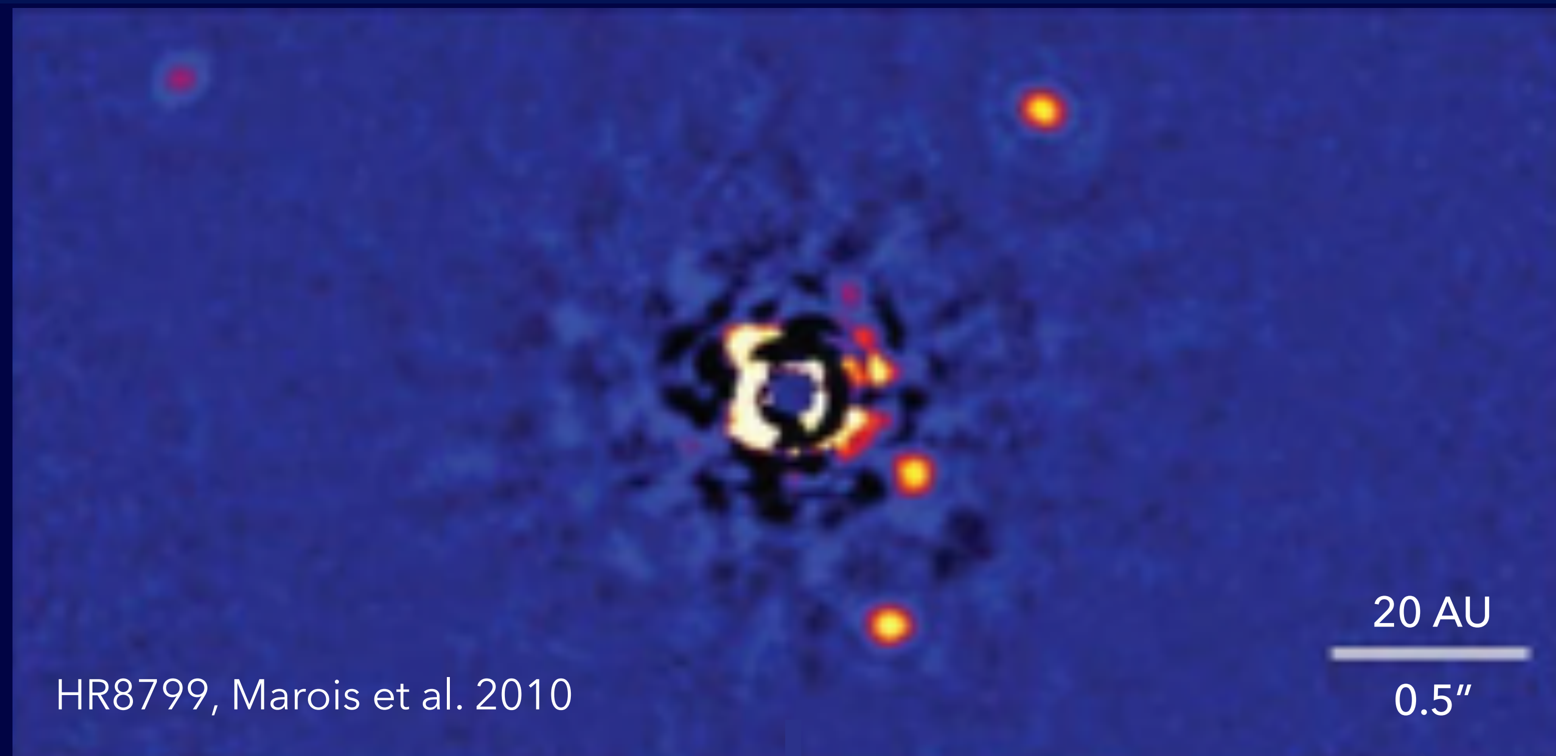

Yep, this is the exoplanet (Beta Pictoris b)

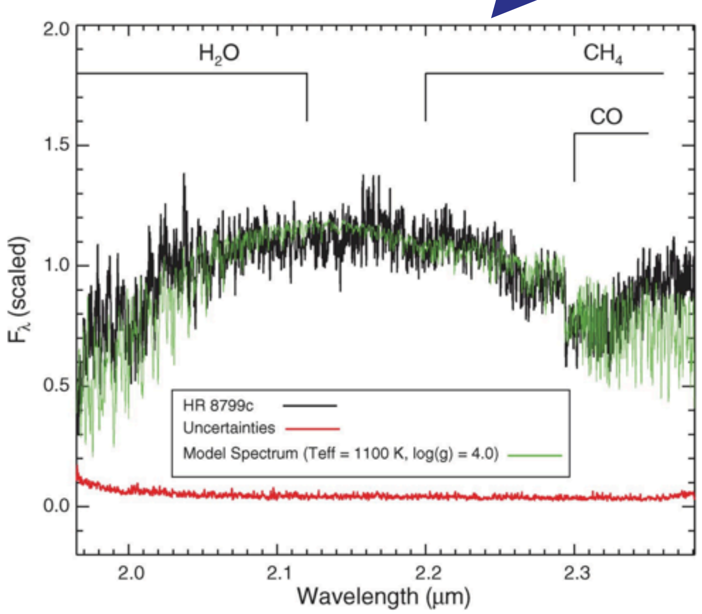
HR8799 bcde (Marois et al. 2008-2010)

Several epochs (final images)

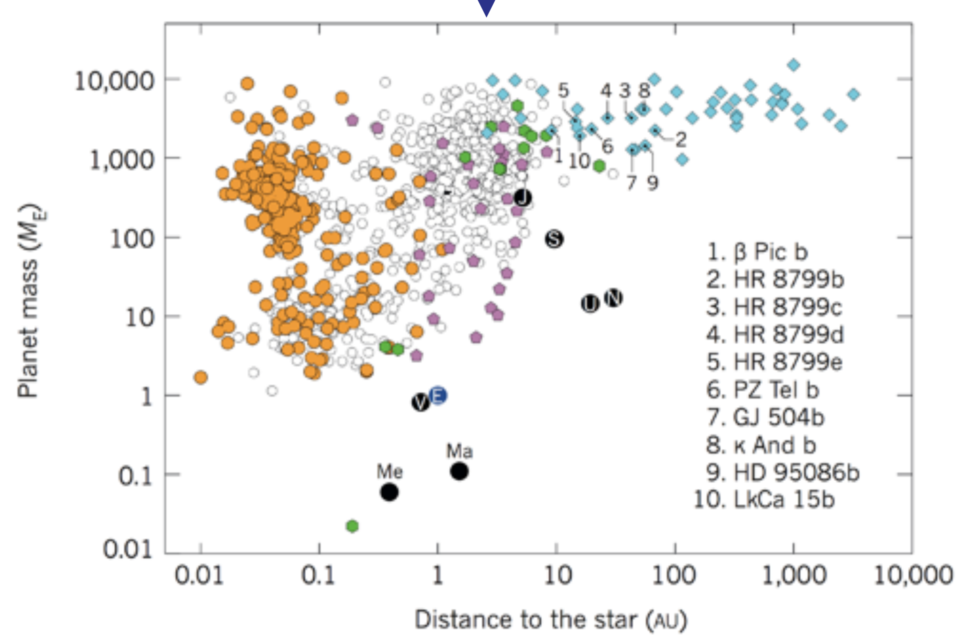


Why direct imaging?

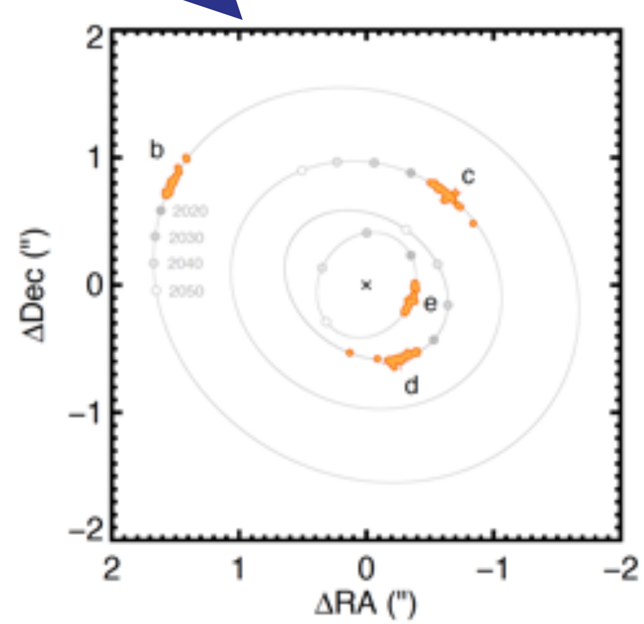




Konopacky et al. 2013



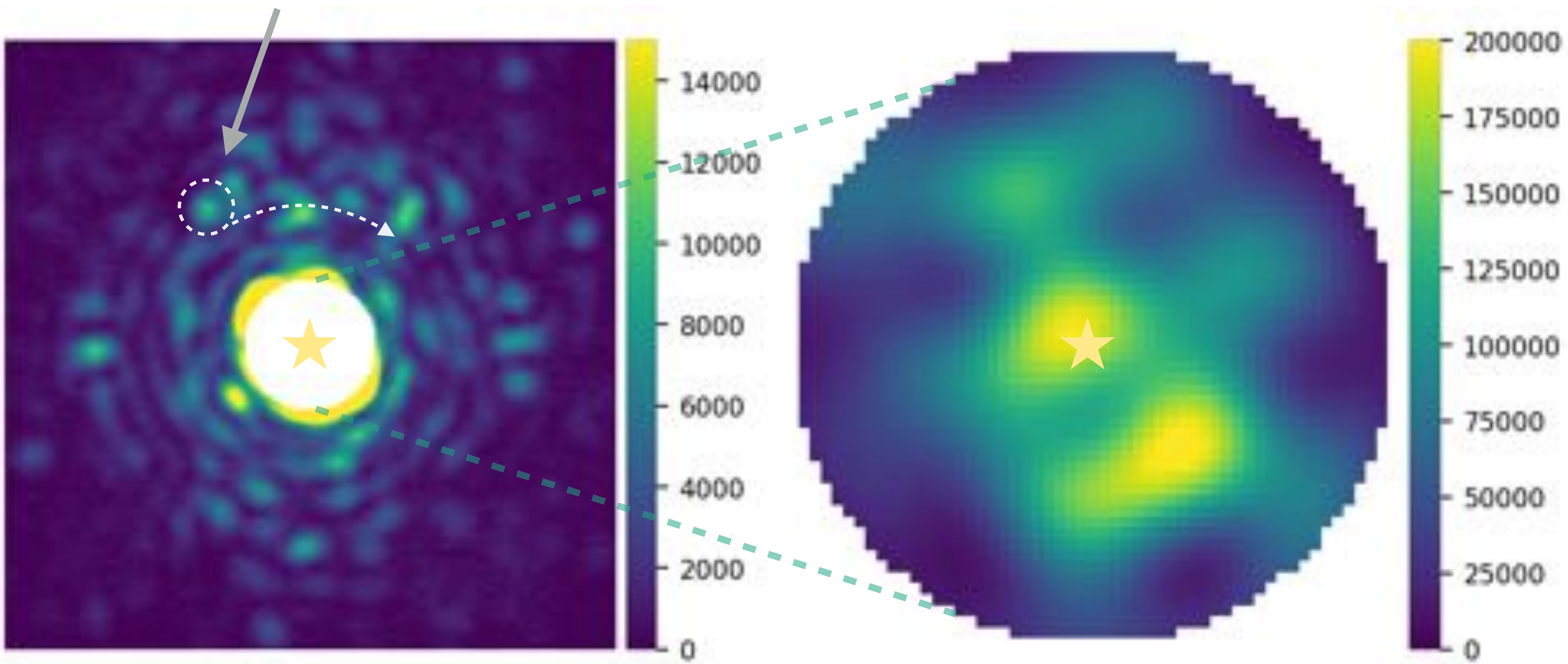
Milli et al. 2016



Bowler 2016

Signal and noise

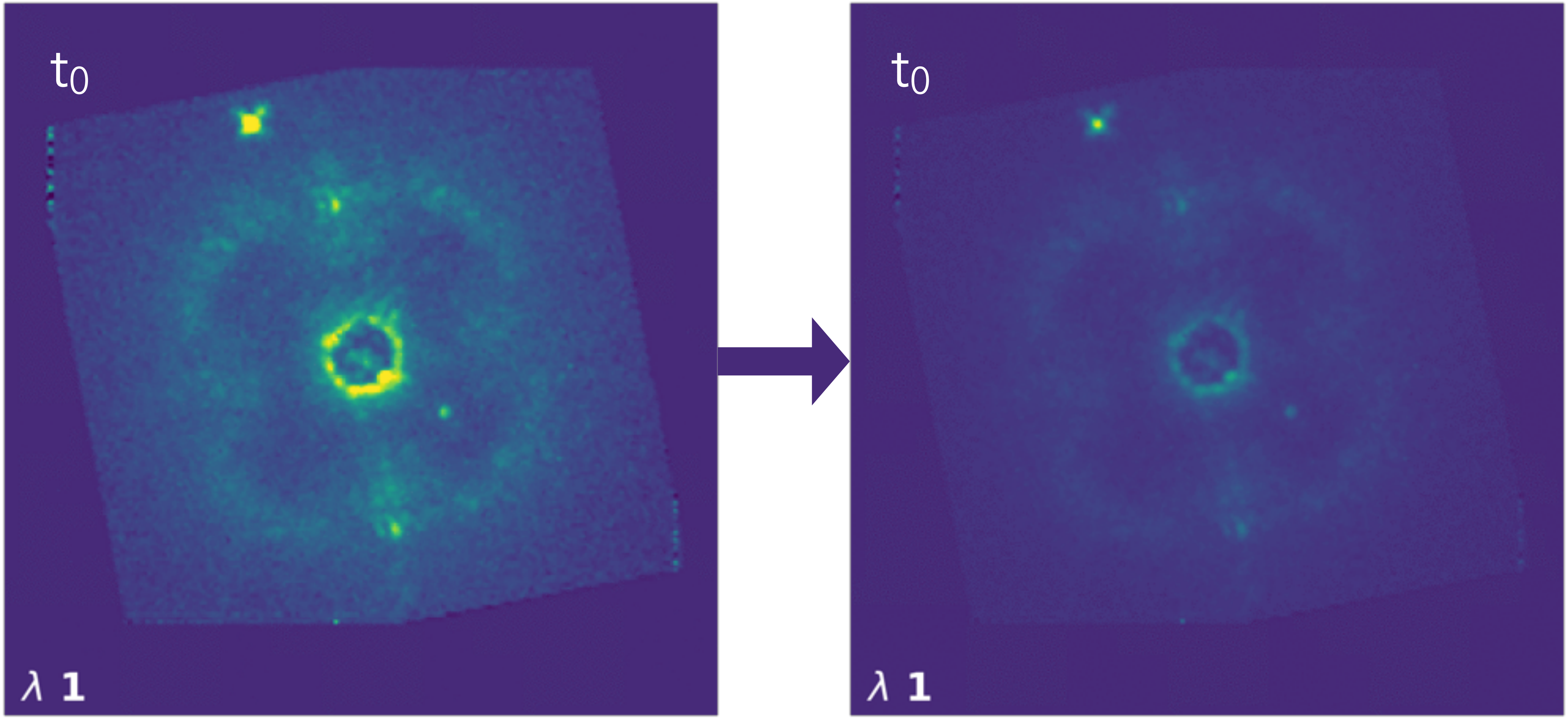
Fake planet (low contrast)



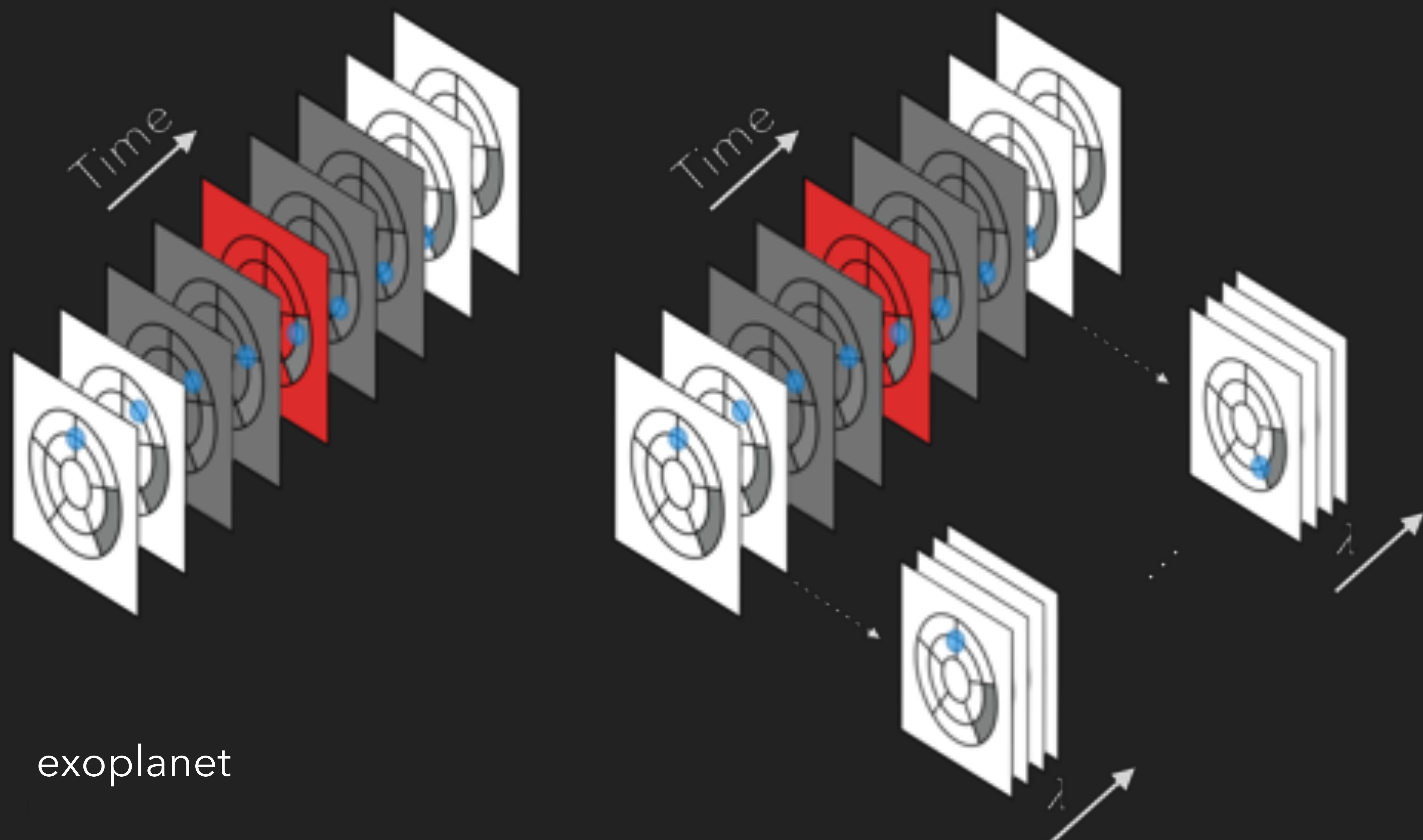
Signal and noise

with an integral field spectrograph

spatially rescaled images



Multi-dimensional arrays (t, λ, x, y)



exoplanet

Image processing pipeline

Image processing pipeline

Getting rid of the noise…

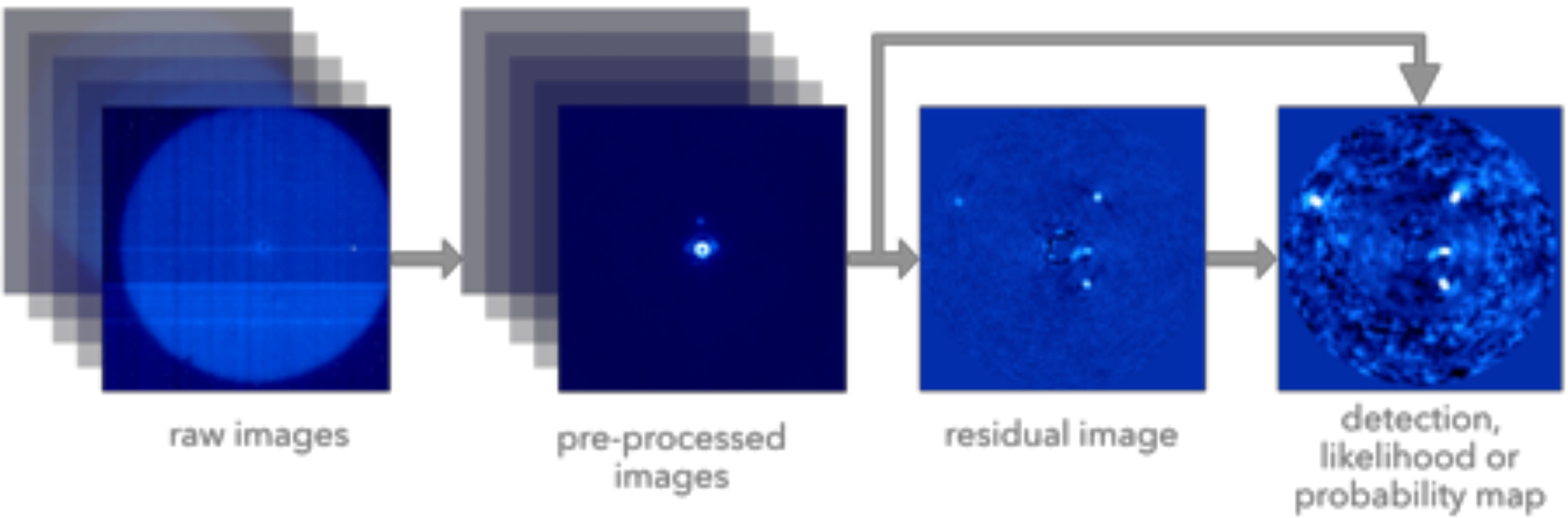
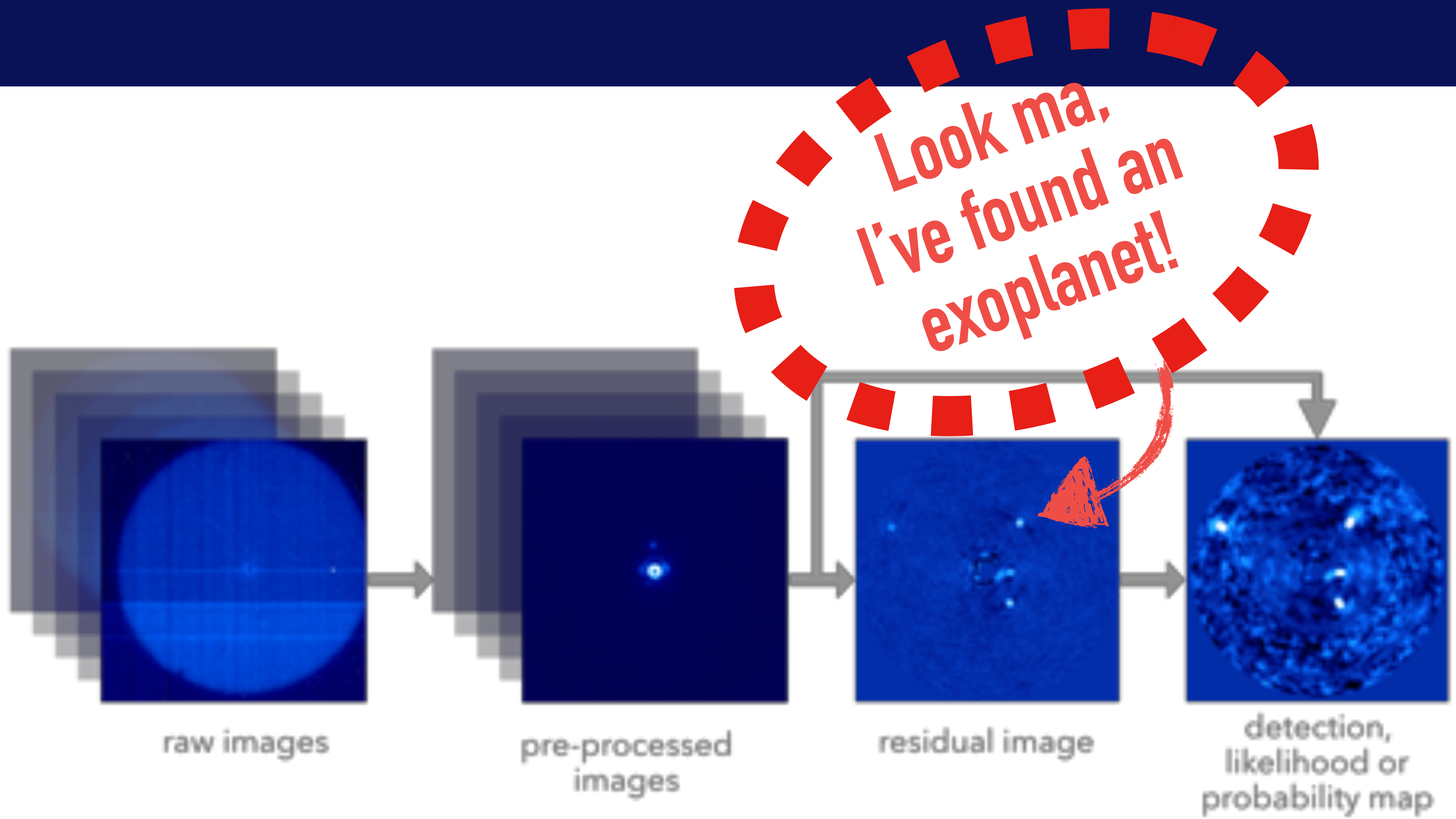
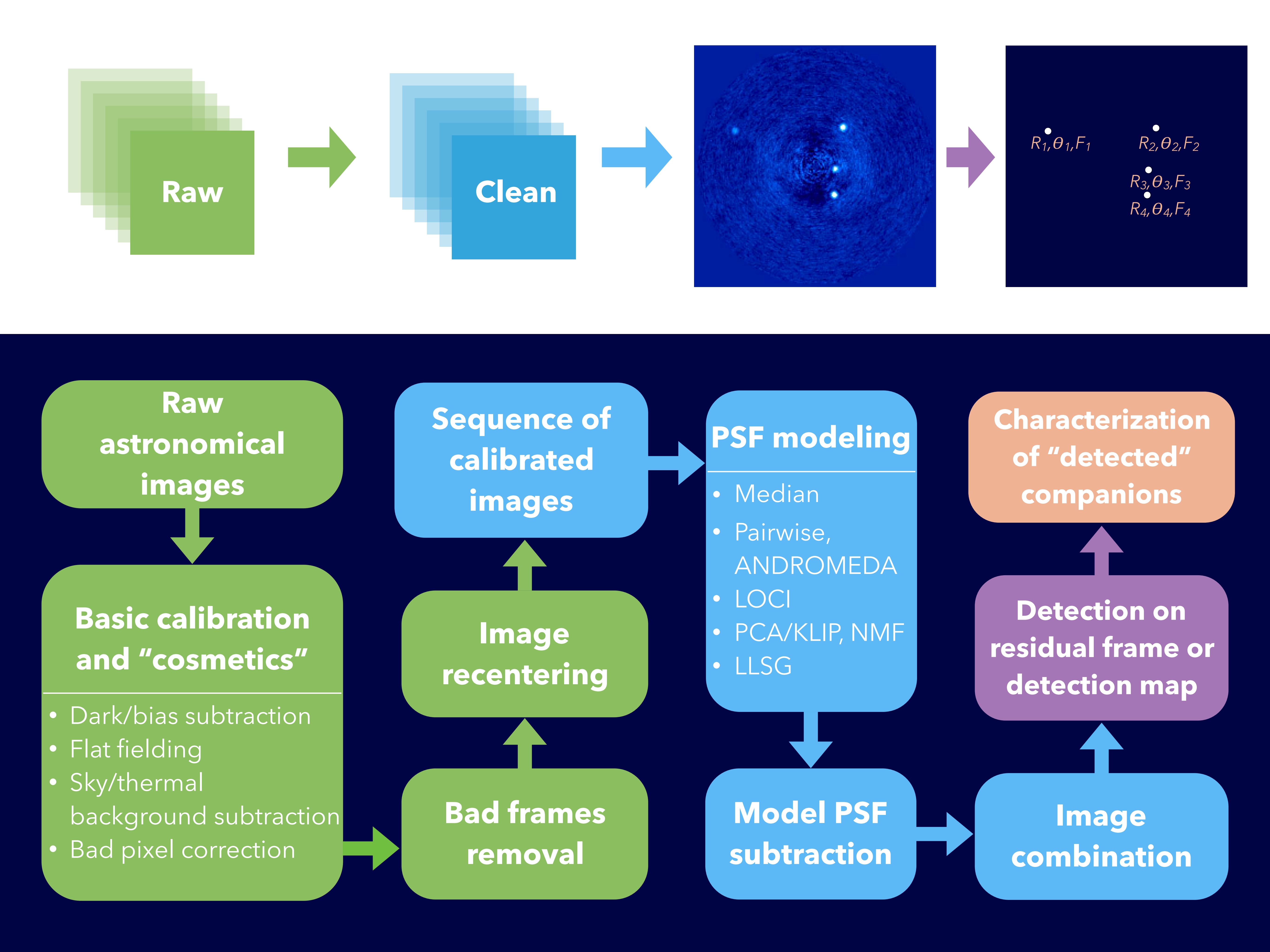


Image processing pipeline




Raw
Clean
R1,θ1,F1
R2,θ2,F2
R3,θ3,F3
R4,θ4,F4
Raw astronomical images
Basic calibration and "cosmetics"
• Dark/bias subtraction
• Flat fielding
• Sky/thermal background subtraction
• Bad pixel correction
Bad frames removal
Image recentering
Sequence of calibrated images
PSF modeling
• Median
• Pairwise, ANDROMEDA
• LOCI
• PCA/KLIP, NMF
• LLSG
Model PSF subtraction
Image combination
Detection on residual frame or detection map
Characterization of "detected" companions

Algorithms

Ten years of research:

- Median frame subtraction
- Pairwise subtraction
- Least squares image combination
- PCA (forward modeling), NMF
- Low-rank plus sparse decompositions
- Matched filtering
- Maximum likelihood estimation

```
In [115]: print('images: {}, angles: {}'.format(cube3d.shape, pa3d.shape))

images: (61, 101, 101), angles: (61,)
```

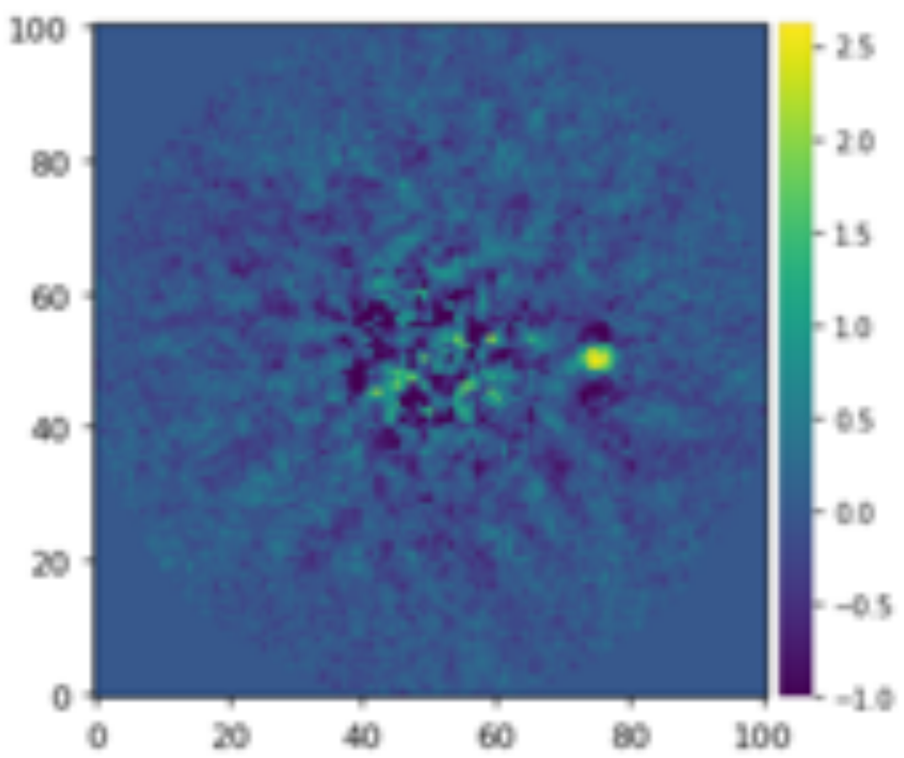
```
In [116]: cube3d_wfc = vip.metrics.cube_inject_companions(cube3d, psf3d, pa3d, 200,
                                                          plsc3d, rad_dists=25, verbose=False)
```

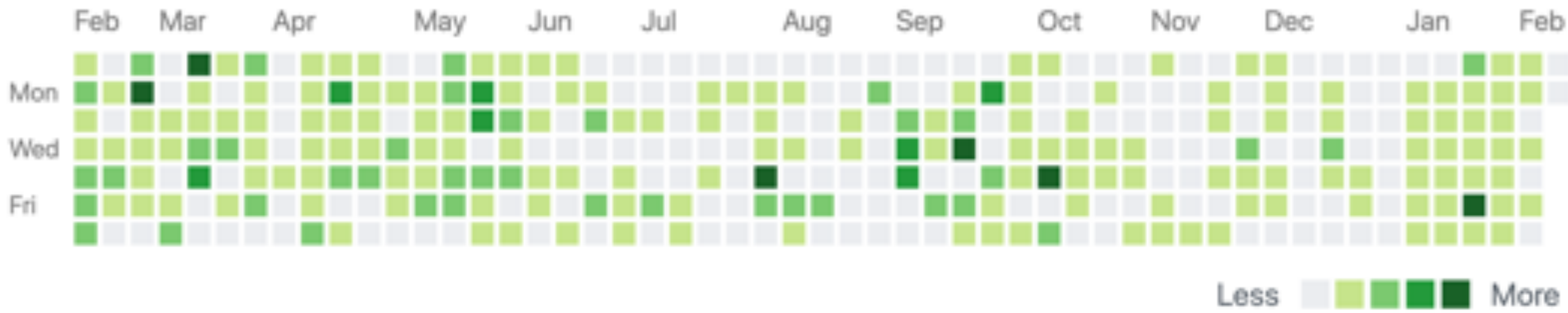
```
In [117]: matrix = cube3d_wfc.reshape(61, -1)    # (61, 10201)
          u, s, v = np.linalg.svd(matrix, full_matrices=False)
```

```
In [118]: v.shape    # right singular vectors

Out[118]: (61, 10201)
```

```
In [119]: projection = np.dot(v[:20], matrix.T)     # using 20 SVs
          model = np.dot(projection.T, v[:20])    # reconstructing the images
          residuals = matrix - model    # subtracting
          cube_res = residuals.reshape(61, 101, 101)
```

```
In [121]: cube_derot = np.zeros_like(cube3d_wfc)
          for i in range(cube3d_wfc.shape[0]):
              M = cv2.getRotationMatrix2D((50.0, 50.0), -pa3d[i], 1)
              cube_derot[i] = cv2.warpAffine(cube_res[i].astype(np.float32), M, (101, 101))

          frame = np.median(cube_derot, axis=0)
          vip.var.pp_subplots(frame, vmin=-1)
```





Researcher creates an open-source library

VIP

scikit learn

astropy

jupyter

NumPy

matplotlib

scikit learn

scikit-image
image processing in python

astropy

NumPy

vortex-exoplanet / VIP

Unwatch 14 | Unstar 33 | Fork 22

VIP - Vortex Image Processing package

pypi package 0.9.8 | python 2.7, 3.6, 3.7 | build passing | license MIT | arxiv 1705.06184 | docs passing | codecov 30%

- https://github.com/vortex-exoplanet/VIP
- http://vip.readthedocs.io/
- CI, test suite, jupyter tutorials, paper on A&A

Code | Issues 19 | Pull requests 3 | Projects 1 | Wiki | Insights | Settings

phot/snr.py #354

Edit | New issue

Open KristinaDavis opened this issue 13 days ago · 3 comments

KristinaDavis commented 13 days ago

import statement needs to import get_annulus_segments instead of old get_annulus

carlgogo commented 12 days ago — Member

I would suggest to install from master instead of using the version from pypi (a bit outdated). You could run: `pip install git+https://github.com/vortex-exoplanet/VIP.git` .

KristinaDavis commented 7 days ago

Hello Carlos,

I have the vip-hci package installed in a conda environment. At first, I
ddid a pip-install of vip-hci using pypi, as you correctly assumed. When I
try and install the new package directly from git, I get the following
error:
Building wheels for collected packages: vip-hci
  Running setup.py bdist_wheel for vip-hci ... done
  Stored in directory:
/tmp/pip-ephem-wheel-cache-
7jqopiq1/wheels/8f/14/88/95225840c2146aaf815de70113580db62accc2753c78d86a6a
Successfully built vip-hci
Could not install packages due to an EnvironmentError: [Errno 2] No such

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

Notifications
Unsubscribe
You're receiving notifications because you commented.

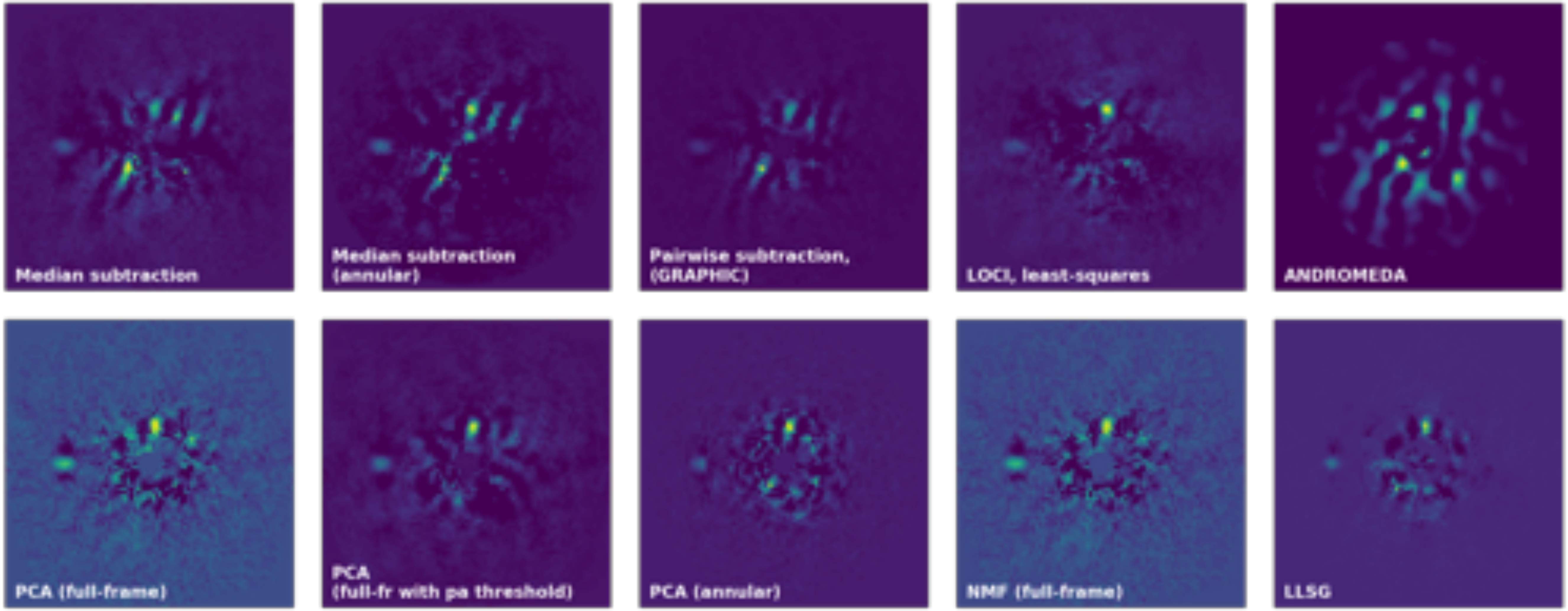
2 participants

Lock conversation

Pin issue

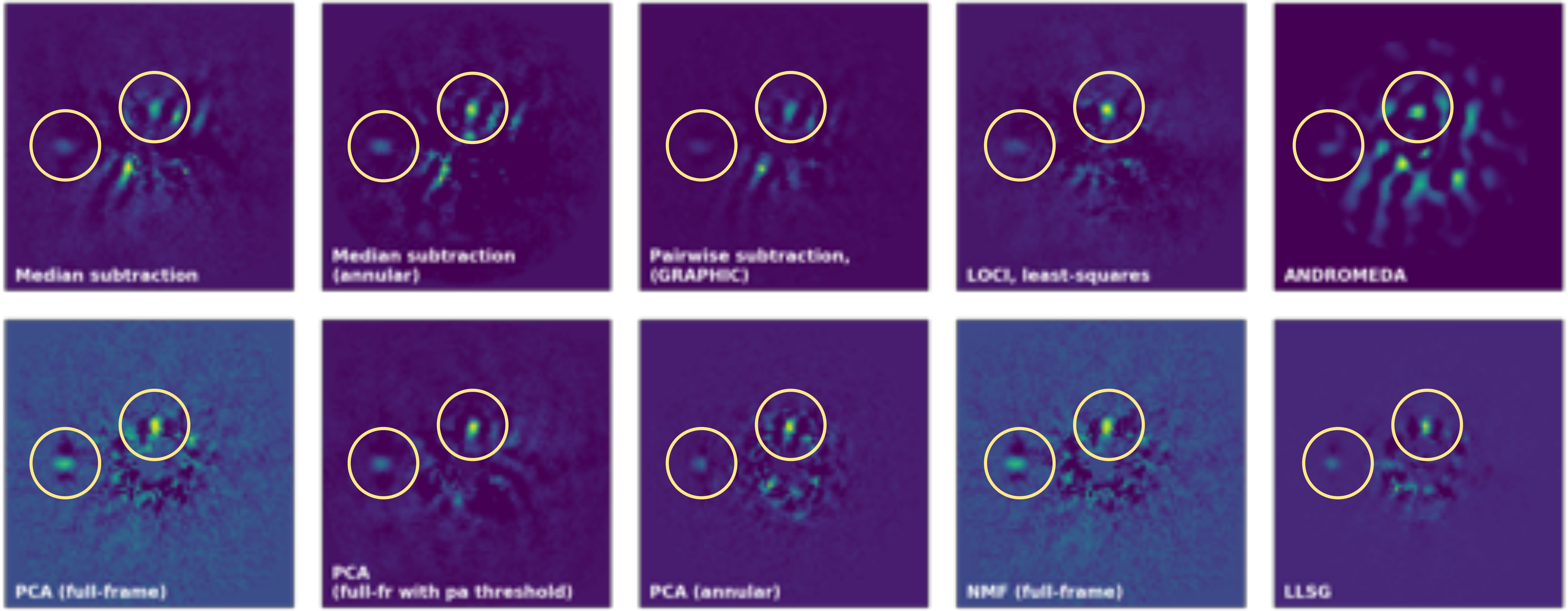
VIP: algorithmic zoo

Comparing the resulting images (using different algorithms)



VIP: algorithmic zoo

Task: remove the noise and reveal the exoplanet signal



Designing APIs for science

```
In [3]: cu3d = vip.HCIDataset(cube=cube, angles=angles, fwhm=4.5, psf=psf, px_scale=vip.conf.VLT_SPHERE['plsc'])
```

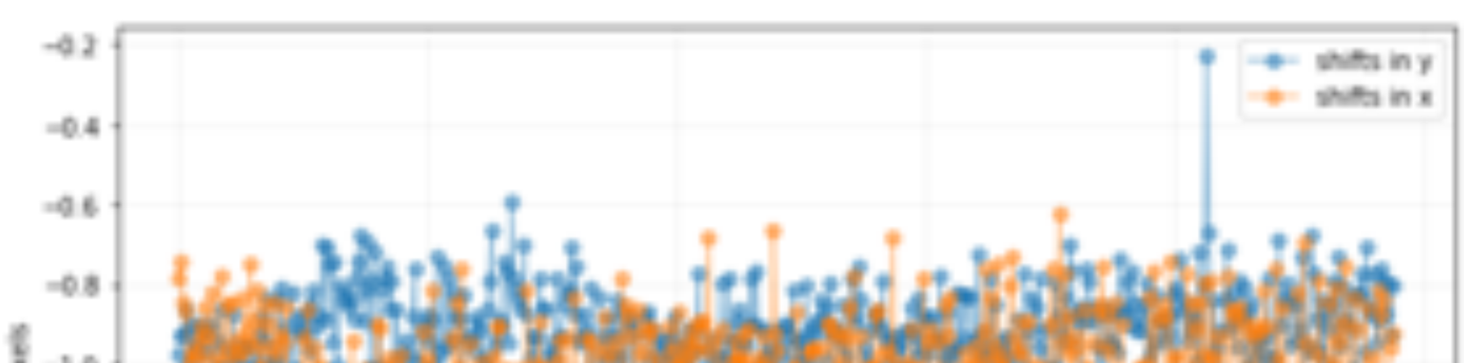
```
Cube array shape: (489, 250, 250)
Angles array shape: (489,)
PSF array shape: (19, 19)
FWHM: 4.5
Pixel/plate scale: 0.01225
```

```
In [ ]: cu3d.
```

```
cu3d.angles          instance
cu3d.collapse        function
cu3d.crop_frames     function
cu3d.cube            instance
cu3d.cuberef         instance
cu3d.derotate        function
cu3d.drop_frames     function
cu3d.filter          function
cu3d.frame_distances function
cu3d.frame_stats     function
```

```
In [21]: cu3d.recenter('2d_fitting', model='gauss')
```

```
--------------------------------------------------------------------------------
Starting time: 2018-05-01 12:28:17
--------------------------------------------------------------------------------
`subi_size` is odd (while frame size is even)
Setting `subi_size` to 6 pixels
2d Gauss-fitting, looping through frames
0% [##############################] 100% | ETA: 00:00:00
Total time elapsed: 00:00:03
Shifting the frames
0% [##############################] 100% | ETA: 00:00:00
Total time elapsed: 00:00:01
Running time:  0:00:04.994335
--------------------------------------------------------------------------------
```

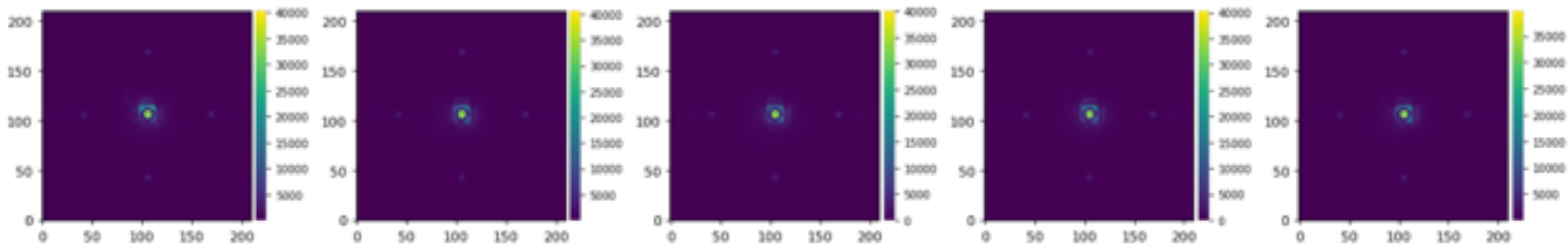


Designing APIs for science

```
In [29]: cu3d.crop_frames(size=210)
```

```
New shape: (489, 210, 210), centered at (124.5, 124.5)
```

```
In [31]: cu3d.plot(maxplots=5)
```



```
In [32]: PcaSub1 = vip.HCIPca(ncomp=5)
```

```
adimsdi: double
check_mem: True
collapse: median
delta_rot: 1
imlib: opencv
interpolation: lanczos4
mask_central_px: None
ncomp: 5
ncomp2: 1
scaling: None
source_xy: None
svd_mode: lapack
```

```
In [33]: fr1 = PcaSub1.run(cu3d)
```

```
--------------------------------------------------------------------------------
Starting time: 2018-05-01 12:35:26
--------------------------------------------------------------------------------
Done vectorizing the frames. Matrix shape: (489, 44100)
Done SVD/PCA with numpy SVD (LAPACK)
Running time:  0:00:05.061552
--------------------------------------------------------------------------------
Done de-rotating and combining
Running time:  0:00:06.739208
--------------------------------------------------------------------------------
```

Building blocks...

Noise reduction algorithm

$$\mathbf{D} - \mathbf{M} = \mathrm{R}$$

R - residuals containing the exoplanet signal

Building blocks...

Noise reduction algorithm → **Observer (classifier)**

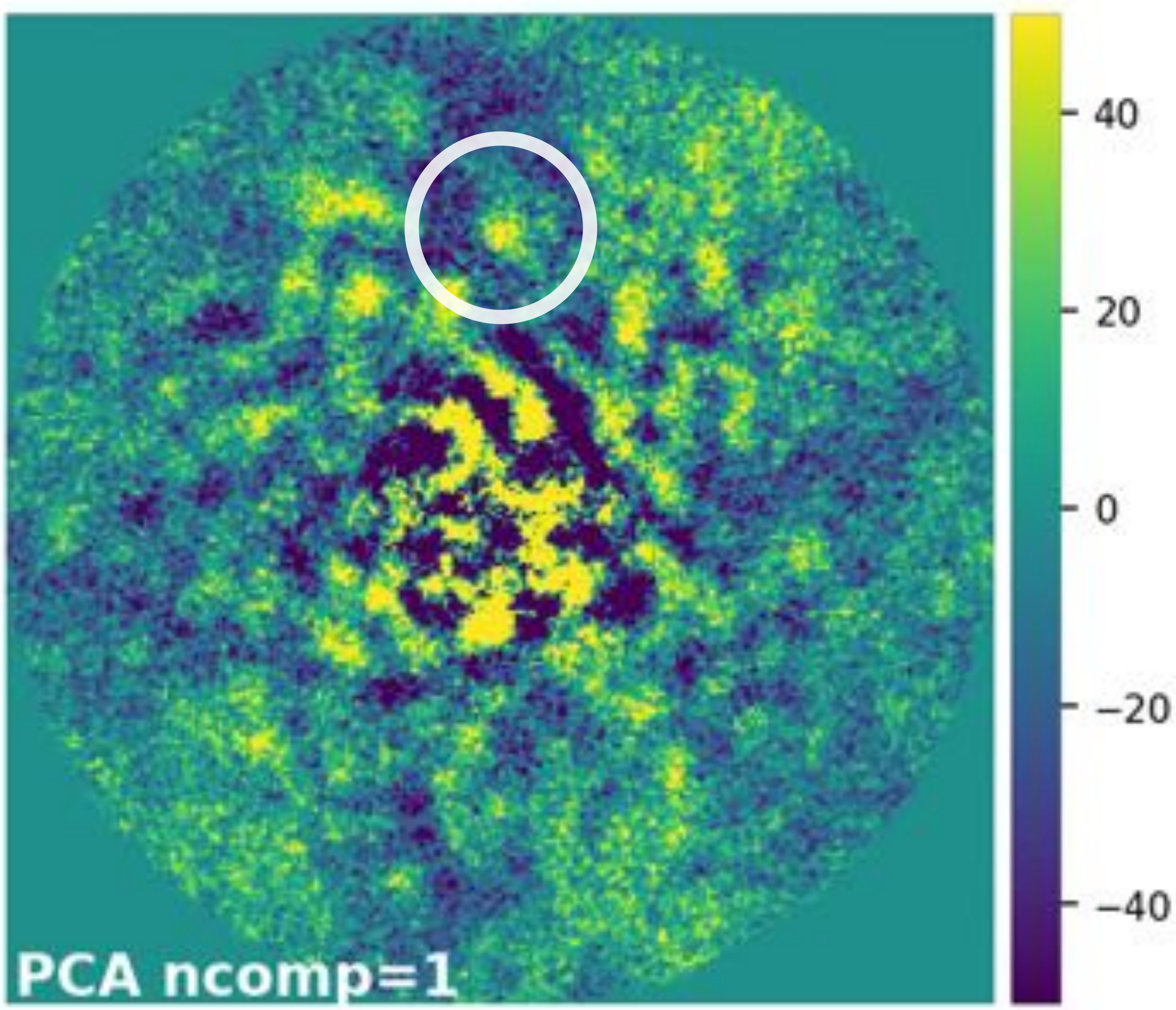
$\mathbf{D} - \mathbf{M} = \mathrm{R}$

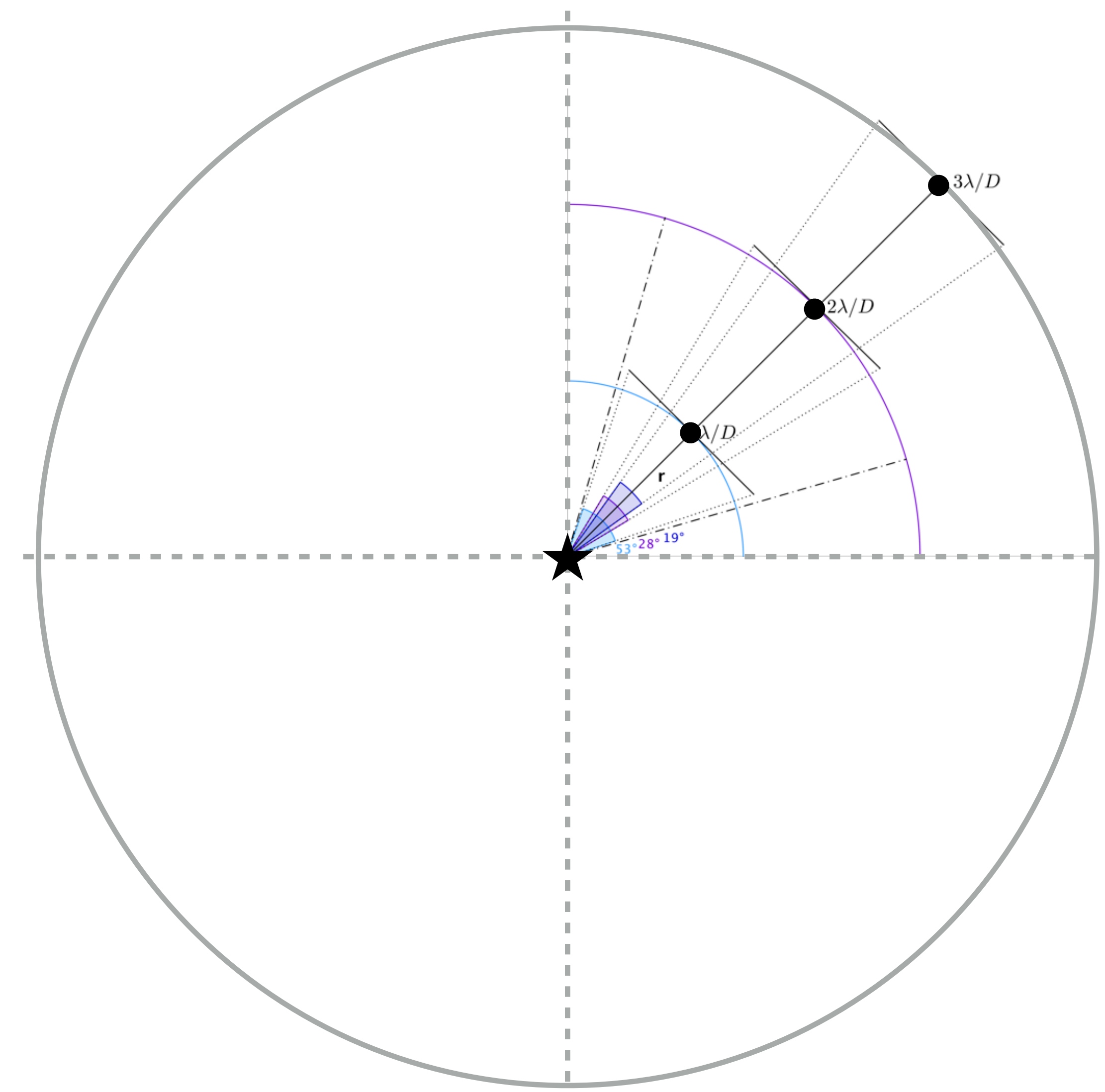
R - residuals containing the exoplanet signal

or

Model PSF subtraction drawbacks

Planet signal is subtracted along with the speckles

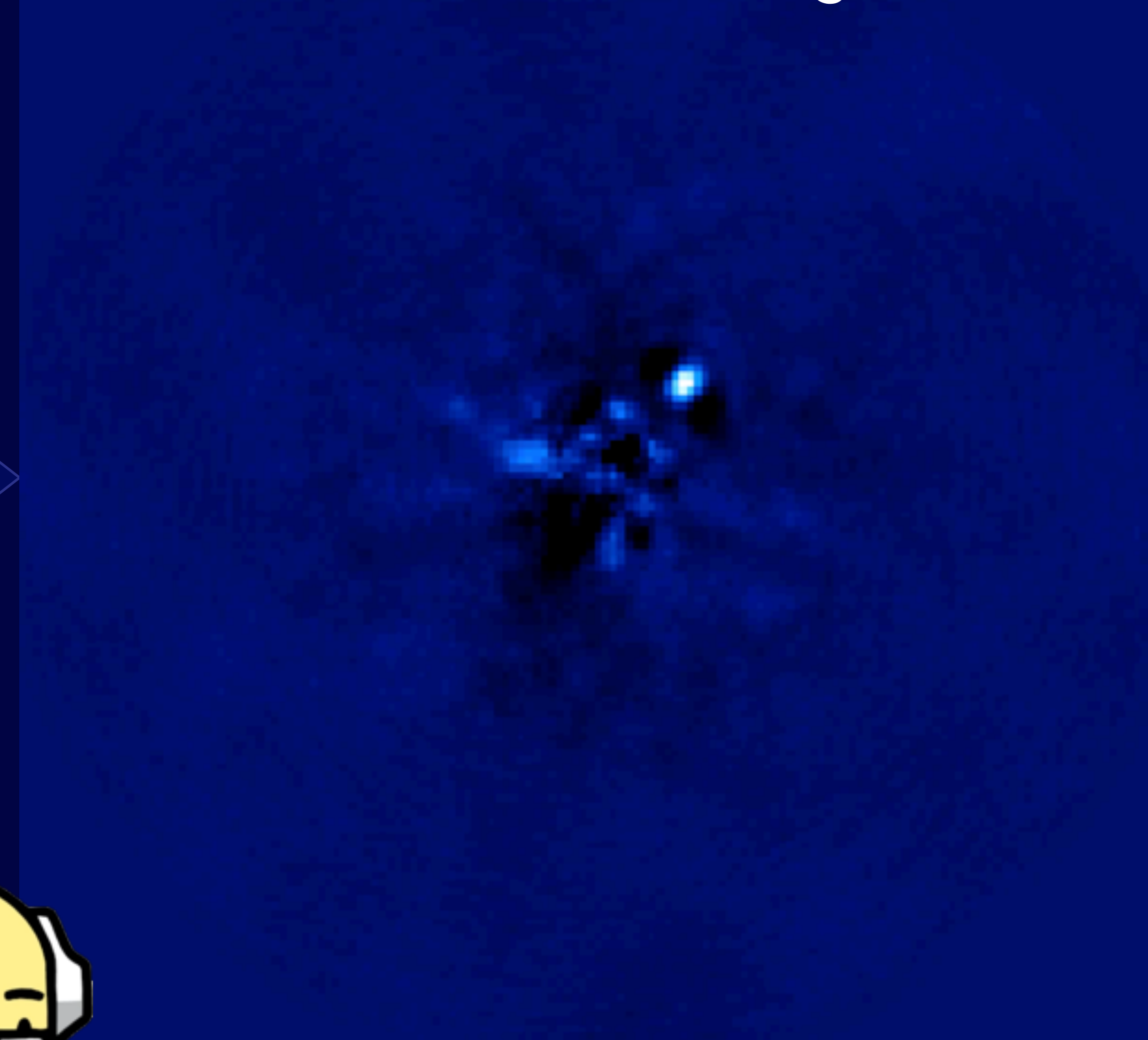




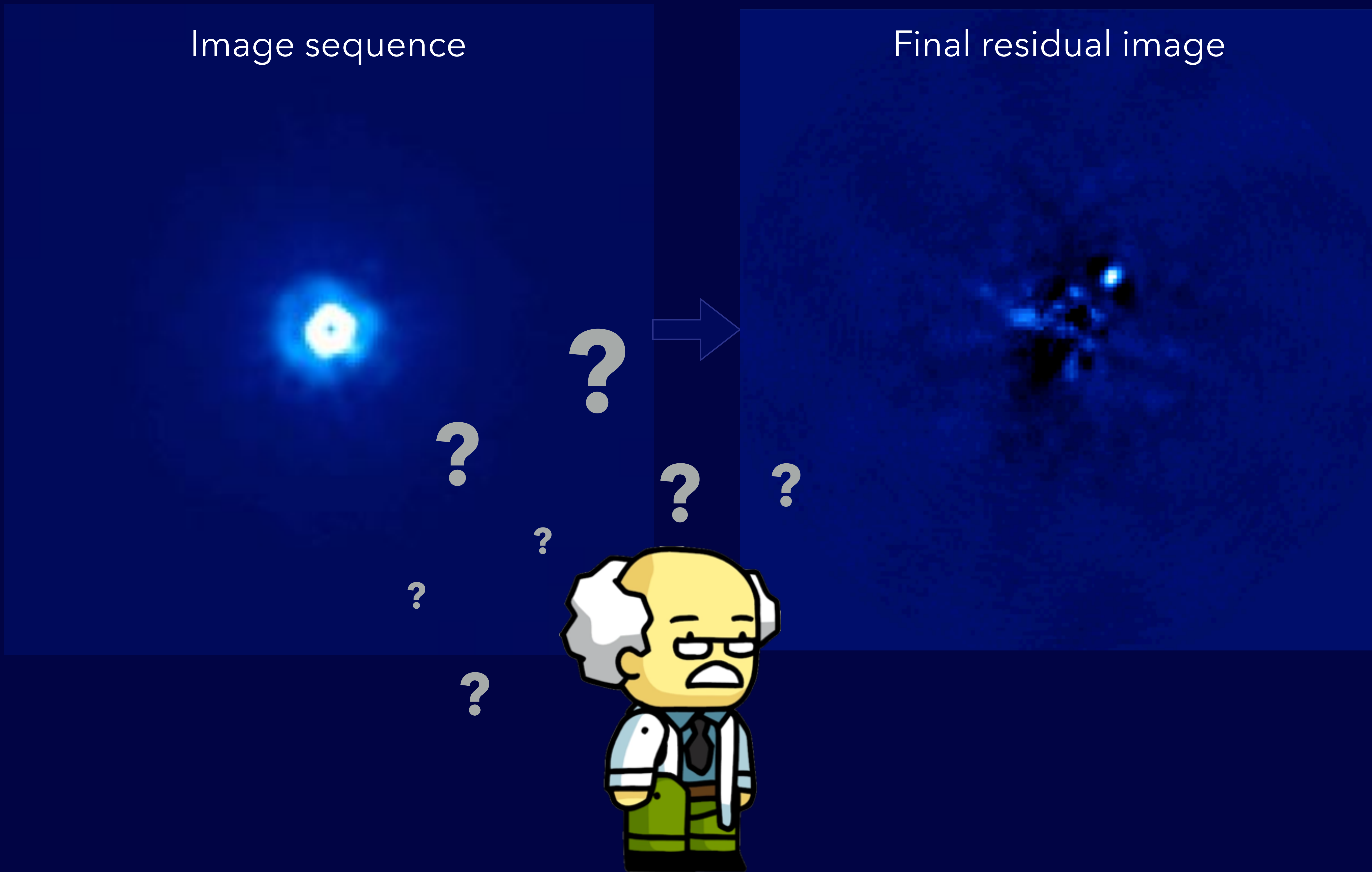
Detection

Image sequence

Final residual image



Detection



Detection

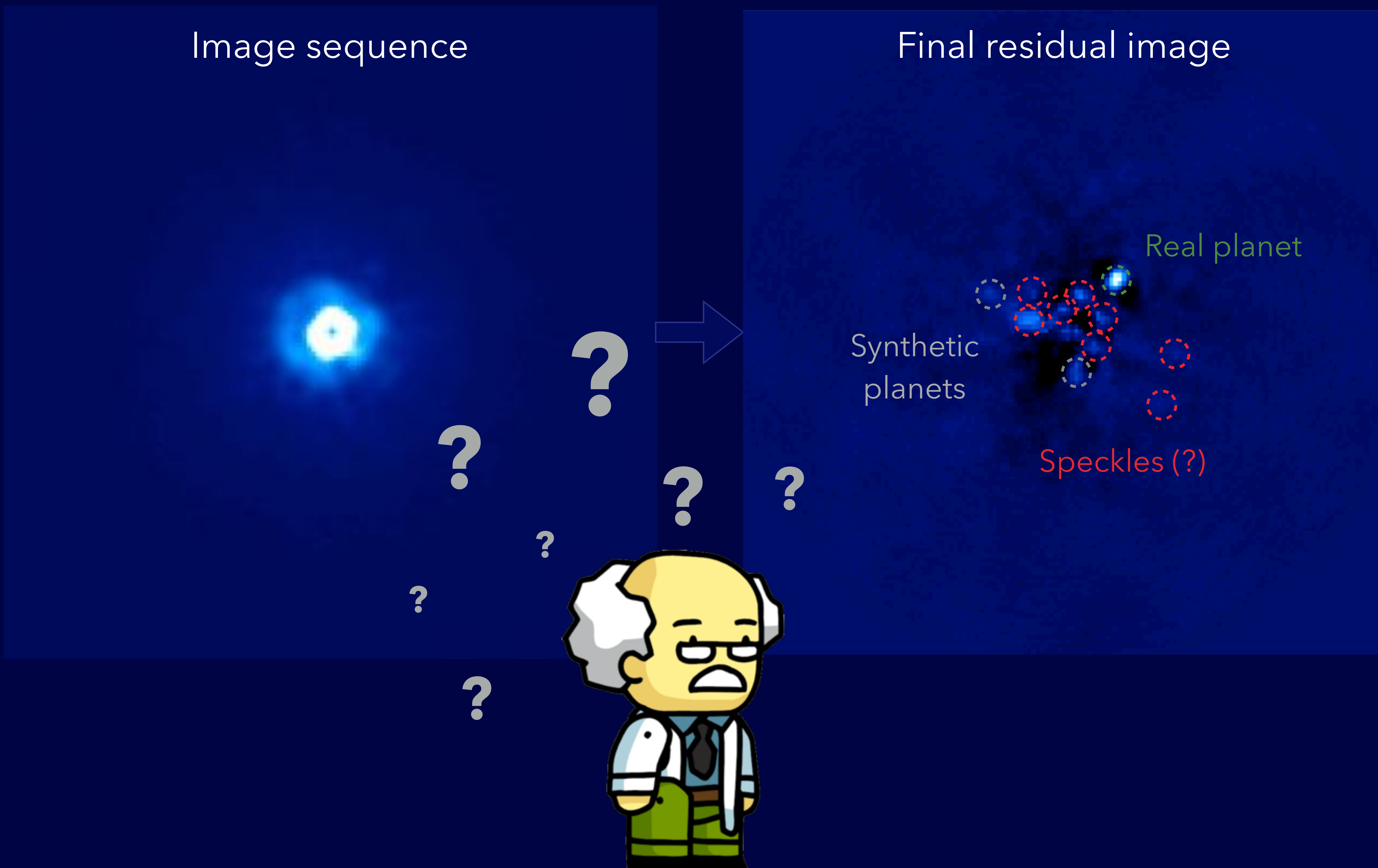
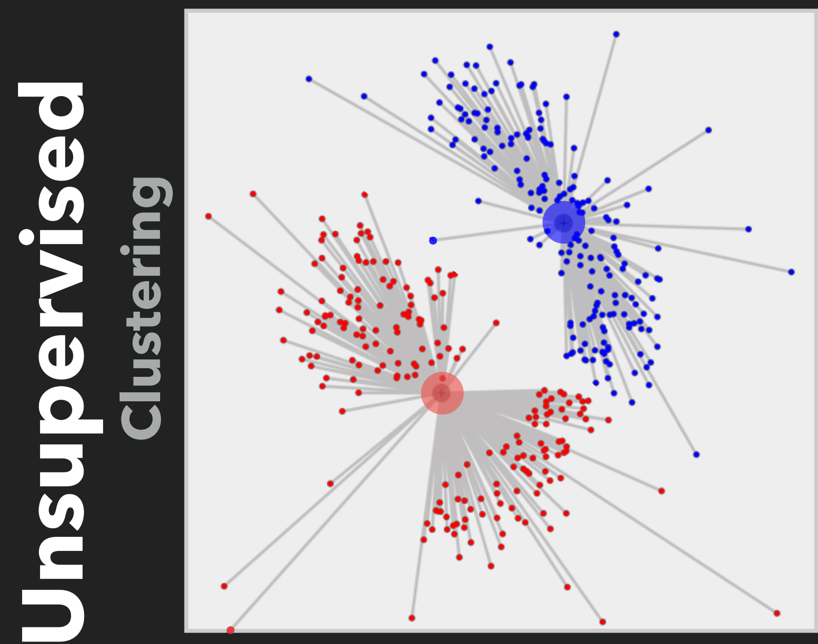
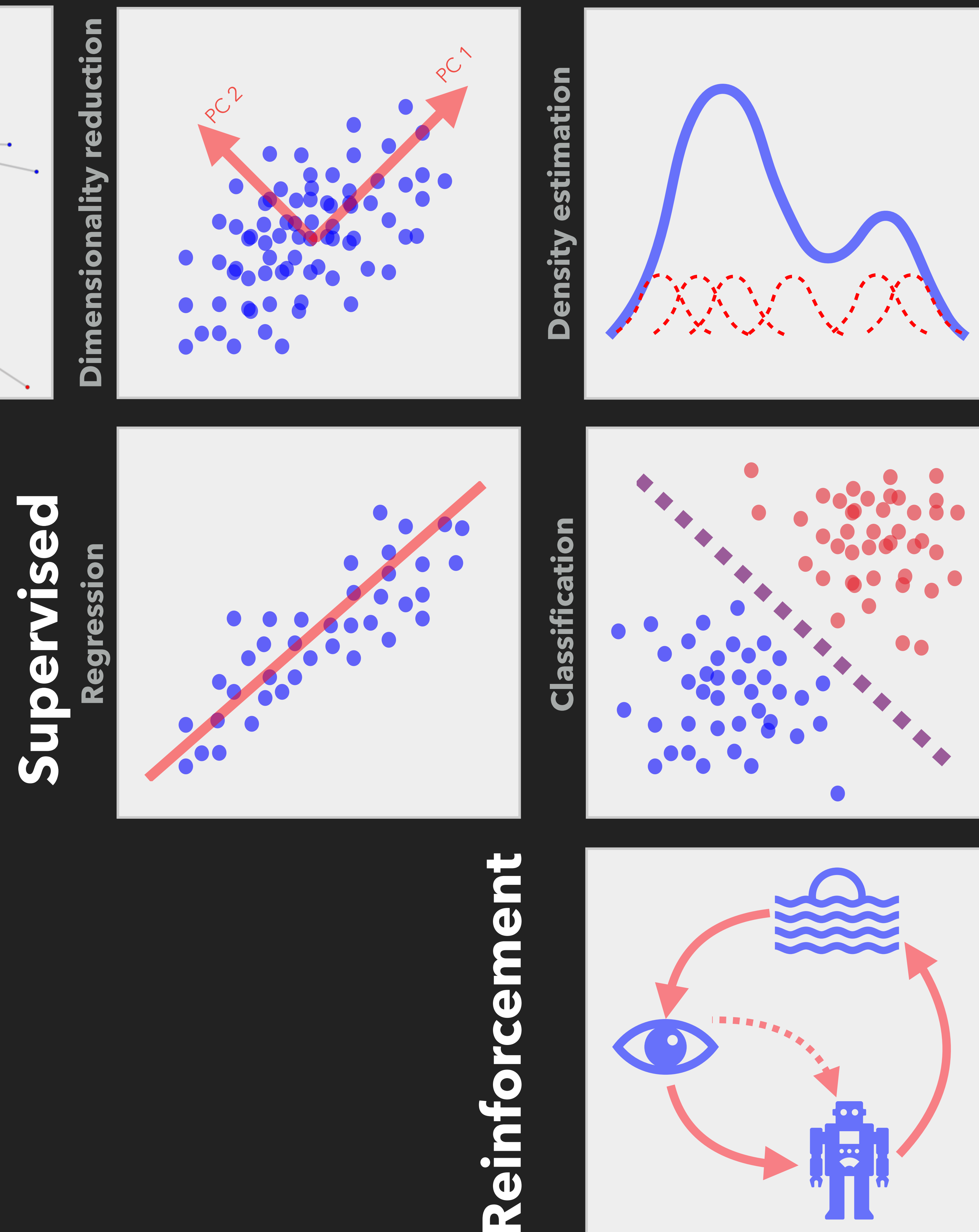


Image sequence
Final residual image
Real planet
Synthetic planets
Speckles (?)


Unsupervised
Clustering


Dimensionality reduction
PC 2
PC 1
Density estimation
Supervised
Regression
Classification
Reinforcement

ML

Supervised learning

$$f : \mathcal{X} \to \mathcal{Y}, \quad (x_i, y_i)_{i=1,\dots,n}$$

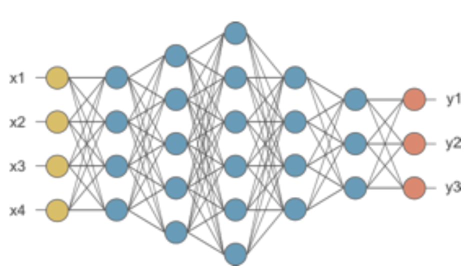
$$f = \underset{f_\theta, \theta \in \Theta}{\arg\min} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) + g(\theta)$$

Training data

Supervised learning

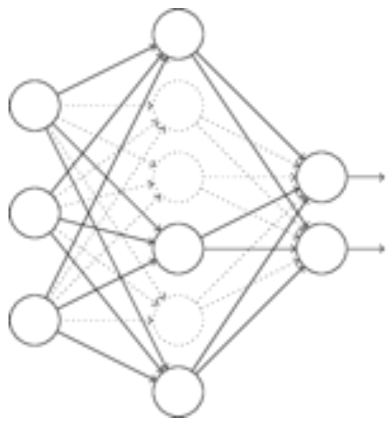
$$f : \mathcal{X} \to \mathcal{Y}, \quad (x_i, y_i)_{i=1,\dots,n}$$

$$f = \underset{\underbrace{f_\theta, \theta \in \Theta}_{\text{Model architecture}}}{\arg\min} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) + g(\theta)$$



Supervised learning

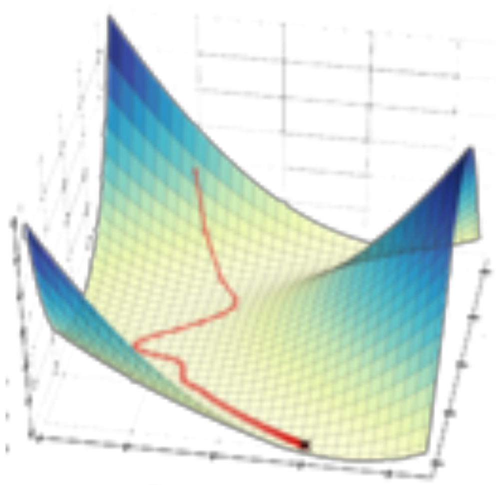
$$f : \mathcal{X} \to \mathcal{Y}, \quad (x_i, y_i)_{i=1,\ldots,n}$$



Loss function and regularization

$$f = \arg\min_{f_\theta, \theta \in \Theta} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) + g(\theta)$$

Supervised learning



$$f : \mathcal{X} \to \mathcal{Y}, \quad (x_i, y_i)_{i=1,\dots,n}$$

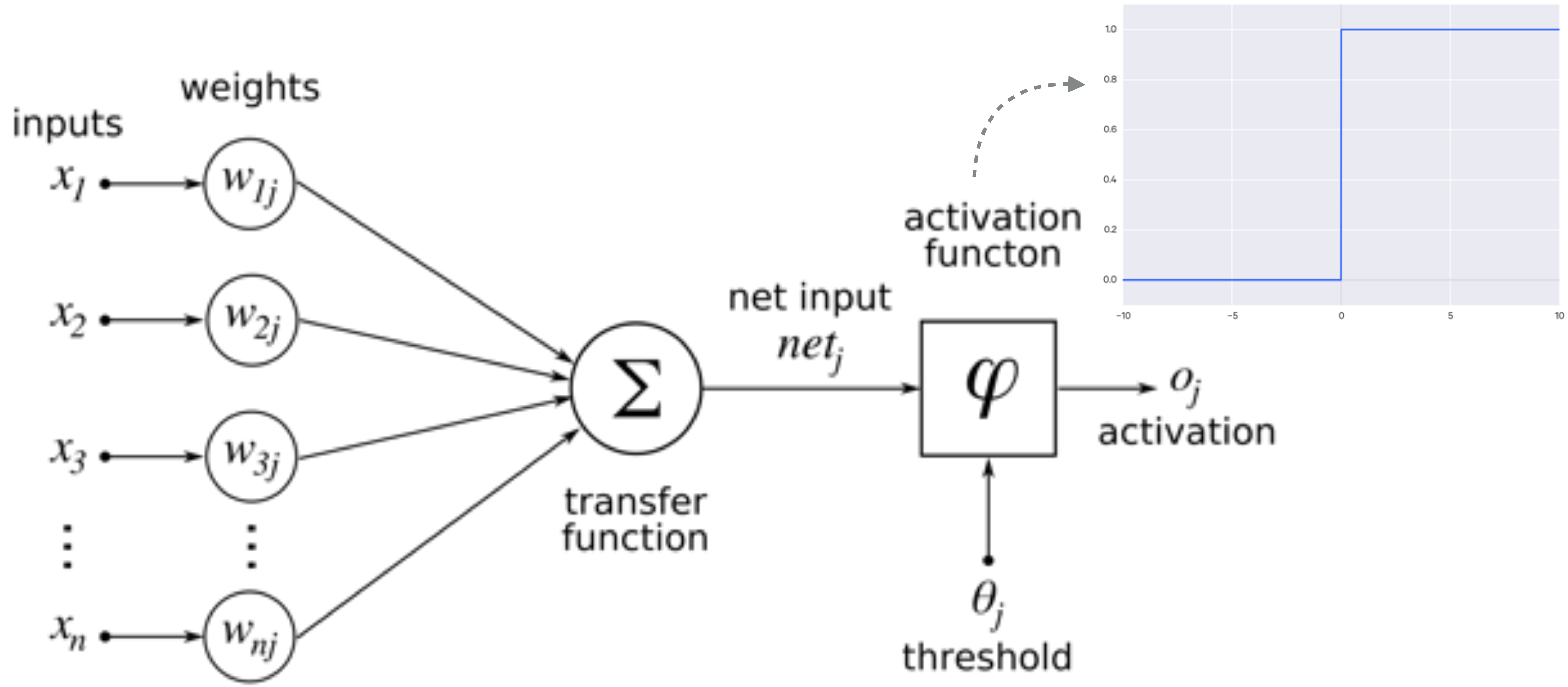
$$f = \overbrace{\arg\min}^{\textbf{Optimization}}_{f_\theta, \theta \in \Theta} \sum_{i=1}^{n} \mathcal{L}(y_i, f_\theta(x_i)) + g(\theta)$$

Input X

1st Layer
(data transformation)

Input X

1st Layer
(data transformation)



Perceptron (Rosenblatt 1958)

Input X

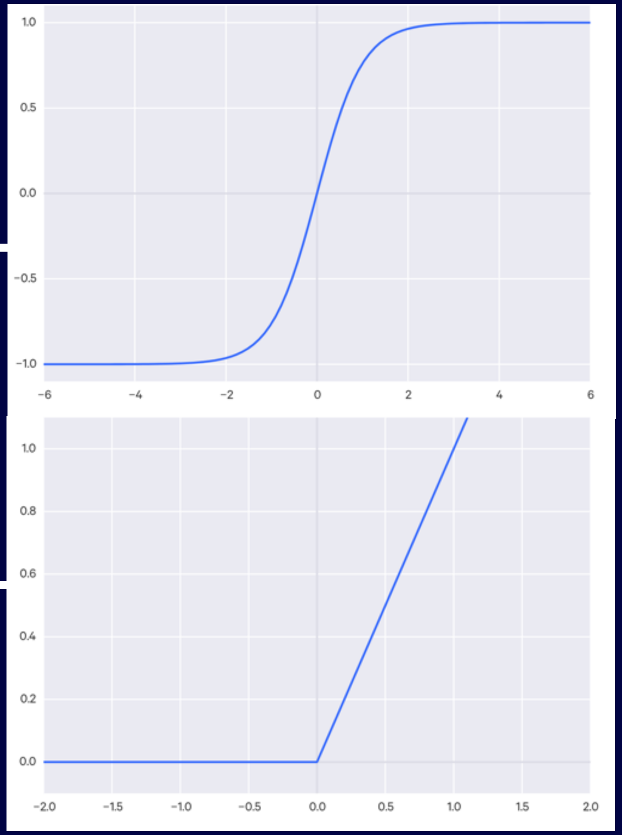
1st Layer
(data transformation)

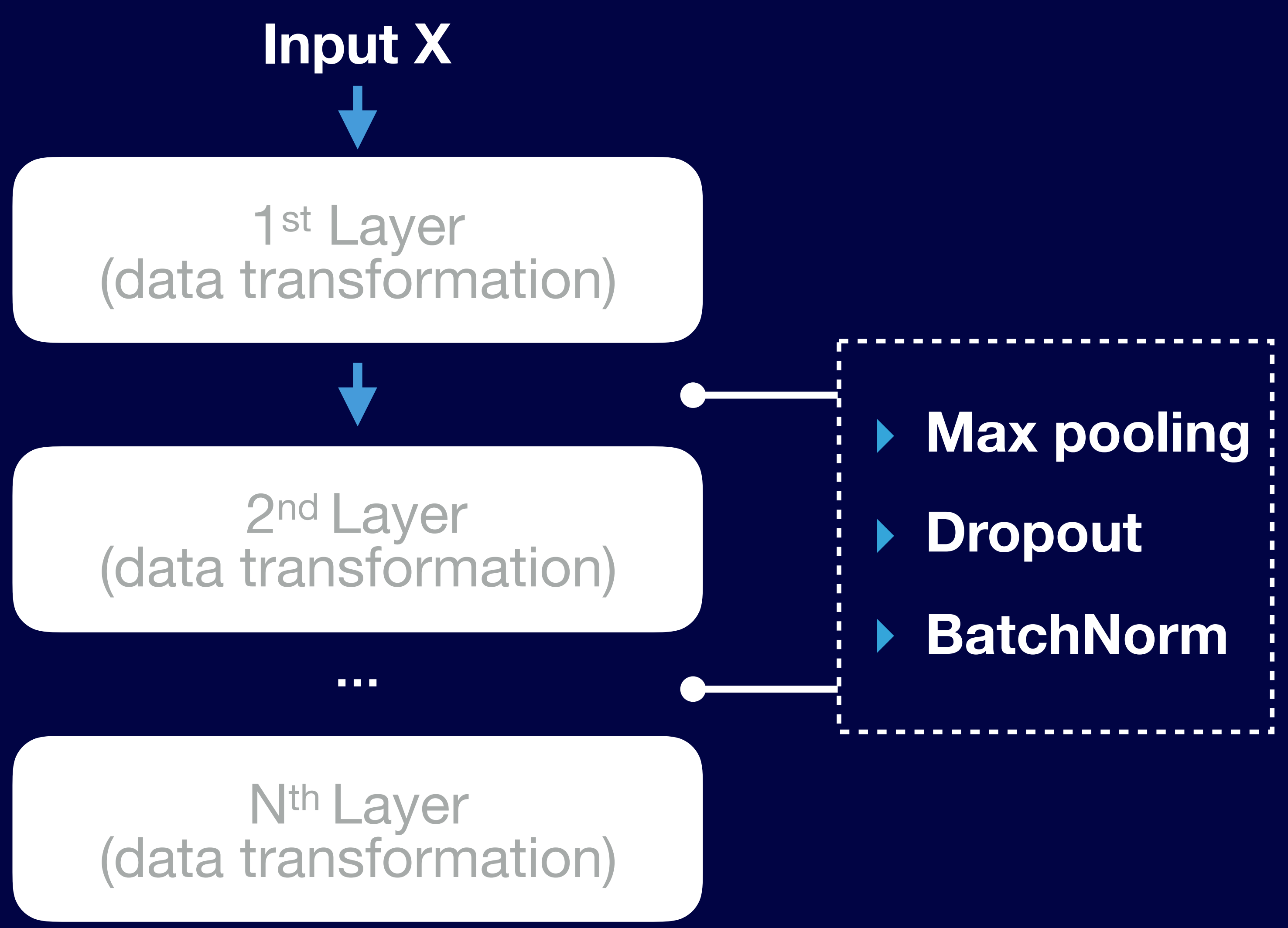
2nd Layer
(data transformation)

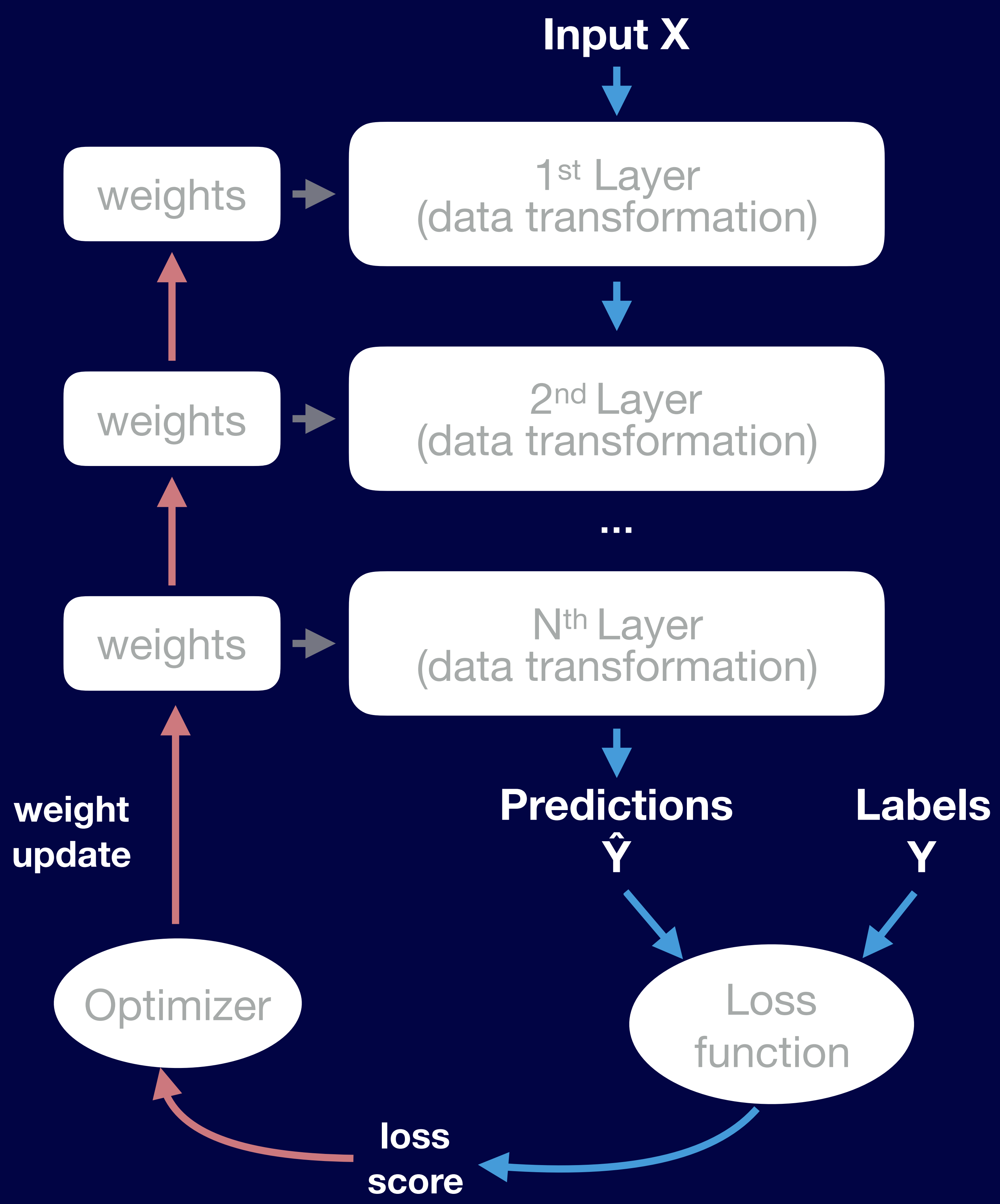
...

Nth Layer
(data transformation)

Activation function

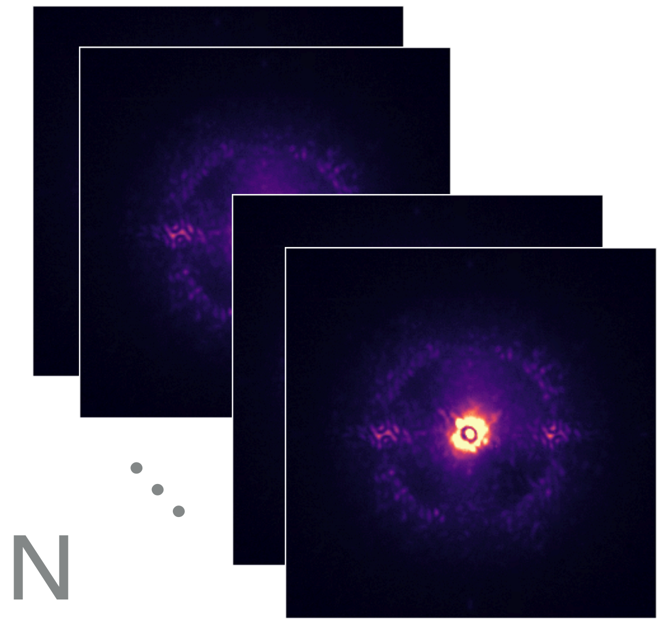


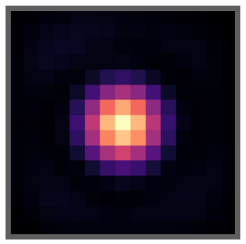

Input X
1st Layer
(data transformation)
2nd Layer
(data transformation)
...
Nth Layer
(data transformation)
Max pooling
Dropout
BatchNorm



Input X
weights
1st Layer
(data transformation)
weights
2nd Layer
(data transformation)
...
weights
Nth Layer
(data transformation)
weight
update
Predictions
Ŷ
Labels
Y
Optimizer
Loss
function
loss
score

Reframing the problem: from unsupervised to supervised learning

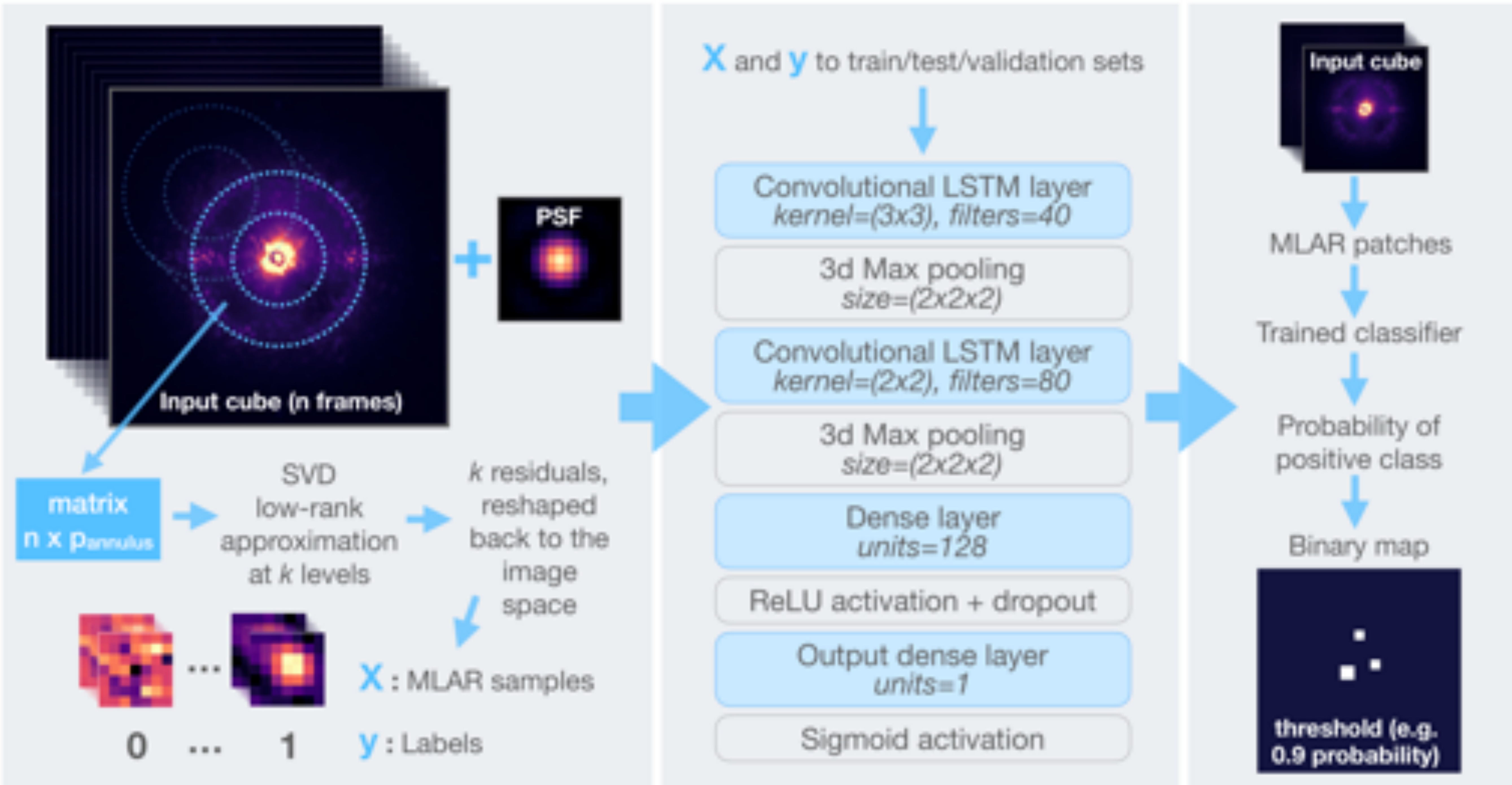




PSF template

- Sequences of images without labels
- Not enough archival data (observed stars)
- We can generate semi-synthetic data by injecting a planet (PSF) template!
- We grab patches: signal/noise

Gomez Gonzalez et al. 2018



Gomez Gonzalez et al. 2018


Sat Oct 6 17:04:47 2018
NVIDIA-SMI 390.25
Driver Version: 390.25
Processes:
GTX 1080Ti

SODINN library

DataLabeler → **Model** → **Predictor**

DataLabeler

- Flux vs S/N sampling
- Fluxes/contrast estimation
- Training data generation
- Data augmentation
- Data persistence (load/save with HDF5)

Model

- Network creation (Keras and Tensorflow)
- Model training
- Model persistence (load/save with HDF5)

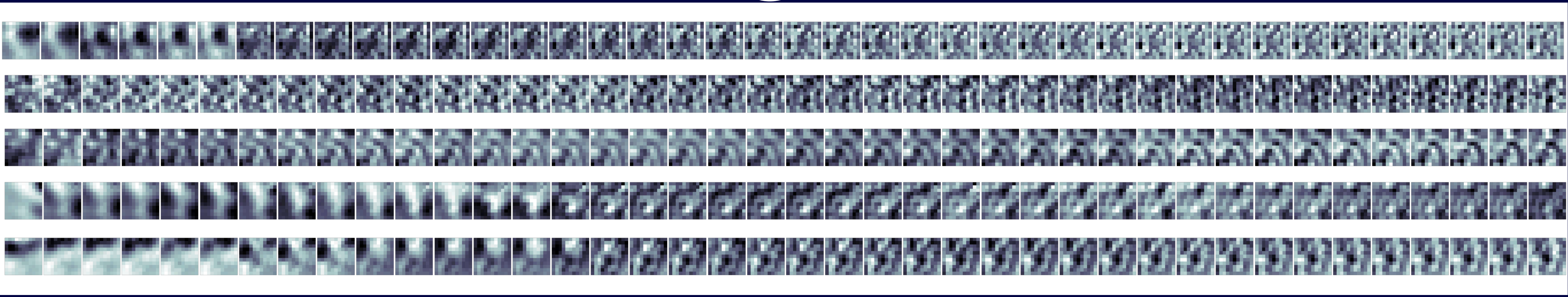
Predictor

- Target samples generation
- Predictions (based on trained model)
- Probability map inspection
- Results to HDF5

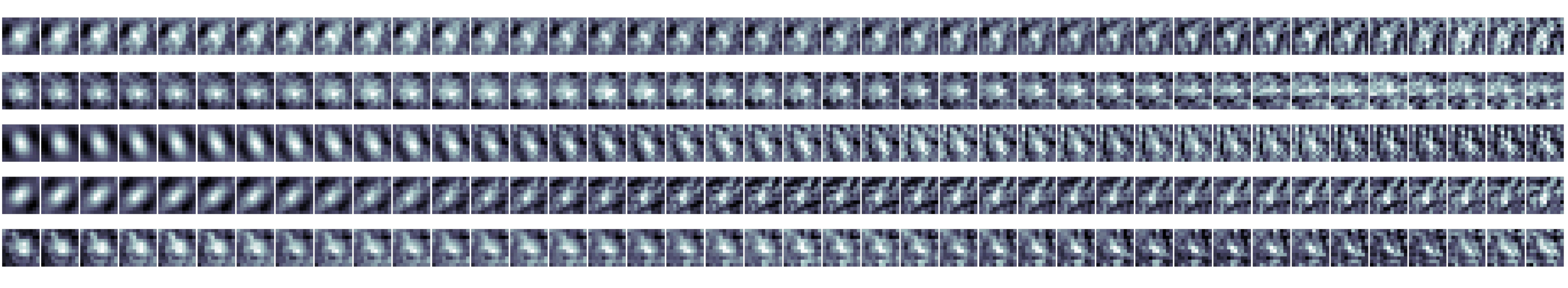
- Reproducible results
- Hyper-parameter and network architecture tuning
- Comparison of labeling strategies

MLAR samples

c^-



c^+



Corresponding labels: $y \in \{c^-, c^+\}$

Discriminative model: Neural Network

Learning a mapping function

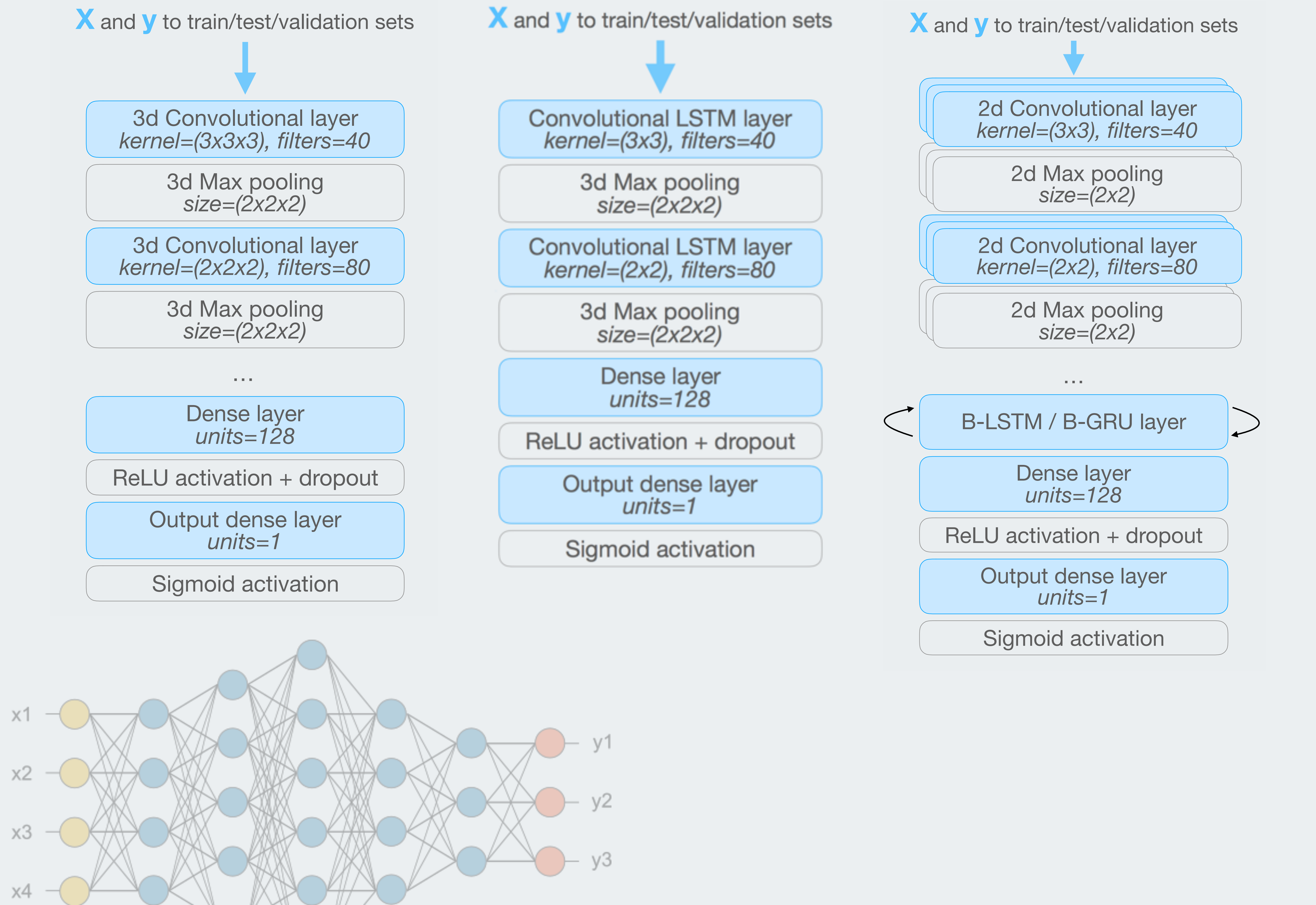
$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

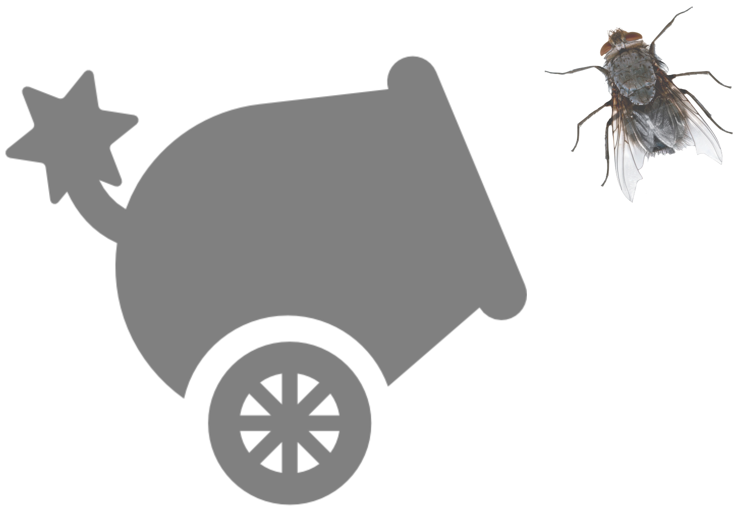
Goal - to make correct predictions on new samples:

$$\hat{y} = p(c^{+} | \,\mathrm{MLAR\ sample})$$

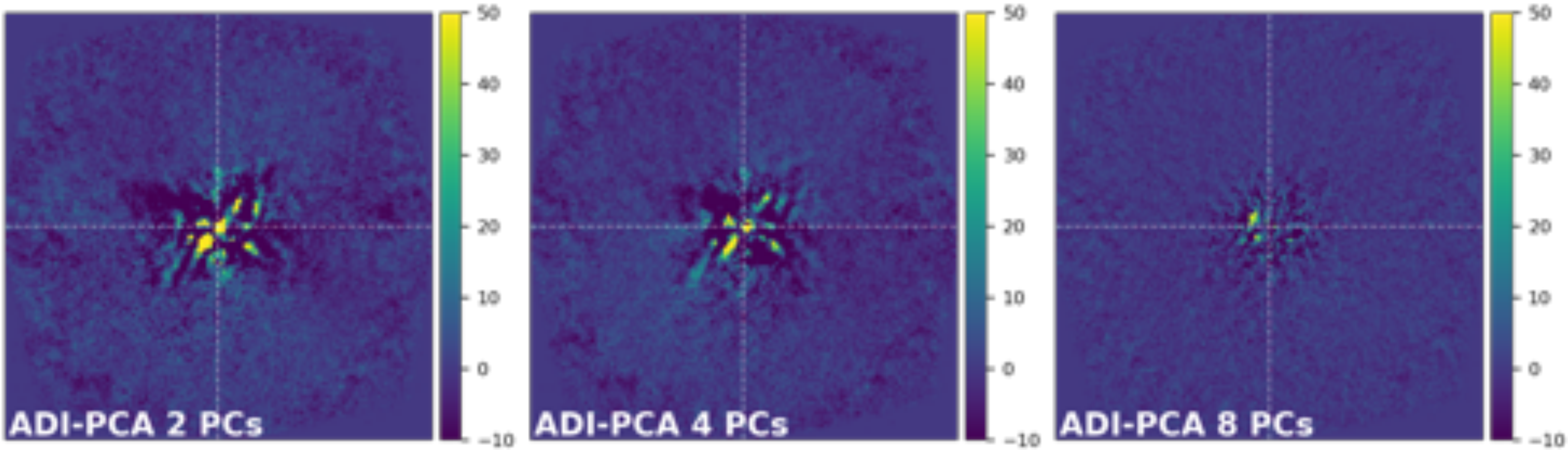
SGD with a binary cross-entropy loss:

$$\mathcal{L} = -\sum_{n} (y_n \ln(\hat{y}_n) + (1 - y_n) \ln(1 - \hat{y}_n))$$

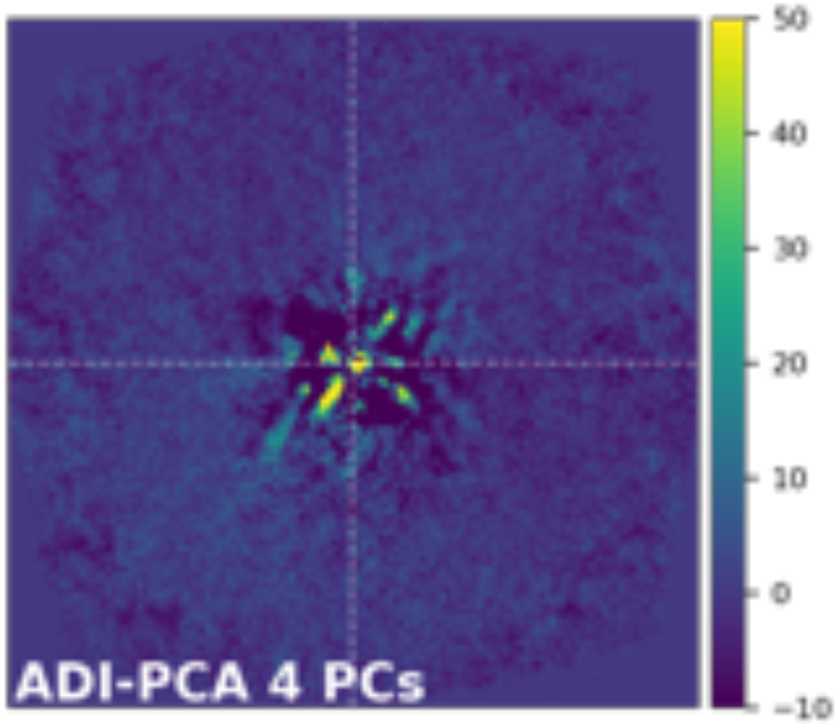

X and y to train/test/validation sets
3d Convolutional layer
kernel=(3x3x3), filters=40
3d Max pooling
size=(2x2x2)
3d Convolutional layer
kernel=(2x2x2), filters=80
3d Max pooling
size=(2x2x2)
...
Dense layer
units=128
ReLU activation + dropout
Output dense layer
units=1
Sigmoid activation
X and y to train/test/validation sets
Convolutional LSTM layer
kernel=(3x3), filters=40
3d Max pooling
size=(2x2x2)
Convolutional LSTM layer
kernel=(2x2), filters=80
3d Max pooling
size=(2x2x2)
Dense layer
units=128
ReLU activation + dropout
Output dense layer
units=1
Sigmoid activation
X and y to train/test/validation sets
2d Convolutional layer
kernel=(3x3), filters=40
2d Max pooling
size=(2x2)
2d Convolutional layer
kernel=(2x2), filters=80
2d Max pooling
size=(2x2)
...
B-LSTM / B-GRU layer
Dense layer
units=128
ReLU activation + dropout
Output dense layer
units=1
Sigmoid activation
x1
x2
x3
x4
y1
y2
y3



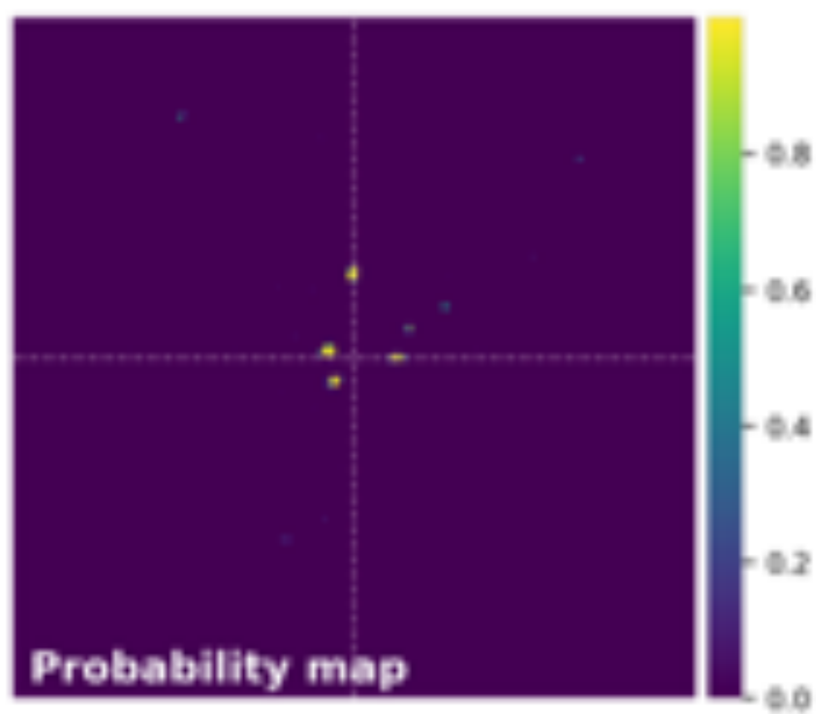
Let’s inject some fake planets

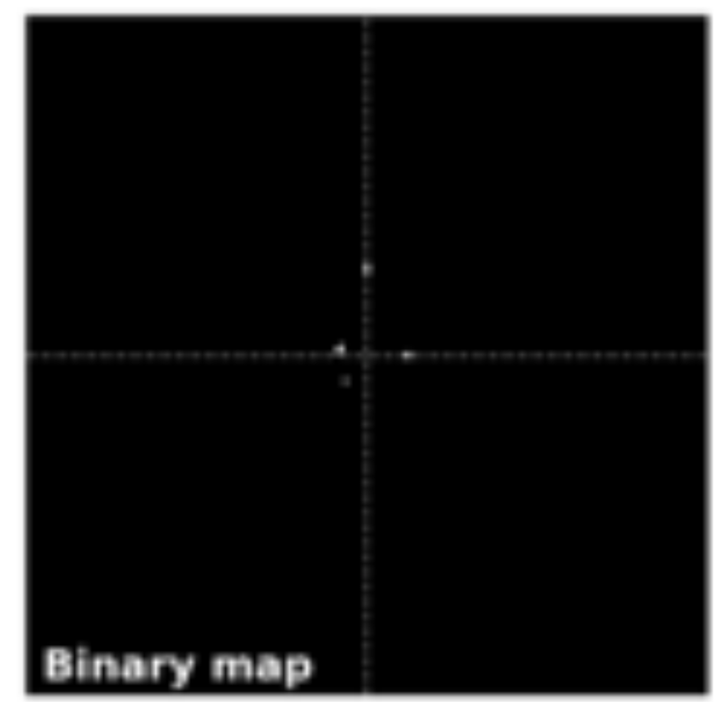


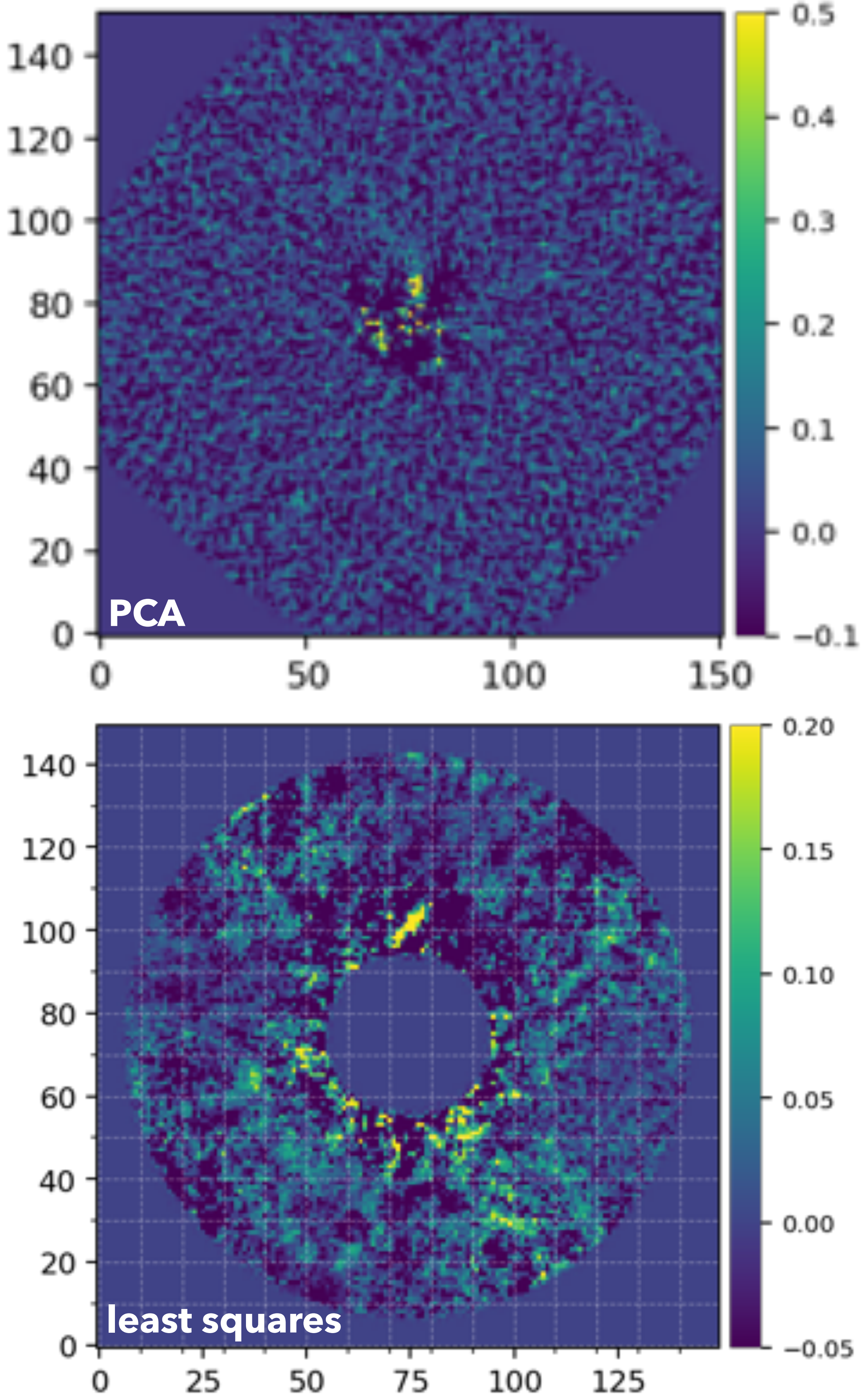
Let’s inject some fake planets



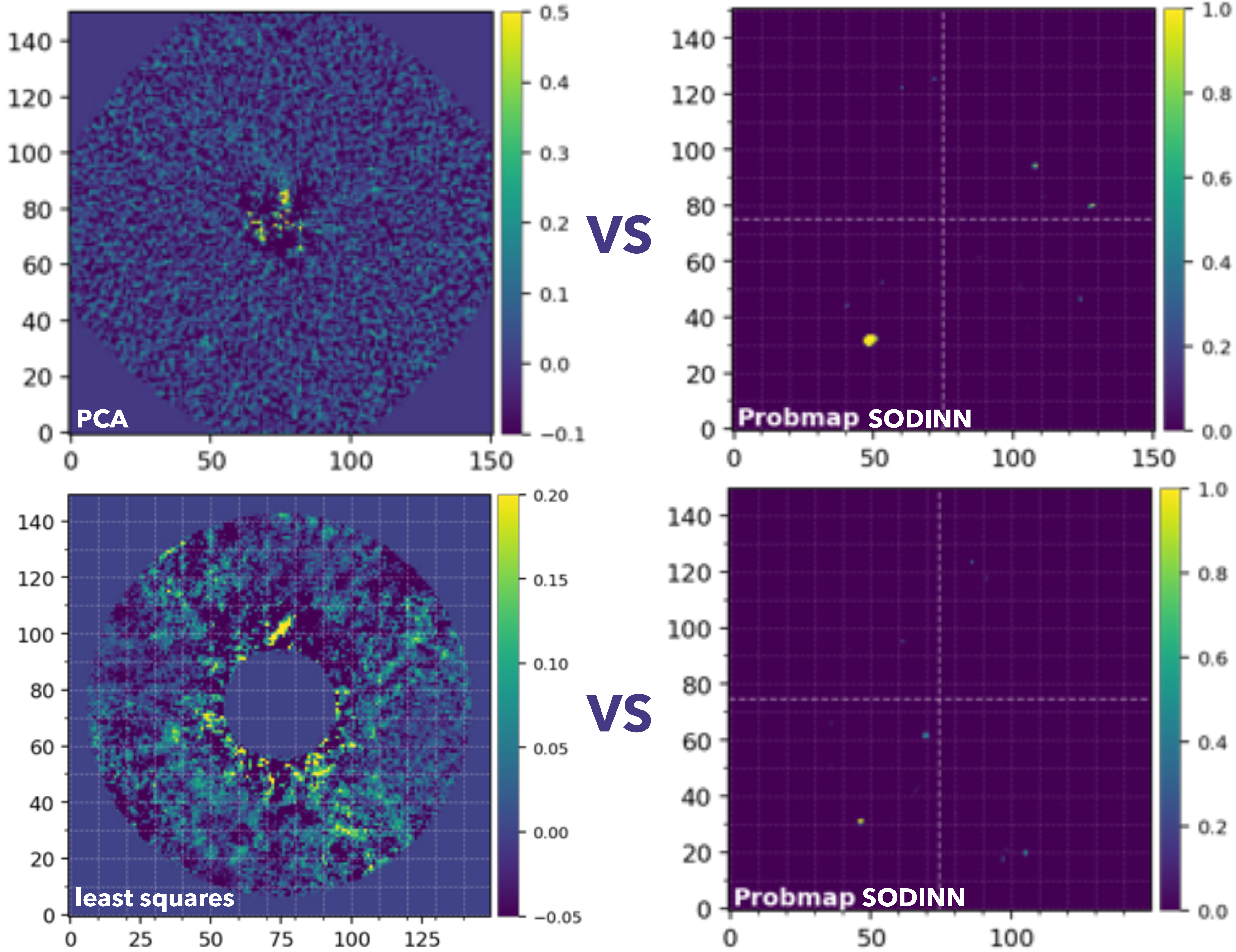
VS







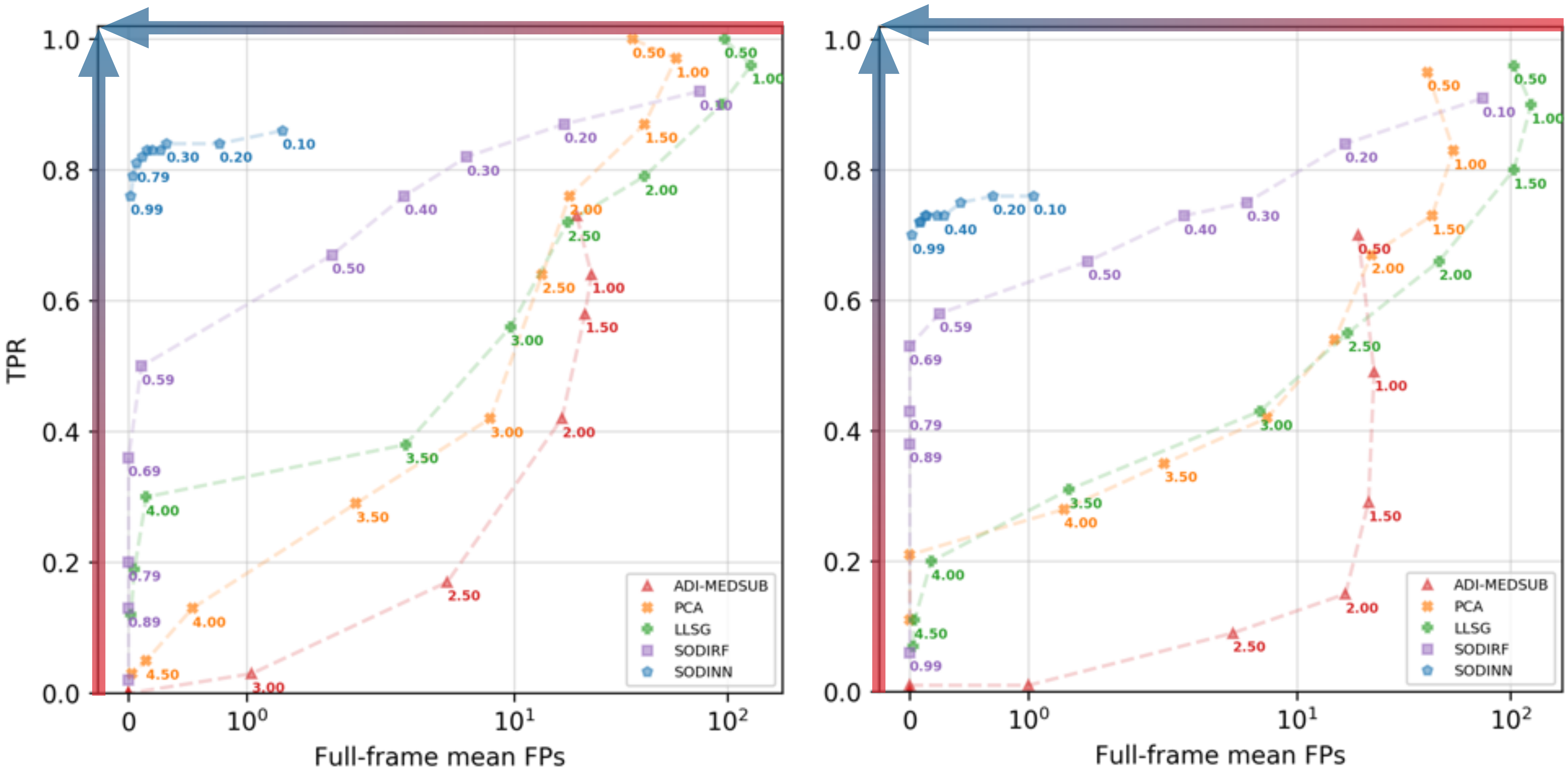
Find the exoplanet ;)



PCA
VS
Probmap SODINN
least squares
VS
Probmap SODINN

Performance assessment

Receiver operating characteristic curves

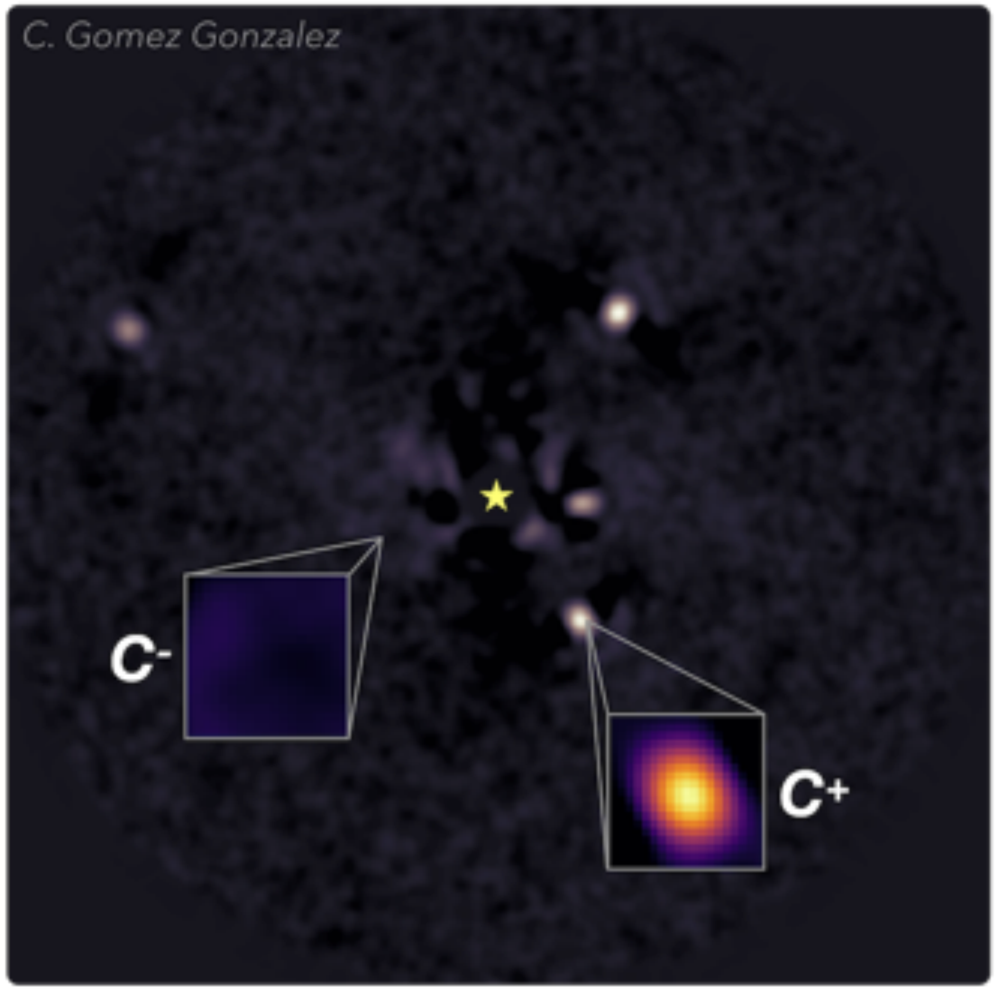


Do you have an idea for a new algorithm?

We are about to launch a data challenge!

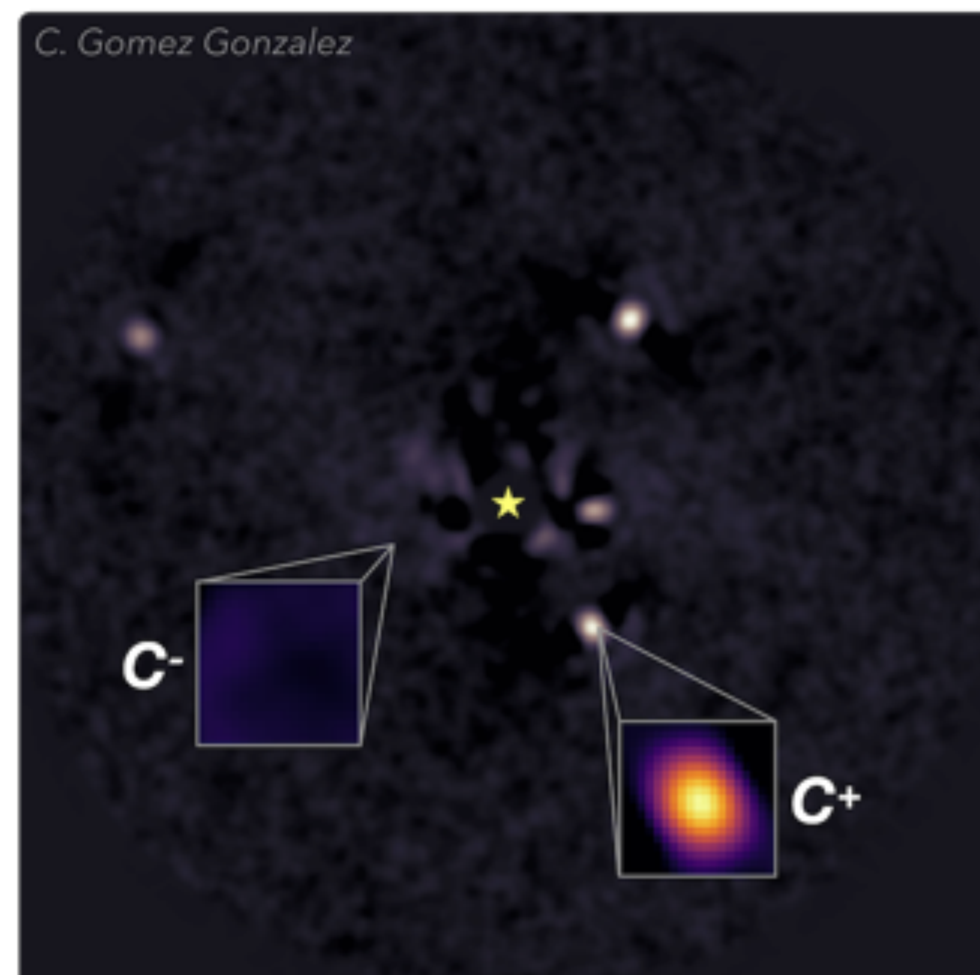
https://carlgogo.github.io/exoimaging_challenge/

https://github.com/carlgogo/exoimaging_challenge_extras



- Data from the most representative instruments
- Metrics:
  - the true positive rate (TPR) also known as sensitivity or recall: `TPR = TPs / Ninj`,
  - the false discovery/detection rate (FDR): `FDR = FPs / Ndet`,
  - the precision or positive predictive value (PPV): `PPV = TPs / Ndet`,
  - the F1-score or harmonic mean of TPR and the precision: `F1 = 2 * PPV * TPR / (PPV + TPR)`.
- Phases:

	3D Sub-challenge - ADI	4D Sub-challenge - ADI+IFS
Phase	**Metric**	**Metric**
1	F1, TPR and FDR	F1, TPR and FDR
2	ROC space	ROC space



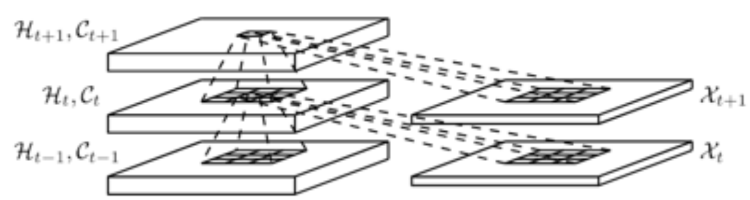
- Will run on Codalab

https://carlgogo.github.io/exoimaging_challenge/

https://github.com/carlgogo/exoimaging_challenge_extras

Connections with other fields

Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting



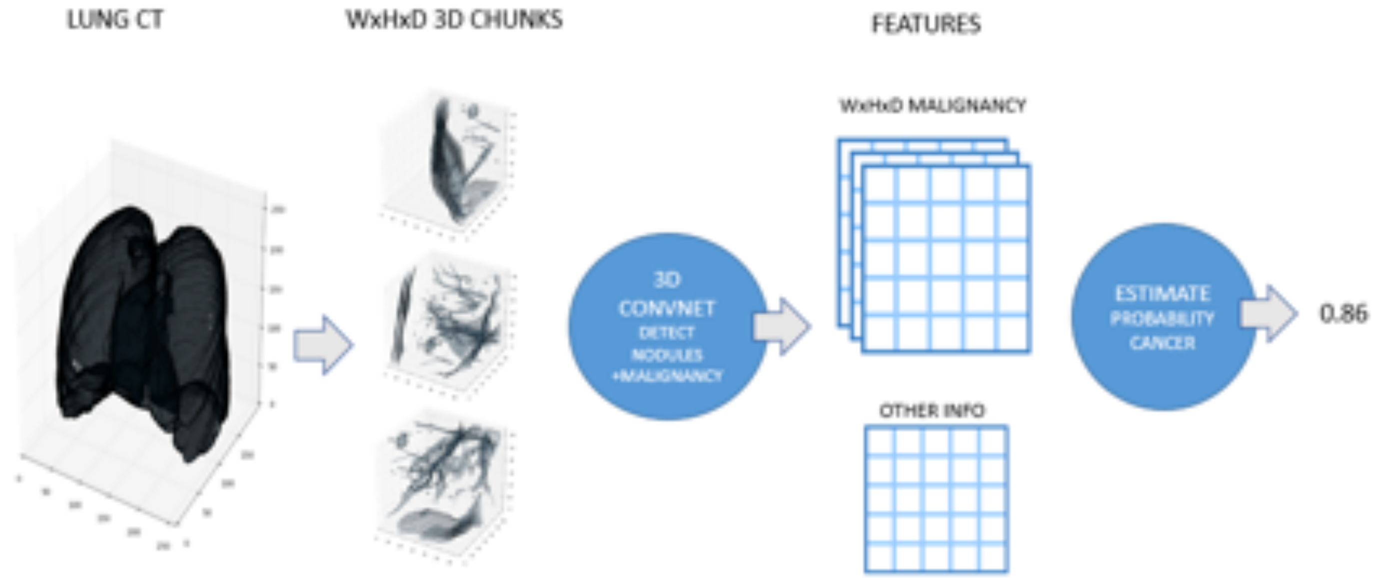
Xingjian Shi, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, Wang-chun Woo

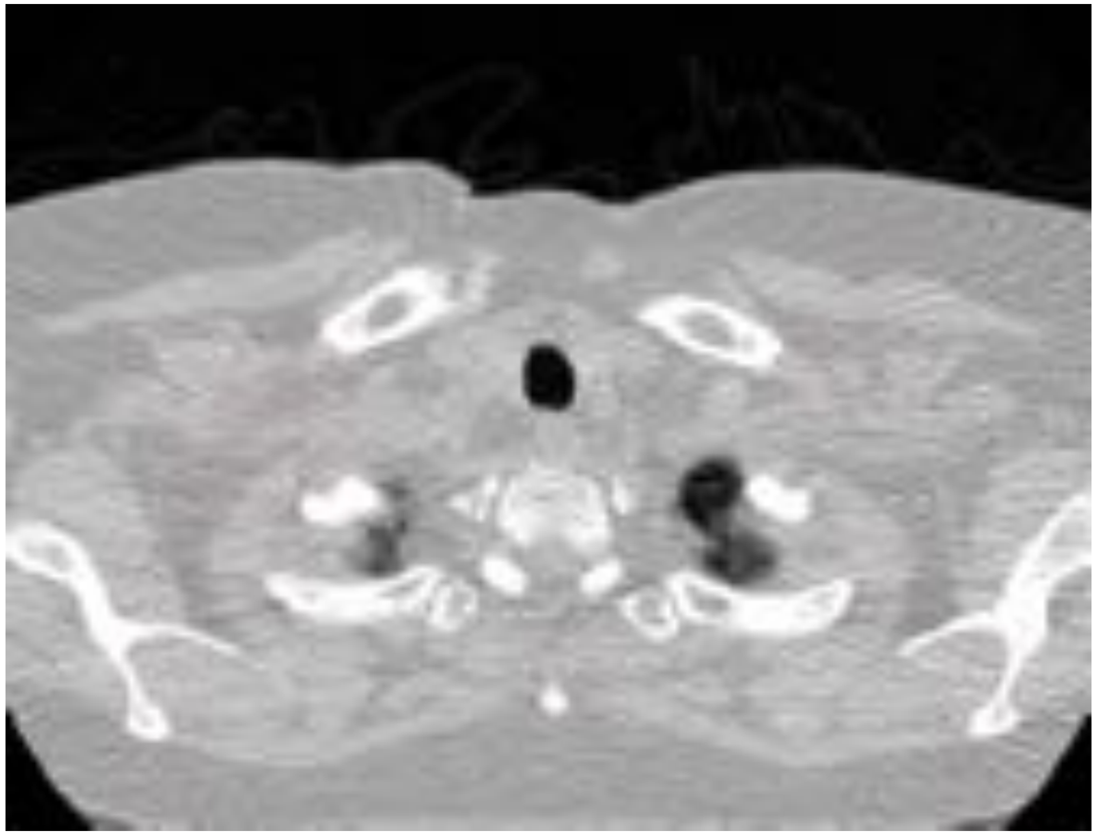
(Submitted on 13 Jun 2015 (v1), last revised 19 Sep 2015 (this version, v2))

The goal of precipitation nowcasting is to predict the future rainfall intensity in a local region over a relatively short period of time. Very few previous studies have examined this crucial and challenging weather forecasting problem from the machine learning perspective. In this paper, we formulate precipitation nowcasting as a spatiotemporal sequence forecasting problem in which both the input and the prediction target are spatiotemporal sequences. By extending the fully connected LSTM (FC-LSTM) to have convolutional structures in both the input-to-state and state-to-state transitions, we propose the convolutional LSTM (ConvLSTM) and use it to build an end-to-end trainable model for the precipitation nowcasting problem. Experiments show that our ConvLSTM network captures spatiotemporal correlations better and consistently outperforms FC-LSTM and the state-of-the-art operational ROVER algorithm for precipitation nowcasting.

https://arxiv.org/pdf/1506.04214.pdf

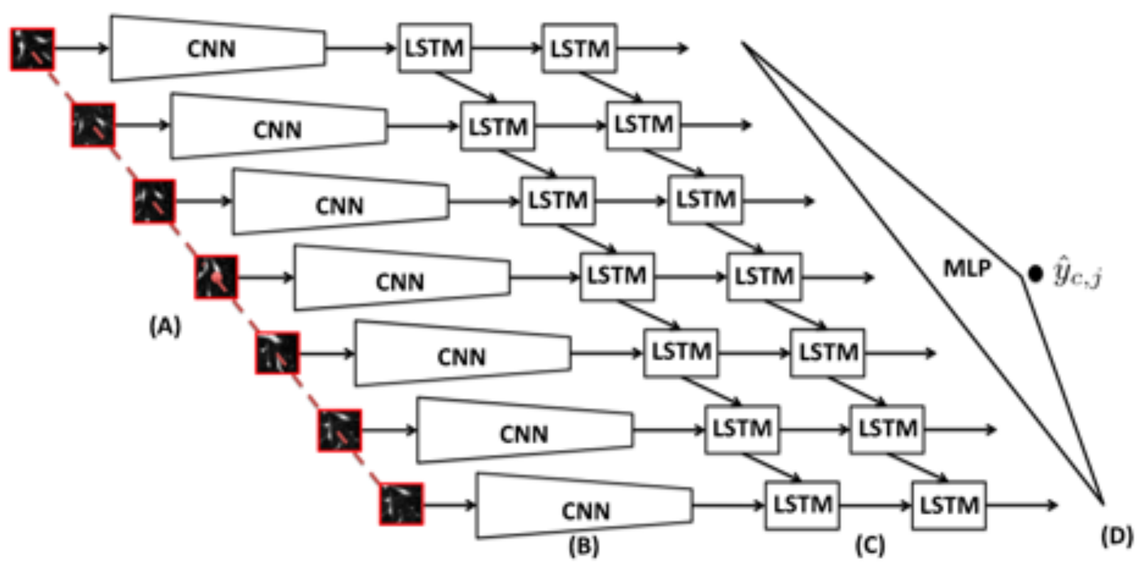
Connections with other fields



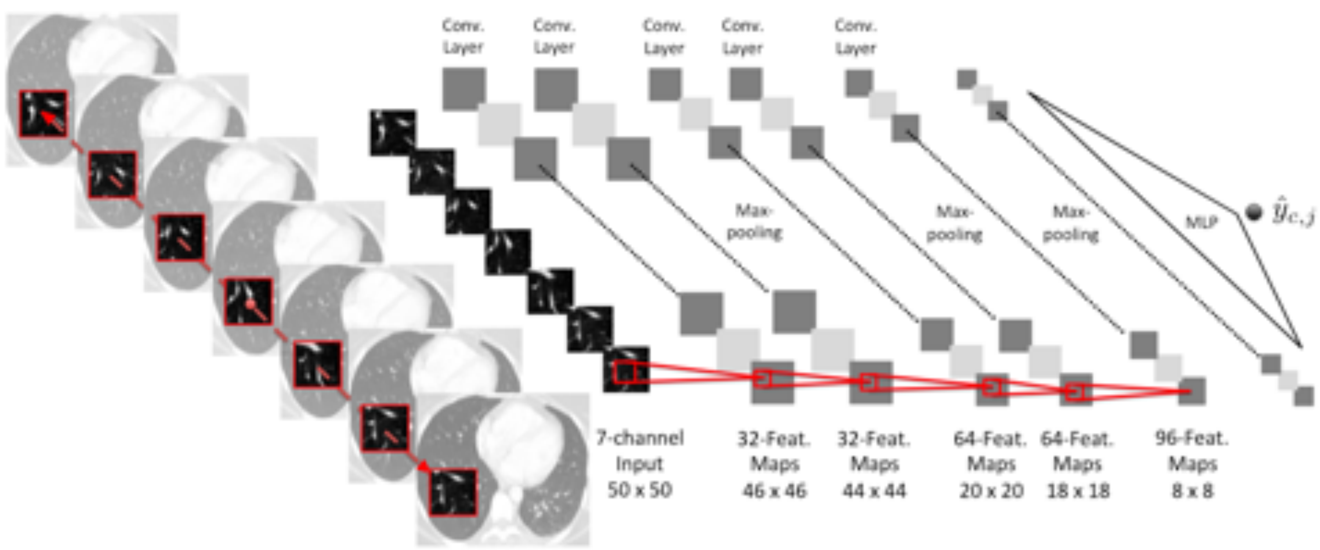


http://juliandewit.github.io/kaggle-ndsb2017/

Connections with other fields

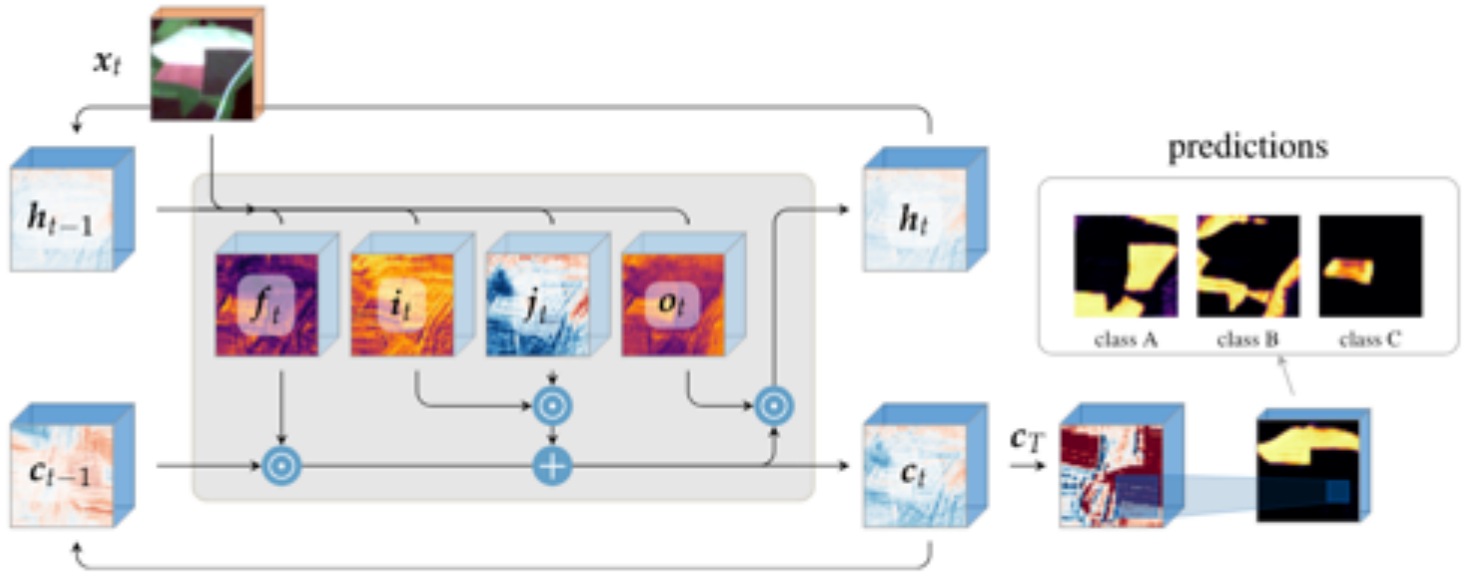


ReCTnet architecture for object detection in multi-slice medical images

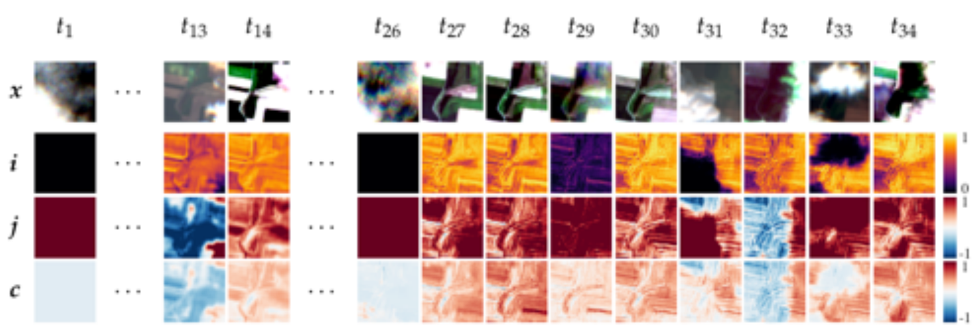


https://arxiv.org/pdf/1609.09143.pdf

Connections with other fields



two-component convolutional long short-term memory network (LSTM)



Convolutional LSTMs for Cloud-Robust Segmentation of Remote Sensing Imagery

https://arxiv.org/pdf/1811.02471.pdf

¡Gracias!

- carlgogo.github.io/
- github.com/carlgogo/
- speakerdeck.com/carlgogo/
- linkedin.com/in/carlgogo/