A3D3 High-Throughput AI Methods and Infrastructure
Seattle - July 10th, 2023

# A3D3 MMA: Caltech/Drexel

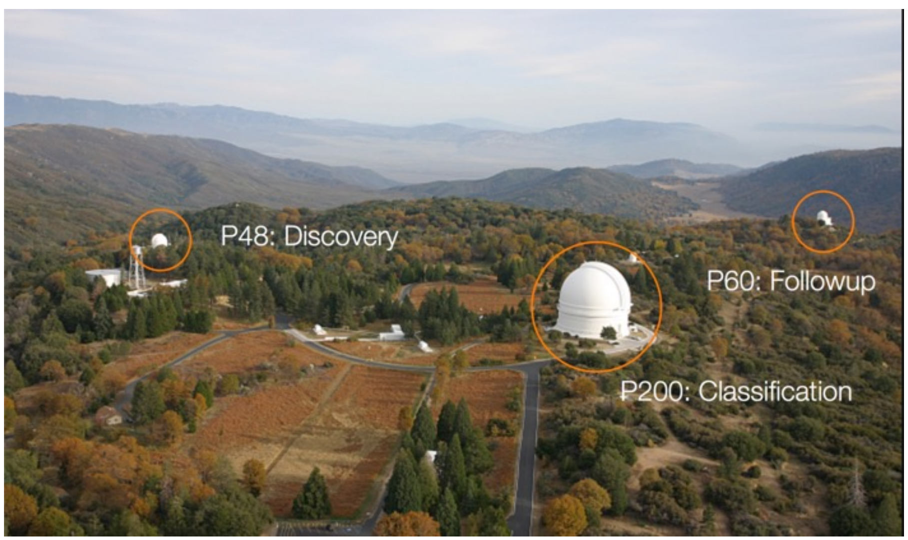
## Optimizing Followup of Transient Alerts

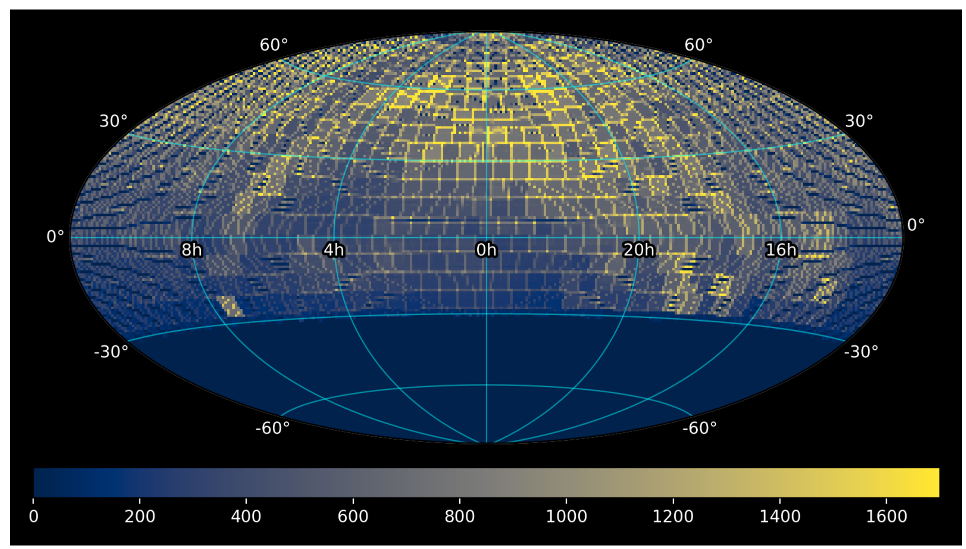
Matthew J. Graham
(mjg@caltech.edu)
Niharika Sravan

Supported by NSF OAC-2117997 and AST-2034437

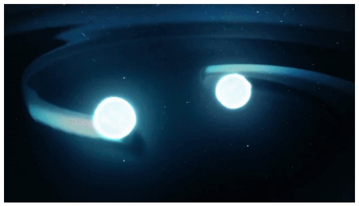
# The Zwicky Transient Facility (ZTF: 2018 – 202?)

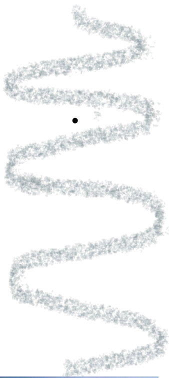
- 47 $deg^2$ field of view camera on Palomar Oschin 48" telescope
- 3750 $deg^2$ / hr to 20.5-21 mag (1.4 TB / night (compressed))
- Full northern sky every two nights in *g, r* and *i* (every three – four nights)
- Over 5 years: 5 PB, 750 billion measurements, ~1000 measurements / source
- First megaevent survey: $10^6$ alerts per night (~300M public since June 2018)
- ZTF is ~10% of Rubin LSST
- Can also be used as a followup facility

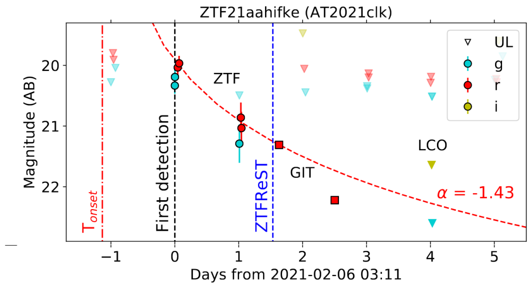


# LIGO followup pipeline

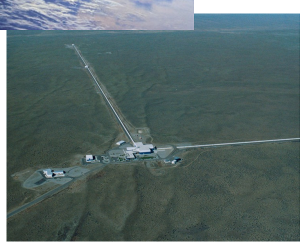






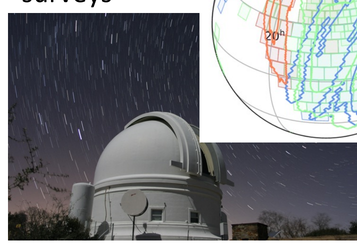

< ~1 week

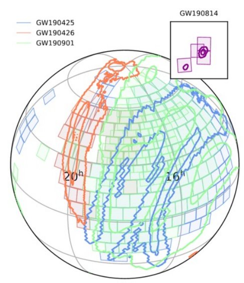
Collect more data given
- Host galaxy (type/redshift)
- Forced photometry light curves
- Updated GW estimates



~mins-hrs

surveys





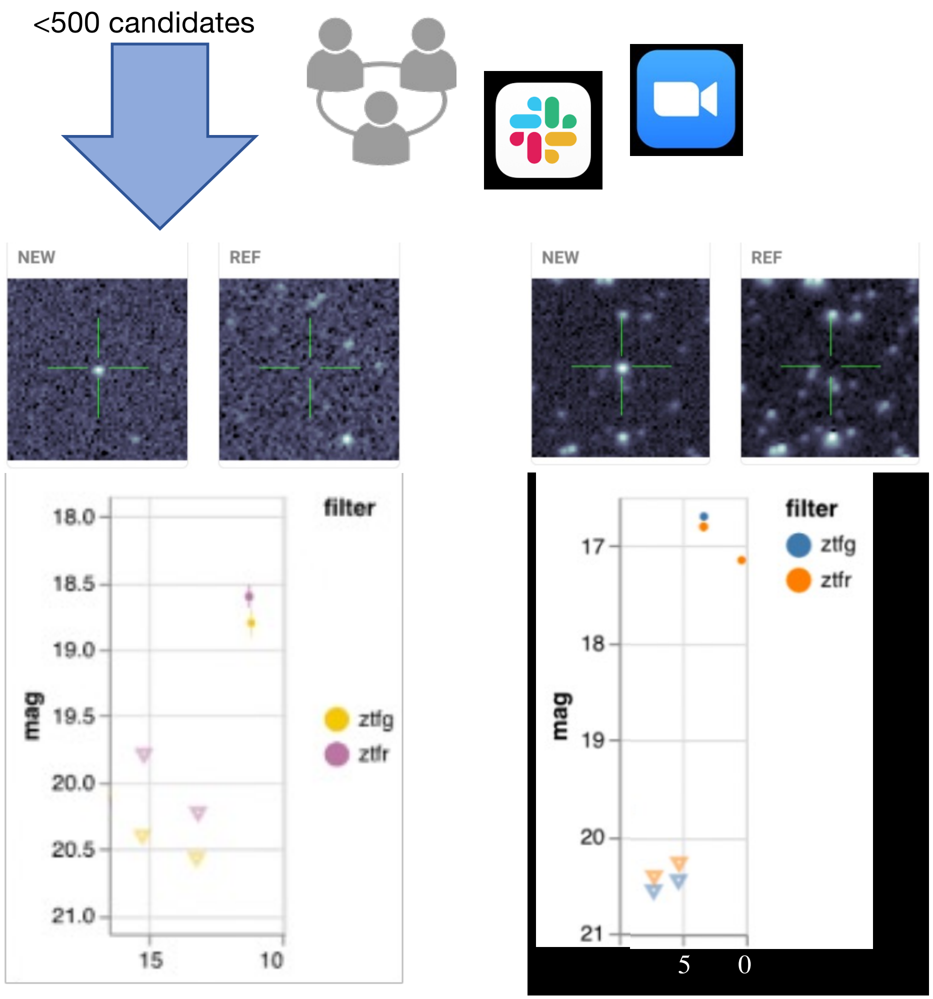

Public alerts
- Event time
- Sky localization
- Distance
- CBC probabilities

~mins

Automated vetting
- Real-bogus classification
- Not a known source
- Evolution rate

~secs-mins
<500 candidates

# The followup bottleneck problem




Goal:

- Use limited resources to acquire more information to:
  - Identify the event
  - Maximize constraints on interesting light curve physics

- Additional follow-up is critical!
- Classifiers don't answer what to do next and how to adapt

Process needs to be:

✓ Free from fatigue/bias
✓ Low-latency
✓ Scalable

# Autonomous decision making

O4 is operating at twice the sensitivity as O3 (eventually)

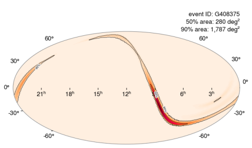
- 50-250 detections a year compared to 20 last time
- Localizations will not improve by much

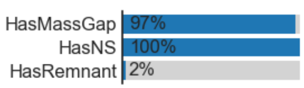
Rubin will produce 10x as many candidates for human experts to analyze

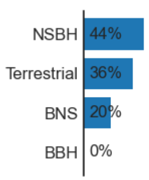
Follow-up resources will not increase at nearly the same rate

Current protocol not sustainable or suitable to get at statistics

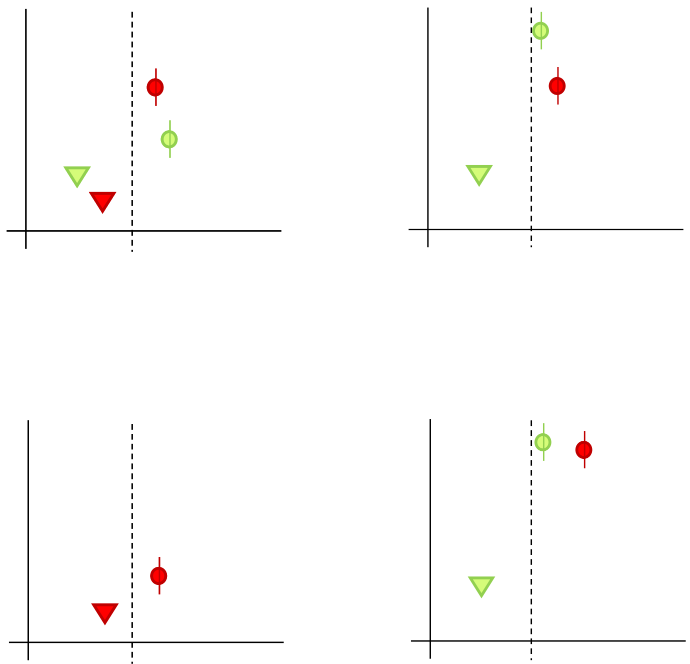
# Autonomous resource allocation






FAR – 9.59 per year



30 hr per semester
g, r, i, z

1 per night
g, r

low res spec
~19mag
up to 5 per night

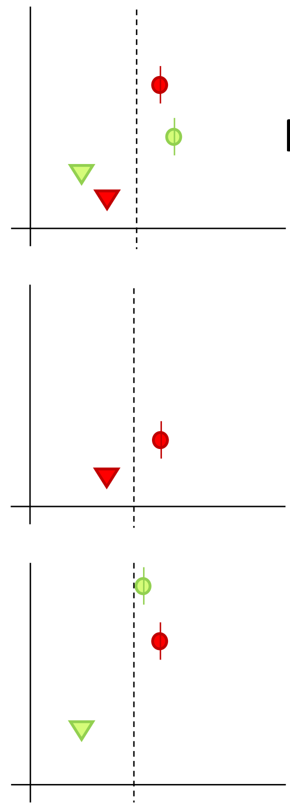
N < 500

Goal: Allocate resources optimally over all events over 7 days

What is optimal?
How make such decisions without complete information?

# Naïve strategy

- Rank by confidence “is kilonova”
  - Allocate full budget to most confident kilonova?
  - Allocate one of each type in order of decreasing confidence until exhausted?
  - Allocate to improve classification scores?
  - Allocate to improve light curve constraints?
  - Other?
- Decision hard because follow-up could be misallocated and this may be apparent with a delay
  - Perhaps it was better overall to rule out borderline cases and later allocate to best guess
- Ideal strategy is optimal given all future allocations and all future outcomes



P(KN)=0.73

P(KN)=0.12

P(KN)=0.11

30 hr per semester
g, r, i, z

1 per night
g, r

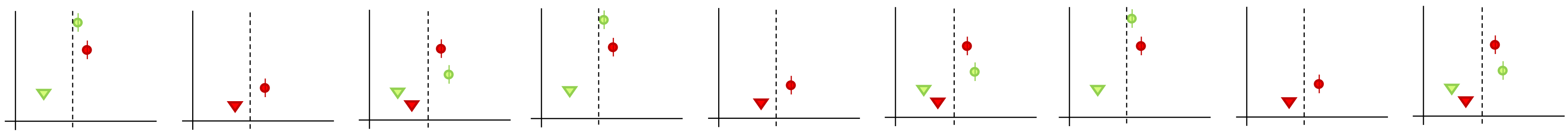
low res spec
~19mag
up to 5 per night

# Reinforcement learning: optimal sequential decision making under uncertainty

- Reinforcement learning agents learn through experience how, given a situation, taking an action now affects returns achieved later
- Formally, the control task needs to be formulated as a Markov Decision Process with a sequence of:
  - *states* – the world that the agent observes
  - *actions* – how an agent responds
  - *policy* – a function mapping the agent's observed state to a distribution of actions it takes
  - *rewards* – the utility of taking a given action in a given state
- RL considers the full distribution of outcomes at every timestep, and all future actions and their outcomes, and so on, and chooses the best action right now that maximizes an overall reward

# Pythia: a toy kilonova follow-up agent

Episodes end day 7



One per day

**Problem:**

9 transients, one of which (always) is the true kilonova (min photometry = 1)

- Contaminants are SNe, unassociated GRB afterglows, shock breakout (do not include observational significance)

Follow-up in ZTF g, r, or i (300s exposure) per day

- Finite horizon – 7 days (observe on day 1)

Reward 1 if agents adds data to the kilonova else 0

**Goal:** Maximize the number of follow-up to the true kilonova (non-model specific objective with the expectation that more data ~ better constraints)

# Algorithm

Need to generalize:
Infinite states!

- Learns online (collecting new experiences) in simulated environment
- Linear VFA (state-action value Q = x(s,a)$^T \omega$)
- x(s,a) is an CNN-autoencoder (for order invariance) representing the light curves with forecasted outcomes per action

On-policy

Bootstrap

**Algorithm** SARSA and TD(0) target

Initialize $w$ to small random weights
Set $\epsilon_0 = 1$
**for** k = 1, M **do** ▷ For each episode
    $\epsilon \leftarrow \epsilon_0 / k^n$
    Initialize $s_1$
    **for** t = 1, horizon **do**
        With probability $\epsilon$ select random action $a_t$
        otherwise select $a_t = \max_a \hat{Q}(s_t, a_t; \hat{w})$
        Execute action and observe reward $r_t$ and next state $s_{t+1}$ from environment
        With probability $\epsilon$ select random action $a_{t+1}$
        otherwise select $a_{t+1} = \max_a \hat{Q}(s_{t+1}, a_{t+1}; \hat{w})$
        Set $\Delta \hat{w} \leftarrow [r_t + \gamma \hat{Q}(s_{t+1}, a_{t+1}; \hat{w}) - \hat{Q}(s_t, a_t; \hat{w})] \nabla_w \hat{Q}(s_t, a_t; \hat{w})$
            ▷ : Loss is MSE between TD(0) target substitute for $Q^*$) and current $Q$
        Update $\hat{w} \leftarrow \hat{w} + \alpha \Delta \hat{w}$ ▷ $\alpha$ is using Adam
    **end for**
**end for**

x(s',a')$^T \omega$

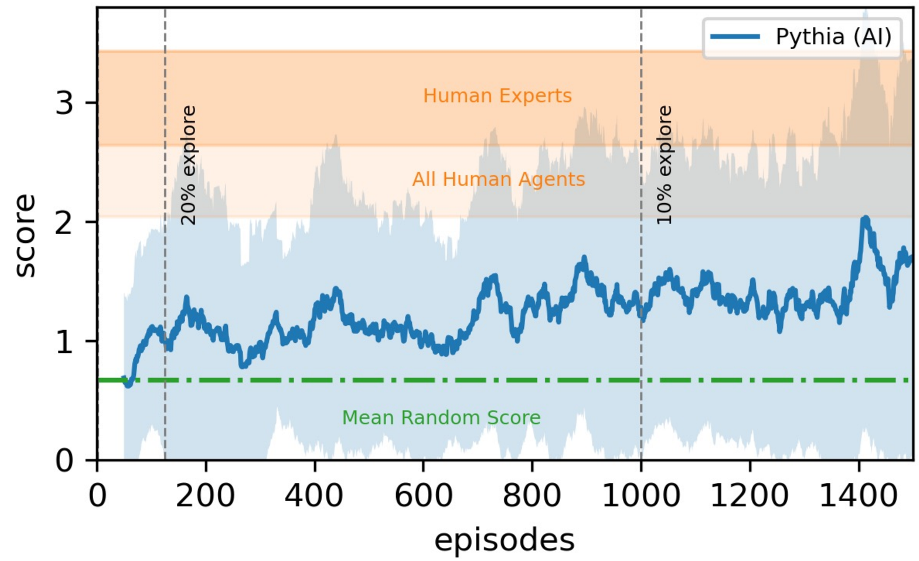
x(s,a)

Per timestep/episode means SGD

L = [Q* – Q(s,a; $\omega$)]$^2$
Q* unknown; sub as TD(0)

Remember gradient descent:
$\nabla \omega$ = - $\alpha \nabla$L

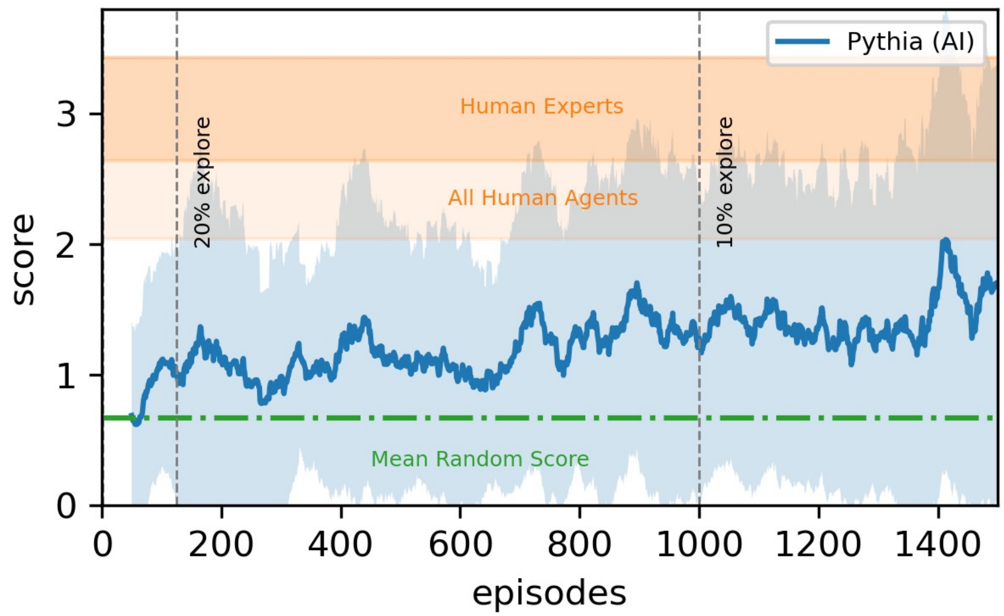
# Pythia




Sravan et al, submitted

- Linear VFA hypothesis class not sufficiently rich representation of true Q function
  - Benefit is theoretical convergence guarantee. Demonstrates problem learnable!
- Shifting to deep Q networks:
  - Will remove two-step learning, one for x(s,a) in supervised/unsupervised learning and one for Q via Bellman updates in RL
- Efficient evaluation of realistic large action space, can have vector instead of scalar output
- SARSA is known to be the weaker of the simple policies, specifically for short horizons with low penaties: QL is expected to perform better
- Use GNN instead of CNN for permutation invariance

# Pythia vs humans




Sravan et al, submitted

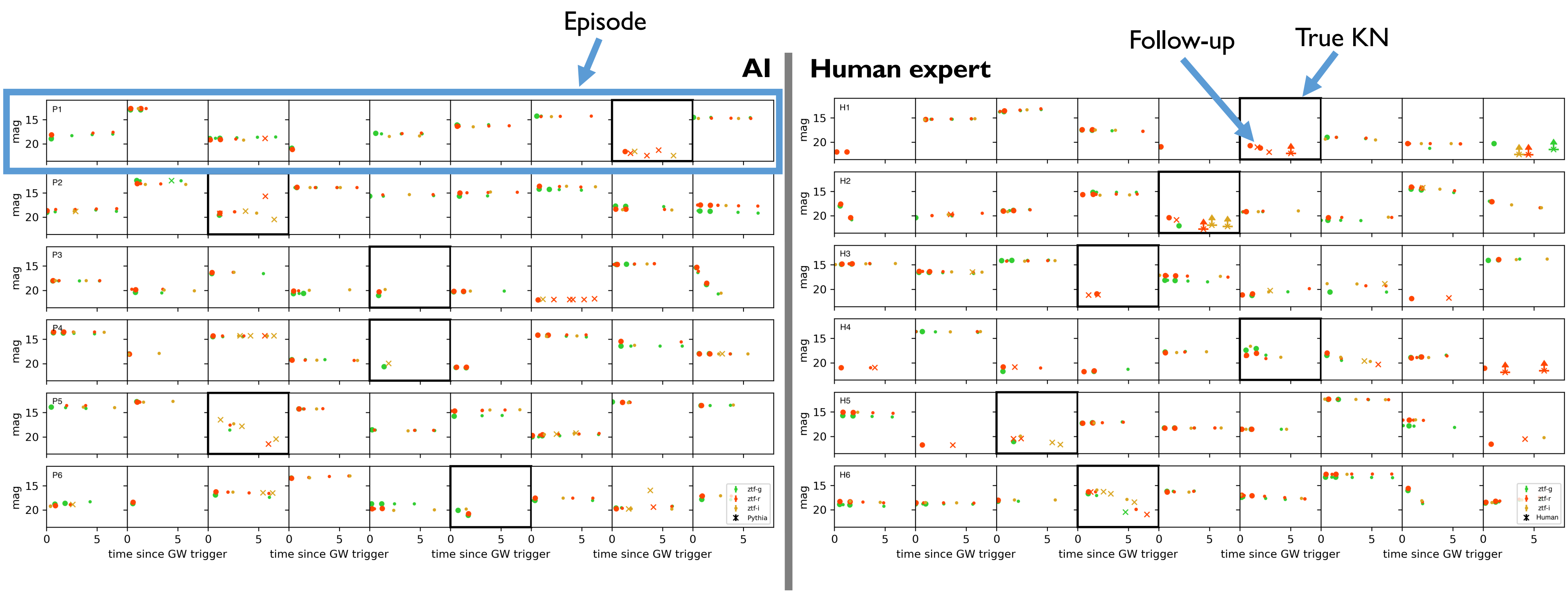
| agent | score | frac KN > 1 follow-up |
|---|---|---|
| Pythia | **1.84** | **0.81** |
| Non-expert 1 | 2.04 | 0.54 |
| Non-expert 2 | 3.15 | 0.86 |
| Expert 1 | 2.64 | 0.76 |
| Expert 2 | 2.74 | 0.78 |
| Expert 3 | 2.94 | 0.72 |
| Expert 4 | **3.43** | **0.9** |

# Pythia vs humans: random test episodes

Goodhart's law: "When a measure becomes a target, it ceases to be a good measure"

- AI prefers greedy when it fails since no benefit for exploring

Human experts have blindspots too



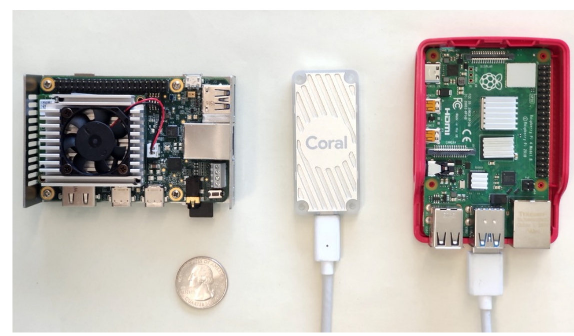

Sravan et al, submitted

# Fast Inferencing for Brokers

- LSST will produce ~$10^7$ alerts per night with a 60s latency from observation => up to ~6 Gbps
- Seven selected community alert brokers: Alerce (Chile), AMPEL (Germany), ANTARES (US), Fink (France), Lasair (UK), Pitt-Google (US), and:

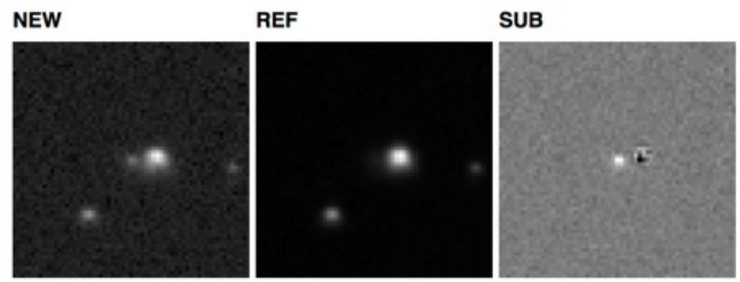
ZTF
BABA·MUL

- Fiducial models from ZTF using alert packets as input:
  - **braai** – real-bogus using VGG6
  - **acai** – 5-class classifier (hosted, orphan, nuclear, variable star, bogus) using a set of independent binary classifiers (CNNs)
  - Time series (RNNs)
- On Google Edge Corel TPU, braai hits 2000/s
  - 100x speed-up over beefy multi-core desktop for $100
- Now porting on FPGAs (Catherine Deng)

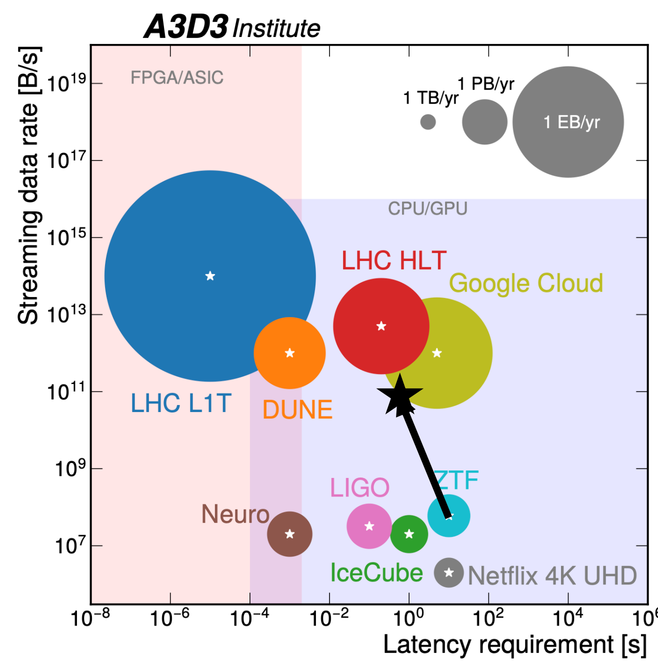


# Fast Inferencing for Brokers - II

- Current alert packets are ~80 kb in size but do we need all this data for the "first tier" models:
  - thumbnail images are sparse:
  - float64 parameter values
- Exploring lightweight reduced representation using autoencoders




- Next generation sky survey: cZTF
  - Replace CCDs with CMOS
  - 1s lucky imaging to probe the **fast** optical sky:
    - Stellar accretion sources
    - Blazars/AGN – jet physics and ISCO events
  - Need fast ML to do *any* science



# Summary

- First AI agent capable of strategizing a sequential transient follow-up
- The problem is learnable by machines and already comes close to human performance with toy solution
- More complex agents to deal with more realistic issues
- Training is expensive ($CO_2$-equivalent to LAX – JFK return flight)
- Larger data volumes/rates require faster/more efficient ML solutions
- Motivates effective low-latency use of expensive space-based follow-up resources