VERA C. RUBIN
OBSERVATORY

# Alert Production Readiness

Eric Bellm
Alert Production Science Lead

Rubin PST | 7 September 2022

NSF AURA U.S. DEPARTMENT OF ENERGY SLAC CHARLES AND LISA SIMONYI FUND FOR ARTS AND SCIENCES LSST CORPORATION

# Is AP ready for commissioning, construction completion, and early science?

| Item | Risk (probability x impact) | | | Comments |
|---|---|---|---|---|
| | Commissioning | Construction Completeness | Early Science | |
| Differencing Algorithms | Medium | Low | Medium | Core performance is solid but still room for improvement (science seeing < template; artifacts) |
| Real/Bogus scoring | Medium | Medium | Medium | Simple models expected to meet requirements but timely (re)training will be needed. |
| Processing Time | Low | **High** | Low | Not yet close to 60 seconds; needs Prompt Processing.  Tradeoffs with other development. |
| Incremental template generation | Medium | Low | **High** | Capability is present, but effort is needed to understand strategy & operationalize. |
| Execution environment* | **High** | Medium | Medium | Prompt Processing system still an early prototype, requires significant work from ARCH + USDF & AP |
| Alert Distribution | Low | Low | Medium | Near-production system needs to be migrated to the USDF and tested at scale.  AVS requires development. |

# I'm proposing the following course of action.

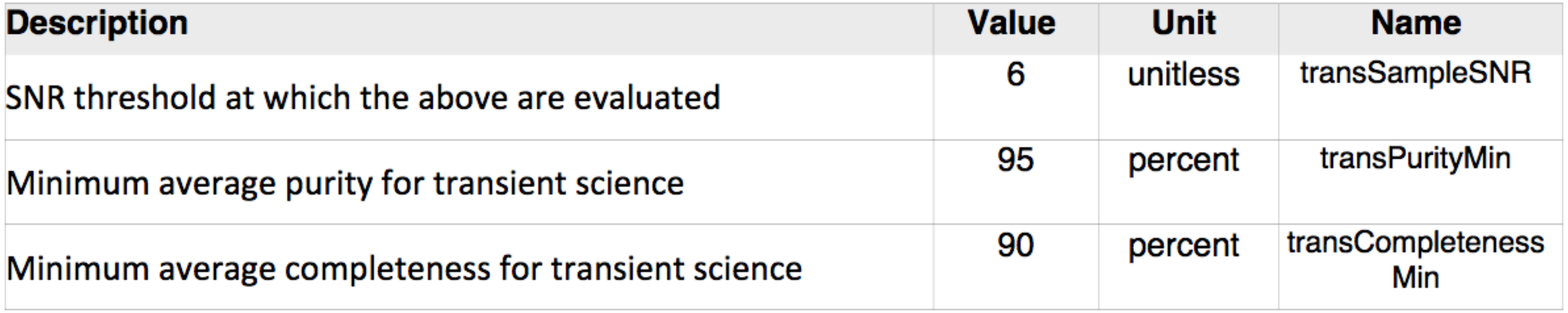
**Execution environment:** immediately after USDF migration stabilizes, begin major cross-DM effort to stand up Prompt Processing at USDF. *immediately*
**Incremental Template Generation:** rapidly confirm alignment on broad strategy; simulate and study detailed selection criteria in parallel with operational tooling development *first half 2023*
**Processing Time:** continue optimization efforts, but target 120 second latency until incremental template generation and routine AP are well-established. *ongoing*

**Differencing Algorithms** and **Real/Bogus:** continue steady algorithmic testing and refinement through commissioning and operations *ongoing*
**Alert Distribution:** stand up production systems and use continuous integration to maintain operational readiness *fall 2022*

# Image Differencing Algorithms

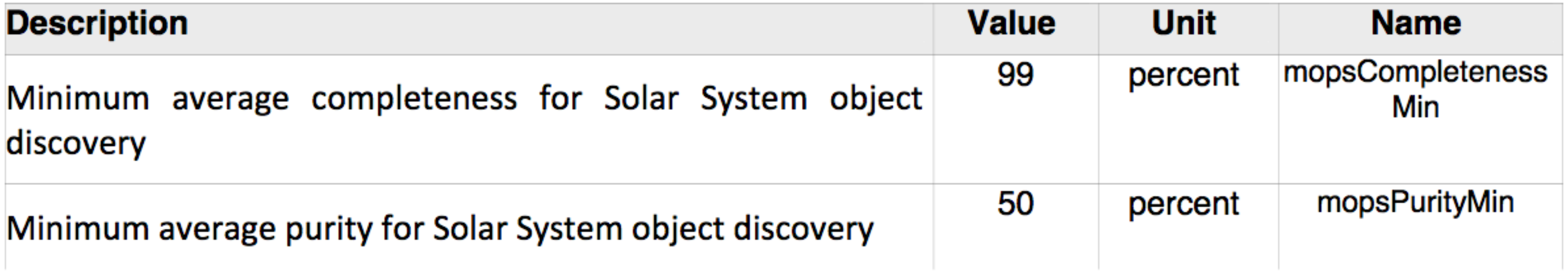
# AP has high-level requirements on alert completeness and purity.

## OSS-REQ-0353: Difference Source Spuriousness Threshold - Transients



| Description | Value | Unit | Name |
|---|---|---|---|
| SNR threshold at which the above are evaluated | 6 | unitless | transSampleSNR |
| Minimum average purity for transient science | 95 | percent | transPurityMin |
| Minimum average completeness for transient science | 90 | percent | transCompleteness Min |

## OSS-REQ-0354: Difference Source Spuriousness Threshold - MOPS



| Description | Value | Unit | Name |
|---|---|---|---|
| Minimum average completeness for Solar System object discovery | 99 | percent | mopsCompleteness Min |
| Minimum average purity for Solar System object discovery | 50 | percent | mopsPurityMin |

# For ground-based surveys we have to account for PSFs that change from image to image.

One option is to PSF-match the images by computing an appropriate convolution kernel in Fourier space, degrading to the resolution of the worse-seeing image (e.g., Ciardullo+90, Phillips & Davis 95).

$$I'_2(k) = I_2(k)\frac{\phi_1(k)}{\phi_2(k)}$$

$$A_1 - A_2 = \frac{\sum_i \left(I_{1,i} - I'_{2,i}\right)\phi_i}{\sum_i \phi_i^2}$$

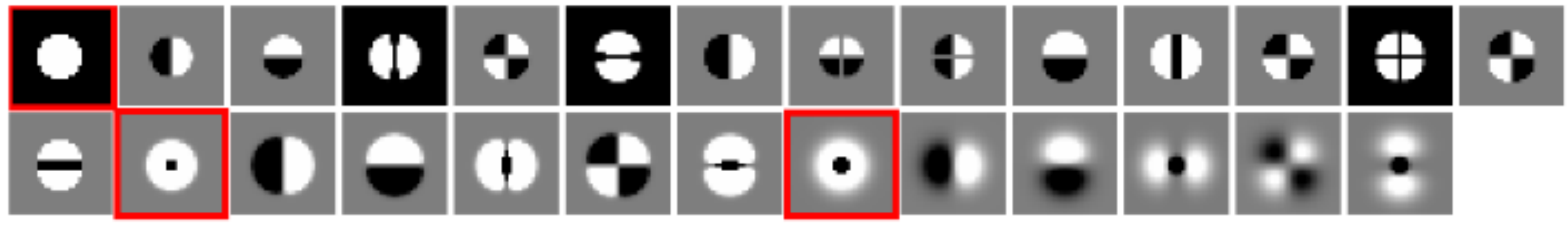
But this requires very good knowledge of the PSF, which may be difficult in crowded fields, and care to avoid numerical instability in the division.

# Alard & Lupton 1998 model the convolution kernel as a series of basis functions.

Defining the convolution kernel as:

$$I'_2 = \kappa \otimes I_2 \qquad \text{where} \qquad \kappa = \sum_r a_r B^r$$

and Br are a set of basis functions



LDM-227

in the background-dominated regime we can use least-squares fitting to determine the kernel basis function coefficients.

$$\left| \frac{I_1 - \sum_r a_r \left(B^r \otimes I_2\right)}{\sigma} \right|^2$$

This works nicely if the template is noise-free and the science image has better seeing than the template.

It also allows for spatial variations in the PSF as well as background matching.

# Zackay, Ofek, and Gal-Yam (2016) recognized that classical A&L is not optimal if the template is noisy.

They proposed an alternative algorithm (now widely called “ZOGY”) in Fourier space:

$$\widehat{D} = \frac{F_r \widehat{P_r} \widehat{N} - F_n \widehat{P_n} \widehat{R}}{\sqrt{\sigma_n^2 F_r^2 |\widehat{P_r}|^2 + \sigma_r^2 F_n^2 |\widehat{P_n}|^2}}$$

- fully symmetric between the science and template images
- maximizes the SNR of detected point sources when both images are noisy
- requires knowledge of the PSFs (or their ratio)

# The ZOGY noise-whitening approach can also be applied within the A&L framework (Reiss & Lupton 16).

See https://dmtn-021.lsst.io/

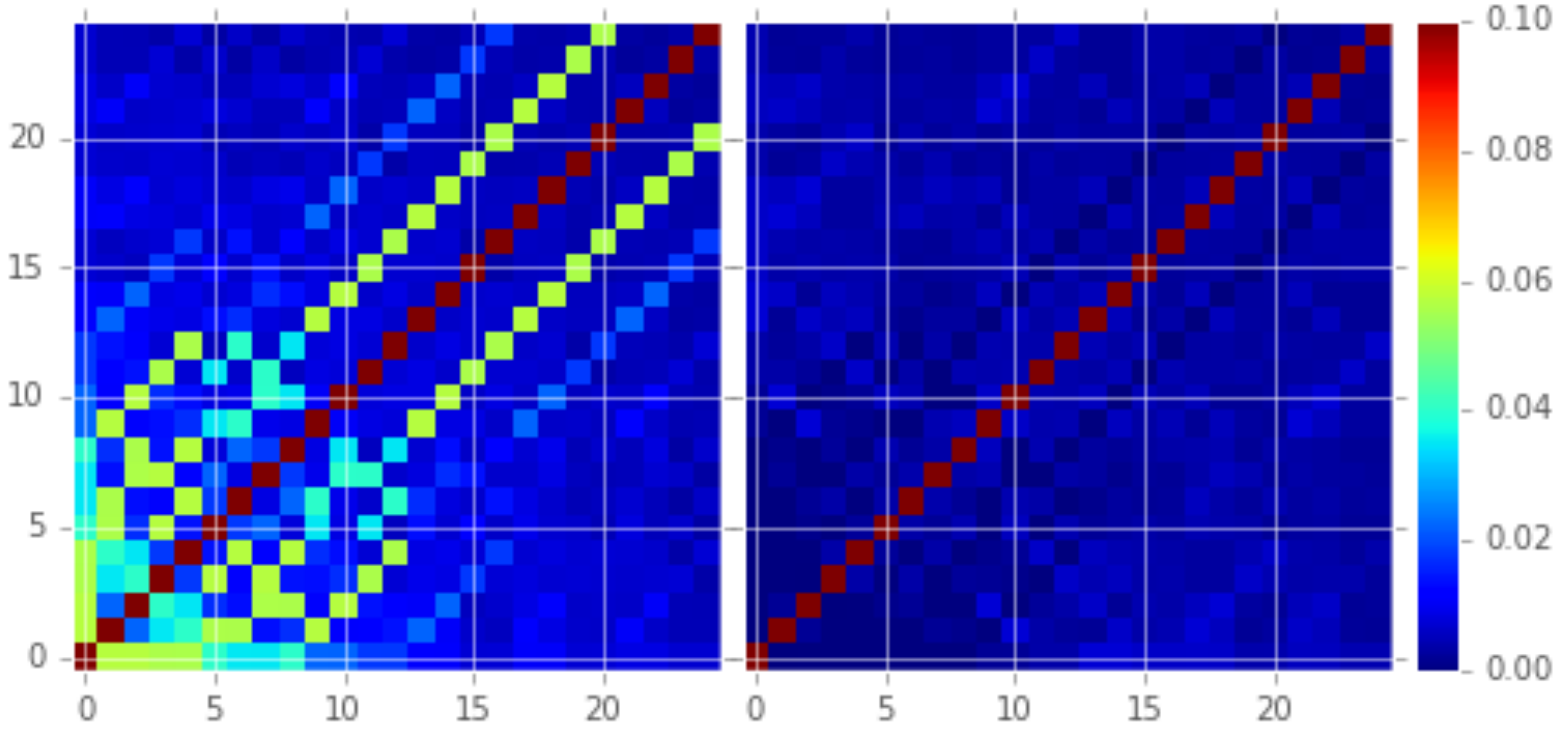
Rather than use the ratio known PSFs, retain the use of basis functions, but whiten the noise in k-space:

$$D(k) = \left[I_1(k) - \kappa(k) I_2(k)\right] \sqrt{\frac{\overline{\sigma}_1^{-2} + \overline{\sigma}_2^{-2}}{\overline{\sigma}_1^{-2} + \kappa^2(k)\overline{\sigma}_2^{-2}}}$$

which is identical up to normalizations to the ZOGY expression when

$$\kappa(k) = \phi_1(k)/\phi_2(k)$$

- avoids need to know the PSFs directly
- removes noise covariance as expected



# Decorrelated A&L and ZOGY are both optimal for background-limited detection but have different practical tradeoffs.

### ZOGY

+ fully symmetric between images
+ handles misaligned elliptical PSFs

- requires knowledge of the PSFs (or ratio)—concern for crowded fields
- Fourier space solution sensitive to gaps, edges, interpolation, zeros
- more challenging to handle spatial PSF variations

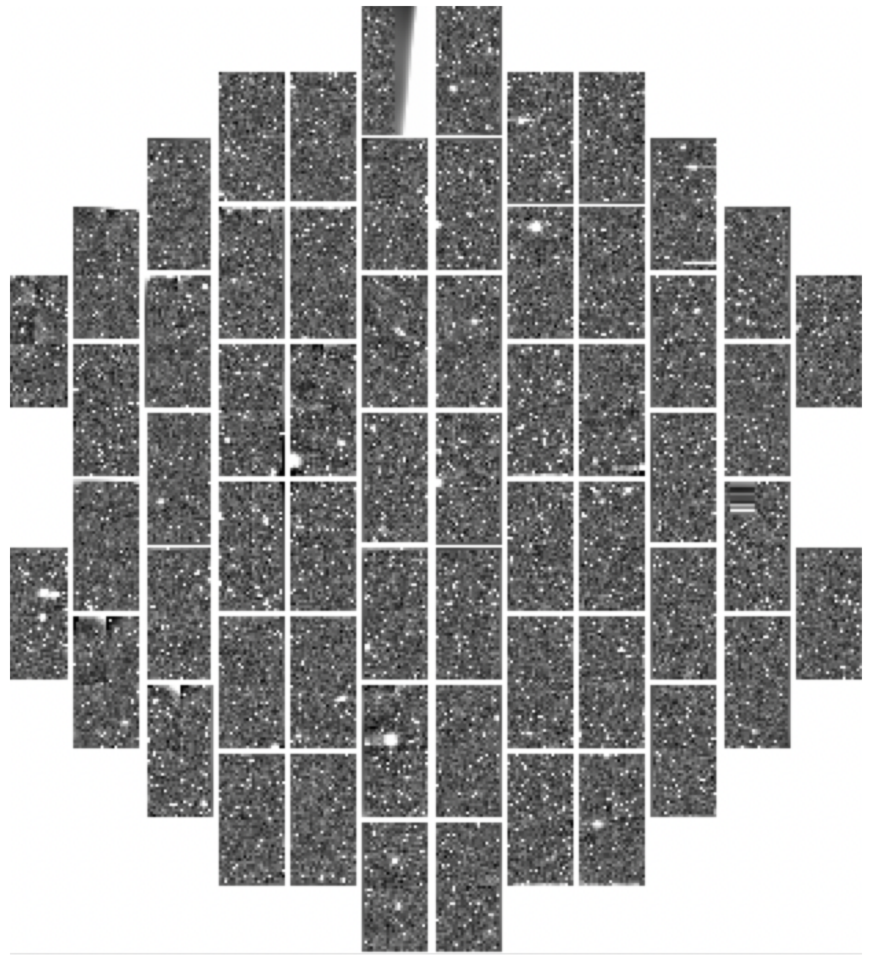
### Decorrelated A&L

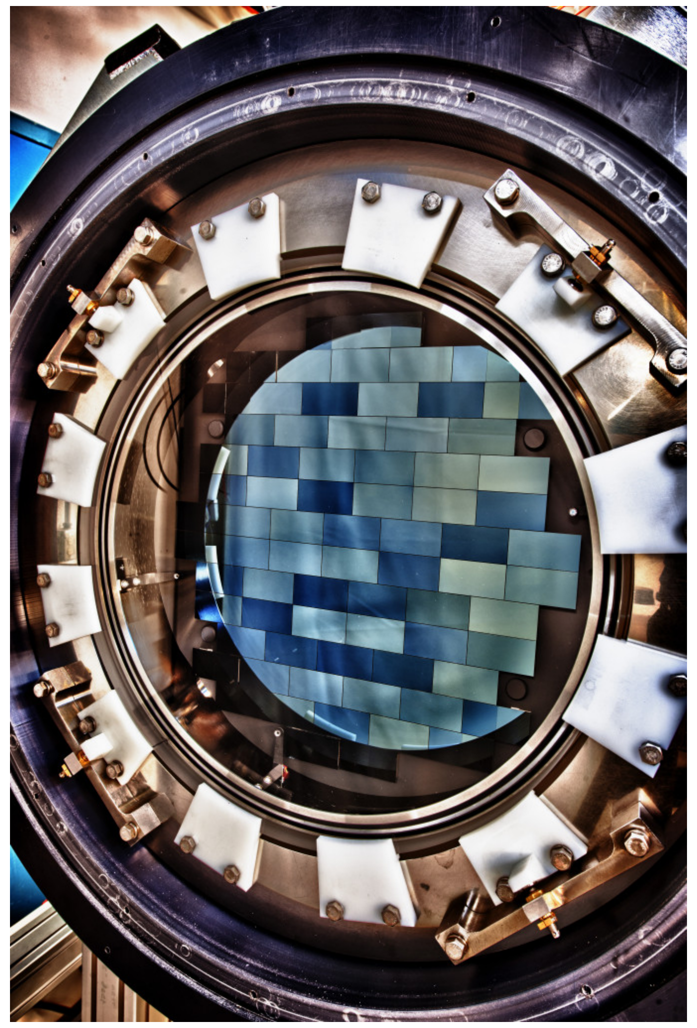
+ image-space solution handles spatial PSF variations, missing data, etc.
+ does not require PSF knowledge

- requires special treatment when science seeing is better than the template
- choice of kernel basis function introduces degrees of freedom

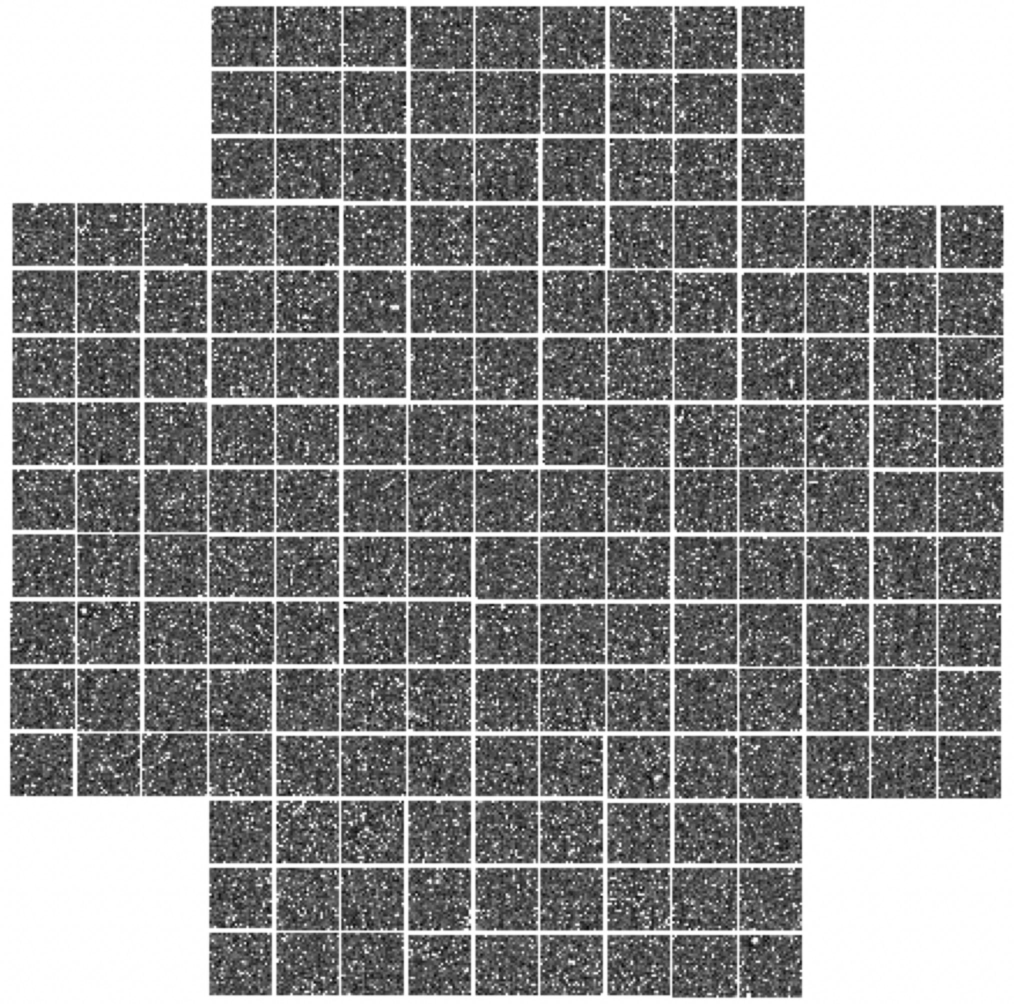
To date we have elected to focus most of our algorithmic development effort on Decorrelated A&L.

# We test against several real and simulated precursor datasets on daily and monthly timescales.

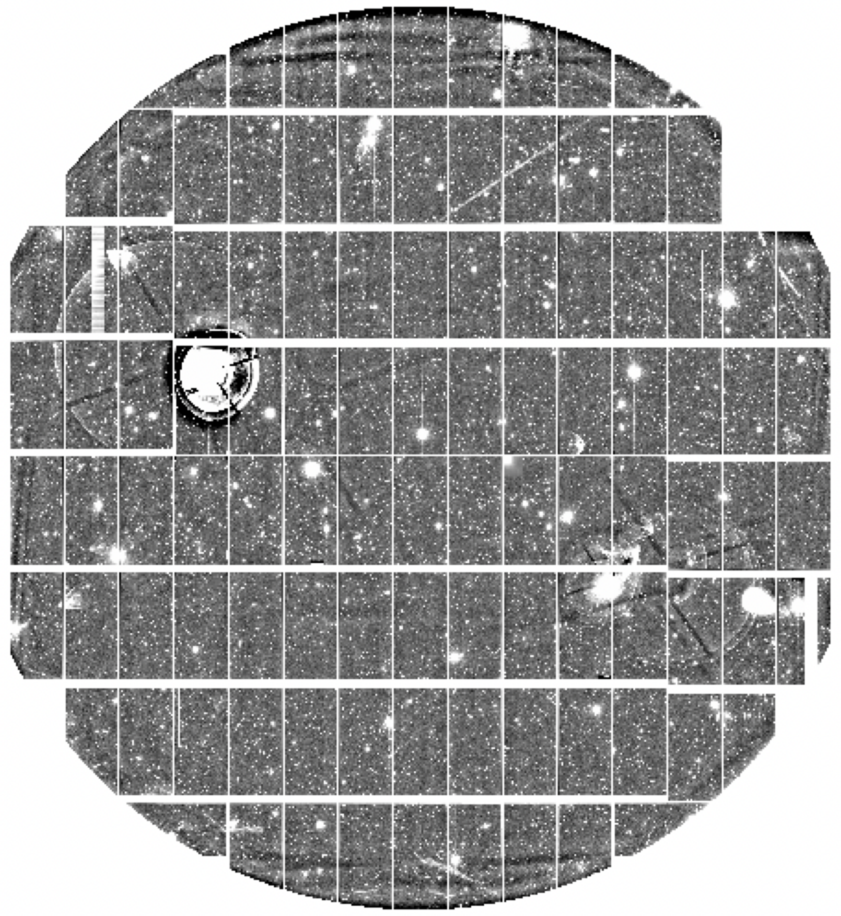


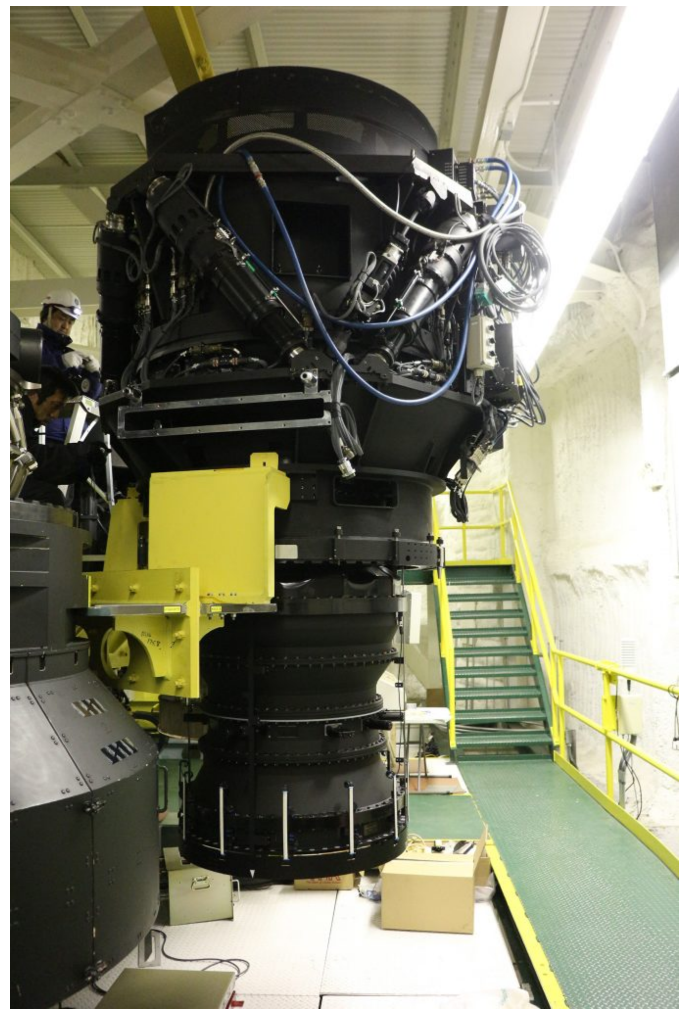


**Dark Energy Camera (DECam):**
HiTS
Saha Bulge
(DECAT)

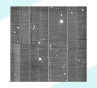


**DESC DC2**
image
simulations





**Hyper SuprimeCam (HSC):**
COSMOS

*(soon)* **AuxTel LATISS** 

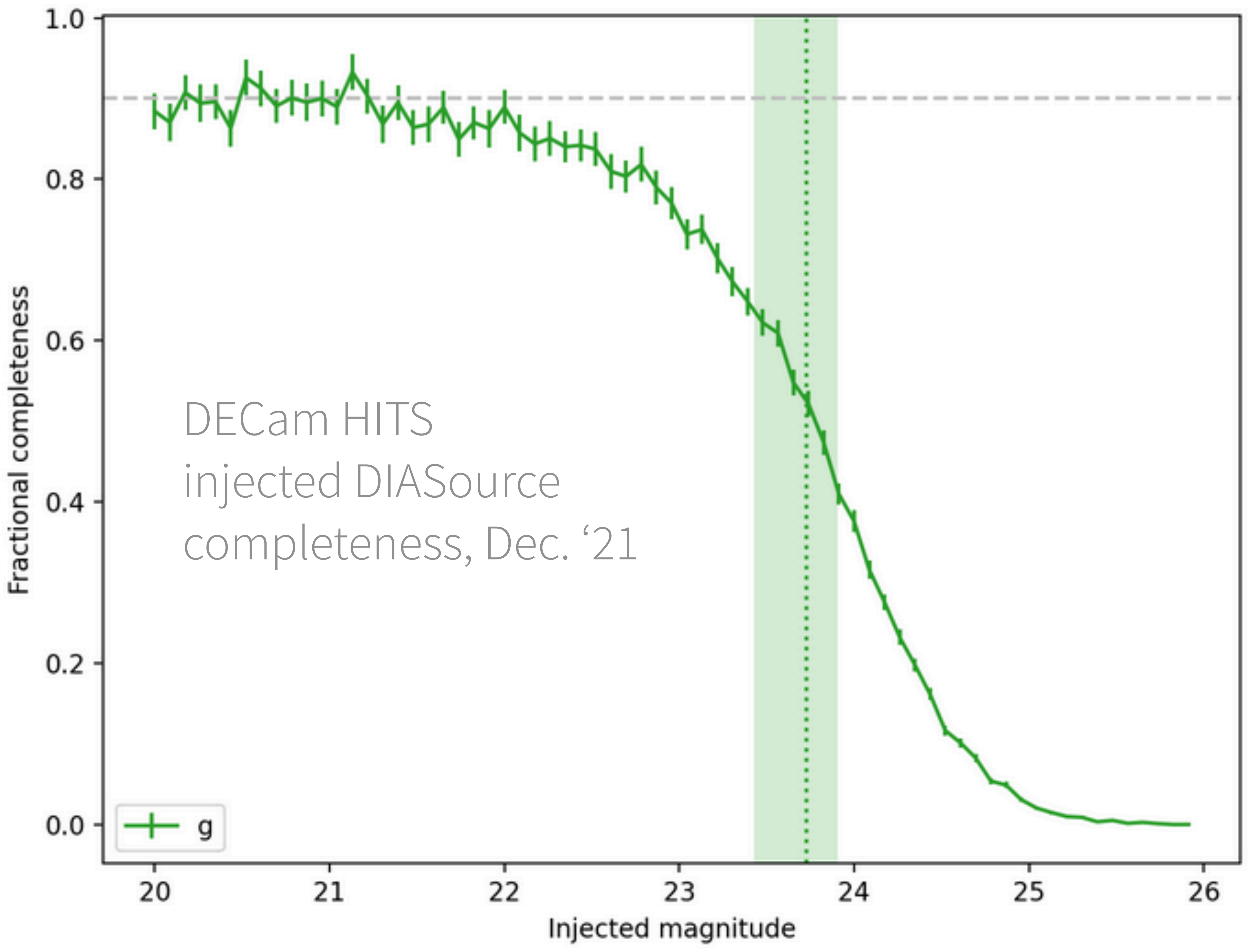
Groups outside AP are successfully using Rubin diffim: Smotherman+21; Smotherman+, Bernardinelli+, Stetzler+in prep; Moore+ in prep

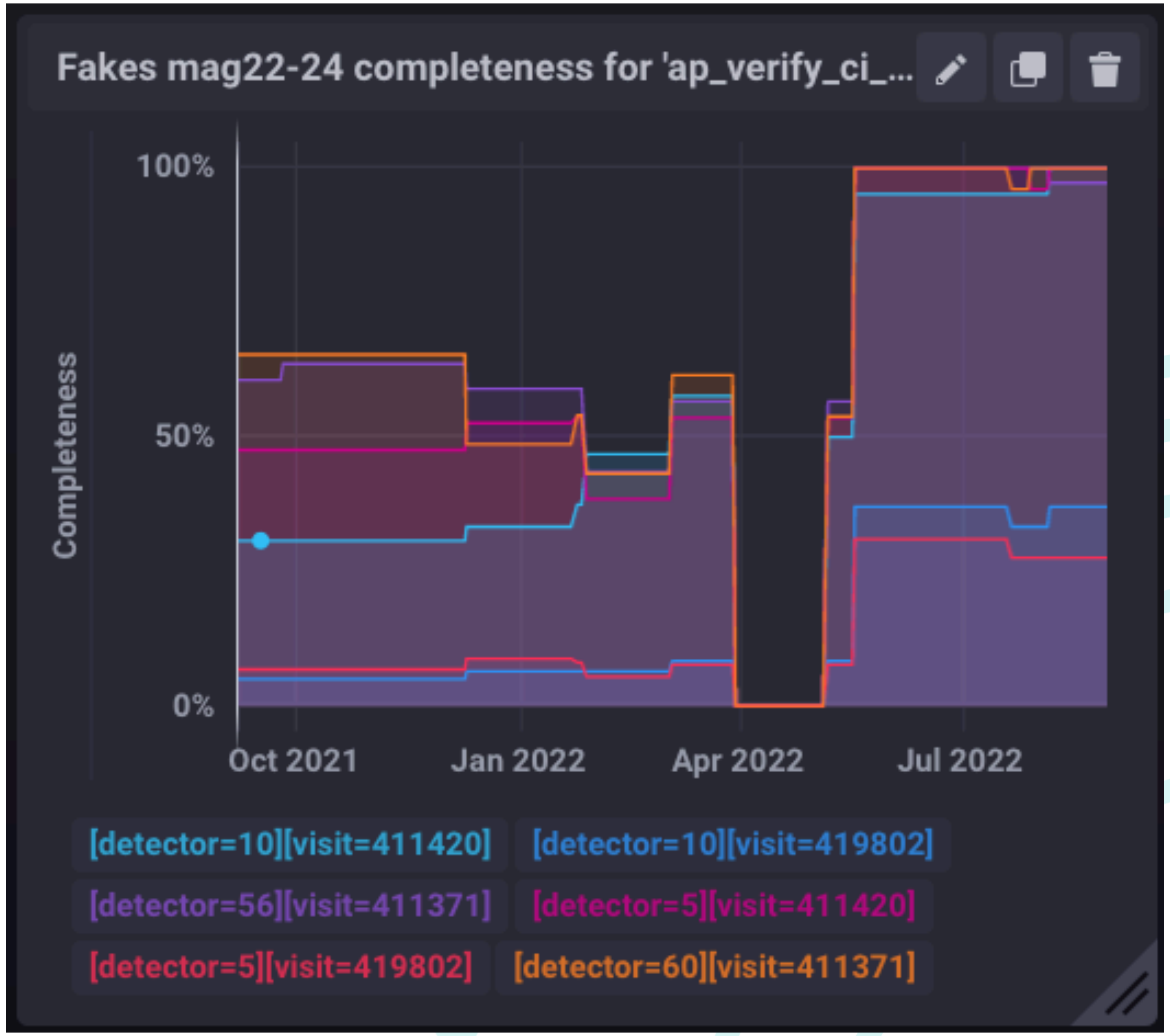
# (The Data Facilities migration prevented a fully updated report of pipelines performance.)

We completed large-scale HSC COSMOS and DECam HiTS reprocessings with fake injection at the end of July at NCSA

Due to the migration to the USDF at SLAC the data have been unavailable to us since then, so the following performance plots are taken from older presentations and our small continuous integration datasets.

# Fake source injection allows us to assess completeness.

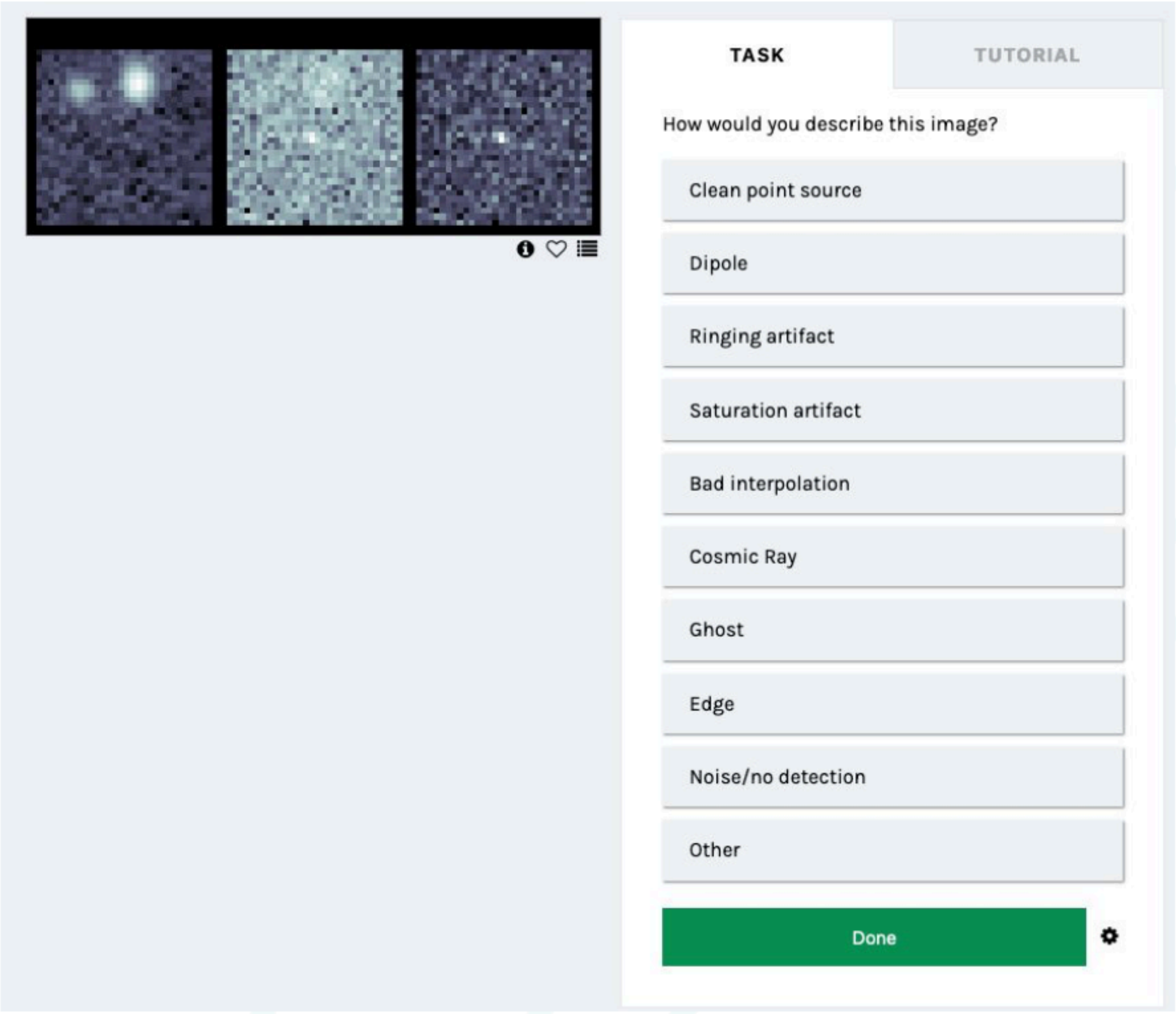



Fakes mag22-24 completeness for 'ap_verify_ci_...

100%
50%
0%
Completeness
Oct 2021
Jan 2022
Apr 2022
Jul 2022
[detector=10][visit=411420]
[detector=10][visit=419802]
[detector=56][visit=411371]
[detector=5][visit=411420]
[detector=5][visit=419802]
[detector=60][visit=411371]

Bugfixes and other improvements have steadily increased our completeness.

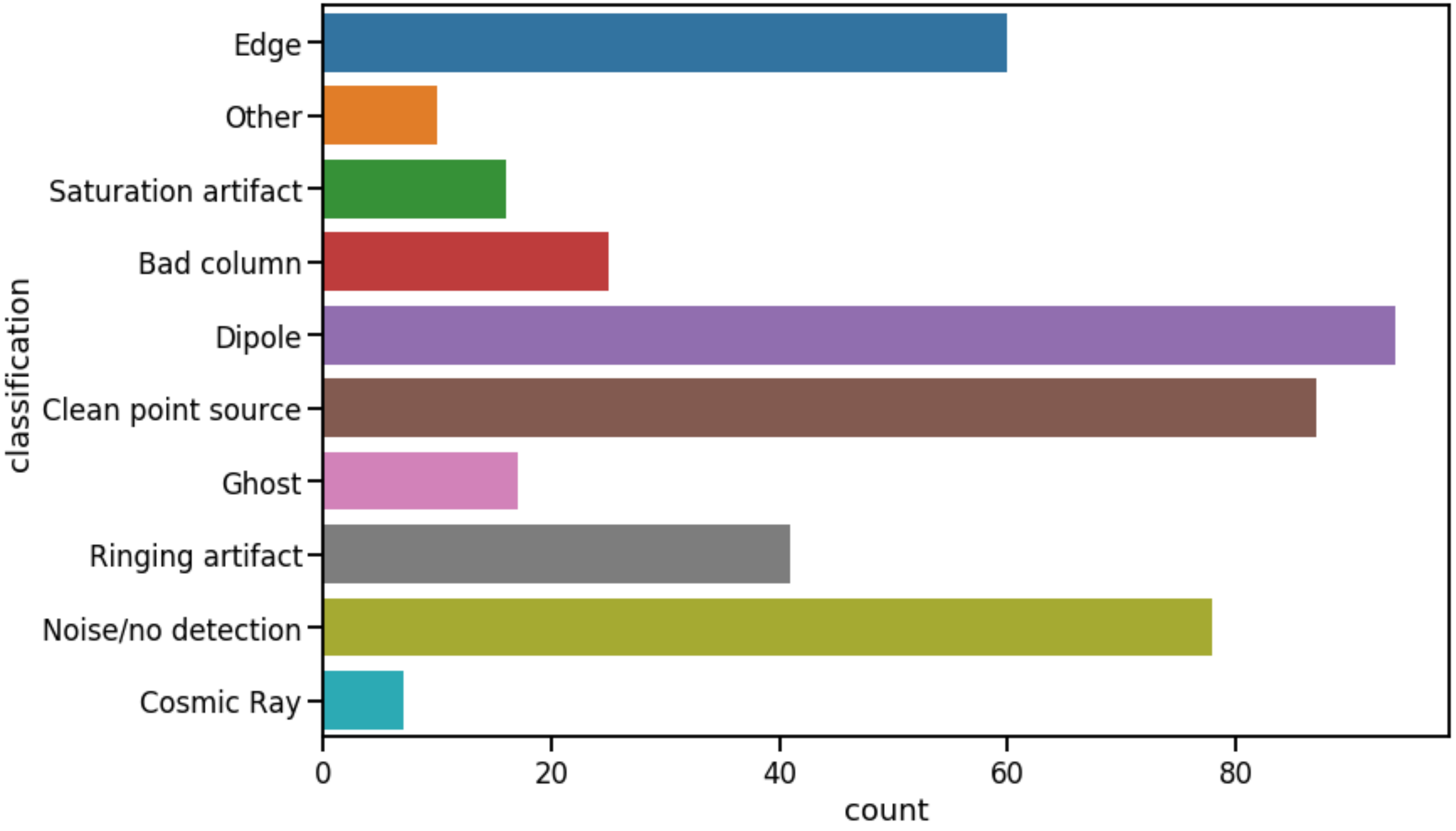
# Human labelling allows us to assess purity.



TASK

TUTORIAL

How would you describe this image?



DECam HITS artifacts, March 2020

# Rubin pipelines provide purer raw outputs than other major surveys.

PS1 (A&L):

- raw artifact/non-artifact ratio of 10:1-50+:1 (Denneau+13)

DES Y1 (A&L; HOTPANTS):

- raw artifact/non-artifact ratio of 13:1 before ML (Goldstein+15)

ZTF (ZOGY):

- raw artifact/non-artifact rate between 2.5:1 and 25:1

Science Pipelines (decorrelated A&L):

- **HiTS (2020): raw artifact/non-artifact rate ~4:1 (no filtering), ~3:1 (coarse flag filtering, minor loss of completeness)**

# We are making progress on the "better science seeing" case for decorrelated A&L.

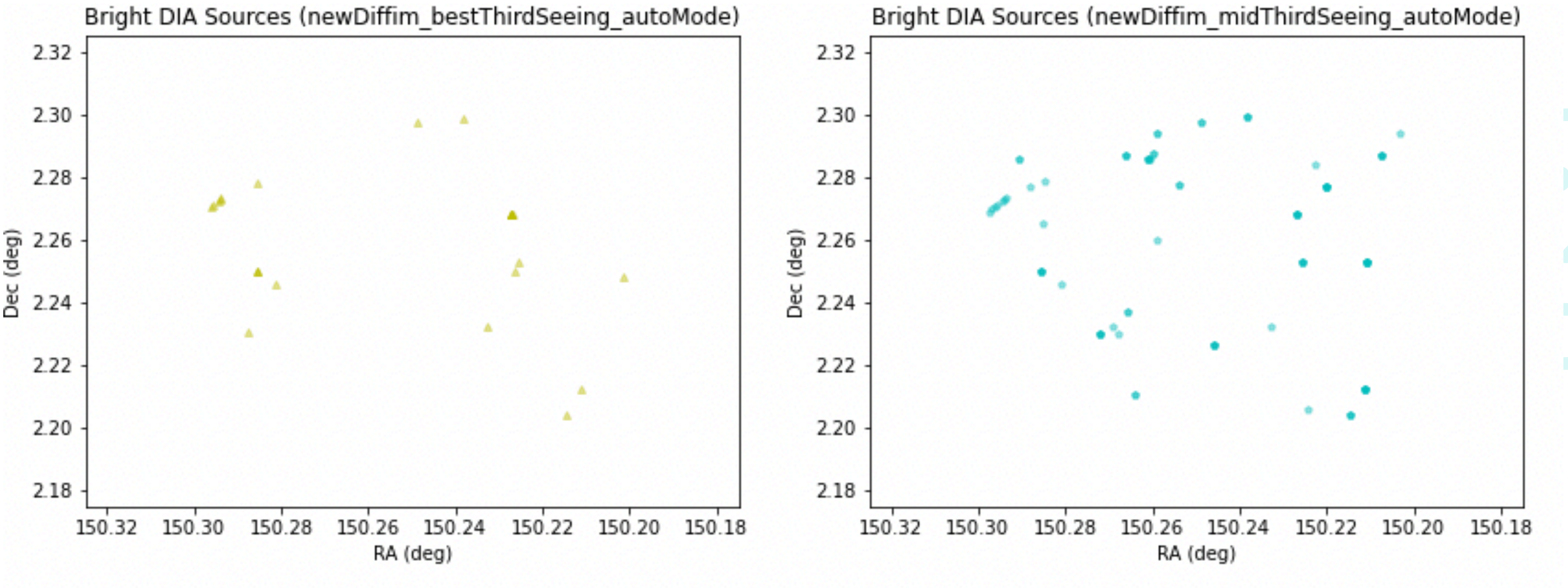
ZOGY is fully symmetric between images; decorrelated A&L is not

Practically: can we just swap the science and template when the science seeing is better? ("convolve science"). Maybe!

Other option: preconvolution, although combining with decorrelation & measurement requires more algorithmic work

Refactored difference imaging supports adjusting algorithm on the fly based on input seeing

# Early results for convolving the science image show some promise.



# In 2022 we refactored the image differencing pipeline.

The new design is more modular, better tested, and leverages pure Gen 3 capabilities.

Instead of `lsst.pipe.tasks.ImageDifferenceTask`, the image differencing pipeline now consists of three steps:

- `lsst.ip.diffim.GetTemplateTask` which constructs the warped and cropped (and possibly DCR-corrected) template exposure.
- `lsst.ip.diffim.AlardLuptonSubtractTask` which PSF-matches either the template or science image, and subtracts the template from the science image.
- `lsst.ip.diffim.DetectAndMeasureTask` which runs source detection and measurement on the difference image, including adding sky sources and forced measurement.

Slide & refactor by Ian Sullivan

# The refactored difference imaging shows comparable performance; some further work remains.

Several bugs found and fixed

Much more complete test coverage

Detection & measurement on likelihood images remains to be implemented (needed for preconvolved A&L, ZOGY)

# New AP hire Bruno Sánchez will strengthen our algorithms expertise beginning (we hope) in December.

Dr. Bruno Sánchez

DESC member with extensive experience with A&L & ZOGY image differencing both in Rubin pipelines and elswhere

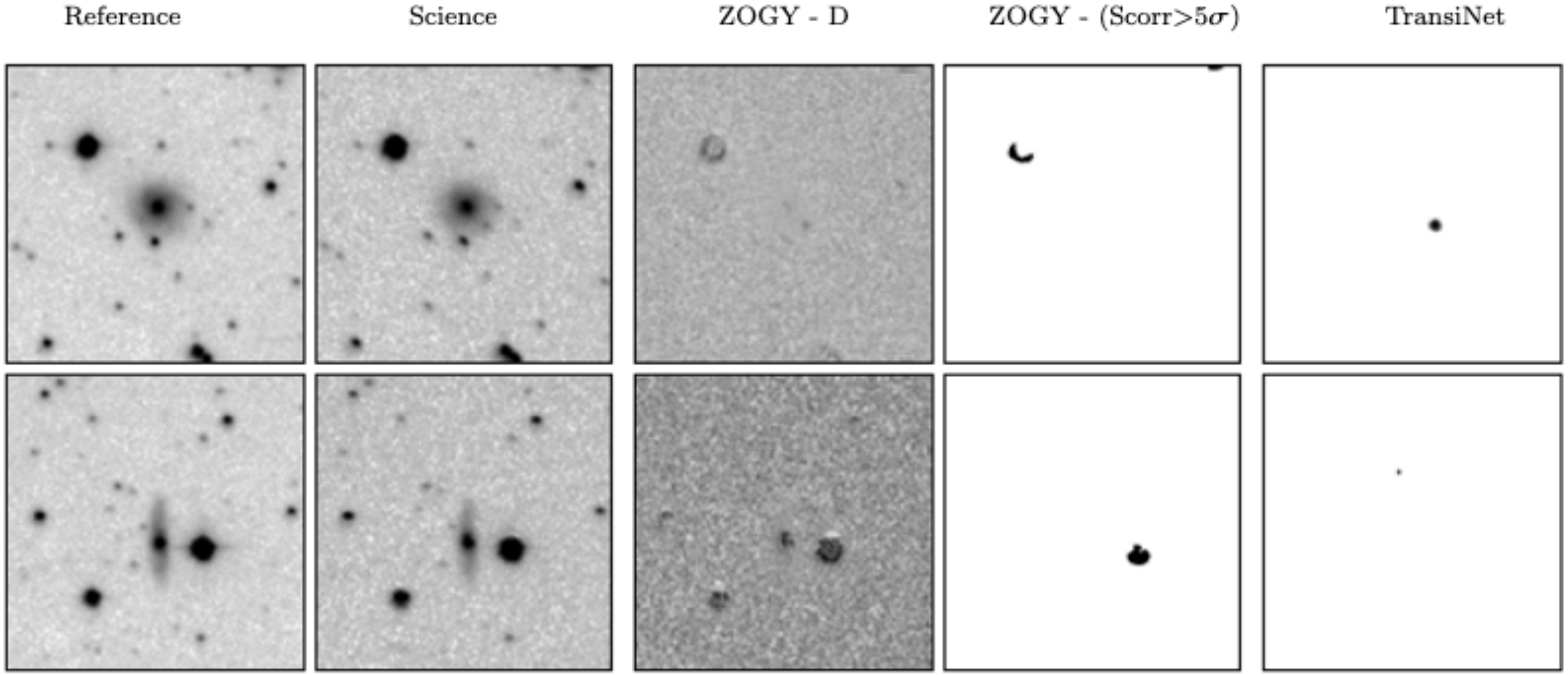
- e.g., Sánchez+22, Sánchez+19

Expect to have him drive forward the likelihood-based preconvolution & ZOGY approaches in the refactored difference imaging, evaluate algorithmic trade space.

# Real/Bogus Scoring

# Machine-learned spuriousness (Real/Bogus) development is in progress.

Dr. Nima Sedaghat

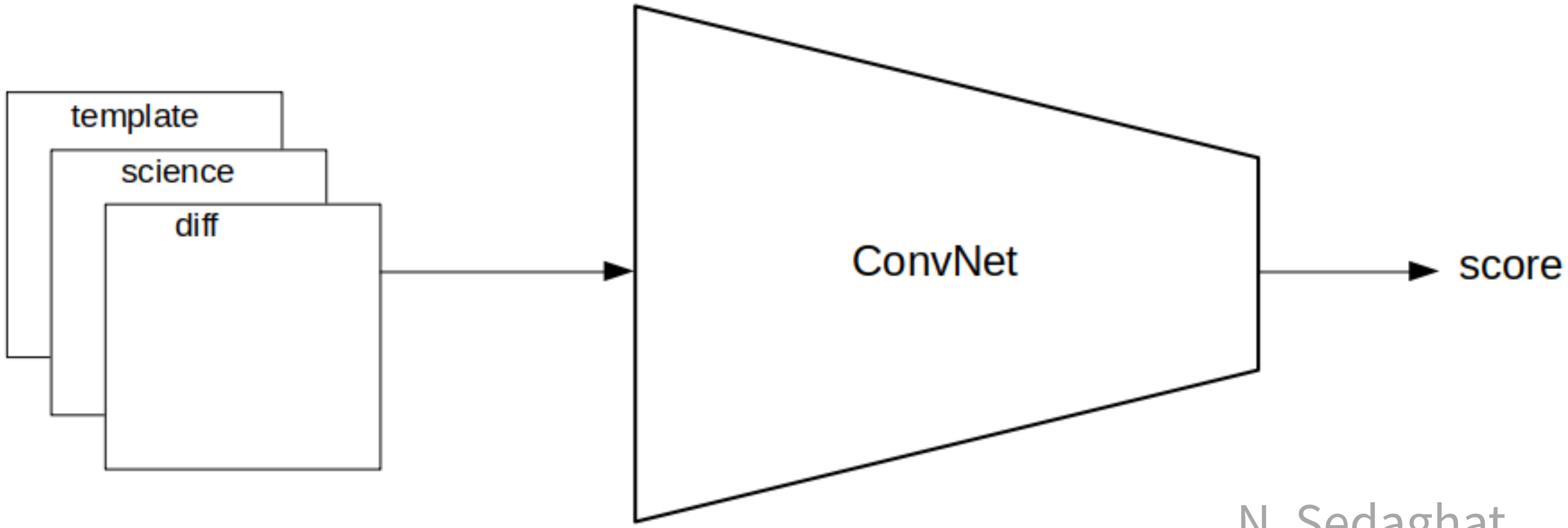


Sedaghat & Mahabal 2018

# We have implemented a traditional Real/Bogus classifier.

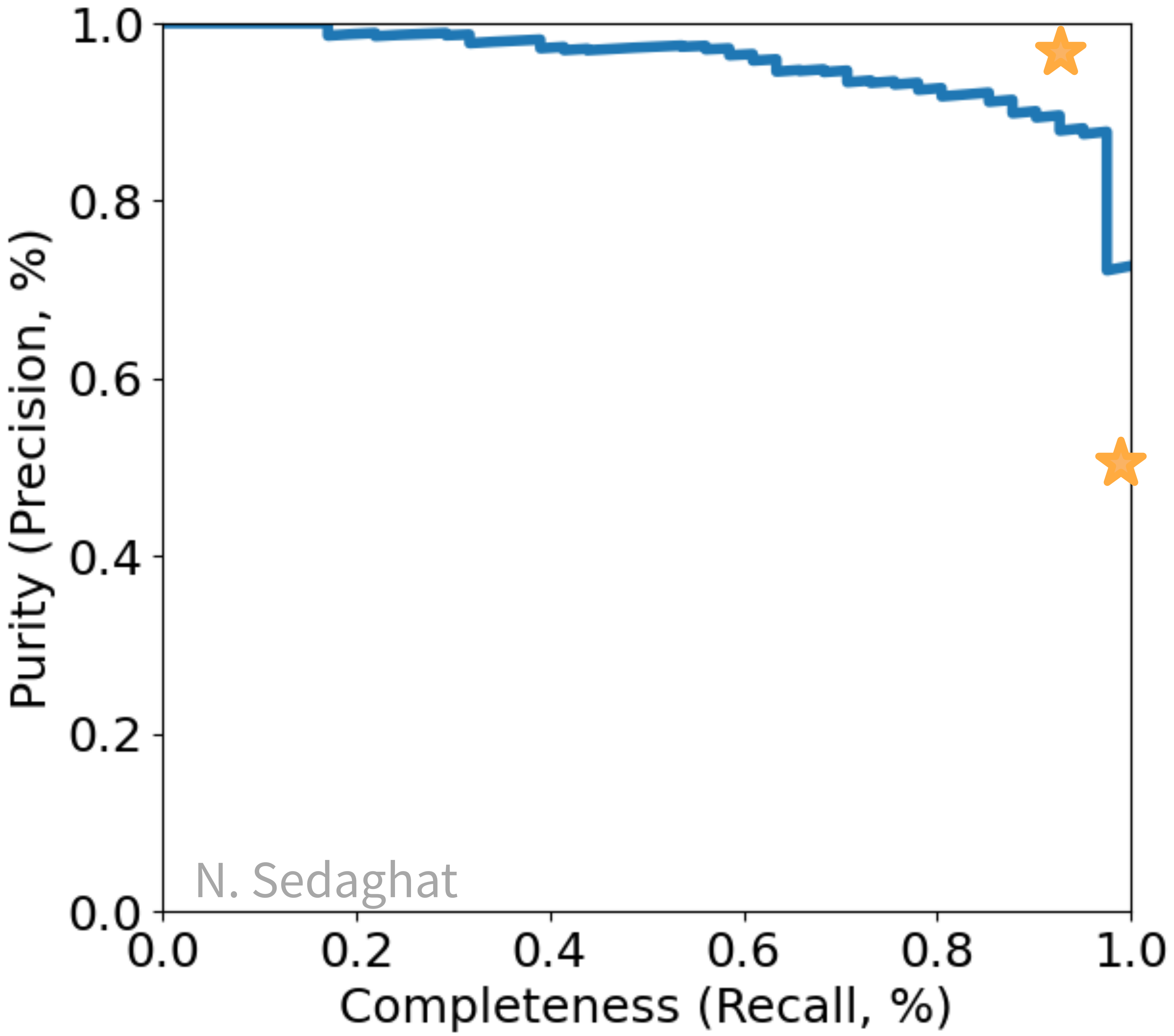
Supervised binary discriminative classifier that runs on image differencing outputs (cf. Bailey+07, Bloom+12, Goldstein+15, Duev+19...).

Improves alert purity but not completeness above the image differencing threshold.



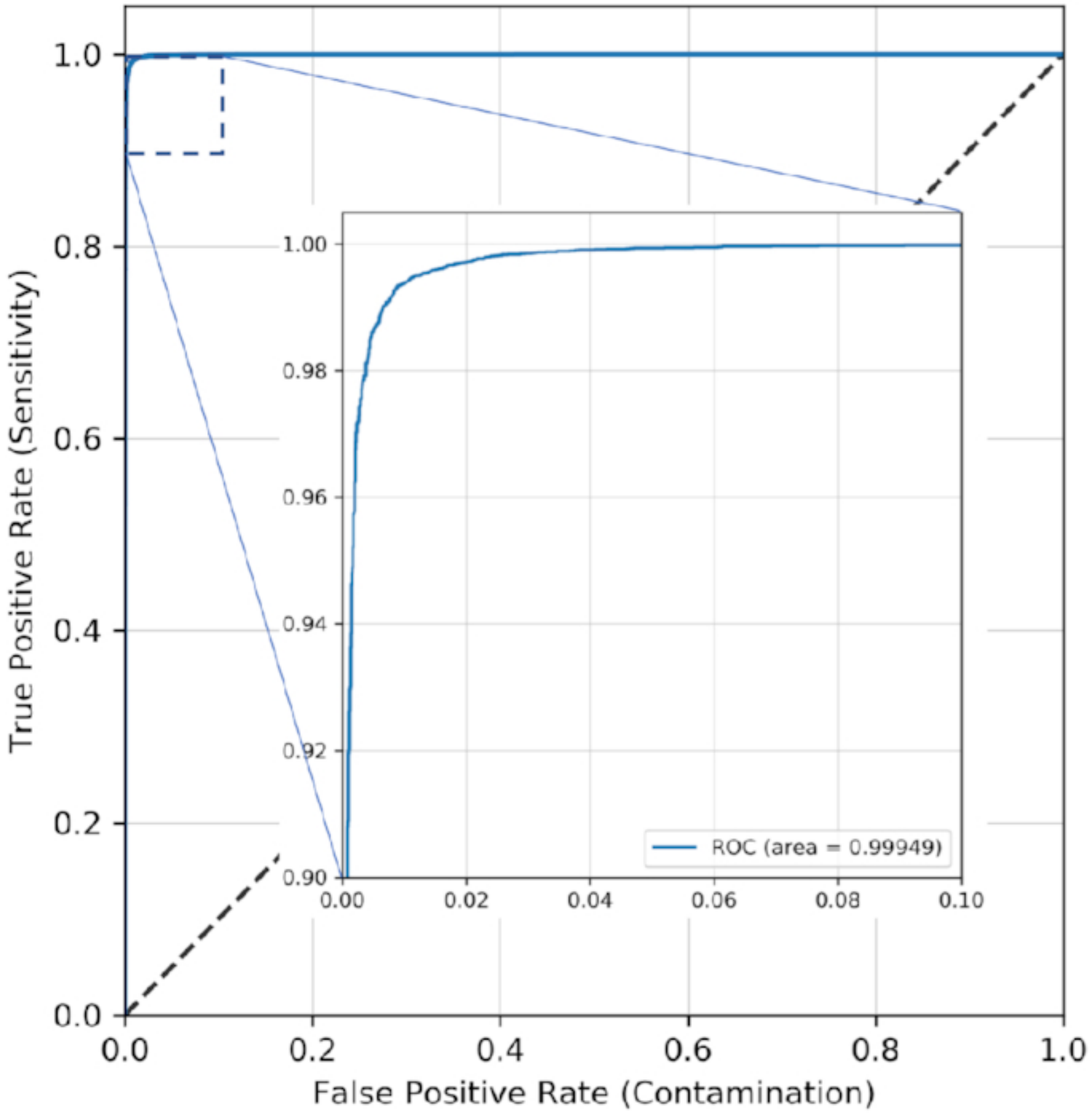
N. Sedaghat

# We are performing initial training on DC2 simulations.



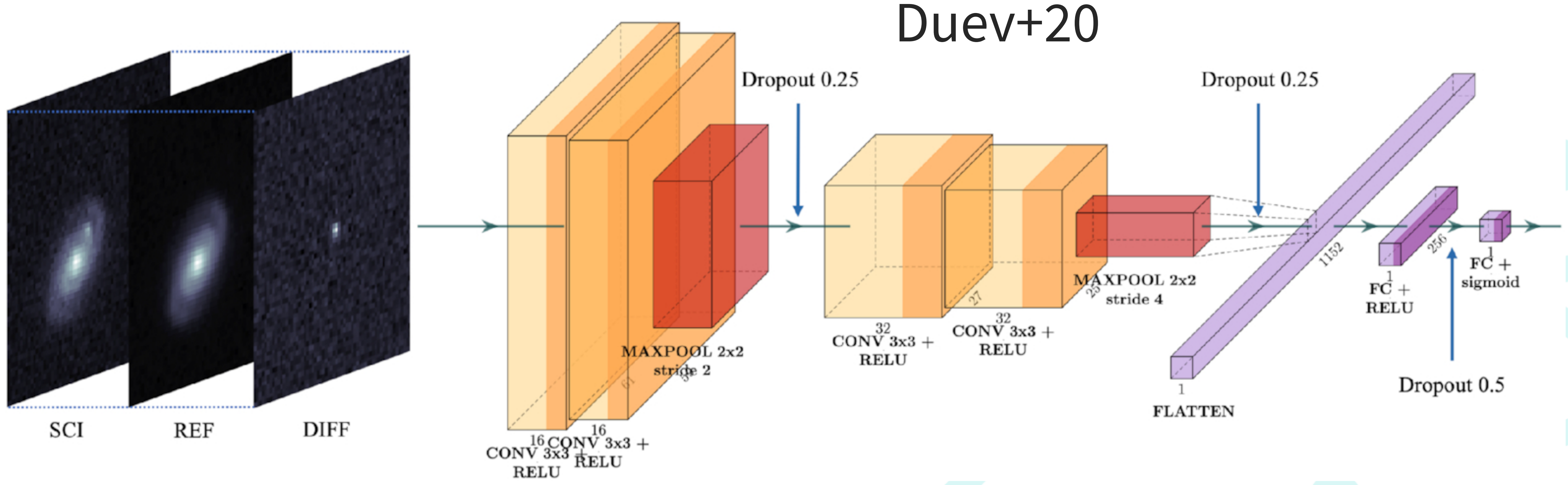
nb.: 5 sigma DIASources rather than 6 sigma

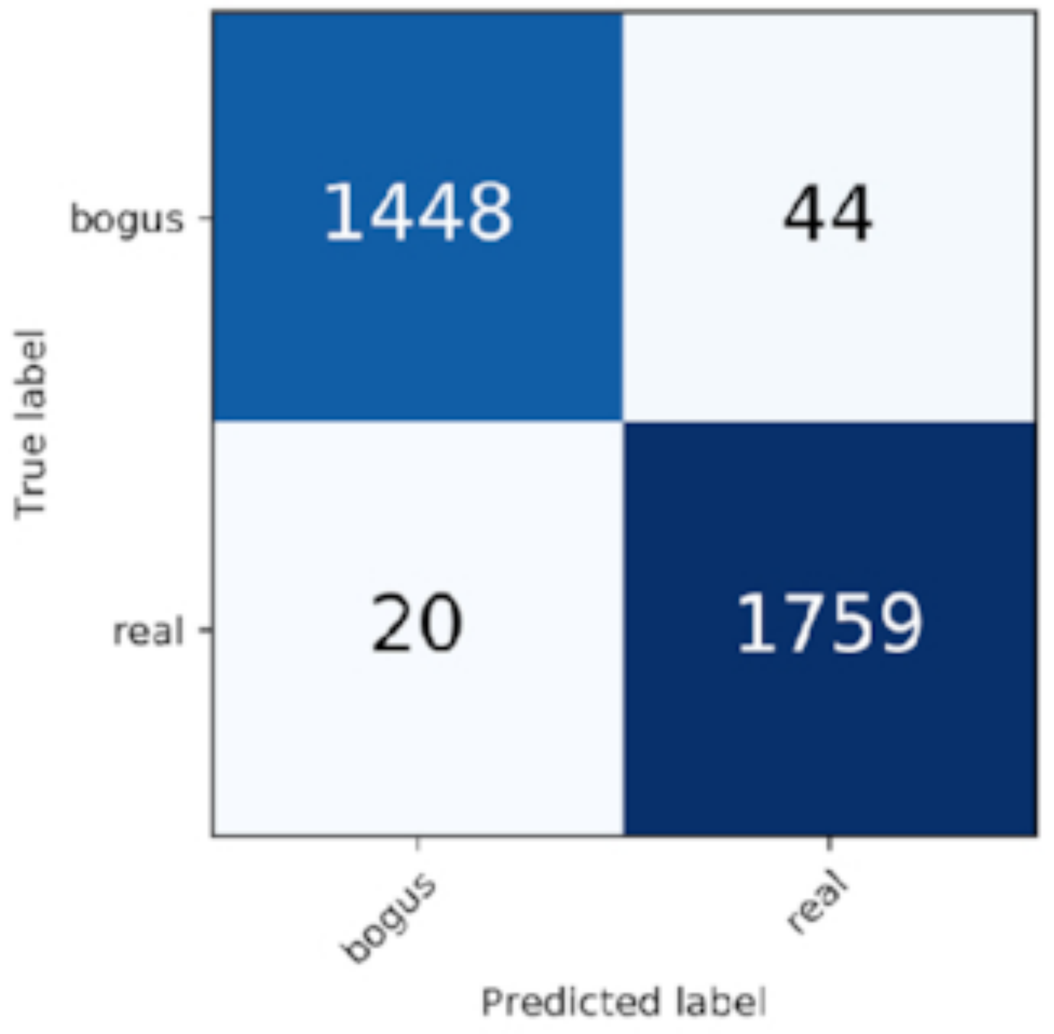
# We expect to be able to reach requirements with a CNN.



ZTF Braai classifier

Duev+20





| True label \ Predicted label | bogus | real |
|---|---|---|
| bogus | 1448 | 44 |
| real | 20 | 1759 |

Completeness: 98.9%

Purity: 97.6%

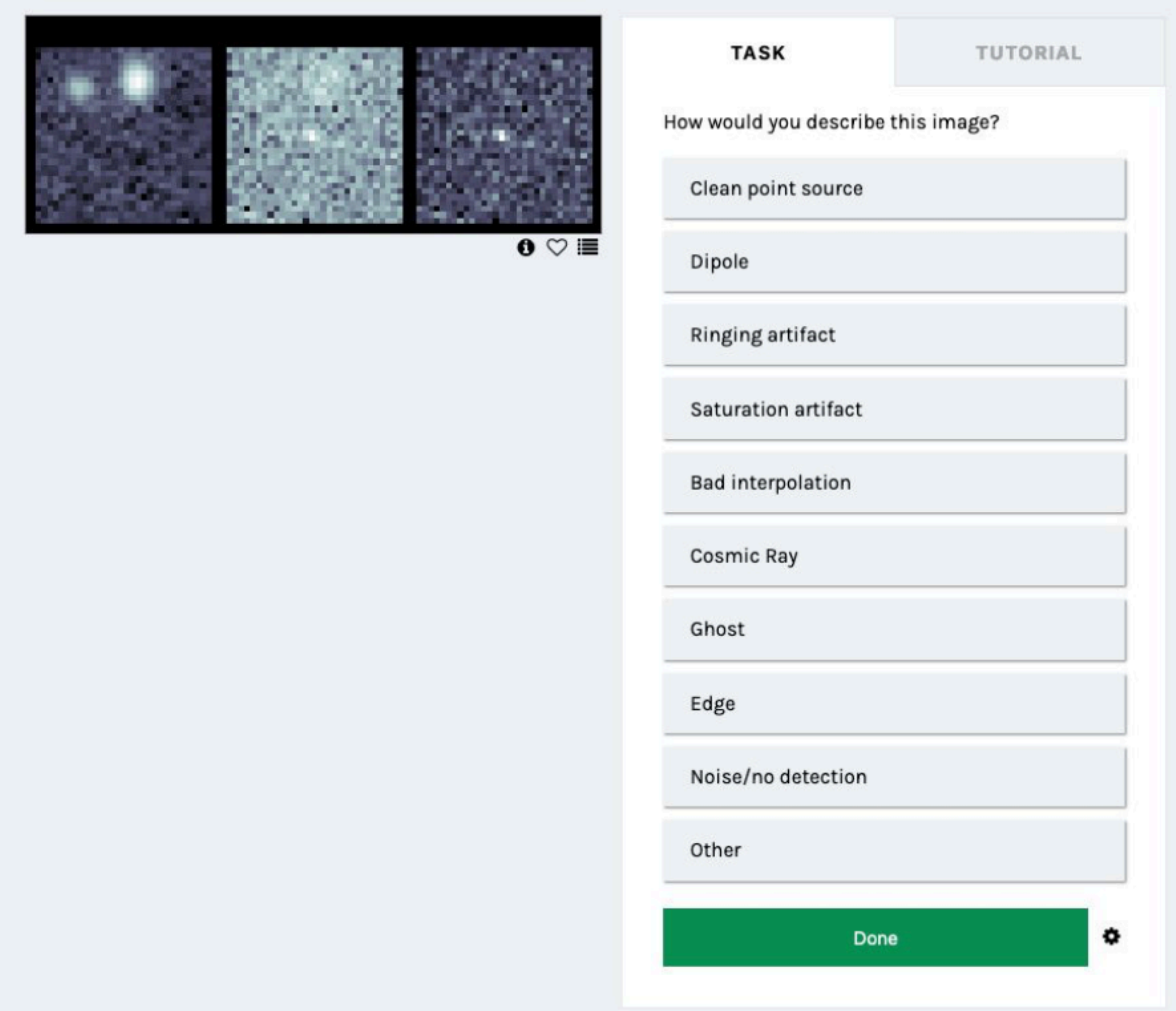
# We plan to use domain adaptation to speed model training on precursor and LSST data.

DC2 model weights provide a head start for learning about new cameras

We will need labels:

- Citizen Science (Zooniverse)
- Active Learning

Repeated model retraining will be needed well into operations.



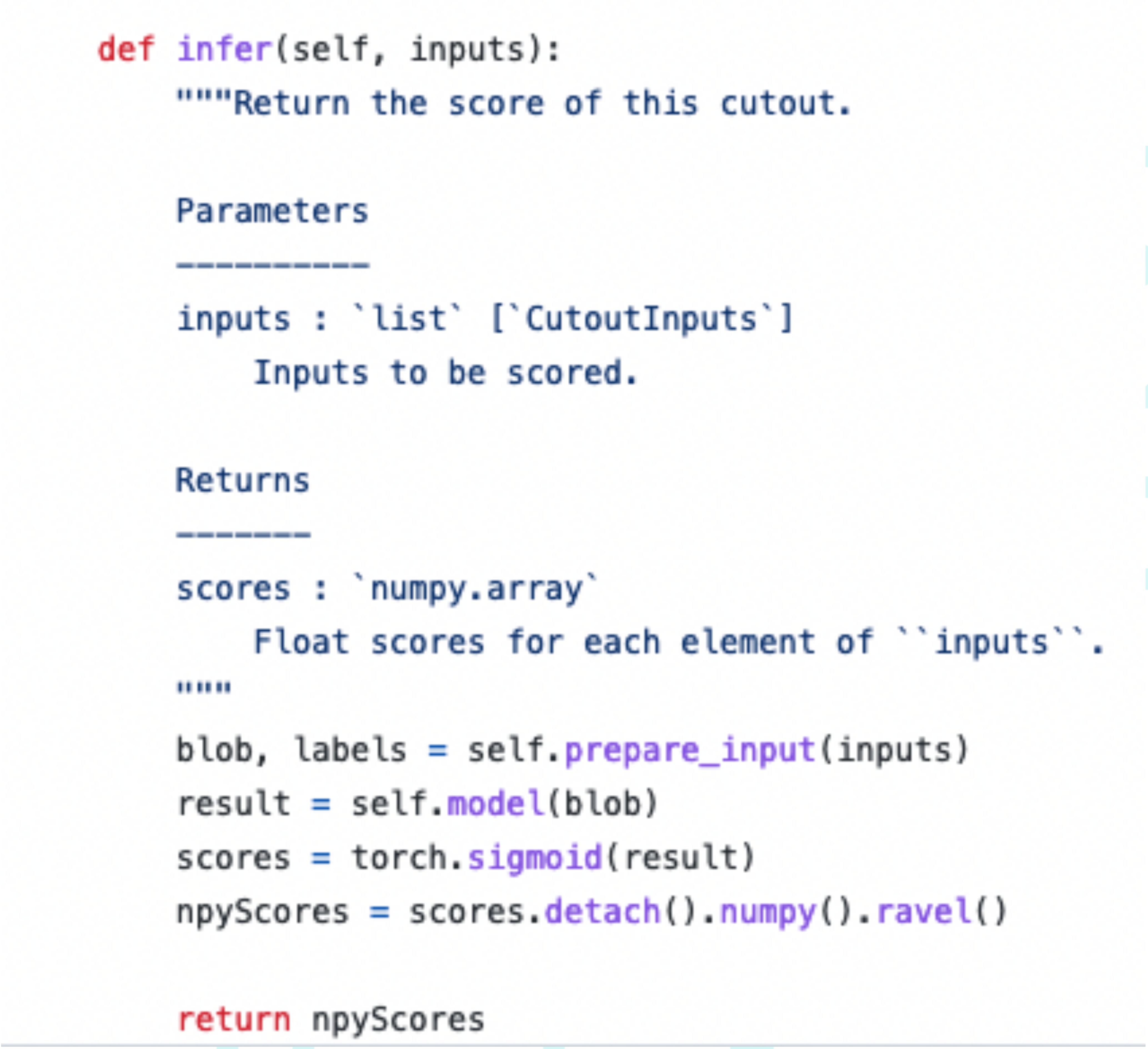
How would you describe this image?

- Clean point source
- Dipole
- Ringing artifact
- Saturation artifact
- Bad interpolation
- Cosmic Ray
- Ghost
- Edge
- Noise/no detection
- Other

# We are close to having RB scoring fully integrated into Science Pipelines.

Model trained on DC2 will be integrated into a new DC2 CI dataset for automated daily reprocessing

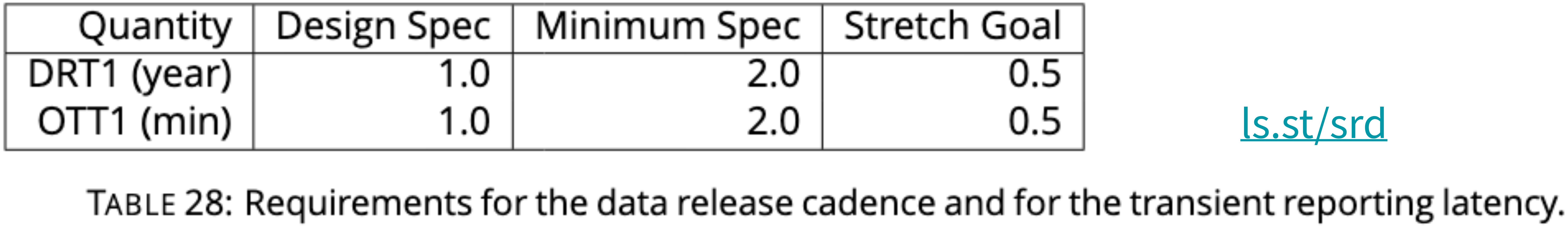
- direct daily assessment of completeness and purity requirements
- provides automated indication of model drift as pipelines are modified



```python
def infer(self, inputs):
    """Return the score of this cutout.

    Parameters
    ----------
    inputs : `list` [`CutoutInputs`]
        Inputs to be scored.

    Returns
    -------
    scores : `numpy.array`
        Float scores for each element of ``inputs``.
    """
    blob, labels = self.prepare_input(inputs)
    result = self.model(blob)
    scores = torch.sigmoid(result)
    npyScores = scores.detach().numpy().ravel()

    return npyScores
```

# Processing Time

# Rubin has an SRD requirement on alert latency.



| Quantity | Design Spec | Minimum Spec | Stretch Goal |
|---|---|---|---|
| DRT1 (year) | 1.0 | 2.0 | 0.5 |
| OTT1 (min) | 1.0 | 2.0 | 0.5 |

TABLE 28: Requirements for the data release cadence and for the transient reporting latency.

ls.st/srd

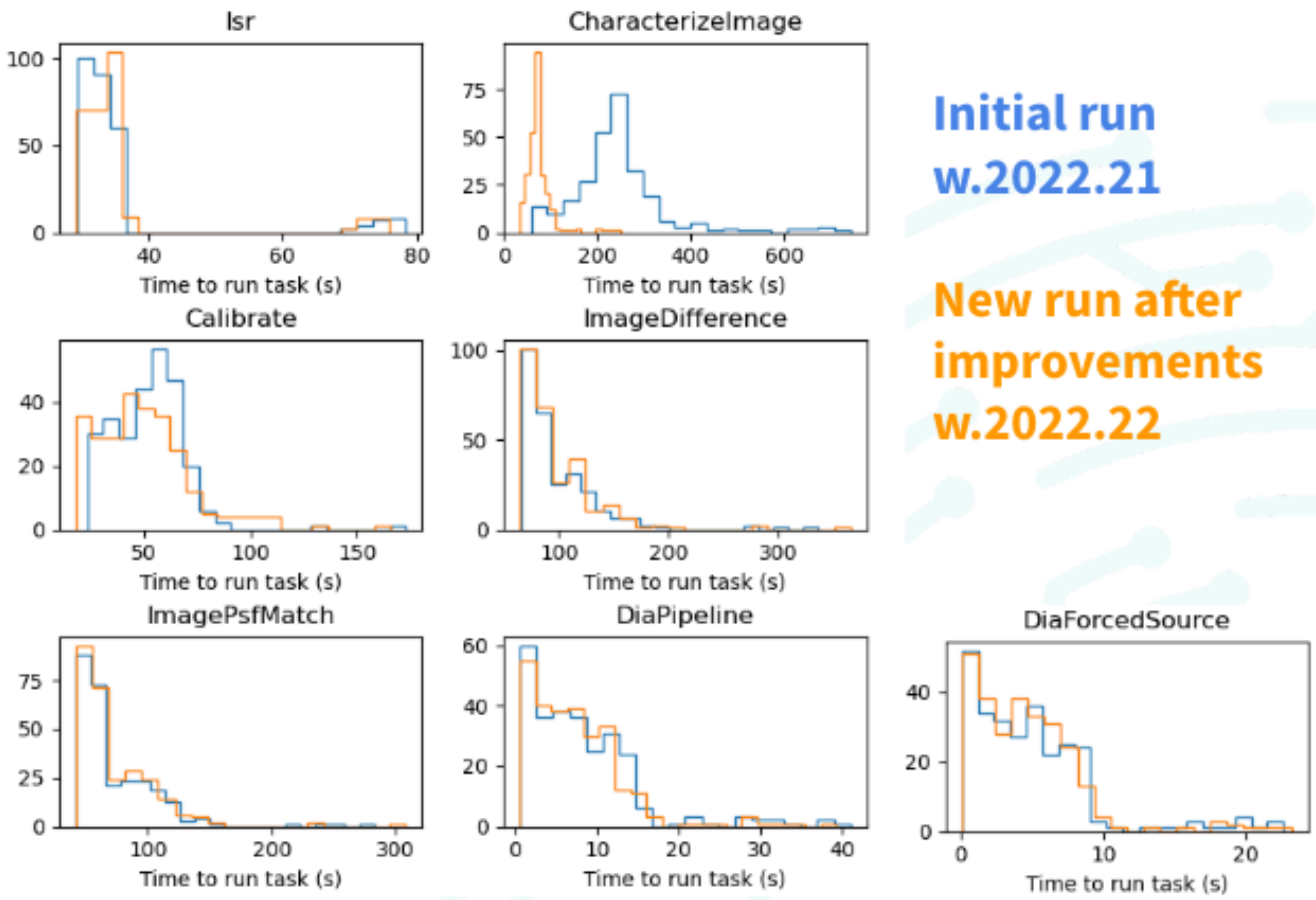
Until late 2021 we had not seriously engaged with latency in order to avoid premature optimization. Gen3 finalization and approaching end of construction make this timely now.

# We undertook a performance sprint using DC2 data in May '22.

Profiling identified nonessential processing, cutting execution time from >400 s to **~200 s** on NCSA hardware.

Further effort is planned (e.g., RFC-857). Maybe 30 more seconds of "easy" wins?

We began development of a DC2 CI dataset for automated estimates of LSST-relevant runtime.



Initial run w.2022.21

New run after improvements w.2022.22

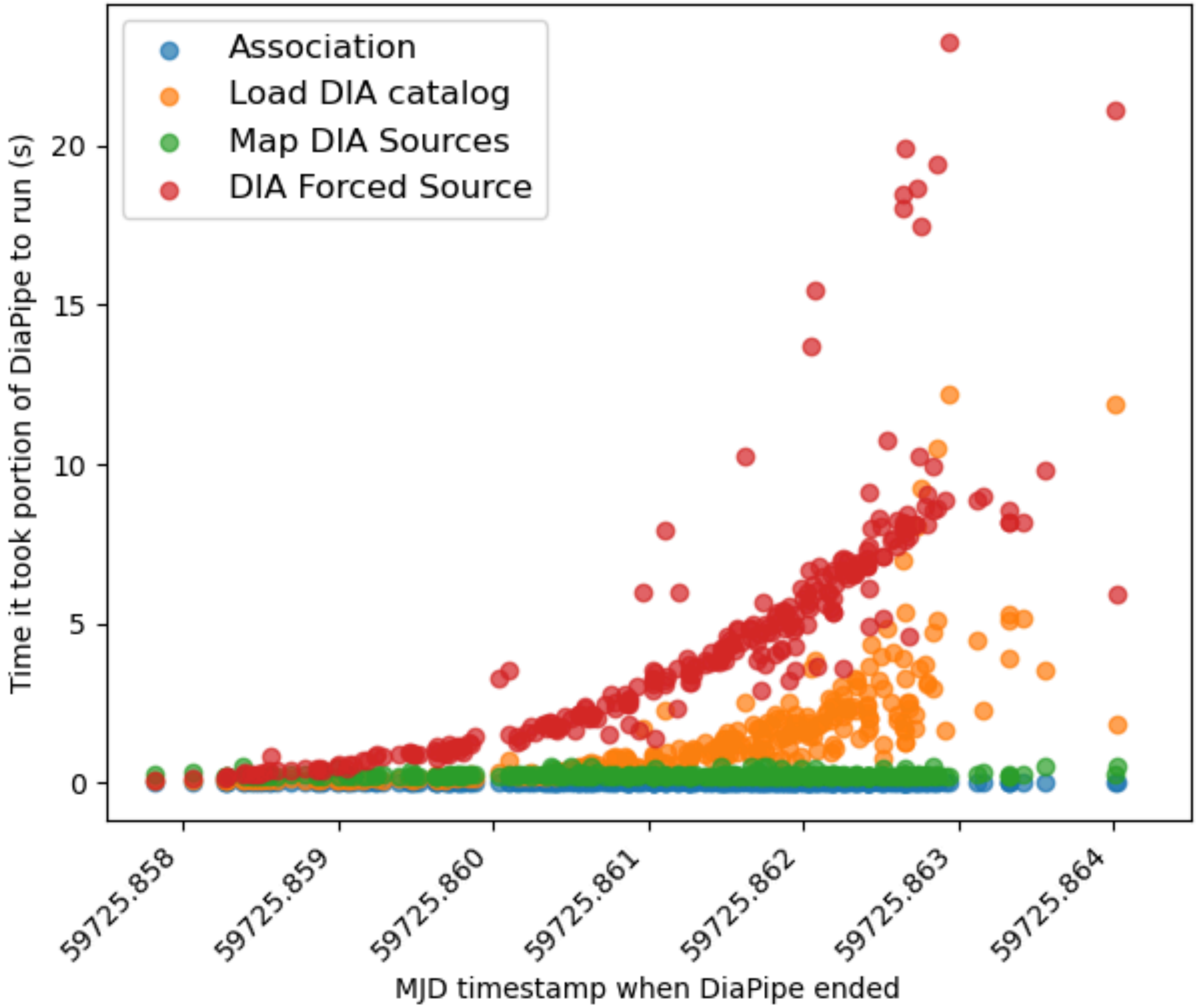
# These performance numbers need some caveats.

Tests weren't done on USDF hardware or production databases.

Prompt Processing was not available, so some preloading was not possible.

More processing-intensive timeseries features are expected to be added.

Profiling focused on single frame processing of a single visit--there is clearly optimization work needed when working with 12 months of historical data.



# PST guidance on prioritization is welcome.

We can choose how to prioritize our effort:

- reach 60 (or 120) second latency sooner (verification)
- focus on enabling early science (diffim quality, incremental templates, etc.; validation)

Proposal:

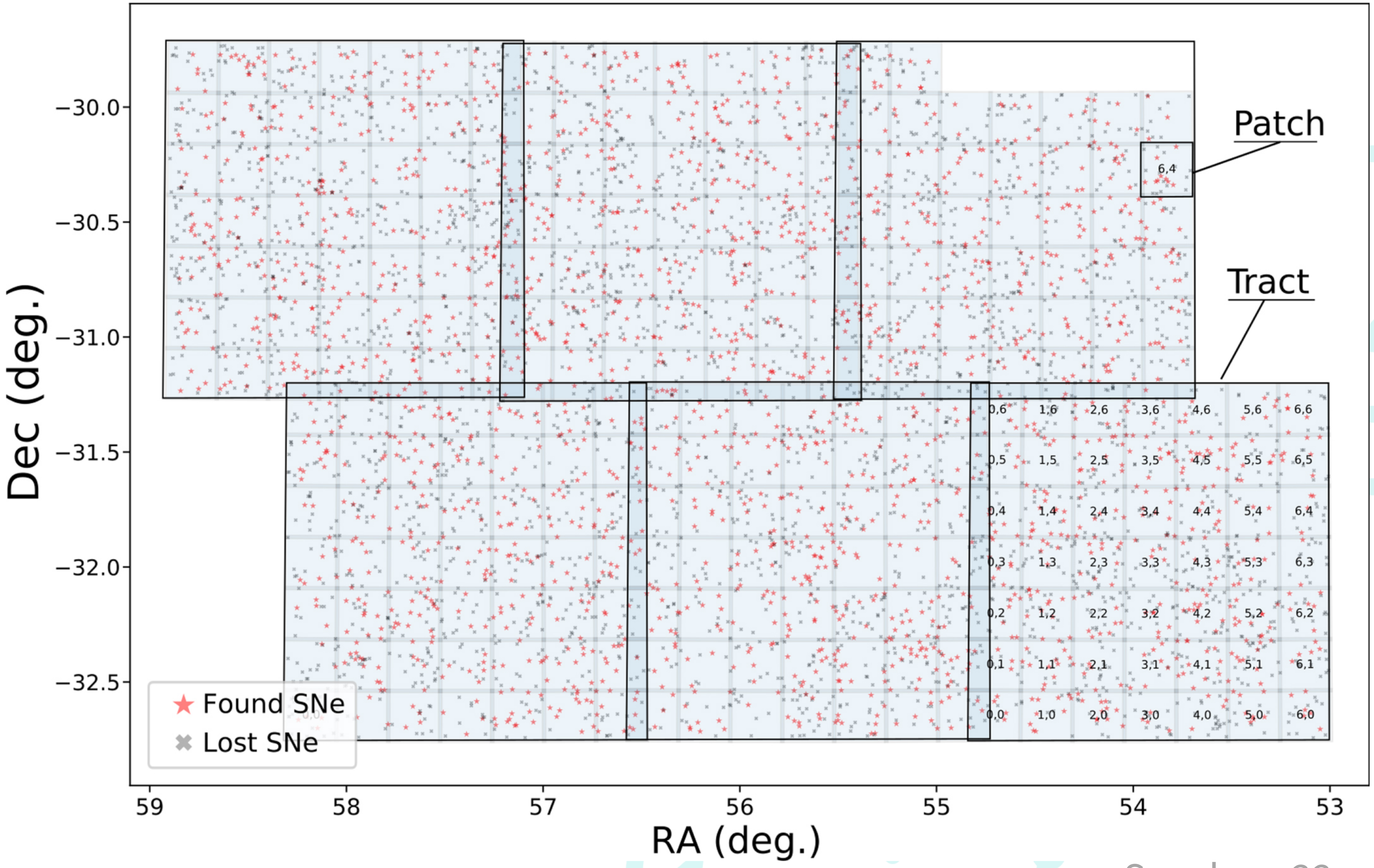
- Continue efforts to optimize the system at the pipelines level over next 12 months, expanding to systems level when Prompt Processing in the USDF is available.
- Target 120 seconds as the initial optimization goal, then divert resources to ensuring prompt processing, incremental template gen, and AP generally are running smoothly
- Push from 120 s → 60 s thereafter

# Incremental Template Generation

# Alert Production relies on the existence of template images.

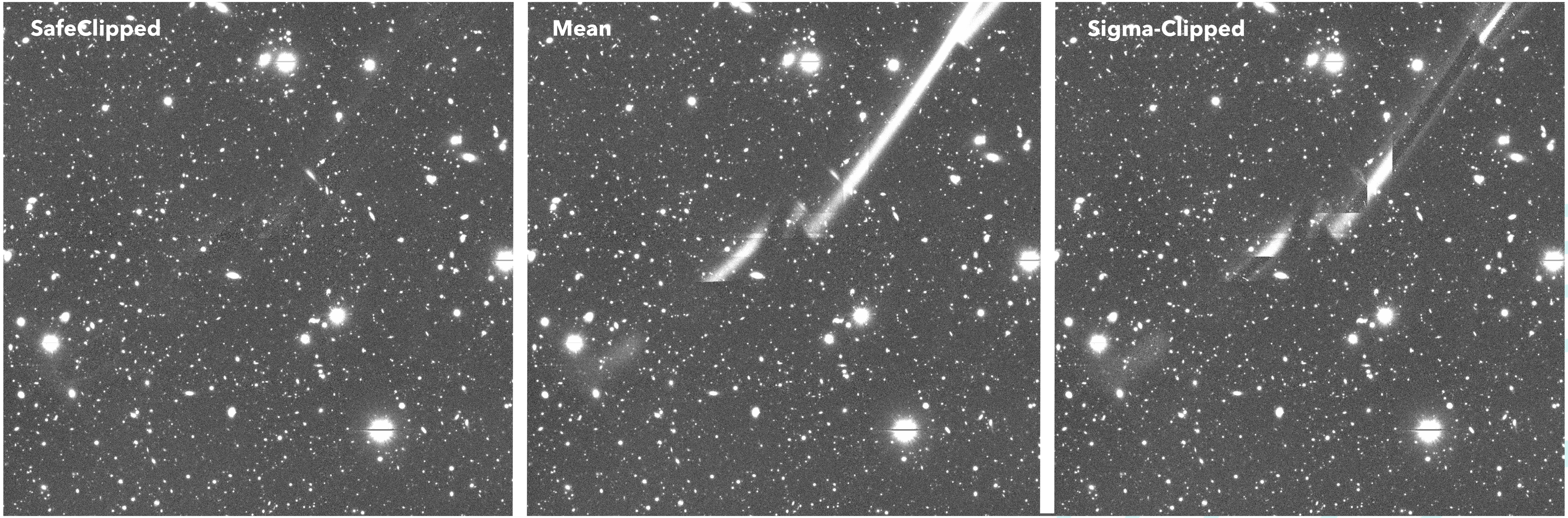
Simply a set of one or more images coadded and warped into (tract, patch) space

A single image can be used as a template, it just has to be appropriately warped



Sanchez+22

# Image differencing performance will be best with 3+ high SNR, good seeing images.



Y. AlSayyad

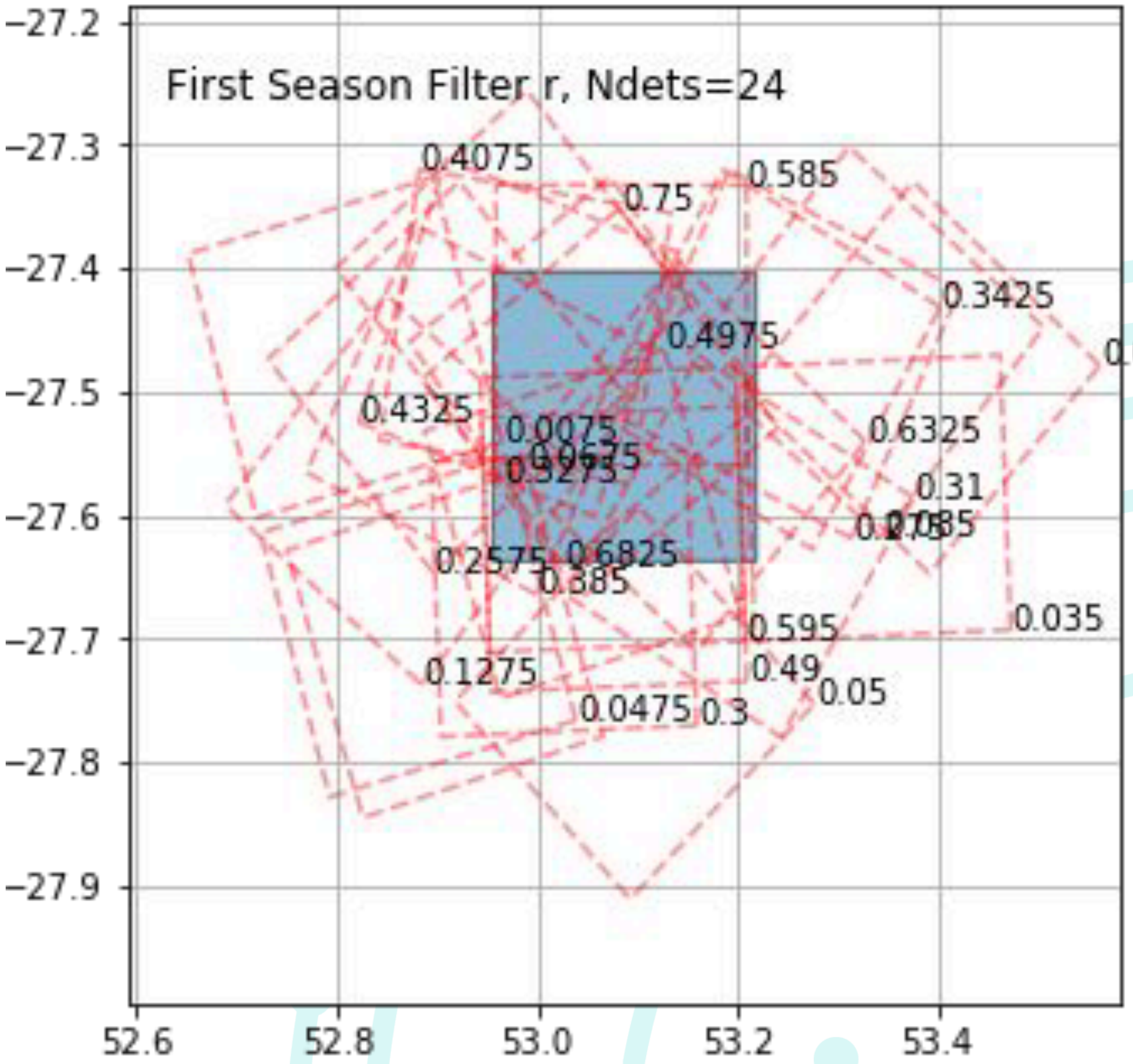
- Multiple images allow artifact rejection (DMTN-080)
- Multiple images reduce coadd noise and improve sensitivity
- Difference imaging performance improves with good-seeing templates

# LSST's large rotational and translational dithers complicate template construction.

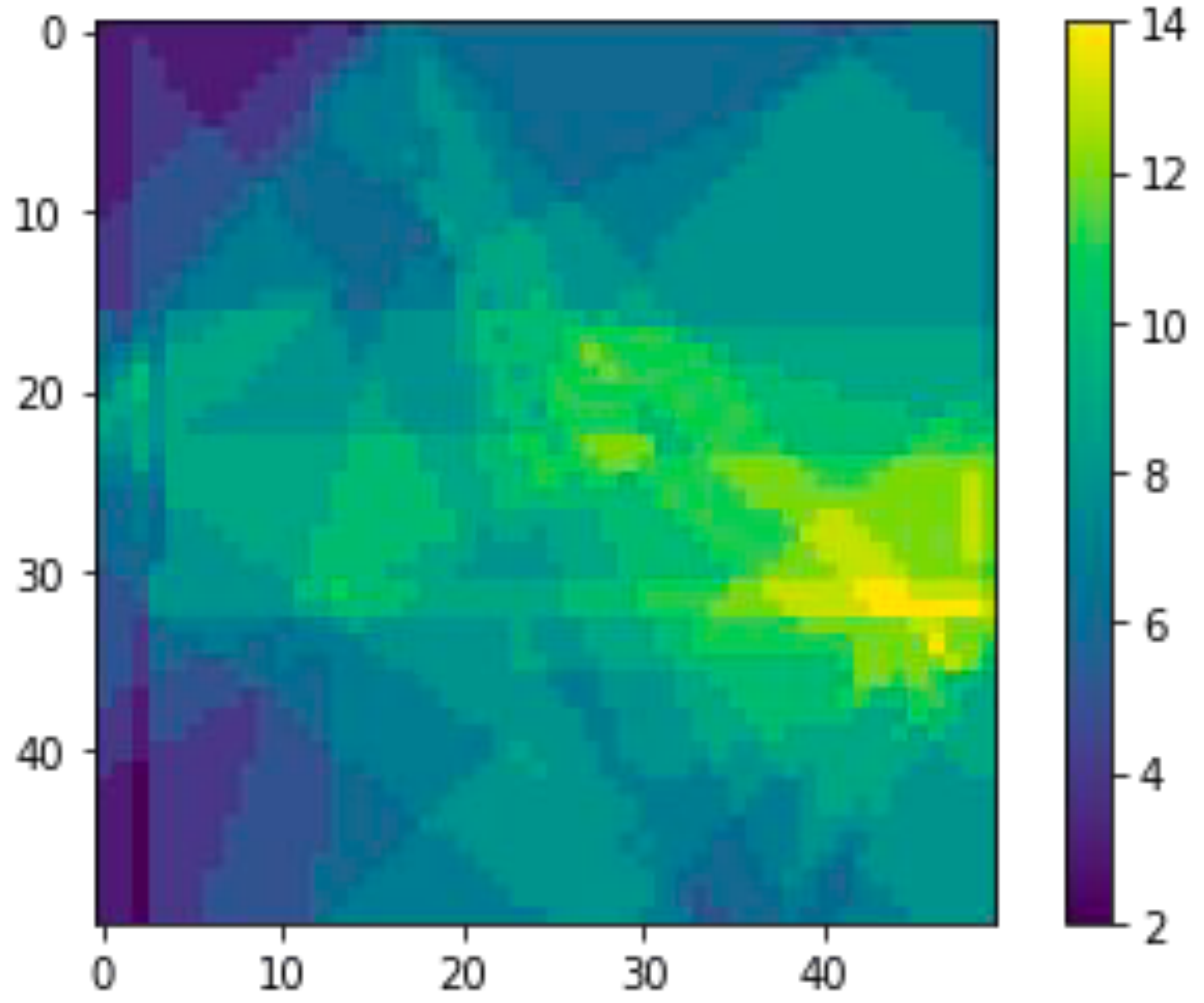
Dithering will leave CCD gaps in coadds at the patch level.

⇒ more images are needed to cover the entire patch

Armstrong & Sánchez



PSF discontinuities are a concern for image differencing



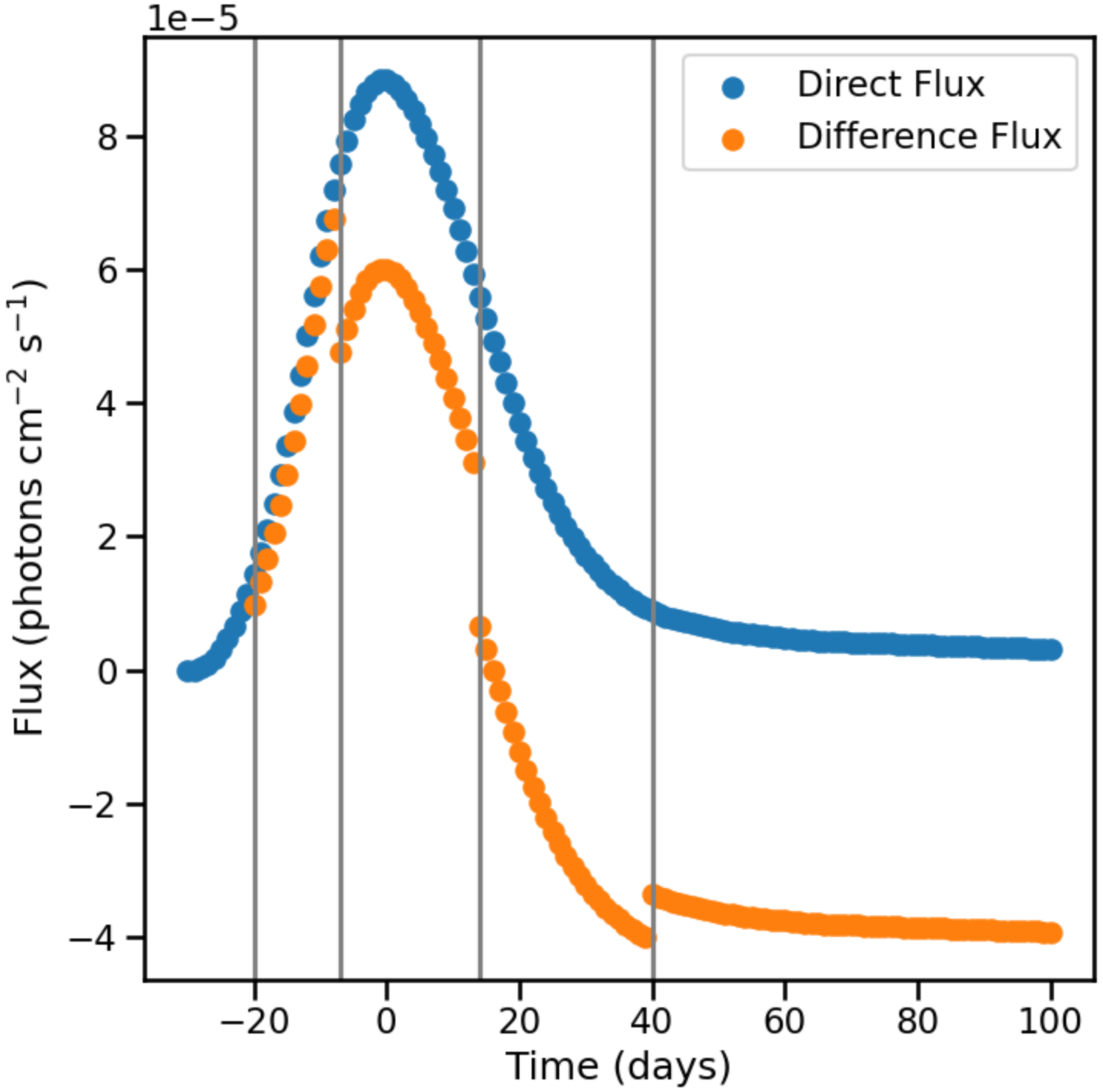
# Scientific use of DIASource history requires consistent template baselines.

Making DIASources from a single image is not very useful: which of 10,000 24th magnitude sources do you follow up? Need lightcurves, not single detections.

Even for Solar System science, need to be able to remove stationary sources to avoid $N^2$ combinatorics.

Changing templates contaminates transient lightcurves, changes baseline offsets for variable sources, and skews timeseries features.

⇒ We want to keep per-patch templates fixed from when we build them until DR1.



# We have to find a balance between how fast we can start alerts and the quality and completeness of the resulting alerts.

**Earliest possible templates**

+ rapid start to scientific utilization
+ potentially quick opportunities for agency press/PR
+ discovery of transients and SSOs that would otherwise be missed
+ longer timeseries baselines

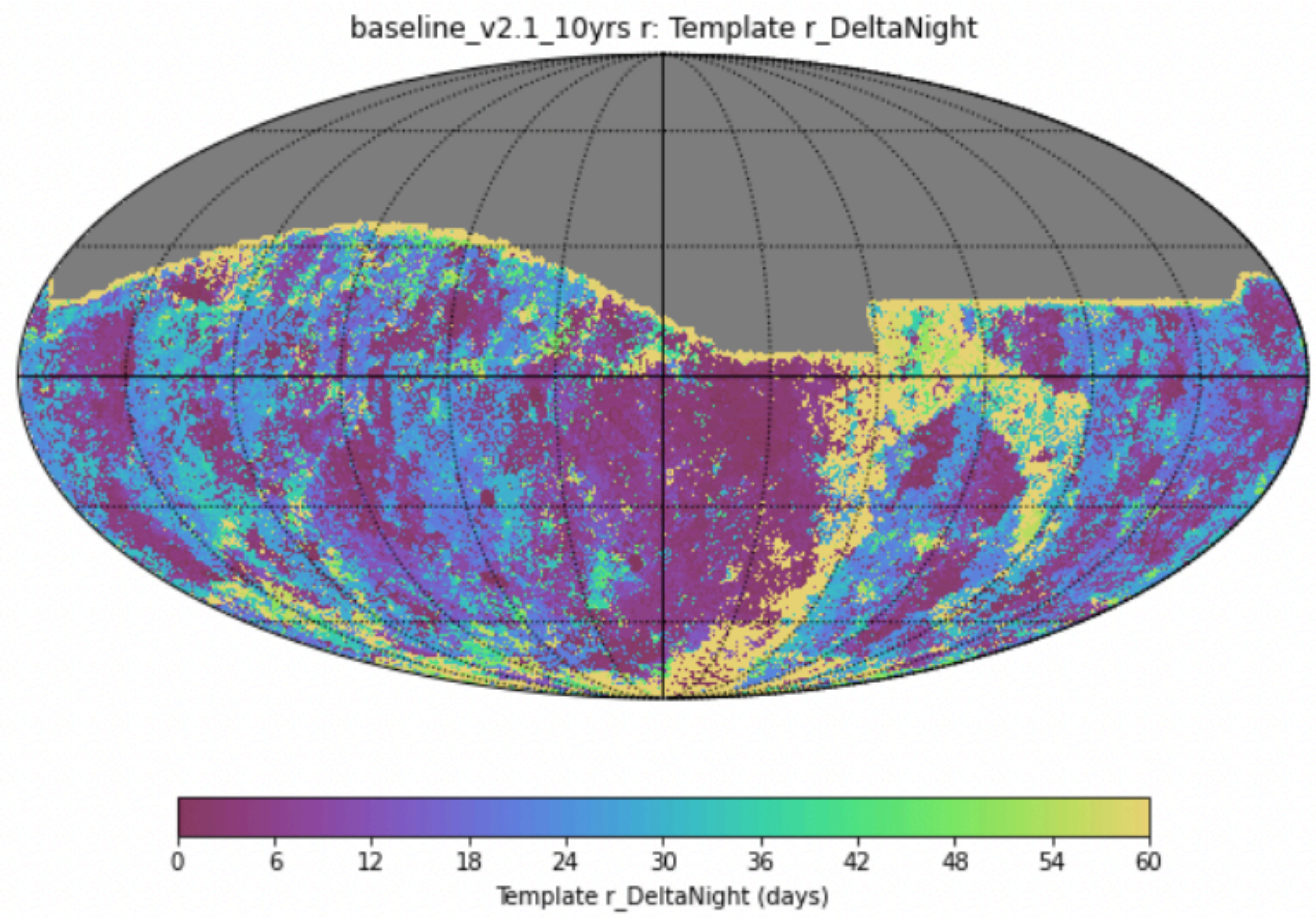
- missing template regions won't generate alerts until DR1
- noisier alerts until DR1
- reputational risk to the project if alerts are artifact-heavy

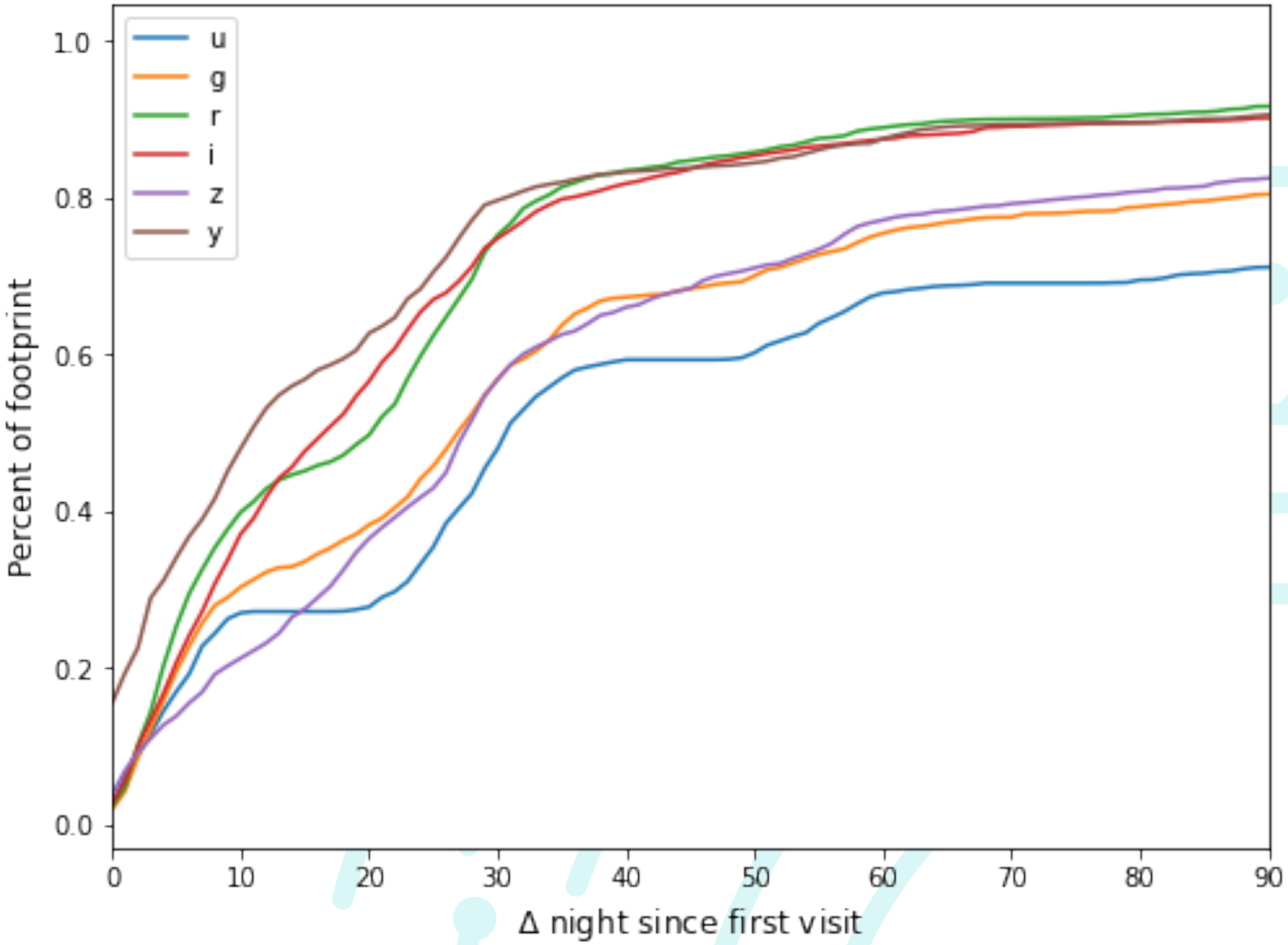
**Later, higher quality alerts**

+ greater spatial completeness
+ better RB & improved alert purity
+ higher SNR difference lightcurves

- slower start to early science
- later or missed discovery of some transients and SSOs

# Simulations suggests good-quality templates should be available relatively quickly.


baseline_v2.1_10yrs r: Template r_DeltaNight
0
6
12
18
24
30
36
42
48
54
60
Template r_DeltaNight (days)



Lynne Jones in #incremental-template-generation

# More work is needed to deliver high-quality templates and alerts as soon as practical.

Simulation studies:

- potential of alternative template-optimized survey strategies
- algorithms for assessing fill-factor
- science trade studies (earlier alerts vs. higher quality and completeness)

Precursor and commissioning data:

- identify practical quality cuts for image inputs to templates

Tooling:

- Software for semi-automated template generation as new images are taken
- Analysis tools for Operations team vetting of new templates prior to deployment

undertake this in the first half of 2023?

# Execution Environment

# Alert Production runs within Prompt Processing.

Meeting our 60 second readout-to-alert latency budget requires a specialized execution environment:

- Event-driven, low-latency
- With the required advanced notice of upcoming visits: pre-loads catalogs, calibrations, and other information before the image pixels arrive.
- Executes different software payloads on the fly
- Ideally: elastically scalable.

The UW Alert Production team is responsible for the software payloads but not the execution environment. Until recently, all AP testing on precursor datasets only used batch processing environments (e.g., SLURM, BPS).

# DMTN-219 provided a draft design and initial prototype of Prompt Processing.

In February 2022, K-T Lim provided a draft design for PP:

- next_visit notifications sent by HTTP POST
- processing conducted in containers using local butlers which synchronize to a central repository.

DMTN-219 included a skeletal prototype running on the Google Compute Platform (GCP), but without the actual science payloads and Butler datastores.

# In March the AP team conducted a short sprint to evaluate the PP prototype.

Sprint goal: critically evaluate the design by adapting the prototype to run the AP payload on real precursor data from HSC.

Major effort by Krzysztof Findeisen and John Parejko, technical assistance from K-T Lim, management supervision by Eric Bellm.

Major work captured on DM-33916; coordination discussion in #dm-prompt-processing.

Code at https://github.com/lsst-dm/prompt_prototype.

Rubin Summit
Google Cloud
Container
Next visit message
Next Visit topic (Pub/Sub)
Cloud Run
activator.py Pipelines Payload
Local Butler
Image upload complete topic (Pub/Sub)
Raw Image
Incoming Image Bucket (Cloud Storage)
Central Butler Repo (Cloud Storage)
Butler Registry (Postgres, Cloud SQL)
APDB (Postgres, Cloud SQL)

# We successfully integrated the AP pipelines into the prototype framework.

Achievements:

- completed run of the AP pipelines from raw images → APDB
- no showstoppers apparent in the design
- migrated code to a more permanent repo and improved tests and documentation.

See the sprint summary for more details & open questions.

# Substantial further work will be needed to productionize the framework and migrate it to the USDF.

GCP version needs further work for e.g. an AuxTel Prompt Processing Campaign

- production next_visit and image transfer tools at the telescope
- post-processing synchronization back to the central Butler (DM-35053, in progress by AP)
- configurable selection of pipeline payload
- error handling & performance improvements

Current prototype relies heavily on GCP technology which will need to be replaced for the USDF.

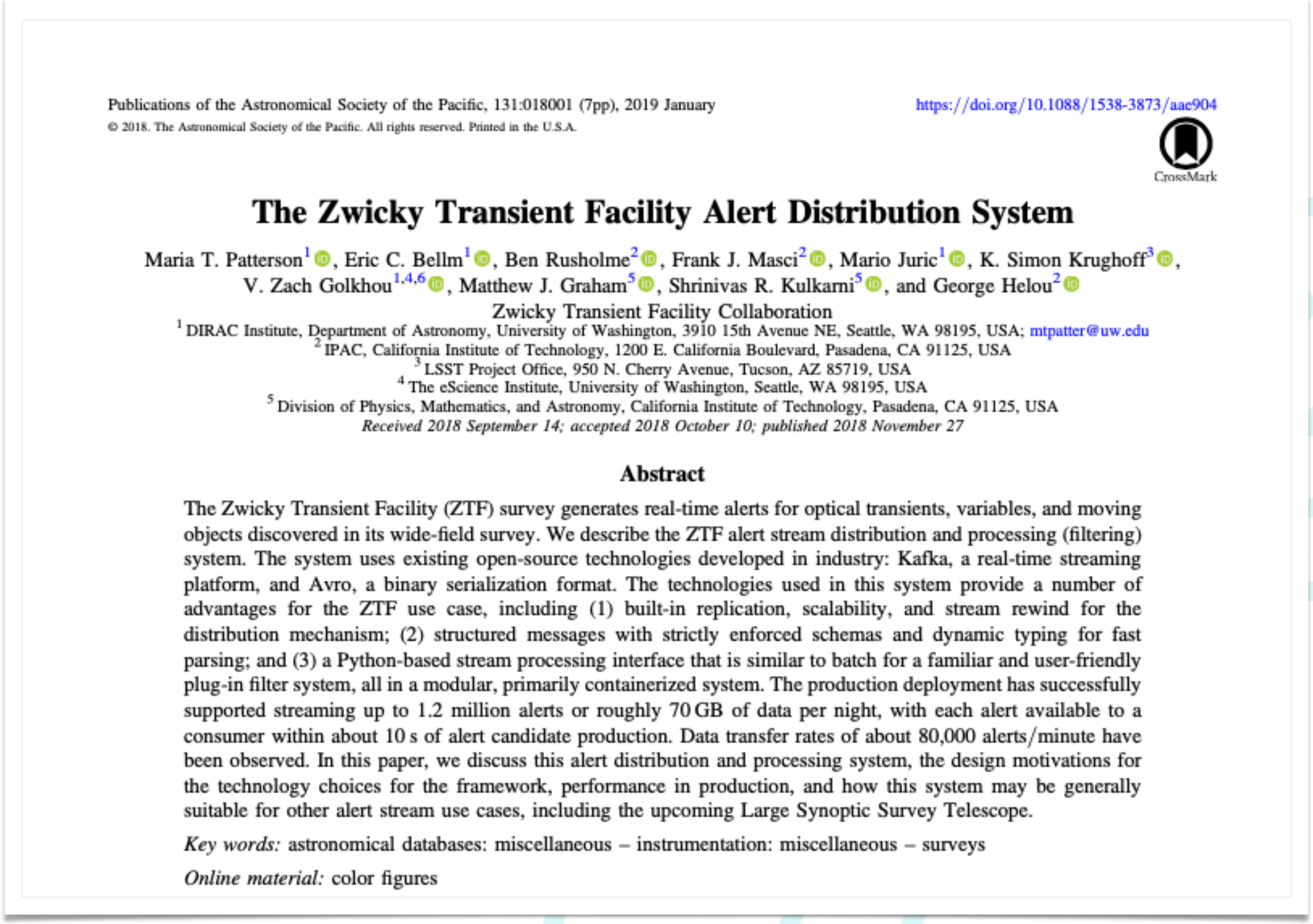
- Pub/Sub → Kafka
- Cloud Run → Kubernetes + Ingress + Horizontal Pod Autoscaler
- Cloud Storage → MinIO or Ceph Object Gateway (RADOS)
- APDB Cloud SQL Postgres → Cassandra

This is substantial work and will need effort from ARCH and an owner in the USDF.

# Alert Distribution

# Rubin Alert Distribution is based on mature components.

Kafka and Avroare both used by ZTF and existing brokers.



Publications of the Astronomical Society of the Pacific, 131:018001 (7pp), 2019 January https://doi.org/10.1088/1538-3873/aae904

## The Zwicky Transient Facility Alert Distribution System

Maria T. Patterson[1], Eric C. Bellm[1], Ben Rusholme[2], Frank J. Masci[2], Mario Juric[1], K. Simon Krughoff[3], V. Zach Golkhou[1,4,6], Matthew J. Graham[5], Shrinivas R. Kulkarni[5], and George Helou[2]
Zwicky Transient Facility Collaboration
[1] DIRAC Institute, Department of Astronomy, University of Washington, 3910 15th Avenue NE, Seattle, WA 98195, USA; mtpatter@uw.edu
[2] IPAC, California Institute of Technology, 1200 E. California Boulevard, Pasadena, CA 91125, USA
[3] LSST Project Office, 950 N. Cherry Avenue, Tucson, AZ 85719, USA
[4] The eScience Institute, University of Washington, Seattle, WA 98195, USA
[5] Division of Physics, Mathematics, and Astronomy, California Institute of Technology, Pasadena, CA 91125, USA

*Received 2018 September 14; accepted 2018 October 10; published 2018 November 27*

### Abstract

The Zwicky Transient Facility (ZTF) survey generates real-time alerts for optical transients, variables, and moving objects discovered in its wide-field survey. We describe the ZTF alert stream distribution and processing (filtering) system. The system uses existing open-source technologies developed in industry: Kafka, a real-time streaming platform, and Avro, a binary serialization format. The technologies used in this system provide a number of advantages for the ZTF use case, including (1) built-in replication, scalability, and stream rewind for the distribution mechanism; (2) structured messages with strictly enforced schemas and dynamic typing for fast parsing; and (3) a Python-based stream processing interface that is similar to batch for a familiar and user-friendly plug-in filter system, all in a modular, primarily containerized system. The production deployment has successfully supported streaming up to 1.2 million alerts or roughly 70 GB of data per night, with each alert available to a consumer within about 10 s of alert candidate production. Data transfer rates of about 80,000 alerts/minute have been observed. In this paper, we discuss this alert distribution and processing system, the design motivations for the technology choices for the framework, performance in production, and how this system may be generally suitable for other alert stream use cases, including the upcoming Large Synoptic Survey Telescope.

*Key words:* astronomical databases: miscellaneous – instrumentation: miscellaneous – surveys

*Online material:* color figures

Patterson+19

# Rubin has agreed to send the full alert stream to seven brokers; others will operate downstream.

Seven brokers were selected for direct access to the full alert stream:

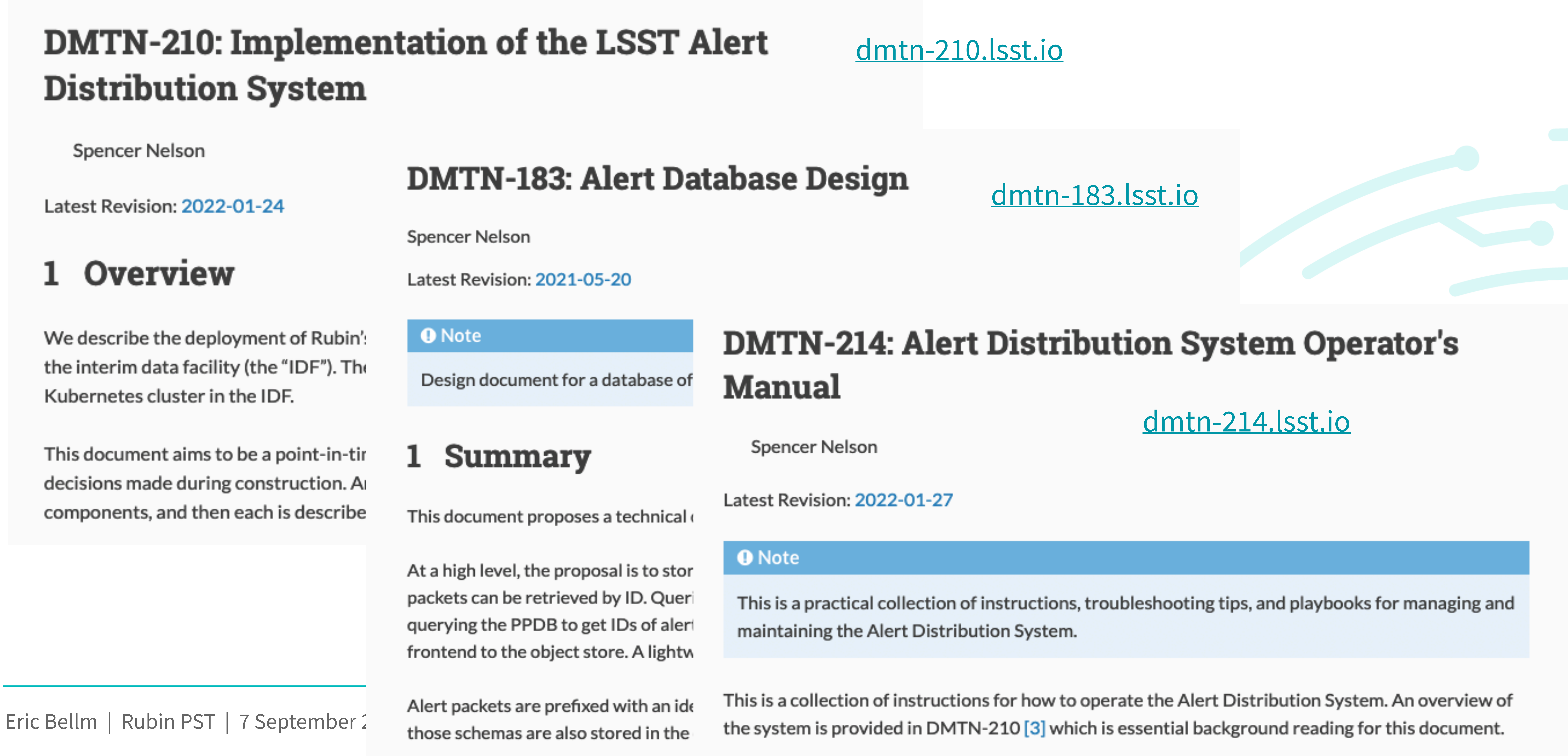
- ALeRCE
- AMPEL
- ANTARES
- Babamul
- Fink
- Lasair
- Pitt-Google

Two additional brokers were recommended to operate downstream:

- SNAPS
- POI/Variables

All of these teams are receiving ZTF alerts.

# We have developed and deployed the alert distribution system in the Rubin Kubernetes environment in the IDF.



dmtn-210.lsst.io

## DMTN-210: Implementation of the LSST Alert Distribution System

Spencer Nelson

Latest Revision: 2022-01-24

### 1 Overview

We describe the deployment of Rubin's
the interim data facility (the "IDF"). Th
Kubernetes cluster in the IDF.

This document aims to be a point-in-ti
decisions made during construction. A
components, and then each is describe

dmtn-183.lsst.io

## DMTN-183: Alert Database Design

Spencer Nelson

Latest Revision: 2021-05-20

Note

Design document for a database of

### 1 Summary

This document proposes a technical

At a high level, the proposal is to stor
packets can be retrieved by ID. Queri
querying the PPDB to get IDs of aler
frontend to the object store. A lightw

Alert packets are prefixed with an ide
those schemas are also stored in the

dmtn-214.lsst.io

## DMTN-214: Alert Distribution System Operator's Manual

Spencer Nelson

Latest Revision: 2022-01-27

Note

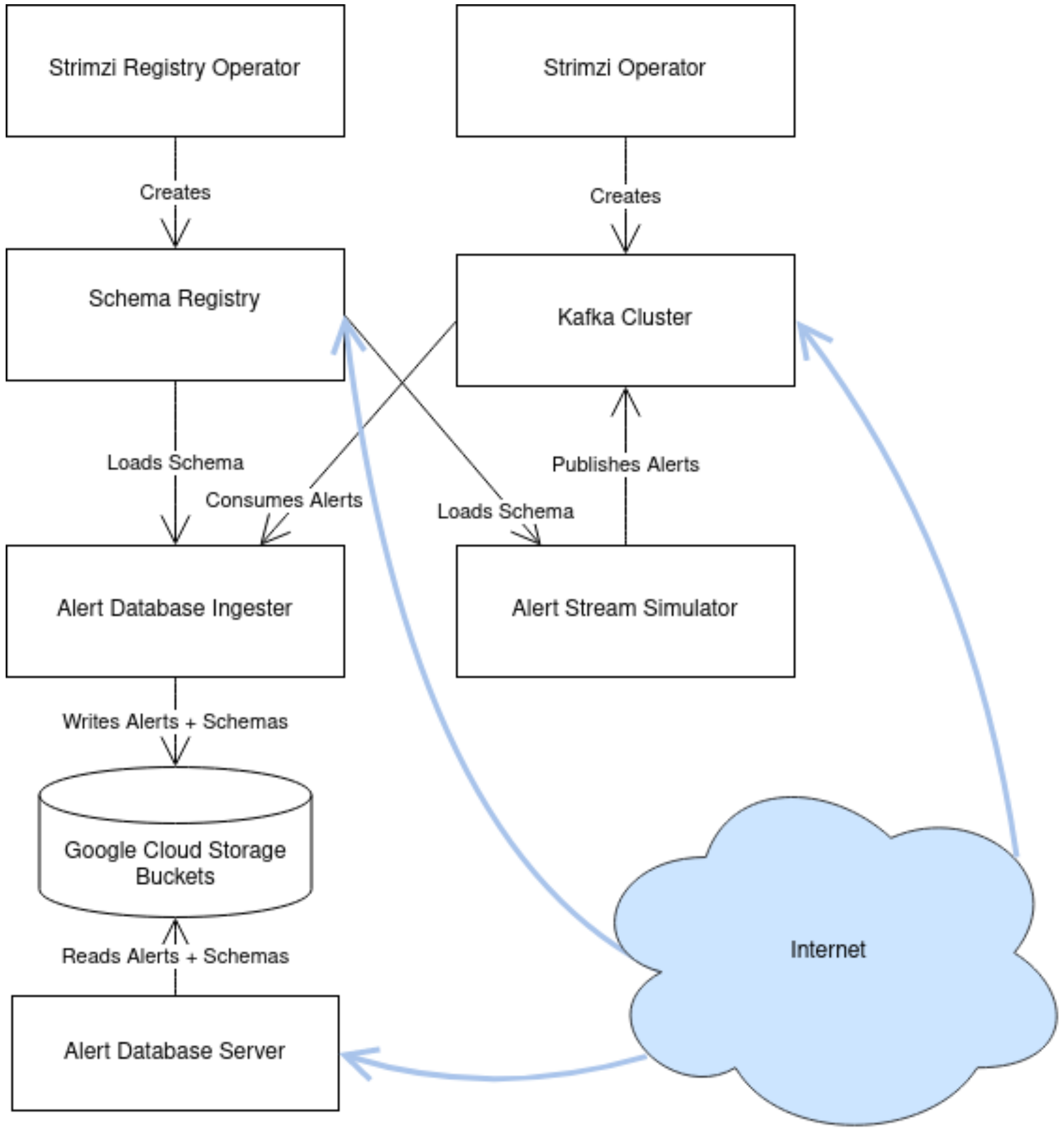
This is a practical collection of instructions, troubleshooting tips, and playbooks for managing and maintaining the Alert Distribution System.

This is a collection of instructions for how to operate the Alert Distribution System. An overview of the system is provided in DMTN-210 [3] which is essential background reading for this document.

# In January 2022 community alert brokers connected to the prototype Rubin Alert Distribution system running in the IDF.

For the first time, we stood up a production-like Kafka cluster and alert archive.

Community broker teams successfully authenticated and received canned sample alerts during a test run in January 2022.



dmtn-210.lsst.io

# We plan to stand up alert distribution at the USDF this October.

Modest changes to Square's Phalanx Helm chart configurations need to be integrated.

Minor GCP-specific implementation details need to be removed.

Reasonable to contemplate testing USDF pipelines → brokers in early 2023?

Dr. Breanna Smart

# The AVS system requires development.

While the broad outline of the AVS needs are clear we have not yet begun technical implementation.

Organizational interfaces have not been developed in detail.

Risk to most alert science should not be high?

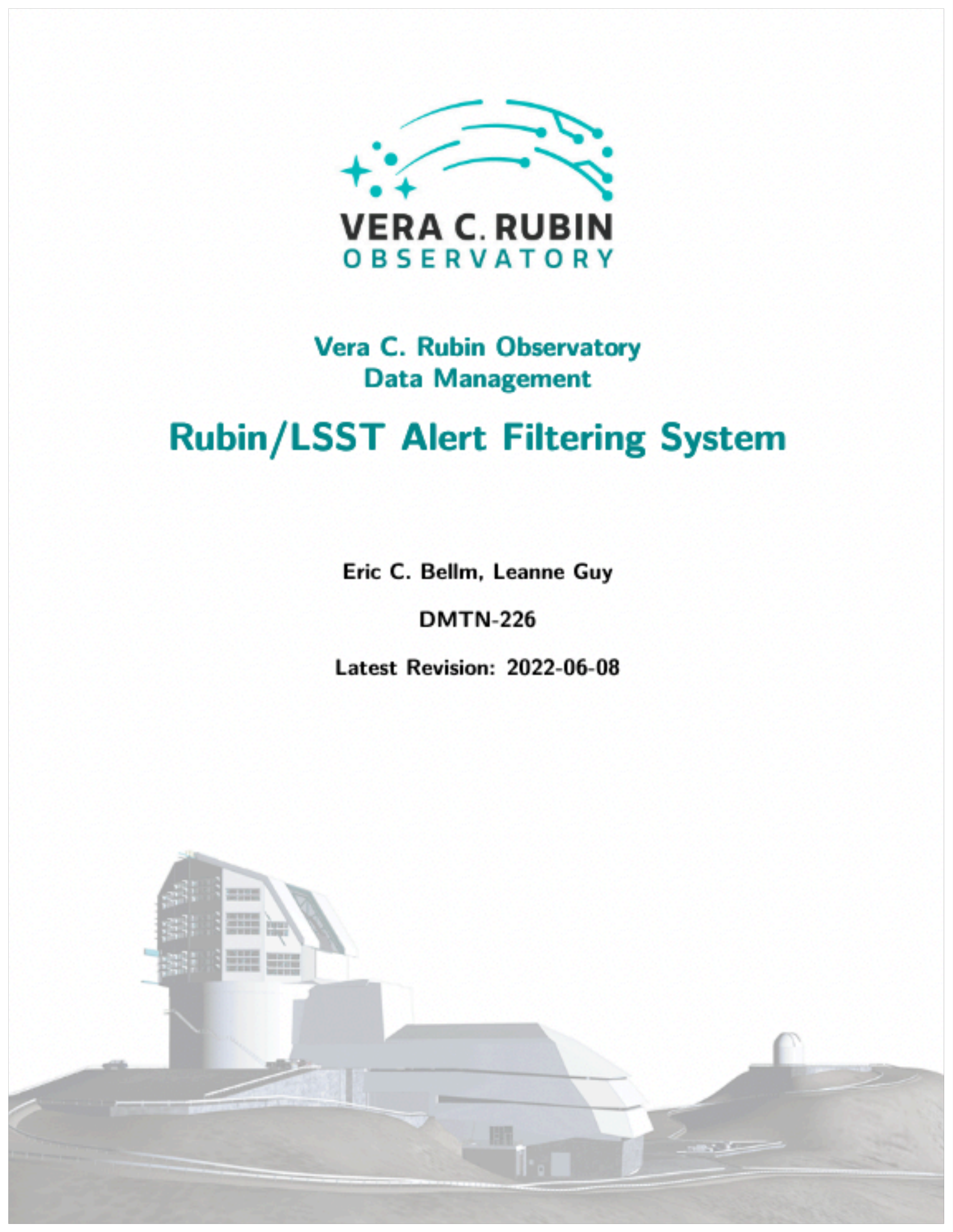
# We are planning to partner with ANTARES on the functionality planned for the Alert Filtering Service.

The SRD recognized the need for user and pre-defined alert filters. The Rubin Alert Filtering System (AFS) was a stopgap if no community brokers became available--with nine operational brokers that concern is reduced. But brokers determine their services independently.

The ANTARES broker has a user filtering service which appears to meet Rubin requirements and operates within NOIRLab along with Rubin operations.

We are partnering with ANTARES to deliver the capabilities envisioned in the Rubin Alert Filtering Service & investigating the programmatic and technical implications.

Need to finalize the details with ANTARES this fall, LCR, and publicize to the community.



VERA C. RUBIN OBSERVATORY

Vera C. Rubin Observatory
Data Management

Rubin/LSST Alert Filtering System

Eric C. Bellm, Leanne Guy

DMTN-226

Latest Revision: 2022-06-08

dmtn-226.lsst.io

# Backup slides

VERA C. RUBIN
OBSERVATORY