

# microJAX: A Differentiable Framework for Microlensing Modeling with GPU-Accelerated Image-Centered Ray Shooting

SHOTA MIYAZAKI <sup>1</sup> AND HAJIME KAWAHARA <sup>1</sup>

<sup>1</sup>*Institute of Space and Astronautical Science, Japan Aerospace Exploration Agency, 3-1-1 Yoshinodai, Chuo, Sagami-hara, Kanagawa 252-5210, Japan*

(Revised October 6, 2025; Accepted October 6, 2025)

Submitted to ApJ

## ABSTRACT

We introduce `microJAX`, the first fully differentiable implementation of the image-centered ray-shooting (ICRS) algorithm for gravitational microlensing. Built on `JAX` and its XLA just-in-time compiler, `microJAX` exploits GPU parallelism while providing exact gradients through automatic differentiation. The current release supports binary- and triple-lens geometries, including limb-darkened extended-source effects, and delivers magnifications that remain differentiable for all model parameters. Benchmarks show that `microJAX` matches the accuracy of established packages and attains up to a factor of  $\sim 5$ -6 speed-up in the small-source, limb-darkened regime on an NVIDIA A100 GPU. Since the model is fully differentiable, it integrates seamlessly with probabilistic programming frameworks, enabling scalable Hamiltonian Monte Carlo and variational inference workflows. Although the present work focuses on standard microlensing magnification models, the modular architecture is designed to support upcoming implementations of microlensing higher-order effects, while remaining compatible with external likelihood frameworks that incorporate advanced noise models. `microJAX` thus provides a robust foundation for precise and large-scale surveys anticipated in the coming decade, including the *Nancy Grace Roman Space Telescope*, where scalable, physically self-consistent inference will be essential for maximizing scientific return.

*Keywords:* gravitational lensing: micro,

## 1. INTRODUCTION

Recent advances in differentiable programming and probabilistic inference have opened new possibilities for precision modeling in astronomy. Tools such as `Theano` (The Theano Development Team et al. 2016), `Tensorflow` (Abadi et al. 2015), `JAX` (Bradbury et al. 2018), and `PyTorch` (Paszke et al. 2019) allow the construction of fully differentiable computational graphs, supporting high-performance automatic differentiation. When combined with probabilistic programming languages (PPLs) such as `Stan` (Carpenter et al. 2017) and `NumPyro` (Bingham et al. 2018; Phan et al. 2019), these frameworks enable the direct application of efficient gradient-based optimization and inference methods, such as Hamiltonian Monte Carlo (HMC; Duane et al. 1987; Hoffman & Gelman 2011; Betancourt 2017), to complex scien-

tific models. This has led to successful applications with high-dimensional Bayesian inference, for example, in exoplanetary system modeling (Agol et al. 2021; Masuda et al. 2024), X-ray spectral fitting (Dupourqué et al. 2024), atmospheric retrievals of exoplanets (Kawahara et al. 2022, 2025), gravitational waves (Isi & Farr 2021; Wong et al. 2023; Edwards et al. 2024), and so on.<sup>12</sup> With the diversification of scientific goals and the advent of next-generation surveys, microlensing modeling faces new demands. The upcoming *Nancy Grace Roman Space Telescope* (Spergel et al. 2015; Akeson et al. 2019) will provide high-cadence, high-precision light curves for tens of thousands of microlensing events, including planetary systems, dark lenses, and multi-component configurations (Bennett & Rhie 2002; Penny et al. 2019; Lam et al. 2023). To fully leverage this dataset,

Corresponding author: Shota Miyazaki  
miyazaki@ir.isas.jaxa.jp

<sup>1</sup> <https://github.com/JAXtronomy/awesome-JAXtronomy>

<sup>2</sup> <https://juliaastro.org/home/>

modeling frameworks must handle diverse lens geometries, extended-source effects, complex surface brightness profiles, and other higher-order effects while remaining scalable and compatible with gradient-based inference. These challenges motivate the development of robust numerical methods that integrate differentiable programming techniques.

Modeling microlensing magnification becomes significantly more challenging when both multiple-lens geometries and finite-source effects are involved. While point-source calculations can be handled analytically or with efficient root-finding methods, extended-source models require numerical integration over the lensed images. There are two main approaches commonly used: contour integration (Gould & Gauchere 1997; Dominik 1998a) and image-centered ray shooting (Bennett & Rhie 1996; Dong et al. 2006; Bennett 2010). The contour integrating method utilizes Green’s theorem to transform the two-dimensional image integral into a one-dimensional integral over the boundaries of the image. This technique is highly efficient and has been widely adopted in binary-lens modeling, particularly in the well-established public package `VBBinaryLensing` (Bozza 2010; Bozza et al. 2018). However, although great efforts have been made to extend the contour integration technique to more complex lens configurations, along with high accuracy and performance (Bozza et al. 2025), the method inherently encounters difficulties in determining the correct orientation and connectivity of image boundaries as the number of lenses increases, which is essential for applying Green’s theorem. Moreover, for sources with spatially complex brightness profiles, the integrals require finer quadrature and significantly higher computational cost (Dominik 1998a).

In contrast, the image-centered ray shooting method (ICRS; Bennett & Rhie 1996; Bennett 2010) straightforwardly performs numerical integration directly on the image plane by mapping rays back to the source plane. This approach is more flexible in handling complex lens configurations and is naturally suited for arbitrary surface brightness profiles, such as those with starspots (e.g., Hendry et al. 2002; Giordano et al. 2015). Its robustness near caustics and general applicability make it a powerful alternative to contour integration. However, the method typically requires dense sampling over each image, thus posing higher computational costs, particularly for small sources where the magnification, defined as the image area per unit source area, demands finer resolution of image areas to maintain accuracy. On the other hand, the inverse ray shooting (Kayser et al. 1986; Wambsganss 1997) is inherently parallelizable across image-plane pixels; a variety of GPU-accelerated schemes have been developed for cosmological microlensing (e.g., Thompson et al. 2014; Zheng et al. 2022; Weisenbach 2025). However, only a few exploratory GPU ports of the ICRS algorithm have

been published (Hundertmark 2011; McDougall & Albrow 2016), and none has yet undertaken a systematic optimisation of accuracy versus computational performance, which likely reflects the difficulty of adaptive image-centered mapping onto GPU architectures.

To address these challenges, we present `microJAX` (Miyazaki & Kawahara 2025), a differentiable, GPU-accelerated microlensing modeling framework based on the ICRS method. The key contributions of this work are summarized as follows:

- **Differentiable ICRS on GPU:** We implement the first fully differentiable ICRS algorithm compatible with JAX, enabling efficient automatic differentiation. The code leverages XLA compilation to exploit GPU parallelism while maintaining a static computation graph, which facilitates fast and stable magnification evaluations.
- **Fast and High-precision modeling of complex lens systems:** `microJAX` supports binary and triple lens systems with arbitrary surface brightness profiles. Using a polar grid discretization and smooth interpolation near image boundaries, we achieve relative magnification errors  $\lesssim 10^{-4}$  even for small sources.<sup>3</sup>
- **End-to-end gradient-based inference:** The framework is designed to interface seamlessly with probabilistic programming tools. We demonstrate HMC inferences on both synthetic and real microlensing light curves, showing that `microJAX` can serve as a building block for full Bayesian pipelines.

In the context of prior work, our approach differs from two dominant lines of development. Traditional ICRS implementations (e.g., Bennett & Rhie 1996; Bennett 2010) are not differentiable and have not been optimized for GPU acceleration. Conversely, recent JAX-based codes such as `caustics` (Bartolić 2023) and `microlux` (Ren & Zhu 2025) achieve differentiability by adapting contour integration techniques. Still, they are currently limited to binary lenses and/or rely on adaptive sampling structures, which pose challenges for parallelization. `microJAX` thus offers a unified solution that is both scalable and differentiable, bridging these two approaches. `microJAX` is available on GitHub as a Python library with documentation and GPU support, aiming to provide a foundation for further development and community contributions.<sup>4</sup>

In this paper, we describe the design and implementation of `microJAX`, benchmark its computational performance and accuracy against existing tools, and demonstrate its ap-

<sup>3</sup> This accuracy refers to the default configuration and is not a fundamental limit. Higher precision can, in principle, be obtained by increasing the radial and angular resolutions in the ray shooting at the expense of computation time and memory usage.

<sup>4</sup> <https://github.com/ShotaMiyazaki94/microjax>

plicability in both idealized and realistic inference scenarios. Section 2 reviews the fundamentals of microlensing magnification modeling and introduces the image-centered ray shooting (ICRS) method. Section 3 details the differentiable implementation of `microJAX` using JAX. Section 4 presents computational benchmarks and accuracy comparisons with existing microlensing tools on both CPU and GPU architectures. In Section 5, we illustrate the integration of `microJAX` into gradient-based inference workflows using synthetic light curves. Finally, Section 6 summarizes the key results and discusses future directions.

## 2. BACKGROUND

To motivate the design of our modeling framework, we first summarize the mathematical foundations of gravitational microlensing with multiple lenses. In particular, we review the lens equation and its implications for the image formation process, which forms the basis for our inverse-ray shooting approach. We then briefly describe the image-centered ray shooting (ICRS) method, which serves as the computational backbone of `microJAX`.

### 2.1. Multiple Lens Microlensing

Assuming that all lenses are point masses, the lens equation for an  $N$ -body system can be written (Kayser et al. 1986; Witt 1990) as

$$\mathbf{w} = \mathbf{z} - \sum_{i=1}^N \frac{\epsilon_i}{\bar{\mathbf{z}} - \mathbf{a}_i}, \quad (1)$$

where  $\mathbf{w}$  and  $\mathbf{z}$  are the source and image positions in complex coordinates, respectively, and  $\mathbf{a}_i$  denotes the position of the  $i$ -th lens.  $\bar{\mathbf{z}}$  is defined as a conjugate of  $\mathbf{z}$ . The parameter  $\epsilon_i$  represents the fractional mass of the  $i$ -th lens normalized by the total lens mass  $M$ , so that  $\sum_i \epsilon_i = 1$ . All coordinates are expressed in units of the Einstein radius corresponding to  $M$ . Because of the complex conjugation, this lens equation defines a nonlinear and non-analytic mapping from the image plane to the source plane. As a result, obtaining the image positions  $\mathbf{z}$  for a given source position  $\mathbf{w}$  generally requires numerical root-finding methods.

When assuming a point source, the total magnification is computed as the sum of the signed magnifications of all images. Each image contributes an amount equal to the inverse of the absolute value of the Jacobian determinant of the lens mapping  $\mathbf{z} \mapsto \mathbf{w}$ , evaluated at the corresponding image position  $\mathbf{z}_j$ :

$$A(\mathbf{w}) = \sum_{j=1}^{N_{\text{image}}} \frac{1}{|\det J(\mathbf{z}_j)|}, \quad (2)$$

where  $N_{\text{image}}$  is the number of images. The Jacobian determinant at  $\mathbf{z}_j$  is given<sup>5</sup> by

$$\det J(\mathbf{z}_j) = 1 - \left| \sum_{i=1}^N \frac{\epsilon_i}{(\bar{\mathbf{z}}_j - \mathbf{a}_i)^2} \right|^2. \quad (3)$$

To evaluate Equation (2), all image positions  $\{\mathbf{z}_j\}$  corresponding to a given source position  $\mathbf{w}$  must first be found by solving the lens equation. For multiple point-mass lenses, Equation (1) can be reformulated as a complex polynomial in  $\mathbf{z}$ :

$$\mathcal{P}(\mathbf{z}|\mathbf{w}, \{\mathbf{a}_i\}, \{\epsilon_i\}) = \sum_{l=0}^{N_{\text{deg}}} b_l \mathbf{z}^l = 0, \quad (4)$$

where the polynomial degree is  $N_{\text{deg}} = 5(N - 1)$  for an  $N$ -lens system (e.g., Rhie 2003), yielding  $N_{\text{deg}} = 5$  and 10 for binary and triple lenses, respectively. The number of physical images ranges from  $N + 1$  to  $5(N - 1)$ , but not all roots of Equation (4) correspond to real images. Some may be spurious or correspond to non-physical solutions, and thus each root of images must be validated by substituting it back into Equation (1). The magnification diverges where  $\det J = 0$ , which defines the critical curves in the image plane and maps to caustics in the source plane. These regions produce strong, distinctive magnification features and are particularly sensitive to modeling errors.

### 2.2. Image-Centered Ray Shooting (ICRS)

For extended sources, the total magnification is given by the surface-brightness-weighted integral of the point-source magnification over the source disk. In the presence of caustic structures, particularly during near-caustic or caustic crossings, the magnification pattern can exhibit sharp spatial gradients, rendering the point-source approximation and analytic methods (e.g., hexadecapole expansions; Gould 2008; Cassan 2017) inaccurate. In such cases, numerical integration is required to evaluate the finite-source magnification.

In the ICRS method, the lens equation is first solved to determine the point-source image positions  $\{\mathbf{z}_j\}$  for a given center of source position  $\mathbf{w}_c$  (and arbitrary positions within the source). Around each image  $\mathbf{z}_j$ , a localized grid is constructed on the image plane, which must fully enclose the extent of the corresponding image to ensure that all contributing rays are captured. Each grid point  $\mathbf{z}_j^{(k)}$  is mapped back to the source plane via the lens equation:  $\mathbf{w}_j^{(k)} = f_{\text{map}}(\mathbf{z}_j^{(k)})$ , where  $f_{\text{map}}$  denotes the mapping defined by Equation (1). We then define an indicator function that determines whether each mapped point lies within the angular radius  $\rho$  of the

<sup>5</sup> The lens equation involves complex conjugation ( $\bar{\mathbf{z}}$ ), making the mapping  $\mathbf{z} \mapsto \mathbf{w}$  non-analytic (non-holomorphic).

source (in units of  $\theta_E$ ):

$$\chi_j^{(k)} = \begin{cases} 1 & \text{if } |\mathbf{w}_j^{(k)} - \mathbf{w}_c| < \rho, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

The total finite-source magnification is then approximated as

$$A(\mathbf{w}_c, \rho) \approx \frac{1}{F_0} \sum_j^{N_{\text{image}}} \sum_k^{N_{\text{grid}}} \chi_j^{(k)} I(\mathbf{w}_j^{(k)}) \Delta(\mathbf{z}_j^{(k)}), \quad (6)$$

where  $N_{\text{grid}}$  is the number of grid points per image,  $I(\mathbf{w})$  is the source surface brightness profile (e.g., uniform or limb-darkened),  $\Delta(\mathbf{z}_j^{(k)})$  is the area element associated with each grid point in the image plane, and the normalization  $F_0$  is the total unlensed source flux:

$$F_0 = \int_{|\mathbf{w}' - \mathbf{w}_c| \leq \rho} I(\mathbf{w}') d^2 \mathbf{w}'.$$

This method enables accurate integration of the lensed surface brightness while maintaining computational efficiency by restricting the calculation to regions around the true image positions. Furthermore, the mapping  $f_{\text{map}}(\mathbf{z})$  is inherently local and independent across grid points, making the algorithm naturally amenable to parallel execution on GPUs. However, achieving high-accuracy magnification estimates requires careful treatment of image boundaries and integration schemes (Bennett 2010). We describe our implementation and associated optimizations in the following section.

### 3. IMPLEMENTATION

To enable fast and high-precision microlensing modeling on GPUs, `microJAX` implements a parallelizable version of the ICRS algorithm. All computational components are designed to be fully compatible with automatic differentiation, thereby supporting gradient-based optimization and inference techniques. In this section, we describe the key algorithmic components of `microJAX`, beginning with the numerical solution of the lens equation.

#### 3.1. Solving the Lens Equation

Numerical solutions to Equation (4) can be obtained by casting the polynomial into an  $N_{\text{deg}} \times N_{\text{deg}}$  companion matrix and computing its eigenvalues. Although this eigenvalue formulation guarantees all  $N_{\text{deg}} = 5(N - 1)$  roots without initial guesses, the computational cost scales as  $\mathcal{O}(N_{\text{deg}}^3)$ , and it can suffer from numerical instability in some cases. Consequently, contemporary microlensing codes prefer iterative root-refinement schemes such as Laguerre’s method and the updated version of the algorithm proposed by Skowron & Gould (2012), which converge quadratically once a reasonable starting estimate is available.

In `microJAX`, we adopt the Ehrlich-Aberth (EA) method (Ehrlich 1967; Aberth 1973; Fatheddin & Sajadian 2022), which simultaneously refines all root estimates. The update rule is given by

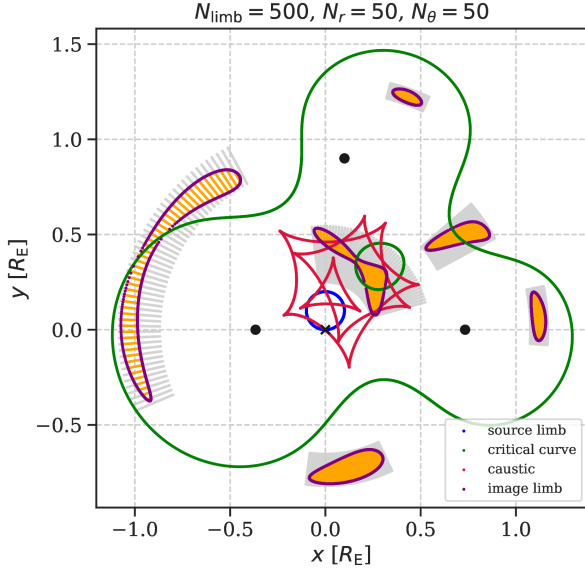
$$z_k^{(n+1)} = z_k^{(n)} - \left( \frac{\mathcal{P}'(z_k^{(n)})}{\mathcal{P}(z_k^{(n)})} - \sum_{j \neq k} \frac{1}{z_k^{(n)} - z_j^{(n)}} \right)^{-1}, \quad (7)$$

where  $\mathcal{P}(z)$  is the polynomial defined in Equation (4), and  $\mathcal{P}'(z)$  is its derivative. This method extends Newton’s method by introducing a repulsive term between root estimates, enhancing convergence stability and mitigating root coalescence. A key advantage of the EA method is its inherent parallelism: all root estimates are updated simultaneously at each iteration without sequential dependencies. This property makes it particularly well-suited for massively parallel execution on GPUs (Weisenbach 2025). Moreover, its deterministic control flow and fixed iteration count enable efficient compilation within JAX’s static computation graph, supporting just-in-time (JIT) compilation and seamless integration with automatic differentiation frameworks.

The initial root estimates  $\{z^{(0)}\}$  are placed uniformly on a circle in the complex plane, with the radius determined by Cauchy’s bound based on the polynomial coefficients. This choice ensures that all true roots are enclosed and that the iteration remains numerically stable from the outset. In practice, this initialization works reliably across a wide range of lens configurations and is used as the default in `microJAX`. Importantly, the EA method’s ability to update all root estimates simultaneously at each step makes it well-suited for vectorized evaluation using `vmap` in JAX. This enables efficient batch evaluation of many source positions in parallel, allowing the root-finding procedure to scale effectively on GPU architectures. For instance, using an NVIDIA A100 GPU, `microJAX` can solve  $\sim 10^6$  10th-degree complex polynomials, corresponding to triple-lens systems, in under one second. This level of throughput is crucial for modeling large microlensing datasets in modern inference pipelines.

#### 3.2. Defining Image-centered Grid Regions

A key step in the ICRS algorithm is defining compact image-centered grid regions that fully enclose each lensed image while minimizing excess area to reduce computational redundancy. In traditional CPU-based implementations, such regions are often constructed by sequentially exploring the image plane. Starting from the image positions obtained by solving the lens equation for the source center  $\mathbf{w}_c$  (and arbitrary positions within the source), inverse-ray shooting is applied iteratively to nearby points until the full image boundary is traced. While adaptive and efficient, this approach involves branching logic and sequential updates, making it poorly suited for GPU architectures and just-in-time (JIT) compilation in JAX.



**Figure 1.** An example of a triple-lens configuration with parameters  $(q, s, q_3, r_3, \psi, \rho) = (0.5, 1.1, 0.91, 1.46, 0.1)$  and source center  $\mathbf{w}_{\text{center}} = (0, 0.1)$ . The red and green curves represent the caustics and critical curves, respectively, and the black dots indicate the lens positions. The source limb is sampled with  $N_{\text{limb}} = 500$  points (blue), which are mapped to the image plane, producing six distinct images (purple). For each image, a polar-coordinate region is defined based on the image-limb points, within which a uniform  $50 \times 50$  grid is placed for integration. Grid points falling inside the source are marked in orange, while those outside are shown in gray.

Unlike traditional region-growing or adaptive boundary tracing methods, `microJAX` adopts a fully static and parallelizable approach to region construction. For each source position, we uniformly sample the boundary of the source with  $N_{\text{limb}}$  points.<sup>6</sup> Each of these limb points is mapped to the image plane using the lens equation, yielding a set of image-limb points that trace the extent of each image. The sampling positions on the source boundary are given by

$$\mathbf{w}^{(i)} = \mathbf{w}_c + \rho e^{2\pi i / N_{\text{limb}}}, \quad i = 0, 1, \dots, N_{\text{limb}} - 1 \quad (8)$$

where  $\mathbf{w}_c$  is the source center and  $\rho$  is the angular source radius in units of the angular Einstein radius  $\theta_E$ . These mapped points on the image plane define the approximate shapes and locations of the lensed images.

Figure 1 illustrates a result of the grid allocating algorithm for clear visualization. Each grid sector corresponds to a distinct image and is processed independently to compute the finite-source magnification. To construct the grid-locating regions efficiently, we convert the mapped image-limb points

into polar coordinates  $(r, \theta)$  and apply one-dimensional binning (i.e., coarse clustering) separately to the radial and angular components. This procedure identifies intervals where the image-limb points are densely populated. All combinations of radial and angular bins define candidate polar sectors in the image plane, which are tested using binary masks to determine whether they contain any image-limb points. Empty sectors are discarded, while periodic or overlapping ones (e.g., those crossing the  $\theta = 0$  boundary) are merged. The total number of sectors is capped based on the maximum expected number of images. For each remaining sector, we refine the boundaries by computing the minimum and maximum values of  $r$  and  $\theta$  among the enclosed points and applying a small safety margin to ensure complete image coverage. The resulting compact annular sectors tightly enclose the images and are used to place integration grids. This binning-based approach scales as  $\mathcal{O}(N_{\text{limb}})$ , in contrast to traditional adaptive methods such as boundary tracing or pairwise clustering, which typically require  $\mathcal{O}(N_{\text{limb}}^2)$  operations and introduce control-flow branching that hinders JIT compilation. By relying entirely on vectorized operations, our method ensures fast, scalable, and fully differentiable region construction compatible with parallel GPU execution within the JAX framework.

### 3.3. Inverse-Ray Shooting and Integration Rules

Given the polar grid regions defined for each image, we compute the magnification by performing inverse-ray integration within each region. For every annular sector specified by  $(r_{\min}, r_{\max}, \theta_{\min}, \theta_{\max})$ , we construct a fixed-resolution polar grid<sup>7</sup> consisting of  $N_r$  radial and  $N_\theta$  azimuthal points. For each radial slice at fixed radius  $r_i$ , the magnification is computed by summing contributions from azimuthal segments, where the inverse-ray mapping determines whether each grid point maps back inside the source disk. At each azimuthal angle  $\theta_\ell$ , the corresponding grid point in the image plane is mapped back to the source plane via the lens equation, yielding the position  $\mathbf{w}_{i,\ell}$ . The Euclidean distance from the source center is computed as  $d_{i,\ell} = |\mathbf{w}_{i,\ell} - \mathbf{w}_c|$  and is used to evaluate the surface brightness profile of the source,  $I(d_{i,\ell})$ . Near the image boundary, where the transition between inside and outside the source occurs, we estimate the precise edge location ( $d = \rho$ ) by performing cubic interpolation using the distances of nearby grid points. This interpolation enables smooth and accurate integration across partially covered pixels and is particularly important when modeling limb-darkened sources. The detailed quadrature rules and in-

<sup>6</sup> This uniform sampling serves only to define integration regions for the ICRS. Our tests in various situations show  $N_{\text{limb}} \sim 500$  is sufficient for stability. Adaptive sampling could in principle improve robustness, but its sequential nature makes it unsuitable for GPU execution.

<sup>7</sup> Adaptive  $(r, \theta)$  grids were tested but found inefficient for GPU parallelization. With our integration scheme, the uniform grid provided a better balance of speed and accuracy.

terpolation formulae used in these calculations are presented in Appendix A.

### 3.4. Optimizing Parallel Computing and Memory Balance

High GPU performance requires both massive parallelism and efficient memory usage. In the ICRS method, which handles extended-source effects with fine image structures, balancing computational throughput with memory constraints is critical. To address this, `microJAX` leverages JAX’s functional primitives, particularly `vmap` and `lax.scan`, to express parallelism in a compiler-friendly and memory-efficient manner. The implementation distinguishes between operations that scale well with data-level parallelism and those limited by memory or control-flow dependencies. Key parallelized components include:

- (i) **Lens equation solving:** At each source-limb point, root-finding is fully vectorized with `vmap`, exploiting the embarrassingly parallel nature of the task without incurring significant memory overhead.
- (ii) **Inverse-ray shooting:** Image-plane grid points are mapped independently to the source plane. Within each sector (corresponding to a distinct image), azimuthal rays are evaluated in parallel via `vmap`, while integration across sectors is done sequentially with `lax.scan`, allowing intermediate results to be discarded and reducing memory usage. The dominant memory cost arises here, scaling with the product of radial and azimuthal grid sizes ( $N_r \times N_\theta$ ).
- (iii) **Light curve modeling:** For extended-source treatment, only a preselected set of  $N_{\text{ext}}$  epochs (e.g., near caustics) are modeled. These are divided into batches of size  $N_{\text{batch}}$ , processed in parallel via `vmap`, while outer iterations are handled by `lax.scan` or explicit loops.

As a result, the memory footprint scales as  $\mathcal{O}(N_r \times N_\theta \times N_{\text{batch}})$ . This memory-aware design allows modeling of thousands of time steps within realistic GPU limits. However, the use of `lax.scan`, which discards the intermediate state, precludes efficient reverse-mode automatic differentiation. Accordingly, `microJAX` currently supports only forward-mode differentiation for these components.

### 3.5. Automatic Differentiation

`microJAX` leverages JAX’s automatic differentiation (AD) framework to compute gradients of the magnification function for model parameters. However, certain model components involve non-differentiable operations, such as Heaviside functions and conditional branches, which obstruct standard differentiation. We therefore define differentiable exten-

sions of such functions with hand-written derivative rules via the `@custom_jvp` decorator in JAX<sup>8</sup>.

`microJAX` supports differentiable root-finding for complex polynomials via a JAX-compatible implementation of the Ehrlich-Aberth (EA) method. Consider the complex polynomial

$$\mathcal{P}(z; \mathbf{b}) = \sum_{k=0}^n b_k z^k, \quad \mathbf{b} = (b_0, \dots, b_n) \in \mathbb{C}^{n+1},$$

with complex variable  $z$ . Its roots  $\mathbf{z}^*(\mathbf{b}) = (z_1^*, \dots, z_n^*)$  are implicitly defined by  $\mathcal{P}(z_i^*; \mathbf{b}) = 0$ . To quantify how each root changes with the coefficients, differentiate the identity  $\mathcal{P}(z_i^*(\mathbf{b}); \mathbf{b}) = 0$  with respect to  $\mathbf{b}$ :

$$\frac{\partial \mathcal{P}}{\partial z}(z_i^*; \mathbf{b}) \frac{dz_i^*}{d\mathbf{b}} + \frac{\partial \mathcal{P}}{\partial \mathbf{b}}(z_i^*; \mathbf{b}) = 0.$$

Hence

$$\boxed{\frac{dz_i^*}{d\mathbf{b}} = - \frac{\partial \mathcal{P} / \partial \mathbf{b}(z_i^*; \mathbf{b})}{\partial \mathcal{P} / \partial z(z_i^*; \mathbf{b})}} \in \mathbb{C}^{1 \times (n+1)}.$$

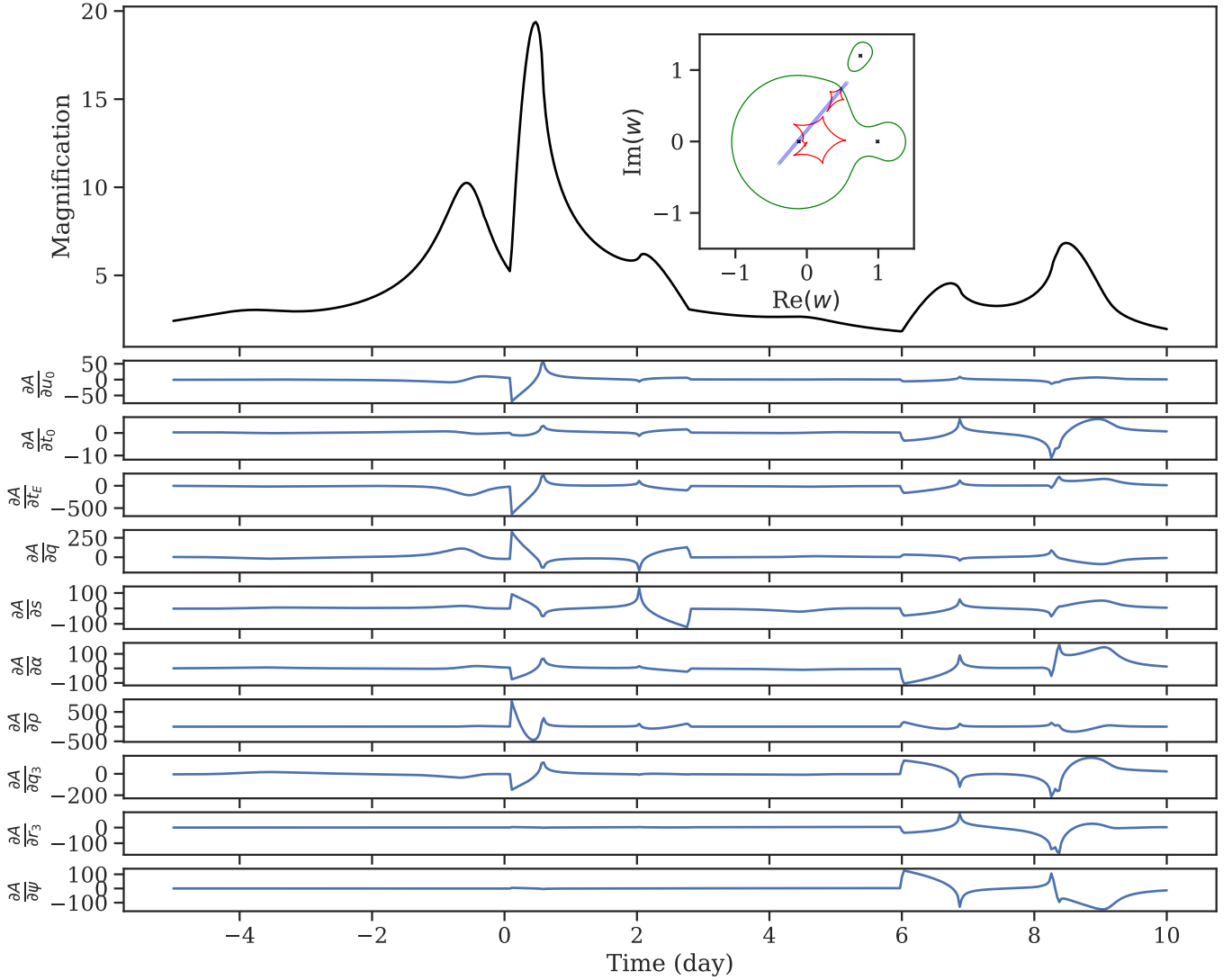
Here  $\partial \mathcal{P} / \partial \mathbf{b} = [(z_i^*)^n, (z_i^*)^{n-1}, \dots, 1]$  is the monomial vector evaluated at  $z_i^*$ , and  $\partial \mathcal{P} / \partial z = \sum_{k=1}^n k b_k z^{k-1}$  is the ordinary partial derivative. Stacking the gradients  $dz_i^* / d\mathbf{b}$  for all roots yields the Jacobian  $d\mathbf{z}^* / d\mathbf{b} \in \mathbb{C}^{n \times (n+1)}$ , which characterizes the full sensitivity of the root vector to the coefficients. This closed-form expression can be registered as a custom JVP rule, so gradients propagate through the roots without back-propagating through the iterative EA updates, yielding stable and efficient gradient computation in JAX.

Binary functions used for inclusion tests, such as determining whether a ray falls within the source radius, are implemented using the Heaviside function  $\Theta(x)$ . To enable smooth automatic differentiation, these functions are overridden by sigmoid-based approximations during gradient computation. In particular, the indicator function  $\chi = \Theta(\rho - d)$  used for segment classification is replaced in its JVP rule by a smooth sigmoid  $\sigma(x)$  with

$$\frac{\partial \chi}{\partial x} = \sigma(x)(1 - \sigma(x)) \times S_{\text{fac}}, \quad (9)$$

where  $S_{\text{fac}}$  is an empirically chosen steepness parameter controlling gradient sharpness and numerical stability. The correction factor  $f(\delta)$  detailed in Appendix A, used for integrating across azimuthal segments that straddle the source boundary in limb-darkened models, is defined piecewise depending on  $\delta_c$ . While the function itself retains its original

<sup>8</sup> JVP (Jacobian-Vector Product) is the forward-mode directional derivative; the reverse-mode counterpart is VJP (Vector-Jacobian Product), which JAX derives automatically from the same rule.



**Figure 2.** The top panel shows the triple-lens magnification curve for a uniform brightness source, with the inset displaying the corresponding caustic structure (red solid line) and source trajectory (blue solid line), where the microlensing parameters are  $(t_0, t_E, u_0, q, s, \alpha, q_3, r_3, \psi) = (0, 10 \text{ day}, 0, 0.5, 1.1, 60^\circ, 0.03, 1.24, 76.0^\circ)$ . The subsequent panels show the gradients of the magnification for each microlensing parameter. The resolutions of the inverse-ray grid are  $(N_r, N_\theta) = (500, 500)$ , respectively.

piecewise definition during primal evaluation, its gradient is manually defined as a fixed smooth expression, typically the derivative of the second-order formula, to ensure continuity in the AD flow near the threshold. This avoids discontinuities in the AD flow that would otherwise result from switching integration rules during evaluation.

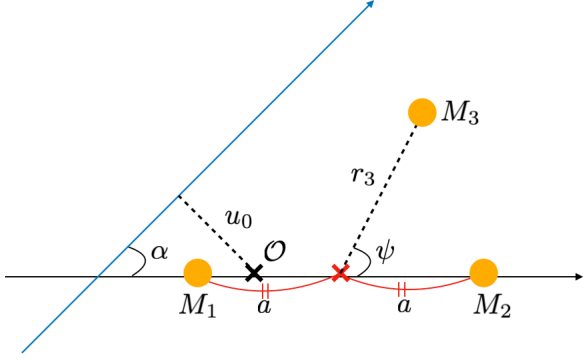
Figure 2 presents the magnification curve for a triple-lens system with a uniform-brightness source, along with the gradients of magnification for each model parameter. The computed gradients reveal regions of high sensitivity, particularly near caustic crossings, demonstrating `microJAX`'s ability to evaluate both magnification and its parameter derivatives in complex microlensing models. Here the microlensing model is parameterized by the time of closest approach to the barycenter of the first and second lenses  $t_0$ , the impact

parameter  $u_0$ , the Einstein radius crossing time  $t_E$ , the binary mass ratio  $q$ , the projected separation in units of the Einstein radius  $s$ , the angle between the source trajectory and the binary axis  $\alpha$ , and the normalized source radius  $\rho$ . For the third lens, three additional parameters are introduced:  $q_3$ , the mass ratio to the primary lens;  $r_3$ , the projected separation between the binary midpoint and the third lens; and  $\psi$ , the angle between the binary axis and the direction to the third lens. The geometric definitions are illustrated in Figure 3.

## 4. PERFORMANCE BENCHMARKS

### 4.1. Accuracy Validation

To assess the accuracy of `microJAX`, we benchmark its magnification calculations against the widely used `VBBinaryLensing` code (Bozza 2010; Bozza et al. 2018,



**Figure 3.** Geometrical configuration of the triple-lens system used in magnification and gradient calculations. The primary and secondary lenses ( $M_1$ ,  $M_2$ ) are separated by  $s \equiv 2a$  along the horizontal axis, with the origin  $\mathcal{O}$  defined as their barycenter. The source (blue line) moves with impact parameter  $u_0$  and angle  $\alpha$  relative to the binary axis. The third lens  $M_3$  is placed at polar coordinates  $(r_3, \psi)$  relative to the midpoint between  $M_1$  and  $M_2$ .

2021). All comparisons are made using version 3.7.0 of `VBBinaryLensing` to ensure consistency and reproducibility. Figure 4 shows the relative error of `microJAX` with respect to `VBBinaryLensing` for source radii  $\rho = (0.1, 0.01, 10^{-3}, 10^{-4})$  and azimuthal resolutions  $N_\theta = (500, 1000, 8000)$ . The radial resolution is fixed at  $N_r = 1000$ , a value that limits radial-integration errors to  $\lesssim 10^{-4}$ . A uniform-brightness source is assumed<sup>9</sup>, and the reference code is run at a nominal accuracy of  $10^{-5}$ . To construct the test dataset, source centers are randomly distributed around the central caustic of a binary lens with parameters  $q = 0.1$  and  $s = 1$ . Specifically, each test point is sampled within a circular region of radius  $\rho$  centered at a randomly selected point along the caustic curve. This ensures that all sources lie within a distance  $\rho$  from the caustic, effectively probing regions of strong magnification variation. The sampling configuration is illustrated in the lower-right panel of Figure 4. We fix the number of limb points to  $N_{\text{limb}} = 500$ , which is sufficiently high to ensure that limb discretization does not significantly affect the relative magnification error.

For large sources ( $\rho \gtrsim 10^{-2}$ ), all configurations achieve relative errors below  $10^{-4}$ . As the source size decreases near the caustic, the image boundary significantly elongates primarily in the angular direction while remaining relatively comparable in the radial direction. Consequently, accurately tracing that elongated boundary requires a finer azimuthal grid ( $N_\theta$ ), whereas the radial grid rarely misses the boundary. Figure 5 illustrates a case where a coarse angular grid determines an inaccurate image boundary compared to the source size, producing sizeable errors in magnification calcu-

lation. When the integrand in the radial direction is smooth, the quadrature error scales as  $N_r^{-2}$  and is averaged over the entire source area; thus, using coarse  $N_r$  only gradually degrades the precision. On the other hand, azimuthal sampling is far more sensitive: any boundary determination directly impacts the magnification, so insufficient  $N_\theta$  alone can introduce spikes and discontinuities in the light curve even if  $N_r$  is adequate. In the most demanding case ( $\rho = 10^{-4}$ ), only the highest resolution tested ( $N_\theta = 8000$ ) keeps the relative error below  $10^{-3}$ , underscoring the need for dense angular sampling in small-source regimes.<sup>10</sup> Although increasing  $N_\theta$  raises computational cost and memory per epoch, the total cost is offset in practice by the shorter source-crossing time  $t_\star = \rho t_E$ , which limits the number of epochs where extended-source effects must be evaluated. As a result, the effective computational burden remains manageable even at high angular resolution.

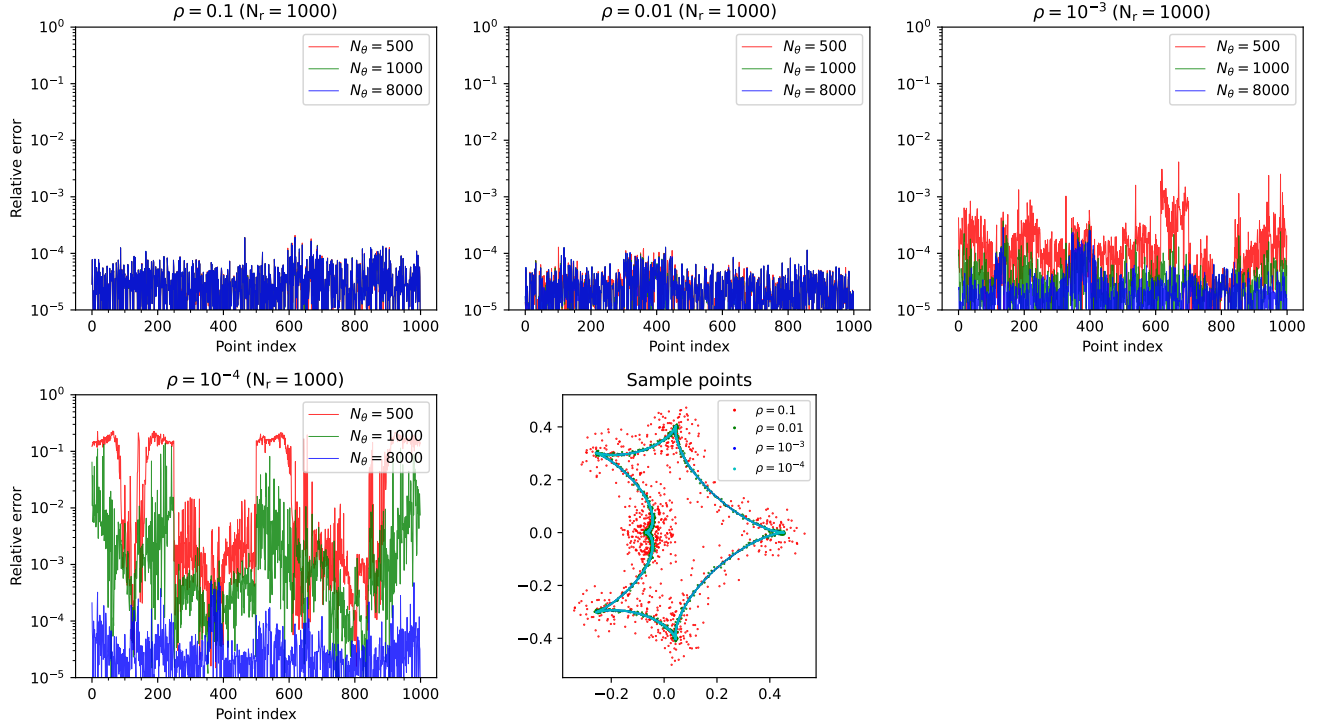
#### 4.2. Computational Performance

Figure 6 compares the total computation time per magnification evaluation as a function of the number of light curve points (epochs), for both `VBBinaryLensing` and `microJAX`, under uniform and limb-darkened source assumptions. Both methods exhibit good scalability with the number of epochs, but the computation time for each point reflects the fundamental differences in how magnification is evaluated. `VBBinaryLensing` employs a contour integration method that is highly optimized for uniform sources, achieving performance up to  $\sim 10\times$  faster than in the limb-darkened case, where brightness weighting and adaptive sampling introduce significant computational overhead. In contrast, `microJAX` shows minimal difference (less than a factor of 2) between uniform and limb-darkened cases. This efficiency arises from its use of a static computation graph, which allows all operations to be compiled efficiently ahead of time. Consequently, the total runtime is nearly independent of the source size  $\rho$ . While `microJAX` can be up to  $\sim 4\text{--}5\times$  slower than `VBBinaryLensing` for large, uniform sources, it significantly outperforms in the small-source, limb-darkened regime, achieving  $5\text{--}6\times$  speedup due to its stable performance profile.

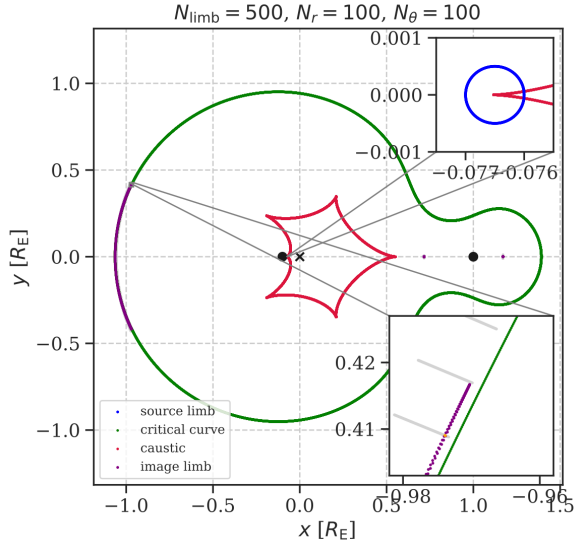
It should be noted, however, that a direct comparison between `VBBinaryLensing` (running on CPU) and `microJAX` (executed on GPU) is inherently limited by differences in hardware architecture and parallelization strategies. The reported speed differences should therefore be interpreted as relative performance under typical usage scenarios, rather than absolute benchmarks.

<sup>9</sup> We also tested limb-darkened sources and confirmed that the conclusions remain consistent.

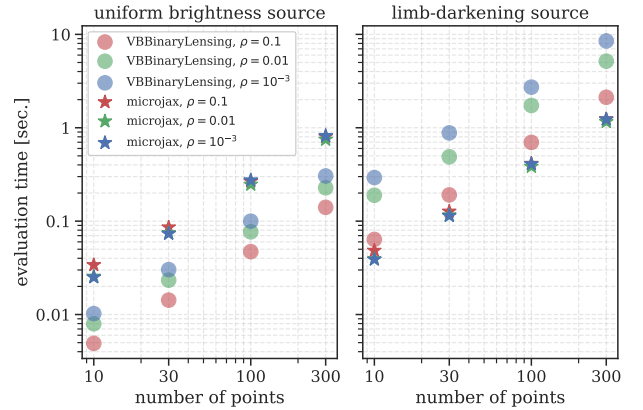
<sup>10</sup> The total angular extent of a lensed image is at most of order  $\pi$ . Therefore, to achieve an angular resolution comparable to the source radius  $\rho$ , the required number of azimuthal grid points is of order  $\sim 1/(\pi\rho)$ .



**Figure 4.** Relative accuracy of `microJAX` magnification estimates compared to `VBinaryLensing`, as a function of source size ( $\rho = 0.1$  to  $10^{-4}$ ) and angular resolution ( $N_\theta = 500, 1000, 8000$ ) with fixed  $N_r = 1000$ . Each panel shows the relative error in magnification computed by `microJAX`, to the reference values from `VBinaryLensing`, for 1000 randomly sampled source positions near the central caustic of a binary lens with  $q = 0.1$  and  $s = 1$ . A uniform-brightness source is assumed, and `VBinaryLensing` is run at a relative accuracy of  $10^{-5}$ . The lower-right panel illustrates the spatial distribution of test points near the caustic.



**Figure 5.** An example of a lensing geometry, similar with Figure 1, where the error in magnification increases, with parameters  $(q, s, \rho) = (0.1, 1.1, 5 \times 10^{-4})$  and  $\mathbf{w}_{\text{center}} = (-0.0765, 0)$ . Each inset panel shows a zoomed-in view of each position. In this case, the angular resolution of the grid is too sparse to accurately determine the image boundary compared to the source size, illustrated in the lower-right panel.



**Figure 6.** Total computation time per magnification evaluation is shown as a function of the number of light curve points (epochs) for both uniform (left) and limb-darkened (right) source profiles. The results compare `microJAX` (stars) and `VBinaryLensing` (circles) across different source sizes  $\rho$ . We set the relative error in `VBinaryLensing` to  $10^{-4}$ , and adopt  $(N_r, N_\theta) = (10^3, 10^3)$  in `microJAX`. All benchmarks were performed using an AMD EPYC 7452 CPU (2.35 GHz base frequency, single-threaded) for `VBinaryLensing`, and an NVIDIA A100 PCIe GPU (40 GB) for `microJAX`. All timings were measured with JIT compilation enabled and exclude compilation overhead.

**Table 1.** Public Codes of Microlensing Magnification Calculation

| Codes           | Method <sup>*</sup> | Max $N_{\text{lens}}$ | GPU | Differentiable |
|-----------------|---------------------|-----------------------|-----|----------------|
| microJAX        | ICRS                | 3                     | ✓   | ✓              |
| eesunhong       | ICRS                | 2                     | ✗   | ✗              |
| VBBinaryLensing | CI                  | 2                     | ✗   | ✗              |
| VBMicrolensing  | CI                  | $\geq 3$              | ✗   | ✗              |
| Twinkle         | CI                  | 2                     | ✓   | ✗              |
| caustics        | CI                  | 3                     | ▲   | ✓              |
| microlux        | CI                  | 2                     | ▲   | ✓              |

<sup>\*</sup> ICRS = Image-centred Ray Shooting method. CI = Contour Integrating method.

NOTE—✓= Supported; ✗= Not supported; ▲= JAX backend available, but no dedicated GPU optimization. All listed codes are designed to compute microlensing magnification as a function of source position and lens configuration. microJAX<sup>a</sup>: This work. eesunhong<sup>b</sup>: CPU-optimized ICRS (Bennett & Rhie 1996; Bennett 2010). VBBinaryLensing<sup>c</sup>: CPU-optimized contour integrating for binary-lens (Bozza 2010; Bozza et al. 2018). VBmicrolensing<sup>d</sup>: CPU-optimized contour integrating for multiple-lens (Bozza et al. 2025). Twinkle<sup>e</sup>: GPU-optimized contour integrating for binary-lens (Wang et al. 2025). caustics<sup>f</sup>: differentiable contour integrating (Bartolić 2023). microlux<sup>g</sup>: differentiable VBBinaryLensing (Ren & Zhu 2025).

<sup>a</sup> <https://github.com/ShotaMiyazaki94/microjax>

<sup>b</sup> <https://github.com/golmschenk/eesunhong>

<sup>c</sup> <https://github.com/valboz/VBBinaryLensing>

<sup>d</sup> <https://github.com/valboz/VBMicrolensing>

<sup>e</sup> <https://github.com/AsterLight0626/Twinkle>

<sup>f</sup> <https://github.com/fbartolic/caustics>

<sup>g</sup> <https://github.com/CoastEgo/microlux>

Overall, these results demonstrate that microJAX offers competitive performance and superior scaling for complex source models, particularly in the small-source, limb-darkened regime most relevant for high-precision microlensing analyses.

#### 4.3. Comparison with Other Microlensing Codes

Table 1 summarizes key features of publicly available codes for computing microlensing magnification, including their numerical methods, GPU support, and differentiability. Among them, microJAX uniquely combines differentiability, GPU acceleration, and support for triple-lens configurations, representing capabilities not simultaneously realized in other tools. Unlike other well-established codes, such as eesunhong (Bennett & Rhie 1996; Bennett 2010) and VBBinaryLensing (Bozza 2010; Bozza et al. 2018), which are non-differentiable and thus unsuitable for gradient-based inference, microJAX provides a fully differentiable framework tailored for modern probabilistic modeling. Al-

**Table 2.** Injection-Recovery Test Setting and Results

| Parameters             | Injected | Recovered<br>(median & 68% CI) |
|------------------------|----------|--------------------------------|
| $t_0$                  | 0        | $0.001^{+0.001}_{-0.002}$      |
| $\log(t_E/\text{day})$ | 1.21     | $1.209^{+0.004}_{-0.004}$      |
| $u_0$                  | 0.06     | $0.060^{+0.001}_{-0.001}$      |
| $\log q$               | -5.0     | $-5.04^{+0.05}_{-0.04}$        |
| $\log s$               | 0.012    | $0.012^{+0.001}_{-0.002}$      |
| $\alpha$               | 4.156    | $4.153^{+0.002}_{-0.002}$      |
| $\log \rho$            | -2.74    | $-2.74^{+0.01}_{-0.02}$        |

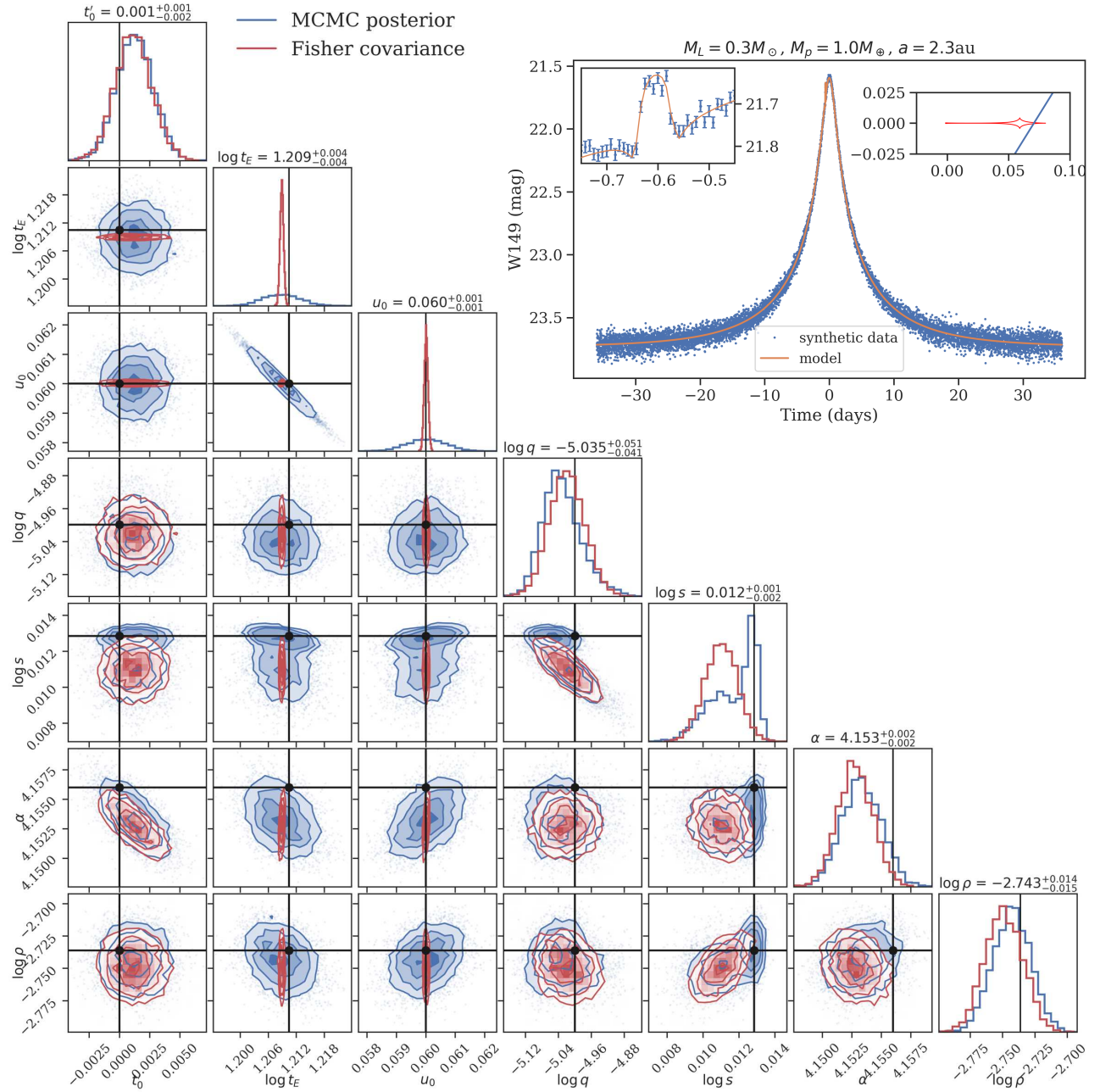
NOTE—Posterior sampling is performed using HMC with 500 warm-up steps and 5,000 retained samples, executed on an NVIDIA A100 GPU. The full run required  $\sim 19$  hours, corresponding to an average of 12.5 second per retained sample. Each proposal was generated with 15 leapfrog steps, and the mean acceptance probability was  $\geq 90\%$ .

though differentiable alternatives such as caustics (Bartolić 2023) and microlux (Ren & Zhu 2025) exist, they are not optimized for complex source profiles. To address this gap, microJAX implements a GPU-accelerated, batched ICRS algorithm that maintains sub-pixel accuracy while supporting arbitrary source profiles. This is not merely a technical optimization but a critical enabler: high-dimensional Bayesian inference demands flexible, physically accurate models. microJAX preserves the differentiability of such models without compromising computational tractability, ensuring coherence between model expressiveness and inference efficiency.

CPU- and GPU-based frameworks may play complementary roles in microlensing inference workflows. CPU-oriented tools are well-suited for large-scale exploratory analyses, such as grid-based searches over a wide range of initial conditions using CPU server clusters, serving as effective guides toward high-precision solutions. In particular, approximate posterior modes identified using CPU-based auto-differentiable frameworks may inform subsequent refinement using microJAX, which is optimized for GPU-accelerated modeling in the final stages of parameter inference. This potential complementarity suggests the importance of continued development of both types of tools to enable scalable and flexible analysis pipelines tailored to different phases of model evaluation.

## 5. APPLICATIONS

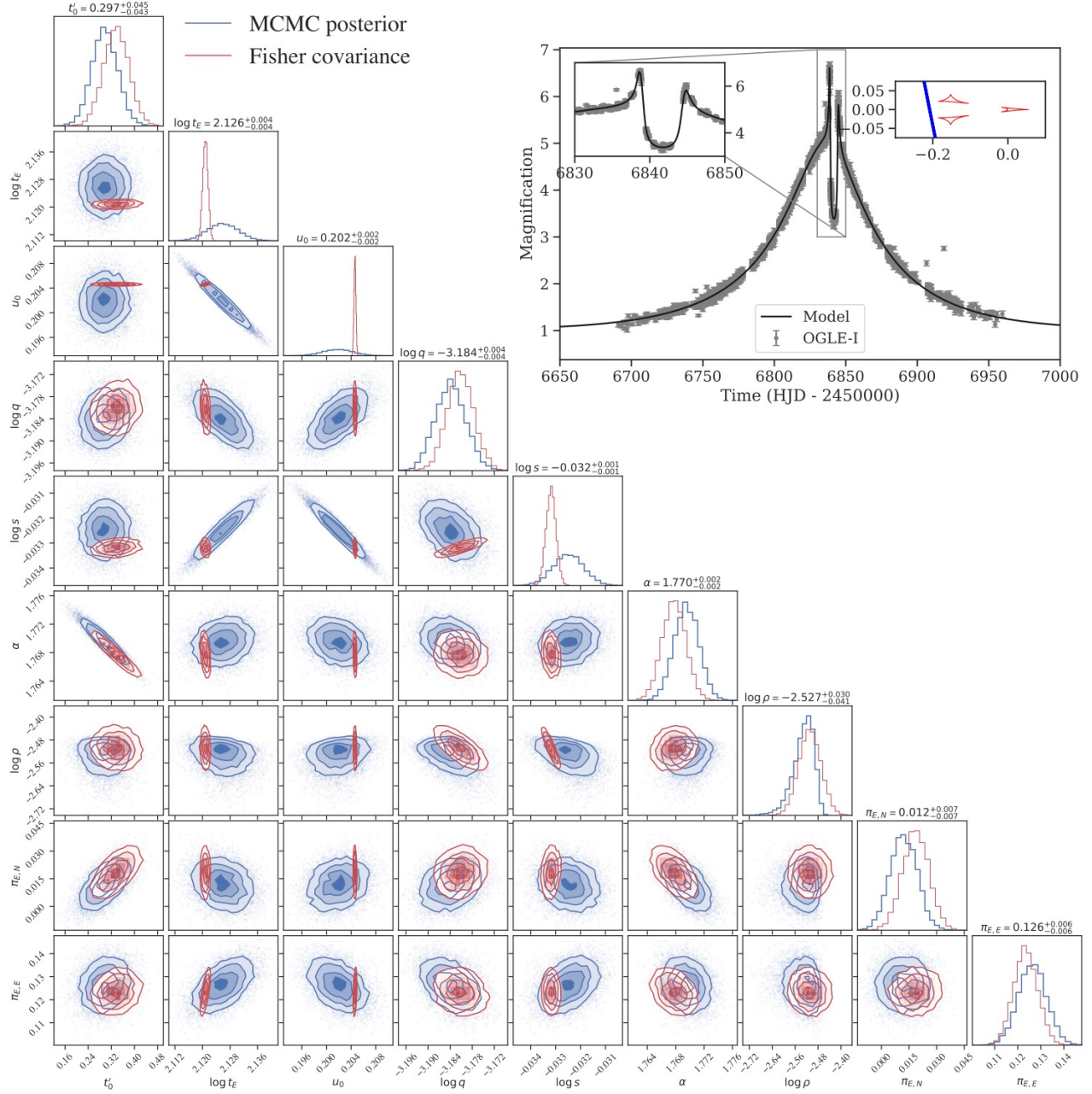
This section demonstrates the practical capabilities of microJAX through two complementary experiments. First,



**Figure 7.** The top right panel shows the synthetic light curve observed by the Roman, where blue error bars are W149-band measurements and the orange solid line is the model curve. The top-right inset in the same panel displays the caustic geometry (red solid line) and the source trajectory (blue line). The corner plot shows the posterior distributions of the model parameters (blue) along with the Fisher covariance contours (red) evaluated at the best-fit parameters obtained with gradient-based optimization before inference. The ground truths (the injected parameters) are shown as black cross-hatched points.

we conduct a parameter recovery test using a synthetic Roman-like light curve with known ground-truth parameters. Second, we apply the method to the well-studied planetary microlensing event OGLE-2014-BLG-0124, utilizing real and heterogeneous photometric data. Together, these experiments illustrate the statistical reliability and robustness of *microJAX* under both idealized and realistic observational conditions. Prior to posterior sampling, we perform gradient-

based optimization using the *optax* library (DeepMind et al. 2020) to obtain a good initial estimate. Throughout the analysis, we employ posterior sampling via Hamiltonian Monte Carlo (HMC) with the No-U-Turn Sampler (NUTS; Duane et al. 1987; Betancourt 2017) as implemented in NumPyro (Phan et al. 2019; Bingham et al. 2019).



**Figure 8.** The top right panel shows the observed light curve of OGLE-2014-BLG-0124. Gray error bars indicate the  $I$ -band photometry from the OGLE-IV survey, and the black solid line denotes the best-fit binary-lens model including orbital parallax. The inset in the same panel displays the caustic geometry (red solid line) and the source trajectory (blue line). The corner plot shows the posterior distributions of the model parameters (blue) along with the Fisher covariance contours (red), evaluated at the best-fit solution obtained with gradient-based optimization before inference.

### 5.1. Parameter Recovery Test with a Synthetic Roman-like Event

To assess inference accuracy under controlled conditions, we generate a 72-day synthetic light curve sampled at a 12-minute cadence, adopting fiducial microlensing parameters. Gaussian photometric uncertainties are independently assigned based on the  $W149$ -band magnitude-uncertainty re-

lation presented in Figure 4 of Penny et al. (2019). The light curve includes a planetary perturbation induced by a lens system in which an Earth-mass planet orbits an M-dwarf host. The injected microlensing parameters are set as  $\phi \equiv (t_0, \log(t_E/\text{day}), u_0, \log q, \log s, \alpha, \log \rho) = (0., 1.21, 0.06, -5.0, 0.012, 4.156, -2.74)$ .

Instead of placing priors directly on the physical parameters, we reparameterize the model using a latent standard

normal variable. Specifically, we draw a latent vector  $\beta \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$  and apply an element-wise affine transformation to obtain the physical parameters:  $\phi = \mu + \sigma \odot \beta$ , where  $\mu$  is the maximum a posteriori (MAP) estimate obtained via gradient-based optimization, and  $\sigma$  is a vector of scale parameters, given by the square roots of the diagonal elements of the Fisher covariance matrix evaluated at  $\mu$  and inflated by a factor of 30. The symbol  $\odot$  denotes the Hadamard (element-wise) product. This approach corresponds to an independent Gaussian prior in the physical space:

$$\phi_i \sim \mathcal{N}(\phi_{i,\text{init}}, 30 \cdot \sigma_{i,\text{Fisher}}), \quad (10)$$

where  $\phi_{i,\text{init}}$  is the MAP estimate, and  $\sigma_{i,\text{Fisher}}$  is the corresponding diagonal element of the Fisher covariance matrix evaluated at that point. While the full Fisher matrix encodes local parameter correlations, we deliberately ignore off-diagonal terms to avoid injecting potentially unreliable structure into the prior. The diagonal approximation with variance inflation provides a conservative, weakly informative prior that respects the local scale of each parameter without overcommitting to possibly inaccurate correlation directions. In other words, this is equivalent to independent Gaussian priors centered on the MAP values with inflated Fisher variances. The reparameterization with latent standard normal variables is introduced only for computational convenience, ensuring stable gradient-based inference.

The microlensing light curve is modeled as

$$F_{\text{obs}}(t) = [A(t; \phi) - 1] F_S + F_{\text{base}}, \quad (11)$$

where  $A(t; \phi)$  is the model magnification as a function of time and microlensing parameters, and  $F_S$  and  $F_{\text{base}}$  denote the source and baseline fluxes, respectively. Given that the photometric uncertainty of the light curve reaches  $\sigma_{W149} \sim 0.01$ , we adopt integration resolutions of  $(N_r, N_\theta) = (10^3, 10^3)$ , which conservatively ensures an accuracy better than  $10^{-4}$ . Since the flux parameters enter linearly, they are analytically marginalized in the likelihood evaluation, as described in Appendix B. Posterior sampling is performed using HMC with 500 warm-up steps and 5,000 retained samples, executed on an NVIDIA A100 GPU. Table 2 shows the settings and results of the injection-recovery test.<sup>11</sup>

Figure 7 presents the synthetic Roman-like light curve (top right panel) alongside a corner plot summarizing the HMC posterior. Splitting the single HMC chain yields Gelman-Rubin diagnostics (Gelman & Rubin 1992) of  $\hat{R} < 1.00$  and effective sample sizes exceeding 800 for all parameters, indicating convergence well. In the corner plot, red ellipses represent a Gaussian approximation based on the Fisher matrix

evaluated at the best-fit point. These ellipses agree with the posterior for well-constrained parameters such as the  $(t_0, \alpha)$  plane. Although all injected values (black cross-hairs) are recovered within the 90% highest-density regions of the posterior, some Fisher ellipses entirely fail to enclose them. This discrepancy is particularly evident in poorly constrained or degenerate directions, such as  $(\log t_E, u_0)$  and  $(\log q, \log s)$ , where the posterior exhibits strong skewness or curvature. In such cases, the Fisher approximation not only underestimates the uncertainty but also misrepresents the correlation structure, with both scale and orientation deviating significantly from the true posterior geometry. By contrast, the HMC-based posterior fully captures these non-Gaussian features, accurately recovering the injection values even in the presence of pronounced degeneracies. This highlights the robustness of `microJAX` with HMC in resolving complex parameter structures beyond the reach of local Gaussian approximations.

## 5.2. Real Planetary Event: OGLE-2014-BLG-0124

We further demonstrate the applicability of `microJAX` to real data by analyzing the well-characterized planetary microlensing event OGLE-2014-BLG-0124 (Udalski et al. 2015b). We use only the *I*-band photometry from the OGLE-IV survey (Udalski et al. 2015a), retrieved from the NASA Exoplanet Archive (Christiansen et al. 2025), and fit the light curve using a binary-lens model that includes the orbital parallax effect. The model consists of nine parameters, defined as  $\phi \equiv (t_0, \log(t_E/\text{day}), u_0, \log q, \log s, \alpha, \log \rho, \pi_{E,N}, \pi_{E,E})$ , where  $\pi_{E,N}$  and  $\pi_{E,E}$  are the north and east components of the microlens parallax vector (Gould 2004), respectively. This event has degenerate solutions of  $u_0 > 0$  and  $u_0 < 0$ , but here we focus on the  $u_0 > 0$  solution. During HMC, we also infer a flux error rescaling factor  $f_{\text{err}}$ , defined such that the rescaled photometric uncertainties are  $\sigma'_i = f_{\text{err}} \sigma_i$  where  $\sigma'_i$  and  $\sigma_i$  are the rescaled and original uncertainties, respectively. Posterior sampling is performed using HMC with 500 warm-up steps and 10,000 retained samples in a single chain. The overall inference procedure follows the same framework as applied in the synthetic test case.

Figure 8 shows the observed light curve (upper right panel) and the resulting corner plot from parameter inference using HMC. The posterior of the error rescaling factor is  $f_{\text{err}} = 1.711^{+0.013}_{-0.013}$  (68% credible interval), and, for all the parameters,  $\hat{R} < 1.00$  and the ESS exceeds 5,000. Since our light curve and model<sup>12</sup> differ from those of Udalski et al. (2015b), we do not attempt a direct comparison of the inferred parameters. Nevertheless, our results are qualitatively consistent

<sup>11</sup> For reproducibility, the code used to perform this test is available at [https://github.com/ShotaMiyazaki94/test\\_microjax](https://github.com/ShotaMiyazaki94/test_microjax).

<sup>12</sup> For example, our model does not account for lens orbital motion.

with the original study, recovering a comparable planetary signal and clear parallax detection. This supports the reliability of our framework under realistic observational conditions. As in the synthetic test, the Fisher matrix does not fully capture the posterior geometry for several parameters in the real-data analysis. These discrepancies imply the limitations of the Fisher-based prior with correlation.

### 5.3. Caveats and Lessons Learned

Our experiments confirm that HMC enables robust and efficient posterior sampling, even in high-dimensional microlensing models. However, several caveats must be noted. The Fisher information matrix, while useful for setting initial parameter scales, often fails to capture the true posterior geometry, particularly in the presence of strong degeneracies, skewness, or multimodal structure. The resulting Gaussian approximation typically underestimates uncertainties and distorts parameter correlations, leading to error ellipses that may misalign with the true posterior and fail to enclose the ground-truth parameters. When reparameterization is performed using the Fisher matrix, these inaccuracies propagate into the prior. Because HMC operates in the latent space defined by this transformation, misalignment between the Fisher-based prior and the posterior can suppress sampling efficiency and hinder exploration of relevant modes. We observe these effects even under idealized conditions, suggesting that full-covariance Fisher priors (e.g., Ren & Zhu 2025) are only valid when the posterior is approximately Gaussian.

These issues are especially pronounced in microlensing, where model degeneracies and parameter correlations are not rare. Even simplified assumptions, such as diagonal priors, may also risk overlooking critical posterior structure. While Fisher-based priors can serve as a practical and scalable starting point, we advocate for flexible prior models and rigorous posterior diagnostics, especially in complex or weakly constrained inference problems.

## 6. DISCUSSION AND CONCLUSION

We have presented `microJAX`, a differentiable microlensing modeling framework that combines GPU-accelerated, image-centered inverse-ray shooting with JAX-based automatic differentiation. Designed for scalability and physical fidelity, `microJAX` supports gradient-based inference in extended-source and multiple-lens models. Benchmarks against `VBBinaryLensing` show that `microJAX` achieves sub-second times of extended-source magnification evaluation for hundreds of epochs on a single GPU while maintaining relative magnification error less than  $10^{-4}$ , making it suitable for precise and high-throughput applications. Case studies further demonstrate its ability to recover model parameters efficiently under both idealized and observationally realistic conditions.

Despite its current capabilities, several key extensions remain for future work. The framework currently relies on forward-mode automatic differentiation, which scales linearly with the number of parameters and becomes costly in high-dimensional models. Reverse-mode differentiation, which offers constant-cost gradient evaluation with respect to parameter dimensionality (Blondel & Roulet 2024), is not yet implemented due to memory constraints: storing intermediate values from the inverse-ray shooting process exceeds typical GPU memory limits. We plan to address this through checkpointing strategies (Griewank 1992), enabling memory-efficient reverse-mode differentiation and significantly faster gradient computation.

Support for higher-order microlensing effects, such as lens orbital motion (Dominik 1998b; Bennett et al. 2010; Skowron et al. 2011), and xallarap (Bennett et al. 2008; Miyazaki et al. 2020, 2021; Rota et al. 2021), as well as an increased number of lens masses (Bozza et al. 2025), is also a priority. Furthermore, high-resolution imaging surveys such as Roman will enable the detection of astrometric microlensing signals, offering complementary constraints on lens properties (Gould & Yee 2014; Wyrzykowski et al. 2020; Lam et al. 2022; Sahu et al. 2022). Incorporating such astrometric data into the modeling framework will be essential for achieving high-precision, fully-consistent inferences. These extensions, while critical, significantly increase model complexity and parameter dimensionality. As such, automatic differentiation will remain central to enabling scalable inference.

An additional direction for future development involves the use of more realistic noise models. The assumption of independent Gaussian errors breaks down in many practical cases, particularly in precise space-based observations such as those from *Kepler* (e.g., Carter & Winn 2009; Stumpe et al. 2012). Incorporating time-correlated and heteroscedastic uncertainties will be critical for robust parameter inference. The growing ecosystem of JAX-based probabilistic tools, including differentiable Gaussian processes and state-space models (e.g., Pinder & Dodd 2022; Foreman-Mackey et al. 2024; Boettner 2025), provides a promising foundation for integrating physical modeling and noise characterization within a unified, end-to-end framework.

In summary, `microJAX` offers a modular and differentiable framework for physically interpretable microlensing inference. With continued development, it has the potential to play a central role in analyzing next-generation microlensing datasets and advancing our understanding of planetary populations.

## ACKNOWLEDGMENTS

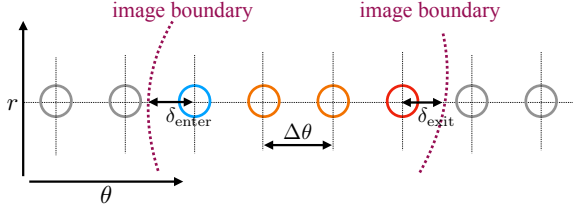
We thank the anonymous referee for constructive feedback that helped improve the clarity of the paper. We are

also grateful to Kohei Miyakawa, Yui Kasagi, Shotaro Tada, and Yamato Ureshino for productive discussions on differentiable programming. We also thank Kento Masuda and Ryou Ohsawa for valuable comments on this manuscript. This study was supported by the Japan Society for the Promotion of Science (JSPS) KAKENHI Grant Numbers 23K19079, 25KJ0428, and 25K17449. We acknowledge the use of Chat-

GPT (OpenAI) in improving the grammar and clarity of the manuscript.

*Software:* `microJAX` (Miyazaki & Kawahara 2025), `JAX` (Bradbury et al. 2018), `Numpyro` (Bingham et al. 2018; Phan et al. 2019), `VBinaryLensing` (Bozza 2010; Bozza et al. 2018), `corner` (Foreman-Mackey 2016)

## APPENDIX



**Figure 9.** A schematic view of the integration method for a given  $r$ -row. The circles represent the grid points assigned for the inverse-ray shooting, and the purple lines represent the image boundary. The orange grid points are fully inside the image, the blue (entering) and red (exiting) ones are partially inside, and the gray ones are outside ( $d > \rho$ ). The boundary position of the image at each row, i.e.,  $d = \rho$ , is estimated using cubic interpolation based on the distances of four nearest points to the boundary.

### A. INTEGRATION RULES FOR SURFACE BRIGHTNESS PROFILES

This appendix provides the full numerical formulation for computing the magnification in `microJAX`, using the inverse-ray shooting method on a fixed-resolution polar grid. Given the annular image regions defined in Section 3.3, we discretize each image into  $N_r \times N_\theta$  grid points and compute their inverse-ray mapped positions in the source plane. The magnification is then obtained by numerically integrating the source brightness over these mapped positions, with quadrature rules adapted to the source profile.

#### A.1. For Uniform Brightness Source Profile

We describe here the method used to compute the magnification when the source has a uniform surface brightness profile. To accurately evaluate the contribution of each azimuthal segment, the algorithm examines local variations in the distance to the source center and uses sub-grid interpolation to resolve image boundaries with high precision. For each fixed radius  $r_i$ , we consider four consecutive azimuthal points,  $(\theta_{\ell-1}, \theta_\ell, \theta_{\ell+1}, \theta_{\ell+2})$ , along with their corresponding distances from the source center,  $(d_{\ell-1}, d_\ell, d_{\ell+1}, d_{\ell+2})$ . These values are used to construct a cubic Lagrange interpolant  $d(\theta)$ , which approximates the distance profile along the azimuthal direction. The boundary crossing angle  $\theta_{\text{cross}}$  is then estimated by solving  $d(\theta_{\text{cross}}) = \rho$ , enabling precise

identification of the image boundary. This interpolation enables accurate integration across segments that partially overlap the source disk, thereby reducing aliasing artifacts and enhancing the fidelity of the computed magnification. The use of cubic interpolation strikes a balance between numerical accuracy and computational efficiency, and is particularly effective for resolving sharp transitions near image edges.

Figure 9 represents a schematic view of the grid allocation for a given  $r$ -row. To determine the contribution of each azimuthal segment, we classify adjacent angular pairs  $(\theta_\ell, \theta_{\ell+1})$  based on whether their corresponding rays map inside or outside the source boundary. For each fixed radius  $r_i$ , we define an inclusion indicator

$$\chi_\ell = \Theta(\rho - d_\ell), \quad (\text{A1})$$

where  $d_\ell = |\mathbf{w}_{i,\ell} - \mathbf{w}_c|$ , and  $\Theta$  is the Heaviside step function. Each segment is then categorized as follows:

- **Fully enclosed** ( $\chi_\ell = \chi_{\ell+1} = 1$ ): Both endpoints lie inside the source, and the segment contributes fully to the integral (orange-orange segments in Figure 9).
- **Entering** ( $\chi_\ell = 0, \chi_{\ell+1} = 1$ ): The segment crosses into the source; the contribution is determined via interpolation at the entry point (gray-blue segments).
- **Exiting** ( $\chi_\ell = 1, \chi_{\ell+1} = 0$ ): The segment crosses out of the source, treated analogously (red-gray segments).

For crossing segments, we estimate the boundary angle  $\theta_{\text{cross}}$  using cubic interpolation based on the distances of nearby points. The fractional coverage is then given by

$$\delta_{\ell,\text{enter}} = \frac{\theta_{\ell+1} - \theta_{\text{cross}}}{\Delta\theta}, \quad \delta_{\ell,\text{exit}} = \frac{\theta_{\text{cross}} - \theta_\ell}{\Delta\theta}, \quad (\text{A2})$$

where  $\Delta\theta$  is the azimuthal grid spacing. The total contribution from radius  $r_i$  is computed by summing over azimuthal segments using a midpoint rule:

$$A_i = r_i \Delta\theta \sum_\ell \left[ \chi_\ell \chi_{\ell+1} + (1 - \chi_\ell) \chi_{\ell+1} \left( \delta_{\ell,\text{enter}} + \frac{1}{2} \right) + \chi_\ell (1 - \chi_{\ell+1}) \left( \delta_{\ell,\text{exit}} + \frac{1}{2} \right) \right]. \quad (\text{A3})$$

Finally, the contributions from all radial rings are integrated using the trapezoidal rule to yield the area of the sec-

tor:

$$A_{\text{sector}} = \sum_{i=0}^{N_r-1} A_i \Delta r, \quad (\text{A4})$$

and the total magnification is given by summing over all sectors and normalizing by the area of the source disk:

$$A = \frac{1}{\pi \rho^2} \sum_{\text{sectors}} A_{\text{sector}}. \quad (\text{A5})$$

### A.2. For Limb-darkened Source Profile

We now extend the integration scheme to sources with a limb-darkened surface brightness profile. In this case, the magnification is computed by weighting each area contribution by the position-dependent source brightness  $I(\tilde{r})$ , where  $\tilde{r} = d/\rho$  is the normalized radial distance from the source center. For the linear (first-order) limb-darkening law, the brightness is given by

$$I(\tilde{r}) = I_0 \left[ 1 - u_1 \left( 1 - \sqrt{1 - \tilde{r}^2} \right) \right], \quad (\text{A6})$$

where  $u_1$  is the limb-darkening coefficient, and  $I_0 = 3/(\pi(3 - u_1))$  is a normalization factor that ensures the total flux over the source integrates to unity. The profile is defined for  $0 \leq \tilde{r} \leq 1$ .

As in the uniform-brightness case, each radial ring is divided into azimuthal segments. For each fixed radius  $r_i$ , we consider five consecutive angular grid points  $(\theta_{\ell-2}, \theta_{\ell-1}, \theta_{\ell}, \theta_{\ell+1}, \theta_{\ell+2})$  and their corresponding distances  $(d_{\ell-2}, d_{\ell-1}, d_{\ell}, d_{\ell+1}, d_{\ell+2})$  to the source center. Based on the inclusion flags  $\chi_{\ell-1}, \chi_{\ell}, \chi_{\ell+1}$ , each segment centered at  $\theta_{\ell}$  (spanning  $\theta_{\ell} \pm \Delta\theta/2$ ) is classified as follows:

- **Fully enclosed** ( $\chi_{\ell-1} = \chi_{\ell} = \chi_{\ell+1} = 1$ ): The segment lies entirely within the source and contributes fully to the integral, weighted by  $I(\tilde{r}_{\ell})$  at the midpoint.
- **Entering** ( $\chi_{\ell-1} = 0, \chi_{\ell} = \chi_{\ell+1} = 1$ ): The image boundary lies between  $\theta_{\ell-1}$  and  $\theta_{\ell}$ . The crossing angle  $\theta_{\text{cross}}$ , satisfying  $d(\theta_{\text{cross}}) = \rho$ , is estimated via cubic interpolation using four of the five neighboring points. The fractional contribution is  $\delta_{\ell, \text{enter}} = (\theta_{\ell+1} - \theta_{\text{cross}})/\Delta\theta$ .
- **Exiting** ( $\chi_{\ell-1} = \chi_{\ell} = 1, \chi_{\ell+1} = 0$ ): The image boundary lies between  $\theta_{\ell}$  and  $\theta_{\ell+1}$ . The crossing angle  $\theta_{\text{cross}}$  is estimated as above, and the fractional contribution is  $\delta_{\ell, \text{exit}} = (\theta_{\text{cross}} - \theta_{\ell})/\Delta\theta$ .

A naive application of the trapezoidal rule to boundary-crossing segments leads to significant numerical errors, particularly near the limb of the source where the surface brightness gradient becomes steep. For the first-order limb-darkening law, the surface brightness in the image plane near the edge behaves approximately as  $f(z) \sim \sqrt{z - z_L}$ , where  $z_L$  denotes a point on the image boundary. As revealed by a Taylor expansion, applying the trapezoidal rule

to such functions introduces a leading-order integration error of  $\mathcal{O}(\Delta\theta^{3/2})$ , due to the divergence of the second derivative near the edge:  $f''(z) \propto (z - z_L)^{-3/2}$ . This singular behavior invalidates the assumptions underlying the trapezoidal approximation, necessitating the use of more suitable integration rules tailored to such edge singularities.

To reduce integration error while preserving the use of a fixed polar grid, we adopt a hybrid quadrature scheme inspired by [Bennett \(2010\)](#). This method modifies the standard midpoint rule for segments that are only partially enclosed within the source disk. Specifically, each segment's contribution is scaled by a correction factor  $f(\delta)$ , which depends on the fractional coverage  $\delta$  of the segment and is defined piecewise as:

$$f(\delta) = \begin{cases} \left(\frac{1}{2} + \delta\right) b(\delta), & \text{if } \delta \geq \delta_c \\ \frac{2}{3}\delta + \frac{1}{2}, & \text{if } \delta < \delta_c, \end{cases} \quad (\text{A7})$$

where  $\delta_c$  is a tunable threshold chosen based on the empirical results of [Bennett \(2010\)](#). This formulation offers a significant improvement in integration accuracy over the naive trapezoidal rule and also over the limiting cases  $\delta_c = 0$  or 1. The correction function

$$b(\delta) = \frac{2}{3} \sqrt{1 + \frac{1}{2\delta}} \quad (\text{A8})$$

is derived to approximate the integral of a square-root brightness profile near the limb of the source. For small  $\delta$ , the second branch of  $f(\delta)$  ensures numerical stability and yields a second-order accurate approximation (see [Bennett \(2010\)](#) for details). Using this scheme, the area contribution for each radius  $r_i$  is computed by summing over all azimuthal segments:

$$A_i = r_i \Delta\theta \sum_{\ell} I(\tilde{r}_{\ell}) \chi_{\ell} \left[ \chi_{\ell-1} \chi_{\ell+1} + (1 - \chi_{\ell-1}) \chi_{\ell+1} f(\delta_{\ell, \text{enter}}) + \chi_{\ell-1} (1 - \chi_{\ell+1}) f(\delta_{\ell, \text{exit}}) \right], \quad (\text{A9})$$

where  $I(\tilde{r}_{\ell})$  is the limb-darkened surface brightness at the segment center. The radial integration is then performed using the trapezoidal rule:  $A_{\text{sector}} = \sum_{i=0}^{N_r-1} A_i \Delta r$ , and the final magnification is obtained by normalizing the total lensed flux by the intrinsic source flux:  $A = \sum_{\text{sector}} A_{\text{sector}} / \rho^2$ , assuming the source brightness profile is already normalized to unit total flux.

## B. MARGINALIZING LINEAR FLUX PARAMETERS WITH NON-LINEAR PARAMETER INFERENCE

### B.1. Bayesian Formulation

Following [Hogg & Villar \(2021\)](#) and [Bartolić \(2023\)](#), consider a dataset  $\mathbf{y} = \{y_i\}$  with covariance matrix  $\mathbf{C}$ , and a

model that includes nonlinear parameters  $\theta$  and linear parameters  $\beta$ . We are interested in evaluating the likelihood:  $p(\mathbf{y}|\theta, \beta)$ . However, if we are not interested in the linear parameters themselves, the computational cost of inference can be reduced if the following marginalization is tractable:

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}|\theta, \beta) p(\beta|\theta) d\beta. \quad (\text{B10})$$

This integral requires a prior distribution  $p(\beta|\theta)$  that may depend on the nonlinear parameters. Alternatively, one can use Bayes' theorem to rewrite the integral in the following form:

$$p(\mathbf{y}|\theta, \beta) p(\beta|\theta) = p(\beta|\mathbf{y}, \theta) p(\mathbf{y}|\theta) \quad (\text{B11})$$

$$\Rightarrow p(\mathbf{y}|\theta) = \frac{p(\mathbf{y}|\theta, \beta) p(\beta|\theta)}{p(\beta|\mathbf{y}, \theta)}. \quad (\text{B12})$$

This expression enables the evaluation of the marginalized likelihood without explicitly performing the integral. If both the prior and the likelihood are multivariate Gaussian distributions, the expressions become:

$$\begin{aligned} p(\mathbf{y}|\theta, \beta) &= \mathcal{N}(\mathbf{y}|\mathbf{M}\beta, \mathbf{C}) \\ &\propto \exp\left(-\frac{1}{2}(\mathbf{y} - \mathbf{M}\beta)^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{M}\beta)\right) \\ p(\beta|\theta) &= \mathcal{N}(\beta|\mu, \Lambda) \\ &\propto \exp\left(-\frac{1}{2}(\beta - \mu)^\top \Lambda^{-1}(\beta - \mu)\right) \end{aligned}$$

$$\begin{aligned} \Rightarrow -\ln \mathcal{N}(\mathbf{y}|\mathbf{M}\beta, \mathbf{C}) \mathcal{N}(\beta|\mu, \Lambda) \\ \propto (\mathbf{y} - \mathbf{M}\beta)^\top \mathbf{C}^{-1}(\mathbf{y} - \mathbf{M}\beta) + (\beta - \mu)^\top \Lambda^{-1}(\beta - \mu). \end{aligned} \quad (\text{B13})$$

The marginalized likelihood  $p(\mathbf{y}|\theta)$  can then be evaluated as:

$$\mathcal{N}(\mathbf{y}|\mathbf{M}\beta, \mathbf{C}) \mathcal{N}(\beta|\mu, \Lambda) = \mathcal{N}(\beta|\mathbf{a}, \mathbf{A}) \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B}), \quad (\text{B14})$$

where

$$\begin{aligned} \mathbf{A}^{-1} &= \Lambda^{-1} + \mathbf{M}^\top \mathbf{C}^{-1} \mathbf{M}, \\ \mathbf{a} &= \Lambda^{-1} \mu + \mathbf{M}^\top \mathbf{C}^{-1} \mathbf{y}, \\ \mathbf{B} &= \mathbf{C} + \mathbf{M} \Lambda \mathbf{M}^\top, \\ \mathbf{b} &= \mathbf{M} \mu. \end{aligned}$$

Thus, since the term involving the linear parameter  $\beta$  can be integrated out, we obtain:

$$p(\mathbf{y}|\theta) = \int \mathcal{N}(\beta|\mathbf{a}, \mathbf{A}) \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B}) d\beta = \mathcal{N}(\mathbf{y}|\mathbf{b}, \mathbf{B}). \quad (\text{B15})$$

## B.2. Application to Microlensing Modeling

To model the observed fluxes in a gravitational microlensing event, we introduce a hierarchical framework in which each dataset  $j$  is described by a set of linear flux parameters  $\beta_j$  and shared non-linear parameters  $\theta$  that characterize the lensing model. Assuming independent Gaussian noise for each dataset, the joint likelihood can be written as

$$p(\mathbf{D}|\beta, \theta) = \prod_{j=1}^J p(\mathbf{F}_j|\beta_j, \theta) \quad (\text{B16})$$

where  $J$  and  $j$  refer to the number of datasets and their index, respectively. When we assume Gaussian priors for linear parameters of flux parameters  $\beta$ , we can marginalize the probabilities of linear parameters for the non-linear parameter inference (Hogg et al. 2020). In this formulation, the design matrix  $\mathbf{M}_j$  depends only on the non-linear parameters  $\theta$ , and relates them to the expected contribution of the linear flux parameters. Using the formula in the previous section, the negative log-likelihood (NLL) for non-linear parameters can be written as

$$\begin{aligned} \text{NLL}(\theta) &= \frac{1}{2}(\mathbf{F}_j - \mathbf{M}_j \mu)^\top (\mathbf{C}_j + \mathbf{M}_j \Lambda \mathbf{M}_j^\top)^{-1} (\mathbf{F}_j - \mathbf{M}_j \mu) \\ &\quad + \frac{1}{2} \ln |\mathbf{C}_j + \mathbf{M}_j \Lambda \mathbf{M}_j^\top| \end{aligned} \quad (\text{B17})$$

where

$$\begin{aligned} (\mathbf{C}_j + \mathbf{M}_j \Lambda \mathbf{M}_j^\top)^{-1} \\ = \mathbf{C}_j^{-1} - \mathbf{C}_j^{-1} \mathbf{M}_j (\Lambda^{-1} + \mathbf{M}_j^\top \mathbf{C}_j^{-1} \mathbf{M}_j)^{-1} \mathbf{M}_j \mathbf{C}_j^{-1} \end{aligned} \quad (\text{B18})$$

$$\begin{aligned} \ln |\mathbf{C}_j + \mathbf{M}_j \Lambda \mathbf{M}_j^\top| \\ = \ln |\mathbf{C}_j| + \ln |\Lambda| + \ln |\Lambda^{-1} + \mathbf{M}_j^\top \mathbf{C}_j^{-1} \mathbf{M}_j|, \end{aligned} \quad (\text{B19})$$

which are derived from the matrix inversion lemma, also known as the Woodbury matrix identity.

Regarding a single-source microlensing model:

$$\{\mathbf{F}\}_j = [\mathbf{A}(t_j; \theta) - 1] F_{S,j} + F_{\text{base},j}, \quad (\text{B20})$$

marginalizing over the linear parameters, we define

$$\mathbf{M}_j = \begin{pmatrix} A(t_{j,1}; \theta) - 1 & 1 \\ \vdots & \\ A(t_{j,n_j}; \theta) - 1 & 1 \end{pmatrix} \in \mathbb{R}^{n_j \times 2}, \quad (\text{B21})$$

$$\mu_j = (F_{S,j}, F_{\text{base},j})^\top, \quad (\text{B22})$$

$$\Lambda = (\sigma_{F_{S,j}}^2, \sigma_{F_{\text{base},j}}^2)^\top, \quad (\text{B23})$$

and by substituting these expressions, it becomes possible to estimate the microlensing parameters without directly estimating the linear parameters  $F_S$  and  $F_{\text{base}}$ . In general, since

we can assume that  $\mu = \mathbf{0}$ ,  $\Lambda$  and  $C_j$  are diagonal, the evaluation of Equation (B17) is not expensive to compute.

If you want to obtain the posterior distribution of the linear parameters, you can sample from the following distribution:

$$p(\beta \mid \mathbf{y}, \theta) = \mathcal{N}(\beta \mid \mathbf{a}(\theta), \mathbf{A}(\theta)) \quad (\text{B24})$$

$$\mathbf{A}(\theta)^{-1} = \Lambda^{-1} + \mathbf{M}(\theta)^\top \mathbf{C}^{-1} \mathbf{M}(\theta) \quad (\text{B25})$$

$$\mathbf{a}(\theta) = \mathbf{A}(\theta) [\Lambda^{-1} \mu + \mathbf{M}(\theta)^\top \mathbf{C}^{-1} \mathbf{y}] \quad (\text{B26})$$

## REFERENCES

- Abadi, M., Agarwal, A., Barham, P., et al. 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. <https://www.tensorflow.org/>
- Aberth, O. 1973, *Mathematics of Computation*, 27, 339, doi: [10.1090/S0025-5718-1973-0329236-7](https://doi.org/10.1090/S0025-5718-1973-0329236-7)
- Agol, E., Hernandez, D. M., & Langford, Z. 2021, *MNRAS*, 507, 1582, doi: [10.1093/mnras/stab2044](https://doi.org/10.1093/mnras/stab2044)
- Akeson, R., Armus, L., Bachelet, E., et al. 2019, arXiv e-prints, arXiv:1902.05569, doi: [10.48550/arXiv.1902.05569](https://doi.org/10.48550/arXiv.1902.05569)
- Bartolić, F. 2023, Ph.d. thesis, University of St Andrews, St Andrews, UK, doi: [10.17630/sta/334](https://doi.org/10.17630/sta/334)
- Bennett, D. P. 2010, *ApJ*, 716, 1408, doi: [10.1088/0004-637X/716/2/1408](https://doi.org/10.1088/0004-637X/716/2/1408)
- Bennett, D. P., & Rhie, S. H. 1996, *ApJ*, 472, 660, doi: [10.1086/178096](https://doi.org/10.1086/178096)
- . 2002, *ApJ*, 574, 985, doi: [10.1086/340977](https://doi.org/10.1086/340977)
- Bennett, D. P., Bond, I. A., Udalski, A., et al. 2008, *ApJ*, 684, 663, doi: [10.1086/589940](https://doi.org/10.1086/589940)
- Bennett, D. P., Rhie, S. H., Nikolaev, S., et al. 2010, *ApJ*, 713, 837, doi: [10.1088/0004-637X/713/2/837](https://doi.org/10.1088/0004-637X/713/2/837)
- Betancourt, M. 2017, arXiv e-prints, arXiv:1701.02434. <https://arxiv.org/abs/1701.02434>
- Bingham, E., Chen, J. P., Jankowiak, M., et al. 2018, arXiv e-prints, arXiv:1810.09538, doi: [10.48550/arXiv.1810.09538](https://doi.org/10.48550/arXiv.1810.09538)
- Bingham, E., Chen, J. P., Jankowiak, M., et al. 2019, *J. Mach. Learn. Res.*, 20, 28:1. <http://jmlr.org/papers/v20/18-403.html>
- Blondel, M., & Roulet, V. 2024, arXiv e-prints, arXiv:2403.14606, doi: [10.48550/arXiv.2403.14606](https://doi.org/10.48550/arXiv.2403.14606)
- Boettner, C. 2025, *A&A*, doi: [10.1051/0004-6361/202554518](https://doi.org/10.1051/0004-6361/202554518)
- Bozza, V. 2010, *MNRAS*, 408, 2188, doi: [10.1111/j.1365-2966.2010.17265.x](https://doi.org/10.1111/j.1365-2966.2010.17265.x)
- Bozza, V., Bachelet, E., Bartolić, F., et al. 2018, *MNRAS*, 479, 5157, doi: [10.1093/mnras/sty1791](https://doi.org/10.1093/mnras/sty1791)
- Bozza, V., Khalouei, E., & Bachelet, E. 2021, *MNRAS*, 505, 126, doi: [10.1093/mnras/stab1376](https://doi.org/10.1093/mnras/stab1376)
- Bozza, V., Saggese, V., Covone, G., Rota, P., & Zhang, J. 2025, *A&A*, 694, A219, doi: [10.1051/0004-6361/202452648](https://doi.org/10.1051/0004-6361/202452648)
- Bradbury, J., Frostig, R., Hawkins, P., et al. 2018, JAX: composable transformations of Python+NumPy programs, 0.3.13. <http://github.com/google/jax>
- Carpenter, B., Gelman, A., Hoffman, M. D., et al. 2017, *Journal of Statistical Software*, 76, 1
- Carter, J. A., & Winn, J. N. 2009, *ApJ*, 704, 51, doi: [10.1088/0004-637X/704/1/51](https://doi.org/10.1088/0004-637X/704/1/51)
- Cassan, A. 2017, *MNRAS*, 468, 3993, doi: [10.1093/mnras/stx849](https://doi.org/10.1093/mnras/stx849)
- Christiansen, J. L., McElroy, D. L., Harbut, M., et al. 2025, arXiv e-prints, arXiv:2506.03299, doi: [10.48550/arXiv.2506.03299](https://doi.org/10.48550/arXiv.2506.03299)
- DeepMind, Babuschkin, I., Baumli, K., et al. 2020, The DeepMind JAX Ecosystem. <http://github.com/google-deepmind>
- Dominik, M. 1998a, *A&A*, 333, L79, doi: [10.48550/arXiv.astro-ph/9804059](https://doi.org/10.48550/arXiv.astro-ph/9804059)
- . 1998b, *A&A*, 329, 361, doi: [10.48550/arXiv.astro-ph/9702039](https://doi.org/10.48550/arXiv.astro-ph/9702039)
- Dong, S., DePoy, D. L., Gaudi, B. S., et al. 2006, *ApJ*, 642, 842, doi: [10.1086/501224](https://doi.org/10.1086/501224)
- Duane, S., Kennedy, A. D., Pendleton, B. J., & Roweth, D. 1987, *Physics Letters B*, 195, 216, doi: [10.1016/0370-2693\(87\)91197-X](https://doi.org/10.1016/0370-2693(87)91197-X)
- Dupourqué, S., Barret, D., Diez, C. M., Guillot, S., & Quintin, E. 2024, *A&A*, 690, A317, doi: [10.1051/0004-6361/202451736](https://doi.org/10.1051/0004-6361/202451736)
- Edwards, T. D. P., Wong, K. W. K., Lam, K. K. H., et al. 2024, *PhRvD*, 110, 064028, doi: [10.1103/PhysRevD.110.064028](https://doi.org/10.1103/PhysRevD.110.064028)
- Ehrlich, L. W. 1967, *Communications of the ACM*, 10, 107, doi: [10.1145/363067.363115](https://doi.org/10.1145/363067.363115)
- Fatheddin, H., & Sajadian, S. 2022, *MNRAS*, 514, 4379, doi: [10.1093/mnras/stac1565](https://doi.org/10.1093/mnras/stac1565)
- Foreman-Mackey, D. 2016, *The Journal of Open Source Software*, 1, 24, doi: [10.21105/joss.00024](https://doi.org/10.21105/joss.00024)
- Foreman-Mackey, D., Yu, W., Yadav, S., et al. 2024, dfm/tinygp: The tiniest of Gaussian Process libraries, v0.3.0, Zenodo, doi: [10.5281/zenodo.10463641](https://doi.org/10.5281/zenodo.10463641)
- Gelman, A., & Rubin, D. B. 1992, *Statistical Science*, 7, 457, doi: [10.1214/ss/1177011136](https://doi.org/10.1214/ss/1177011136)
- Giordano, M., Nucita, A. A., De Paolis, F., & Ingrosso, G. 2015, *MNRAS*, 453, 2017, doi: [10.1093/mnras/stv1764](https://doi.org/10.1093/mnras/stv1764)
- Gould, A. 2004, *ApJ*, 606, 319, doi: [10.1086/382782](https://doi.org/10.1086/382782)
- . 2008, *ApJ*, 681, 1593, doi: [10.1086/588601](https://doi.org/10.1086/588601)
- Gould, A., & Gaucherel, C. 1997, *ApJ*, 477, 580, doi: [10.1086/303751](https://doi.org/10.1086/303751)
- Gould, A., & Yee, J. C. 2014, *ApJ*, 784, 64, doi: [10.1088/0004-637X/784/1/64](https://doi.org/10.1088/0004-637X/784/1/64)

- Griewank. 1992, *Optimization Methods and Software*, 1, 35, doi: [10.1080/10556789208805505](https://doi.org/10.1080/10556789208805505)
- Hendry, M. A., Bryce, H. M., & Valls-Gabaud, D. 2002, *MNRAS*, 335, 539, doi: [10.1046/j.1365-8711.2002.05496.x](https://doi.org/10.1046/j.1365-8711.2002.05496.x)
- Hoffman, M. D., & Gelman, A. 2011, arXiv e-prints, arXiv:1111.4246, doi: [10.48550/arXiv.1111.4246](https://doi.org/10.48550/arXiv.1111.4246)
- Hogg, D. W., Price-Whelan, A. M., & Leistedt, B. 2020, arXiv e-prints, arXiv:2005.14199, doi: [10.48550/arXiv.2005.14199](https://doi.org/10.48550/arXiv.2005.14199)
- Hogg, D. W., & Villar, S. 2021, *PASP*, 133, 093001, doi: [10.1088/1538-3873/ac20ac](https://doi.org/10.1088/1538-3873/ac20ac)
- Hundertmark, M. P. G. 2011, PhD thesis, Georg August University of Gottingen, Germany
- Isi, M., & Farr, W. M. 2021, arXiv e-prints, arXiv:2107.05609, doi: [10.48550/arXiv.2107.05609](https://doi.org/10.48550/arXiv.2107.05609)
- Kawahara, H., Kawashima, Y., Masuda, K., et al. 2022, *ApJS*, 258, 31, doi: [10.3847/1538-4365/ac3b4d](https://doi.org/10.3847/1538-4365/ac3b4d)
- Kawahara, H., Kawashima, Y., Tada, S., et al. 2025, *ApJ*, 985, 263, doi: [10.3847/1538-4357/adcba2](https://doi.org/10.3847/1538-4357/adcba2)
- Kayser, R., Refsdal, S., & Stabell, R. 1986, *A&A*, 166, 36
- Lam, C. Y., Lu, J. R., Udalski, A., et al. 2022, *ApJL*, 933, L23, doi: [10.3847/2041-8213/ac7442](https://doi.org/10.3847/2041-8213/ac7442)
- Lam, C. Y., Abrams, N., Andrews, J., et al. 2023, arXiv e-prints, arXiv:2306.12514, doi: [10.48550/arXiv.2306.12514](https://doi.org/10.48550/arXiv.2306.12514)
- Masuda, K., Libby-Roberts, J. E., Livingston, J. H., et al. 2024, *AJ*, 168, 294, doi: [10.3847/1538-3881/ad83d3](https://doi.org/10.3847/1538-3881/ad83d3)
- McDougall, A., & Albrow, M. D. 2016, *MNRAS*, 456, 565, doi: [10.1093/mnras/stv2609](https://doi.org/10.1093/mnras/stv2609)
- Miyazaki, S., Johnson, S. A., Sumi, T., et al. 2021, *AJ*, 161, 84, doi: [10.3847/1538-3881/abcec2](https://doi.org/10.3847/1538-3881/abcec2)
- Miyazaki, S., & Kawahara, H. 2025, ShotaMiyazaki94/microjax: micorjaxx 0.1.1, v0.1.1, Zenodo, doi: [10.5281/zenodo.17247893](https://doi.org/10.5281/zenodo.17247893)
- Miyazaki, S., Sumi, T., Bennett, D. P., et al. 2020, *AJ*, 159, 76, doi: [10.3847/1538-3881/ab64de](https://doi.org/10.3847/1538-3881/ab64de)
- Paszke, A., Gross, S., Massa, F., et al. 2019, arXiv e-prints, arXiv:1912.01703, doi: [10.48550/arXiv.1912.01703](https://doi.org/10.48550/arXiv.1912.01703)
- Penny, M. T., Gaudi, B. S., Kerins, E., et al. 2019, *ApJS*, 241, 3, doi: [10.3847/1538-4365/aafb69](https://doi.org/10.3847/1538-4365/aafb69)
- Phan, D., Pradhan, N., & Jankowiak, M. 2019, arXiv e-prints, arXiv:1912.11554. <https://arxiv.org/abs/1912.11554>
- Pinder, T., & Dodd, D. 2022, *Journal of Open Source Software*, 7, 4455, doi: [10.21105/joss.04455](https://doi.org/10.21105/joss.04455)
- Ren, H., & Zhu, W. 2025, *AJ*, 169, 170, doi: [10.3847/1538-3881/adb1b2](https://doi.org/10.3847/1538-3881/adb1b2)
- Rhie, S. H. 2003, arXiv e-prints, astro, doi: [10.48550/arXiv.astro-ph/0305166](https://doi.org/10.48550/arXiv.astro-ph/0305166)
- Rota, P., Hirao, Y., Bozza, V., et al. 2021, *AJ*, 162, 59, doi: [10.3847/1538-3881/ac0155](https://doi.org/10.3847/1538-3881/ac0155)
- Sahu, K. C., Anderson, J., Casertano, S., et al. 2022, *ApJ*, 933, 83, doi: [10.3847/1538-4357/ac739e](https://doi.org/10.3847/1538-4357/ac739e)
- Skowron, J., & Gould, A. 2012, arXiv e-prints, arXiv:1203.1034, doi: [10.48550/arXiv.1203.1034](https://doi.org/10.48550/arXiv.1203.1034)
- Skowron, J., Udalski, A., Gould, A., et al. 2011, *ApJ*, 738, 87, doi: [10.1088/0004-637X/738/1/87](https://doi.org/10.1088/0004-637X/738/1/87)
- Spergel, D., Gehrels, N., Baltay, C., et al. 2015, arXiv e-prints, arXiv:1503.03757, doi: [10.48550/arXiv.1503.03757](https://doi.org/10.48550/arXiv.1503.03757)
- Stumpe, M. C., Smith, J. C., Van Cleve, J. E., et al. 2012, *PASP*, 124, 985, doi: [10.1086/667698](https://doi.org/10.1086/667698)
- The Theano Development Team, Al-Rfou, R., Alain, G., et al. 2016, arXiv e-prints, arXiv:1605.02688, doi: [10.48550/arXiv.1605.02688](https://doi.org/10.48550/arXiv.1605.02688)
- Thompson, A. C., Vernardos, G., Fluke, C. J., & Barsdell, B. R. 2014, GPU-D: Generating cosmological microlensing magnification maps, *Astrophysics Source Code Library*, record ascl:1403.001
- Udalski, A., Szymański, M. K., & Szymański, G. 2015a, *AcA*, 65, 1, doi: [10.48550/arXiv.1504.05966](https://doi.org/10.48550/arXiv.1504.05966)
- Udalski, A., Yee, J. C., Gould, A., et al. 2015b, *ApJ*, 799, 237, doi: [10.1088/0004-637X/799/2/237](https://doi.org/10.1088/0004-637X/799/2/237)
- Wambsganss, J. 1997, *MNRAS*, 284, 172, doi: [10.1093/mnras/284.1.172](https://doi.org/10.1093/mnras/284.1.172)
- Wang, S., Wang, L., & Dong, S. 2025, *ApJS*, 276, 40, doi: [10.3847/1538-4365/ad9b8d](https://doi.org/10.3847/1538-4365/ad9b8d)
- Weisenbach, L. 2025, arXiv e-prints, arXiv:2506.02121, doi: [10.48550/arXiv.2506.02121](https://doi.org/10.48550/arXiv.2506.02121)
- Witt, H. J. 1990, *A&A*, 236, 311
- Wong, K. W. K., Isi, M., & Edwards, T. D. P. 2023, *ApJ*, 958, 129, doi: [10.3847/1538-4357/acf5cd](https://doi.org/10.3847/1538-4357/acf5cd)
- Wyrzykowski, Ł., Mróz, P., Rybicki, K. A., et al. 2020, *A&A*, 633, A98, doi: [10.1051/0004-6361/201935097](https://doi.org/10.1051/0004-6361/201935097)
- Zheng, W., Chen, X., Li, G., & Chen, H.-Z. 2022, *ApJ*, 931, 114, doi: [10.3847/1538-4357/ac68ea](https://doi.org/10.3847/1538-4357/ac68ea)