relensing

Reconstructing the mass profile of galaxy clusters from gravitational lensing

Based on arXiv:2201.10076

Daniel Alexdy Torres Ballesteros (daatorresba@unal.edu.co)
Supervisor: Prof. Leonardo Castañeda Colorado (lcastanedac@unal.edu.co)

Universidad Nacional de Colombia

Cosmology from home 2022

relensing

Reconstructing the mass profile of galaxy clusters from gravitational lensing

Based on arXiv:2201.10076

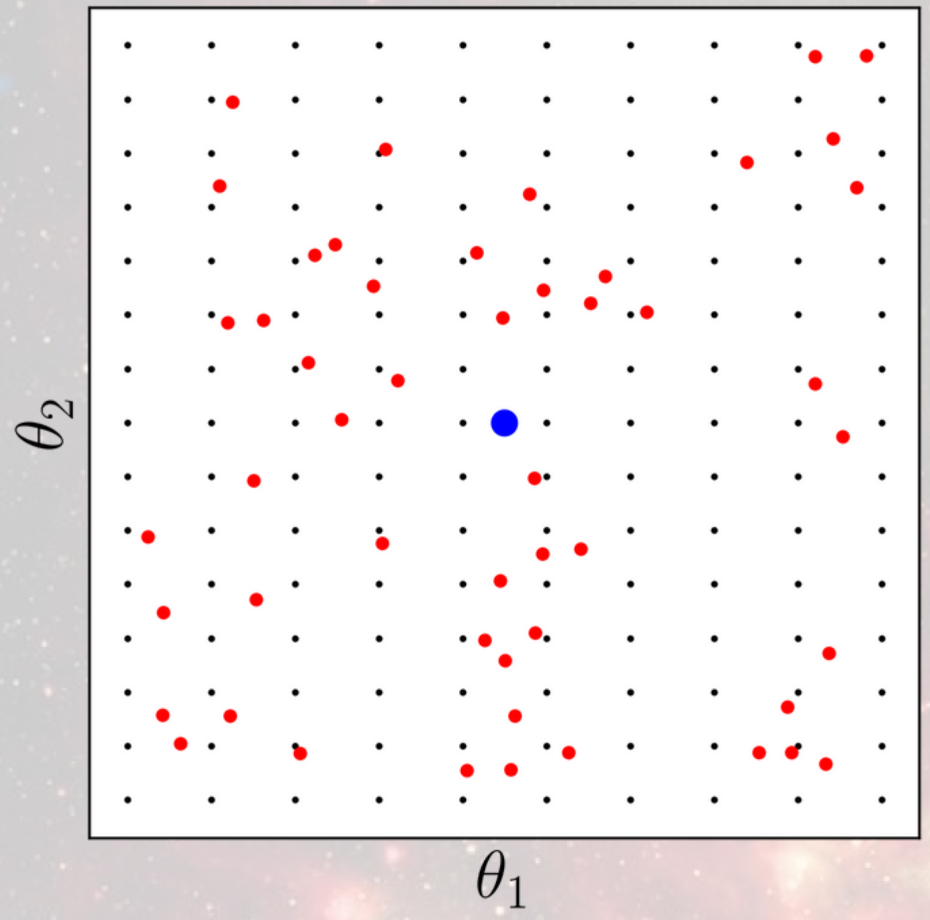
Daniel Alexdy Torres Ballesteros (daatorresba@unal.edu.co)

Supervisor: Prof. Leonardo Castañeda Colorado (lcastanedac@unal.edu.co)

Universidad Nacional de Colombia

Cosmology from home 2022

GRID

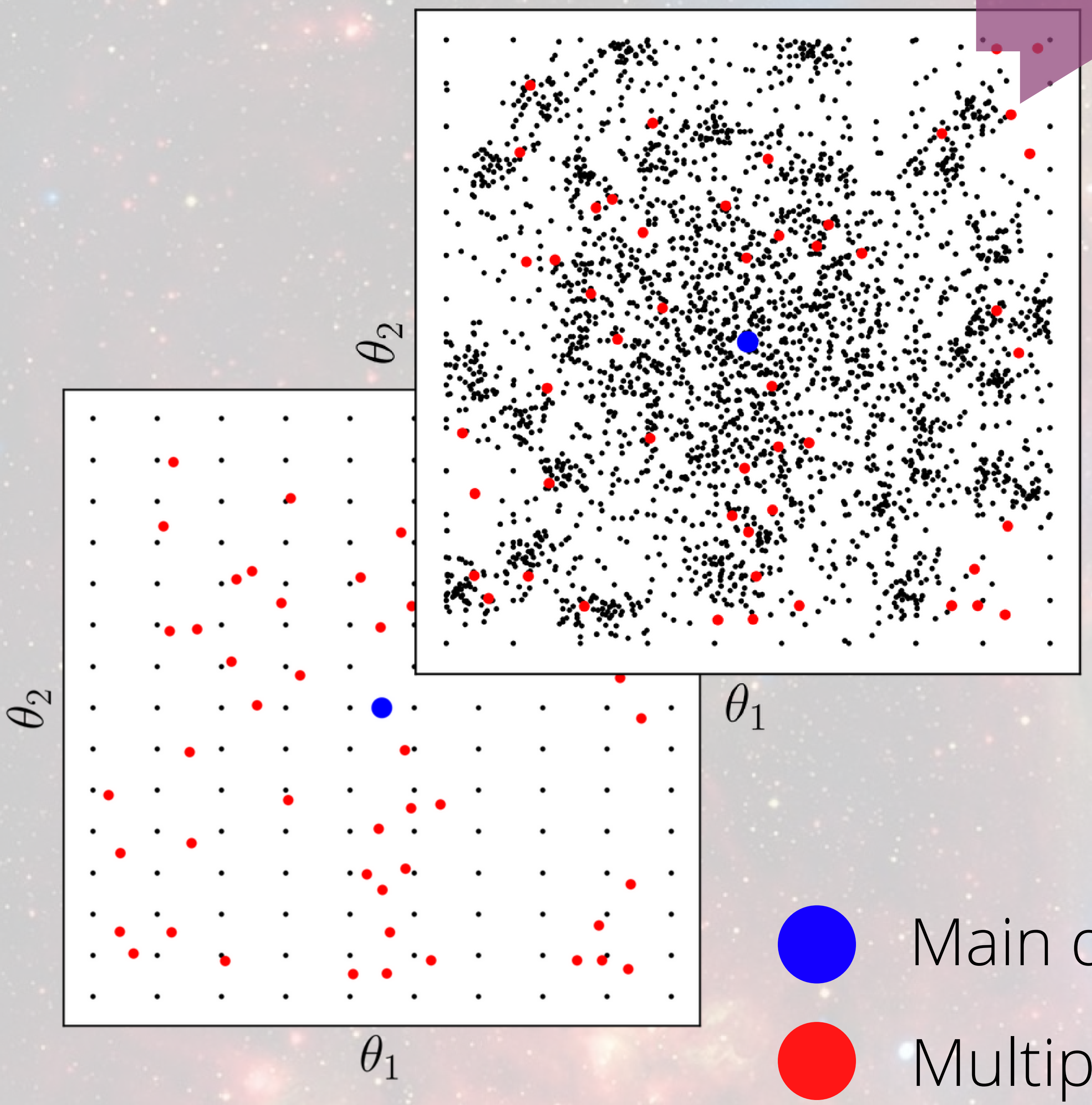


Main deflector

Multiple images

GRID

N nodes in total

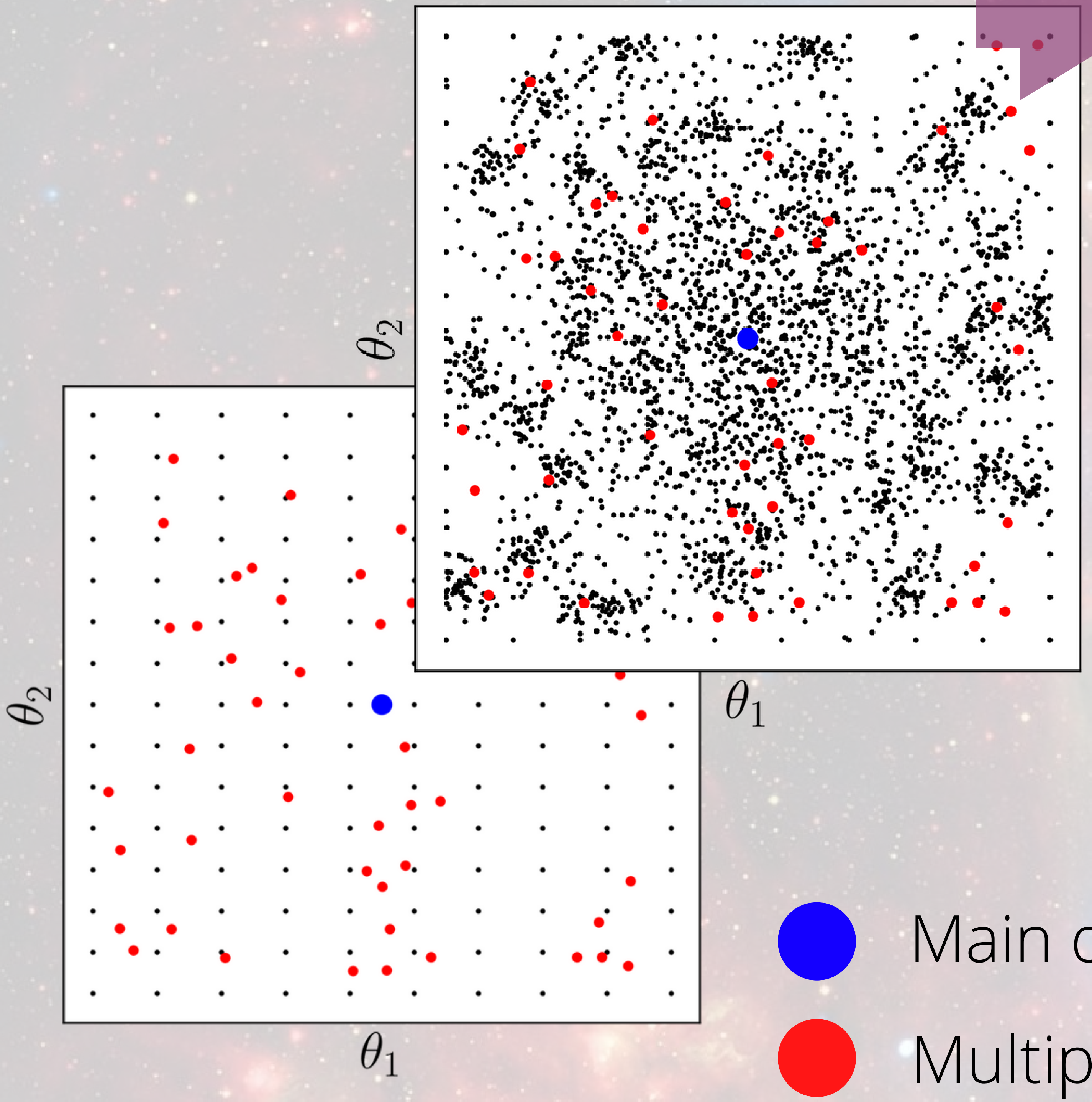


Main deflector

Multiple images

GRID

N nodes in total



For an arbitrary angular position $\boldsymbol{\theta}_i$

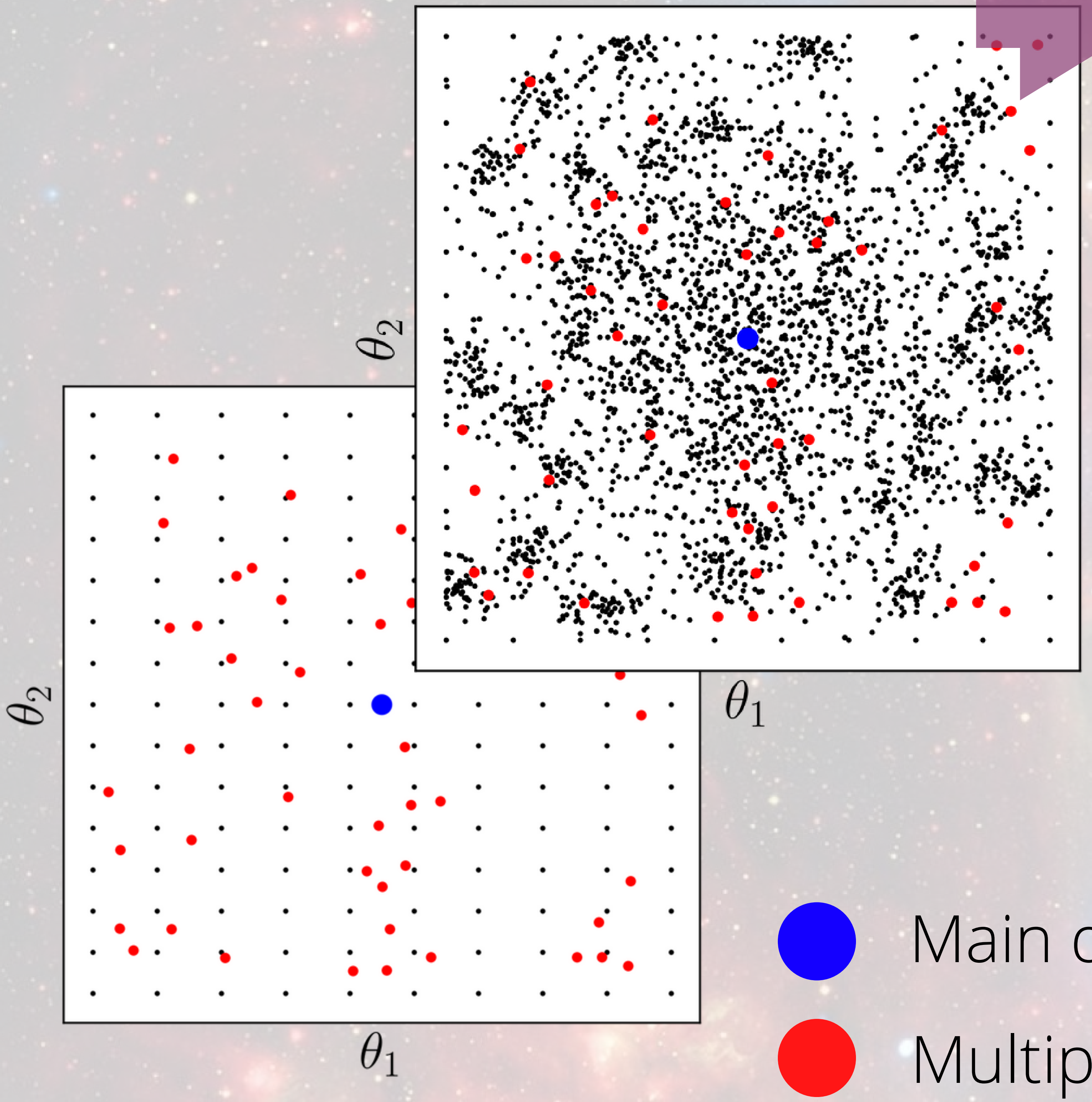
$$\psi_i = \sum_{j=1}^{N} \mathcal{P}_{ij}\psi_j; \qquad \kappa_i = \sum_{j=1}^{N} \mathcal{K}_{ij}\psi_j,$$

$$\gamma_{n,i} = \sum_{j=1}^{N} \mathcal{G}_{n,ij}\psi_j; \qquad \alpha_{n,i} = \sum_{j=1}^{N} \mathcal{D}_{n,ij}\psi_j \quad (n = 1, 2)$$

- Main deflector
- Multiple images

GRID

N nodes in total



For an arbitrary angular position $\boldsymbol{\theta}_i$

$$\psi_i = \sum_{j=1}^{N} \mathcal{P}_{ij}\psi_j; \qquad \kappa_i = \sum_{j=1}^{N} \mathcal{K}_{ij}\psi_j,$$

$$\gamma_{n,i} = \sum_{j=1}^{N} \mathcal{G}_{n,ij}\psi_j; \qquad \alpha_{n,i} = \sum_{j=1}^{N} \mathcal{D}_{n,ij}\psi_j \quad (n = 1, 2)$$

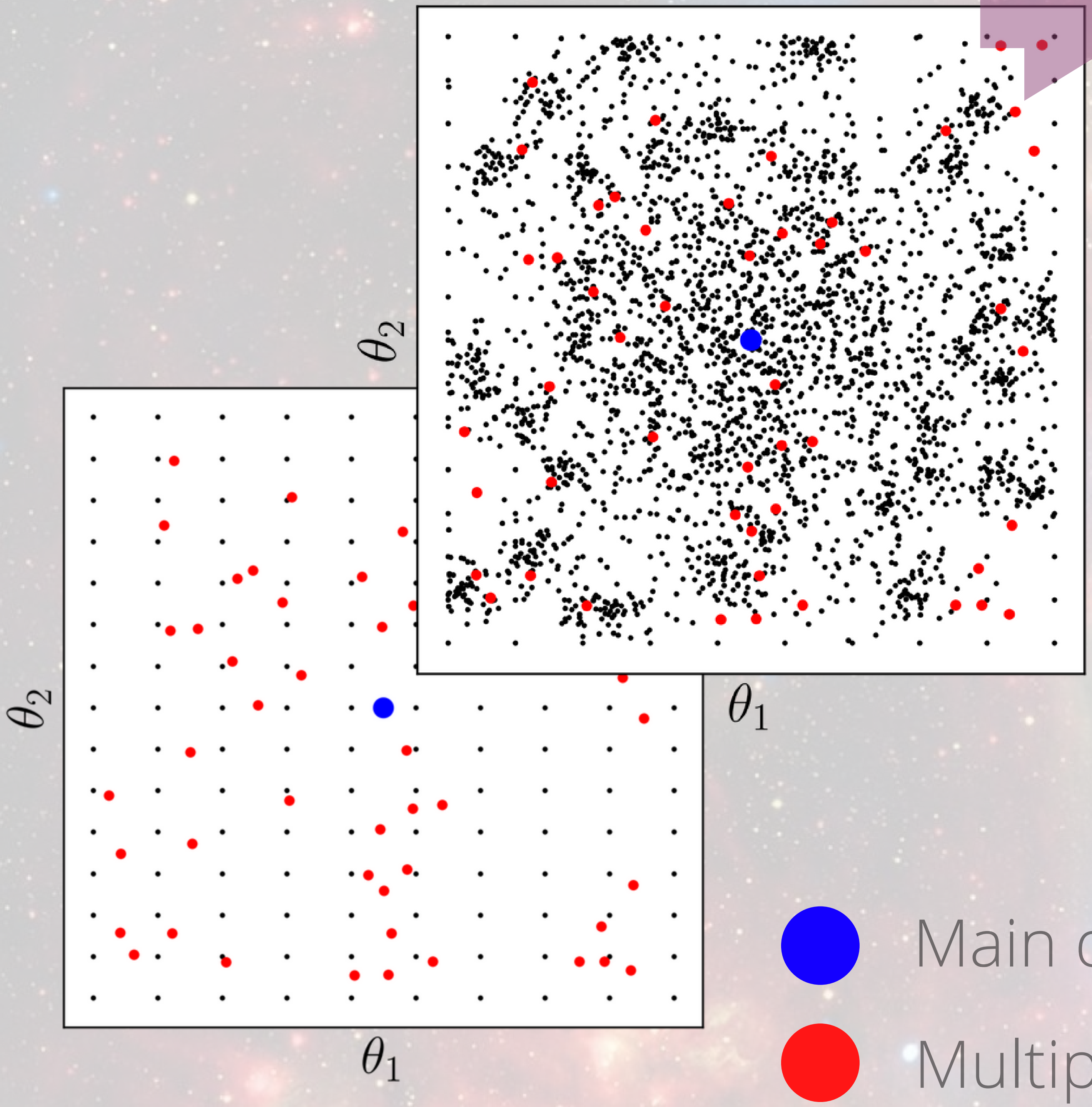
Generalized Finite Differences

Main deflector

Multiple images

GRID

N nodes in total



Main deflector

Multiple images

For an arbitrary angular position $\boldsymbol{\theta}_i$

$$\psi_i = \sum_{j=1}^{N} \mathcal{P}_{ij}\psi_j; \quad \kappa_i = \sum_{j=1}^{N} \mathcal{K}_{ij}\psi_j,$$

$$\gamma_{n,i} = \sum_{j=1}^{N} \mathcal{G}_{n,ij}\psi_j; \quad \alpha_{n,i} = \sum_{j=1}^{N} \mathcal{D}_{n,ij}\psi_j \quad (n = 1, 2)$$

Free-Form reconstruction!

Generalized Finite Differences

ALGORITHM

Input: Initial conditions and the initial guess $\psi^{(0)}$.
Compute $\kappa^{(0)}$, $\gamma_1^{(0)}$ and $\gamma_2^{(0)}$ from $\psi^{(0)}$.
while outer_end = False **do**
  Compute the nodes weight.
  while inner_end = False **do**
    Compute the system of linear equations needed and solve it $\longrightarrow \psi^{(n)}$.
    Smoothing of $\psi^{(n)} \longrightarrow$ New $\psi^{(n)}$ (optional).
    Compute $\kappa^{(n)}$ from $\psi^{(n)}$.
    if $\left|\kappa_j^{(n)} - \kappa_j^{(n-1)}\right| \leq$ tolerance **then**
      inner_end = True.
  Compute χ_s^2.
  if $\chi_s^2 \leq q(2N_{\mathrm{img}})$ **then**
    outer_end = True.
  if outer_end = False **then**
    Apply grid refinement.
    Compute $\psi^{(m)}$, $\kappa^{(m)}$, $\gamma_1^{(m)}$ and $\gamma_2^{(m)}$ on the new grid.
return ψ

ALGORITHM

Outer level

```
Input: Initial conditions and the initial guess ψ^(0).
Compute κ^(0), γ_1^(0) and γ_2^(0) from ψ^(0).
while outer_end = False do
    Compute the nodes weight.
    while inner_end = False do
        Compute the system of linear equations needed and solve it ⟶ ψ^(n).
        Smoothing of ψ^(n) ⟶ New ψ^(n) (optional).
        Compute κ^(n) from ψ^(n).
        if |κ_j^(n) − κ_j^(n−1)| ≤ tolerance then
            inner_end = True.
    Compute χ_s^2.
    if χ_s^2 ≤ q(2N_img) then
        outer_end = True.
    if outer_end = False then
        Apply grid refinement.
        Compute ψ^(m), κ^(m), γ_1^(m) and γ_2^(m) on the new grid.
return ψ
```

ALGORITHM

Outer level

Input: Initial conditions and the initial guess $\psi^{(0)}$.
Compute $\kappa^{(0)}$, $\gamma_1^{(0)}$ and $\gamma_2^{(0)}$ from $\psi^{(0)}$.
while outer_end = False **do**
  Compute the nodes weight.
  while inner_end = False **do**
    Compute the system of linear equations needed and solve it $\longrightarrow \psi^{(n)}$.
    Smoothing of $\psi^{(n)} \longrightarrow$ New $\psi^{(n)}$ (optional).
    Compute $\kappa^{(n)}$ from $\psi^{(n)}$.
    if $\left|\kappa_j^{(n)} - \kappa_j^{(n-1)}\right| \leq$ tolerance **then**
      inner_end = True.
  Compute χ_s^2.
  if $\chi_s^2 \leq q(2N_{\mathrm{img}})$ **then**
    outer_end = True.
  if outer_end = False **then**
    Apply grid refinement.
    Compute $\psi^{(m)}$, $\kappa^{(m)}$, $\gamma_1^{(m)}$ and $\gamma_2^{(m)}$ on the new grid.
return ψ

ALGORITHM

Outer level

Inner level

Input: Initial conditions and the initial guess $\psi^{(0)}$.
Compute $\kappa^{(0)}$, $\gamma_1^{(0)}$ and $\gamma_2^{(0)}$ from $\psi^{(0)}$.
while outer_end = False **do**
    Compute the nodes weight.
    while inner_end = False **do**
        Compute the system of linear equations needed and solve it $\longrightarrow \psi^{(n)}$.
        Smoothing of $\psi^{(n)} \longrightarrow$ New $\psi^{(n)}$ (optional).
        Compute $\kappa^{(n)}$ from $\psi^{(n)}$.
        if $\left|\kappa_j^{(n)} - \kappa_j^{(n-1)}\right| \leq$ tolerance **then**
            inner_end = True.
    Compute χ_s^2.
    if $\chi_s^2 \leq q(2N_{\mathrm{img}})$ **then**
        outer_end = True.
    if outer_end = False **then**
        Apply grid refinement.
        Compute $\psi^{(m)}$, $\kappa^{(m)}$, $\gamma_1^{(m)}$ and $\gamma_2^{(m)}$ on the new grid.
return ψ

ALGORITHM

Outer level

Inner level

Input: Initial conditions and the initial guess $\psi^{(0)}$.
Compute $\kappa^{(0)}$, $\gamma_1^{(0)}$ and $\gamma_2^{(0)}$ from $\psi^{(0)}$.
while outer_end = False **do**
    Compute the nodes weight.
    while inner_end = False **do**
        Compute the system of linear equations needed and solve it $\longrightarrow \psi^{(n)}$.
        Smoothing of $\psi^{(n)} \longrightarrow$ New $\psi^{(n)}$ (optional).
        Compute $\kappa^{(n)}$ from $\psi^{(n)}$.
        if $\left|\kappa_j^{(n)} - \kappa_j^{(n-1)}\right| \leq$ tolerance **then**
            inner_end = True.
    Compute χ_s^2.
    if $\chi_s^2 \leq q(2N_{\mathrm{img}})$ **then**
        outer_end = True.
    if outer_end = False **then**
        Apply grid refinement.
        Compute $\psi^{(m)}$, $\kappa^{(m)}$, $\gamma_1^{(m)}$ and $\gamma_2^{(m)}$ on the new grid.
return ψ

ALGORITHM

Outer level

Inner level

```
Input: Initial conditions and the initial guess ψ^(0).
Compute κ^(0), γ_1^(0) and γ_2^(0) from ψ^(0).
while outer_end = False do
    Compute the nodes weight.
    while inner_end = False do
        Compute the system of linear equations needed and solve it ⟶ ψ^(n).
        Smoothing of ψ^(n) ⟶ New ψ^(n) (optional).
        Compute κ^(n) from ψ^(n).
        if |κ_j^(n) − κ_j^(n−1)| ≤ tolerance then
            inner_end = True.
    Compute χ_s^2.
    if χ_s^2 ≤ q(2N_img) then
        outer_end = True.
    if outer_end = False then
        Apply grid refinement.
        Compute ψ^(m), κ^(m), γ_1^(m) and γ_2^(m) on the new grid.
return ψ
```

$$\chi^2(\psi_j) := \underbrace{\chi_s^2(\psi_j)}_{\text{Strong lensing}} + \overbrace{\chi_w^2(\psi_j)}^{\text{Weak lensing}} + \underbrace{\chi^2_{\kappa(R)}(\psi_j) + \chi^2_{\gamma(R)}(\psi_j)}_{\text{Regularization}}$$

ALGORITHM

Input: Initial conditions and the initial guess $\psi^{(0)}$.
Compute $\kappa^{(0)}$, $\gamma_1^{(0)}$ and $\gamma_2^{(0)}$ from $\psi^{(0)}$.
while outer_end = False **do**
    Compute the nodes weight.
    while inner_end = False **do**
        Compute the system of linear equations needed and solve it $\longrightarrow \psi^{(n)}$.
        Smoothing of $\psi^{(n)} \longrightarrow$ New $\psi^{(n)}$ (optional).
        Compute $\kappa^{(n)}$ from $\psi^{(n)}$.
        if $\left|\kappa_j^{(n)} - \kappa_j^{(n-1)}\right| \leq$ tolerance **then**
            inner_end = True.
    Compute χ_s^2.
    if $\chi_s^2 \leq q(2N_{\text{img}})$ **then**
        outer_end = True.
    if outer_end = False **then**
        Apply grid refinement.
        Compute $\psi^{(m)}$, $\kappa^{(m)}$, $\gamma_1^{(m)}$ and $\gamma_2^{(m)}$ on the new grid.
return ψ

Outer level

Inner level

$$\chi^2(\psi_j) := \underbrace{\chi_s^2(\psi_j)}_{\text{Strong lensing}} + \overbrace{\chi_w^2(\psi_j)}^{\text{Weak lensing}} + \underbrace{\chi^2_{\kappa(R)}(\psi_j) + \chi^2_{\gamma(R)}(\psi_j)}_{\text{Regularization}}$$

$$\frac{\partial \chi^2(\psi_j)}{\partial \psi_k} = 0$$

ALGORITHM

Outer level

Inner level

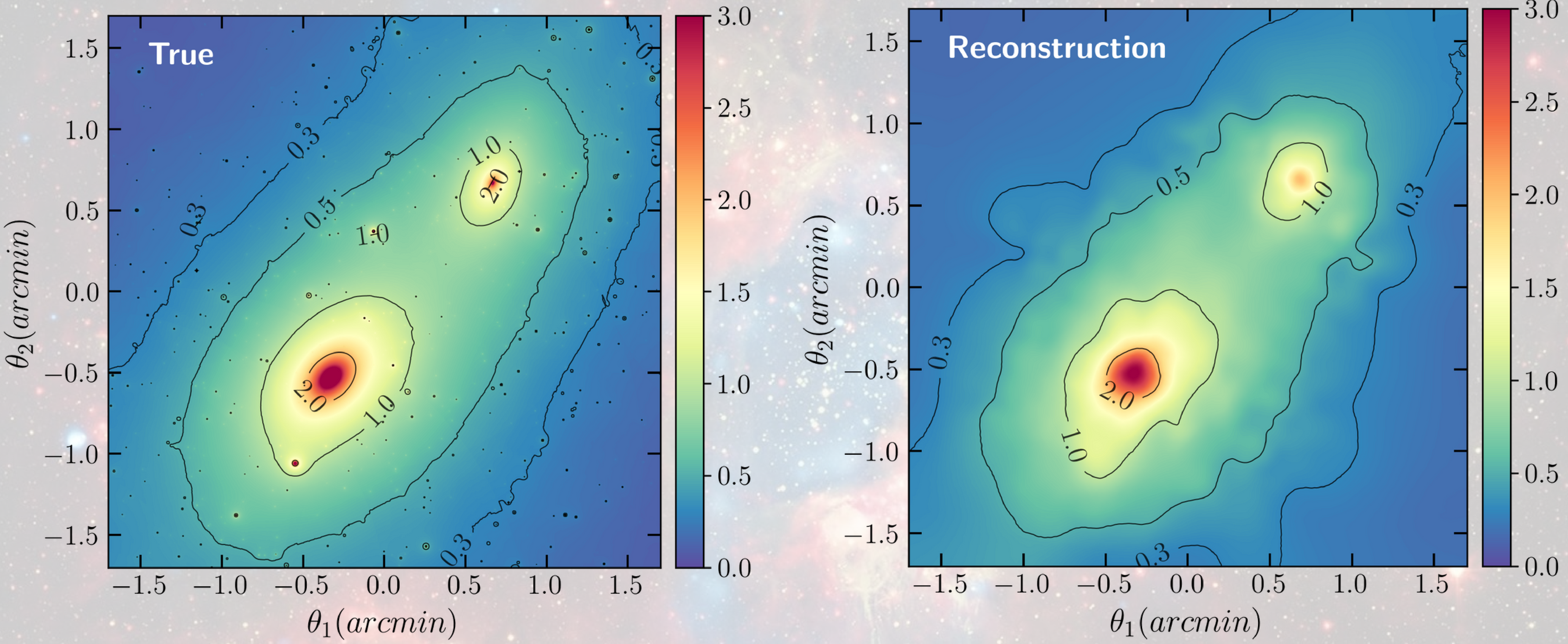
```
Input: Initial conditions and the initial guess ψ^(0).
Compute κ^(0), γ_1^(0) and γ_2^(0) from ψ^(0).
while outer_end = False do
    Compute the nodes weight.
    while inner_end = False do
        Compute the system of linear equations needed and solve it ⟶ ψ^(n).
        Smoothing of ψ^(n) ⟶ New ψ^(n) (optional).
        Compute κ^(n) from ψ^(n).
        if |κ_j^(n) − κ_j^(n−1)| ≤ tolerance then
            inner_end = True.
    Compute χ_s^2.
    if χ_s^2 ≤ q(2N_img) then
        outer_end = True.
    if outer_end = False then
        Apply grid refinement.
        Compute ψ^(m), κ^(m), γ_1^(m) and γ_2^(m) on the new grid.
return ψ
```

$$\chi^2(\psi_j) := \underbrace{\chi_s^2(\psi_j)}_{\text{Strong lensing}} + \overbrace{\chi_w^2(\psi_j)}^{\text{Weak lensing}} + \underbrace{\chi^2_{\kappa(R)}(\psi_j) + \chi^2_{\gamma(R)}(\psi_j)}_{\text{Regularization}}$$

$$\frac{\partial \chi^2(\psi_j)}{\partial \psi_k} = 0 \quad \Rightarrow \quad \sum_{j=1}^{N} \mathcal{W}_{kj}\psi_j = \mathcal{V}_k$$

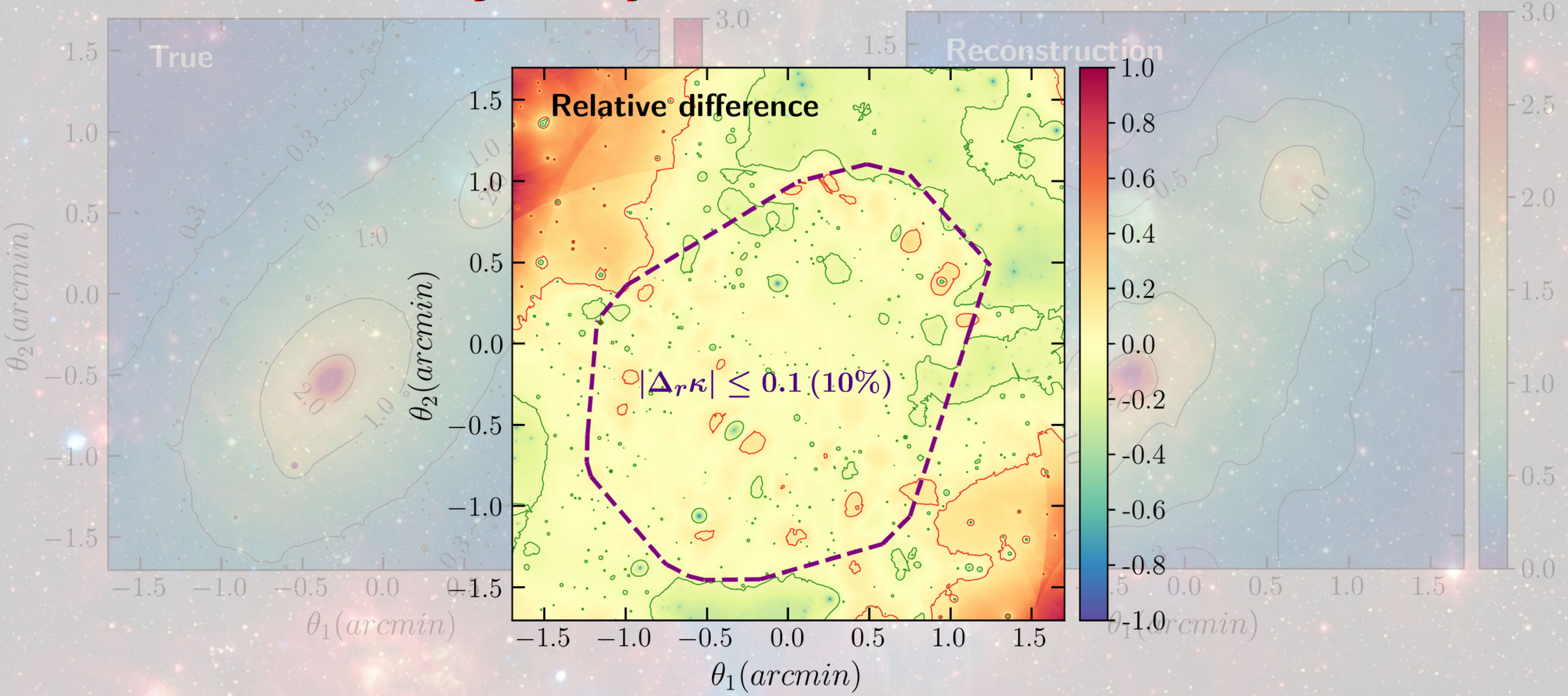
Leads to a system of linear equations!

RECONSTRUCTION (ARES)



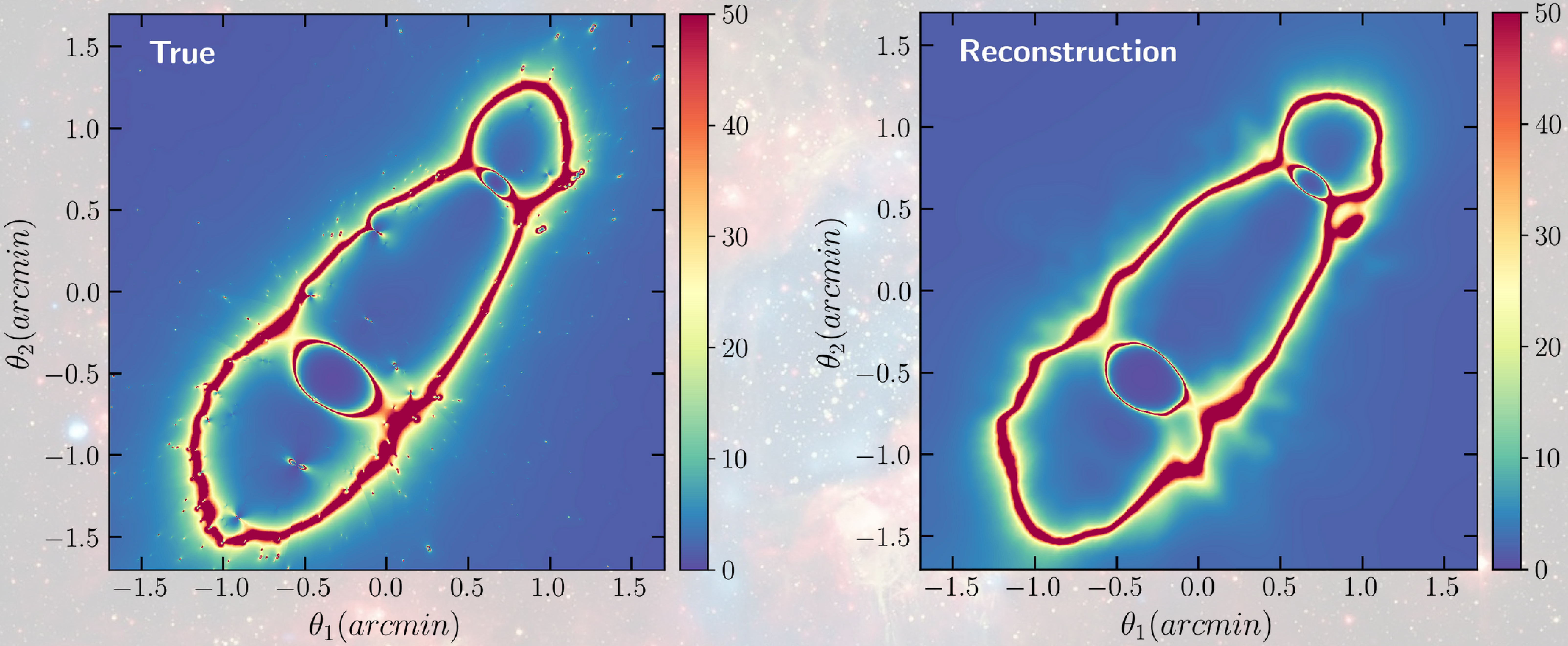
Convergence maps (κ) for a source with $z = 9$

RECONSTRUCTION (ARES)



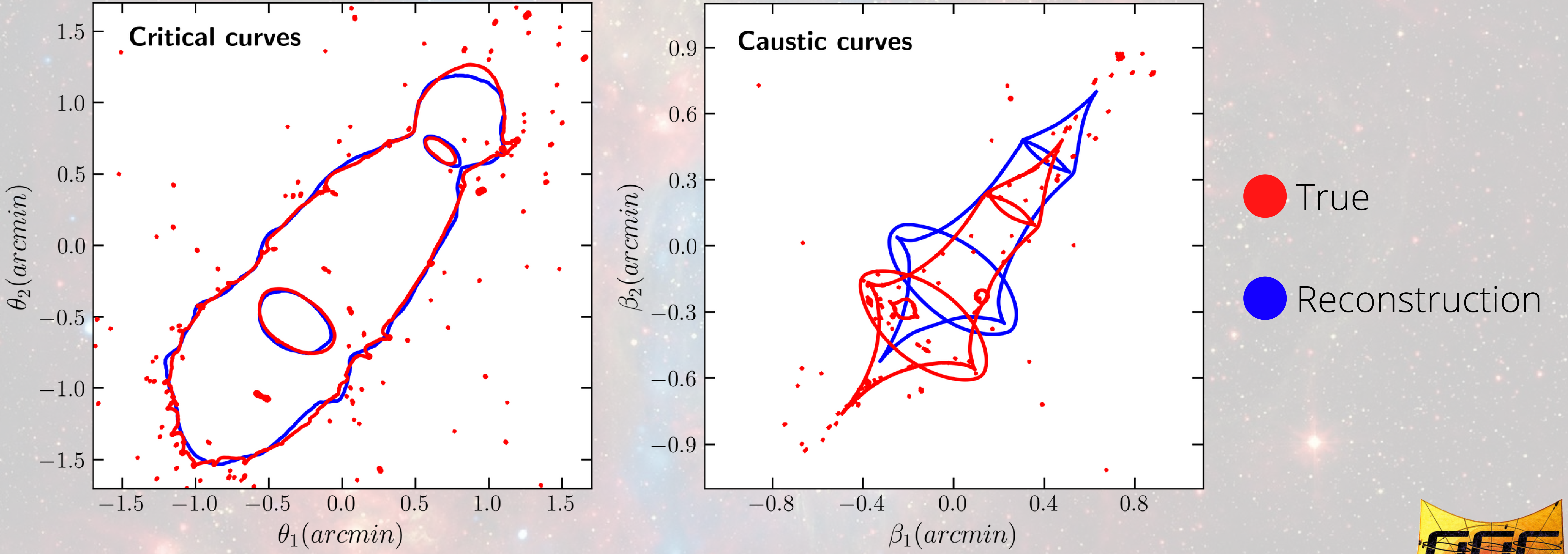
Convergence maps (κ) for a source with $z = 9$

RECONSTRUCTION (ARES)



Magnification maps ($|\mu|$) for a source with $z = 9$

RECONSTRUCTION (ARES)



relensing arXiv:2201.10076