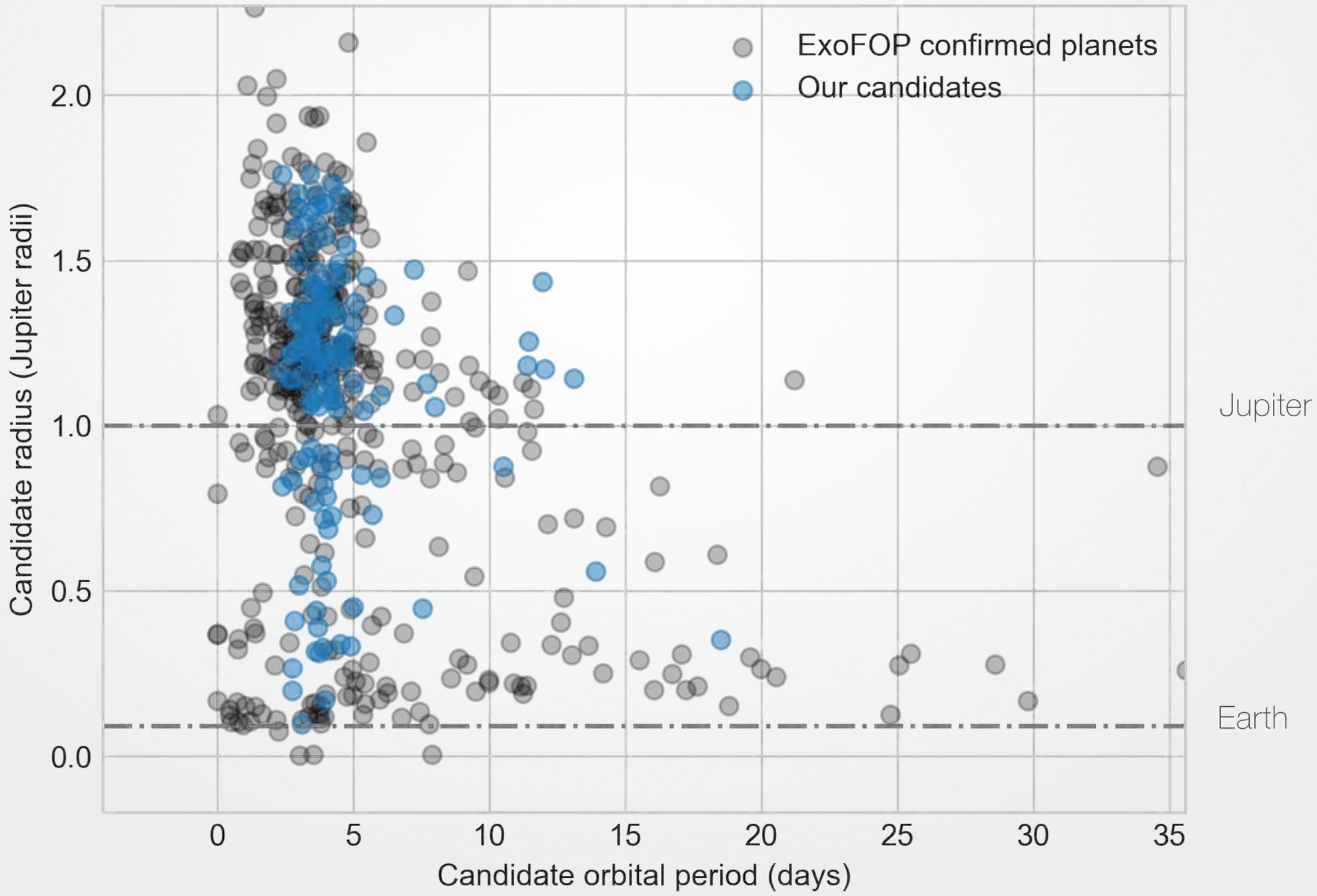
A GENERALIZED PHOTOMETRIC NEURAL NETWORK FRAMEWORK

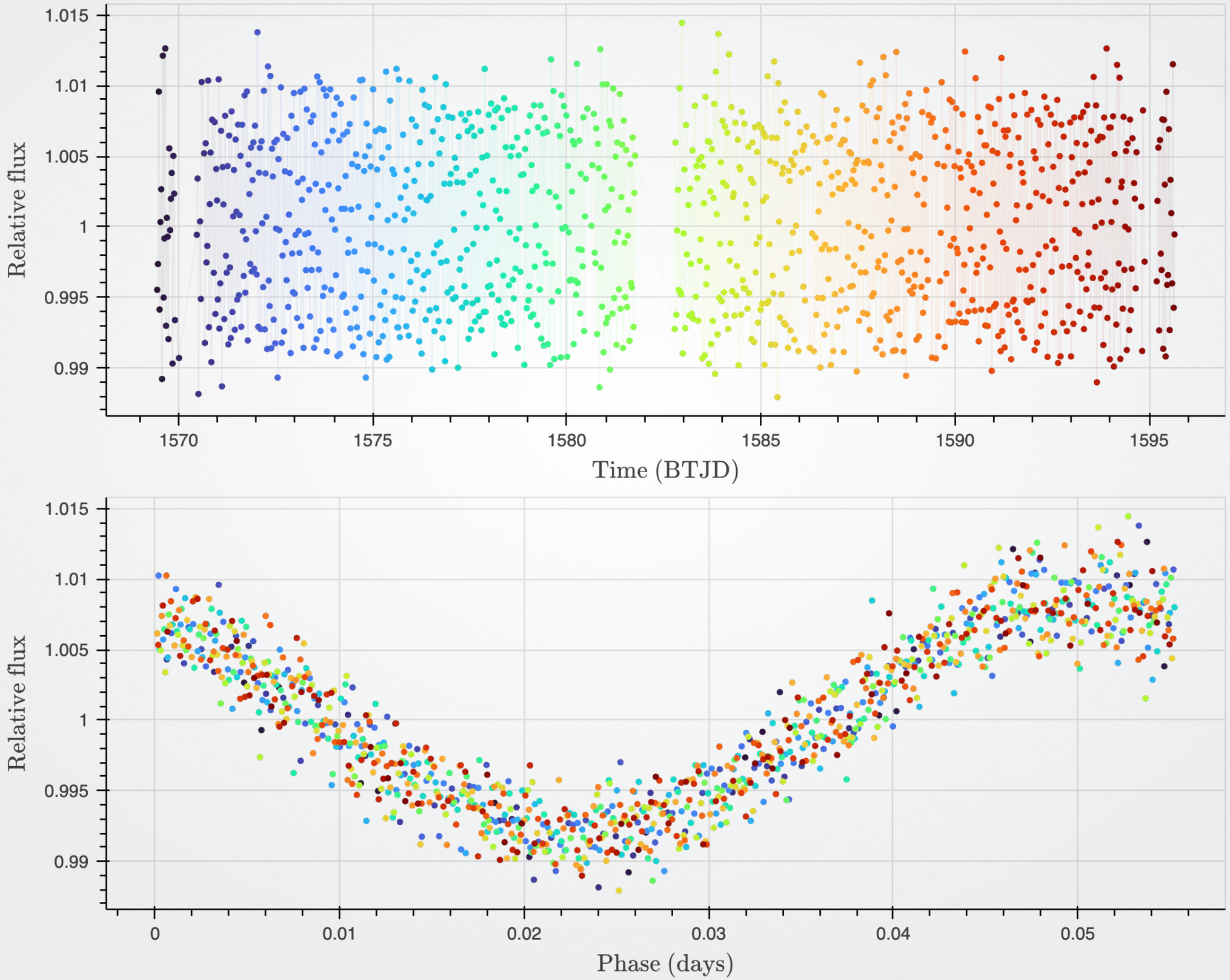
RESULTS

~180 new transiting planet candidates

7 out of 8 with follow-up have been confirmed

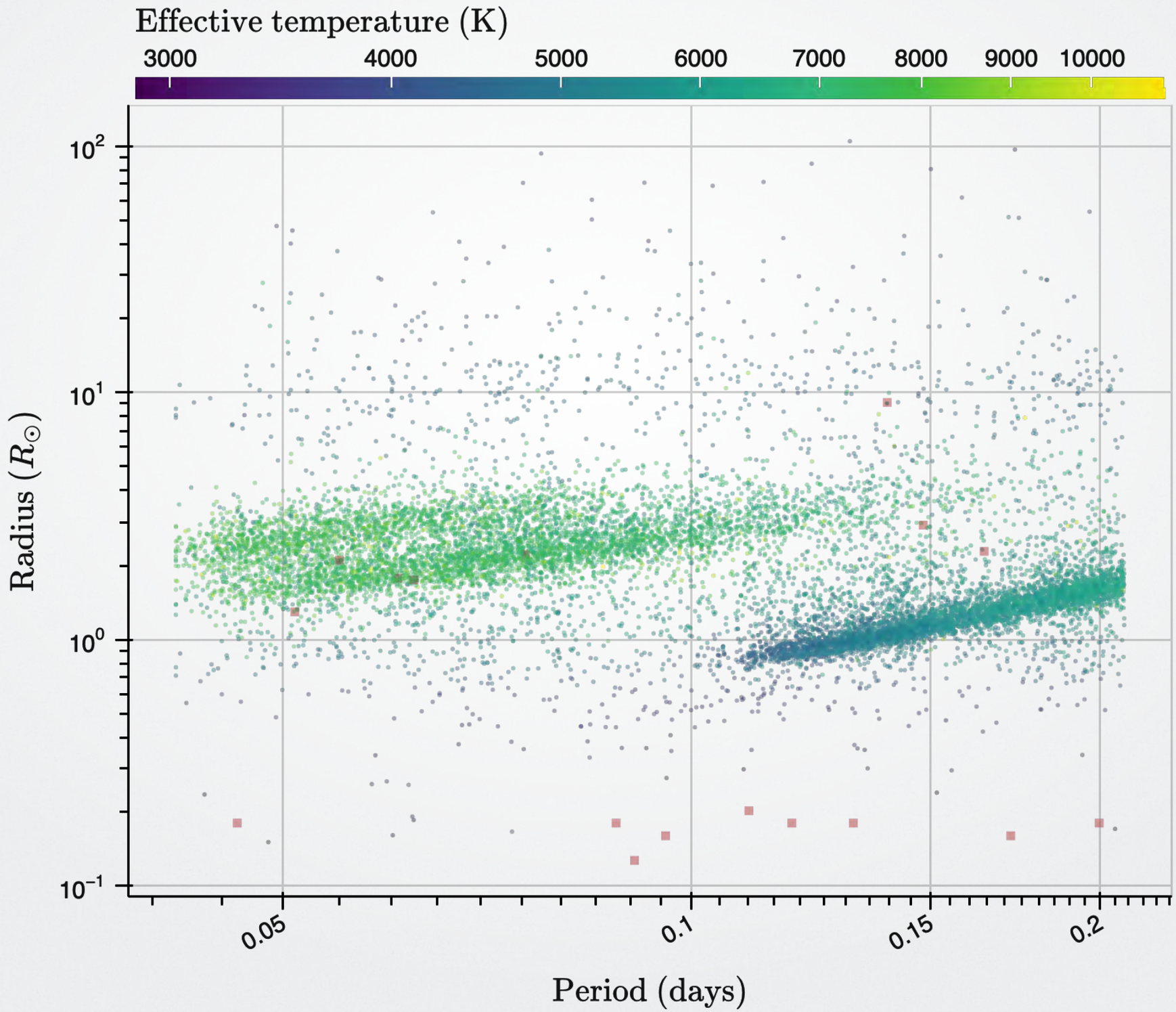


RESULTS


Relative flux
1.015
1.01
1.005
1
0.995
0.99
1570
1575
1580
1585
1590
1595
Time (BTJD)
Relative flux
1.015
1.01
1.005
1
0.995
0.99
0
0.01
0.02
0.03
0.04
0.05
Phase (days)

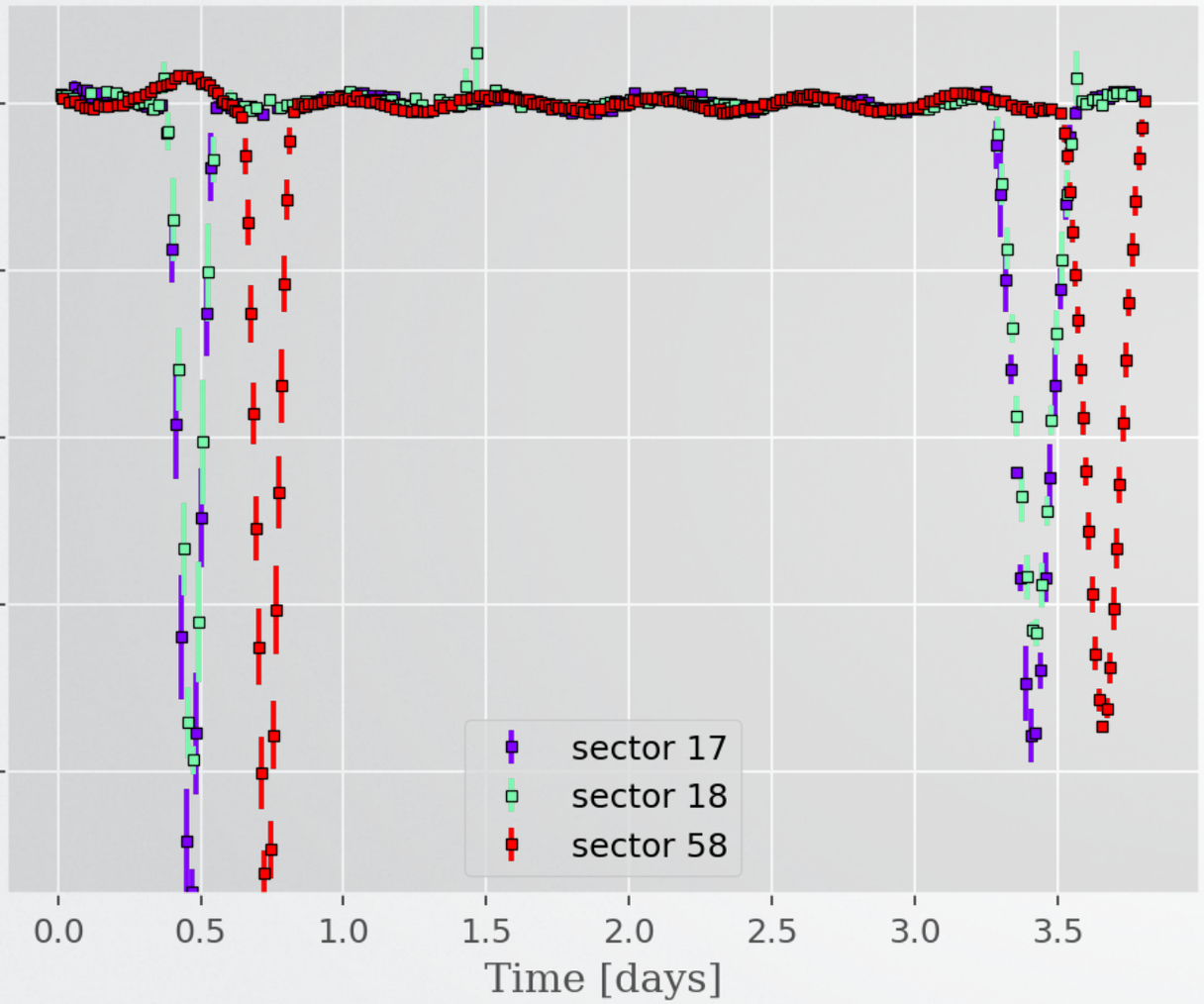
RESULTS

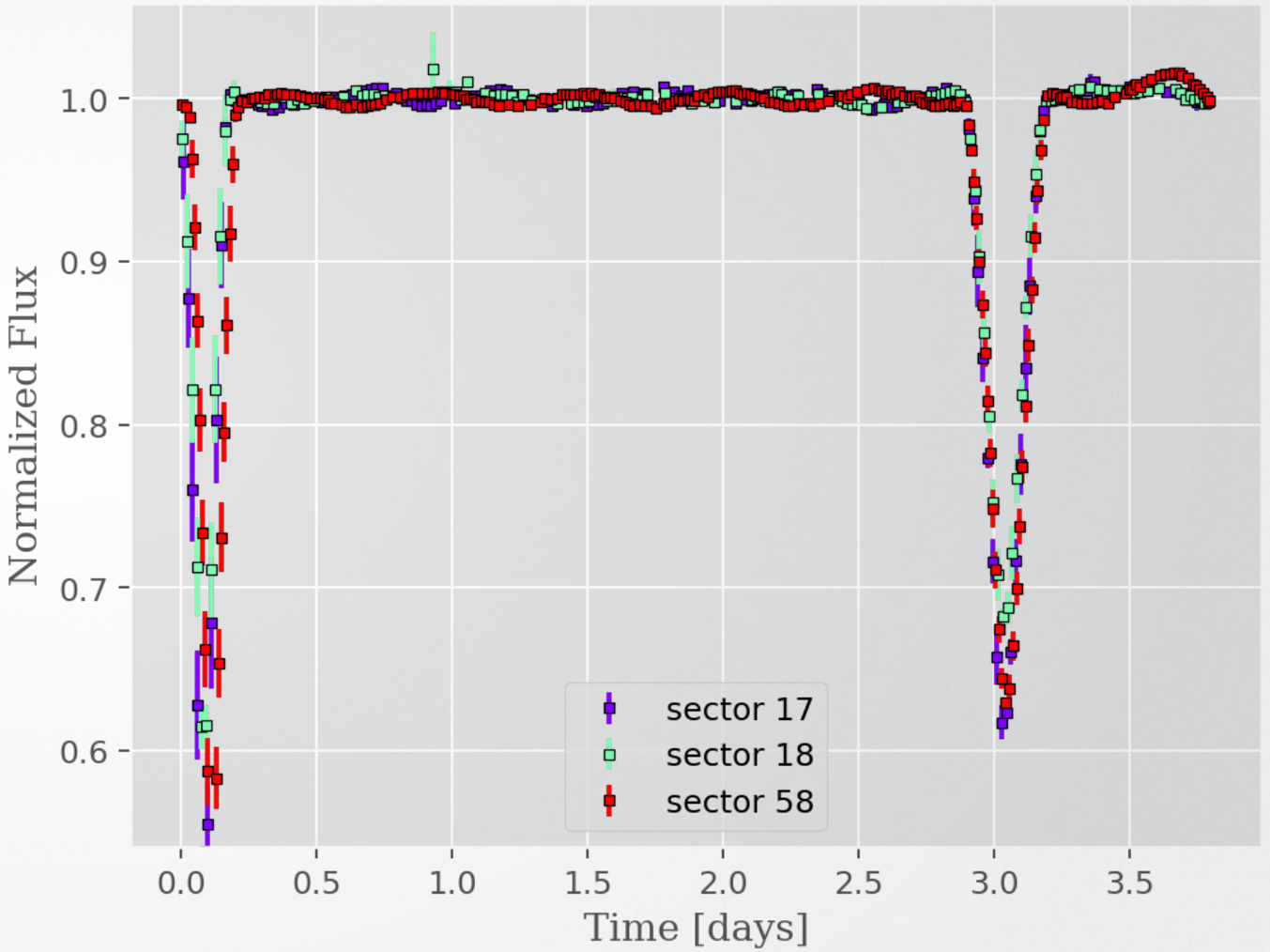
>14,000 short-period variables



RESULTS

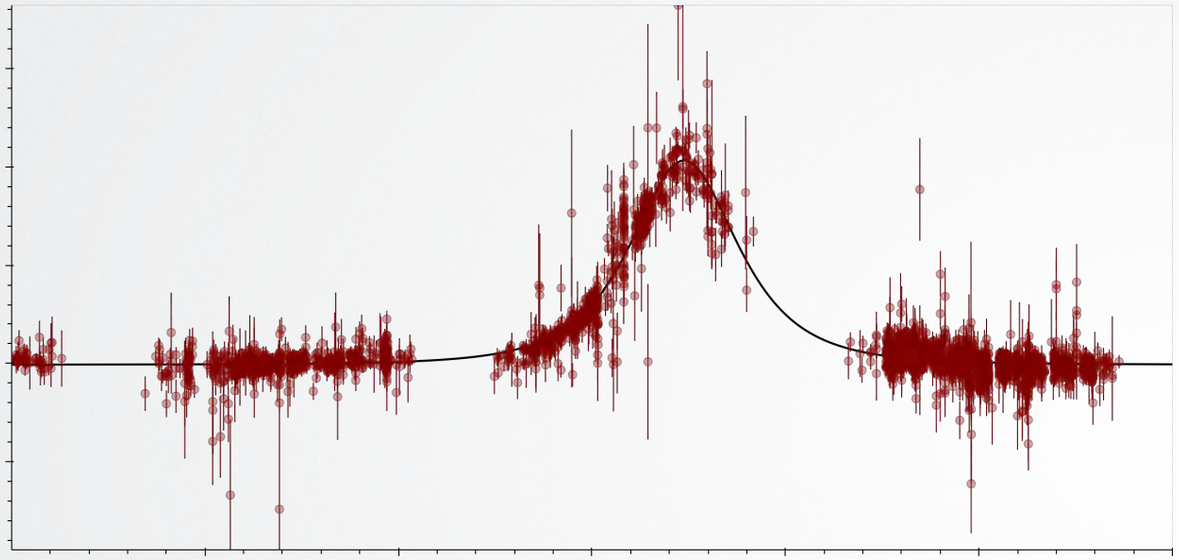
>300 heartbeat stars



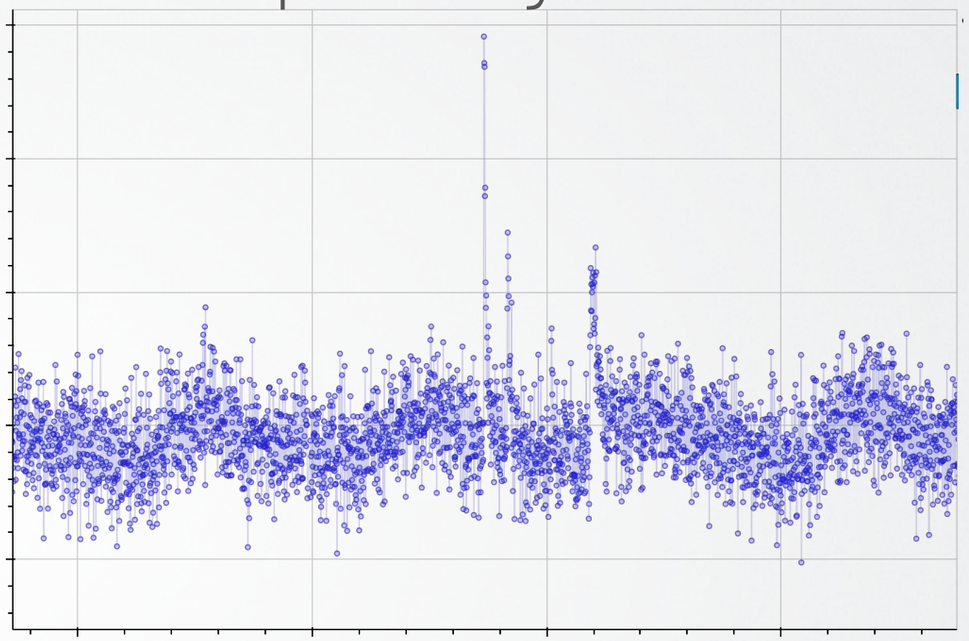


RESULTS

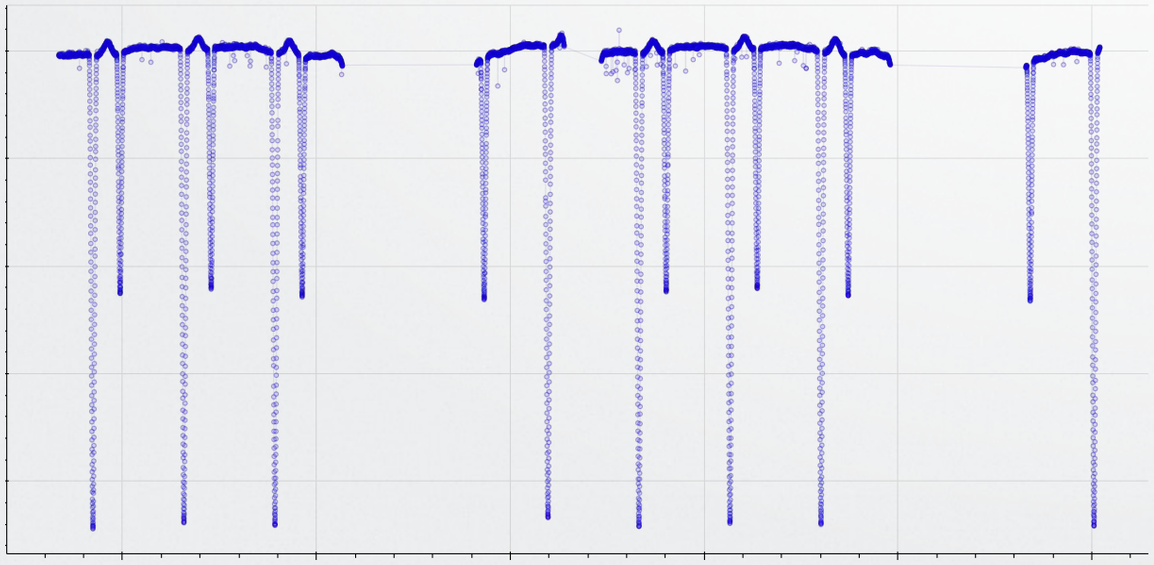
Microlensing



Flare frequency distribution

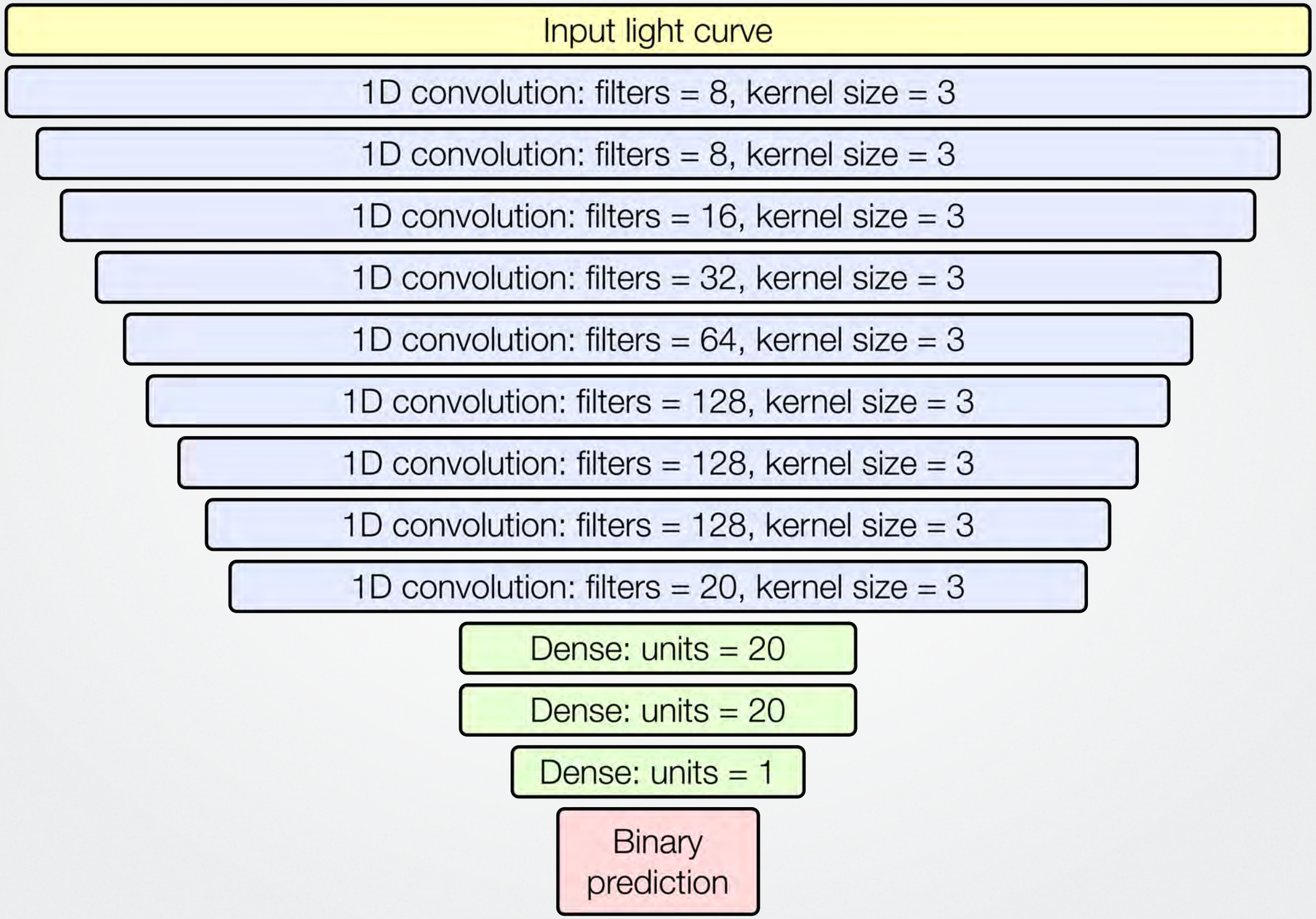


Eclipsing binaries



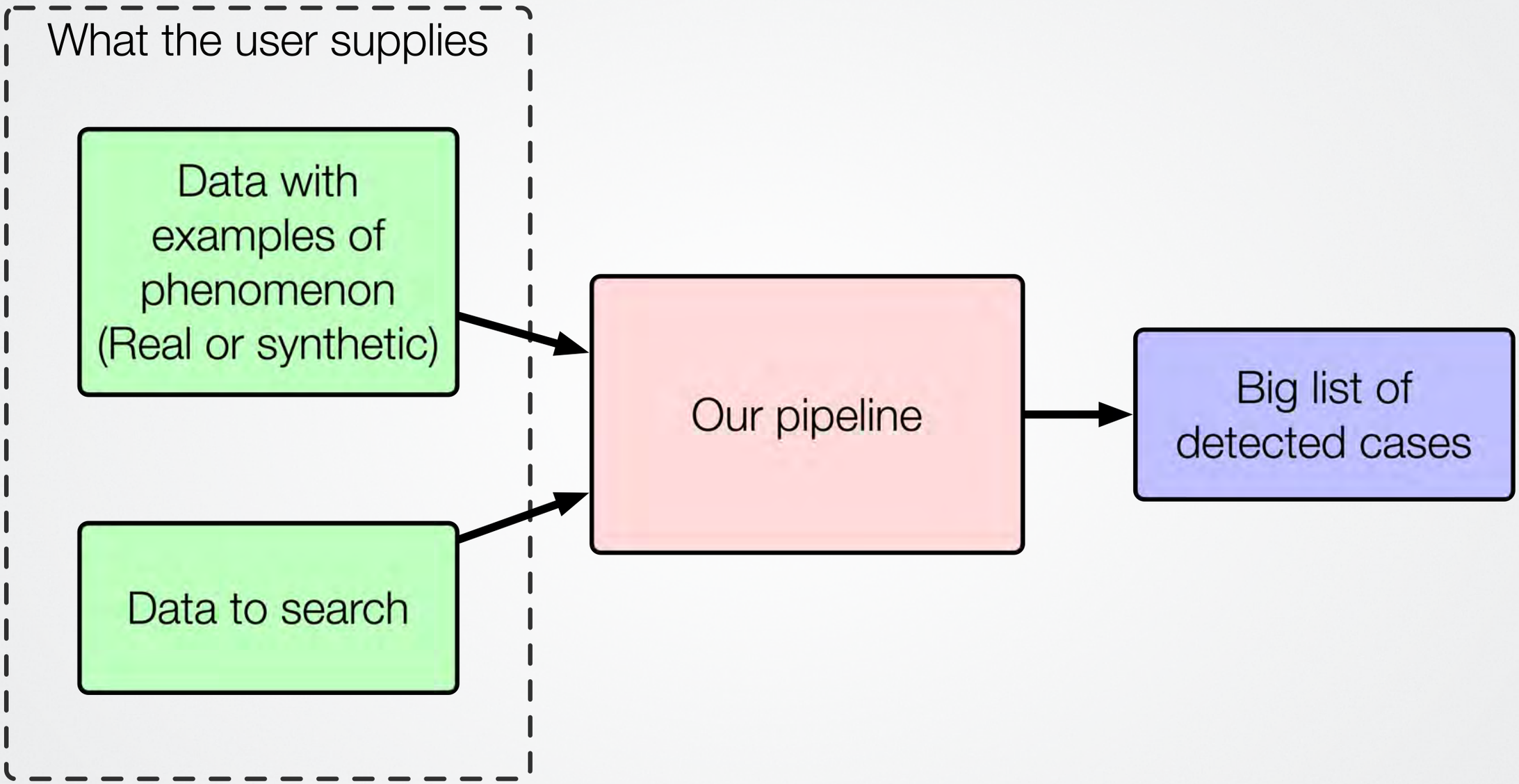
RR Lyrae

NEURAL NETWORK

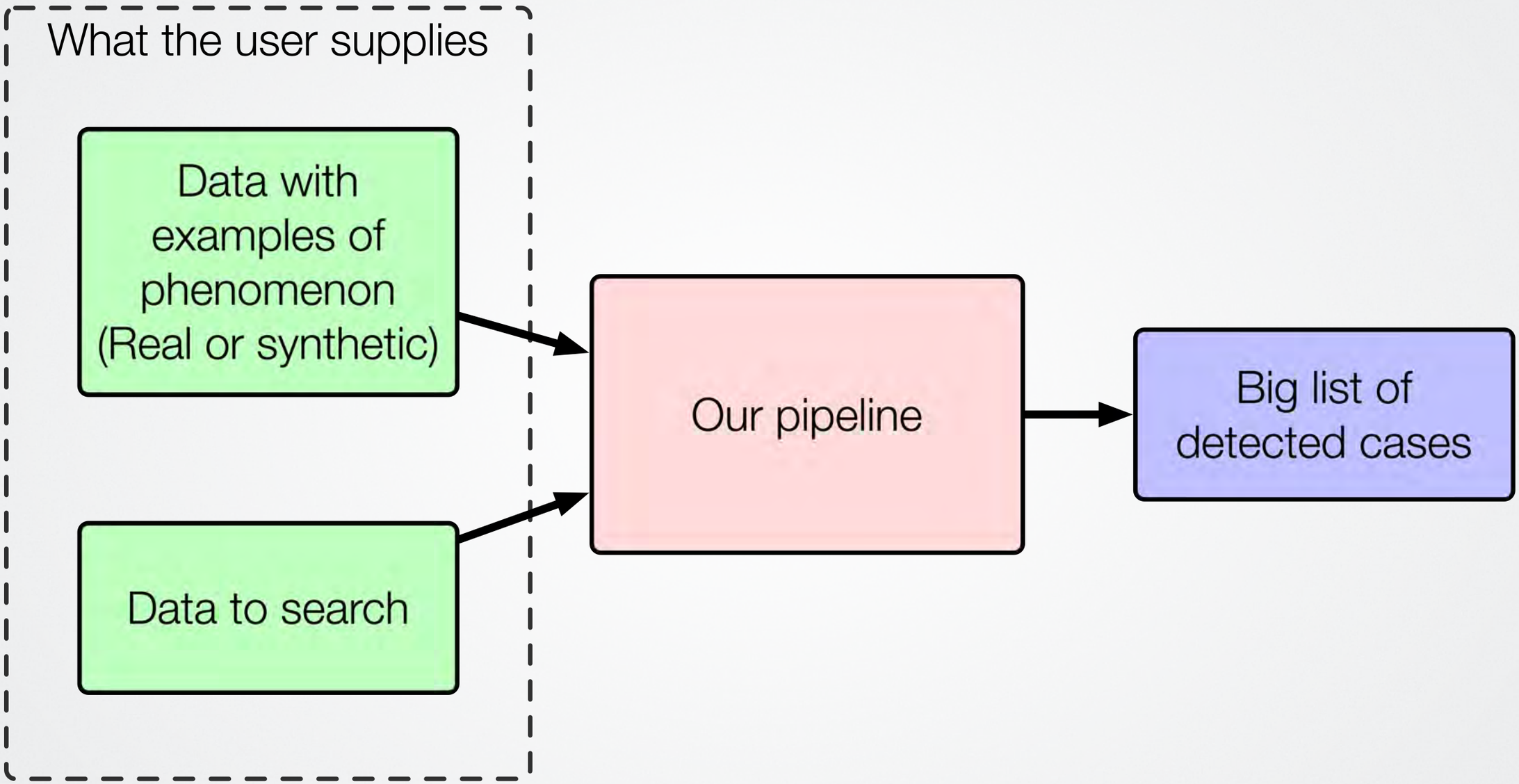


*simplified

A GOAL


What the user supplies
Data with examples of phenomenon (Real or synthetic)
Data to search
Our pipeline
Big list of detected cases

A GOAL



Have researchers spend less time staring at irrelevant data.

REASONS TO CONSIDER USING

- Speed
  - ~50ms to infer on a 10,000 data point light curve
- Automatically learn the task (mostly)
  - For a new data source
  - For sources of false positives and noise

FEATURES

- Data balancing
- Augmentation and preprocessing
  - With reasonable defaults
- Automatic network scaling
- Data injection
- Designed to be accessible to researchers without a ML background

BASIC CASE USER WORK

```
def get_paths():
    # Code that gets the list of paths here.
    return list_of_paths

def load_times_and_fluxes_from_path(path):
    # Code that loads the times and fluxes from a path
    # here.
    return times, fluxes

def load_label_from_path(path):
    # Code that loads the label from a path here.
    return label
```

CURRENT STATUS

- Recently re-written from TensorFlow to PyTorch
  - Some functionality not yet ported
- First stable release in preparation
- Binary classification tutorials available

Basic transit identification with prebuilt components

This tutorial will get you up and running with a neural network (NN) that can identify transiting exoplanets in data from the Transiting Exoplanet Survey Satellite (TESS). Many of the components used in this example will be prebuilt bits of code that we'll import from the package's example code. However, in later tutorials, we'll walkthrough how you would build each of these pieces yourself and how you would modify it for whatever your use case is.

ON THIS PAGE

Getting the example code
Downloading the dataset
Preparing for training
Train the network
Test the fitted model

Getting the example code

First, create a directory to hold the project named `qusi_example_project`, or some other suitable name. Then get the example scripts from the `qusi` repository. You can download just that directory by clicking here. Move this `examples` directory into your project directory so that you have `qusi_example_project/examples`. The remainder of the commands will assume you are running code from the project directory, unless otherwise stated.

Downloading the dataset

The next thing we'll do is download a dataset of light curves that include cases both with and without transiting planets. To do this, run the example script at `examples/download_spoc_transit_light_curves`. For now, don't worry about how each part of the code works. You can run the script with:

```
python examples/download_spoc_transit_light_curves.py
```

The main thing to know is that this will create a `data` directory within the project directory and within that will be a `spoc_transit_experiment` directory, referring to the data for the experiment of finding transiting planets within the TESS SPOC data. This will further contain 3 directories. One for train data, one for validation data, and one for test data. Within each of those, it will create a `positive` directory

NEAR-TERM GOALS

- First stable release
- Large-scale distributed training
- Wider selection of public architectures
- Tutorials for multi-class and regression targets

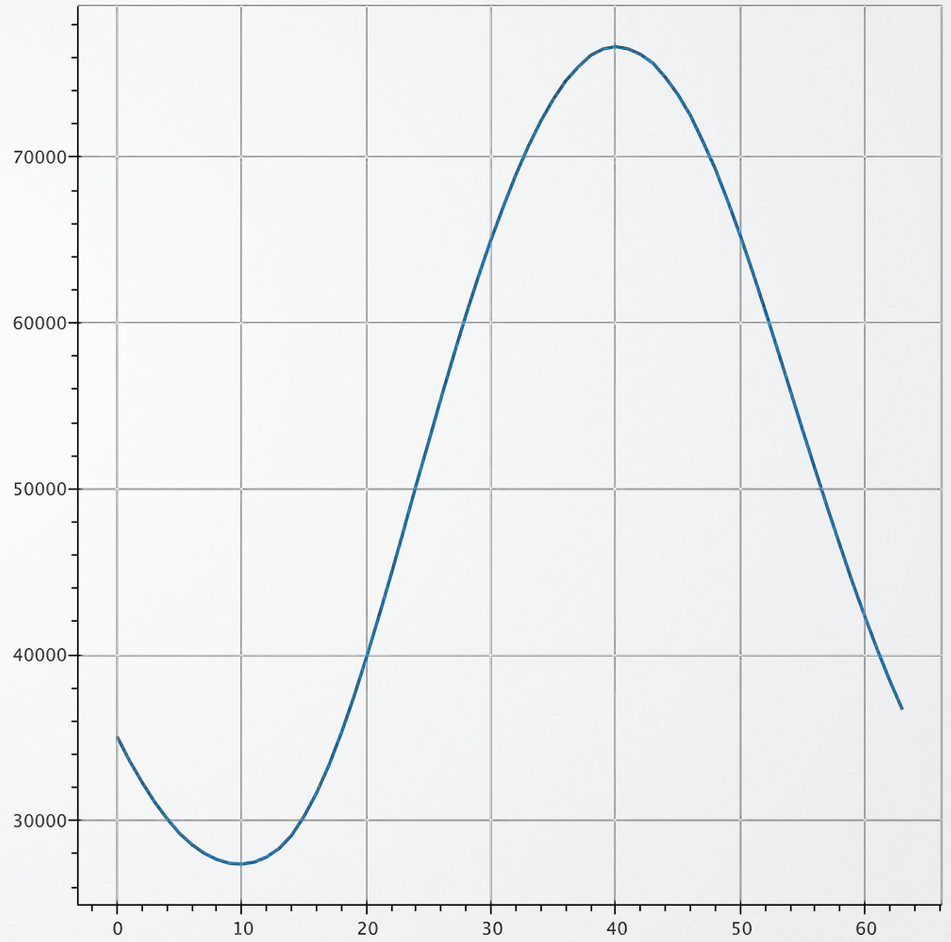
LONGER-TERM GOALS

- Time-series image inputs
- Light curve generative models

GENERATING LIGHT CURVES

0.2900
0.0038
0.1690
2.9928
3.4996
-0.2018
-0.1605
-0.0980
1.1392
5.2350
3.7653

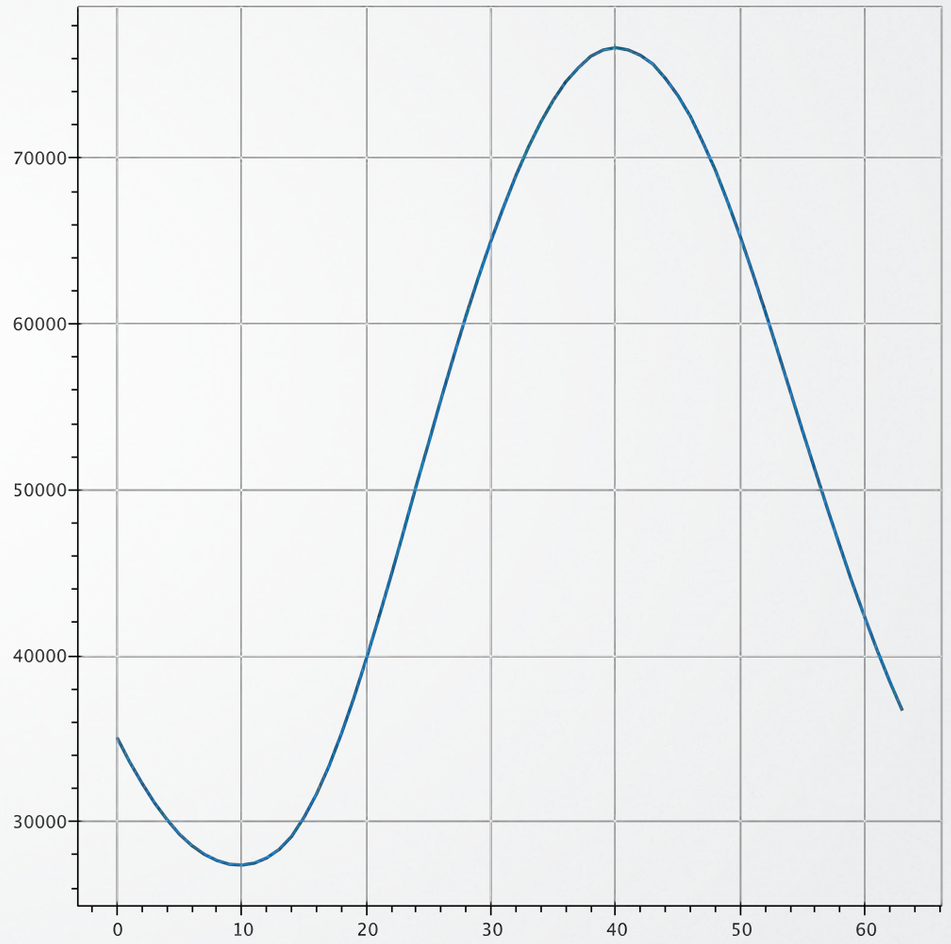
Existing algorithm



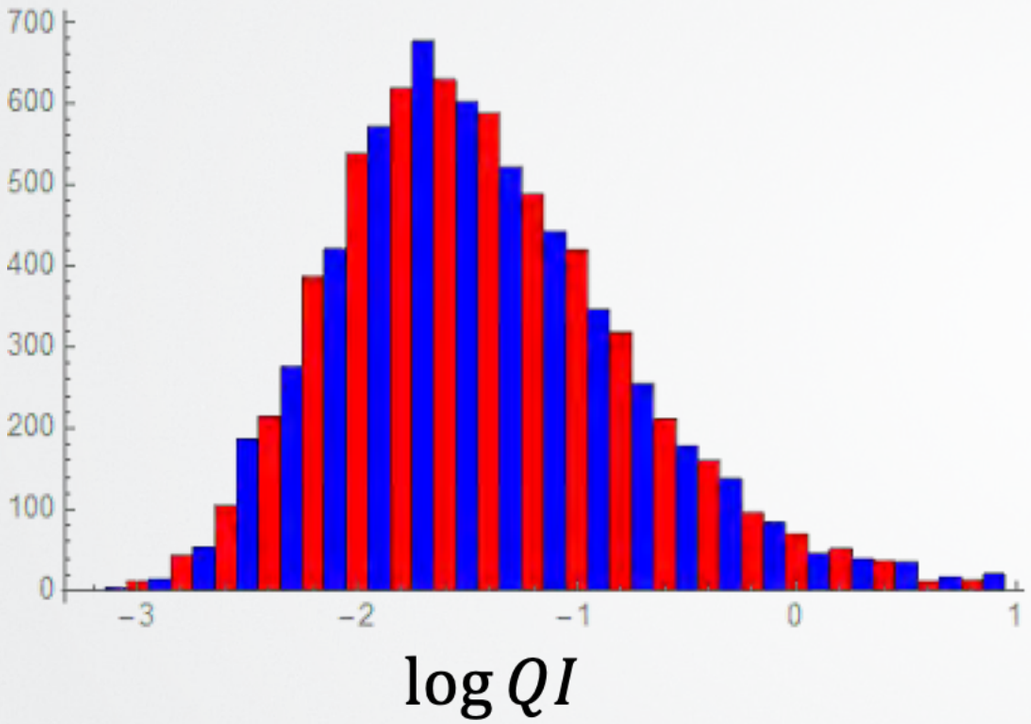
GENERATING LIGHT CURVES

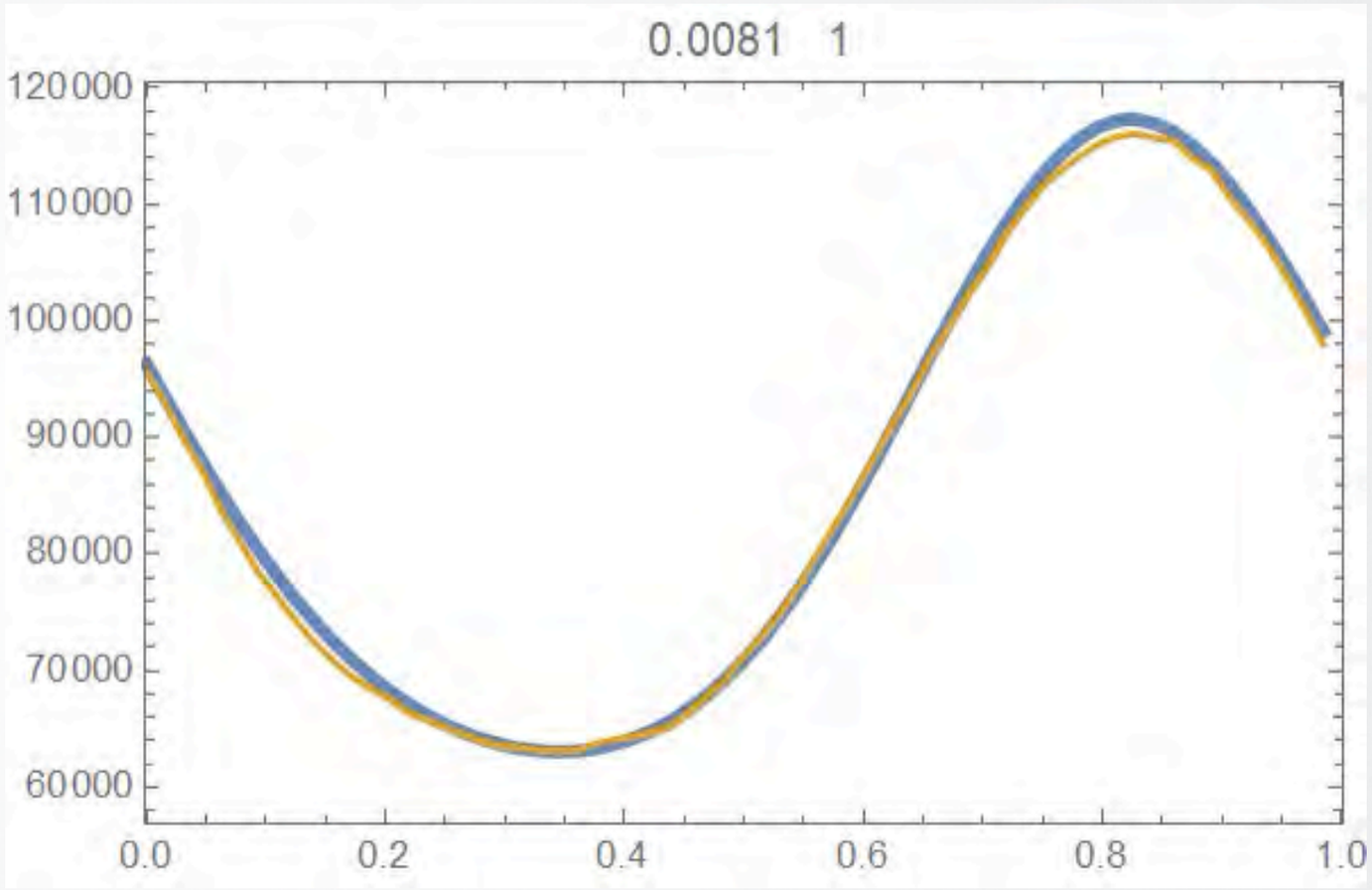
0.2900
0.0038
0.1690
2.9928
3.4996
-0.2018
-0.1605
-0.0980
1.1392
5.2350
3.7653

Neural network



GENERATING LIGHT CURVES



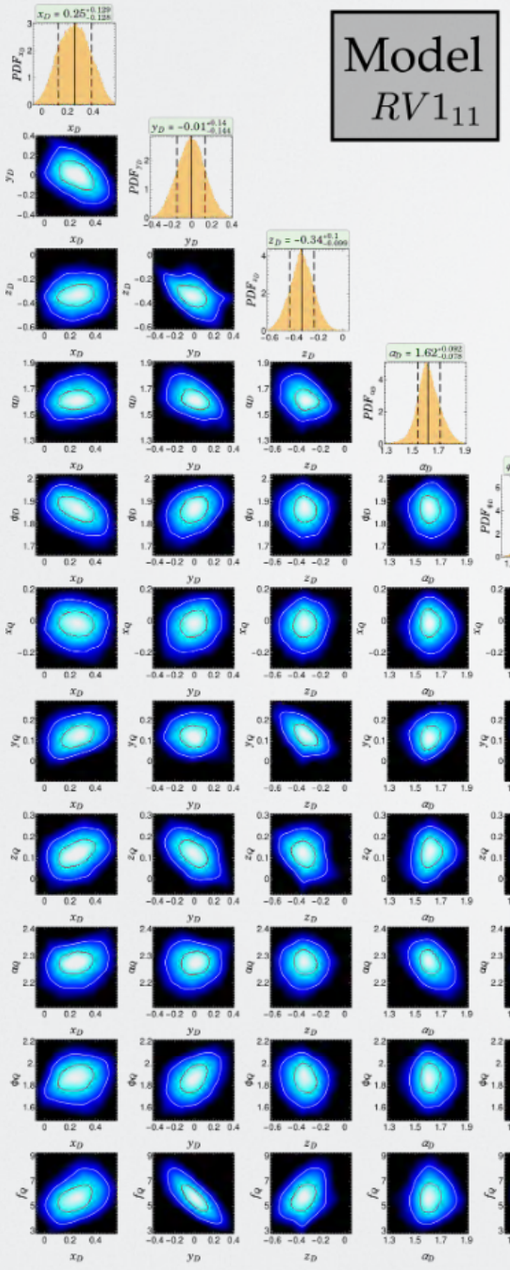


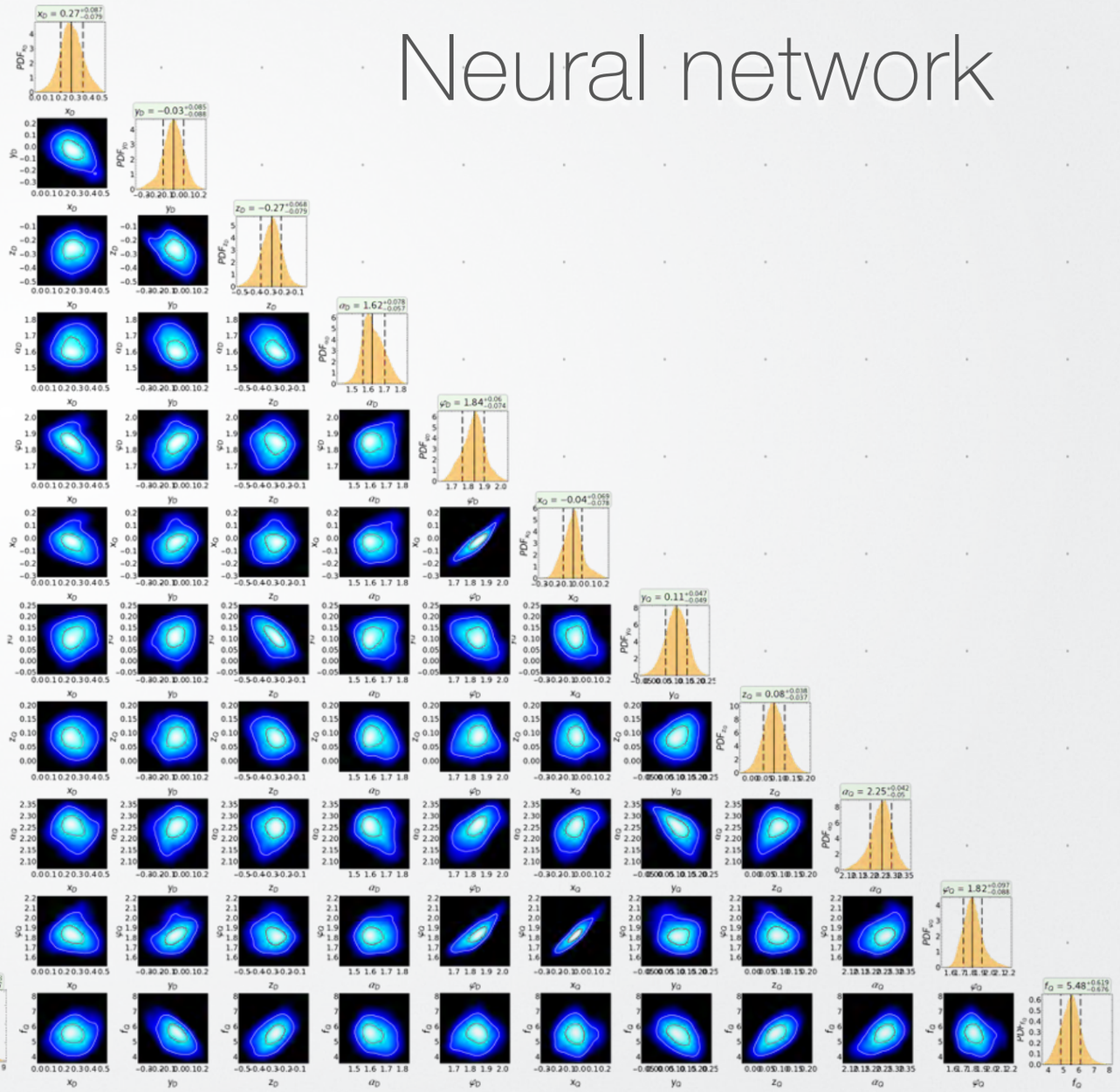
$$QI = \frac{1}{Median_{SA}^2} \sum_{bins} \left(Y_{SA_i} - Y_{ML_i}\right)^2$$

Semi-analytic

Neural network

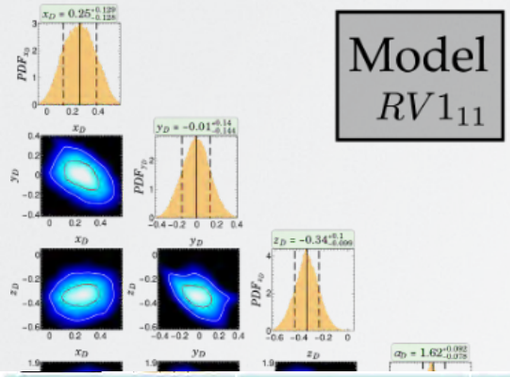
GENERATING LIGHT CURVES



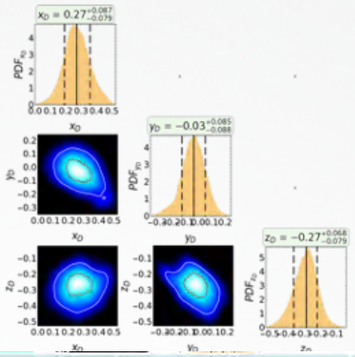


Neural network

GENERATING LIGHT CURVES

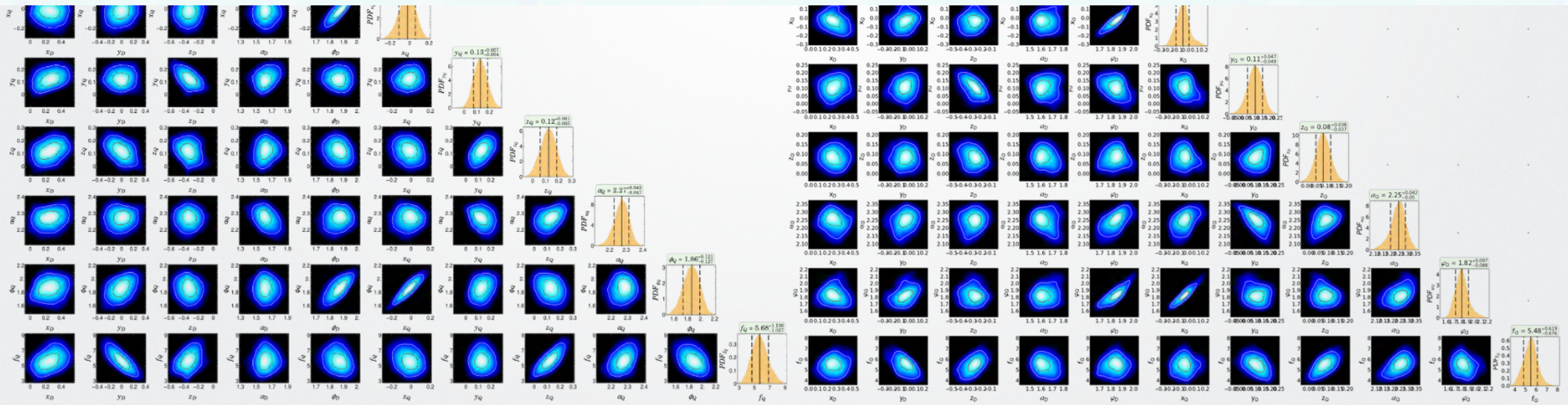


Model
$RV1_{11}$

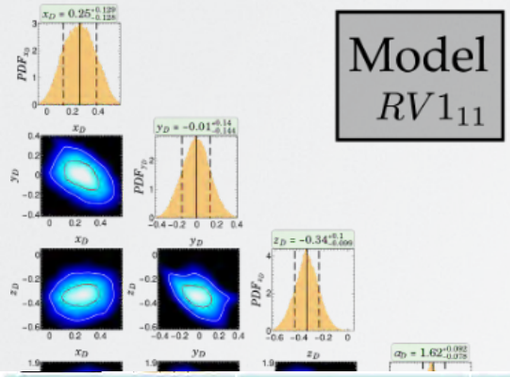


Neural network

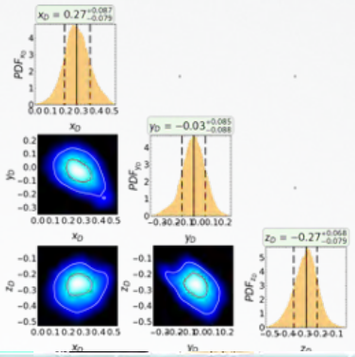
x_D	y_D	z_D	a_D	φ_D	x_Q	y_Q	z_Q	a_Q	φ_Q	Q/D
$0.25^{+0.129}_{-0.128}$	$-0.01^{+0.14}_{-0.144}$	$-0.34^{+0.1}_{-0.099}$	$1.62^{+0.092}_{-0.078}$	$1.86^{+0.053}_{-0.059}$	$-0.02^{+0.082}_{-0.088}$	$0.13^{+0.057}_{-0.054}$	$0.12^{+0.061}_{-0.065}$	$2.27^{+0.042}_{-0.047}$	$1.86^{+0.121}_{-0.127}$	$5.68^{+1.36}_{-1.027}$
$0.27^{+0.087}_{-0.079}$	$-0.03^{+0.085}_{-0.088}$	$-0.27^{+0.068}_{-0.079}$	$1.62^{+0.078}_{-0.057}$	$1.84^{+0.06}_{-0.074}$	$-0.04^{+0.069}_{-0.078}$	$0.11^{+0.047}_{-0.049}$	$0.08^{+0.038}_{-0.037}$	$2.25^{+0.042}_{-0.05}$	$1.82^{+0.097}_{-0.088}$	$5.48^{+0.619}_{-0.676}$



GENERATING LIGHT CURVES

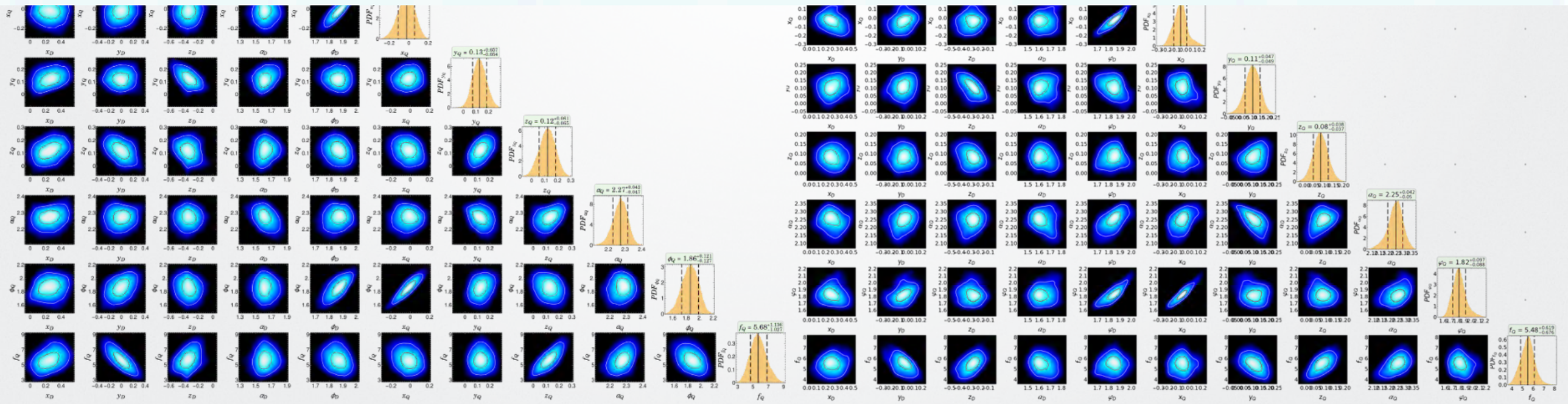


Model
$RV1_{11}$



Neural network
>300x faster

x_D	y_D	z_D	a_D	φ_D	x_Q	y_Q	z_Q	a_Q	φ_Q	Q/D
$0.25^{+0.129}_{-0.128}$	$-0.01^{+0.14}_{-0.144}$	$-0.34^{+0.1}_{-0.099}$	$1.62^{+0.092}_{-0.078}$	$1.86^{+0.053}_{-0.059}$	$-0.02^{+0.082}_{-0.088}$	$0.13^{+0.057}_{-0.054}$	$0.12^{+0.061}_{-0.065}$	$2.27^{+0.042}_{-0.047}$	$1.86^{+0.121}_{-0.127}$	$5.68^{+1.36}_{-1.027}$
$0.27^{+0.087}_{-0.079}$	$-0.03^{+0.085}_{-0.088}$	$-0.27^{+0.068}_{-0.079}$	$1.62^{+0.078}_{-0.057}$	$1.84^{+0.06}_{-0.074}$	$-0.04^{+0.069}_{-0.078}$	$0.11^{+0.047}_{-0.049}$	$0.08^{+0.038}_{-0.037}$	$2.25^{+0.042}_{-0.05}$	$1.82^{+0.097}_{-0.088}$	$5.48^{+0.619}_{-0.676}$



LONGER-TERM GOALS

- Time-series image functionality
- Light curve generative models

FOR SOFTWARE DEVELOPERS

- Unit and integration tests
  - 181 tests currently
- Semantic versioning
  - Starting at first stable release
- Typehinted

FOR MACHINE LEARNING SPECIALISTS

- Modular PyTorch scaffolding
- Can be treated as a library of components

github.com/golmschenk/qusi

Questions?