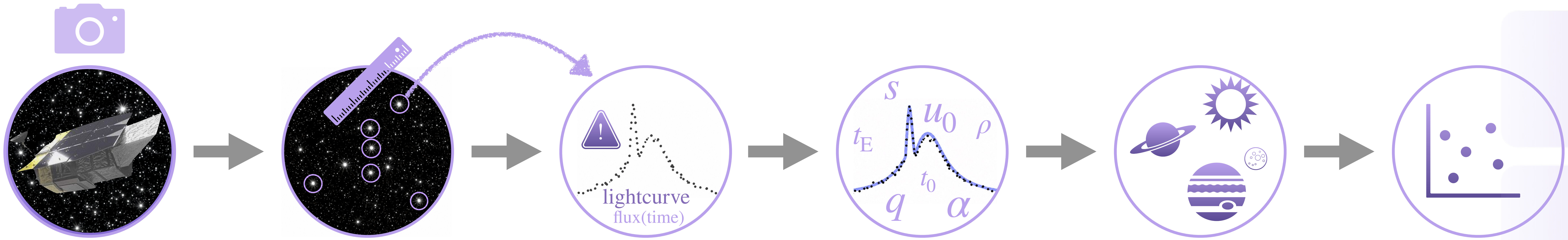
Microlensing Pipelines: From Lightcurves to Characterization

Stela Ishitani Silva (Caltech/IPAC-Roman MSOS)

2026 Sagan Summer Workshop | Exoplanets with Roman Surveys: Microlensing and Transits

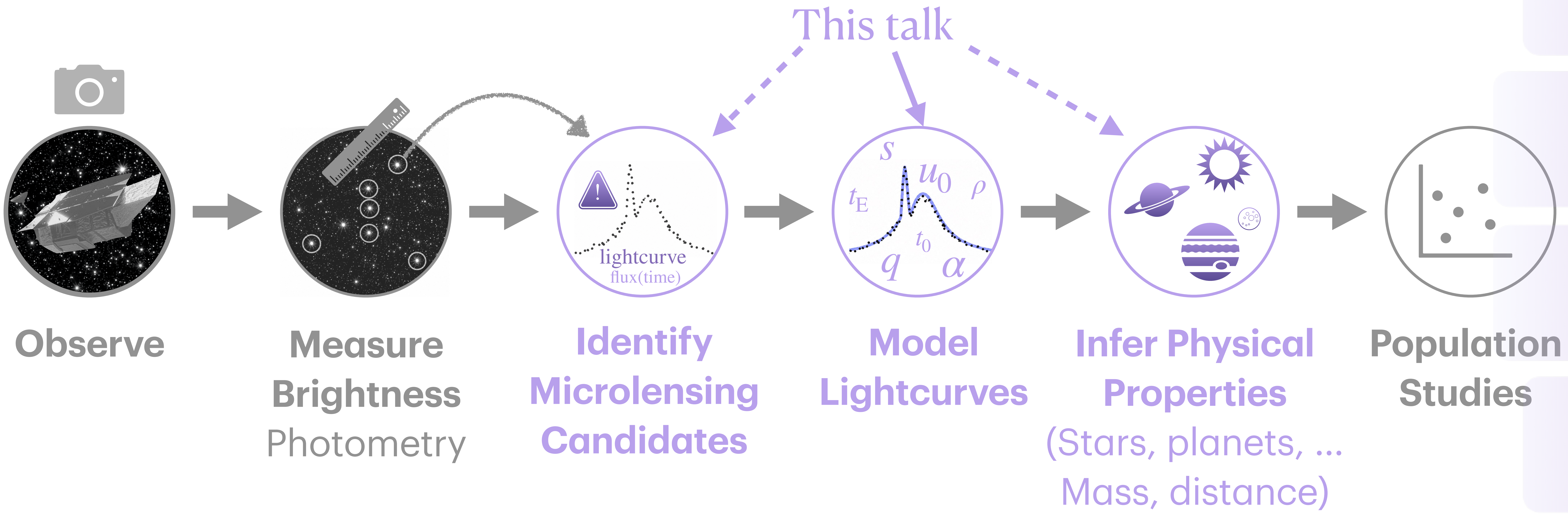
July 23, 2026

From Images to Microlensing Exoplanets

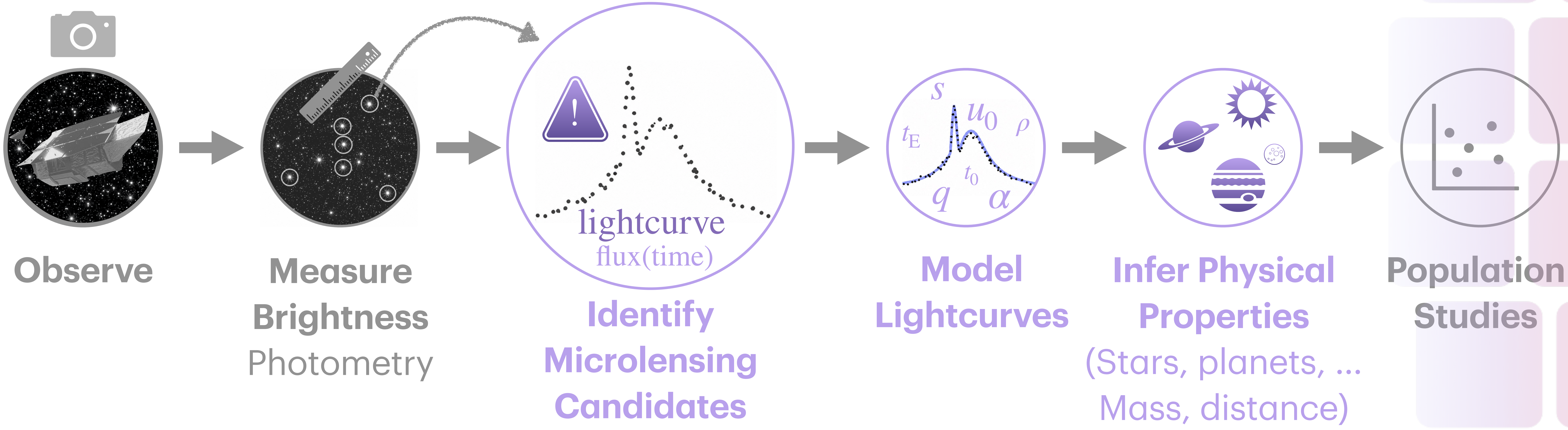


Observe → **Measure Brightness** Photometry → **Identify Microlensing Candidates** → **Model Lightcurves** → **Infer Physical Properties** (Stars, planets, ... Mass, distance) → **Population Studies**

From Images to Microlensing Exoplanets



From Images to Microlensing Exoplanets



lightcurve
flux(time)

Identify Microlensing Candidates

Which Lightcurves Should We Model?

- Most lightcurves don't have a microlensing event

Special quiz:
If you randomly select a star toward the Galactic bulge,
what is the approximate chance that it is being microlensed?
(A) 1 in 500 (B) 1 in 5,000 (C) 1 in 50,000 (D) 1 in 500,000

(Refer to Bachelet's talk)

  - They are quite rare!
  - "But it's okay! We just have to observe millions of stars" 😎
    (said OGLE, MOA, KMTNet, PRIME, ...)
- Roman will deliver hundreds of millions of precise lightcurves
- We need to identify what lightcurves contain a possible microlensing signal

Identify Microlensing Candidates

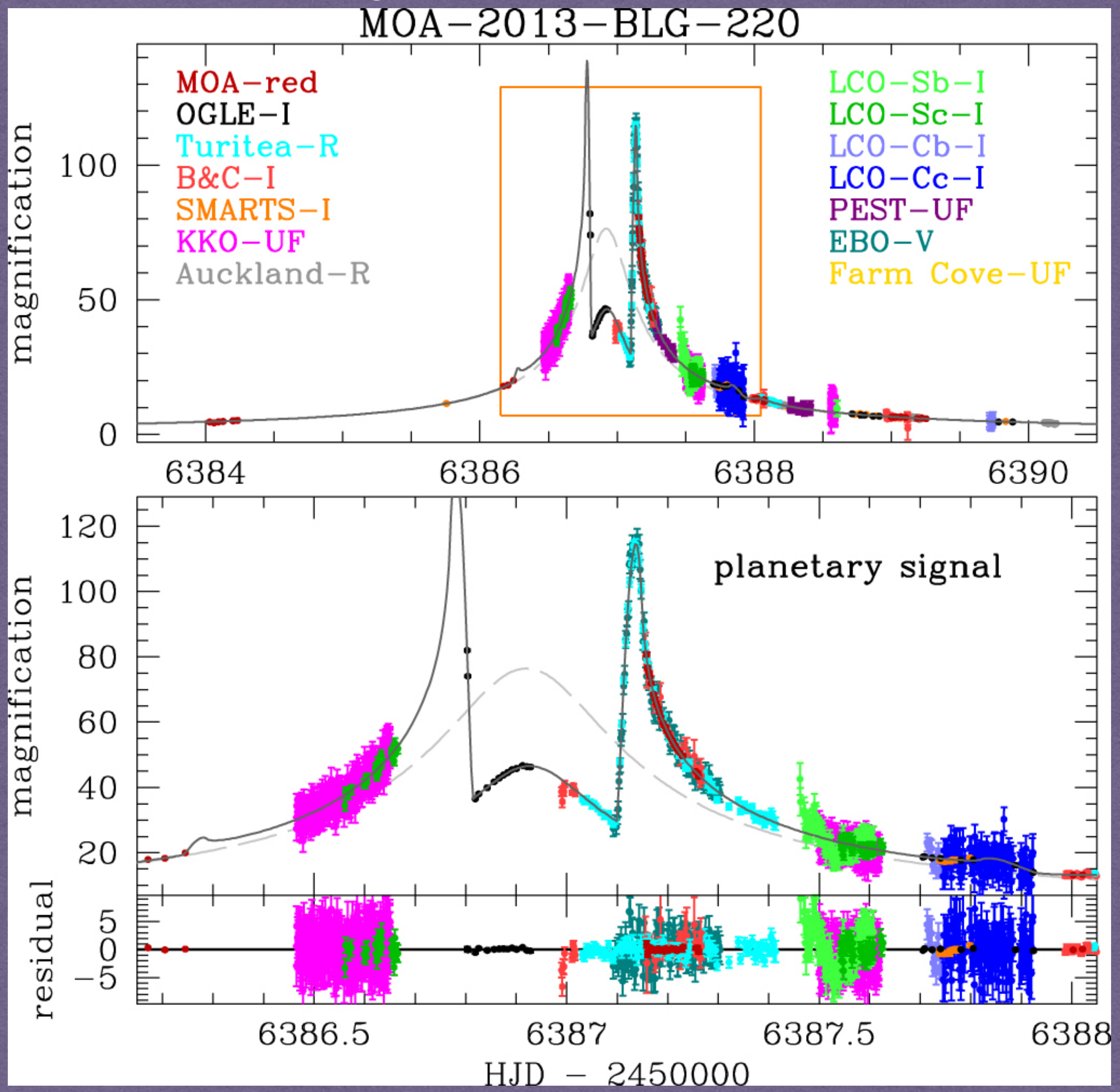
What Does a Microlensing Event Look Like?

- Not flat
- Non-periodic
- Peak(s)
- Achromatic

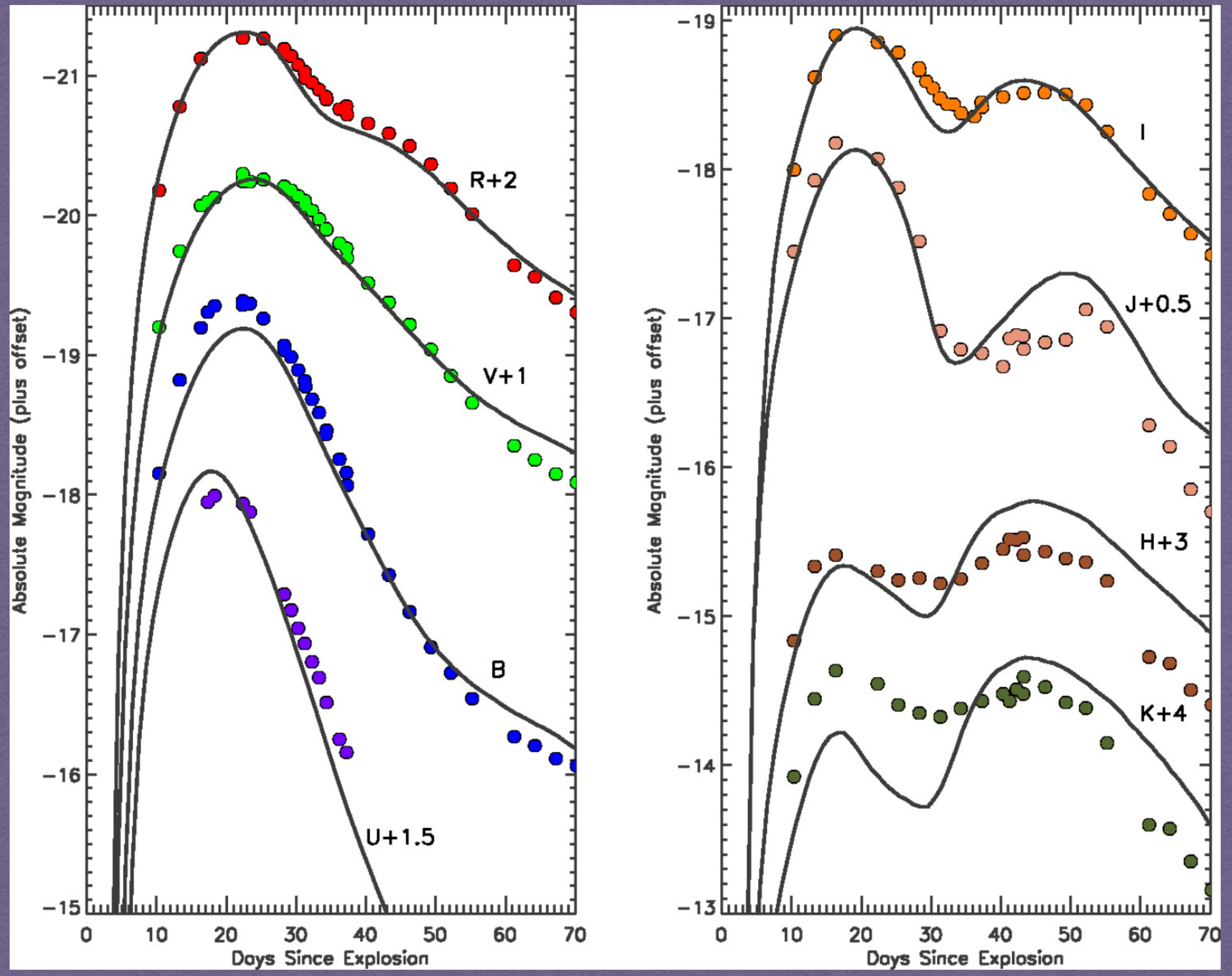
+

- Point-source point-lens fit

Microlensing Event - Vandorou *et al* 2020

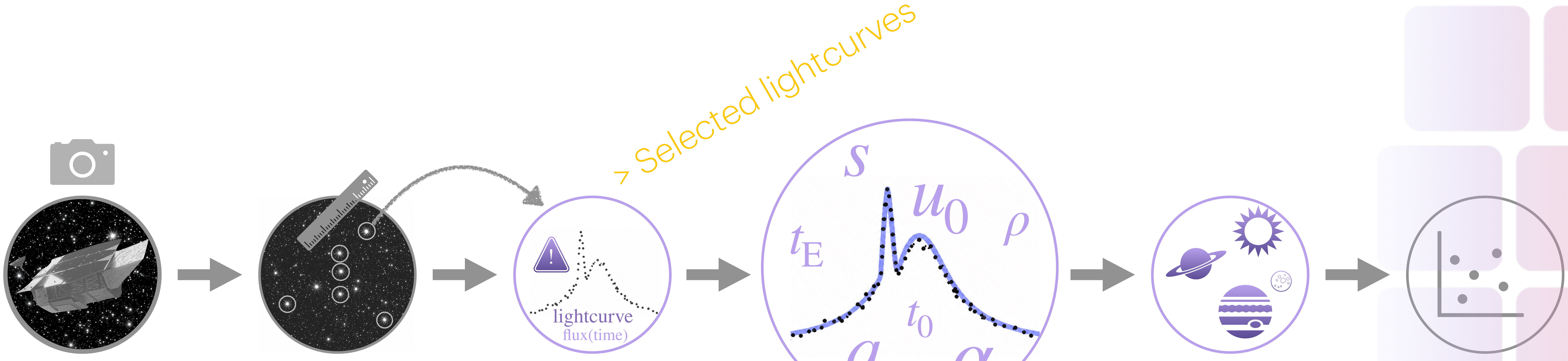


NOT MICROLENSING:
Supernova (SN 2001el) - Kasen 2006

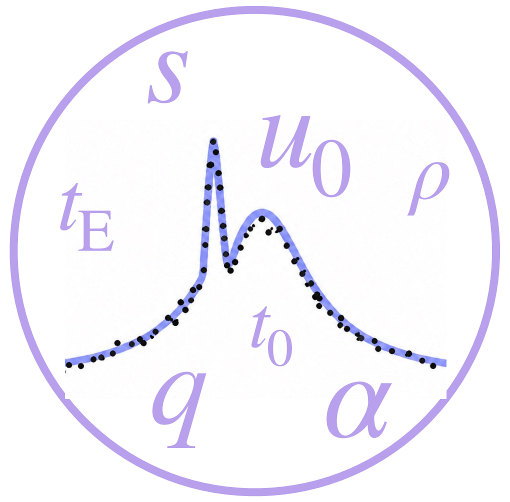


The output of this step should be a collection of lightcurves that you believe are worth modeling

From Images to Microlensing Exoplanets



Observe → **Measure Brightness** Photometry → **Identify Microlensing Candidates** → **Model Lightcurves** → **Infer Physical Properties** (Stars, planets, … Mass, distance) → **Population Studies**

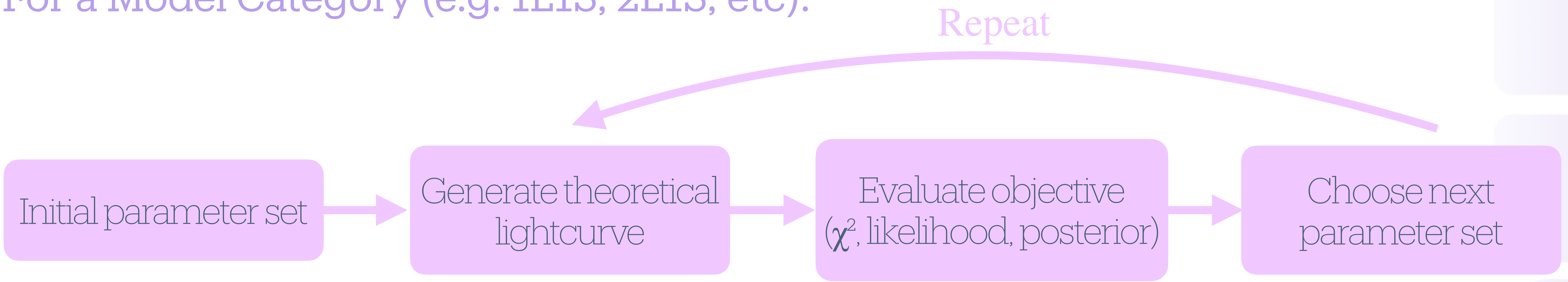


Model Lightcurves

Fitting Microlensing Models to Observed Lightcurves

What set of parameters?

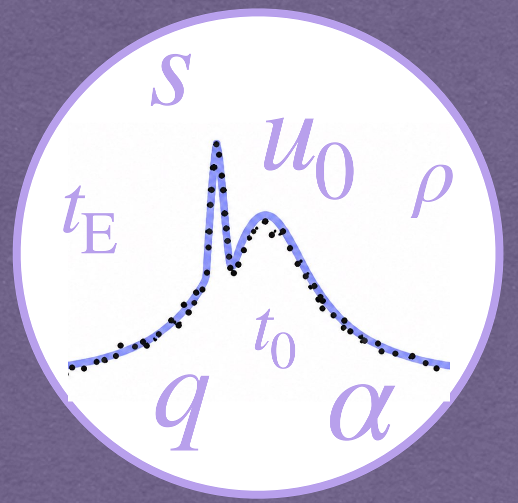
- Find the theoretical microlensing model that best reproduces the observed lightcurve
    - For a Model Category (e.g. 1L1S, 2L1S, etc):



Model Lightcurves

Fitting Microlensing Models to Observed Lightcurves

- Model Category (e.g. 1L1S, 2L1S, etc)



Model Lightcurves

Model Categories

(Refer to talks: Bachelet, Bennett & Bozza)

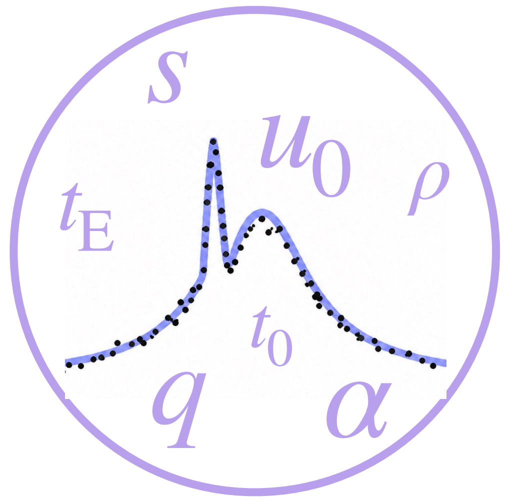
- **Start with the simplest model** type and increase complexity if justified by the data.

Avoids overfitting

Common acronyms	Source	Lens	Other high-order effects		
PSPL (a type of 1L1S)	Point source	Point lens	Parallax		
FSPL (a type of 1L1S)	Finite source	Point lens	Parallax		
FSBL (a type of 2L1S)	Finite source	Binary lens	Parallax	Orbital Motion (lens)	
BSPL, DSSL (a type of 1L2S)	Binary source	Point lens	Parallax	Xallarap	
BSBL (2L2S)	Binary source	Binary lens	Parallax	Xallarap	Orbital Motion (lens)
3L1S	Finite source	Triple lens	Parallax	Orbital Motion (lens)	

+

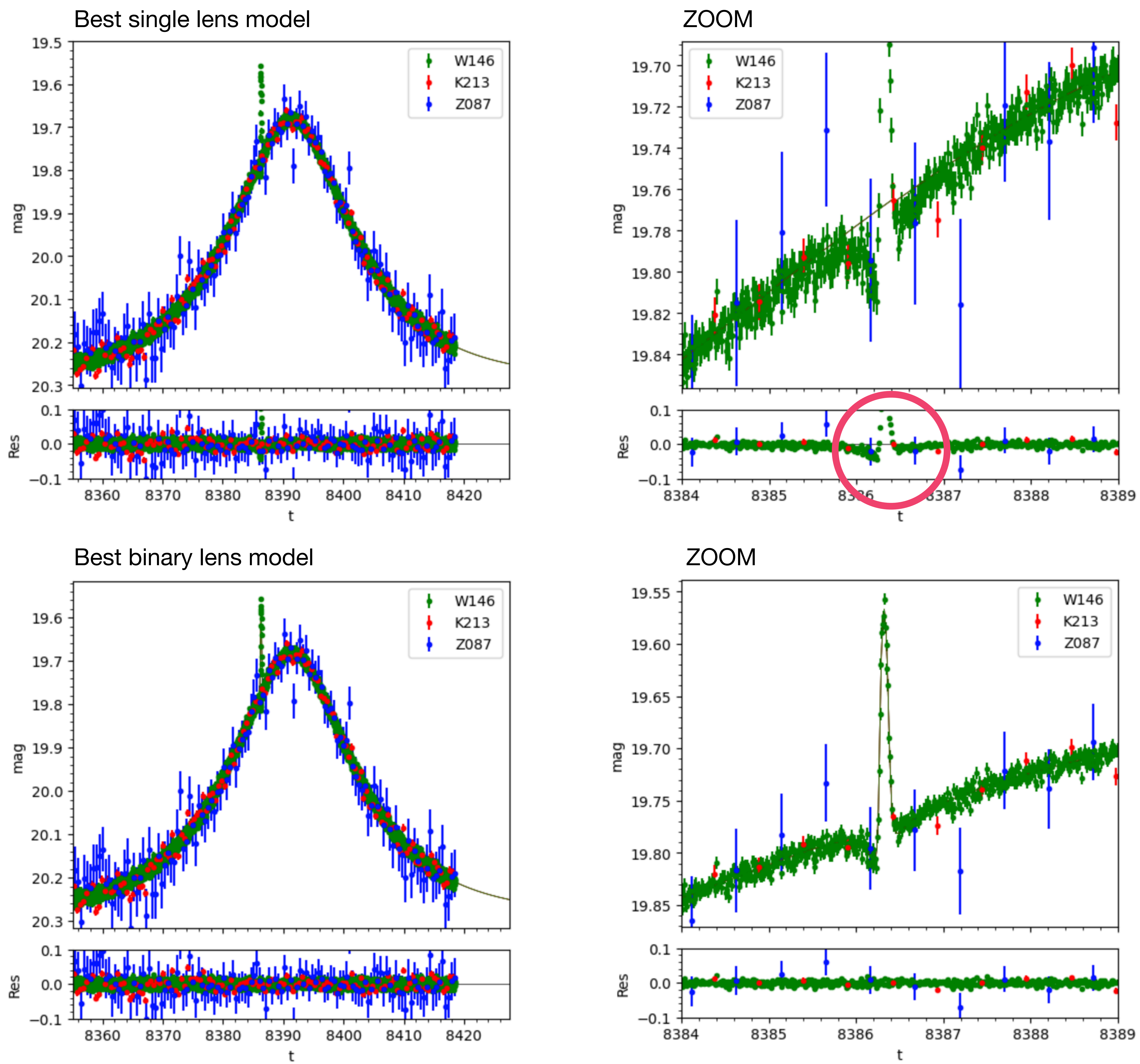
Event 0_45_2904



Model Lightcurves

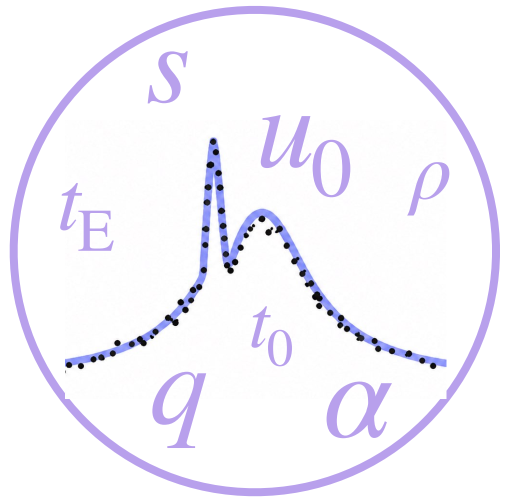
Do I need a different model category?

residuals: data – model



For more examples, see Bozza and Bennett's talk

Credit: RGES PIT - simulated lightcurve (Zohrabi et al) + model fitting (Ishitani Silva et al)



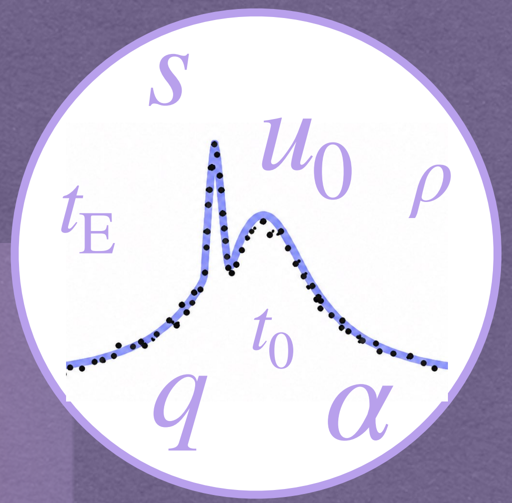
Model Lightcurves

Fitting Microlensing Models to Observed Lightcurves

- For a Model Category (e.g. 1L1S, 2L1S, etc):

Repeat

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior) → Choose next parameter set



Model Lightcurves

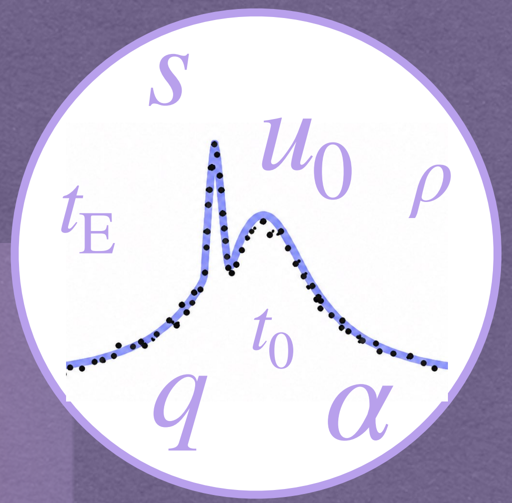
How Fitting Works

Initial parameter set

$$x = (t_E, t_0, u_0, \pi_{EN}, \pi_{EE}, s, \alpha, \rho, [etc])$$

Initial parameters depend on the method

- Educated guess, e.g. t0 = peak time
- Previous quick fit, e.g. alert-system PSPL fit
- A set of parameters from a predefined grid
- Drawn from a prior distribution



Model Lightcurves

How Fitting Works

Initial parameter set → Generate theoretical lightcurve

$$x = (t_E, t_0, u_0, \pi_{EN}, \pi_{EE}, s, \alpha, \rho, [etc])$$

$$A = \sum_{j=1}^{N_{\text{image}}} \frac{1}{\left|\det J(z_j)\right|}$$

(Refer to Bachelet's talk)

Per instrument/filter i :

Linear fit

constants

$$F_i(t) = f_{\mathrm{s,i}} \times \underbrace{A(t, x)}_{\text{magnification}} + f_{b,i}$$

flux of the unlensed source

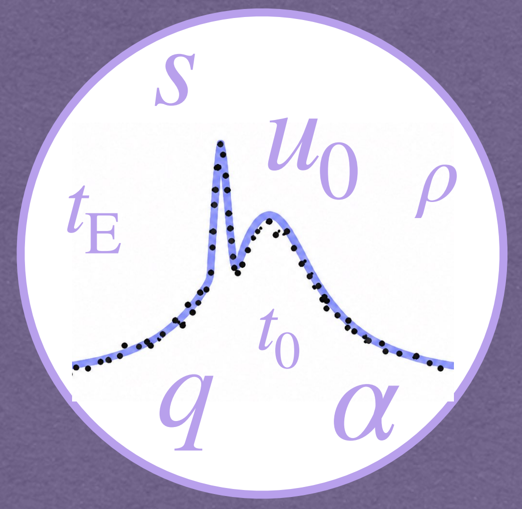
blend flux

For PSPL : (Paczynski 1986)

$$A(u) = \frac{u^2 + 2}{u\sqrt{u^2 + 4}} \qquad u(t) = \sqrt{\left(\frac{t - t_0}{t_E}\right)^2 + u_0^2}$$

For PSBL + : (Rhie, 1997; Witt, 1990; Witt & Mao, 1995)

numerically $$w = z - \sum_{j=1}^{N_{\text{lens}}} \frac{\epsilon_j}{\bar{z} - \bar{r}_j}$$



Model Lightcurves

Two ways to compute finite-source magnification

Initial parameter set → Generate theoretical lightcurve

$$F_i(t) = f_{s,i} \times A(t, x) + f_{b,i}$$

$$A(t) = \frac{f_{\text{images}}(t)}{f_s}$$

flux

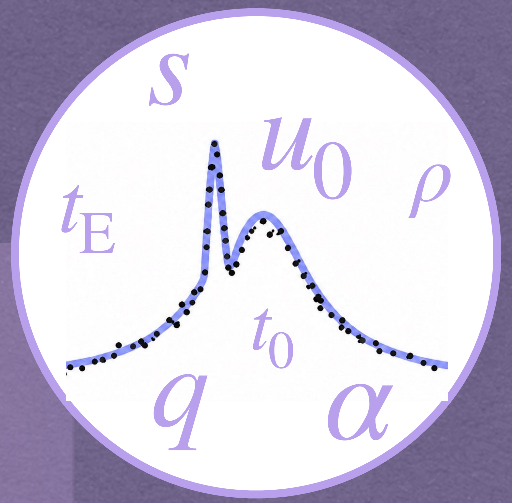
Surface brightness

$$f = \int I(\boldsymbol{\theta})\, d\Omega$$

angular area

(Refer to Bachelet's talk)

Contour Integration	Image-Centered Ray Shooting
Map the source boundary into the image plane and use Green's theorem to compute the areas enclosed by the image boundaries.	Build grids around the images, map each image-plane grid point back to the source plane, and sum the area elements that land inside the source.
Usually faster for binary lenses and uniform sources	Slower, but handles better complex geometries (i.e. 3L, 4L, etc)
Adaptive boundary sampling are harder to run in parallel	Grid points are independent, so easier to run in parallel
Limb darkening requires annuli (rings) for the contour formulae	Weight each accepted area element by local source brightness
Gould & Gaucherel 1997. E.g. VBMicrolensing, BAGLE	Bennett & Rhie 1996. E.g. eesunhong



Model Lightcurves

How Fitting Works

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior)

$$x = (t_E, t_0, u_0, \pi_{EN}, \pi_{EE}, s, \alpha, \rho, [etc])$$

$$F_i(t) = f_{s,i} \times A(t, x) + f_{b,i}$$

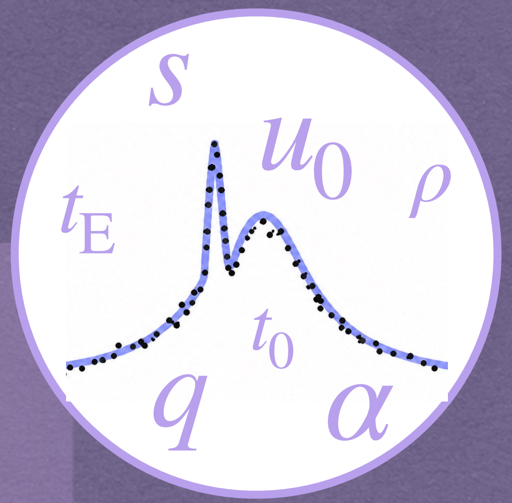
- Compare model vs. data
- Check residuals: data – model
- Objective functions:
  - χ2: weighted residuals
  - Likelihood: probability of data given model
  - Posterior: likelihood + priors

$$\chi^2 = \sum_i \left(\frac{\text{data} - \text{model}}{\text{error}} \right)^2$$

$$\ln(\text{likelihood}) = -\frac{1}{2}\chi^2 + C$$

$$\text{posterior} \propto \text{likelihood} \times \text{prior}$$

Save this number so you can compare how different theoretical lightcurves fit the data!

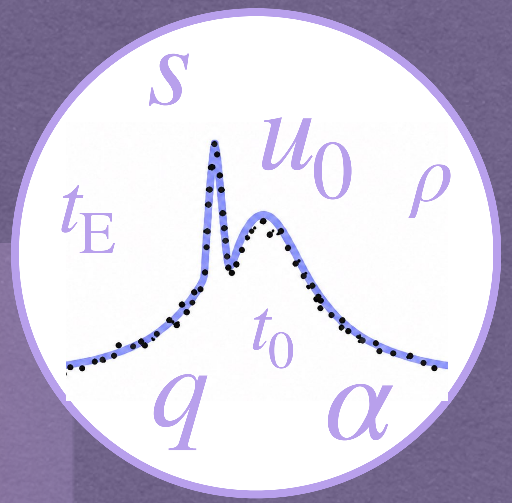


Model Lightcurves

How Fitting Works

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior) → Choose next parameter set

Stage	Method	Next Parameter Set	Strength	Limitation
Explore	Grid search	Next set of the predefined grid	Systematic broad exploration; can reveal separated solutions	Computationally expensive; may miss narrow solutions if the grid is coarse
Refine	Gradient-based methods	Take a step in the direction of the slope	Fast near a good solution	Sensitive to initialization, may find local minima
Refine	Derivative-free methods	Based on previous successes and failures	No derivatives required;, Works with irregular objectives	Often requires many evaluations, may still find local minima
Characterize posterior	MCMC	Random new parameter set near the current one	Posterior distributions and uncertainties	Slow; may struggle with separated modes
Characterize posterior	Nested sampling	New point from the prior where the likelihood is higher	Posterior distributions and Bayesian evidence; can handle multiple modes	Computationally expensive



Model Lightcurves

How Fitting Works

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior) → Choose next parameter set

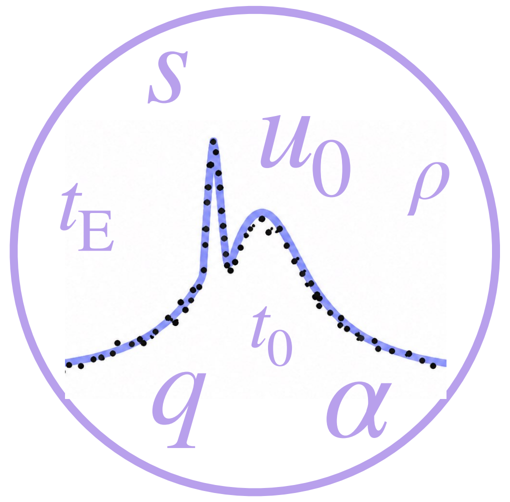
Repeat

When do we stop the "repeat"?

- Grid search: after the grid is complete
- Optimization: when the fit stops improving
- MCMC: when the posterior is well sampled
- Nested sampling: when the remaining contribution to the evidence is small

Compare the saved objective values to identify your best model.

Always check: residuals, convergence, degeneracies, and physical plausibility



Model Lightcurves

Fitting Microlensing Models to Observed Lightcurves

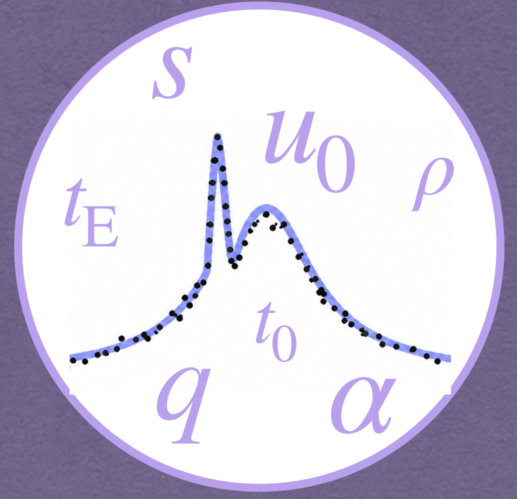
There are microlensing tools for this!

- For a Model Category (e.g. 1L1S, 2L1S, etc):

Repeat

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior) → Choose next parameter set

Microlensing Model-Fitting Tools



Model Lightcurves

BAGLE (Bayesian Analysis of Gravitational Lensing Events) (Lu et al. 2026)

- Open-source Python package for modeling and fitting microlensing events
- Designed for **joint photometric and astrometric analysis**
- Uses PyMultiNest **nested sampling** to explore degeneracies, multimodal solutions, and calculate Bayesian evidence
- **Especially powerful for measuring the masses of dark lenses,** including isolated stellar-mass black holes with Roman and Gaia

eesunhong (in honor of Sun Hong Rhie) (Bennett & Rhie 1996)

- 1st developed in 1995 to demonstrate **microlensing can detect Earth-mass planets**
- Focus on low mass planets from space (2000 Galactic Exoplanet Survey Telescope; Bennett & Rhie 2002)
- **Uses Initial Condition Grid Search** , followed by Metropolis–Hastings $\chi2$ minimization and, optionally, MCMC posterior sampling
- Performs best for **low-mass planets**
- Used to model many published ground-based planetary events

MulensModel (Poleski & Yee 2019)

- Open-source Python package for modeling single- and binary-lens microlensing light curves
- Combines **multiple magnification methods to balance speed and accuracy**
- Calculates $\chi2$, while **allowing users to choose their own fitting algorithm**, such as MCMC or MultiNest
- Used to model many published ground-based planetary events

More options: https://rges-pit.org/tools/

pyLIMA (python Lightcurve Identification and Microlensing Analysis) (Bachelet et al. 2017)

- **First fully open-source** microlensing software
- Primarily designed to fit real microlensing data, with additional simulation module
- Supports Differential Evolution for global searches, LM/TRF for local optimization, and MCMC for posterior sampling
- **Included in the MSOS pipeline** to model single lens events and to perform MCMC
- Used to model many published ground-based planetary events

RTModel (Real Time Modeling) (Bozza 2024)

- Developed to model **OGLE & MOA alerts with real-time follow-up**
- Uses a lightcurve catalog to identify binary lens model initial conditions: **great for stellar binary lens**, then uses Levenberg–Marquardt fitting with a "bumper" mechanism
- **Fast, automated**
- Obvious choice as basis for the **MSOS pipeline to model binary events** (including planetary events)
- Helped identify promising planetary candidates and used for many published events

Microlensing Model-Fitting Tools

Model Lightcurves

Lu et al. 2026

Table 3. PSPL Model-Generation Runtimes for Photometry-Only Models Without Parallax

Model	Runtime (ms) Full	Pre-Instantiated
BAGLE	0.066 ± 0.008	0.049 ± 0.005
pyLIMA	12.76 ± 1.02	0.20 ± 0.02
MulensModel	1.03 ± 0.71	0.24 ± 0.01
VBMicrolensing	0.109 ± 0.003	0.113 ± 0.002

Table 4. PSPL Model-Generation Runtimes for Models With Parallax

	Runtime (ms)			
	Photometry-Only		Photometry+Astrometry	
Model	Full	Pre-Instantiated	Full	Pre-Instantiated
BAGLE	0.32 ± 0.03	0.29 ± 0.18	0.87 ± 0.17	0.76 ± 0.36
pyLIMA	57.07 ± 33.32	0.20 ± 0.05	106.91 ± 31.91	0.50 ± 0.12
MulensModel	2.11 ± 0.34	0.31 ± 0.16	-	-
VBMicrolensing	8.57 ± 7.63	0.11 ± 0.01	129.12 ± 8.43	0.57 ± 1.54

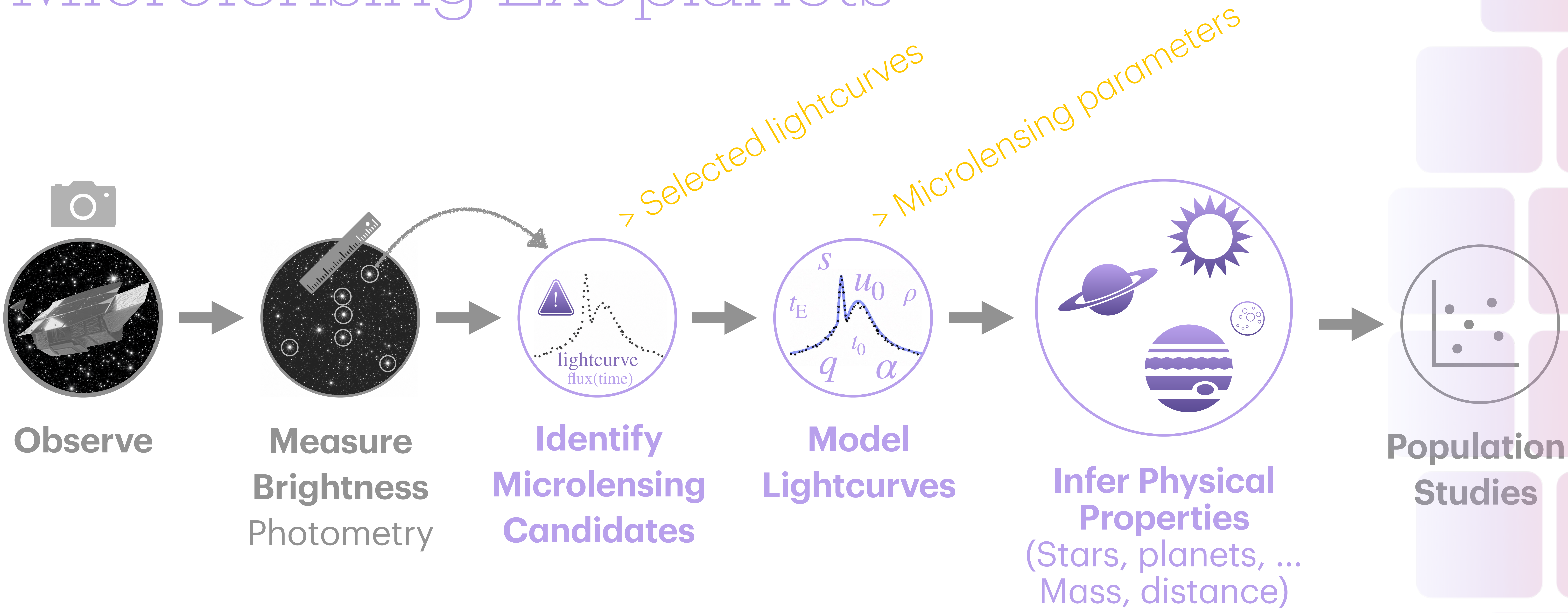
Table 5. FSPL Model-Generation Runtimes for Photometry-Only Models Without Parallax

Model	Runtime (ms) Full	Pre-Instantiated
BAGLE	60.498 ± 0.293	61.085 ± 0.513
VBMicrolensing	0.052 ± 0.027	0.020 ± 0.012

Table 6. Model-Fit Runtimes for Photometry-Only with Parallax

Model	Runtime (s)	$\bar{\chi}^2$	Notes
BAGLE	6.97	0.95	MultiNest Nested Sampling
RTModel	9.48	0.96	Levenberg–Marquardt Fitting
pyLIMA	18.48	0.96	MCMC Fitting
MuLens	8.72	0.96	MultiNest Nested Sampling

From Images to Microlensing Exoplanets



Physical properties

Infer Physical Properties

(Refer to Bhattacharya's talk)

We want: m_{planet} $M_{\text{host star}}$ D_{L}

$$M_L = m_{\text{planet}} + M_{\text{host star}}$$

Planetary lightcurve model gives:

- q: planet-to-host mass ratio $q = \dfrac{m_{\text{planet}}}{M_{\text{host star}}}$
- Finite-source effects (source angular size / ρ = θE: angular Einstein radius) $\theta_{\text{E}} = \sqrt{\dfrac{4GM_L}{c^2 D_S}\left(\dfrac{D_S}{D_L} - 1\right)}$

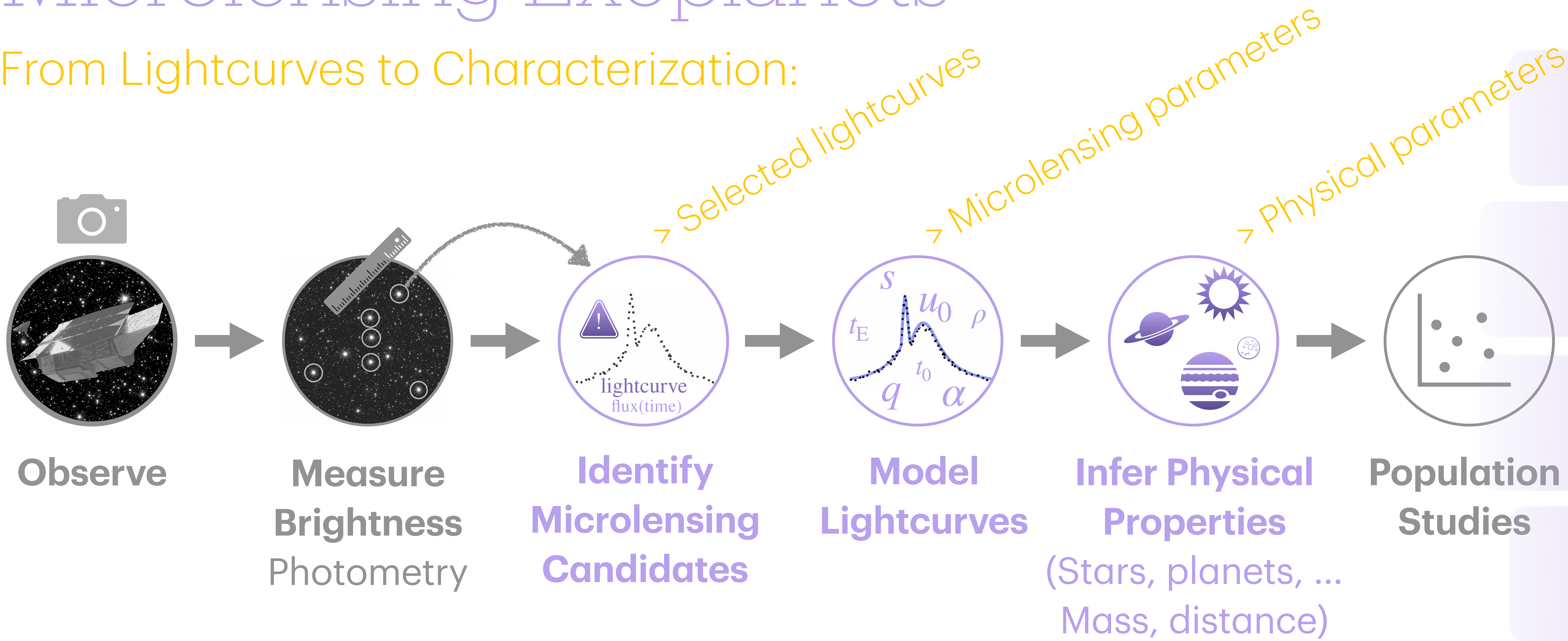
To infer masses and distances, we need **two of three**:

- θE: Einstein-radius constraint ✅
- πE: microlensing parallax constraint $\pi_E = \sqrt{\dfrac{c^2\left(D_S - D_L\right)}{4GM_L D_S D_L}}$
- Lens flux: brightness constraint from high-resolution follow-up (Refer to Bhattacharya, Terry, Beaulieu's talk)

If fewer constraints are available,
we use Galactic models (refer to Huston's talk) to estimate the physical parameters statistically.

From Images to Microlensing Exoplanets

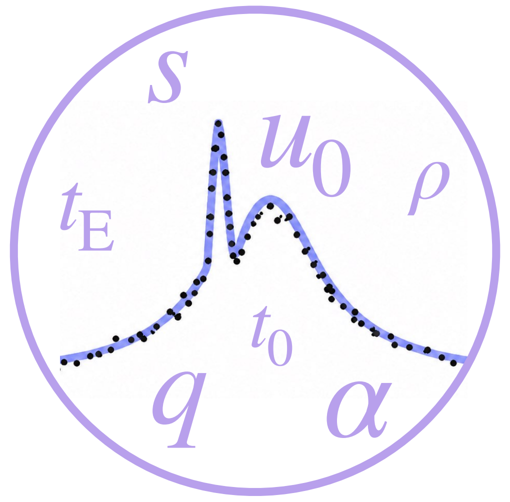
From Lightcurves to Characterization:



Summary:

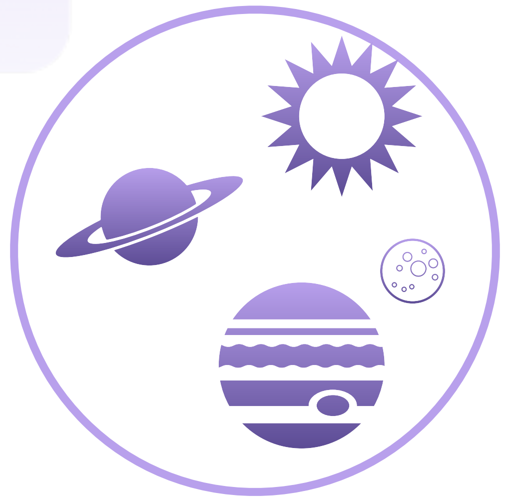
Identify Microlensing Candidates

- We cannot model every lightcurve
- Select promising candidates using the expected features of microlensing events (how it looks like)



Model Lightcurves

- Start with the simplest model
- Add complexity only when needed
- Generate theoretical lightcurves and compare them to the data
- Explore the parameter space to find good solutions
- Use different methods for different goals: exploration, optimization, posterior distribution
- Different microlensing tools are available



Infer Physical Properties

- We need two of these three - θE, πE, and lens flux - to infer mass and distance.
- If direct constraints are missing, estimate physical parameters statistically using Galactic models + the parameters from your lightcurve model

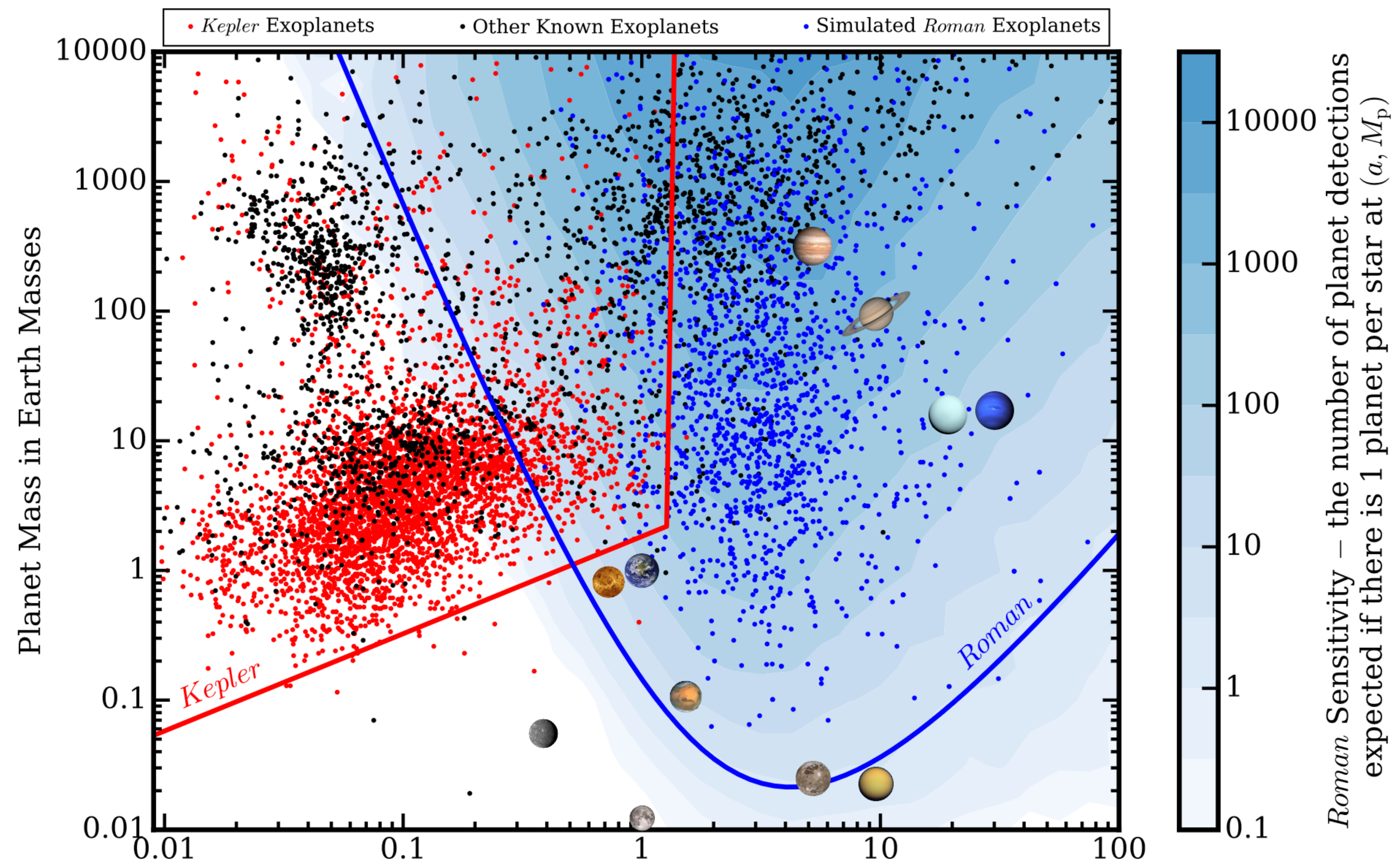
Roman is launching next month!!! ✨

NANCY GRACE ROMAN SPACE TELESCOPE

- Hundred of millions of precise light curves!
- Thousands of microlensing planets!
- You can be part of the microlensing community and help us!

Thank you!

These results won't be only simulations 🤩



Credit: Penny et al. (2019) ApJS 241, 3

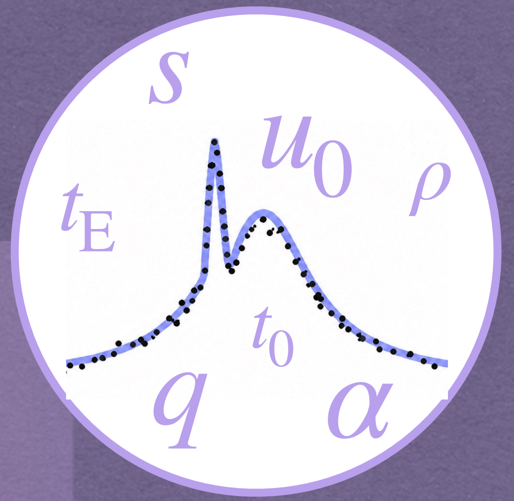
BACK UP SLIDES

A Collection of Lightcurves that We Believe Are Worth Modeling

Identify Microlensing Candidates

- Stricter Selection
  - Higher purity
  - More genuine microlensing events missed
- More Permissive Selection
  - Higher completeness
  - More non-microlensing contaminants

The optimal balance depends on your scientific goals and available resources.

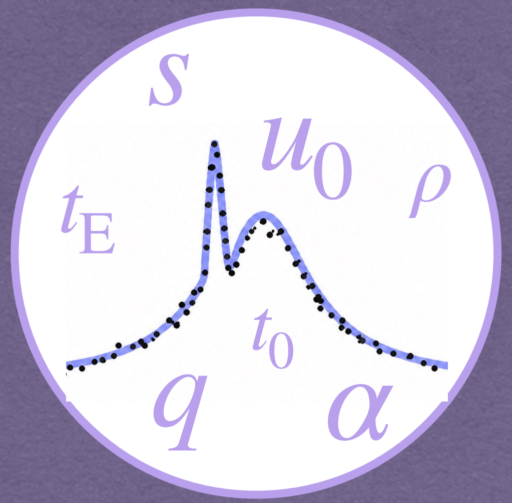


Model Lightcurves

How fitting works

Initial parameter set → Generate theoretical lightcurve → Evaluate objective (χ^2, likelihood, posterior) → Choose next parameter set

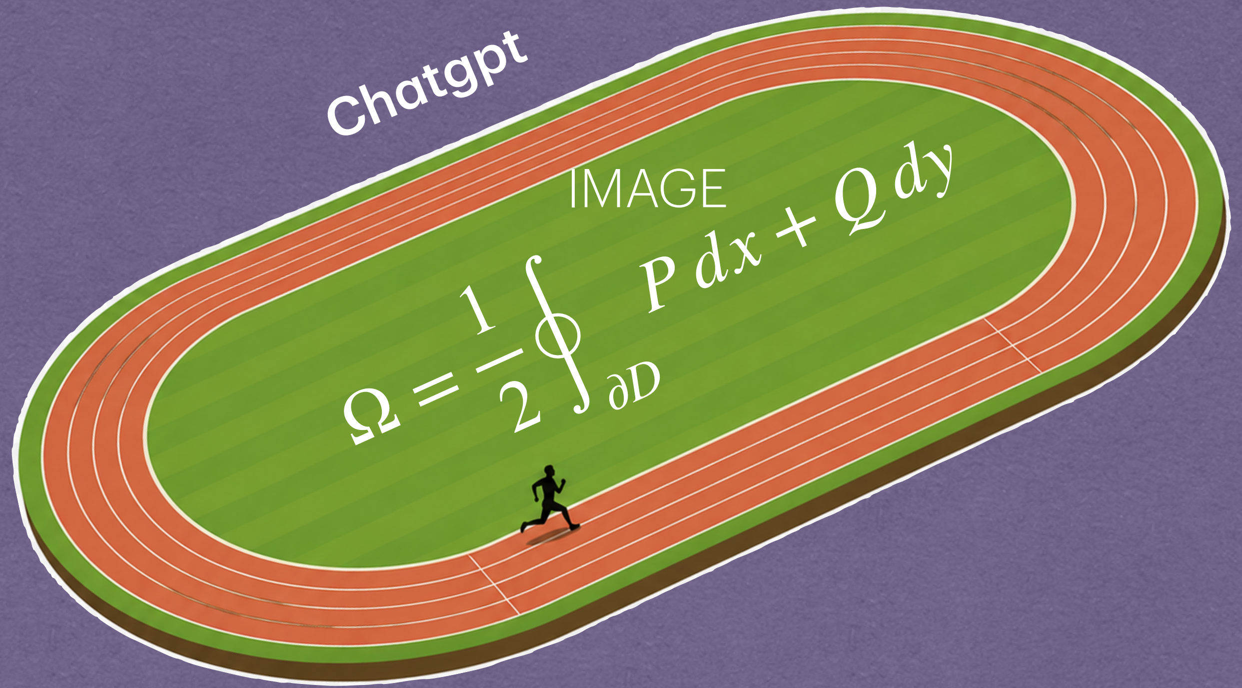
Repeat multiple times



Model Lightcurves

Two ways to compute finite-source magnification

Generate theoretical lightcurve



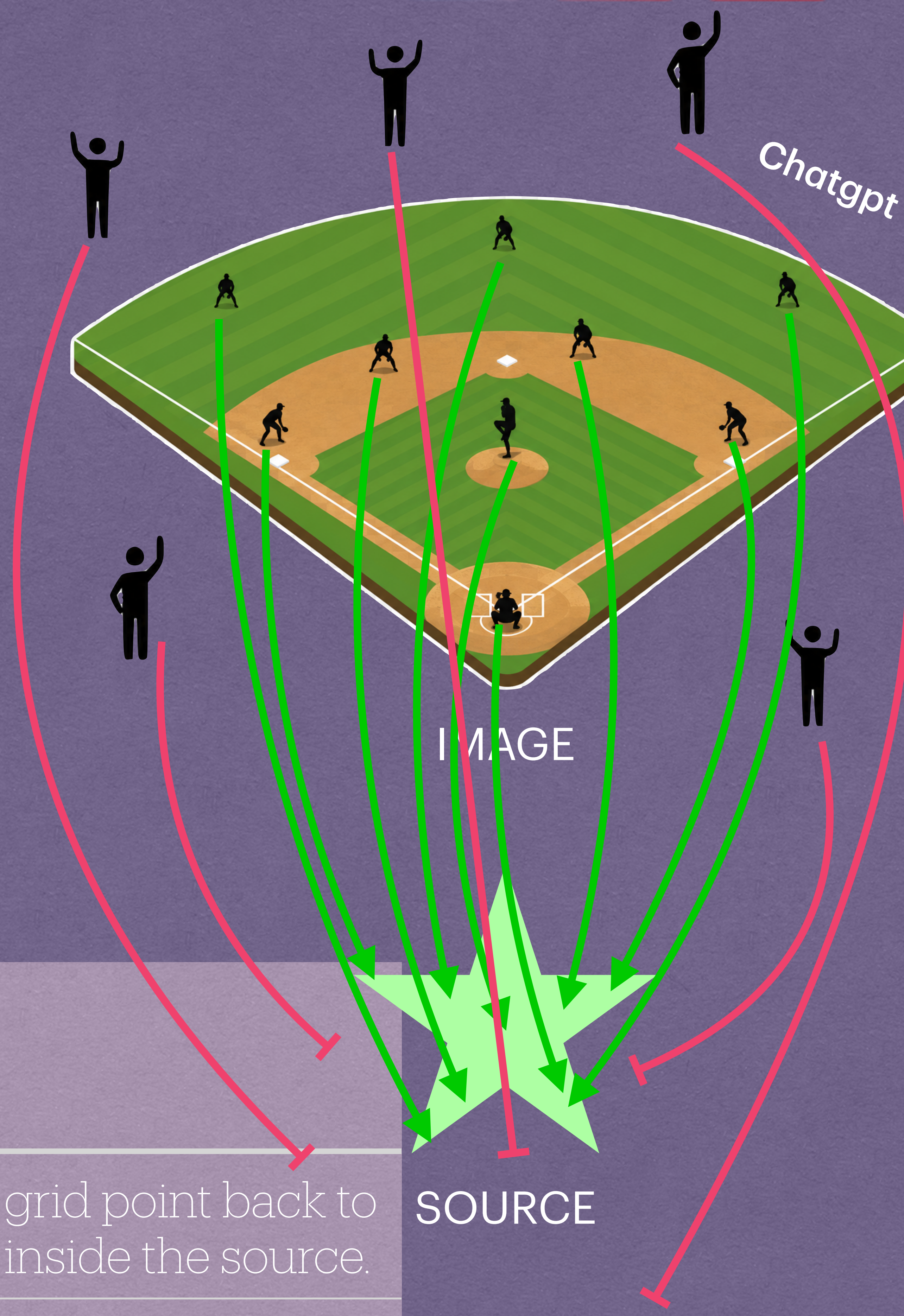
$$A(t) = \frac{f_{\text{images}}(t)}{f_s}$$

Flux

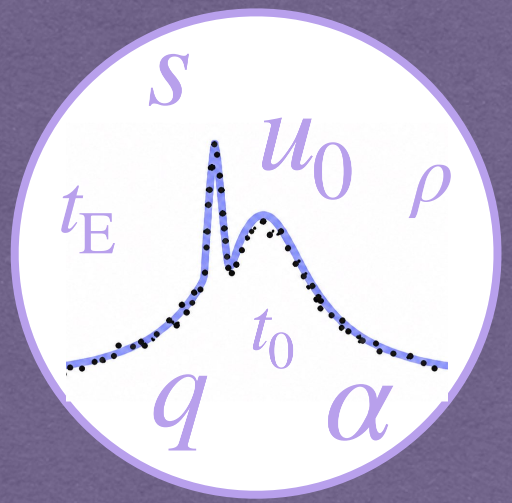
Surface brightness

$$f = \int I(\boldsymbol{\theta})\, d\Omega$$

angular area



Contour Integration	Image-Centered Ray Shooting
Map the source boundary into the image plane and use Green's theorem to compute the areas enclosed by the image boundaries.	Build grids around the images, map each image-plane grid point back to the source plane, and sum the area elements that land inside the source.
Usually faster for binary lenses and uniform sources	Slower, but handles better complex geometries (i.e. 3L, 4L, etc)
Adaptive boundary sampling are harder to run in parallel	Grid points are independent, so easier to run in parallel
Limb darkening requires annuli (rings) for the contour formulae	Weight each accepted area element by local source brightness
Gould & Gaucherel 1997. E.g. VBMicrolensing, BAGLE	Bennett & Rhie 1996. E.g. eesunhong



Model Lightcurves

What software should I use for my microlensing model?

The one you're most likely to succeed using it!

BAGLE — photometric and astrometric microlensing; Maintained: Yes (Moving Universe Lab); Lu et al. (2025)
MulensModel — Roman Microlensing Data Challenge workshop ntained: Yes (Poleski); Poleski & Yee (2019)
pyLIMA — MSOS ling and simulation toolkit; Maintained: Yes (Bachelet); Bachelet et al. (2017)
RTModel — Roman preparation / MSOS ith built-in interpretation; Maintained: Yes (Bozza); Bozza (2024)
VBBinaryLensing — muLAn ır integration approach to binary lenses; Maintained: No; Bozza (2010)
VBMicrolensing — Microlensing Book Chapter (in press) Vandorou & Ishitani Silva ained: Yes (Bozza); Bozza et al. (2025)
eesunhong— MOA-2020-BLG-135Lb; Ishitani Silva et al 2022 Yes (Bennett /Olmschenk); Bennett & Rhie (1996), Bennett (2010)
muLAn — WFIRST Microlensing Data Challenge 2018 ; Street et al (in prep) Cassan/Ranc); Ranc & Cassan (2018)
triplelens — light curves and image positions for triple systems; Maintained: No (Kuang); Kuang et al. (2021)
sfit_minimizer — fitting utilities; Maintained: Yes; Yee & Gould (2025)
SingleLensFitter — single-lens fits with finite-source effects; Maintained: No (Albrow)

Source: https://rges-pit.org/tools/