# Kepler’s Sieve

## Learning Asteroid Orbits
## from Telescopic Observations

Masters of Data Science Thesis

Michael S. Emanuel

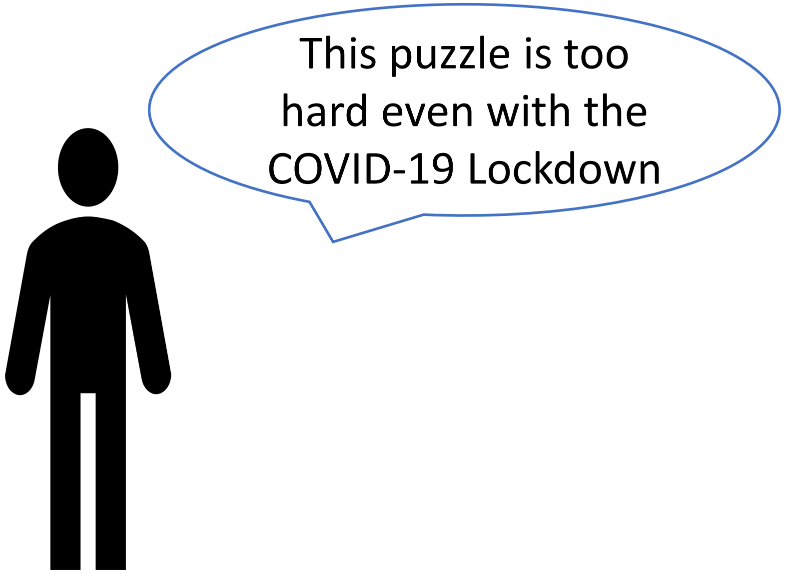
# Acknowledgments


- Advisor: Pavlos Protopapas
- Secondary Advisor: Chris Rycroft

# Introduction

# The Asteroid Search Problem

- Many asteroids (about 958,000 known) in the Solar System
- We want to learn their orbits
- Biggest data source: telescope detections
- Easy once you know which detection matches which asteroid
- This is like a jigsaw puzzle with millions of pieces!



# Combining Tracklets vs. Orbital Element Search

- "Tracklet": two detections close to each other in time and direction
- Existing search methods: greedy search over tracklets
  - Try to extrapolate a tracklet to find additional detections
  - Attempt to fit an orbit when you have enough tracklets
- Drawbacks
  - Myopic – can only connect detections made close in time
  - Suffers from combinatorial explosion
- Proposed novel method: search Orbital Elements
  - 6D space; large, but scales well
  - Cost scales as $N_{\mathrm{ast}} \cdot N_{\mathrm{obs}}$ rather than $N_{\mathrm{obs}}^{r}$
  - But can we make it  work?

# Search Overview

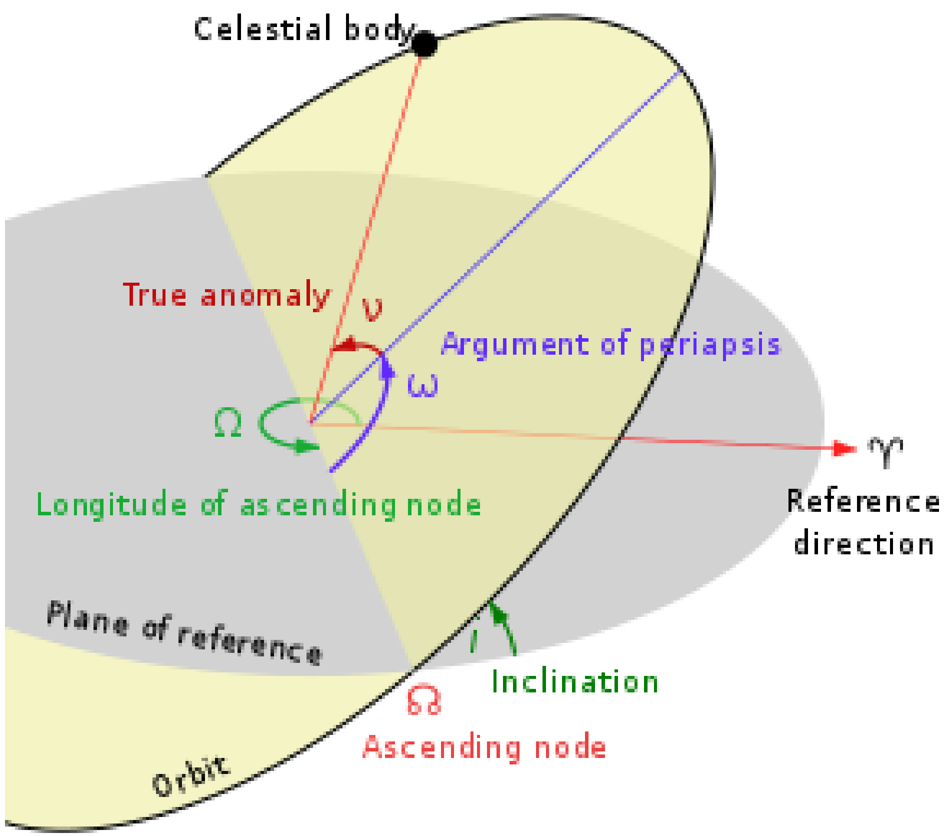
- Initialize candidate orbital elements *a*, *e*, *i*, $\Omega$, $\omega$, *f*
- Mixture parameters: $N_h$, R, $\tau$
- Compute position **q** and velocity **v** from candidate elements
- Compute direction $\mathbf{u}_{pred}$ from **q**, **v;** include light time and topos
- Compute distance s from $\mathbf{u}_{pred}$ to $\mathbf{u}_{obs}$ for ZTF observations
- Compute log likelihood $\mathscr{L}_i$ for each candidate element
- Gradient descent…
- The rest is details! Which you will now hear all about…

# Integrating the Solar System

# REBOUND Integrator for N-Body Problem

- REBOUND is a modern, open source integrator
  - github.com/hannorein/rebound
- It numerically solves the gravitational N-body problem
- Considered the “gold standard” for orbits in this thesis
- IAS15 adaptive integrator uses Gauss-Radau quadrature and a “predictor-corrector” scheme
- Horizons: API provided by NASA JPL to obtain state vectors (position and velocity) of objects in the Solar System

# Keplerian Orbital Elements



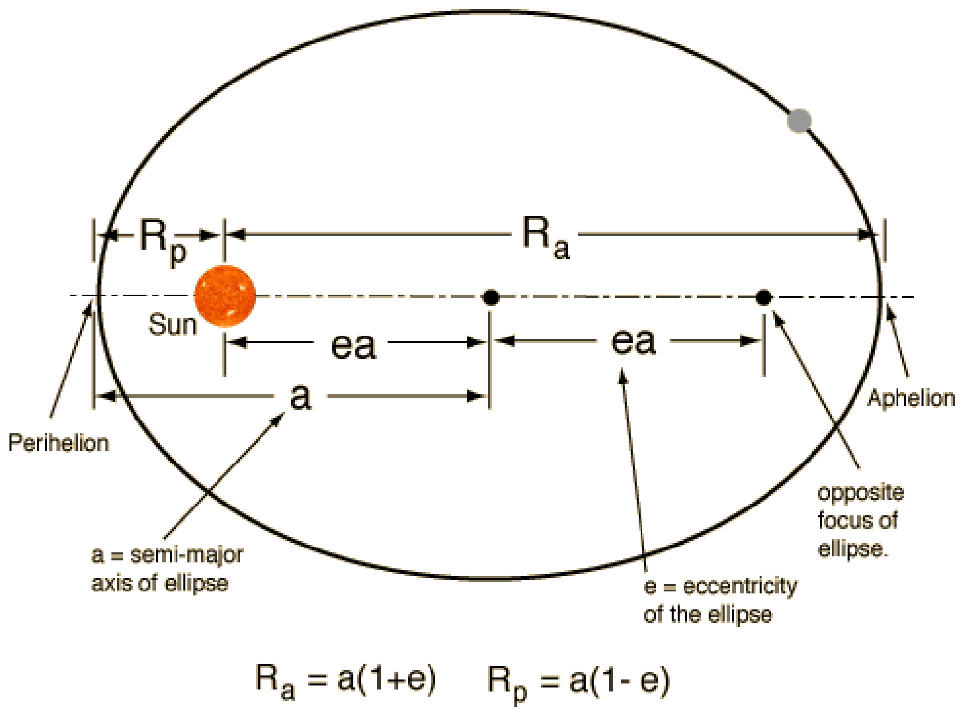
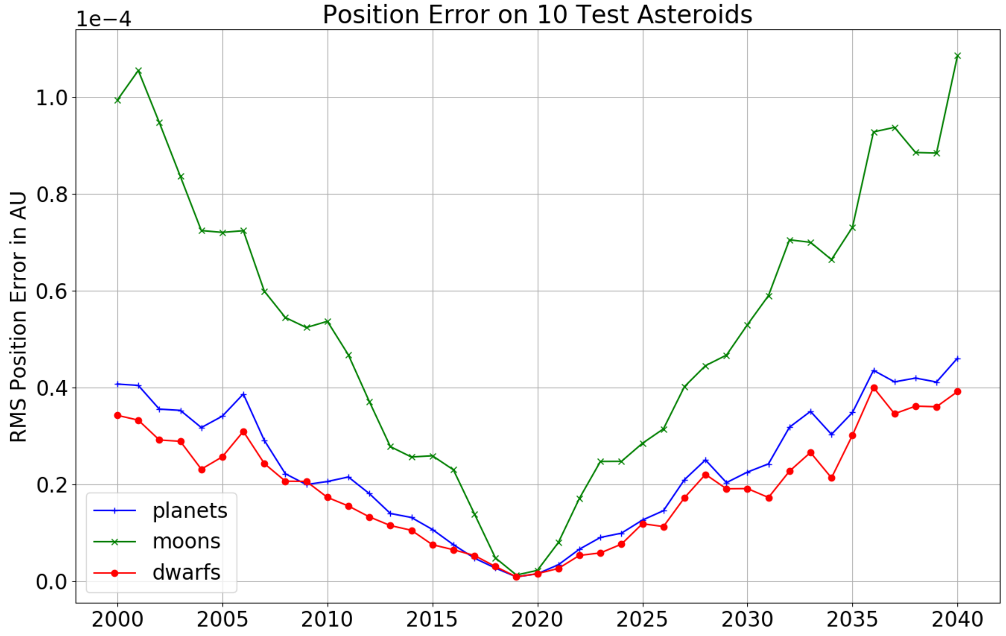



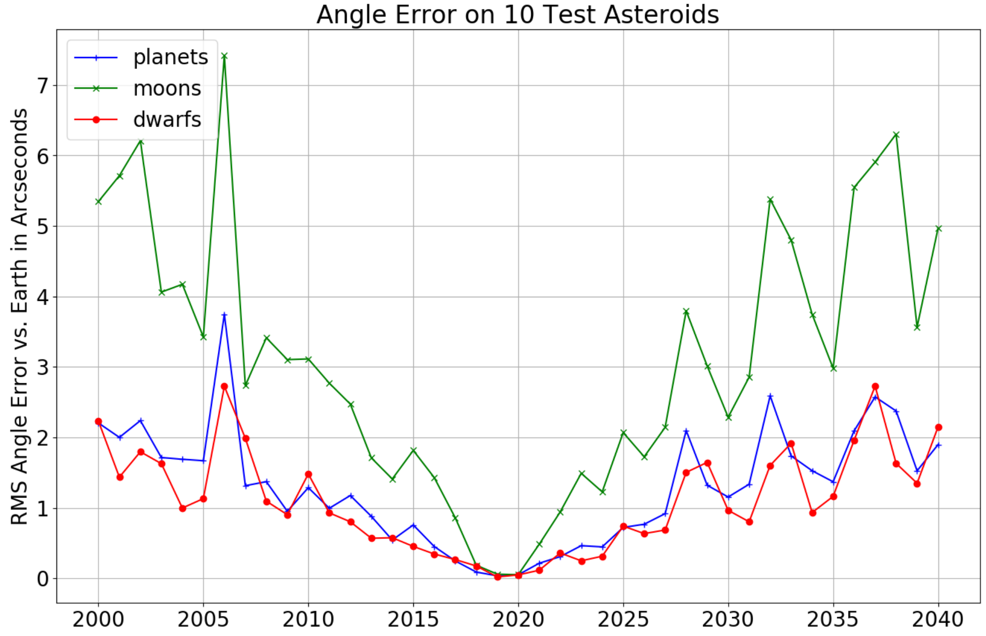

Image Credits: WikiPedia, Cool Cosmos

- Semi-major axis *a* and eccentricity *e* describe the size and shape of the orbital ellipse
- Inclination *i*, ascending node Ω, perihelion ω are angles orienting orbit in the ecliptic plane
- True anomaly *f* is location of the body on its orbital ellipse

# Validating Integration vs. Horizons

- Integrate three collections of massive bodies for 40 years at daily interval
- Initial conditions from Horizons at MJD 58600 / 2019-04-27
- Test results on first 10 IAU asteroids; query their positions from Horizons
- Report error in position (AU) and instantaneous angle from asteroid to Earth (arc seconds)
- Accuracy is excellent!
    - RMS error on planets is 5.4E-6 AU
    - Angle error from asteroids to planets 0.8 arc seconds



# Bulk Integration of 733,489 Asteroids

- Download asteroid orbital elements from JPL
- Data available for 733,489
- Integrate these daily for 40 years
- Save results to disk
  - REBOUND simulation archives
  - Numpy arrays
- Job takes 4:30 on 40 CPU cores
- Writes 1.37 TB output to disk

```
# Load all the asteroid elements
ast_elt = load_ast_elt()
```

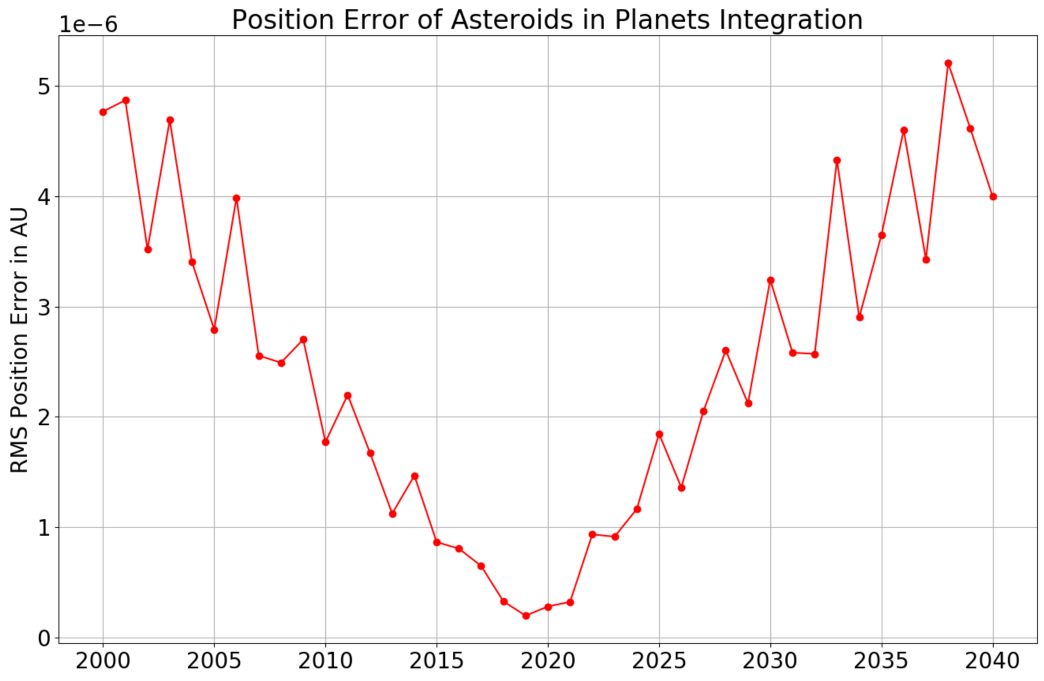
```
ast_elt
```

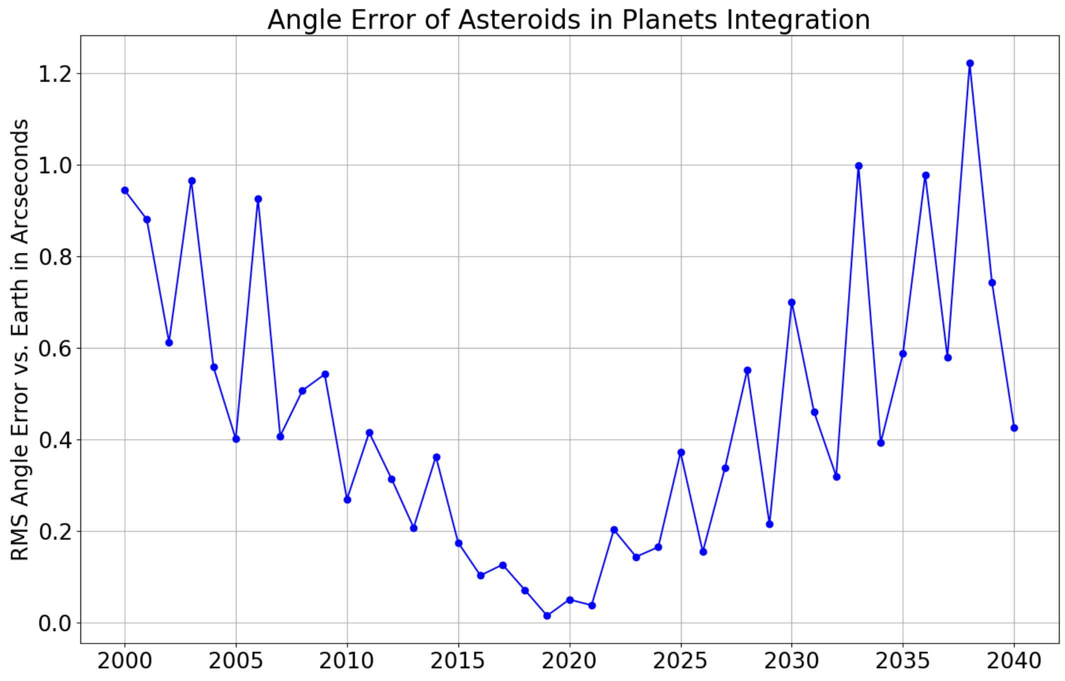
| Num | Num | Name | epoch | a | e | inc | Omega | omega | M | H | G | Ref | f |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **1** | 1 | Ceres | 58600.0 | 2.769165 | 0.076009 | 0.184901 | 1.401596 | 1.284522 | 1.350398 | 3.34 | 0.12 | JPL 46 | 1.501306 |
| **2** | 2 | Pallas | 58600.0 | 2.772466 | 0.230337 | 0.608007 | 3.020817 | 5.411373 | 1.041946 | 4.13 | 0.11 | JPL 35 | 1.490912 |
| **3** | 3 | Juno | 58600.0 | 2.669150 | 0.256942 | 0.226699 | 2.964490 | 4.330836 | 0.609557 | 5.33 | 0.32 | JPL 108 | 0.996719 |
| **4** | 4 | Vesta | 58600.0 | 2.361418 | 0.088721 | 0.124647 | 1.811840 | 2.630709 | 1.673106 | 3.20 | 0.32 | JPL 34 | -4.436417 |
| **5** | 5 | Astraea | 58600.0 | 2.574249 | 0.191095 | 0.093672 | 2.470978 | 6.260280 | 4.928221 | 6.85 | 0.15 | JPL 108 | -1.738676 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **1255499** | 1255499 | 2019 QG | 58600.0 | 0.822197 | 0.237862 | 0.220677 | 5.066979 | 3.770460 | 0.503214 | 21.55 | 0.15 | JPL 1 | 0.807024 |
| **1255501** | 1255501 | 2019 QL | 58600.0 | 2.722045 | 0.530676 | 0.113833 | 4.741919 | 2.351059 | 5.297173 | 19.21 | 0.15 | JPL 1 | -2.082964 |
| **1255502** | 1255502 | 2019 QQ | 58600.0 | 1.053137 | 0.389091 | 0.172121 | 5.648270 | 2.028352 | 3.266522 | 25.31 | 0.15 | JPL 1 | -3.081905 |
| **1255513** | 1255513 | 6331 P-L | 58600.0 | 2.334803 | 0.282830 | 0.141058 | 6.200287 | 0.091869 | 2.609695 | 18.50 | 0.15 | JPL 8 | 2.827595 |
| **1255514** | 1255514 | 6344 P-L | 58600.0 | 2.812944 | 0.664688 | 0.081955 | 3.199363 | 4.094863 | 2.738525 | 20.40 | 0.15 | JPL 17 | 3.032066 |

733489 rows × 19 columns

# Validate Asteroid Integration vs. Horizons

- Test bulk asteroid integration on first 25 IAU asteroids
- Report position error in AU and angle error to Earth in arc seconds
- Excellent results! RMS 2.49E-6 AU and 0.45 arc seconds



# Integrate Kepler Two Body Problem in TensorFlow

- Analytical solution to Kepler problem is an ellipse
- 5 of the 6 orbital elements *a*, *e*, *i*, Ω, ω constant
- The Mean Anomaly *M* is linear in time (2nd Law)

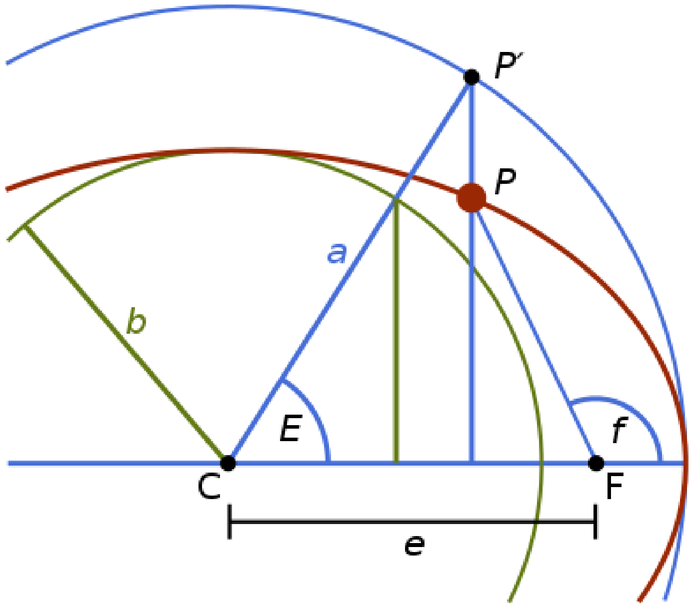
$$r(\theta) = \frac{a \cdot (1 - e^2)}{1 - e \cdot \cos(\theta - \theta_0)}$$

$$M(t) = M_0 + N \cdot (t - t_0)$$

- Kepler's Equation relates orbital anomalies:

$$M = E - e \sin(E) \qquad \text{Kepler's Equation}$$

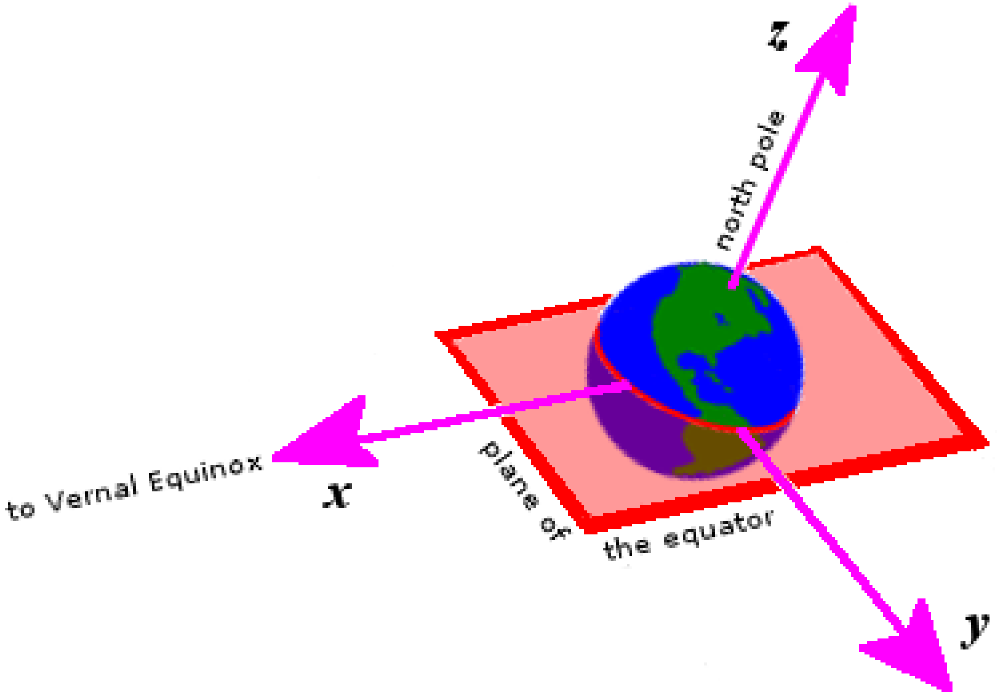
$$\tan\left(\frac{f}{2}\right) = \sqrt{\frac{1+e}{1-e}} \cdot \tan\left(\frac{E}{2}\right) \qquad \text{true to eccentric}$$

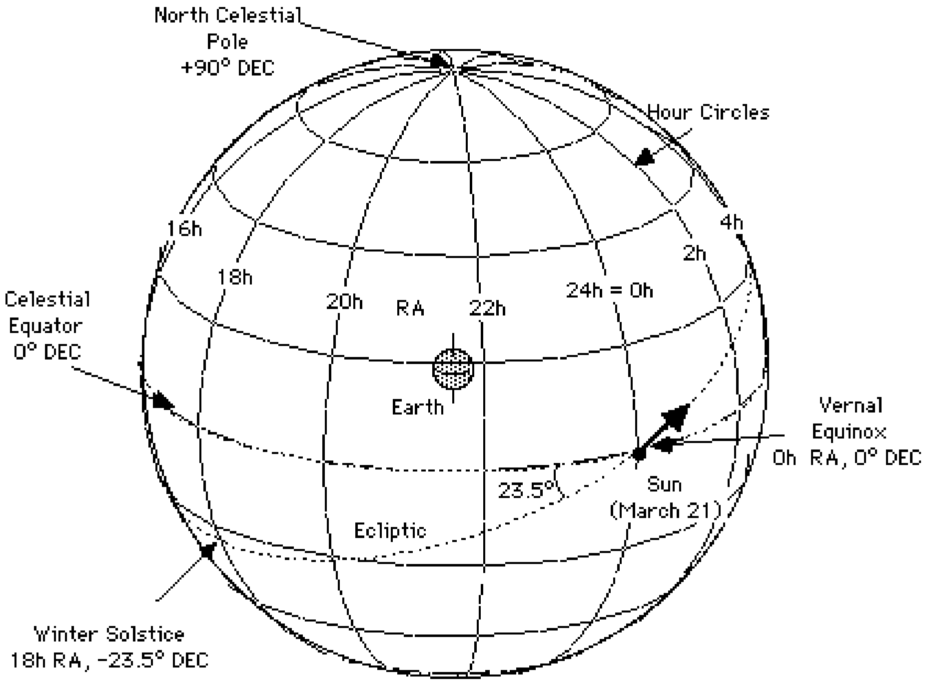



- Convert *M* to *E* to *f*, then to Cartesian coordinates
- TensorFlow is fast! 5000 time points in ~300 μ sec
- Apply calibration d**q**, d**v** to match REBOUND integration at input orbital elements

# Predicting Directions from Positions

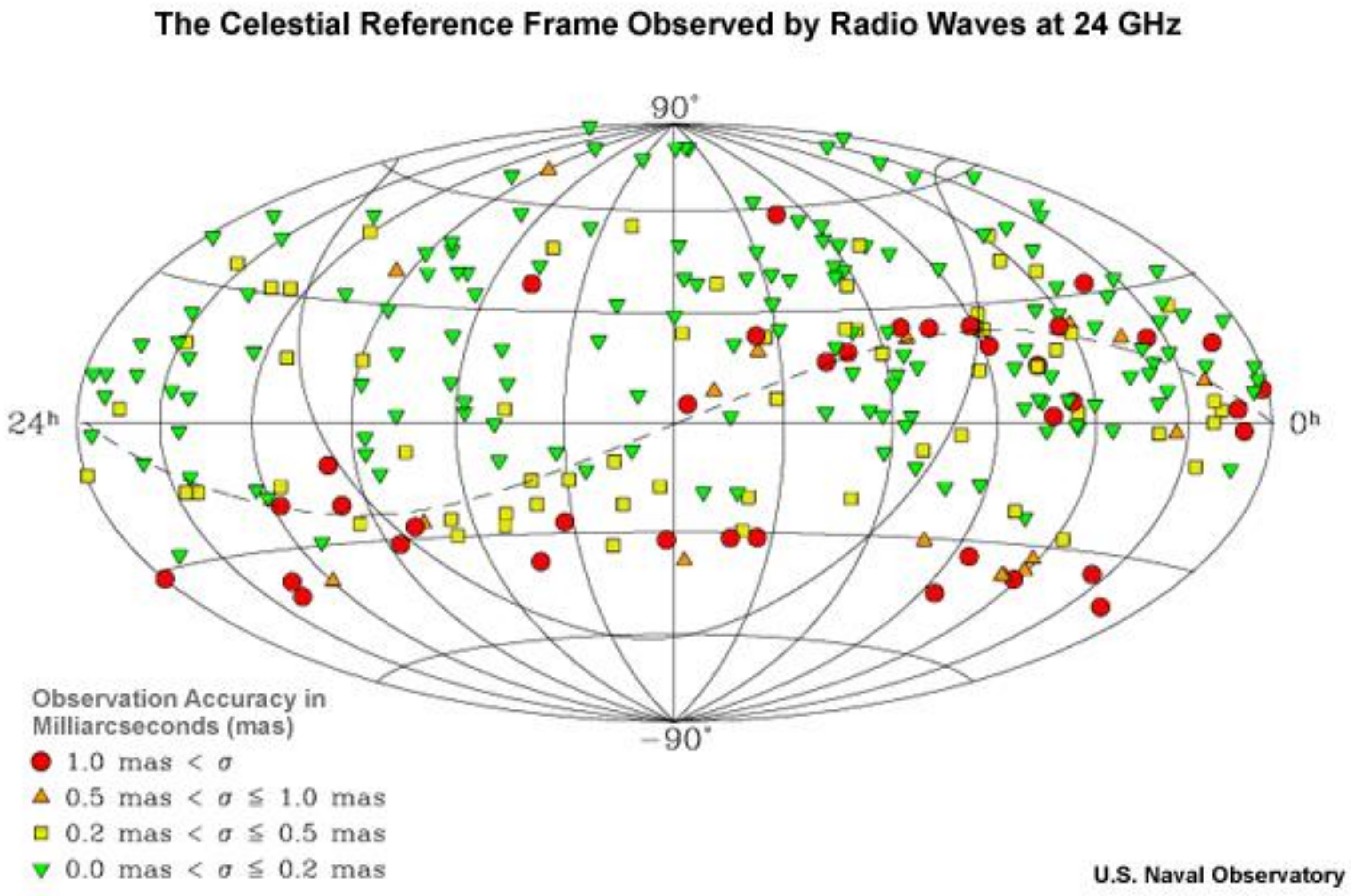
# Right Ascension and Declination



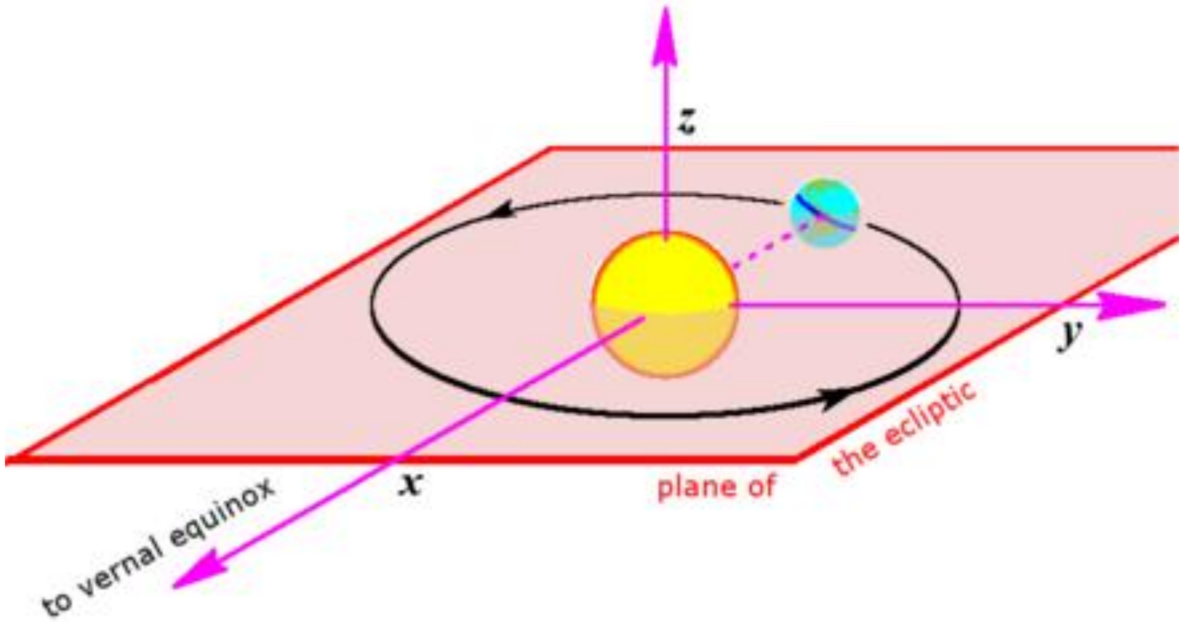

- Fundamental plane is aligned with Earth's equator
- Intuitive, dates to ancient astronomers
- Two problems: precession (drift) and nutation (wobbles) in direction of North Pole

# International Celestial Reference Frame (ICRF)




- Modern system for RA/Dec
- Based on 232 extragalactic objects
- Addresses precession and nutation
  - Quasars don't move!
  - Not in **direction** anyway
- Amazingly accurate
  - ~2 milliarc-seconds
- Intuition: like using Polaris instead of Earth's axis for the North Pole
- Except you use 232 stars to get a highly accurate composite direction

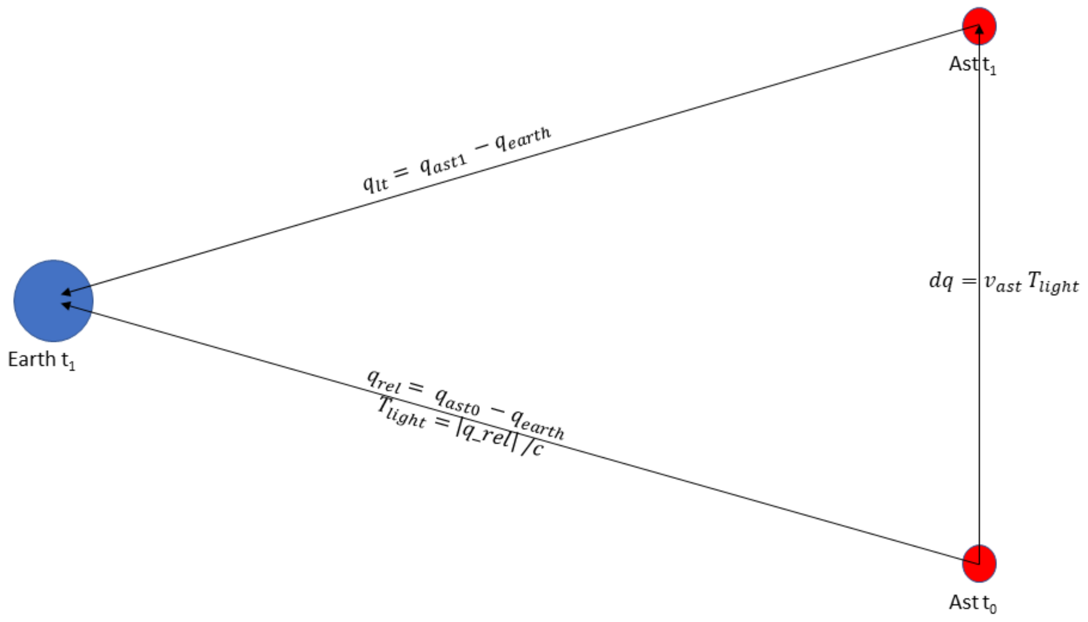
# From RA/Dec to Barycentric Mean Ecliptic




- RA/Dec is ideal for observing the stars
- But not for calculations involving orbits in the Solar System
- Inside the Solar System we want an inertial frame aligned with the ecliptic: BME
- Convert between ICRF and BME using `astropy` library

```
obs_icrs = astropy.SkyCoord(ra=ra, dec=dec, obstime=obstime, frame=ICRS)
obs_ecl = obs_icrs.transform_to(BarycentricMeanEcliptic)
```

# Calculate Direction from Position and Velocity




- Need to remember light speed *c* is finite!
- Otherwise wrong by ~285 arc seconds

$$\mathbf{q}_{\mathrm{rel}} = \mathbf{q}_{\mathrm{ast}} - \mathbf{q}_{\mathrm{earth}}$$
$$T_{\mathrm{light}} = \left\|\mathbf{q}_{\mathrm{rel}}\right\| / c$$
$$\Delta\mathbf{q}_{\mathrm{ast}} = \mathbf{v}_{\mathrm{ast}} \cdot T_{\mathrm{light}}$$
$$\mathbf{q}_{\mathrm{lt}} = \mathbf{q}_{\mathrm{rel}} - \Delta\mathbf{q}_{\mathrm{ast}}$$
$$\mathbf{u} = \mathbf{q}_{\mathrm{lt}} / \left\|\mathbf{q}_{\mathrm{lt}}\right\|$$

- Earth velocity doesn't matter, only asteroid velocity
- BME is an inertial frame

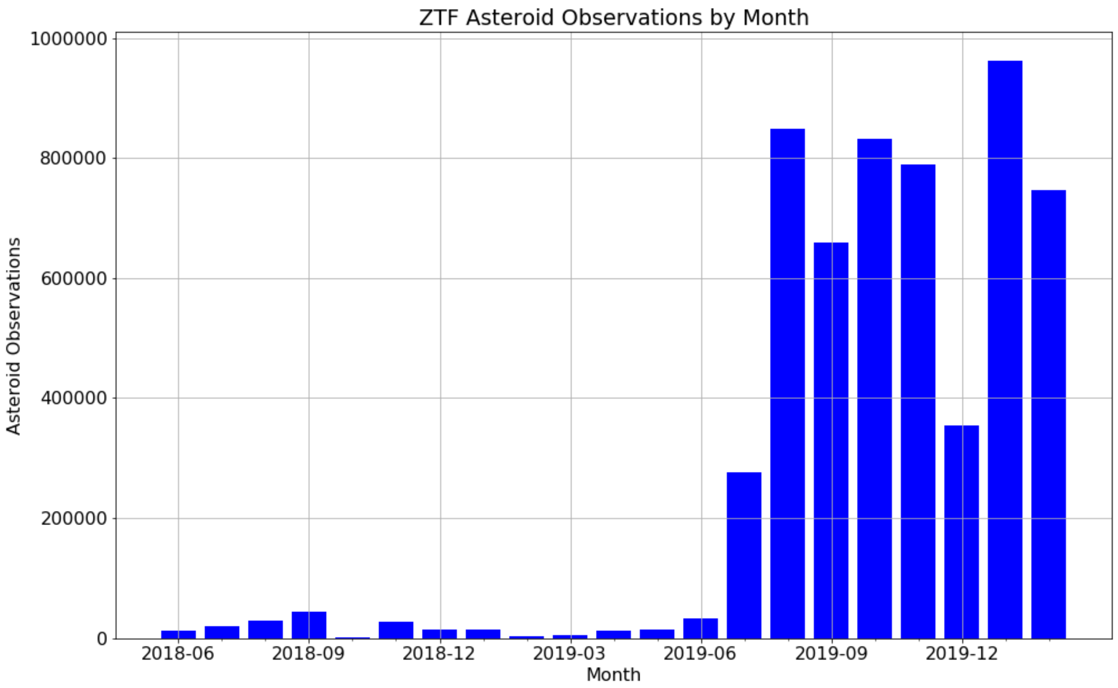
- Also need "topos adjustment" for observatory: Palomar Mountain, not geocenter!
- Topos adjustment worth 0-5 arc seconds on first 16 asteroids

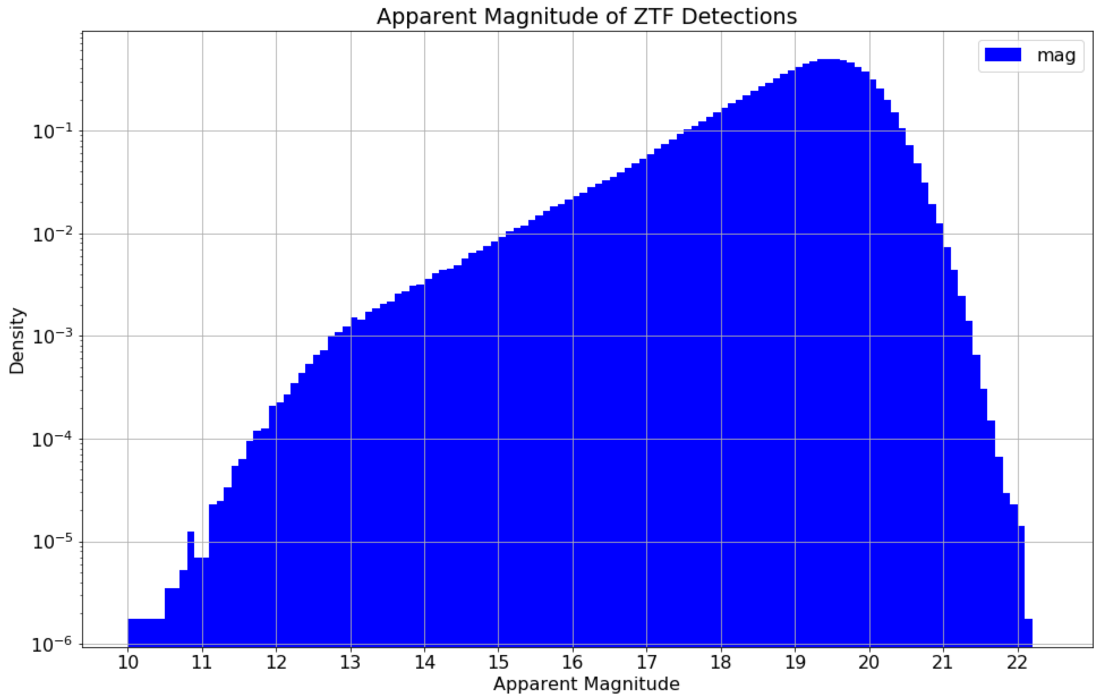
# Validating Astrometric Direction

- Check these results by comparing vs. JPL, SkyField
- Downloaded Mars at 3 hour intervals over 10 years (~29000 rows)
  - Both state vectors ***q***, ***v*** and observer RA / Dec
- Computed astrometric directions from Earth to Mars
  - MSE and SkyField identical: 0.027 arc seconds
  - Both MSE and SkyField differ from JPL by 1.6 arc seconds
- Separately downloaded JPL RA/Dec on first 16 asteroids
- Compared to MSE direction calculated from integrated orbits
- RMS error: 0.873 arc seconds!

# Analysis of ZTF Asteroid Detections

# EDA of ZTF Detections



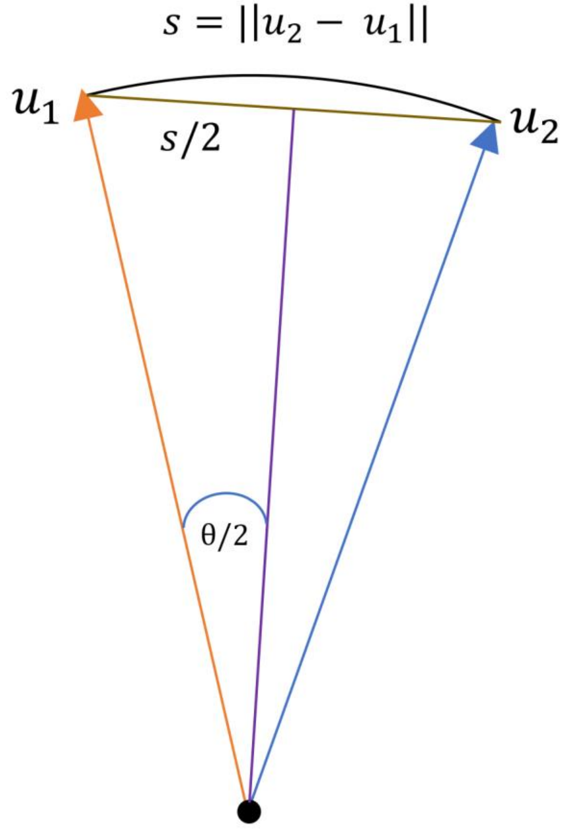




- ZTF: Zwicky Transient Facility; survey of northern sky at Palomar Mountain by Cal Tech
- Fast, deep survey: 3750 square degrees / hour to depth of 20.5 mag
- First light in 2017, but asteroid detections ramp up in July 2019; 7 months of data
- Enriched with machine learning pipeline that filters probable asteroid detections
- 5.69 million possible asteroid detections
- Data includes: MJD, RA, DEC, MAG

# Converting Cartesian to Angular Distance

- How far apart are two directions in the sky?
- Convert RA/Dec to directions $u_1$ and $u_2$ in the BME
- Compute Cartesian distance s between $u_1$ and $u_2$
- Angular distance $\theta$ is geodesic (great circle distance)

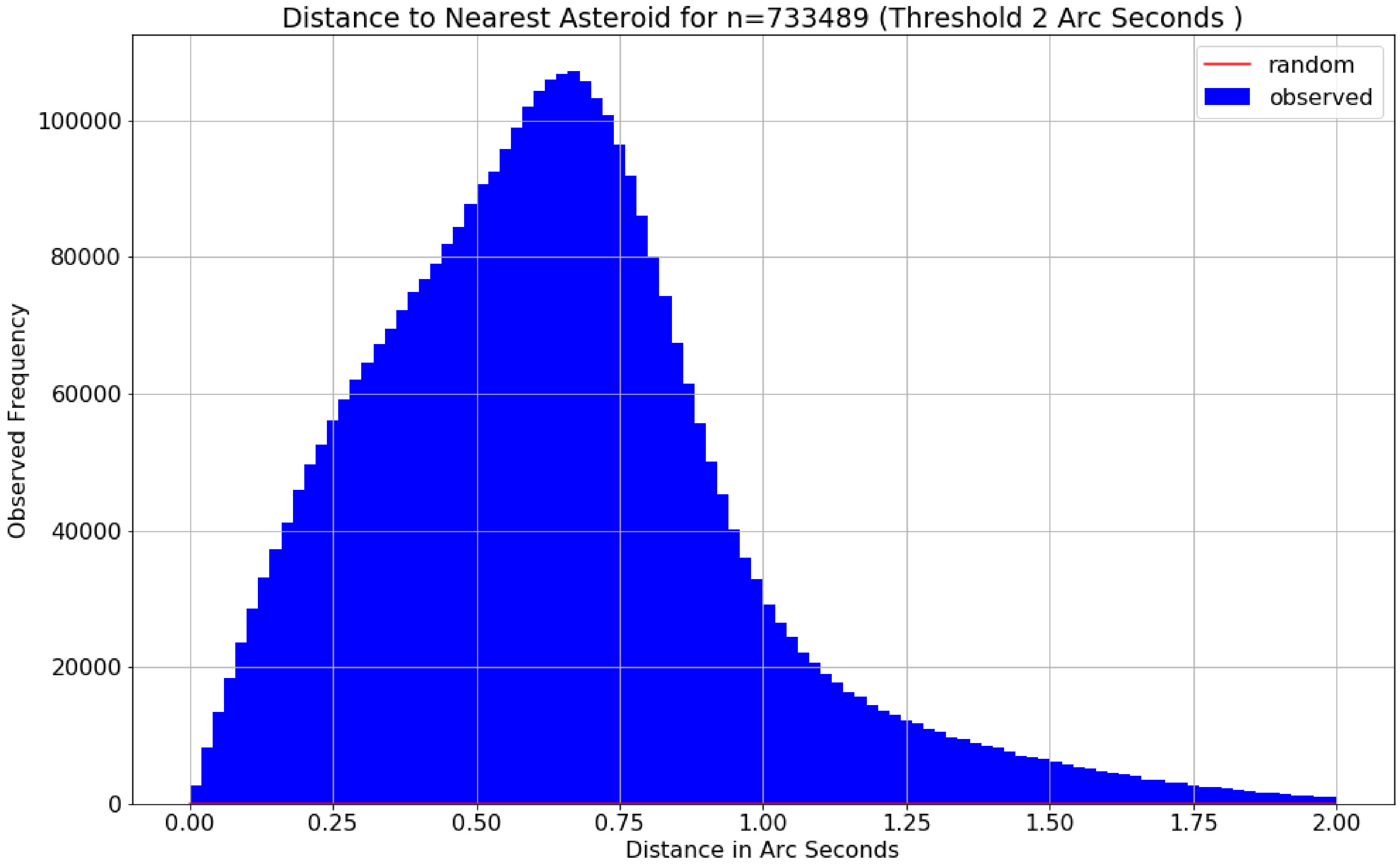
$$\sin(\theta/2) = s/2$$



# Nearest Asteroid to Each ZTF Detection

- Compute direction $u_{obs} = (u_x, u_y, u_z)$ from RA/Dec for each detection
- Compute direction $u_{ast}$ for every asteroid in the catalogue
- 5.7E6 detections x 7.3E5 asteroids = 4.2E12 (4.2 trillion) interactions!
  - Too big for naïve brute force attack
- “Only” 97,111 different MJDs with ZTF detections
- Work in chunks of 1000 asteroids at a time, find nearest to each ZTF
- Then perform reduction operation to find globally nearest asteroid
- Still large compute job: 25 hours on 40 CPUs, 256 GB RAM server

# Nearest Asteroid: 65.7% Within 2.0 Arc Seconds!



# Statistical Distribution of Distance on Sphere

- What is the statistical distribution of s if we guessed directions uniformly at random?

$$s^2 = 2 \cdot (1 - z) \qquad z = 1 - s^2/2$$

- This is useful parameterization because…
- "Orange Slicing Theorem" for solid angle measure:

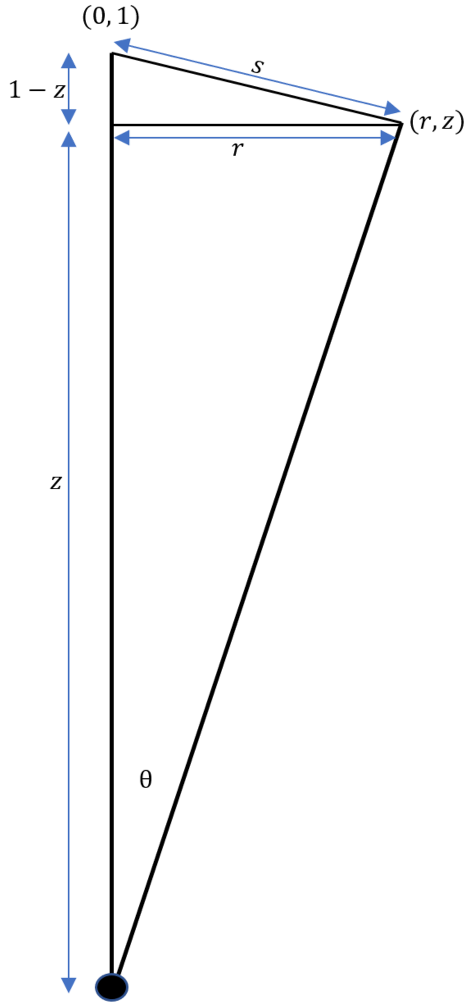
$$d\Omega = dz \cdot d\phi$$

- Think of Z and S as random variables:

$$Z \sim \text{Unif}(-1, 1) \quad S^2 \sim \text{Unif}(0, 4)$$

- Conditional on a max (threshold) distance τ

$$S^2 | S \leq \tau \sim \text{Unif}(0, \tau^2)$$



# Distribution of Nearest Asteroid Distance

- Set a threshold distance τ and define relative squared distance V

$$V = (S/\tau)^2 \qquad V \sim \mathrm{Unif}(0,1)$$

- We have n = 733,489 guesses and are picking closest

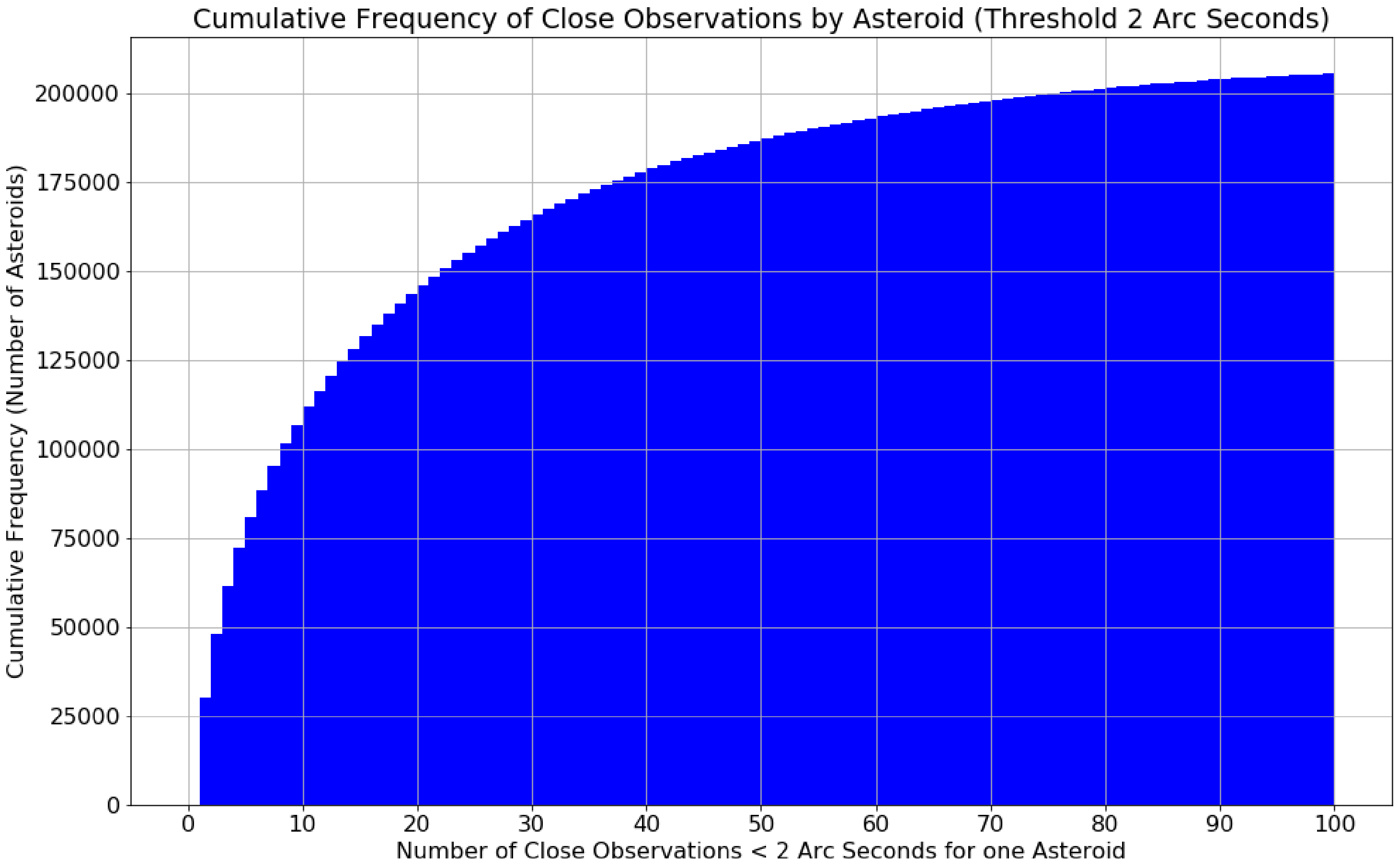
$$V_1, \ldots V_n \overset{i.i.d.}{\sim} \mathrm{Unif}(0,1)$$

- Stat 110: The minimum of n i.i.d. uniforms has a Beta distribution

$$U_{(1)} \sim \mathrm{Beta}(1, n)$$

- How many hits at 2.0 arc seconds would we get by luck?
- Only 98. But we got 3.75 million of them!
- Conclusion: This whole apparatus works to a tolerance of 2.0 arc seconds

# Cumulative Distribution of Hits per Asteroid




- Suppose we want to rebuild the asteroid catalog…
- How many asteroids do we have a sporting chance of finding?
- Count hits at 2.0 arc seconds
- How many asteroids have at least
  - 20 hits? 63,746
  - 10 hits? 100,508
- We have a shot at 13.6% of the catalogue if we require 10+ hits

# Asteroid Search Using Orbital Elements

# Assemble ZTF Detections Near Elements

```
# Load unperturbed element batch
ztf_elt_ast = load_ztf_batch(elts=elts_ast, thresh_deg=1.0, near_ast=False)
```

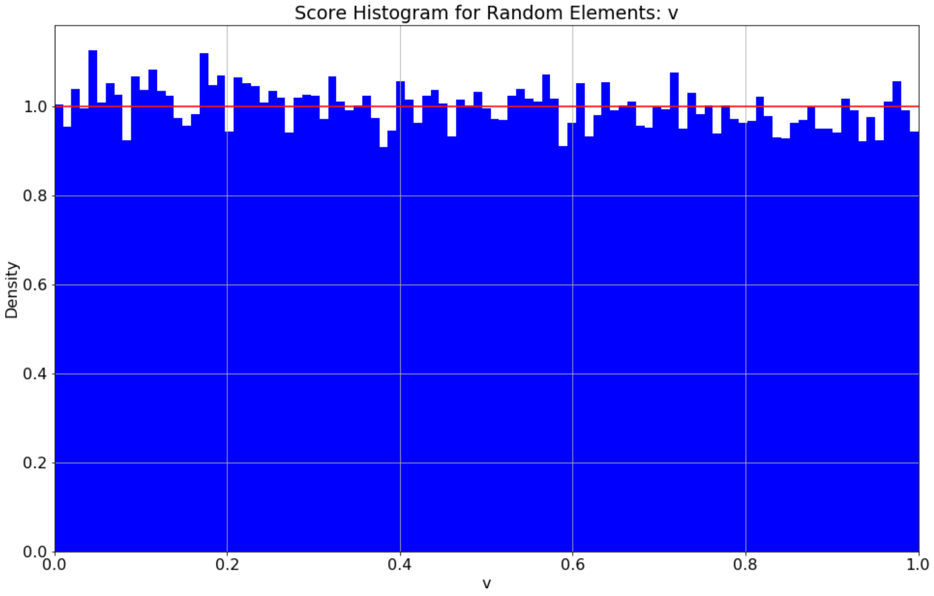
```
# Review
ztf_elt_ast[cols]
```

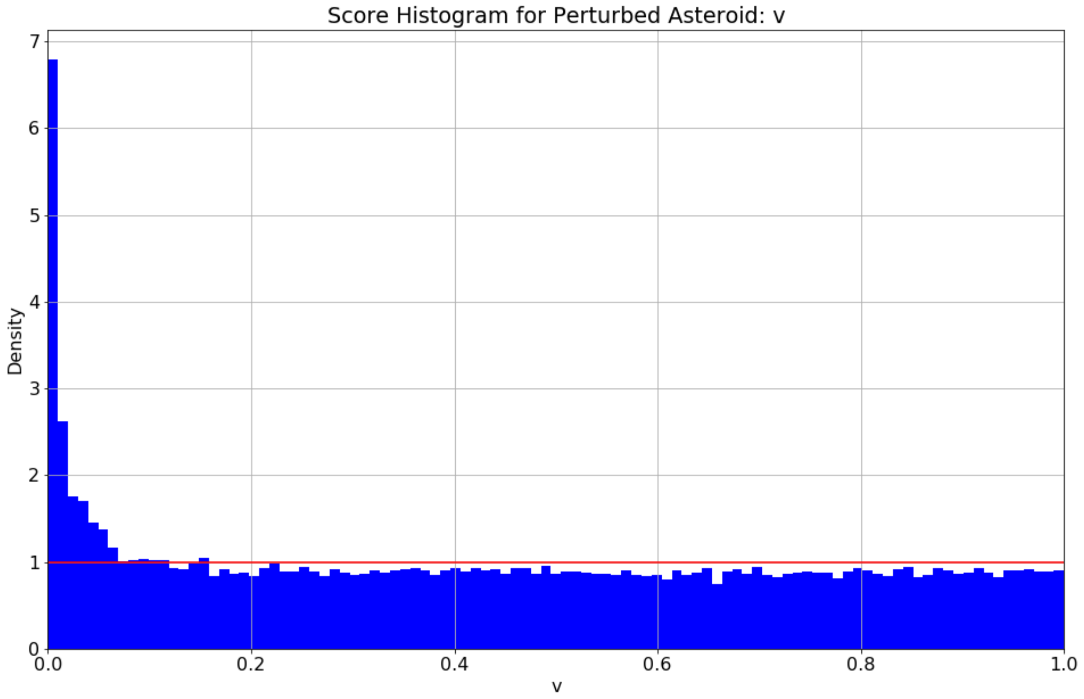
| | element_id | ztf_id | mjd | ra | dec | ux | uy | uz | mag_app | elt_ux | elt_uy | elt_uz | s_sec | v | is_hit |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 733 | 53851 | 58348.197581 | 266.229165 | -13.513802 | -0.063945 | -0.983101 | 0.171530 | 16.755600 | -0.057300 | -0.982042 | 0.179751 | 2191.408734 | 0.370552 | False |
| **1** | 733 | 73604 | 58348.197581 | 265.761024 | -13.509148 | -0.071871 | -0.982578 | 0.171389 | 16.035999 | -0.057300 | -0.982042 | 0.179751 | 3467.151428 | 0.927559 | False |
| **2** | 733 | 82343 | 58389.193252 | 270.331454 | -11.244934 | 0.005674 | -0.977422 | 0.211222 | 17.196199 | 0.000919 | -0.977996 | 0.208622 | 1124.103915 | 0.097503 | False |
| **3** | 733 | 257221 | 58685.471227 | 29.693832 | 42.180412 | 0.643725 | 0.603886 | 0.470042 | 19.289200 | 0.639004 | 0.610779 | 0.467571 | 1797.091521 | 0.249197 | False |
| **4** | 733 | 327000 | 58691.465972 | 33.104905 | 44.059131 | 0.601970 | 0.636719 | 0.481893 | 17.725201 | 0.606278 | 0.637608 | 0.475272 | 1639.539679 | 0.207419 | False |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **90206** | 324582 | 5650588 | 58904.176701 | 44.164238 | 29.650540 | 0.623416 | 0.752309 | 0.213037 | 18.084700 | 0.627640 | 0.750696 | 0.206212 | 1688.638104 | 0.220027 | False |
| **90207** | 324582 | 5650589 | 58904.176250 | 44.164062 | 29.650536 | 0.623417 | 0.752307 | 0.213038 | 18.165199 | 0.627641 | 0.750695 | 0.206213 | 1688.601889 | 0.220018 | False |
| **90208** | 324582 | 5650665 | 58904.176250 | 44.368640 | 28.490480 | 0.628284 | 0.753618 | 0.193182 | 19.025200 | 0.627641 | 0.750695 | 0.206213 | 2757.856412 | 0.586871 | False |
| **90209** | 324582 | 5650697 | 58904.176250 | 43.296207 | 29.505908 | 0.633424 | 0.743491 | 0.214467 | 19.852800 | 0.627641 | 0.750695 | 0.206213 | 2555.278205 | 0.503822 | False |
| **90210** | 324582 | 5650705 | 58904.176250 | 44.621045 | 29.303550 | 0.620689 | 0.756675 | 0.205398 | 19.647400 | 0.627641 | 0.750695 | 0.206213 | 1898.912116 | 0.278236 | False |

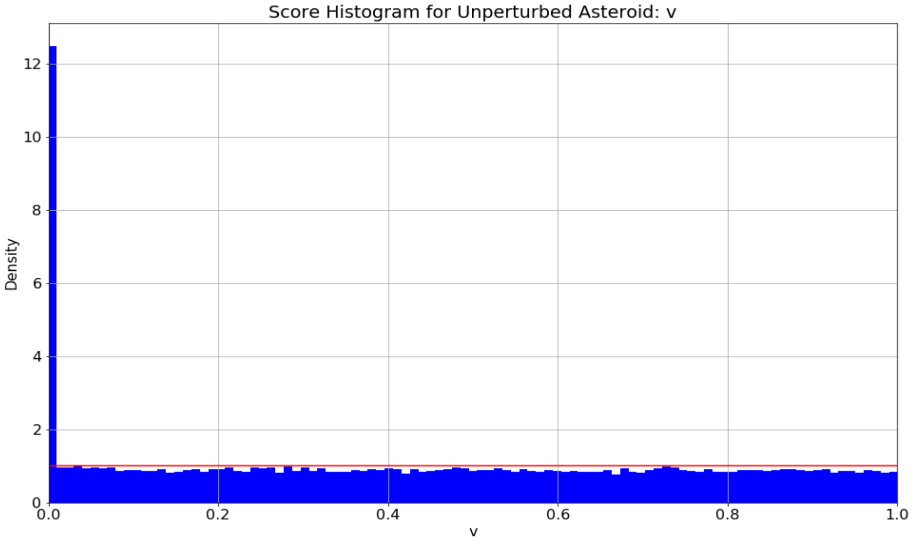
90211 rows × 15 columns

- Integrate the candidate elements on the fly in REBOUND and compute directions
- Filter the ZTF detections to those within threshold of the elements

# Distribution of $V = (S/\tau)^2$ for 3 Element Batches




- Plot V for 3 batches: random, unperturbed, perturbed
- Results match theory perfectly!
- Random elements close to uniform distribution
- Unperturbed: uniform on misses with spike in first bucket
- Perturbed: in between; hits leak out to ~250 arc seconds

# Log Likelihood Objective Function

- Mixture probability model: V mixture of h hits, (1-h) misses

$$V|\text{Hit} \sim \text{Expo}(\lambda) \qquad V|\text{Miss} \sim \text{Unif}(0,1)$$

- Relate decay rate to "resolution" parameter R

$$f(v) \propto e^{-\lambda v} = e^{-\lambda s^2/\tau^2} \qquad f(v) \propto e^{-s^2/2R^2} \qquad \lambda = \frac{\tau^2}{2R^2}$$

- The resolution R controls how tightly the model focuses
- Mixture PDF:

$$h \cdot \frac{\lambda \cdot e^{-\lambda v}}{1 - e^{-\lambda}} + (1-h)$$

- Log Likelihood:

$$\mathcal{L}(\mathbf{v}, h, \lambda) = \sum_{j=1}^{n} \log\left(h_j \cdot \frac{\lambda \cdot e^{-\lambda_j v_j}}{1 - e^{-\lambda_j}} + 1 - h_j\right)$$

# Search Overview

- Six trainable orbital elements *a, e, i,* $\Omega$, $\omega$, *f;* epoch not trainable
- Three trainable mixture parameters: $N_h$, R, $\tau$
- Compute position **q** and velocity **v** from candidate elements
- Compute direction $\mathbf{u}_{\text{pred}}$ from **q**, **v;** include light time and topos
- Compute distance s from $\mathbf{u}_{\text{pred}}$ to $\mathbf{u}_{\text{obs}}$ for ZTF observations
- Compute log likelihood $\mathscr{L}_i$ for each candidate element
- Gradient descent…
- The rest is details!

# Search Techniques I: Uniform Scale, Gradient Clipping and Independent Weights

- Control variables on uniform scale in [0, 1]
  - e.g. a = a_min x exp(a_ x log(a_max / a_min)); a_ trainable in [0,1]
- Clip gradients by norm; max || Grad $\mathscr{L}$ || = 1
  - would be better to do this elementwise, but requires custom class
- Track log likelihood and hits for each candidate element before summing them in the objective function
- Revert changes only on elements that got worse during an episode
- Weight log likelihood for each element in batch independently

$$\mathcal{L} = \sum_{i=1}^{b} w_i \cdot \mathcal{L}_i$$

  - equivalent to controlling 64 learning rates independently
  - reduce learning rate on an element when it overshoots

# Search Techniques II: Mixture vs. Joint Mode, Encouraging Convergence

- Joint mode: learn all parameters jointly
- Mixture mode: only learn $N_h$, R, τ
- Learning rate: $2^{-12}$ in mixture mode vs. $2^{-16}$ in joint mode
- Modified objective function in mixture mode

$$\mathcal{L} = \sum_{i=1}^{b} w_i \cdot \frac{\mathcal{L}_i}{R_i^{\alpha} \cdot \tau_i^{\beta}}$$

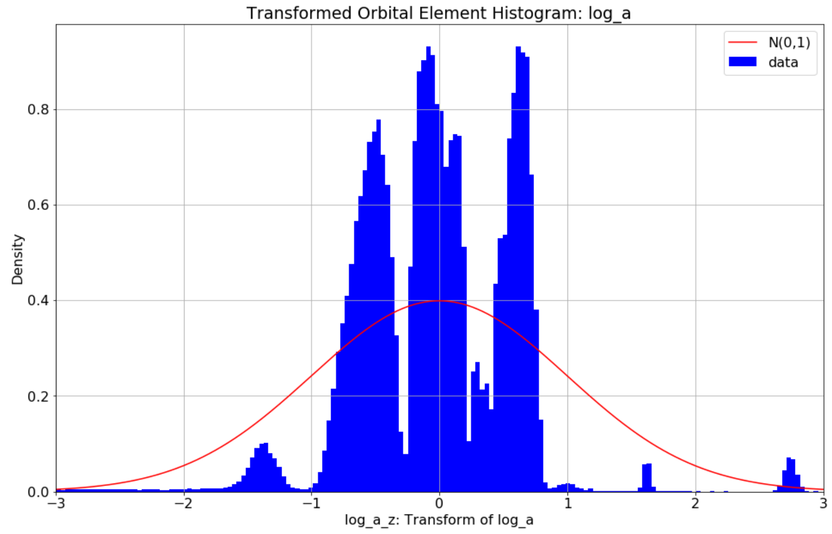
- Theoretical motivation: likelihood would always look better with a larger τ; this encourages the model to converge
- Like adjusting score for degree of difficulty in diving and gymnastics

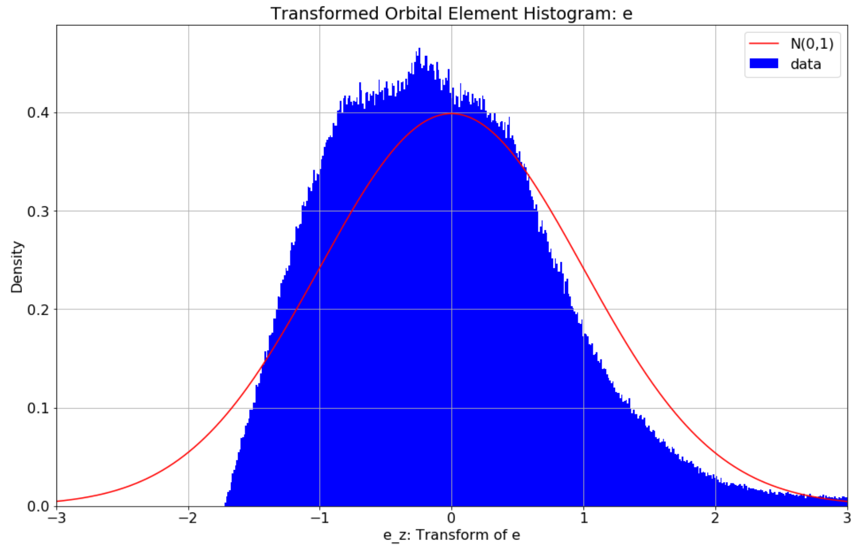
# Asteroid Search Results

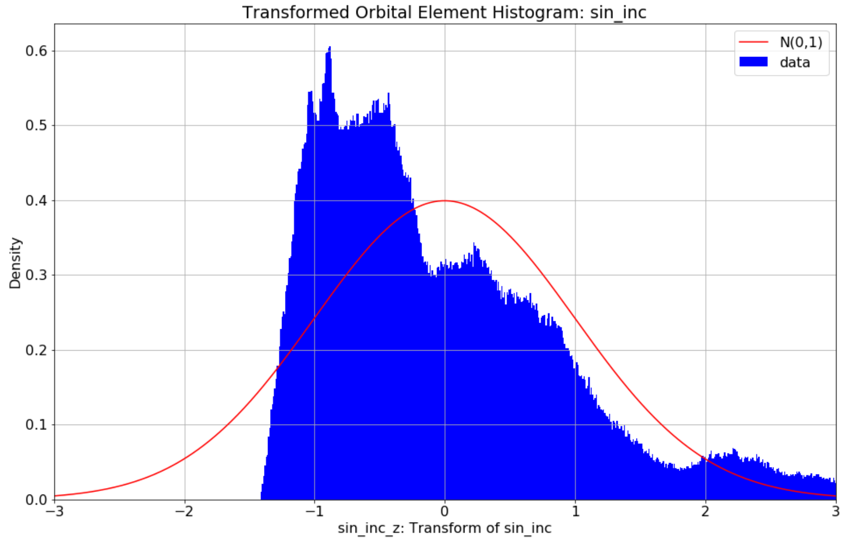
# Comparing Two Orbital Elements

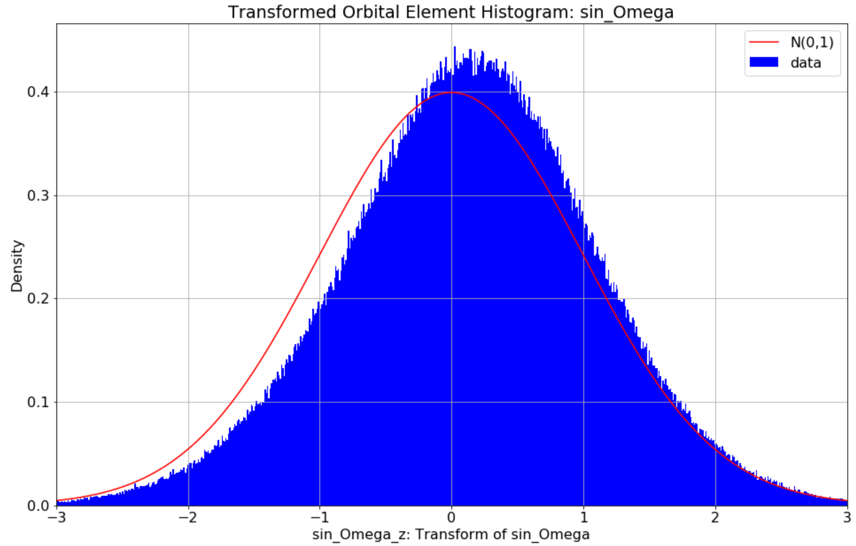
- How far apart are two 6D orbital elements $\varepsilon_1$ and $\varepsilon_2$?
- A naïve Euclidean norm makes no sense at all
- Idea 1: inject the elements into space at a set of times
  - The distance between two elements is the mean distance in AU between the orbits they describe
  - Set 240 sample time points at monthly intervals from 2010 to 2030
- Idea 2: transform elements into low dimensional Cartesian space
  - Try to make each component approximately normal
  - Try to make joint distribution approximately multivariate normal
  - Use the Mahalanobis distance on these transformed elements

# Transforming Elements for Covariance Norm




- Standardize log(*a*)
- Standardize *e*
- Standardize sin(*i*)
- Same transformation for sin and cos of $\Omega$, $\omega$, $f$

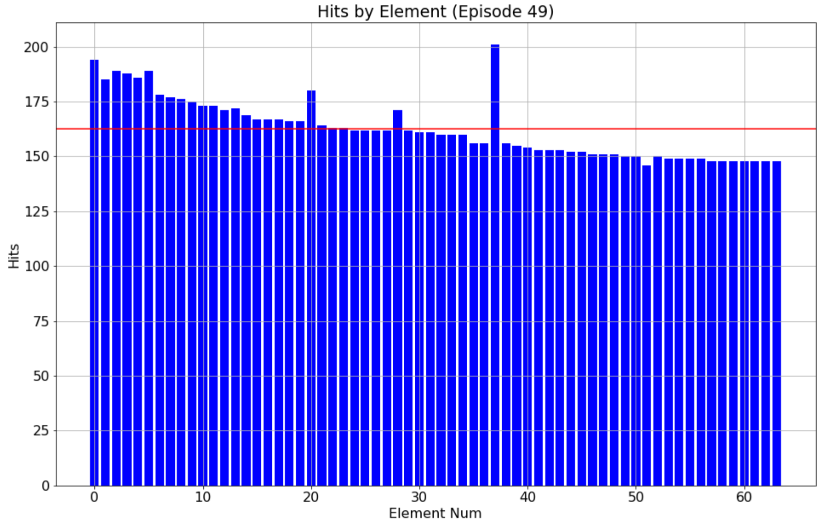
$$u = \frac{1/2 + \arcsin(x)}{\pi}$$

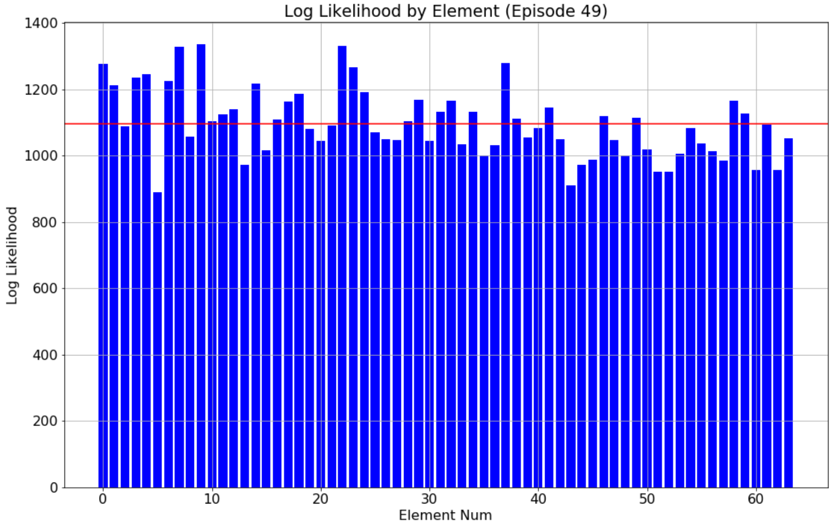
$$z = \Phi^{-1}(u)$$

- This injects elements into $R^9$
- Apply importance weights
  - 1.0 for *a*, *e*
  - 0.5 for *i*
  - 0.1 for sin, cos of $\Omega$, $\omega$, $f$
- Use PCA to find $\beta$ such that X $\beta$ has covariance matrix $I_9$

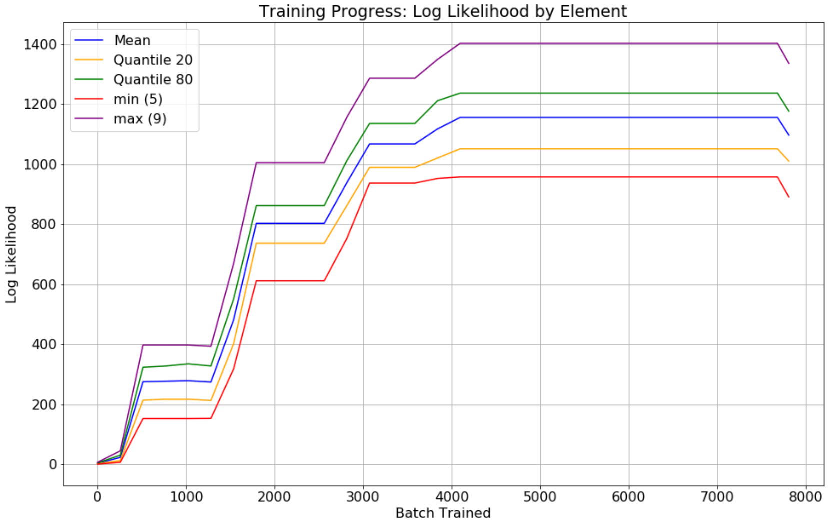
$$\|\epsilon_2 - \epsilon_2\|_{\text{cov}} = \|X_2\beta - X_1\beta\|$$

# Train Known Asteroid Elements: Unperturbed



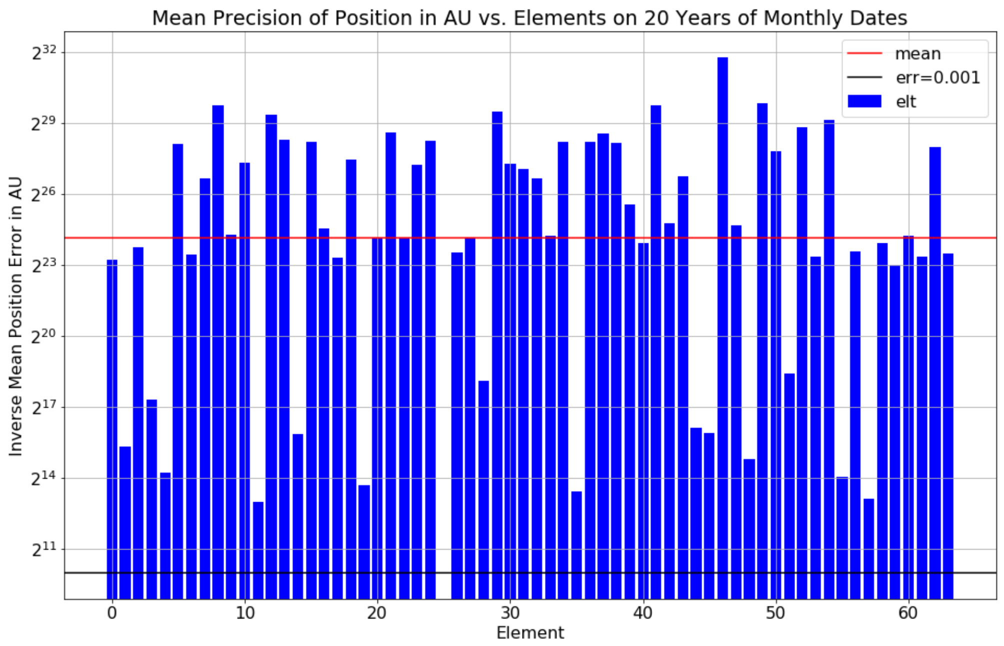



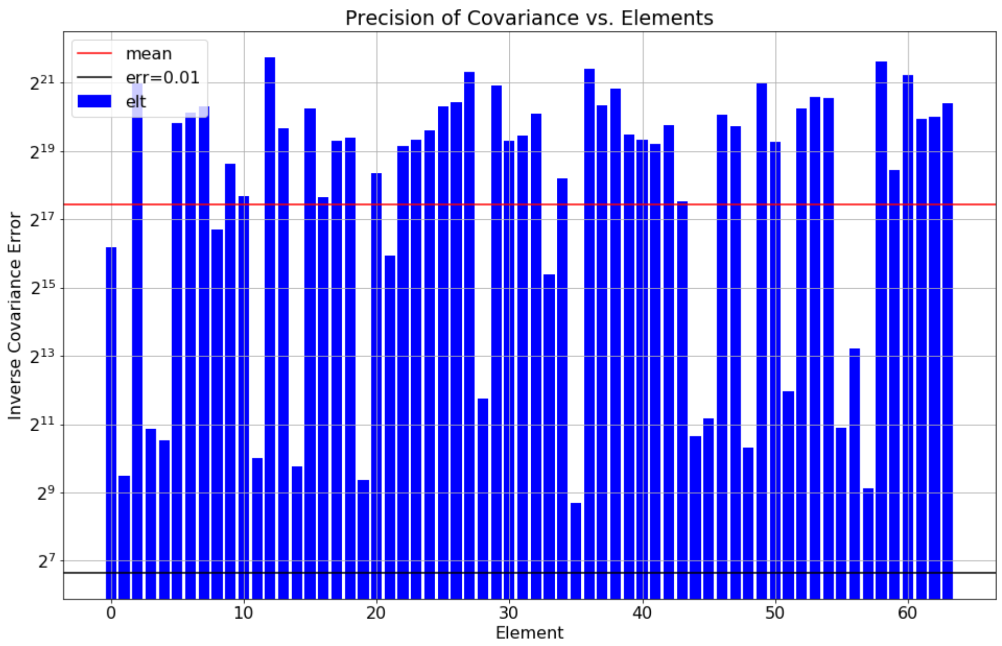





- Start with correct elements but uniformed mixture parameters
- Convergence is almost perfect
- Recovered Elements: 64 (100%)
- Hits: 162.6
- Resolution: 3.0 arc seconds
- Log Like: 1097

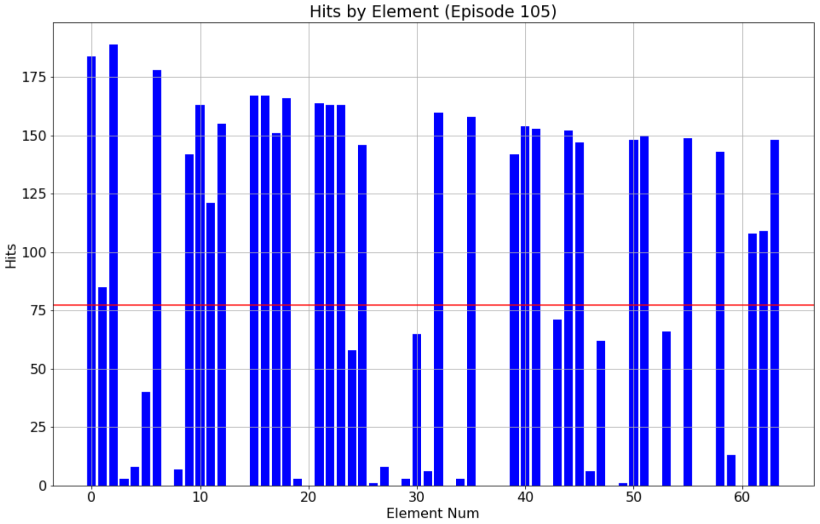
# Fit Quality: Unperturbed Elements

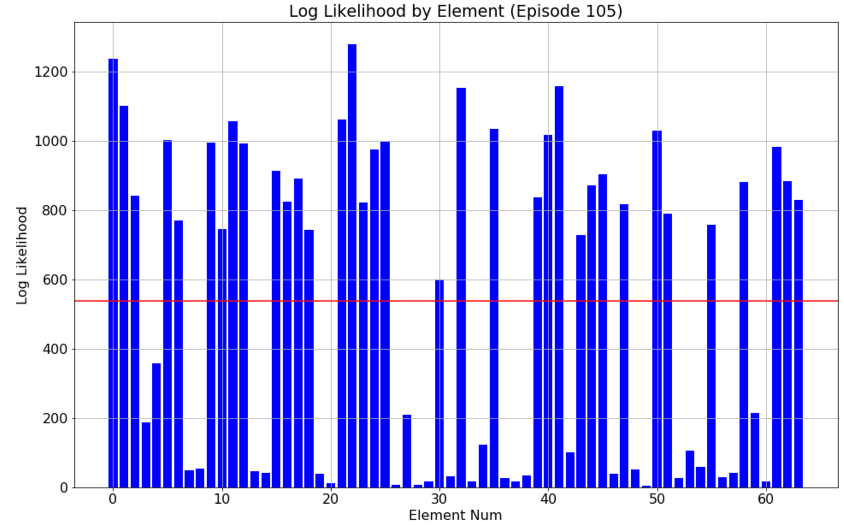


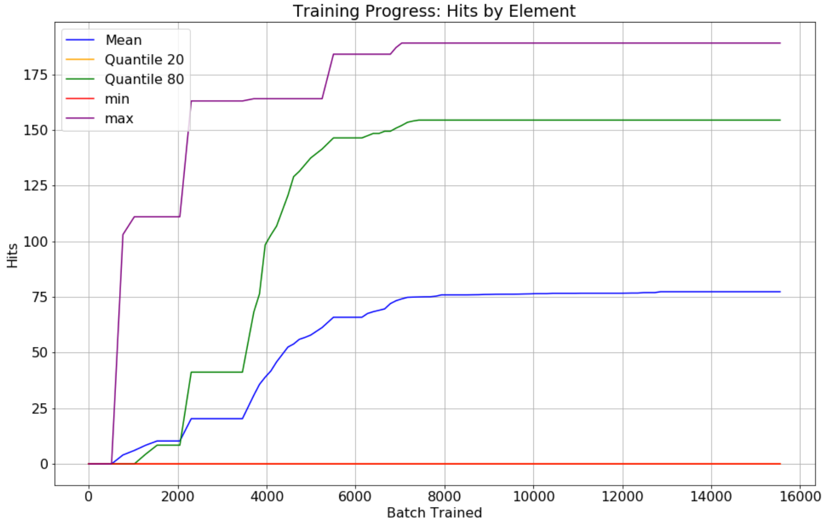



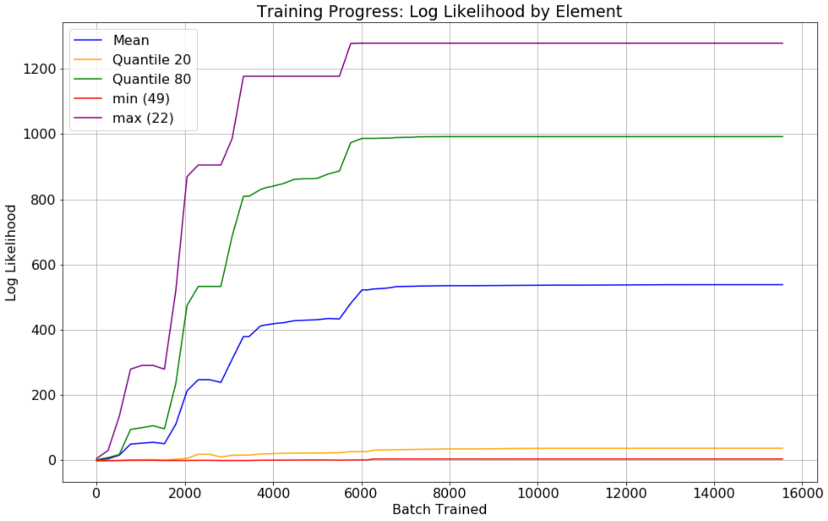

- Do recovered elements match the nearest asteroid?
- 4.6E-8 AU mean distance
- 5.7E-6 covariance norm
- The fit is almost perfect
- Big deal, this is about as hard as hitting a baseball off a tee…

# Train Known Asteroid Elements: Small Perturbation



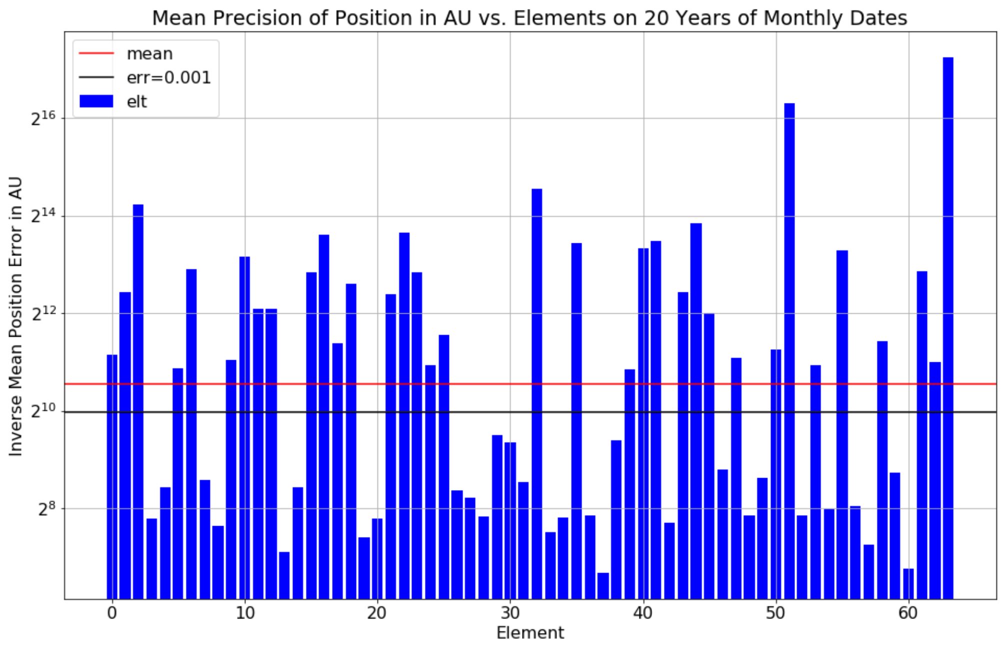



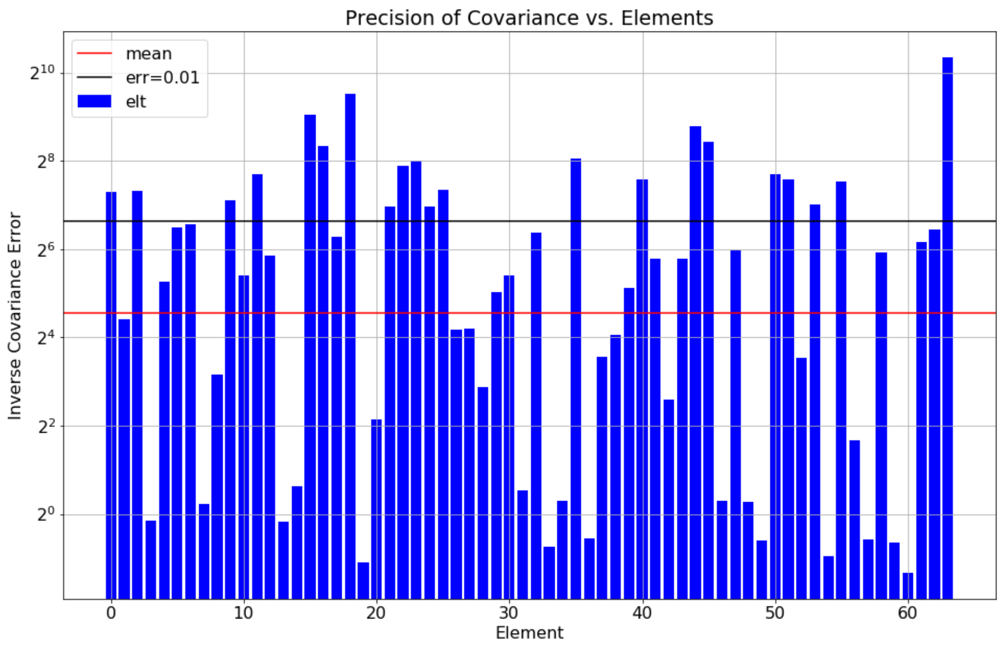




- Small perturbation:
  - 1.0% to *a*
  - 0.25% to *e*
  - 0.05 degrees to *i*
  - 0.25 degrees to $\Omega$, $\omega$, $f$
- Convergence is very good
- Recovered Elements: 42 (65.6%)
- Hits: 117.5
- Resolution: 18.2 arc seconds
- Log Like: 798

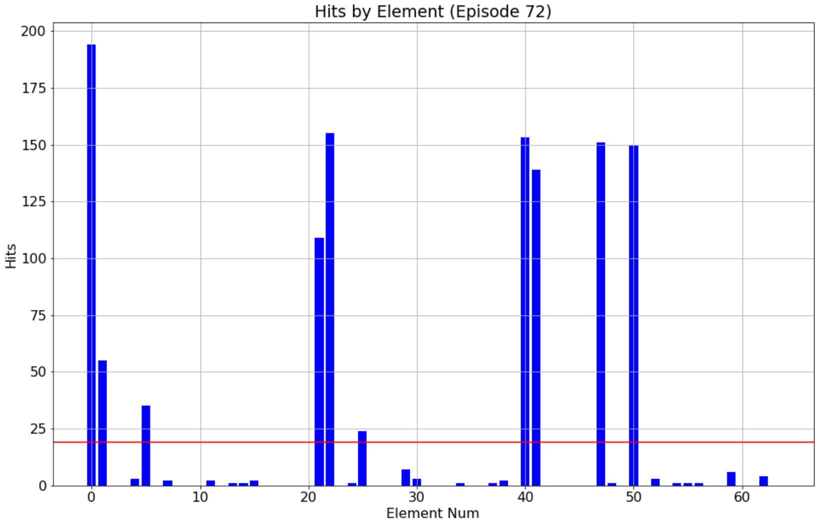
# Fit Quality: Small Perturbation

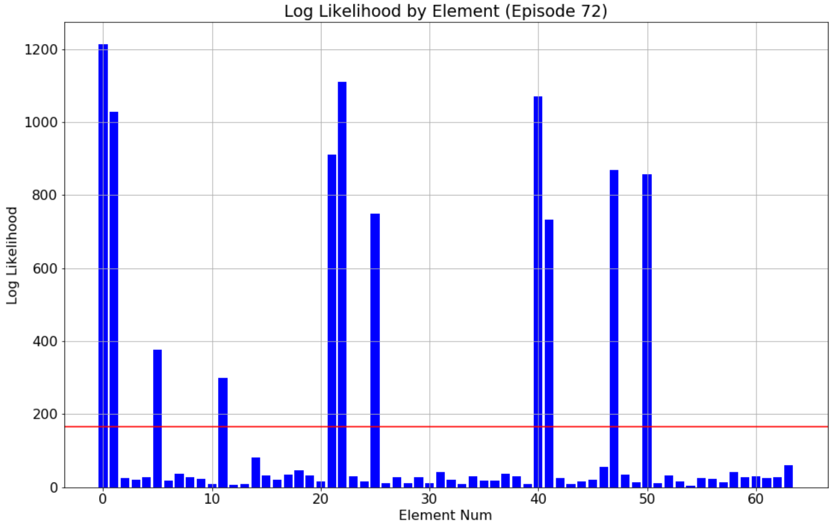


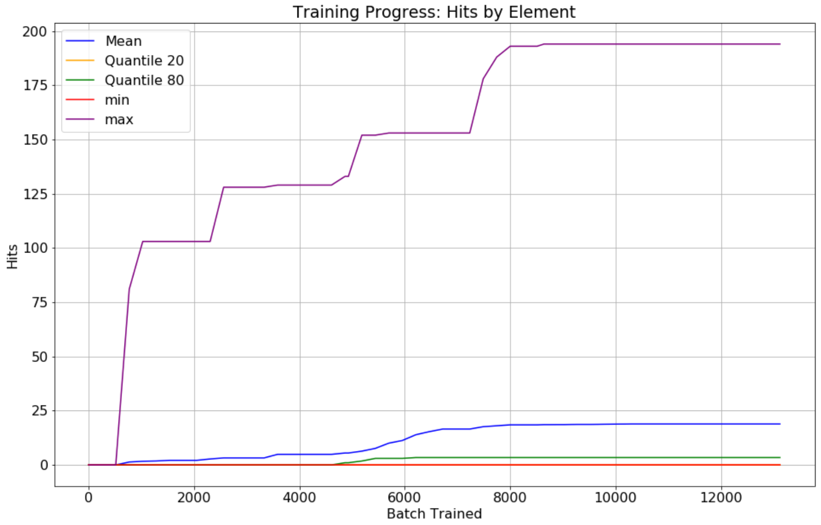



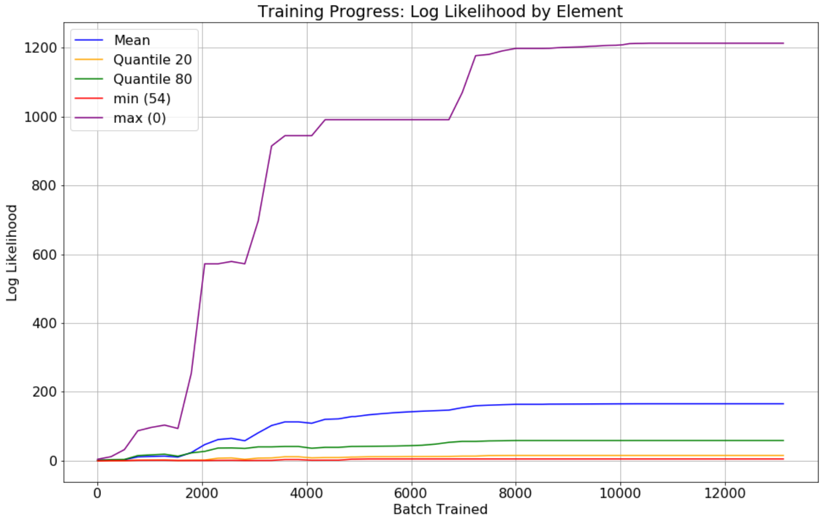

- Do recovered elements match the nearest asteroid?
- 2.6E-4 AU mean distance
- 0.012 covariance norm
- This is still a very good fit on the 42 elements that have been recovered
- This is like your little league coach lobbing the ball over the plate in batting practice…

# Train Known Asteroid Elements: Large Perturbation



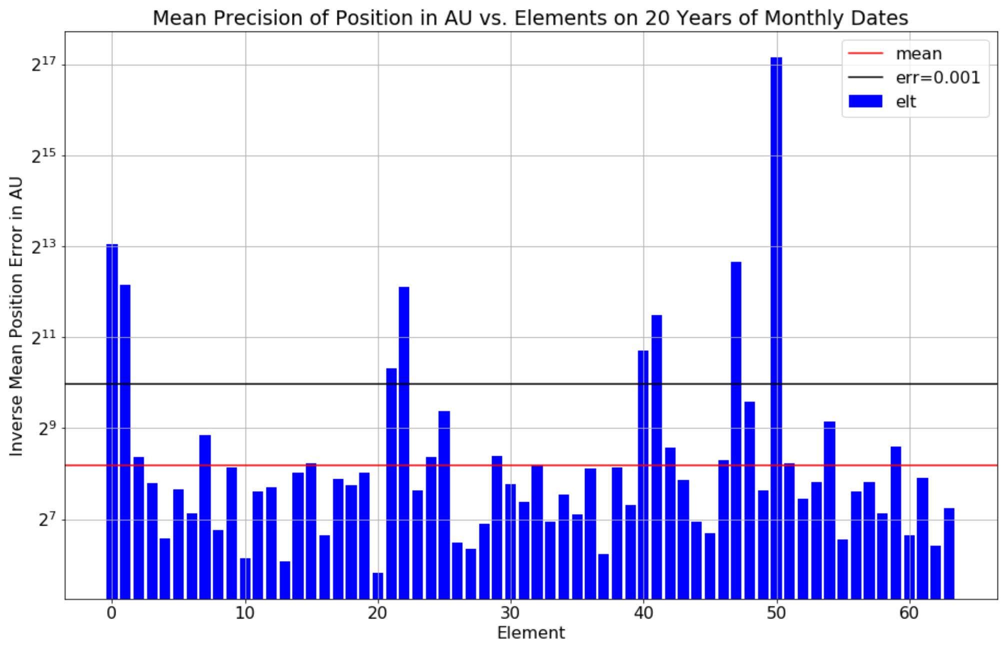



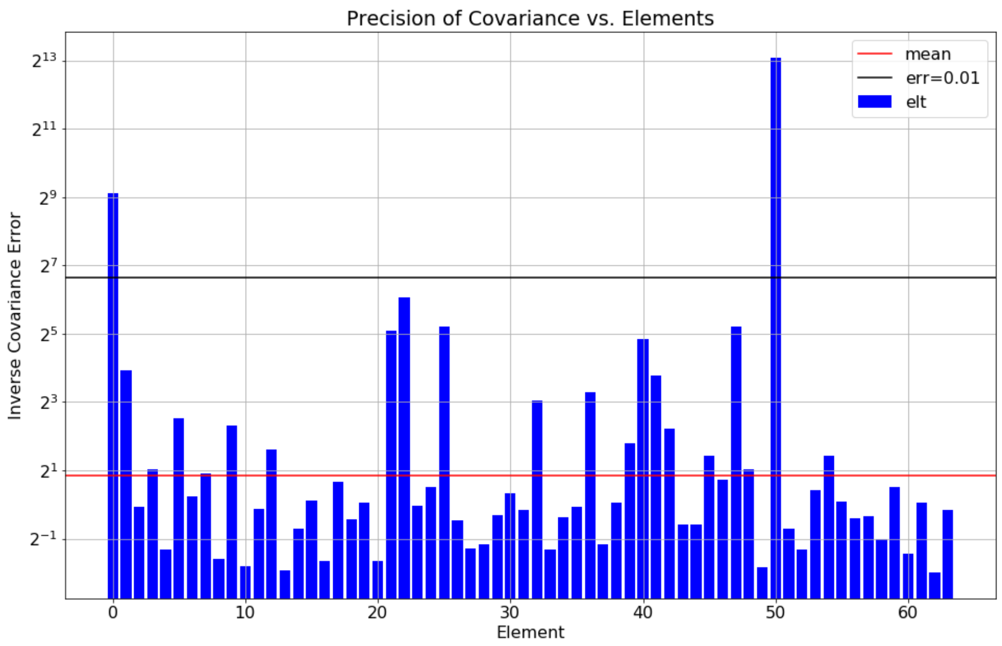




- Small perturbation:
  - 5.0% to *a*
  - 1.0% to *e*
  - 0.25 degrees to *i*
  - 1.0 degrees to $\Omega$, $\omega$, $f$
- Convergence is decent
- Recovered Elements: 12 (18.8%)
- Hits: 98.2
- Resolution: 32.4 arc seconds
- Log Like: 748
- Many of these elements were perturbed so far the original is no longer even the nearest asteroid!

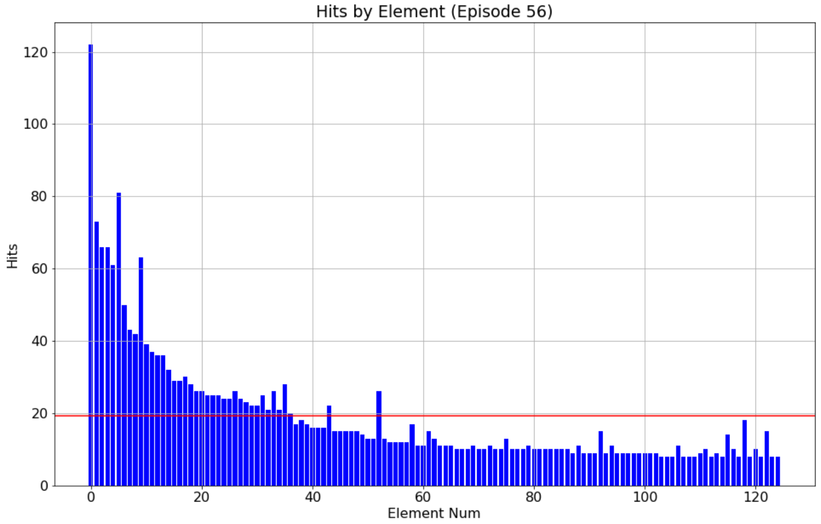
# Fit Quality: Large Perturbation

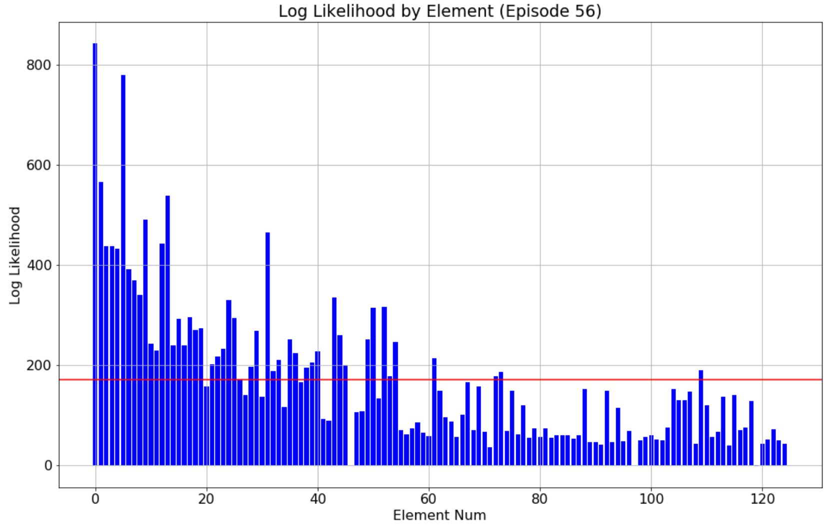


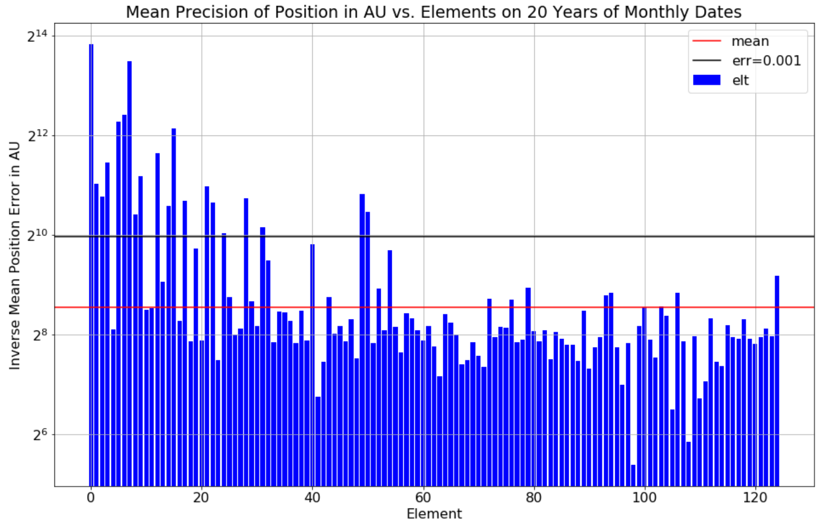



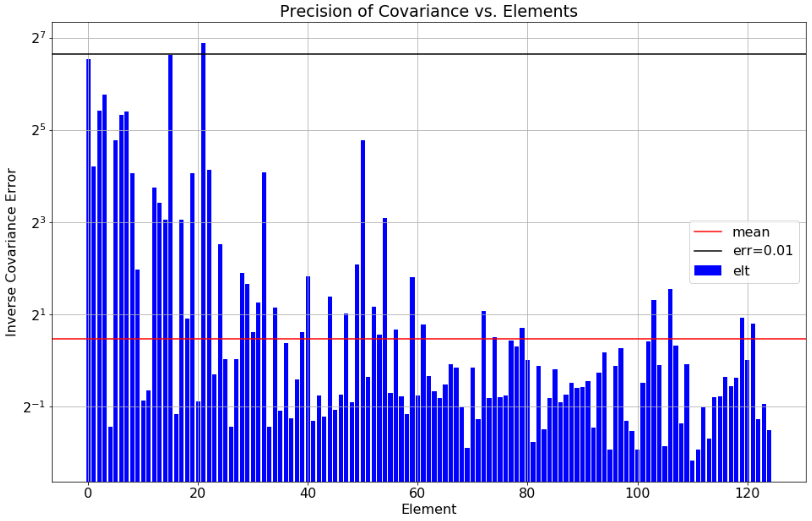

- Do recovered elements match the nearest asteroid? Quite well.
- 4.5E-4 AU mean distance
- 0.032 covariance norm
- This is a decent fit on the 12 elements that have been recovered
- This is a lot harder than the last task-like facing a high school pitcher…

# Search Known Asteroids with Random Initializations



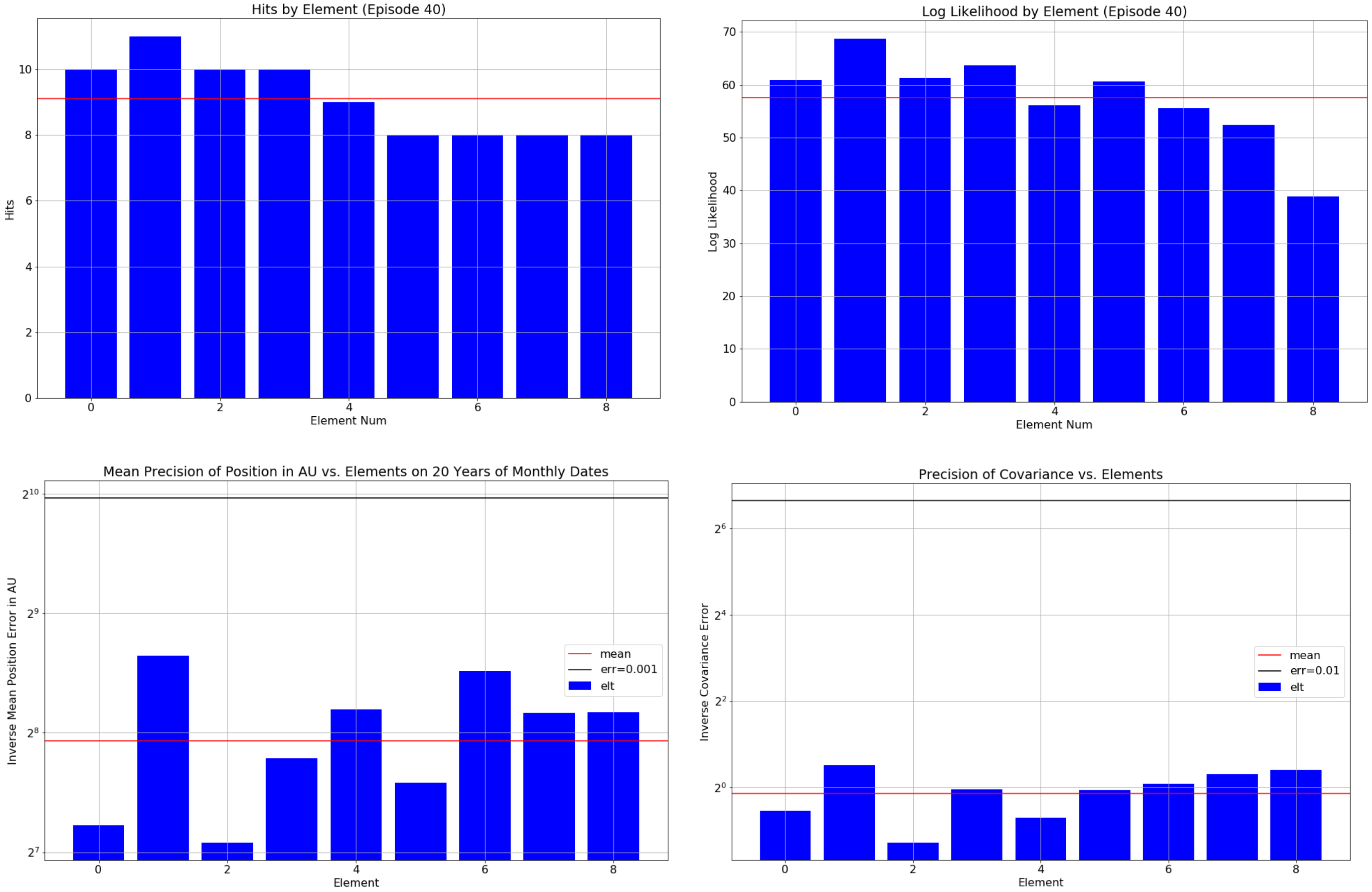




- ~4096 random seeds
- Each batch started with 1024 random elements
- Selected best 64 by mean log(v)
- Trained for ~5 days on 4 GPUs
- Reported fits with ≥ 8 hits and resolution < 20 arc seconds
- Recovered Elements: 125
- Hits: 19.2
- Resolution: 9.3 arc seconds
- Nearest Asteroid Distance: 2.66E-3 AU
- Nearest Asteroid Cov. Norm: 0.73
- Comments on fit quality:
  - Decent fit on some
  - Probably spurious on others
  - Overall shows that this can work
  - But not ready for production
- Baseball analogy continued: facing Roger Clemens, but trying to get hit by the pitch to get on base cheaply

# Search Unknown Asteroids with Random Initializations




- 4096 random seeds
- Each batch started with 1024 random elements
- Selected best 64 by mean log(v)
- Trained for ~5 days on 4 GPUs
- Reported fits with ≥ 8 hits and resolution < 20 arc seconds
- Recovered Elements: 9
- Hits: 9.1
- Resolution: 5.3 arc seconds
- Nearest Asteroid Distance: 4.10E-3 AU
- Nearest Asteroid Cov. Norm: 1.10
- Comments on fit quality:
    - Pretty good resolution
    - But not many hits
    - Not surprising: searching for new asteroids!
- Would prefer to see greater differentiation in distance to nearest known asteroid vs. previous run

# Presenting 9 New Asteroid Candidates

| element_id | a | e | inc | Omega | omega | f | epoch | num_hits | R_sec | thresh_sec | num_rows_close | log_like |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 178421 | 3.160327 | 0.089064 | 0.153620 | 2.668766 | 4.773995 | -0.463213 | 58600.0 | 10.996569 | 3.773662 | 350.229950 | 15.0 | 68.716019 |
| 3308 | 3.026962 | 0.119945 | 0.129409 | 3.903006 | 4.525979 | 4.819063 | 58600.0 | 9.996422 | 3.821569 | 351.971649 | 14.0 | 63.716438 |
| 44117 | 2.935863 | 0.187419 | 0.124516 | 3.166528 | 1.230836 | -3.122138 | 58600.0 | 9.994445 | 5.225097 | 357.466431 | 16.0 | 61.298466 |
| 170789 | 2.735335 | 0.152867 | 0.403704 | 6.038029 | 3.016815 | -3.443415 | 58600.0 | 9.996418 | 7.083936 | 347.111053 | 13.0 | 60.860794 |
| 113970 | 2.897024 | 0.068932 | 0.209250 | 5.663728 | 3.868474 | 4.450756 | 58600.0 | 7.999145 | 3.958026 | 348.778046 | 9.0 | 60.584415 |
| 45801 | 2.754677 | 0.047293 | 0.118126 | 3.139070 | 5.767782 | -1.949476 | 58600.0 | 8.996888 | 4.321019 | 326.387482 | 14.0 | 56.090981 |
| 50775 | 2.374712 | 0.100280 | 0.165483 | 4.280114 | 5.955170 | 2.995417 | 58600.0 | 9.948814 | 9.705722 | 809.804260 | 29.0 | 55.641151 |
| 96507 | 2.820351 | 0.068964 | 0.080233 | 2.222034 | 0.960931 | -2.630966 | 58600.0 | 8.821539 | 3.905472 | 262.245087 | 12.0 | 52.416988 |
| 191915 | 2.315446 | 0.192885 | 0.057156 | 2.130249 | 2.865086 | -4.122024 | 58600.0 | 7.982198 | 9.668445 | 388.469055 | 21.0 | 38.837540 |

# ZTF Hits for Selected Asteroid Candidates

| element_id | ObjectID | mjd | ra | dec | mag_app | s_sec |
|---|---|---|---|---|---|---|
| 178421 | b'ZTF18aboluox' | 58430.166620 | 346.704046 | -10.675142 | 15.787200 | 1.260372 |
| 178421 | b'ZTF18aboluox' | 58430.170313 | 346.704079 | -10.675160 | 15.800000 | 0.982009 |
| 178421 | b'ZTF18aboluox' | 58430.166620 | 346.704001 | -10.675028 | 15.587700 | 1.699850 |
| 178421 | b'ZTF18aboluox' | 58430.170313 | 346.704073 | -10.675099 | 15.654600 | 1.175734 |
| 178421 | b'ZTF18acewaex' | 58863.138472 | 67.229666 | 17.295098 | 19.328100 | 6.219038 |
| 178421 | b'ZTF18acewaex' | 58863.138472 | 67.229789 | 17.295055 | 19.343500 | 5.789846 |
| 178421 | b'ZTF18acewaex' | 58863.152465 | 67.229835 | 17.295083 | 19.283501 | 2.389730 |
| 178421 | b'ZTF18acewaex' | 58863.152465 | 67.229783 | 17.295077 | 19.263100 | 2.392794 |

| element_id | ObjectID | mjd | ra | dec | mag_app | s_sec |
|---|---|---|---|---|---|---|
| 3308 | b'ZTF18abtpdzg' | 58670.439884 | 354.565806 | -8.964600 | 17.605200 | 3.286346 |
| 3308 | b'ZTF18abtpdzg' | 58670.440336 | 354.565888 | -8.964426 | 17.601101 | 3.888707 |
| 3308 | b'ZTF18abtpdzg' | 58670.462604 | 354.565787 | -8.964429 | 16.858500 | 0.697156 |
| 3308 | b'ZTF18abtpdzg' | 58670.463056 | 354.565909 | -8.964536 | 17.128700 | 1.329765 |
| 3308 | b'ZTF18abtpdzg' | 58670.462604 | 354.565824 | -8.964445 | 17.092899 | 0.837276 |
| 3308 | b'ZTF18abtpdzg' | 58670.463056 | 354.565809 | -8.964412 | 16.658600 | 0.769468 |
| 3308 | b'ZTF18abspkzw' | 58863.110058 | 346.642277 | 1.047679 | 16.541800 | 8.282505 |
| 3308 | b'ZTF18abspkzw' | 58863.109606 | 346.642261 | 1.047839 | 16.946899 | 7.604068 |

| element_id | ObjectID | mjd | ra | dec | mag_app | s_sec |
|---|---|---|---|---|---|---|
| 191915 | b'ZTF18abtxtgd' | 58430.170313 | 341.181109 | -12.234676 | 19.693501 | 0.036218 |
| 191915 | b'ZTF18abtxtgd' | 58430.166620 | 341.181092 | -12.234595 | 19.260099 | 2.621655 |
| 191915 | b'ZTF19abtsqmn' | 58899.139884 | 94.436593 | 22.583914 | 18.914301 | 6.521414 |
| 191915 | b'ZTF19abtsqmn' | 58899.140336 | 94.436579 | 22.583982 | 18.900700 | 6.623162 |
| 191915 | b'ZTF19abtsqmn' | 58899.192569 | 94.436720 | 22.583899 | 19.819700 | 0.946245 |
| 191915 | b'ZTF19abtsqmn' | 58899.222442 | 94.436672 | 22.583855 | 19.873501 | 4.829147 |
| 191915 | b'ZTF19abtsqmn' | 58899.220162 | 94.436545 | 22.583886 | 20.234699 | 4.129297 |
| 191915 | b'ZTF19abtsqmn' | 58899.220613 | 94.436587 | 22.583902 | 20.670900 | 4.263810 |

| element_id | ObjectID | mjd | ra | dec | mag_app | s_sec |
|---|---|---|---|---|---|---|
| 170789 | b'ZTF17aaaqwwg' | 58903.113588 | 84.725827 | 15.284655 | 19.437901 | 7.917067 |
| 170789 | b'ZTF17aaaqwwg' | 58903.116806 | 84.725856 | 15.284678 | 19.699699 | 7.043546 |
| 170789 | b'ZTF17aaaqwwg' | 58903.129097 | 84.725897 | 15.284685 | 20.146000 | 3.737848 |
| 170789 | b'ZTF17aaaqwwg' | 58903.126840 | 84.725899 | 15.284678 | 19.689501 | 4.333303 |
| 170789 | b'ZTF17aaaqwwg' | 58903.128194 | 84.725824 | 15.284668 | 19.916201 | 3.767049 |
| 170789 | b'ZTF17aaaqwwg' | 58903.149248 | 84.726023 | 15.284693 | 19.143600 | 3.802050 |
| 170789 | b'ZTF17aaaqwwg' | 58903.149699 | 84.725870 | 15.284728 | 19.707399 | 4.209481 |
| 170789 | b'ZTF17aaaqwwg' | 58903.151053 | 84.725889 | 15.284665 | 19.904301 | 4.672150 |

- Should we believe these new asteroid candidates? Look at ZTF hits to decide.
- Element `178421` has 4 hits on `ZTF18aboluox` and 4 hits on `ZTF18acewaex`, made 433 days apart
    - Magnitudes are too different (4), spurious connection of 2 different objects
- Element `3308` has 6 hits on `ZTF18abtpdzg` and 2 hits on `ZTF18abspkzw`, made 193 days apart
- Element `191915` has 6 hits on `ZTF19abtsqmn` and 2 hits on `ZTF18abtxtgd`, made 469 days apart
    - Magnitudes are compatible: The model has made a non-obvious connection on compatible tracks!
- Element `170789` has 8 hits on `ZTF17aaaqwwg` made in a 55 minute interval
    - Model agrees with ZTF that this is one track for the same object

# Conclusions

- Prove that asteroid search over orbital elements works
  - Need an adequate initialization and representation in data set
- Built a working prototype in TensorFlow
  - First demonstration of efficient astrometric computations on GPU?
- High quality integration of the Solar System and astrometric directions
  - Associated each of 5.7E6 ZTF observations to nearest asteroid
  - 4.2E12 interactions, possibly novel and useful data set
- Proposed candidate orbital elements for 9 new asteroids
- Proof of concept for an automated pipeline to search for new asteroids

# Future Work I: Initialization, More Data

- Intelligent initialization of candidate elements
  - Random initialization was just a quick and dirty placeholder; ran out of time
  - ZTF ObjectID is a great starting point for initializations
  - Provisionally assume that all the detections belong to same object
  - Build least squares fit for candidate element
  - Code mostly there now; modify AsteroidSearch with new loss function
- Add a second data set
  - ZTF is great, but it only dates back to July 2019
  - Want to add a second data source
  - Ideally this should have an ML pipeline to classify probable asteroid hits
  - Failing that, can use any data set with a real-bogus classifier, then subtract known stars and galaxies
  - Is Pan-STARRS a good choice?
  - Advice from astronomers on my Committee would be welcome here!

# Future Work II: Magnitude, Automated Pipeline

- Incorporate magnitude into log likelihood
  - Have prototype to predict magnitude and incorporate it
  - It was too finicky, needed to turn it off to get first version working on time
- Develop an automated pipeline
  - Initial goal: Accurately rebuild a large fraction of the asteroid catalogue
  - ZTF alone has over 100,000 asteroids with 10 or more hits in the data
  - Plausible that with intelligent initialization, we can recover many of these
  - With a second data set, we could really go far
  - An automated process that can accurately recreate the known catalogue…
  - …is also an automated process that can provisionally classify new asteroids!

# Mille Grazie: Thank you for Your Attention!

- Questions?
- Comments?
- Suggestions?