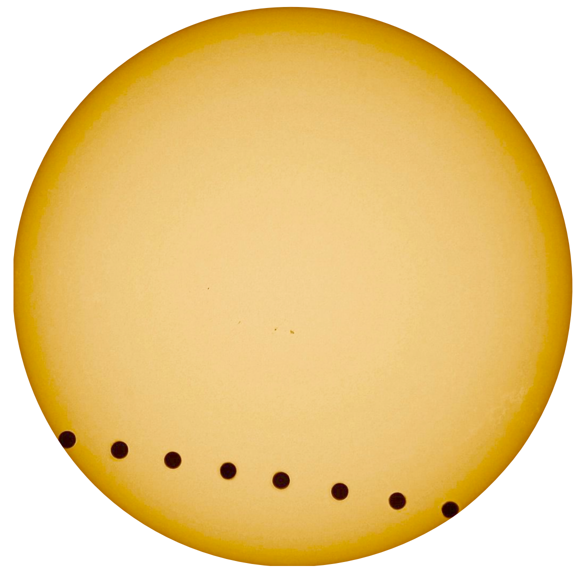


EVEREST

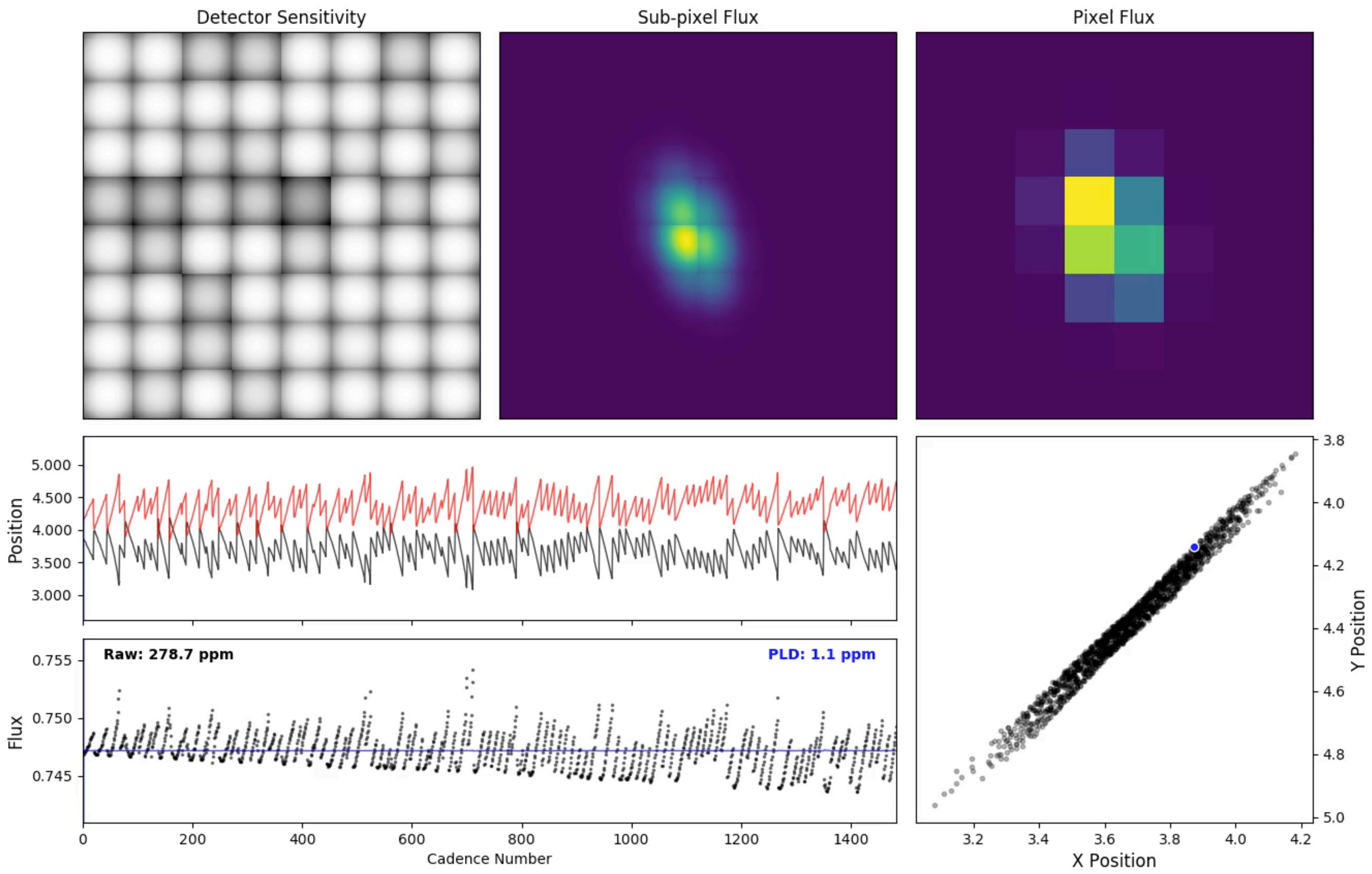
EPIC VARIABILITY EXTRACTION AND REMOVAL FOR EXOPLANET SCIENCE TARGETS

RODRIGO LUGER

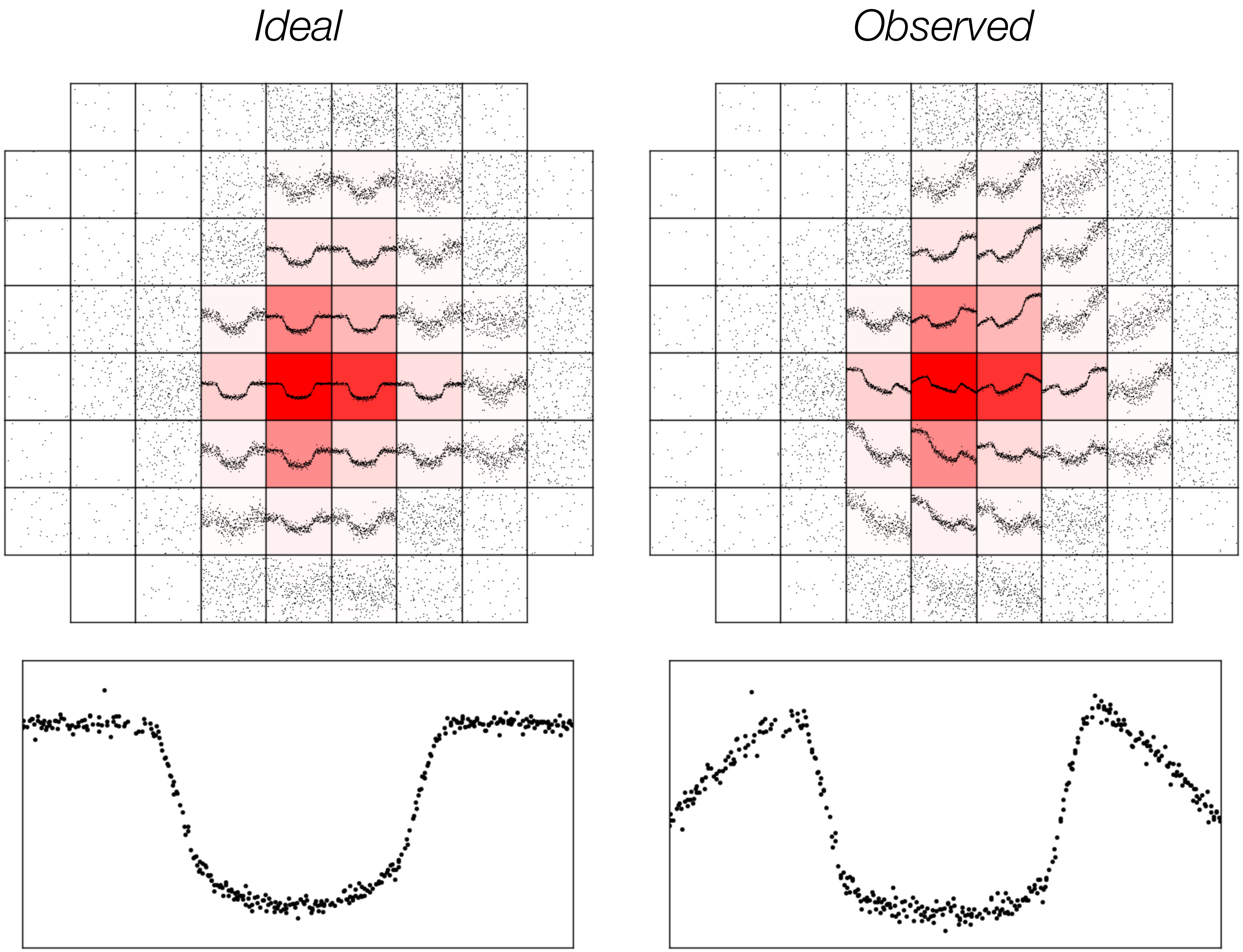
with ETHAN KRUSE, DANIEL FOREMAN-MACKEY, ERIC AGOL, AND NICHOLAS SAUNDERS

PART I

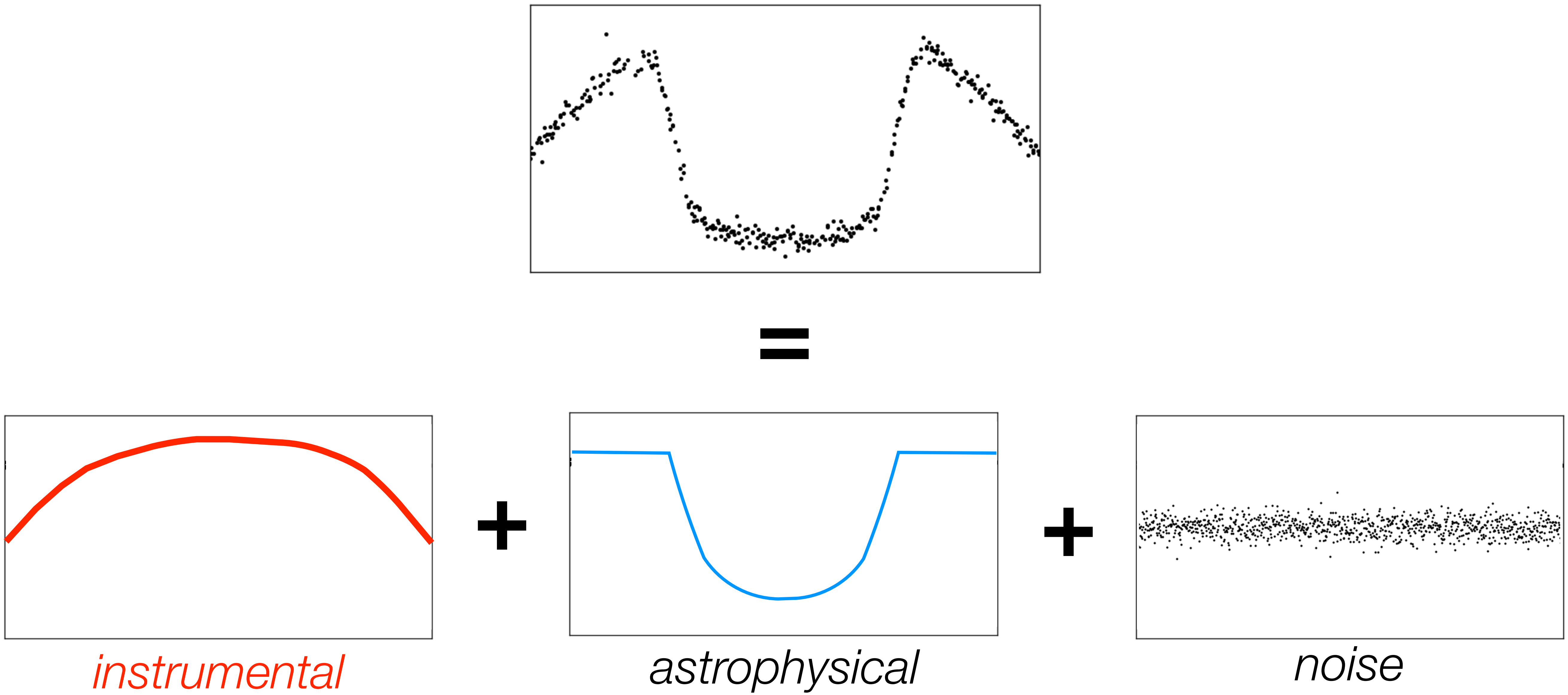
INTRODUCTION


Detector Sensitivity
Sub-pixel Flux
Pixel Flux
Position
5.000
4.500
4.000
3.500
3.000
Flux
0.755
0.750
0.745
Raw: 278.7 ppm
PLD: 1.1 ppm
0
200
400
600
800
1000
1200
1400
Cadence Number
3.2
3.4
3.6
3.8
4.0
4.2
X Position
3.8
4.0
4.2
4.4
4.6
4.8
5.0
Y Position

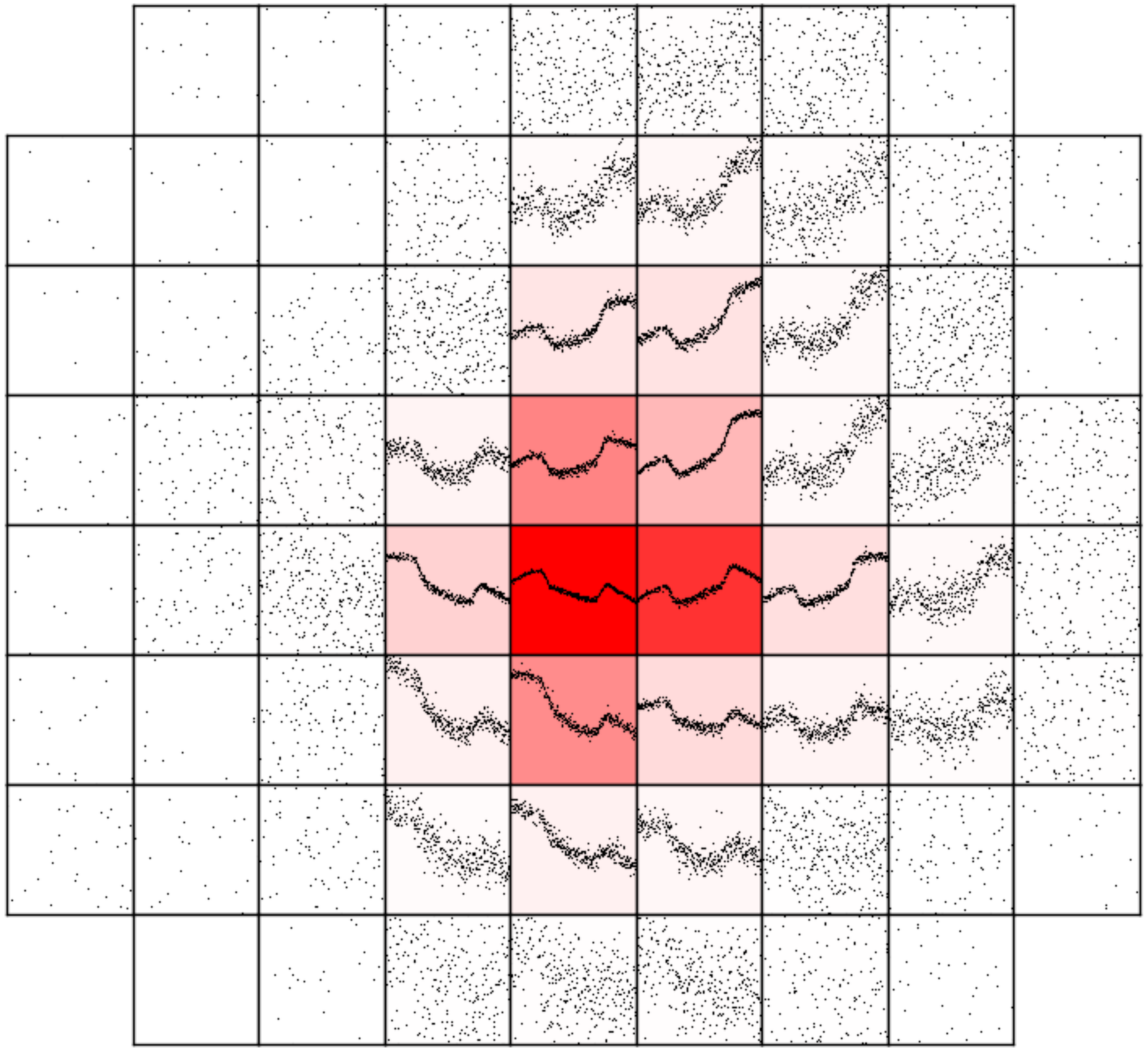
A typical postage stamp in the real world



Back to the simple example…



Pixel Level Decorrelation

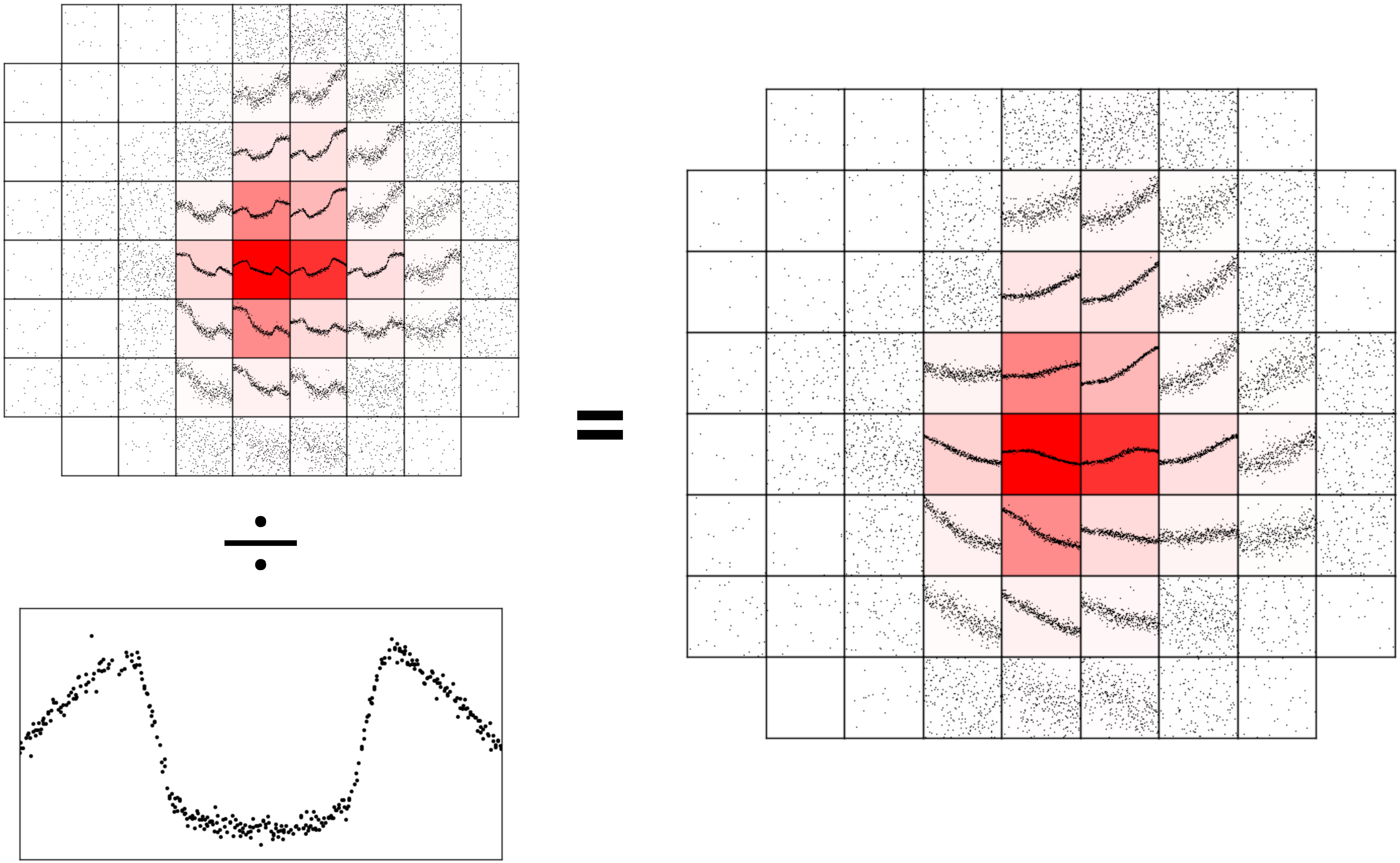


Astrophysical signals are the same in all pixels

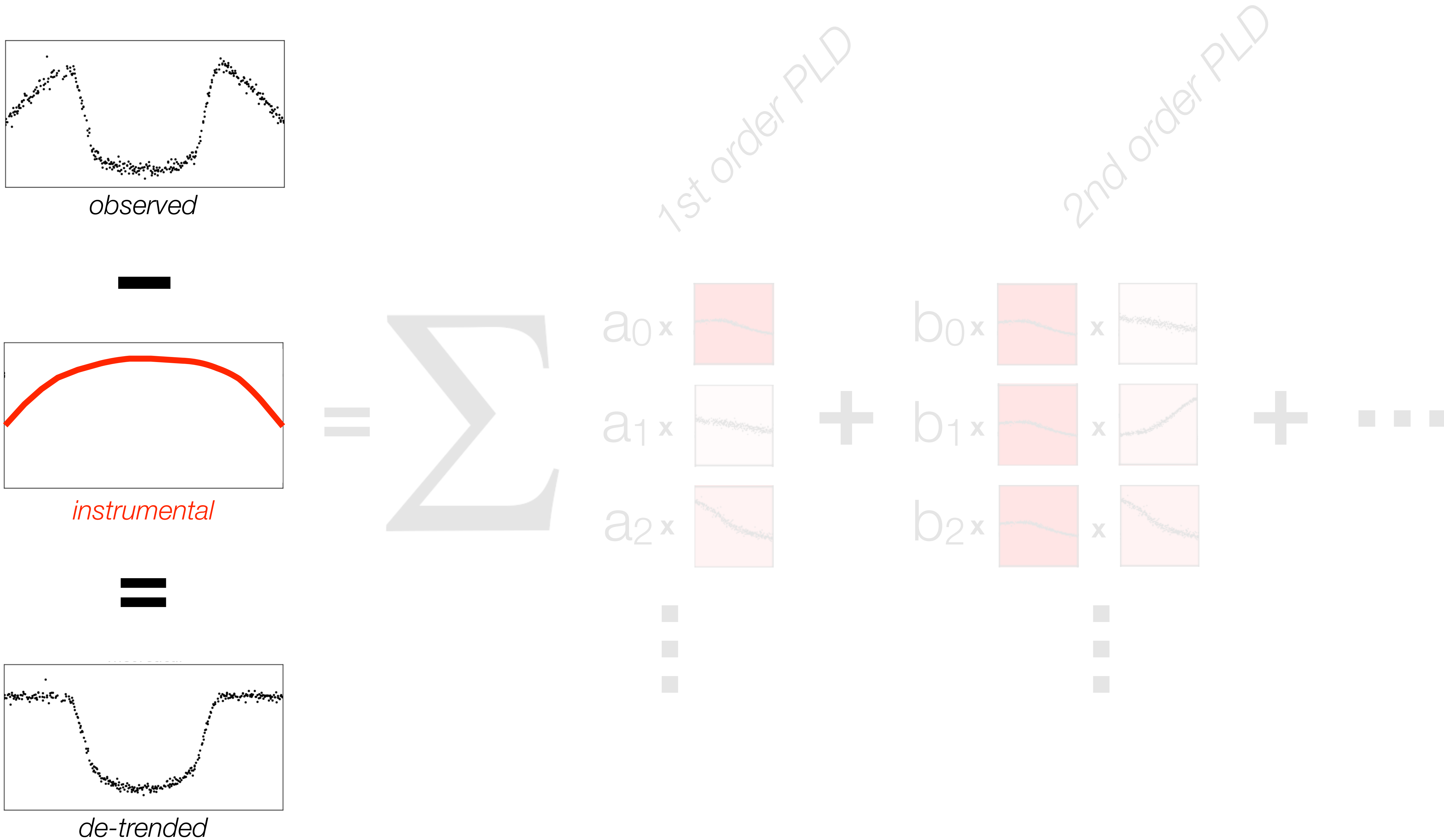
Instrumental signals are different

Deming et al. (2015)

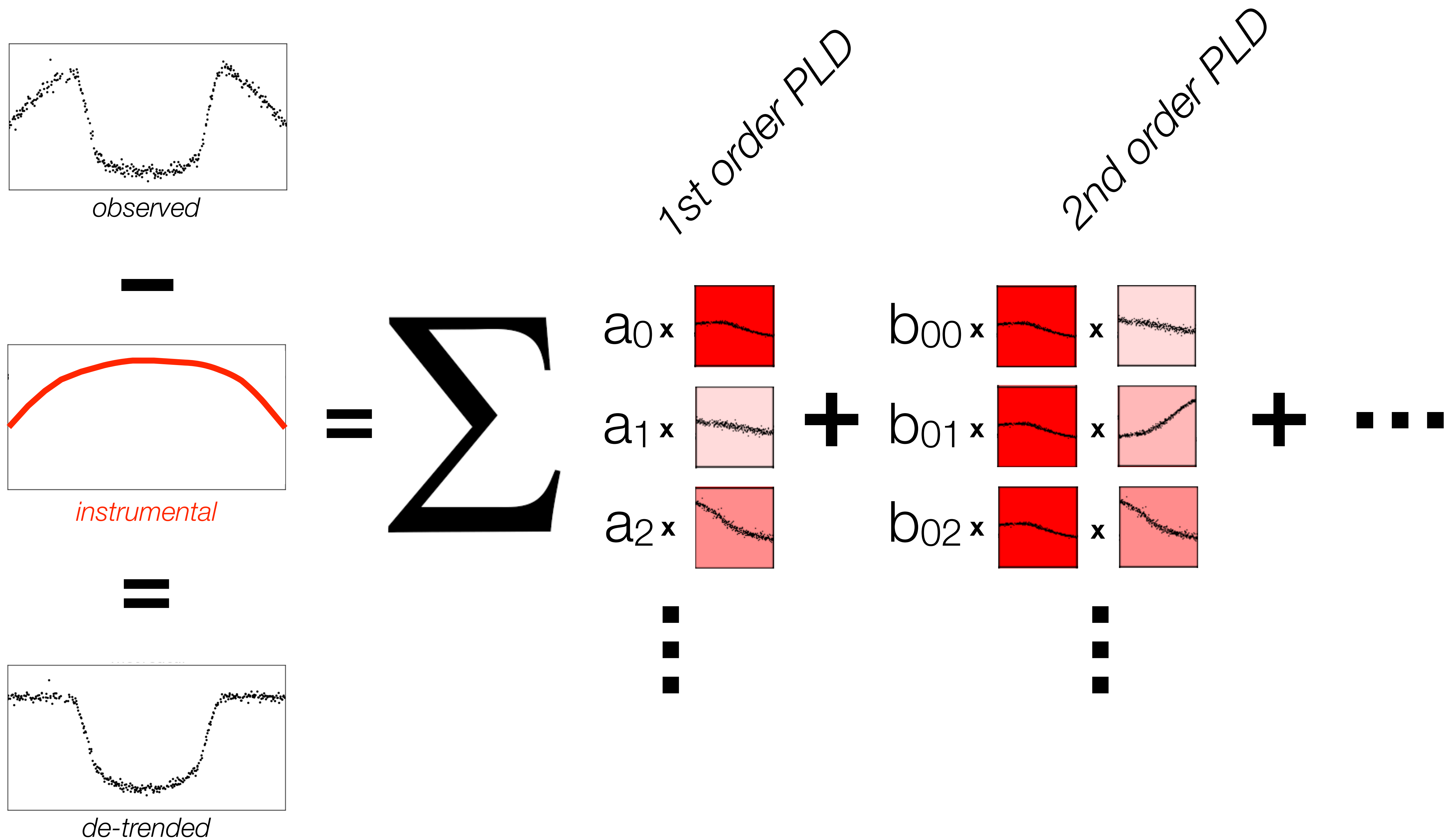
Pixel Level Decorrelation



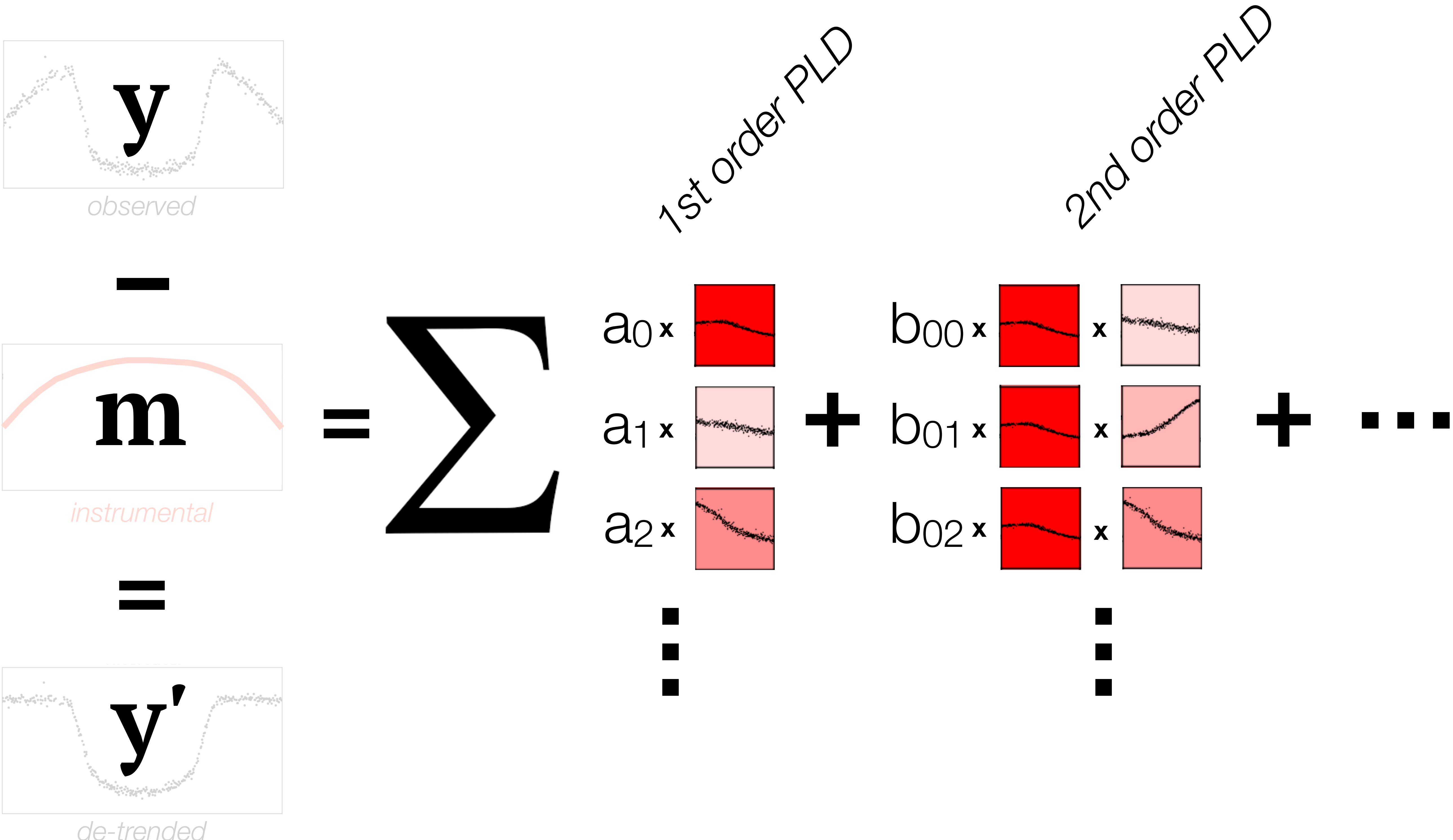
Pixel Level Decorrelation



Pixel Level Decorrelation



Pixel Level Decorrelation



Pixel Level Decorrelation

$$\mathbf{y}$$

observed

$$-$$

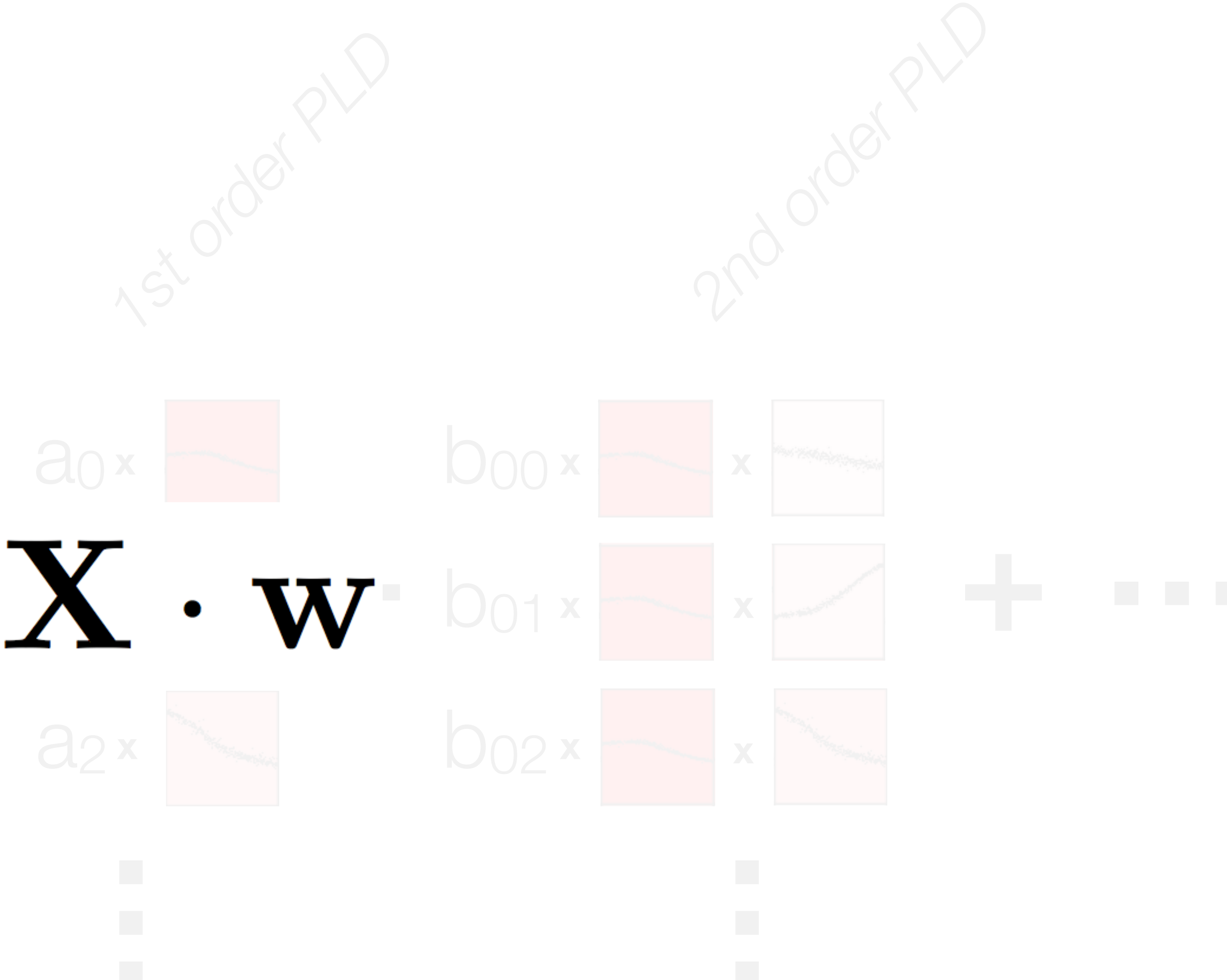
$$\mathbf{m} = \sum \mathbf{X} \cdot \mathbf{w}$$

instrumental

$$=$$

$$\mathbf{y}'$$

de-trended



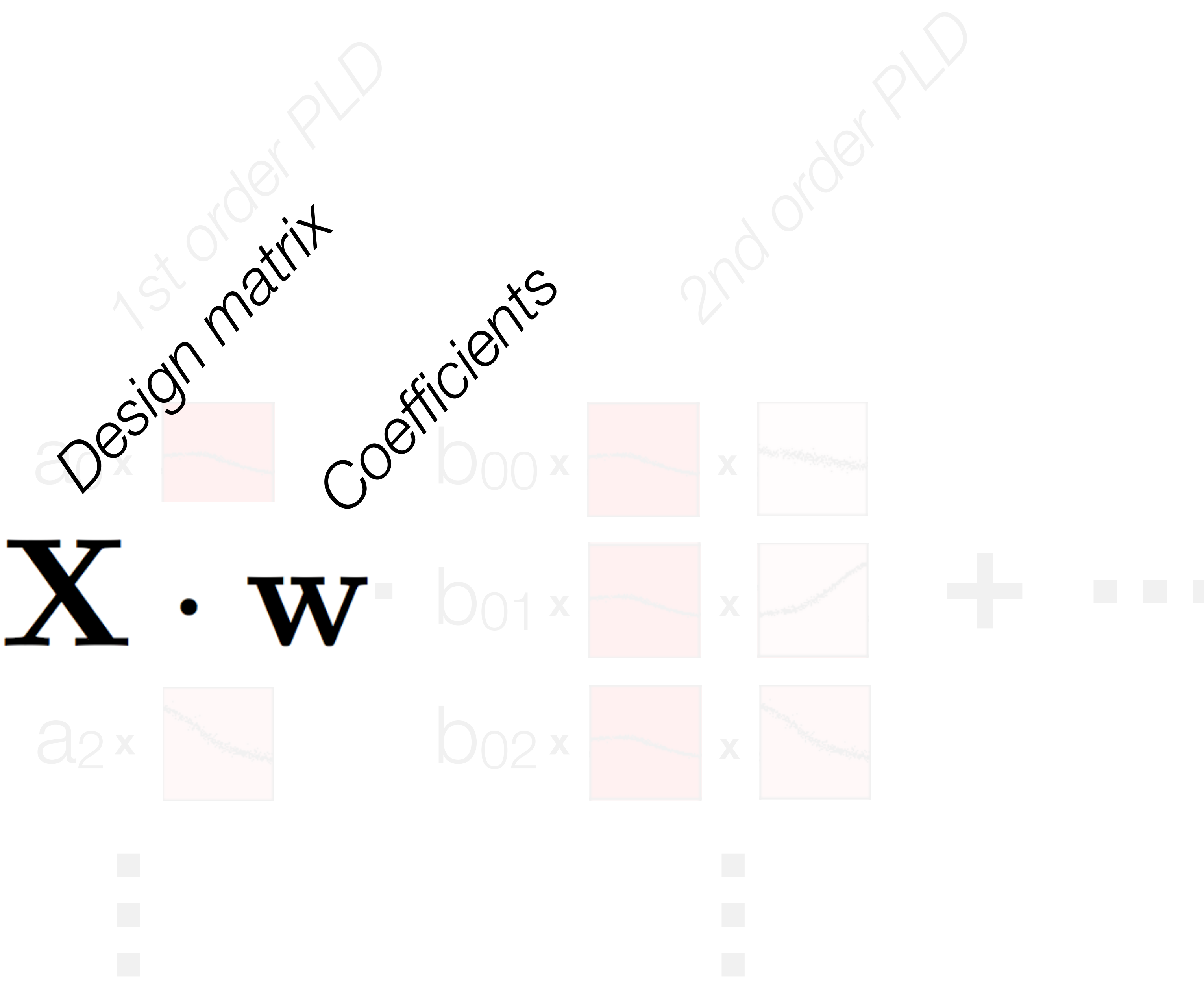
Pixel Level Decorrelation

$-$

$$\mathbf{m} = \sum \mathbf{X} \cdot \mathbf{w}$$

Design matrix

Coefficients



$=$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}^{???}$$

Pixel Level Decorrelation

???

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Arbitrary covariance

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \mathbf{X}\right)^{-1} \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \cdot \mathbf{y}\right)$$

Pixel Level Decorrelation

???

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Arbitrary covariance

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \mathbf{X}\right)^{-1} \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \cdot \mathbf{y}\right)$$

Homoscedastic, diagonal covariance

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{X}\right)^{-1} \left(\mathbf{X}^\top \cdot \mathbf{y}\right)$$

Pixel Level Decorrelation

Let's do this

```
pip install everest-pipeline
```

PART II

USING EVEREST

Installation

Using `everest` is easy once you get it set up.

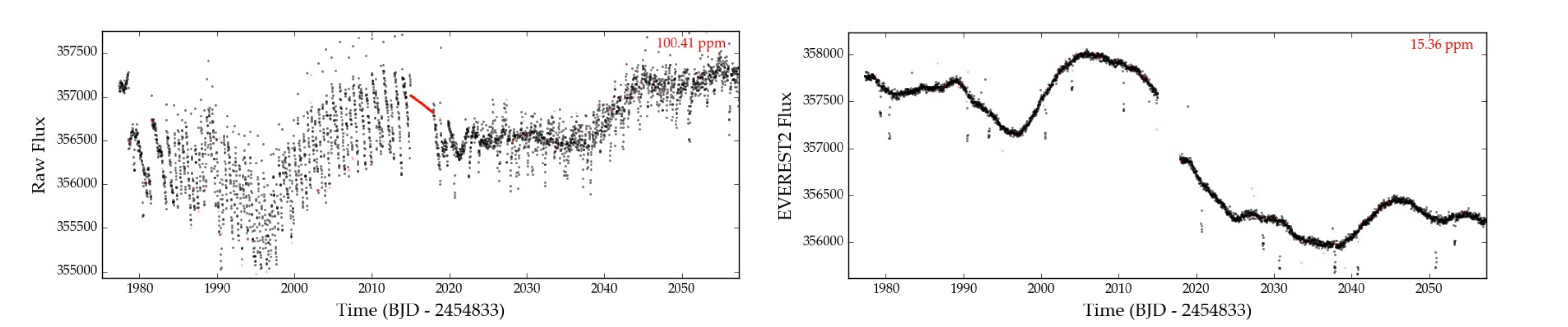
```
pip install everest-pipeline
```

Quick Access

For quick access to the light curves in the catalog, you can use the `everest` command line tool.

Open up a terminal and call

```
everest 205071984
```



User Tools

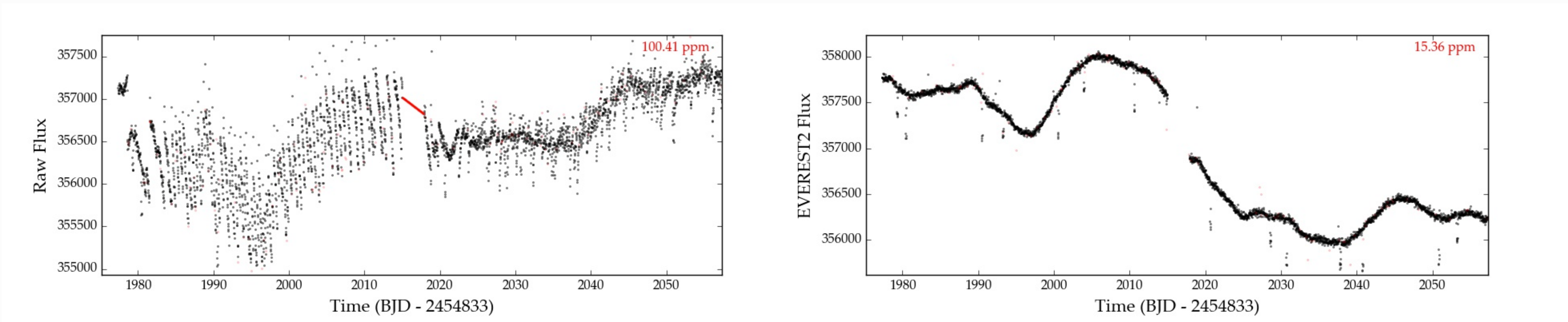
Once you've installed `everest`, you can easily import it in Python:

```
import everest
```

Say we're interested in **EPIC 201367065**, a `K2` transiting exoplanet host star. Let's instantiate the `Everest` class for this target:

```
star = everest.Everest(201367065)
```

```
star.plot()
```



http://staff.washington.edu/rodluger/everest

Masking Transits

If you're using `everest` for exoplanet/eclipsing binary science, you will likely want to apply a mask to any transits in the light curve to prevent them from getting washed out by the least-squares fitting step. The de-trended light curves provided in the catalog automatically mask large outliers, but it is still strongly recommended that all transits be masked during the de-trending step to minimize de-trending bias. This can be done **easily** and **quickly** as follows:

```
star.mask_planet(t0, per, dur = 0.2)
star.compute()
```

where `t0` is the time of first transit, `per` is the period, and `dur` is the full transit duration (all in days).

Try it:

```
star.mask_planet(1980.42, 10.054)
star.compute()
star.plot_folded(1980.42, 10.054)
```

Masking Transits

If you're using `everest` for exoplanet/eclipsing binary science, you will likely want to apply a mask to any transits in the light curve to prevent them from getting washed out by the least-squares fitting step. The de-trended light curves provided in the catalog automatically mask large outliers, but it is still strongly recommended that all transits be masked during the de-trending step to minimize de-trending bias. This can be done **easily** and **quickly** as follows:

```
star.mask_planet(t0, per, dur = 0.2)
star.compute()
```

where `t0` is the time of first transit, `per` is the period, and `dur` is the full transit duration (all in days).

Alternatively, you can specify directly which indices in the light curve should be masked by setting the `star.transitmask` attribute:

```
star.transit_mask = np.array([0, 1, 2, ...], dtype = int)
star.compute()
```

star.plot()	Plot the de-trended light curve
star.dvs()	Data validation summary
star.plot_pipeline(*pipeline_name*)	Plot a different pipeline's light curve
star.mask_planet()	Mask a planet's transits
star.compute()	Compute (after masking)
star.time	Time array
star.flux	De-trended flux array
star.fraw	Raw flux array
star.fraw_err	Raw flux error array
star.cdpp	De-trended photometric precision

https://archive.stsci.edu/prepds/everest/

`everest-stats` - De-trending Statistics

The `everest-stats` command accepts several options, which we list below.

`season`	The season number. For `K2`, this is the campaign number. Note that fractional seasons are allowed (i.e., 6.0). Default is 0
`model`	The `everest` model name. Default `nPLD`
`compare_to`	The `everest` model or pipeline to compare against. Default `everest1`
`-m` *mission*	The mission name (`k2` | `kepler` | `tess`). Default `k2`
`-s`	Plot the short cadence versus long cadence CDPP statistics. If no campaign is specified, shows all campaigns
`-p`	Plot the CDPP comparison for all confirmed planet hosts
`-i`	Plot the transit injection/recovery results.

https://archive.stsci.edu/prepds/everest/

Barbara A. MIKULSKI ARCHIVE FOR SPACE TELESCOPES

MAST | STScI | Tools | Mission Search | Search Website | Follow Us | Register | Forum

About MAST | Getting Started

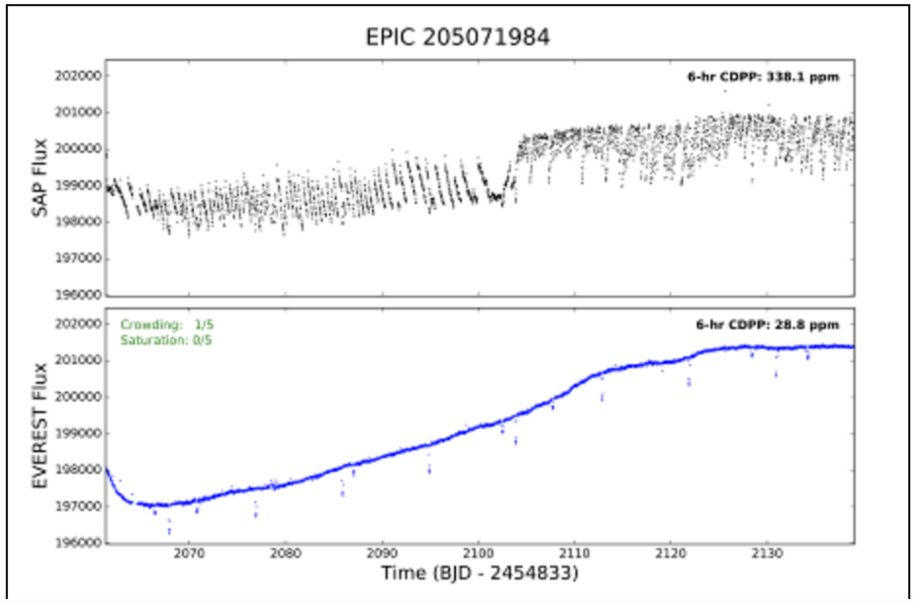
EPIC Variability Extraction and Removal for Exoplanet Science Targets ("EVEREST")

Luger et al. 2016, arXiv:1607.00524

Source code available on GitHub. Documentation is available at MAST (applicable to data archived here) and at Washington (updated by the team as their software evolves).

Introduction | Description of Data Products | Data Access | Download README

Introduction



EVEREST is an open-source pipeline for removing instrumental systematics from K2 light curves, using a combination of pixel-level decorrelations to remove spacecraft pointing error and Gaussian processes to capture astrophysical variability. Corrected light curves have precision comparable to the original Kepler mission for targets brighter than Kp = 13, and within a factor of 2 for fainter targets. Transit injection and recovery tests have been performed to validate the pipeline, and comparisons with other K2 detrended HLSPs is performed (consult the Luger et al. 2016 paper for further details).

NOTE: Since EVEREST performs least-squares fits to reduce the noise in K2 light curves, astrophysical features such as transits and eclipses can sometimes be slightly shallower in the de-trended dataset. In order to prevent this, EVEREST automatically masks outliers prior to computing the fits. However, low signal-to-noise staff.washington.edu/rodluger/everest/catalog.html

everest
2.0.6

Search docs

- Overview
- The Catalog
  - Data Release Notes
  - FITS Files
  - Data Validation Summaries
  - Bulk Download
  - Known Issues
- The Code
- The Papers
- Old Versions

everest » The Catalog

The Catalog

Search the EVEREST Catalog

Target ID: e.g., 201367065

Mission: K2 Kepler TESS

The **EVEREST** catalog is available here and includes `.fits` files with the de-trended light curve data as well as `.pdf` data validation summaries associated with each EPIC target. Follow the links below for detailed information about the catalog. Note that it is **highly recommended** that users access the catalog through the *everest interface*, as this allows for post-processing of the light curves with custom masks.

- Data Release Notes
  - Updates to the source code
  - Updates to the catalog
- FITS Files
  - `[0]` Primary HDU
  - `[1]` Lightcurve HDU
  - `[2]` Pixels HDU
  - `[3]` Aperture HDU
  - `[4]` Images HDU
  - `[5]` HiRes HDU

EVEREST

Papers THE ASTRONOMICAL JOURNAL, 152:100 (14pp), 2016 October
LUGER ET AL. (2017), ARXIV:1702.05488

User-friendly code https://github.com/rodluger/everest

Documentation http://staff.washington.edu/rodluger/everest

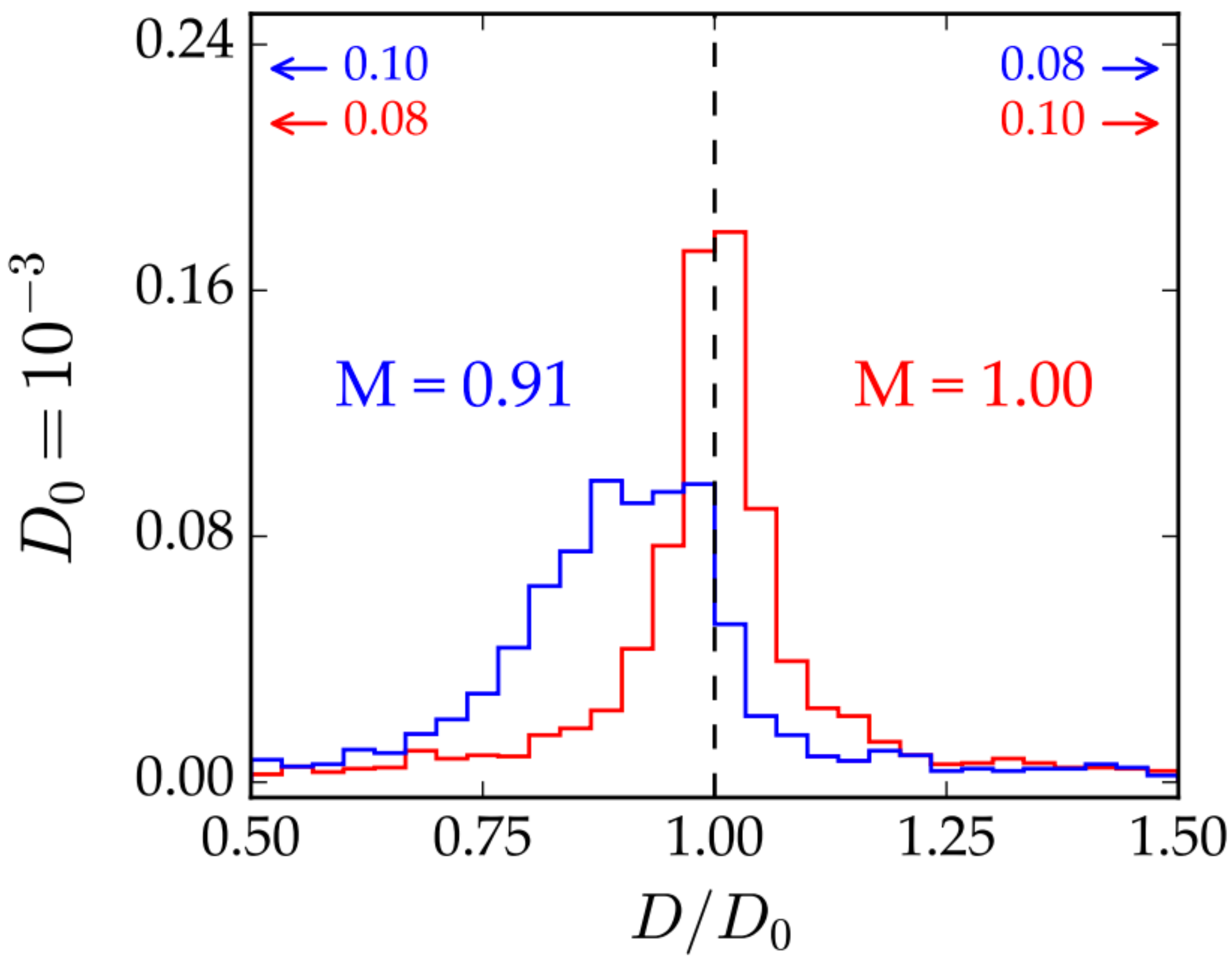
Light curves https://archive.stsci.edu/prepds/everest

PART III

TRANSIT SEARCH

Transit Search

Luger et al. (2017a)



- **Control**
- **Injection recovery**

De-trending can lead to overfitting!

If you don't include a transit model.

Delta-Chi Squared

Likelihood of data, no transit model

$$\log \mathcal{L}_0 = -\frac{1}{2}\underbrace{\mathbf{y}}_{\text{raw data}}^\top \underbrace{\mathbf{\Sigma}^{-1}}_{\text{covariance}}\mathbf{y} + C$$

Likelihood of data, transit model at t = t_0

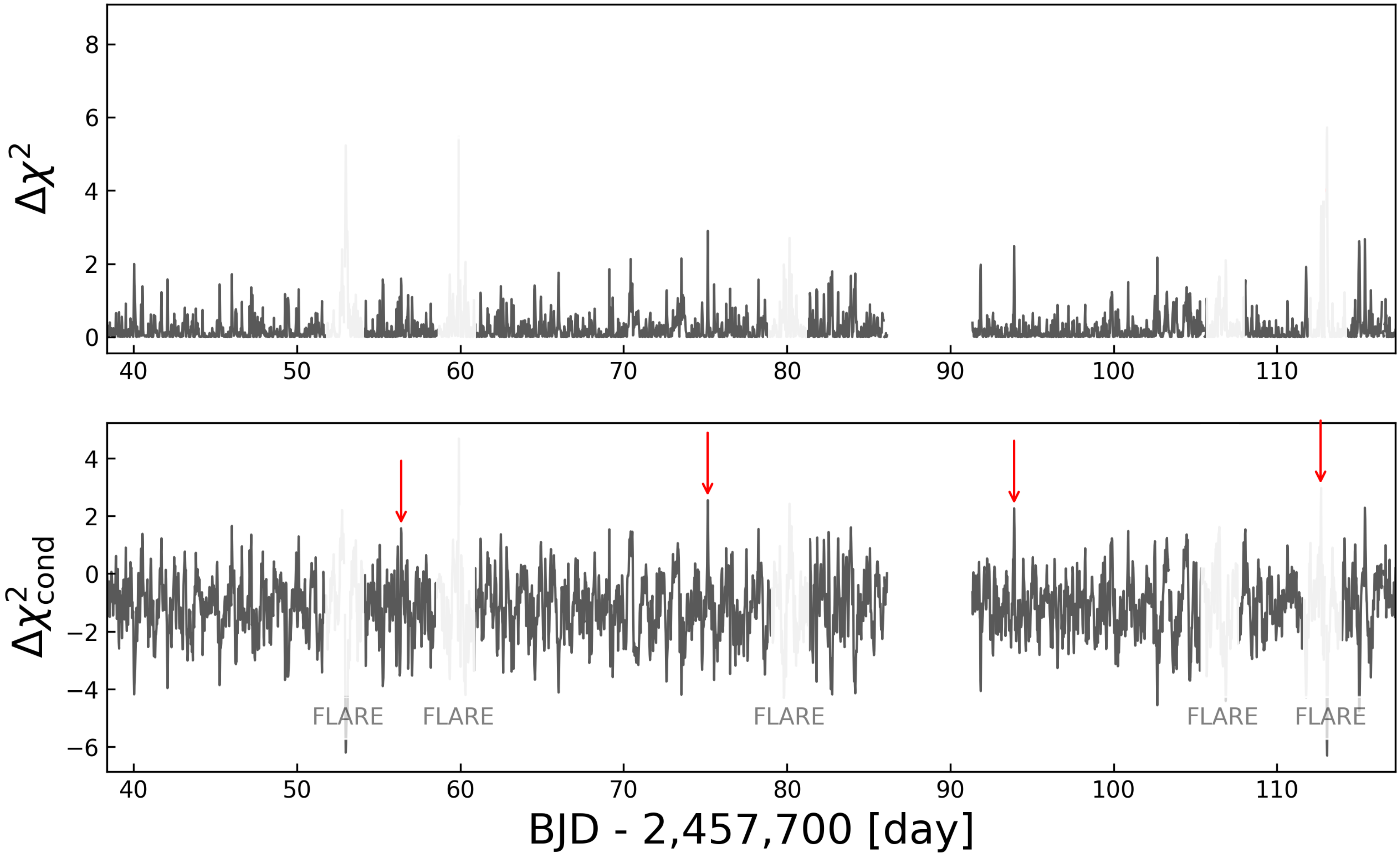
$$\log \mathcal{L} = -\frac{1}{2}(\mathbf{y} - \underbrace{\mathbf{m_{t_0}}}_{\text{transit model}})^\top \mathbf{\Sigma}^{-1}(\mathbf{y} - \mathbf{m_{t_0}}) + C$$

Improvement in chi-squared metric

$$\Delta\chi^2 = 2(\log \mathcal{L} - \log \mathcal{L}_0)$$

Delta-Chi Squared

Luger et al. (2017b)



Delta-Chi Squared

Let's search some light curves!

```
git clone https://github.com/rodluger/everest-workshop.git
cd everest-workshop
```

```
pip install tqdm
```

Group 1

*De-trend **then** search*

```
python group1.py
```

Group 2

*De-trend **and** search*

```
python group2.py
```

Troubleshooting

```
touch matplotlibrc
```

↓

```
backend: MacOSX
```

or

```
backend: Qt4Agg
```

or

```
backend: TkAgg
```

PART IV

BACKUP SLIDES

Pixel Level Decorrelation

???

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right)^{\top} \mathbf{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right) + C$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right)^\top \mathbf{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right) + C$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right)^{\top} \mathbf{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right) + C$$

Kind of like

$$\chi^2 = \sum_i \left(\frac{y_i - m_i}{\sigma_i} \right)^2$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2}\left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right)^\top \boldsymbol{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right) + C$$

$$-\tfrac{1}{2}\chi^2$$

(white noise)

Related to

$$\chi^2 = \sum_i \left(\frac{y_i - m_i}{\sigma_i}\right)^2$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w} \right)^{\top} \boldsymbol{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w} \right) + C$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right)^\top \mathbf{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w}\right) + C$$

$$\frac{\mathrm{d} \log \mathcal{L}}{\mathrm{d}\mathbf{w}} = 0$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

Covariance matrix

$$\log \mathcal{L} = -\frac{1}{2} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w} \right)^\top \mathbf{\Sigma}^{-1} \left(\mathbf{y} - \mathbf{X} \cdot \mathbf{w} \right) + C$$

$$\frac{\mathrm{d} \log \mathcal{L}}{\mathrm{d} \mathbf{w}} = 0$$

$$\hat{\mathbf{w}} = \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \mathbf{X} \right)^{-1} \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \cdot \mathbf{y} \right)$$

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \mathbf{X} \right)^{-1} \left(\mathbf{X}^\top \mathbf{\Sigma}^{-1} \cdot \mathbf{y} \right)$$

*That's it!**

Pixel Level Decorrelation

$$\mathbf{y}' = \mathbf{y} - \mathbf{X} \cdot \mathbf{w}$$

$$\mathbf{y}' = \mathbf{y} - \mathbf{X}\left(\mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \mathbf{X}\right)^{-1} \left(\mathbf{X}^\top \boldsymbol{\Sigma}^{-1} \cdot \mathbf{y}\right)$$

*That's it!**

**optimizing ∑ can be tricky*