

SN la scene modeling pipeline

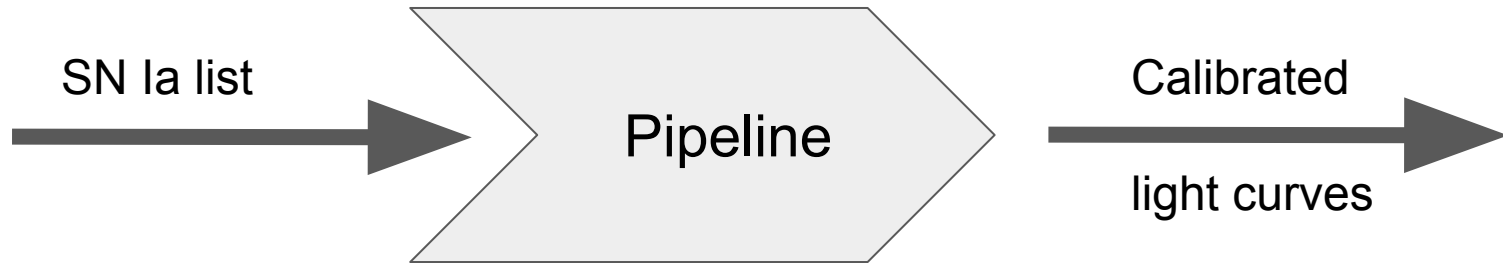
Leander Lacroix

ZTF-France meeting
09/12/2021

Phd advisor: Nicolas Regnault

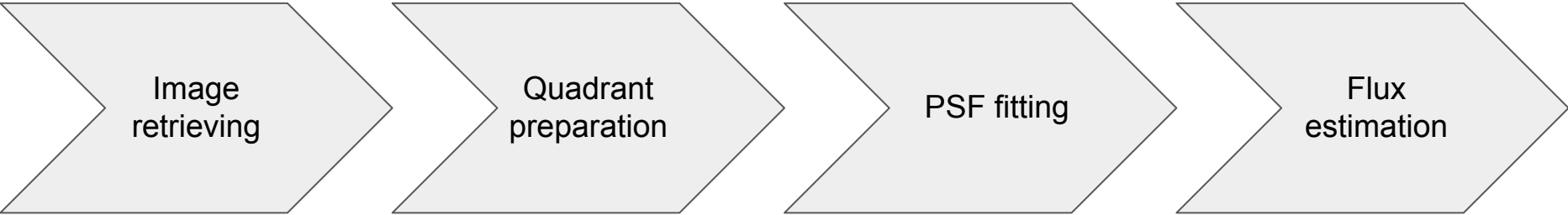
V0 SN Ia scene modeling pipeline

Goal: PoC by combining existing tools



First iteration to identify potential pitfalls, shortcomings and bottlenecks

Pipeline steps



```
graph LR; A[Image retrieving] --> B[Quadrant preparation]; B --> C[PSF fitting]; C --> D[Flux estimation];
```

Image
retrieving

Quadrant
preparation

PSF fitting

Flux
estimation

Step 1: Image retrieving

For each SN Ia, define relevant time interval

- Done using Salt2 model fits

Download:

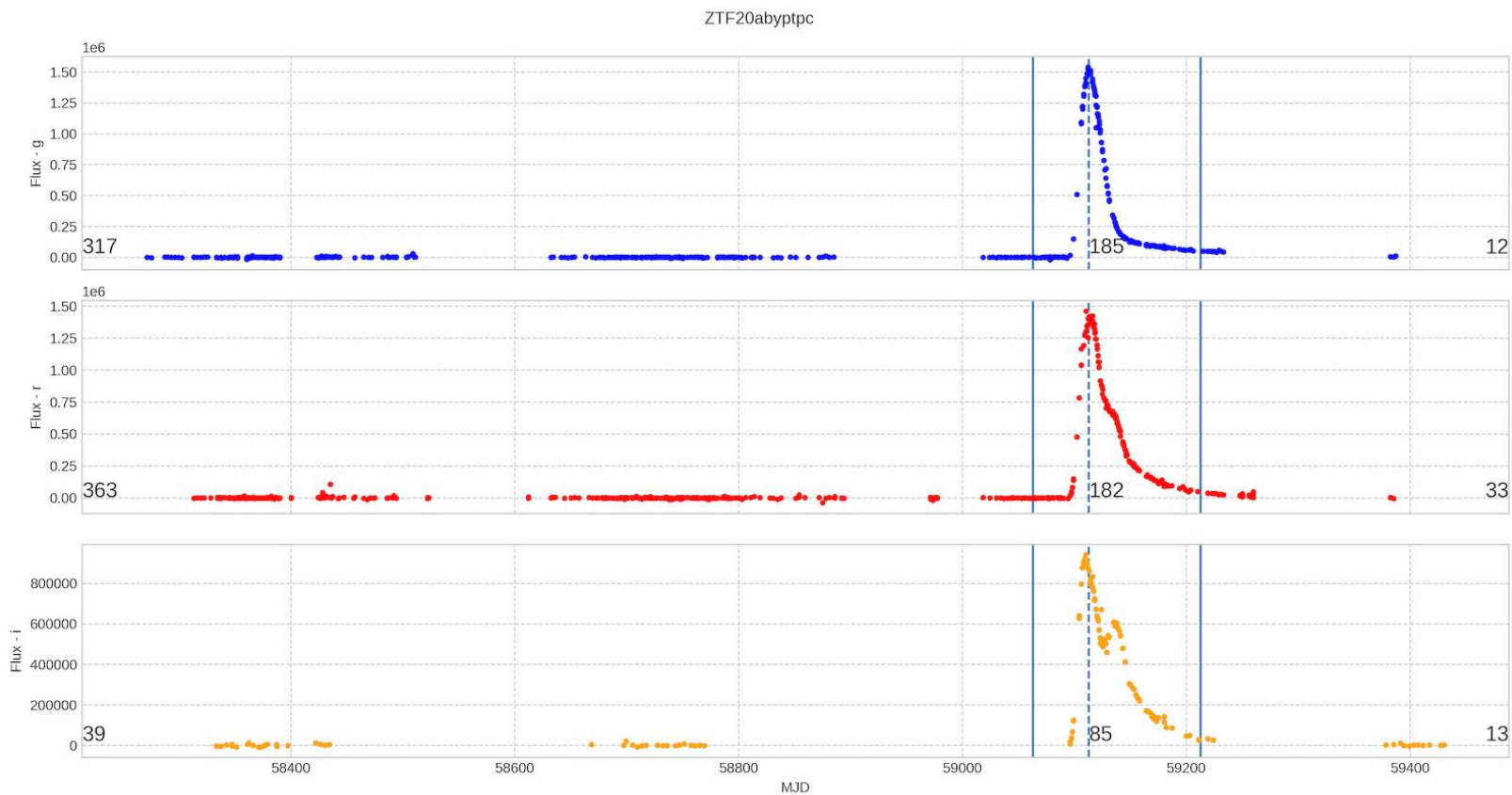
- relevant quadrants (science + mask) at each epoch (~28 GB/sn)

Identify, for each quadrant:

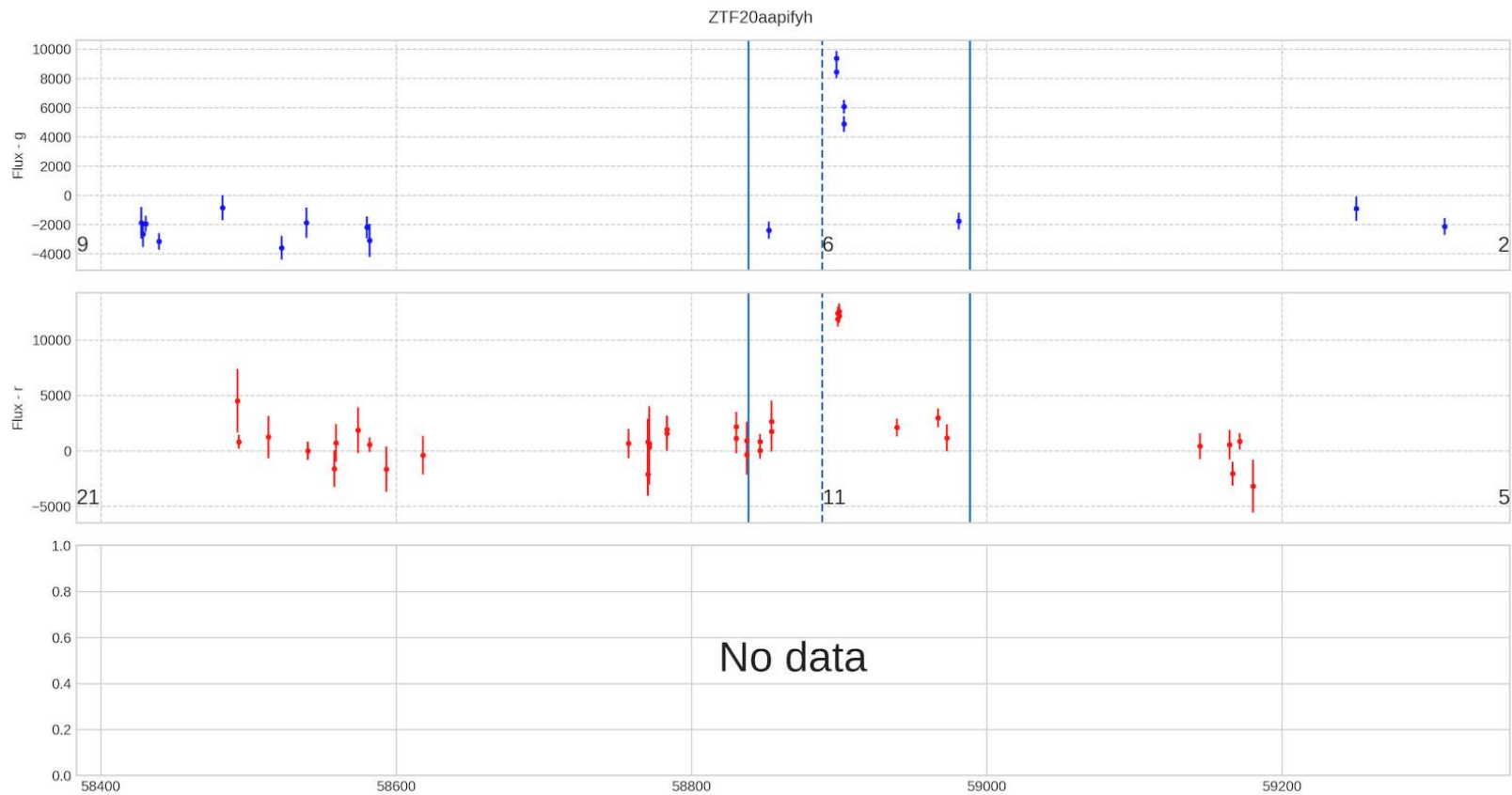
- Gaia callibrators
- Host galaxy

Tools : ztfquery, ztfimg, cosmoidr2, fgallery

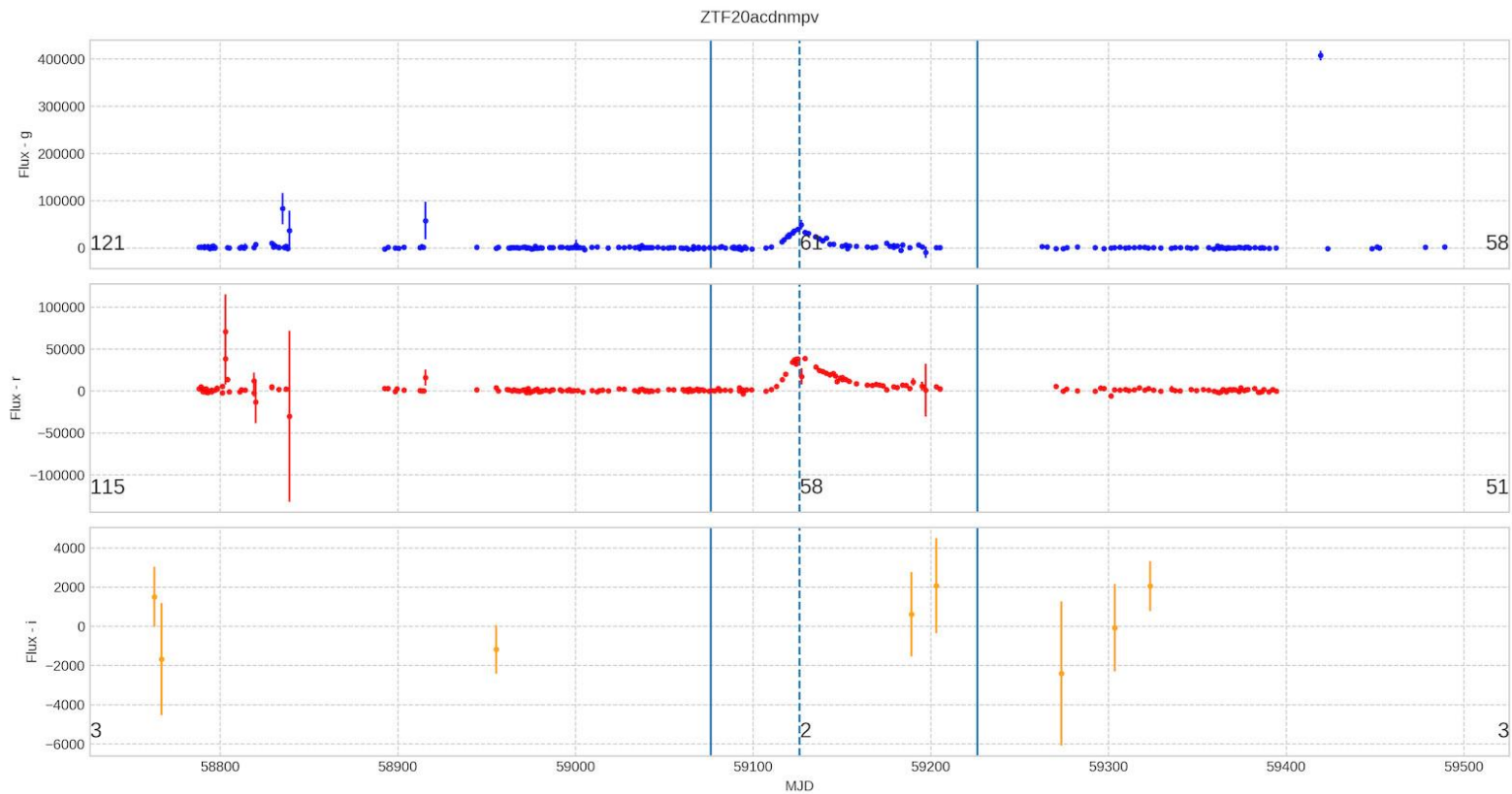
Step 1 bis: selecting relevant SN 1a



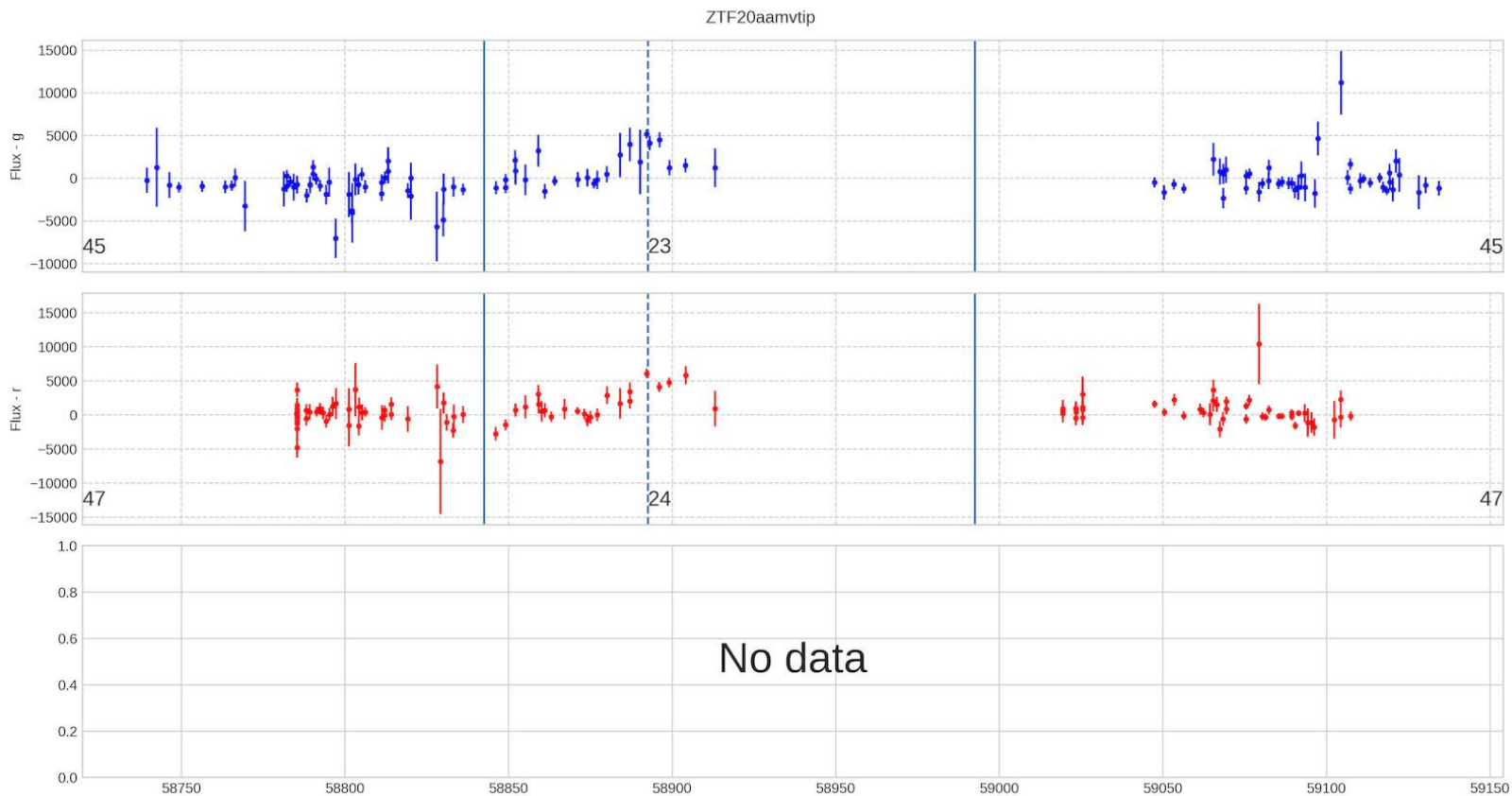
Step 1 bis: selecting relevant SN 1a



Step 1 bis: selecting relevant SN 1a



Step 1 bis: selecting relevant SN 1a



Step 2: Quadrant preparation

Weight map (combination of mask, hot pixels, cosmics)

Sky background model

Aperture photometry for each object + statistical moments

Define reference quadrant (best seeing)

Tools: poloka-core (inhouse SNLS legacy pipeline), astropy

Step 3: PSF fitting

Fit PSF on Gaia stars for each quadrant

PSF model (function of the position):

- Analytical part
- Empirical part (residuals, ...)

Requires ~100 stars/quadrant (we have ~2000)

Tool : poloka-psf (inhouse SNLS legacy software)

Step 4: Flux estimation - scene modeling

Estimate SN Ia flux at each epoch

- Scene modeling lightcurve of nearby stars
- Absolute calibration on Ubercal/Calspec

Method:

- Maximum likelihood model for SN 1a + galaxy

$$M_{ij} = f_i \psi_i(x_j) + K_i * G(x_j)$$

Tool: poloka-simphot (inhouse SNLS legacy software)

Roadmap and potential pitfalls

Roadmap:

- By January: pipeline on ~20 SN
- By March: Streamlining on whole dataset

Potential pitfalls:

- From precedent studies/projects: 30 min/SN/band
- Might benefit from a GPU implementation (for model evaluation)
- Other optimisation scheme ?
 - We know we start close to the solution