Machine learning and scalable analysis of light curves

Lessons I've learned and open questions I have

Konstantin (Kostya) Malanchev[1], SNAD team,
ELAsTiCC team, LINCC Frameworks team

[1] Carnegie Mellon University, LINCC Frameworks

2026.03.26 Rapid Response Workshop, Pasadena

Light-Curve Infrastructure and ML Stories

- Data storage for bulk light curve analysis.
- Light curve feature extraction.
- Active anomaly detection.
- Classification

Lessons learnt from SNAD, ELAsTiCC, LINCC Frameworks teams and others.

SNAD

Collaboration for light-curve anomaly detection, https://snad.space

SNAD 2024 Workshop in Rio de Janeiro

LINCC Frameworks

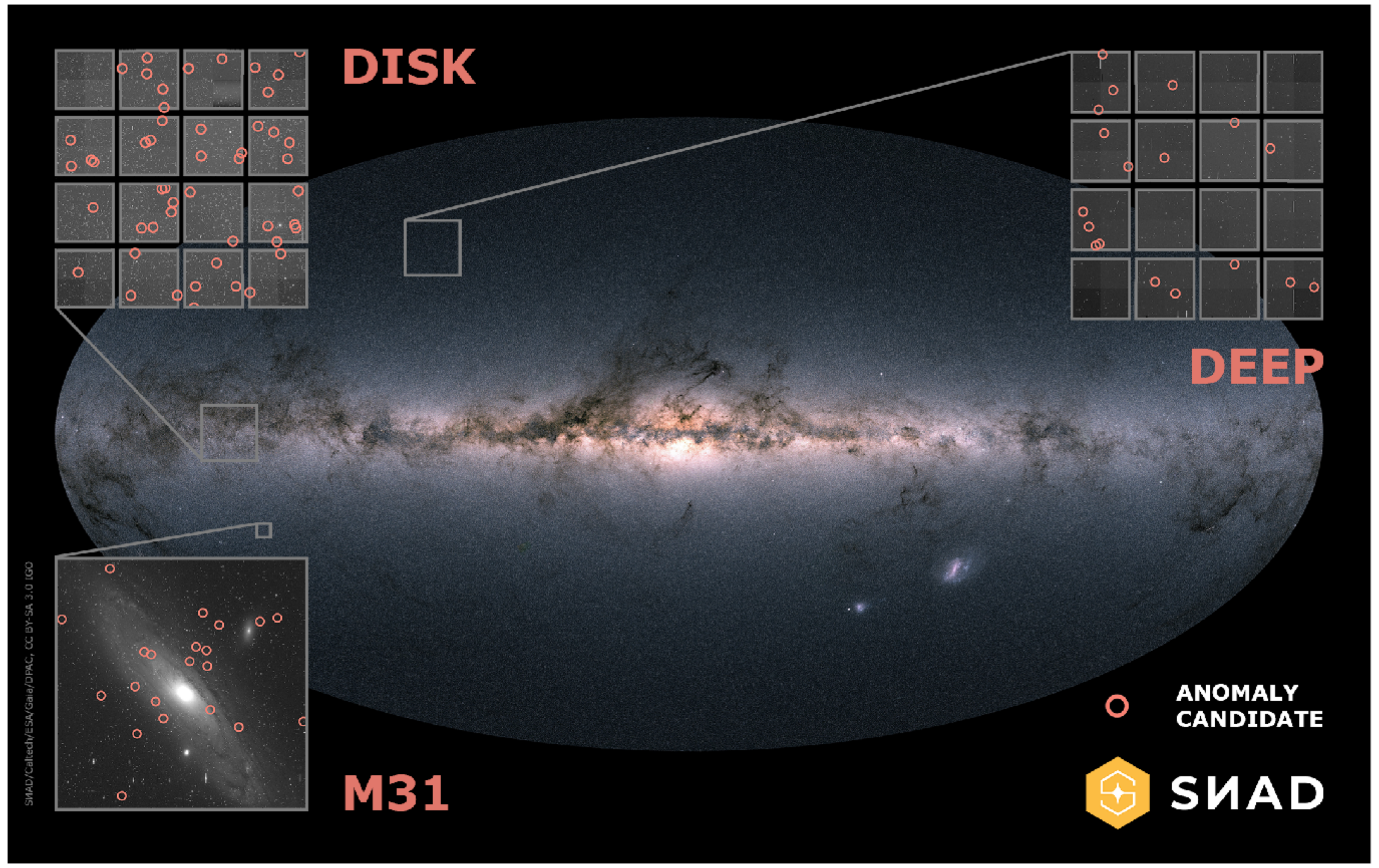
Team working on software for large survey users

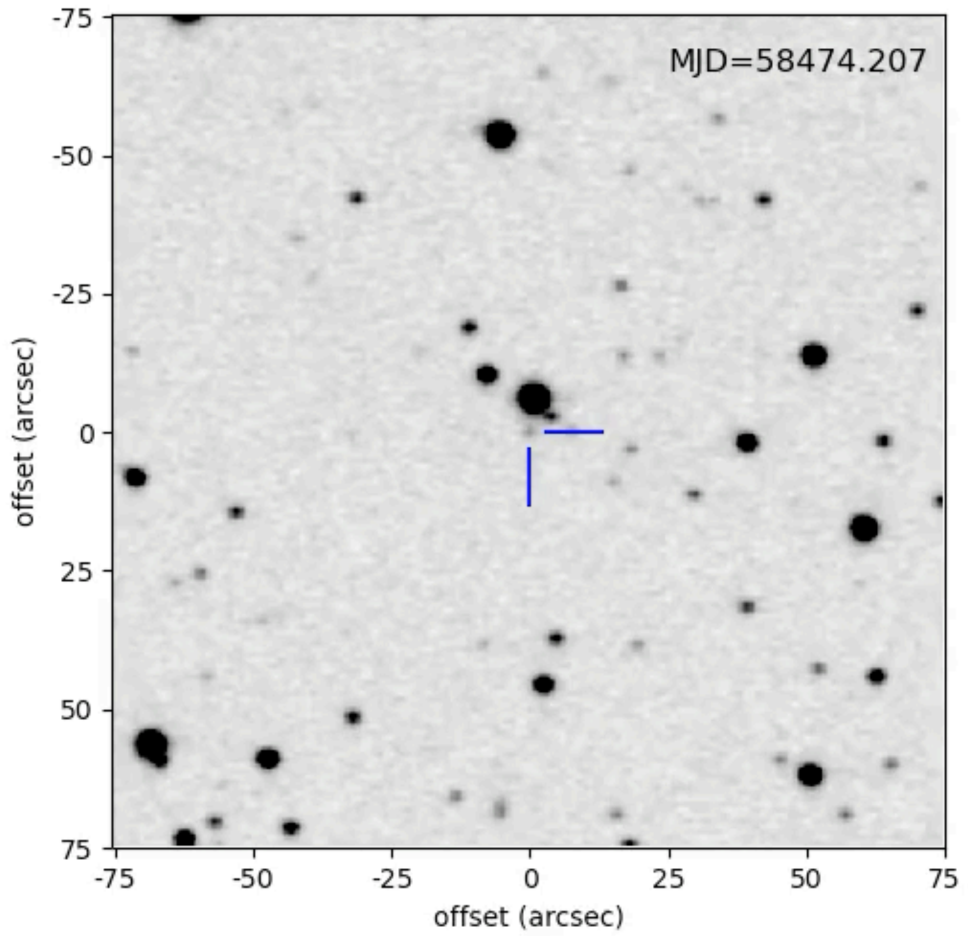
LINCC Up 2023, Pittsburgh

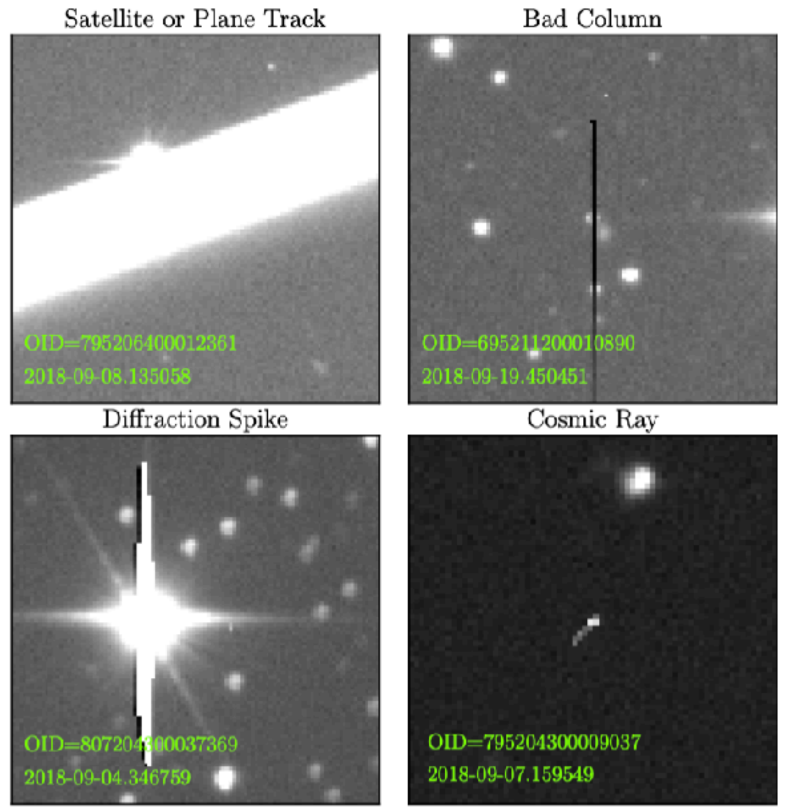
Case: Zwicky Transient Facility DR3

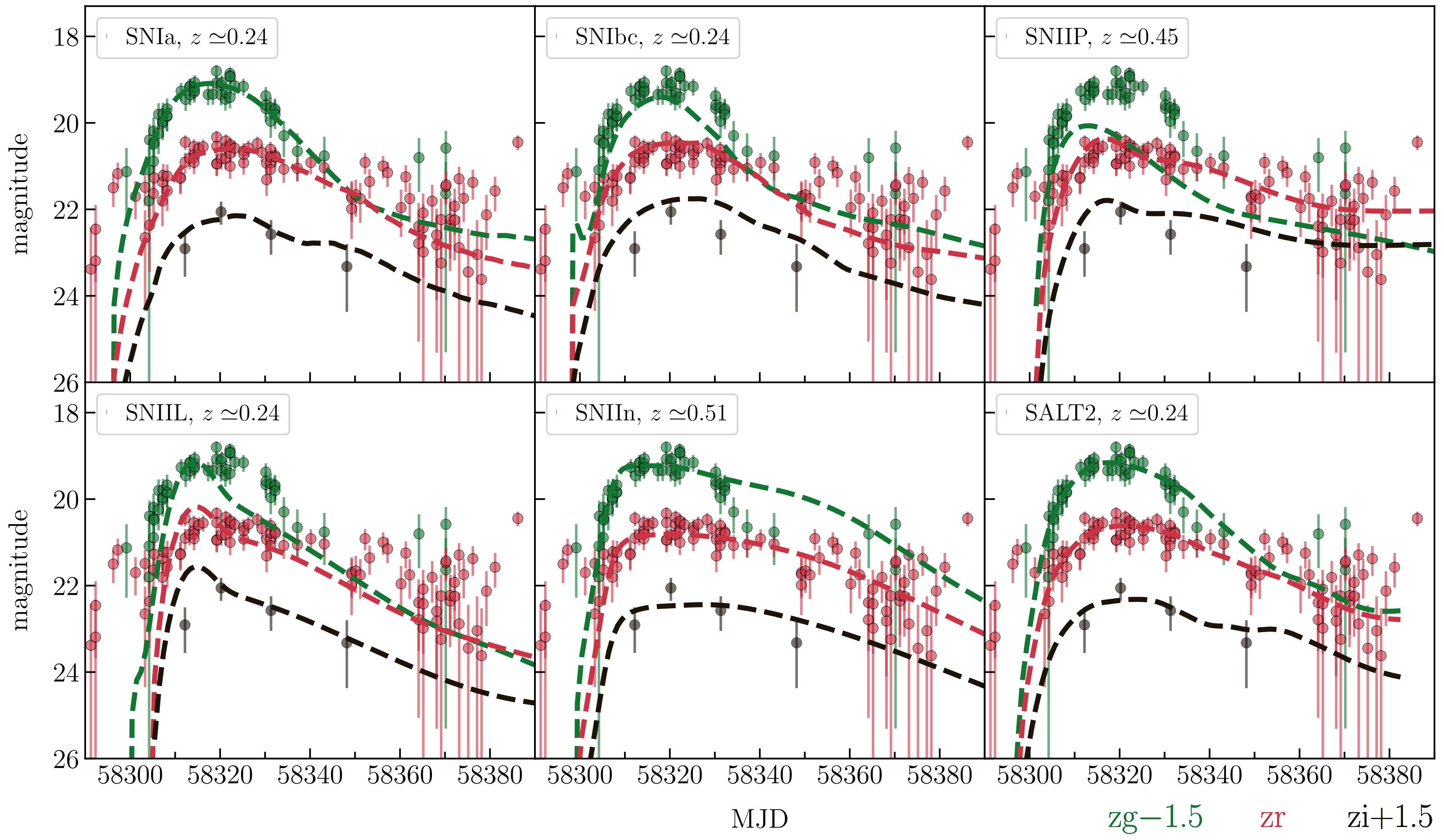
KM+20

- Three fields of ZTF DR3
- **~2×10^6 objects total**
- Four outlier detection algorithms
- 89/227 anomaly candidates
  - **Six (5/6 are new!) SN Ia candidates**
  - RS CVn (confirmed by our spectra)
  - Mira binary candidate
- 188/277 bogus light curves
  - Double star defocusing
  - Bright Mira "echos"
  - Asteroid overlap
  - Bad columns, satellites, spikes, ghosts, etc









Billion Light-Curve Dataset

For me the challenge started with DR1 in 2019

- ZTF DR1, May 2019
  - I was trying to get the whole data
  - 1.7 billion light curves, 59 billion
  - In 16 threads
  - Wikipedia: "In [computing], a **deni**
    (**DoS attack** /dɒs/ *doss*[1]) is a cyb
    perpetrator seeks to make a mac
    resource unavailable".
  - The bulk downloadable textual fi
    were released later. 2.2 TB (0.5 TB compressed).

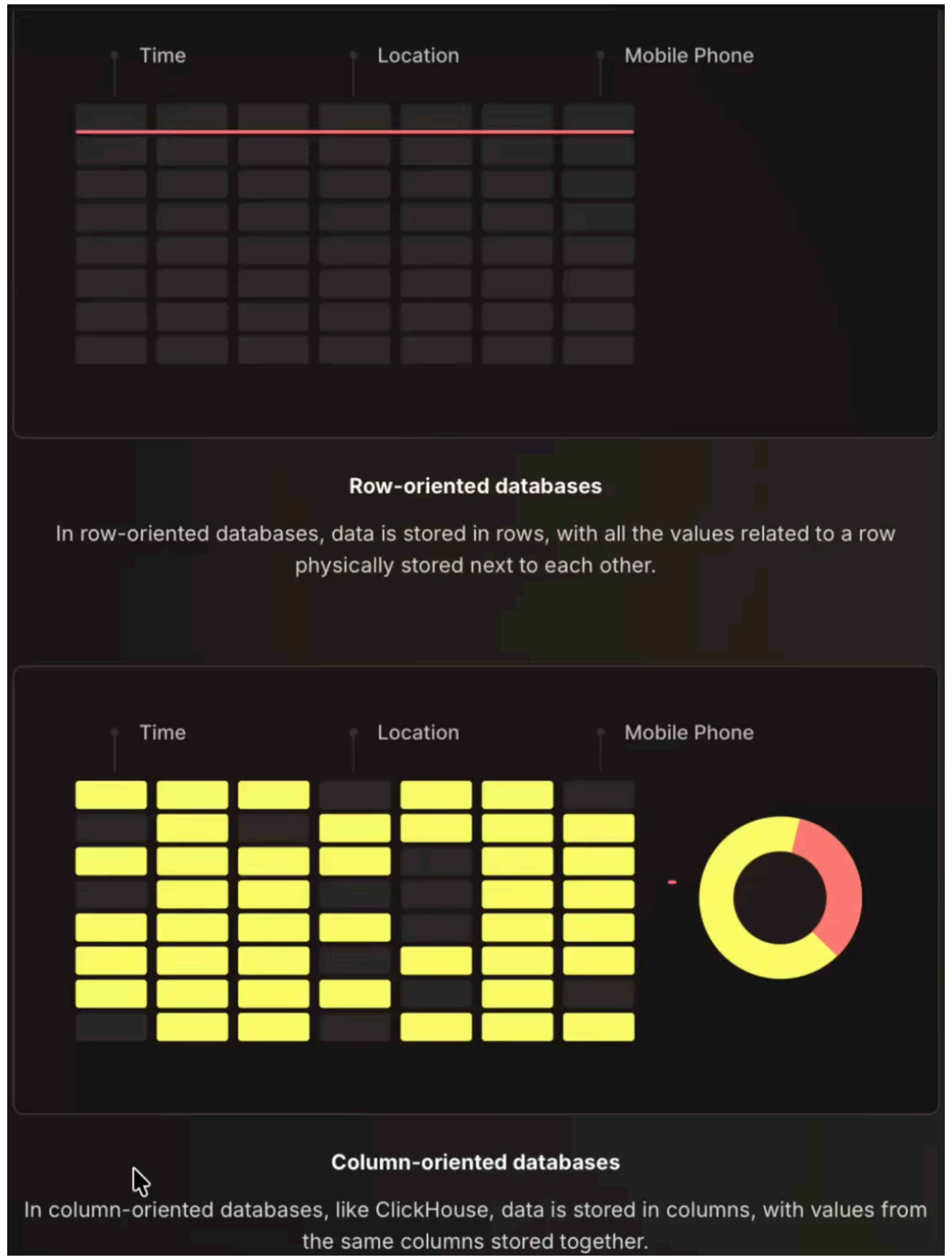
Lesson: do DoS science APIs

~~Lesson: do DoS science APIs~~
Lesson: do not DoS science APIs
Lesson: release catalogs in a bulk-analyzable format

Storage Problems

How do we even count objects?

1. Python parser
2. Classical astronomical DBMS approach: PostgreSQL
    - `SELECT count(*) FROM ztf;`
3. Columnar database: Clickhouse
    - Data is encoded and compressed, 1/4 of PostgreSQL size
    - `SELECT src.mjd, src.mag, src.err FROM obj JOIN src USING (objectid);`
4. Nested columns



clickhouse.com

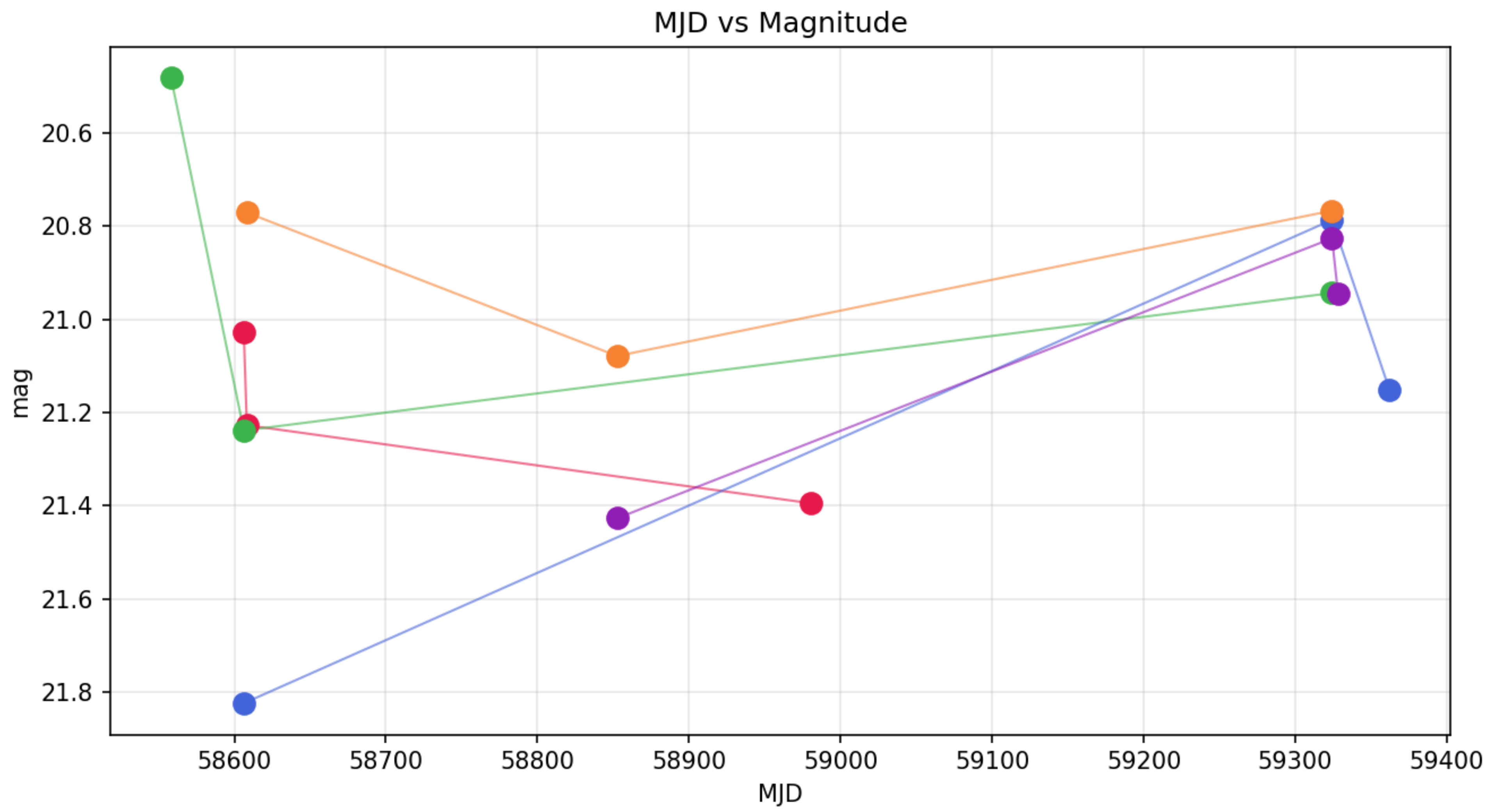
Nested Columns

Packing everything into a single table

List-columns efficiently handle time-domain data.

```
SELECT
    oid,
    ra,
    dec,
    mjd,
    mag
FROM ztf.dr23_olc
WHERE length(mag) = 3
LIMIT 5

Query id: f9f6208f-13c6-4d07-ba2a-c18a83e0cd82
```



	oid	ra	dec	mjd	mag
1.	1757215200007396	209.07936096191406	47.032291412353516	[58606.25854,58608.26613,58980.33208]	[21.028872,21.227367,21.394724]
2.	1757215200007008	209.10459899902344	47.07343673706055	[58558.25143,58606.25854,59324.25145]	[20.482203,21.23867,20.943895]
3.	1757215200007355	208.9701385498047	47.03392791748047	[58606.25854,59324.25145,59362.31036]	[21.823616,20.786997,21.152752]
4.	1757215200006327	209.10702514648438	47.14353942871094	[58608.26612,58852.56924,59324.25144]	[20.771364,21.078943,20.767294]
5.	1757215200005946	209.09503173828125	47.18548583984375	[58852.5688,59324.25144,59328.25079]	[21.426971,20.826607,20.946402]

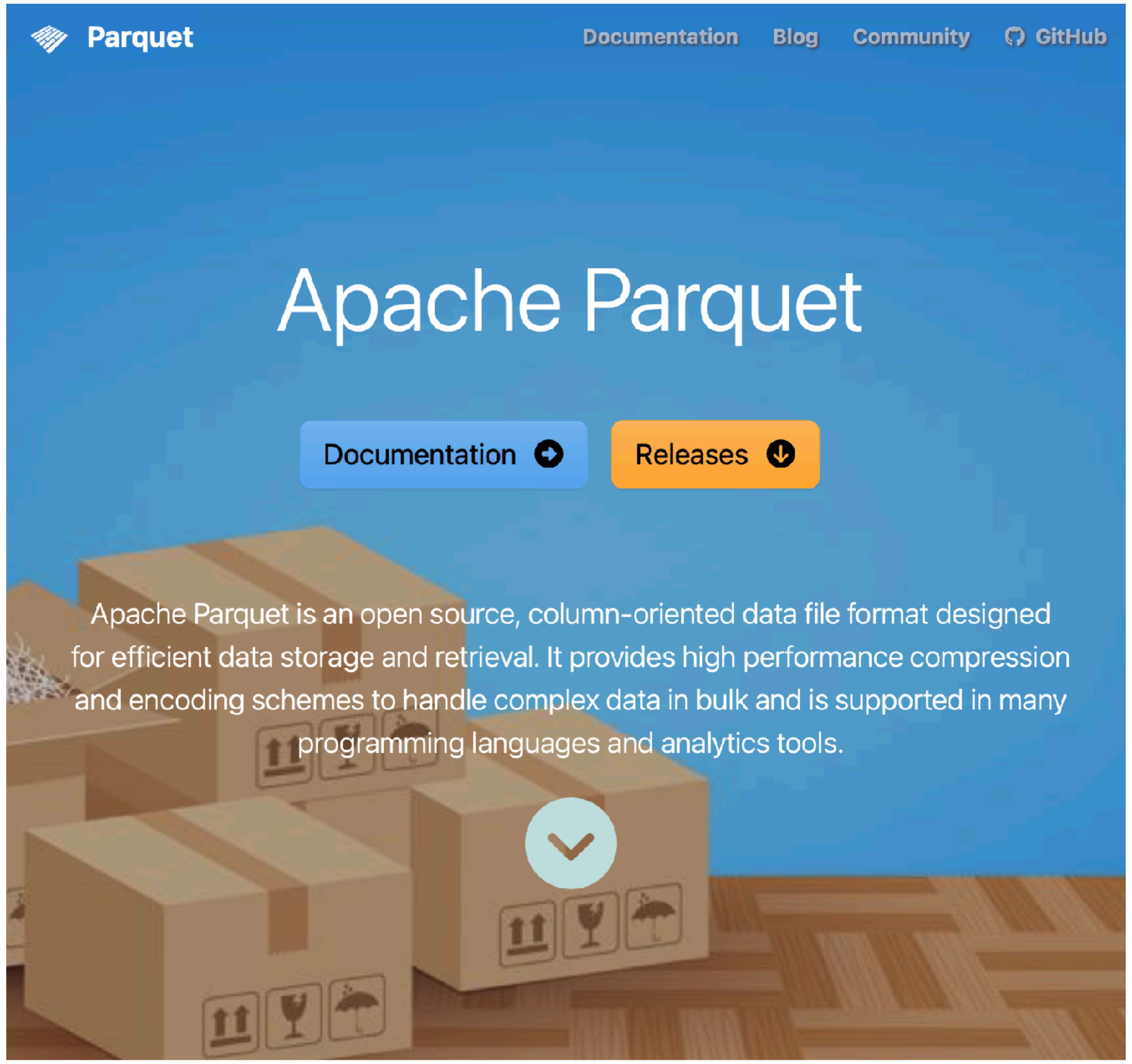
Lesson: use industry practices
Lesson: columnar-based formats help with bulk-processing

Lesson: sometimes you need two systems with the same data for two use-cases [*See the broker panel yesterday*]

Moving Computations to a Client

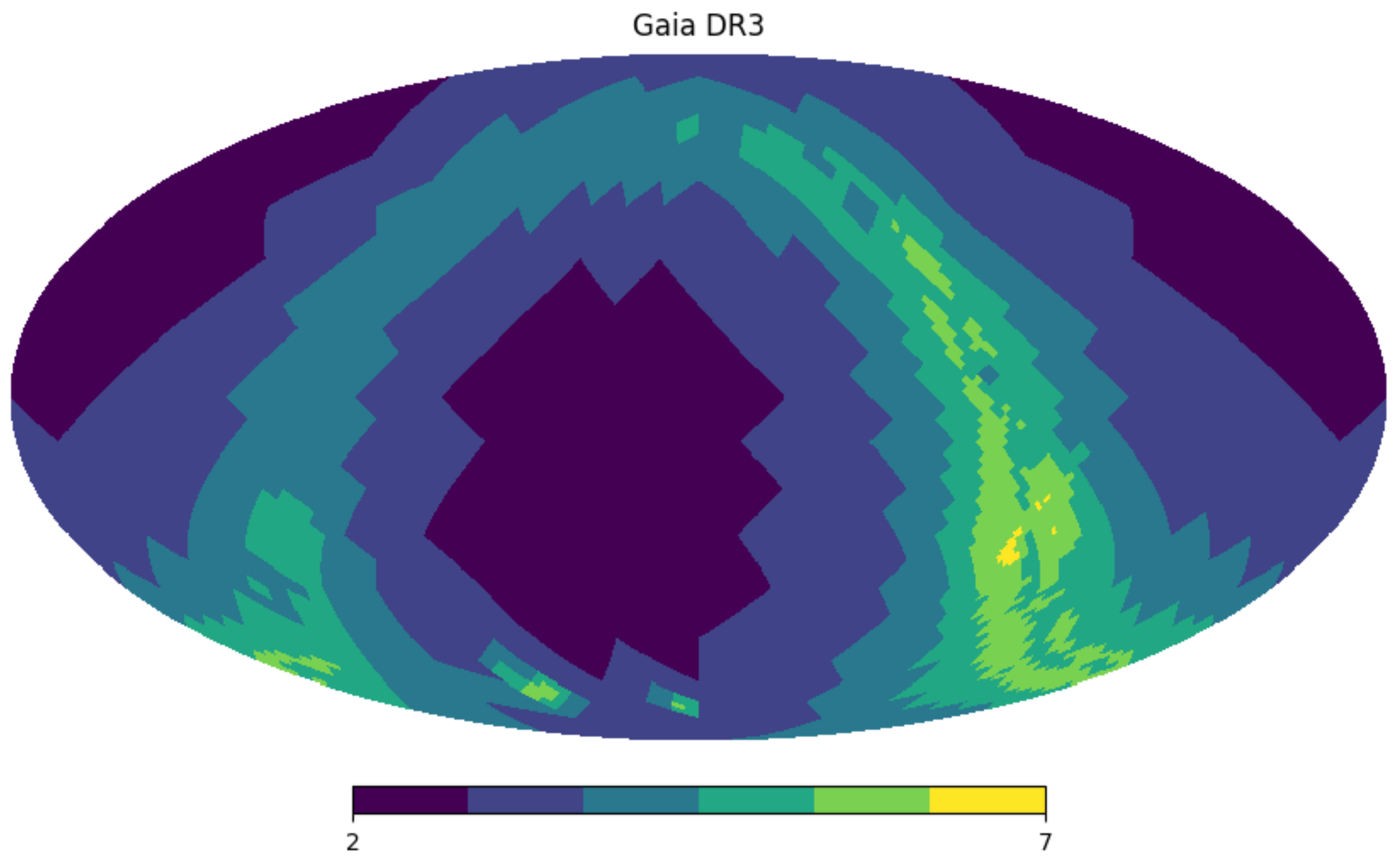
Columnar storage formats

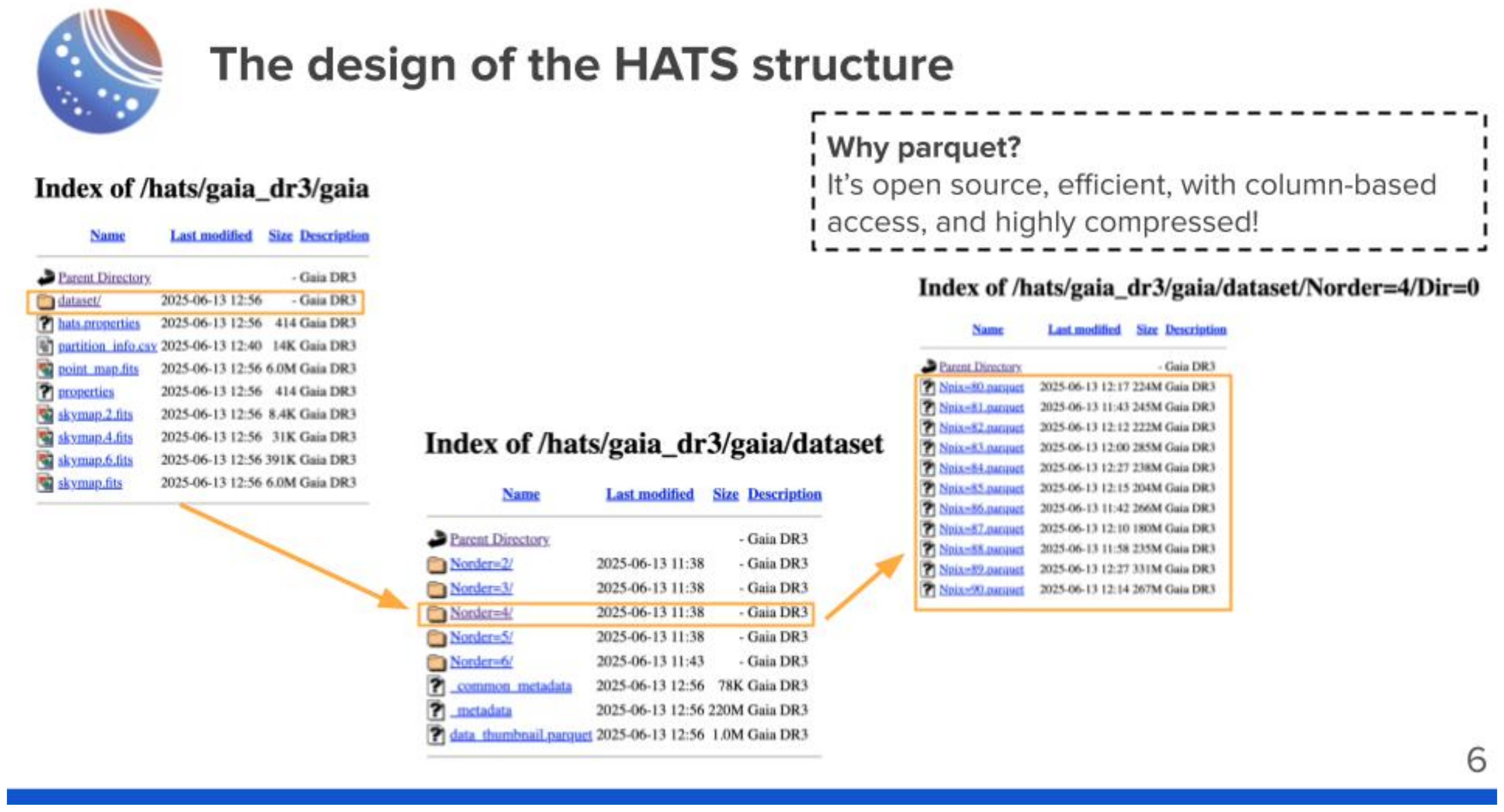
- ZTF DR5 was released in parquet, with nested list-columns
    - Size per source is 1/3—1/4 of that for DR4
- Parquet is a binary columnar format
    - Encoding and compression
    - Partitioning
    - Column selection is fast
- Could we use it to replace a database server?



HATS

- Catalog format based on Parquet and spatial partitioning.
- Parquet natively handles nested columns
    - Images
    - Light curves
    - Spectra
- [*Time-travel to yesterday for Neven Caplar's talk for more details.*]





nested-pandas and LSDB

Querying and running user-defined functions

```
[12]: # Applies the query to "nested", filtering based on "time > 60676.0"
      nf_g = nf.query("lightcurve.time > 60676.0")
      nf_g
```

[12]:

	ra	dec	lightcurve		
			time	brightness	band
0	10.000000	0.000000	60677.0	101.0	r
			+1 rows	...	...
1	15.000000	-1.000000	60676.5	5.01	g
			+1 rows	...	...
2	12.100000	0.500000	60676.6	20.1	g
			+3 rows	...	...

3 rows x 3 columns

```
[14]: import numpy as np

      # use hierarchical column names to access the flux column
      # passed as an array to np.mean
      # row_container signals how to pass the data to the function, in this
      nf.map_rows(np.mean, "lightcurve.brightness", row_container="args")
```

[14]:

id	0
0	100.266667
1	4.996667
2	20.275000

3 rows × 1 columns

_healpix_29	oid	RA	DEC	sed		lc			
				flux	ivar	lc_mjd	lc_magpsf	lc_sigmapsf	lc_fid
1999457337019883863	ZTF18aajkmuy	169.455780	29.272161	27.092773	0.0	59321.255255	20.4715	0.225765	1
				3846 rows × 2 columns		29 rows × 4 columns			
2011433646118926470	ZTF18aajiyzf	170.153050	29.048093	55.046791	0.0	58789.517326	18.663141	0.203879	1
				3844 rows × 2 columns		111 rows × 4 columns			
2011520457775535896	ZTF19aatlkjg	171.027630	30.056040	13.624688	0.0	58580.224572	20.5623	0.267047	2
				3844 rows × 2 columns		113 rows × 4 columns			
2011561936332417334	ZTF20aajrxiz	169.432130	29.446391	19.610645	0.0	58883.46162	20.3355	0.405637	1
				3860 rows × 2 columns		462 rows × 4 columns			
2011561936332430768	ZTF20aakcgwr	169.432130	29.446391	19.610645	0.0	58871.436285	20.2904	0.298648	1
				3860 rows × 2 columns		422 rows × 4 columns			
2011570170024596533	ZTF18aawplub	169.297020	29.570721	60.126614	0.0	58283.19728	19.951263	0.195744	2
				3860 rows × 2 columns		123 rows × 4 columns			

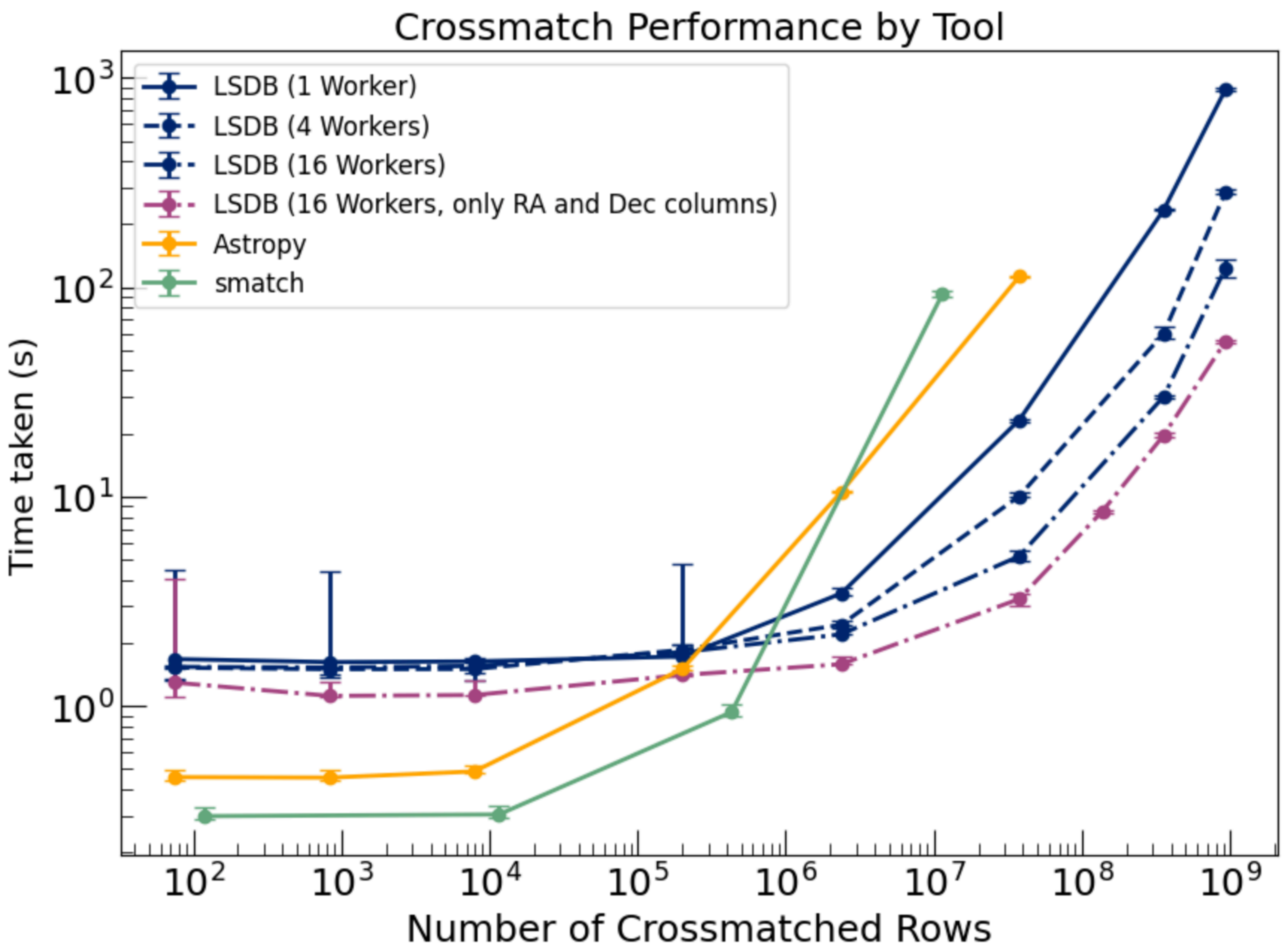
6 rows x 5 columns

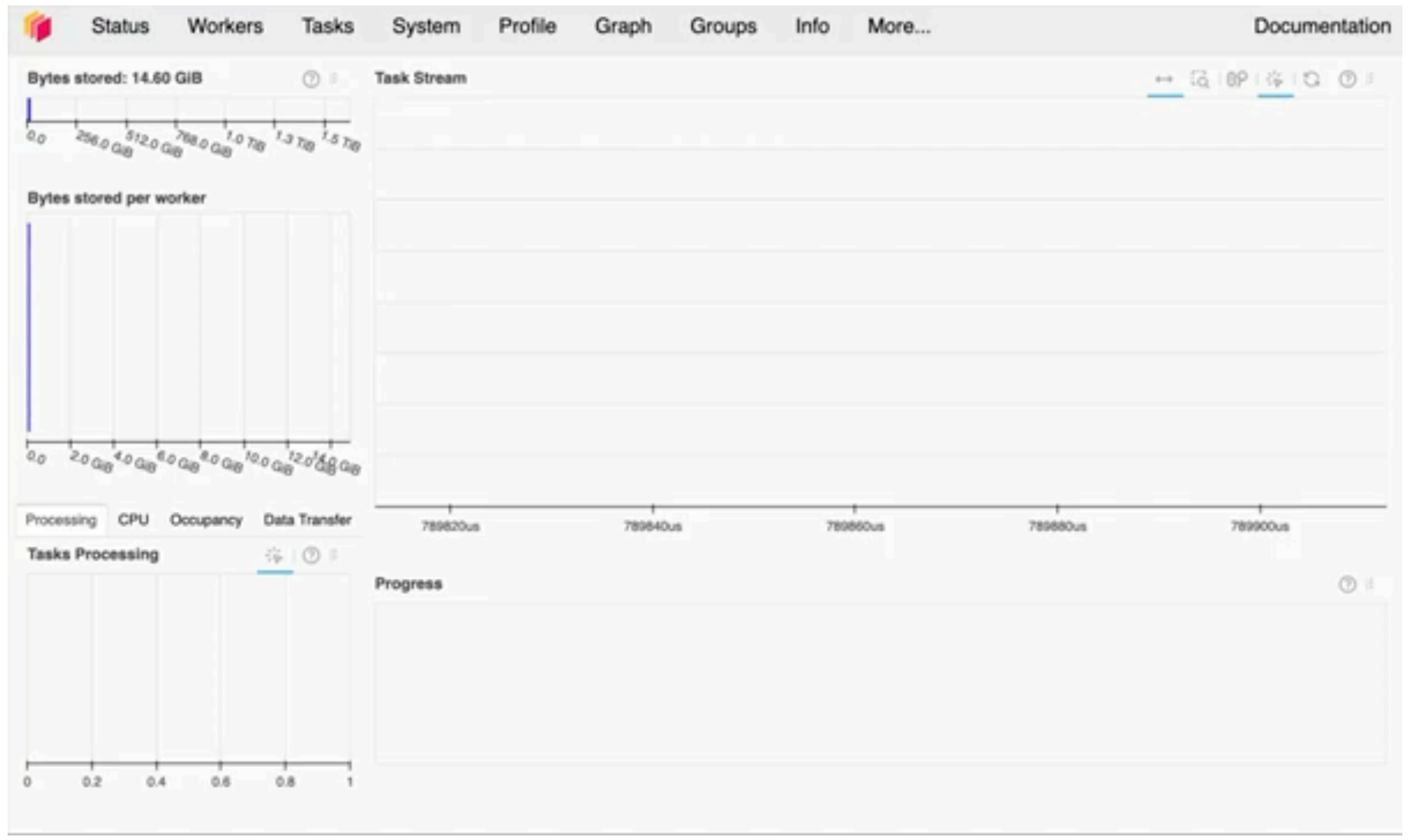
See nested-pandas.rtfd.io and docs.lsdb.io

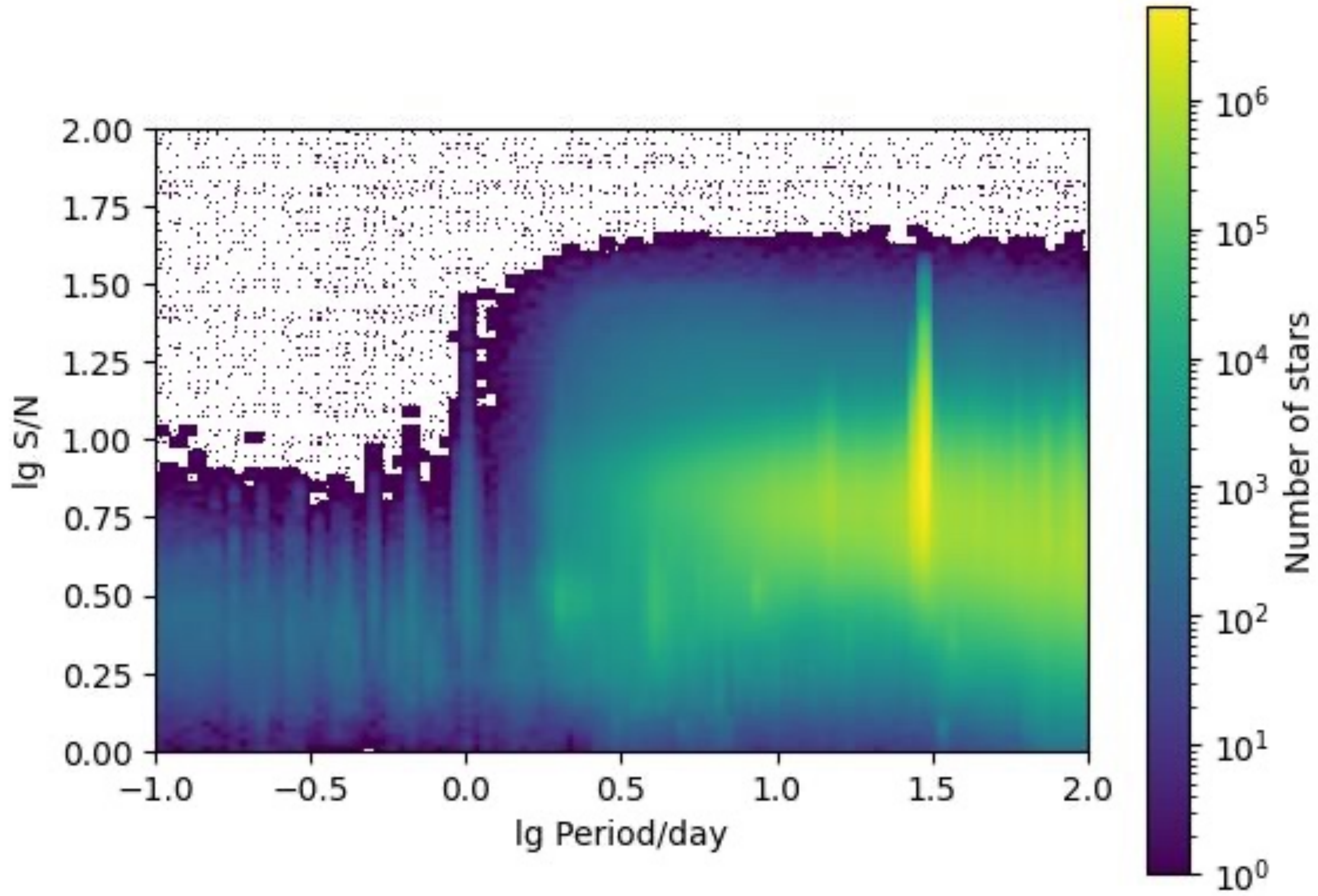
SDSS spectra and ZTF light curves

LSDB performance

[*Go back to yesterday again for Neven's talk!*]







Low-res Lomb—Scargle, 1G ZTF LCs
~90 minutes on 7 cluster nodes

Lesson: client-side computation enables new use-cases, e.g. cross-matching.

Lesson: client-side computation enables new use-cases, e.g. cross-matching.

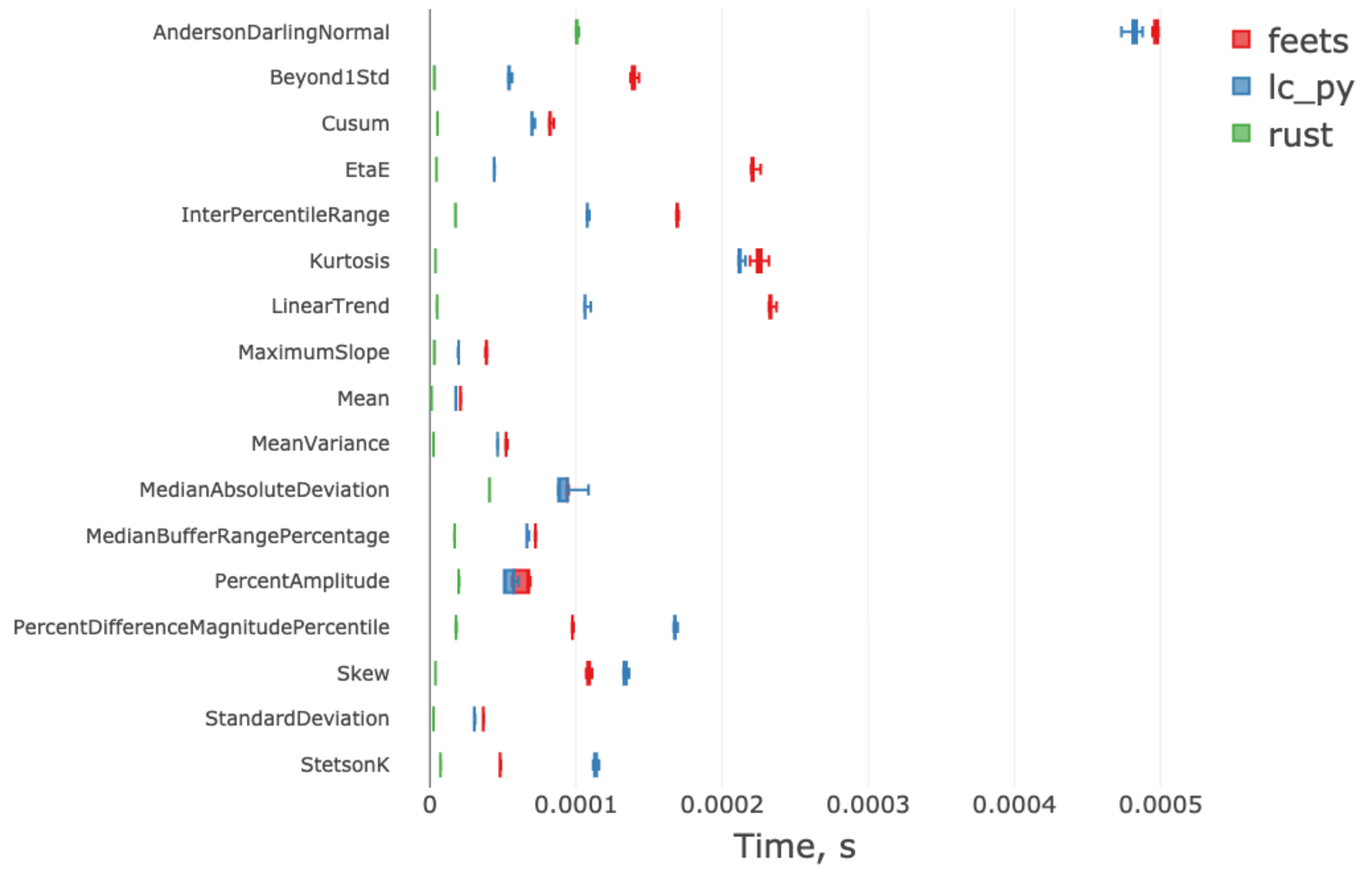
Lesson: but data locality is still useful.

Light-curve ML Stories

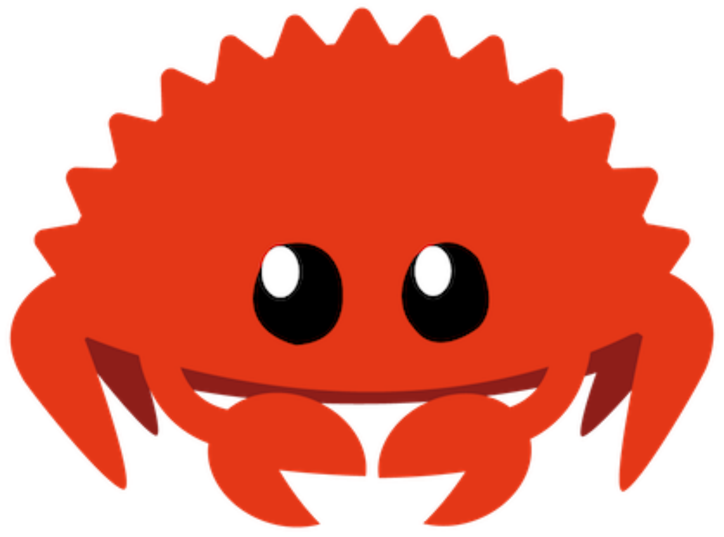
Light Curve ML: feature extraction

`pip install light-curve`

- Feature extraction is important for traditional ML and pre-filtering
- Existing Python libraries are slow for ~100M-1G catalogs
- SNAD developed `light-curve` to use in Rust
    - It was wrapped for Python
    - Up to 100× Feets performance
    - Can process the whole Rubin alert stream on a single node
    - Used by Antares, AMPEL and Fink (thank you!)
- Rust was adopted for Boom/Babamul



Benchmarks, smaller is better



Ferris the crab,
Rust mascot

Lesson: it is OK to rewrite for a speed-up.
[*See Matthew Graham talk on Boom/ Babamul*]

Lesson: it is OK to rewrite for a speed-up.
[*See Matthew Graham talk on Boom/Babamul*]

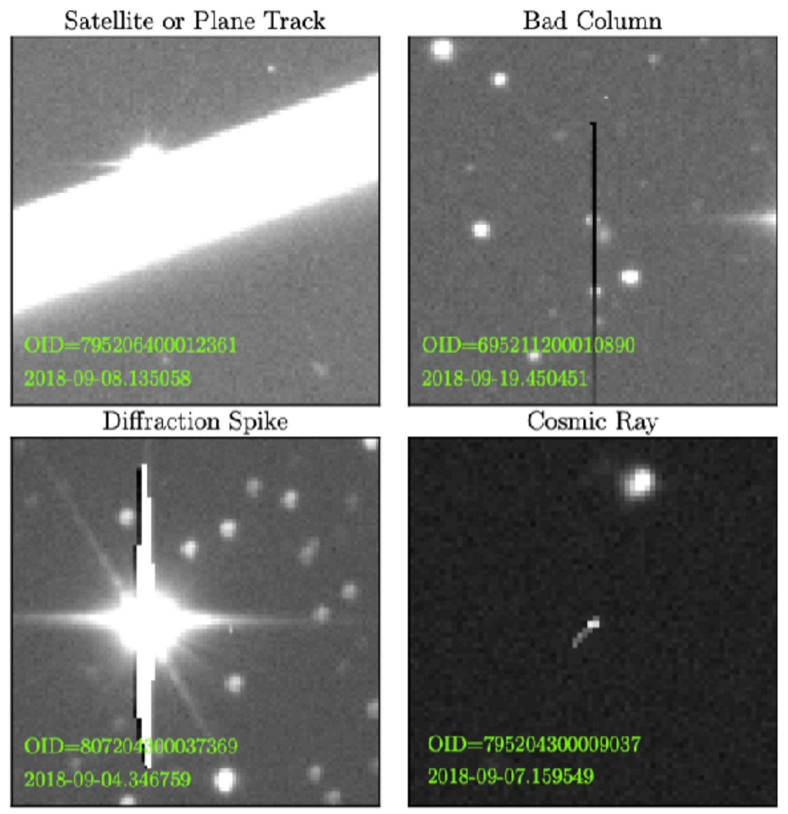
Lesson: it is less scary with an AI agent.

Open question: personalized ML/AI?

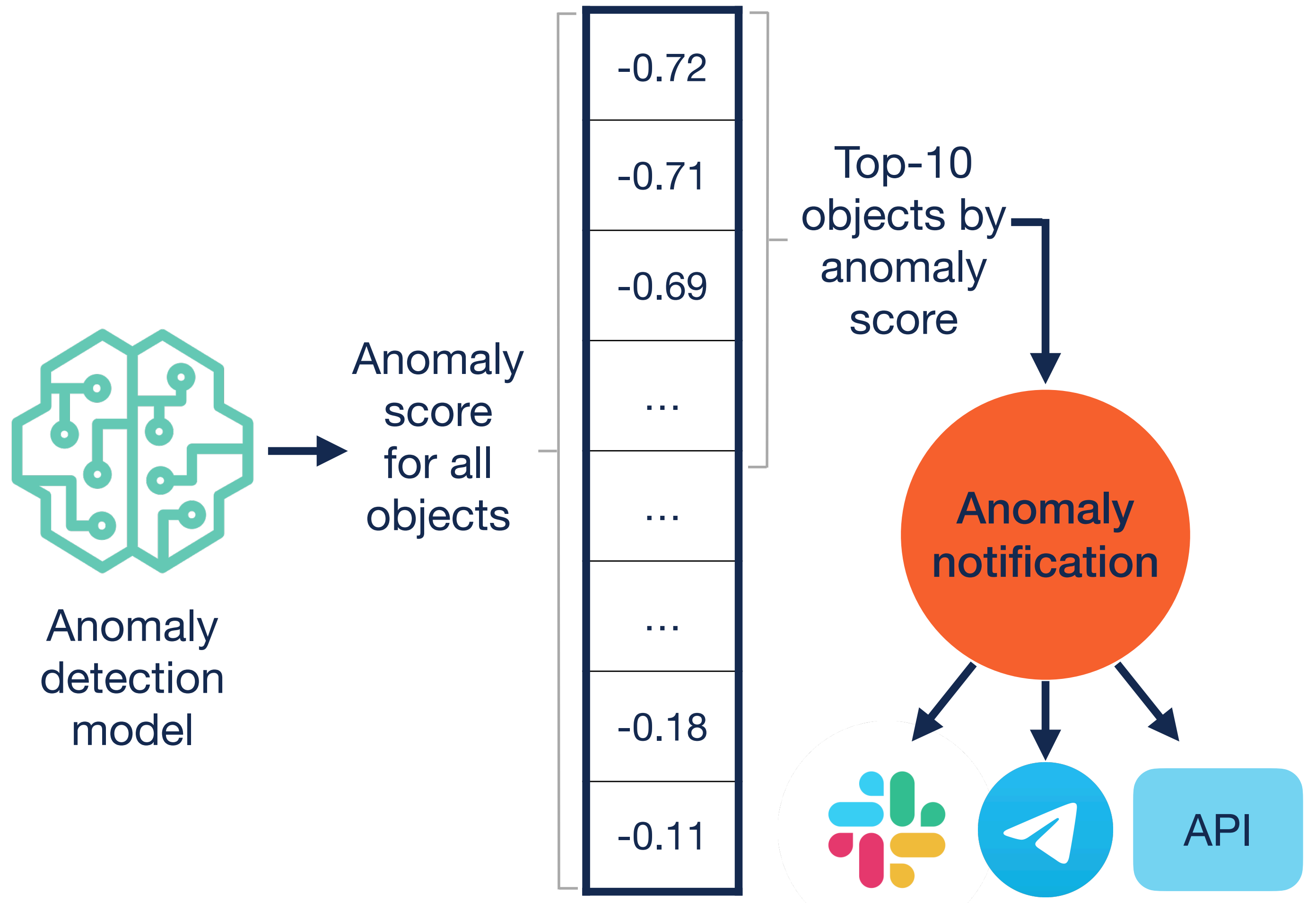
Active Learning: expert in the loop

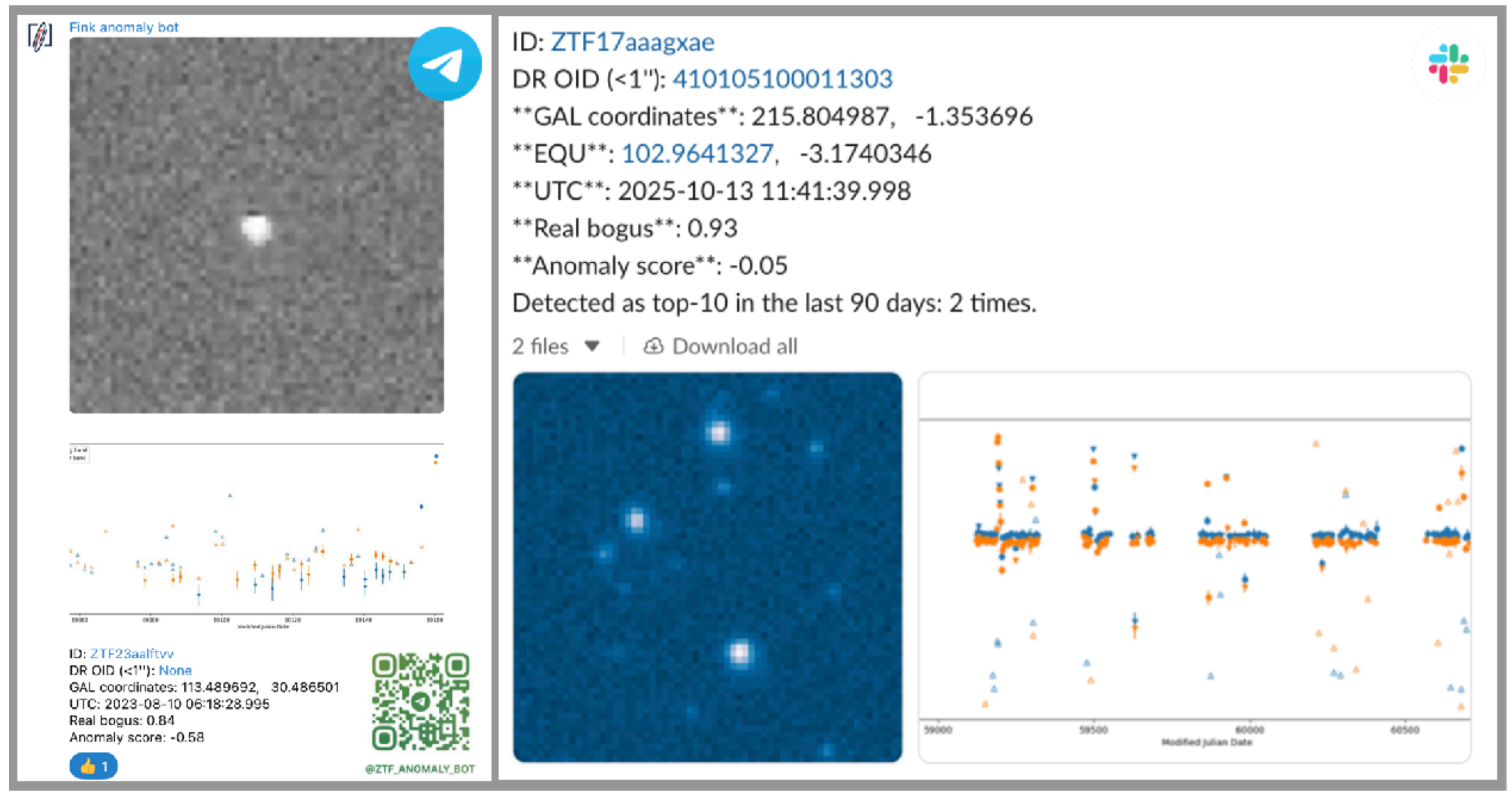
`pip install coniferest`

- Unsupervised anomaly detection frequently detects bogus events (Sánchez-Sáez+21, Pruzhinskaya+23, KM+20, ...)



- No stats definition for "anomaly"
- It would be useful to guide the machine over the feature space
- SNAD's `coniferest` package provides tooling for active anomaly detection.
- Used by Fink for personalized AD models





Fink Anomaly Telegram Bot
Provided by M. Pruzhinskaya

Open question: what do we need classification for?

Classification: question to ask

Is classification is really what we need?

A&A, 694, A130 (2025)
https://doi.org/10.1051/0004-6361/202346891

Astronomy & Astrophysics

Are light curve classification metrics good proxies for SN Ia cosmological constraining power?

Alex I. Malz[1,★], Mi Dai[2], Kara A. Ponder[3], Emille E. O. Ishida[4], Santiago Gonzalez-Gaitain[5], Rupesh Durgesh[6], Alberto Krone-Martins[7,8], Rafael S. de Souza[9], Noble Kennamer[10], Sreevarsha Sreejith[11], Lluís Galbany[12,13], The LSST Dark Energy Science Collaboration (DESC), and The Cosmostatistics Initiative (COIN)

"We thus recommend the use of cosmology-based metrics in place of classification metrics when optimising analysis pipeline designs "

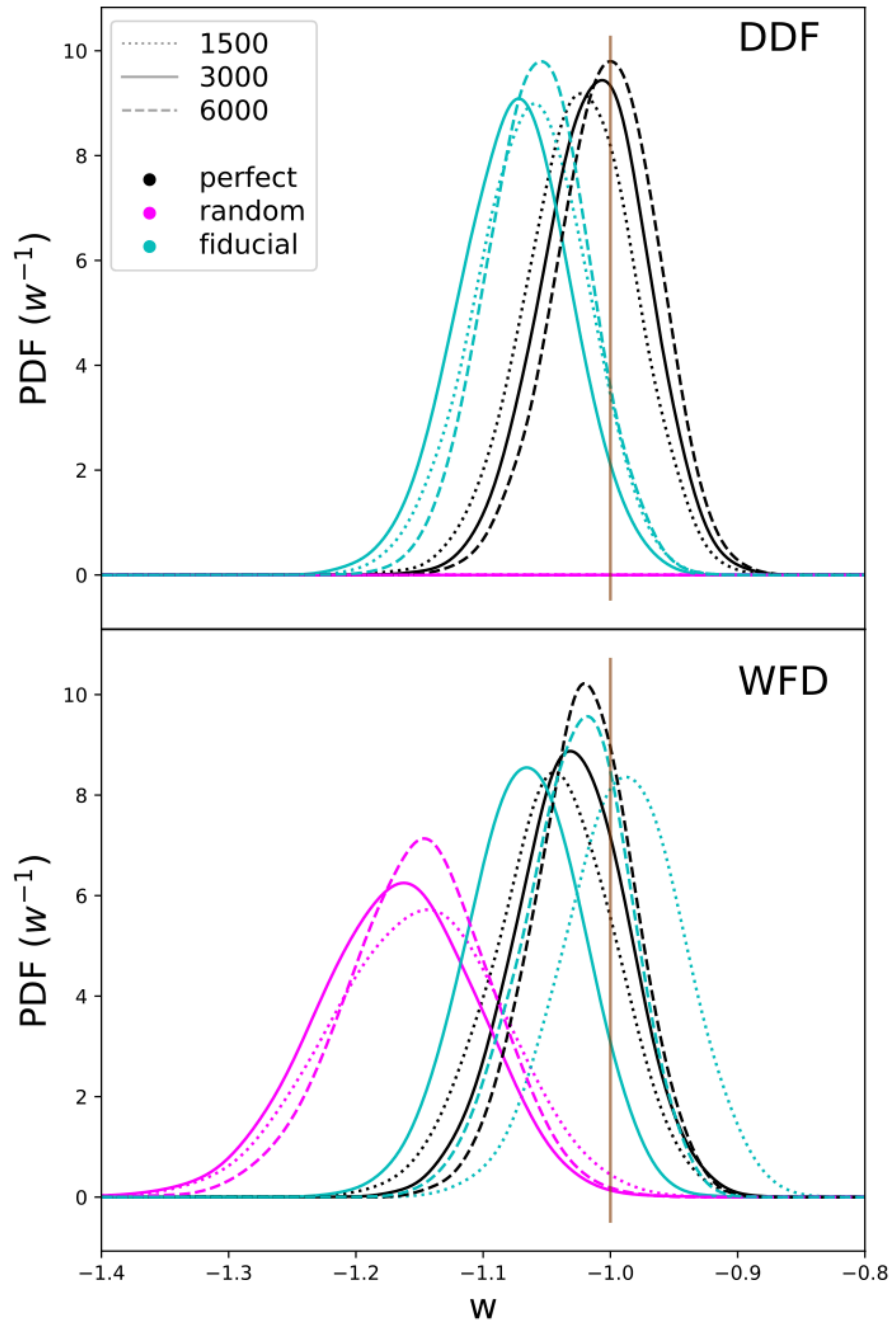
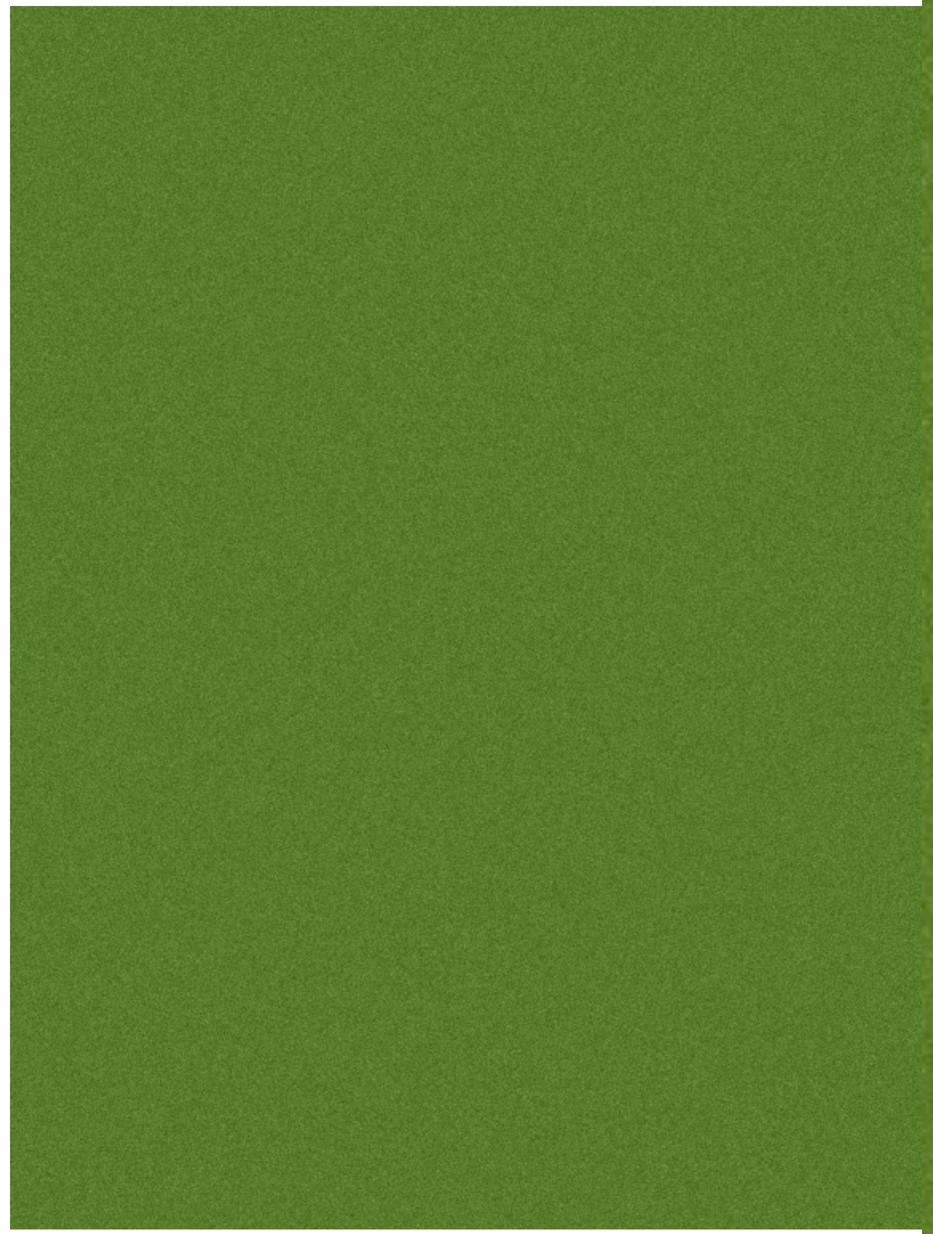


Fig. 2. StanIa posterior PDFs of w for different cosmological sample sizes (line styles) from the DDF and WFD observing strategies (panels) under the 'perfect' (black), 'random' (magenta), and 'fiducial' (cyan) mock classification schemes, with the true value of w indicated by a brown vertical line. We note that the 'random' classifier's constraints in the DDF underestimate w by so much that they cannot be shown on these axes. As the constraints are not very sensitive to the sample size, we can safely use a cosmological sample of 3000 light curves in our tests.

Classific

What is the s

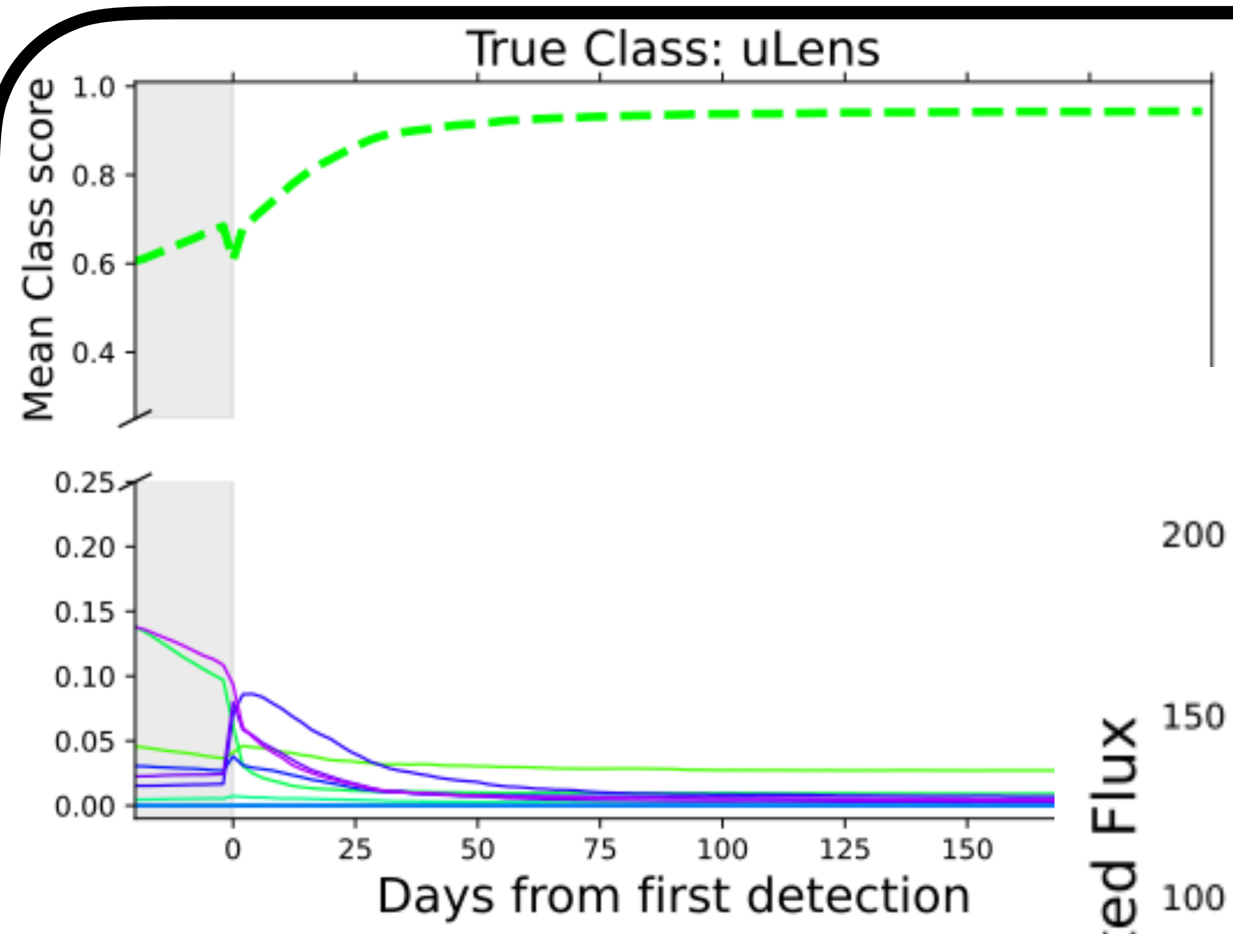


-5 classifications:

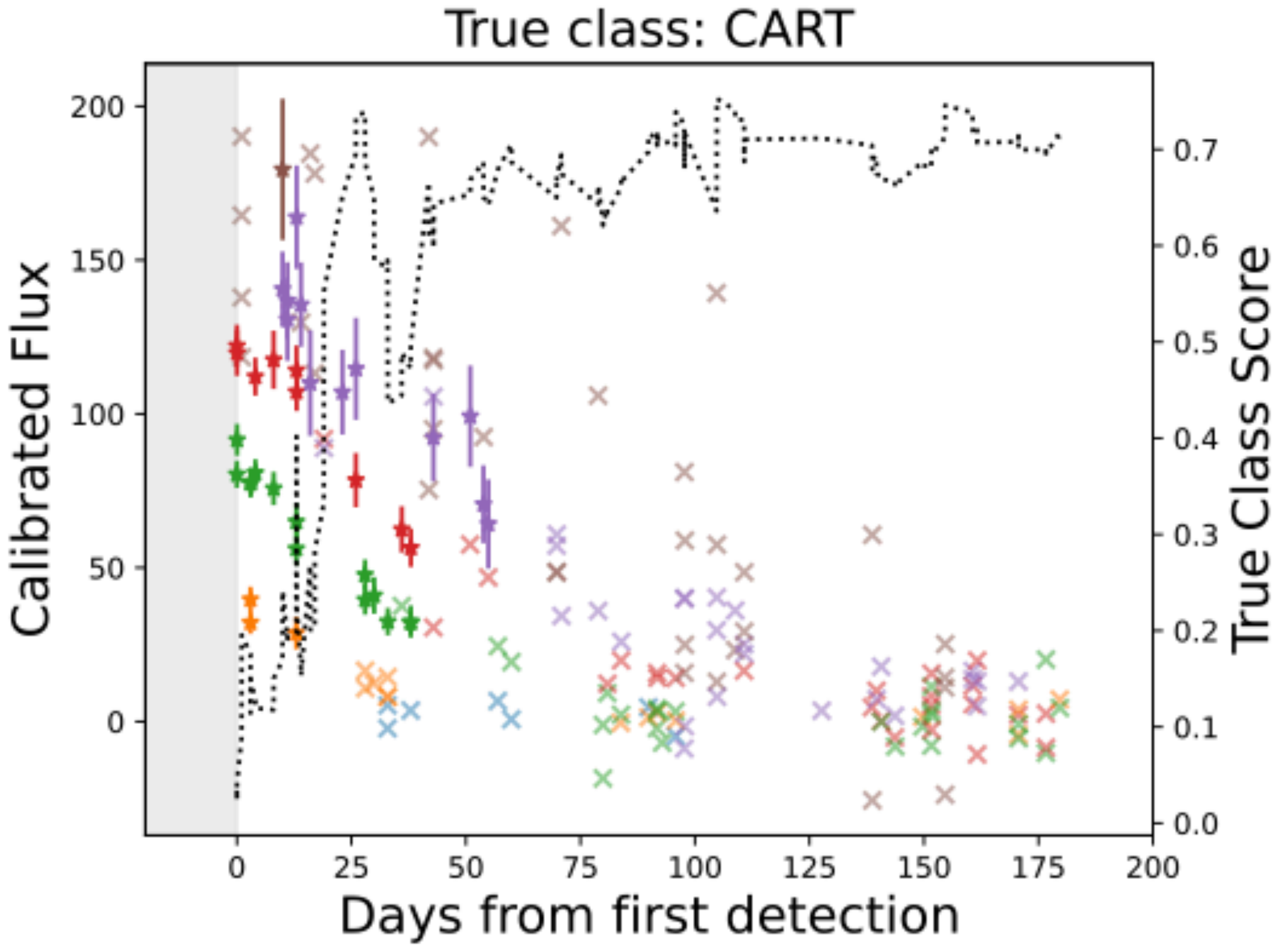
ass	6.3%
oss	4.1%
rest	3.2%
wn	2.8%
hen	1.2%

Classification: question to ask

P(SN Ia | training data, observations)? ELAsTiCC examples

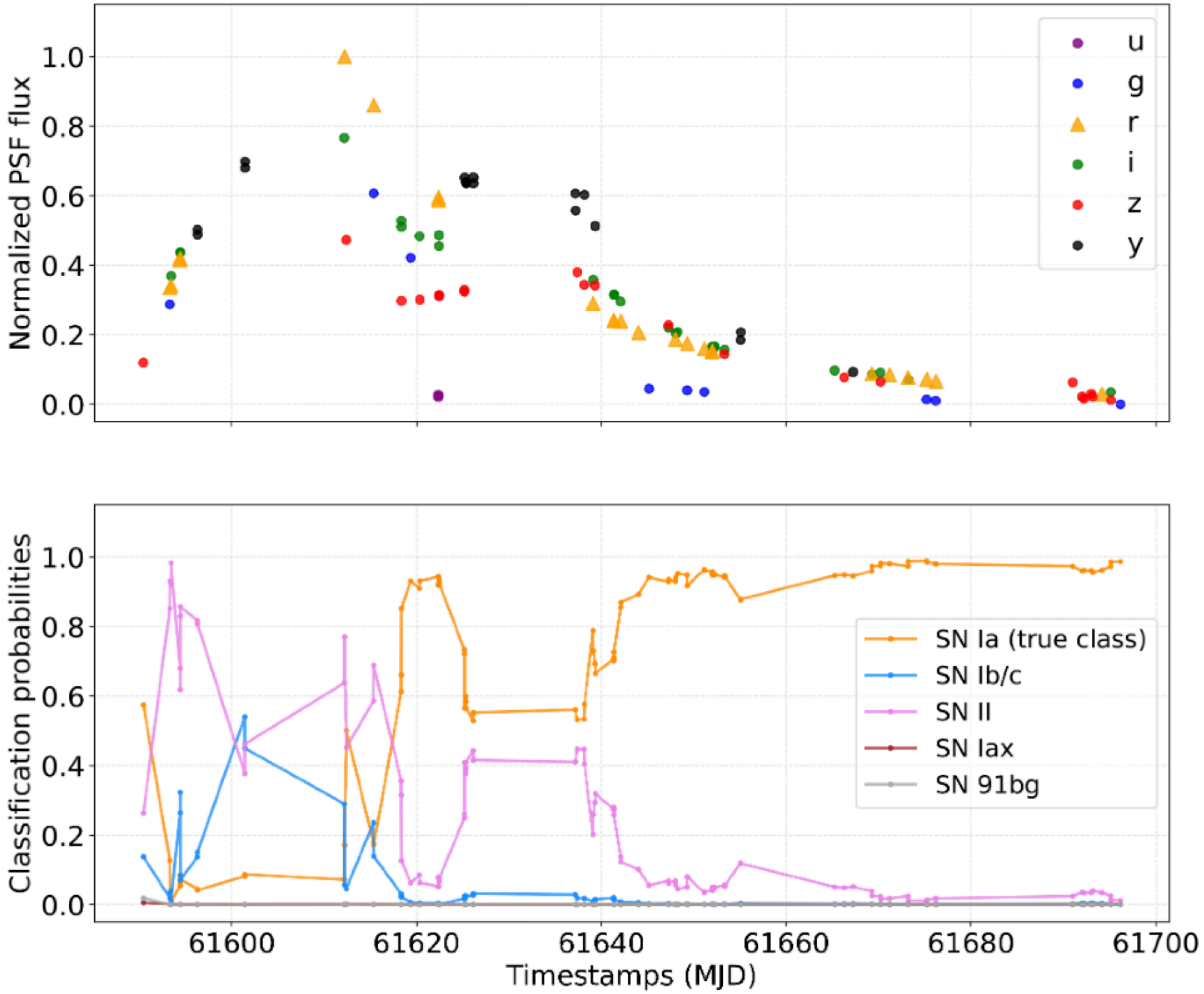


Mean class prob for μ-lens



Ca-Rich SN classification

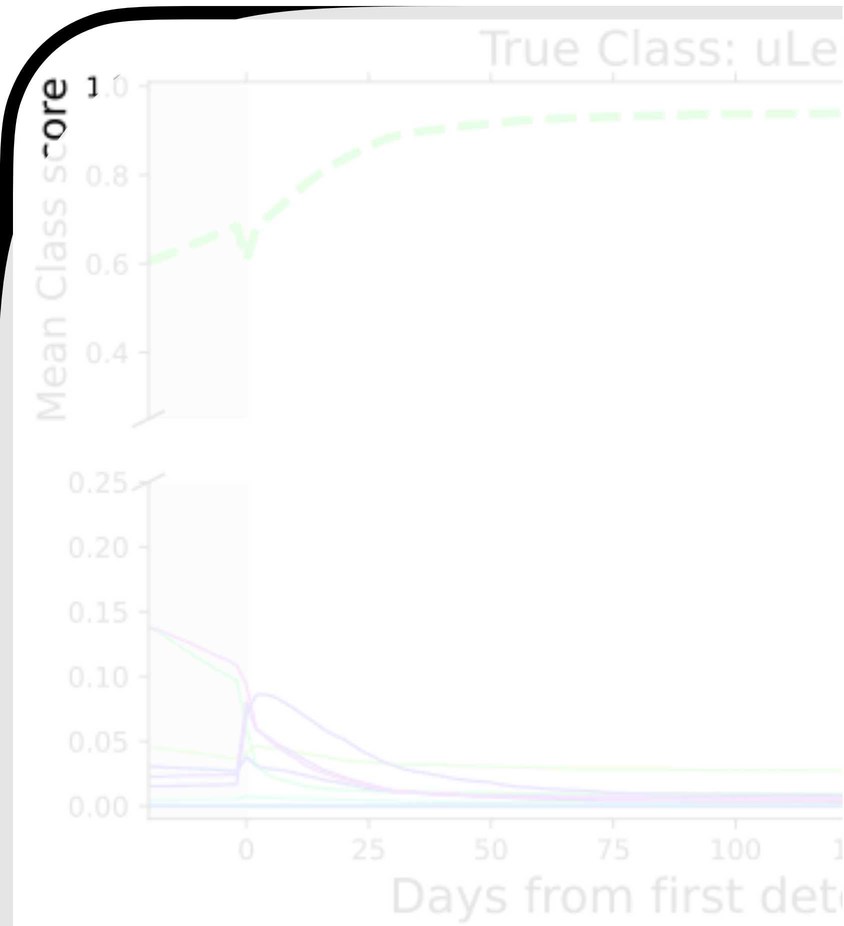
Trained on ELAsTiCC dataset
Shah+25



ELAsTiCC alert stream classification
Zhou+, in prep.

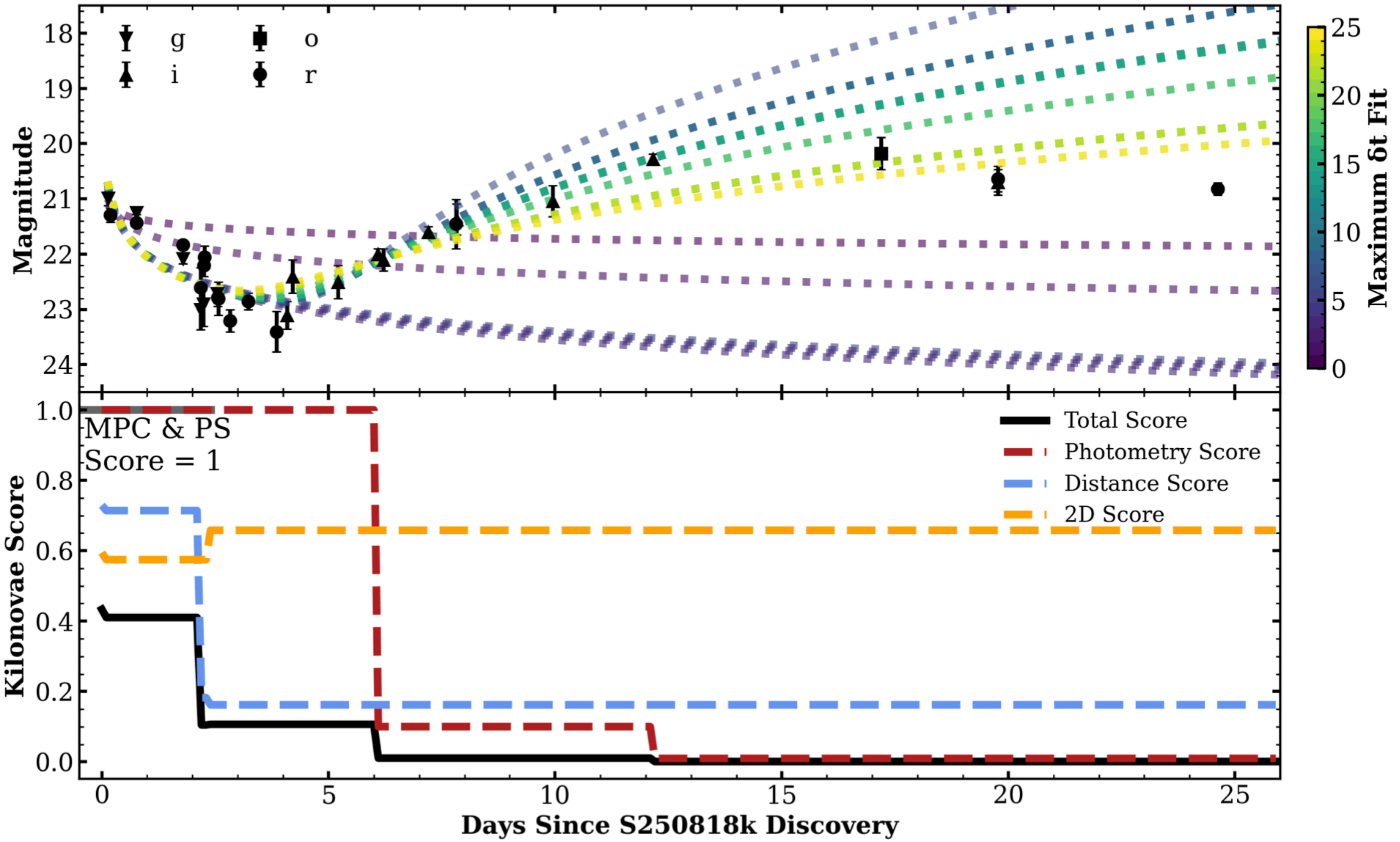
Classification: question to ask

P(SN Ia | training data, observations)? ELAsTiCC examples



Mean class pr
for μ-lens

Traine
Shah+25



ELAsTiCC alert stream classification
Zhou+, in prep.

Franz+25, *see Griffin Hosseinzadeh's talk on Tuesday*

Thank you!

- LSDB: https://lsdb.io
- `light-curve`: https://github.com/light-curve/light-curve-python
- `coniferest`: https://coniferest.snad.space
- SNAD Viewer for ZTF DRs: https://ztf.snad.space