



SPACIOUS **science cases**

**ADAPTED FROM:
code-to-data session
DPAC consortium meeting**

Alfred Castro Ginard

Context

Exploit code-to-data analysis (SPACIOUS) for the PVPs:

- Star Clusters: detection, dynamics, and stellar variability.
- The Bottom of the Main Sequence.

} Both related to challenge 2: clustering

The Gaia DR3 Selection Function: related to Challenge 4: Gaia enhanced products

Limitations:

- Very tight schedule
- SPACIOUS deployments:

• Scorpio/capricorn	←	high CPU/memory	✗
• Aquarius	←	main hub	🔨
• Google Cloud	←	high Flexibility	
• Sagittarius	←	approved data partner	✓
		3 nodes, 32 vCPU, 110 Gb	

Star Clusters: detection, dynamics, and stellar variability

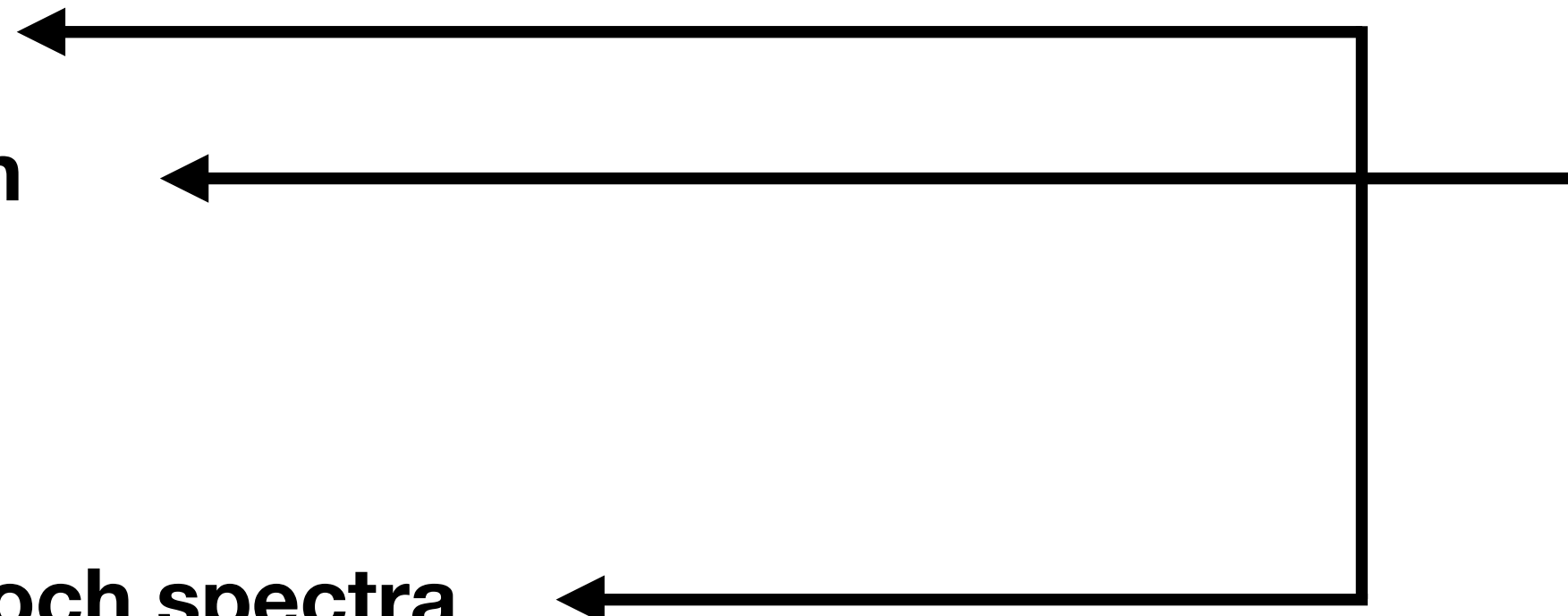
Goal: to characterise the star cluster population in Gaia DR4

focusing on improvements wrt DR3 and new science opportunities

Description of tasks (show improvement on):

- **Detection of new clusters**
- **Cluster membership determination**
- **Cluster internal kinematics**
- **Cluster population kinematics**
- **Signs of stellar variability in XP epoch spectra**

Big Data



Star Clusters: detection, dynamics, and stellar variability

Detection of new clusters

Explore how improved DR4 astrometry pushes the cluster detection threshold

- Task on SPACIOUS: clustering in 5D astrometry

1. Tile the sky, e.g., HEALPix

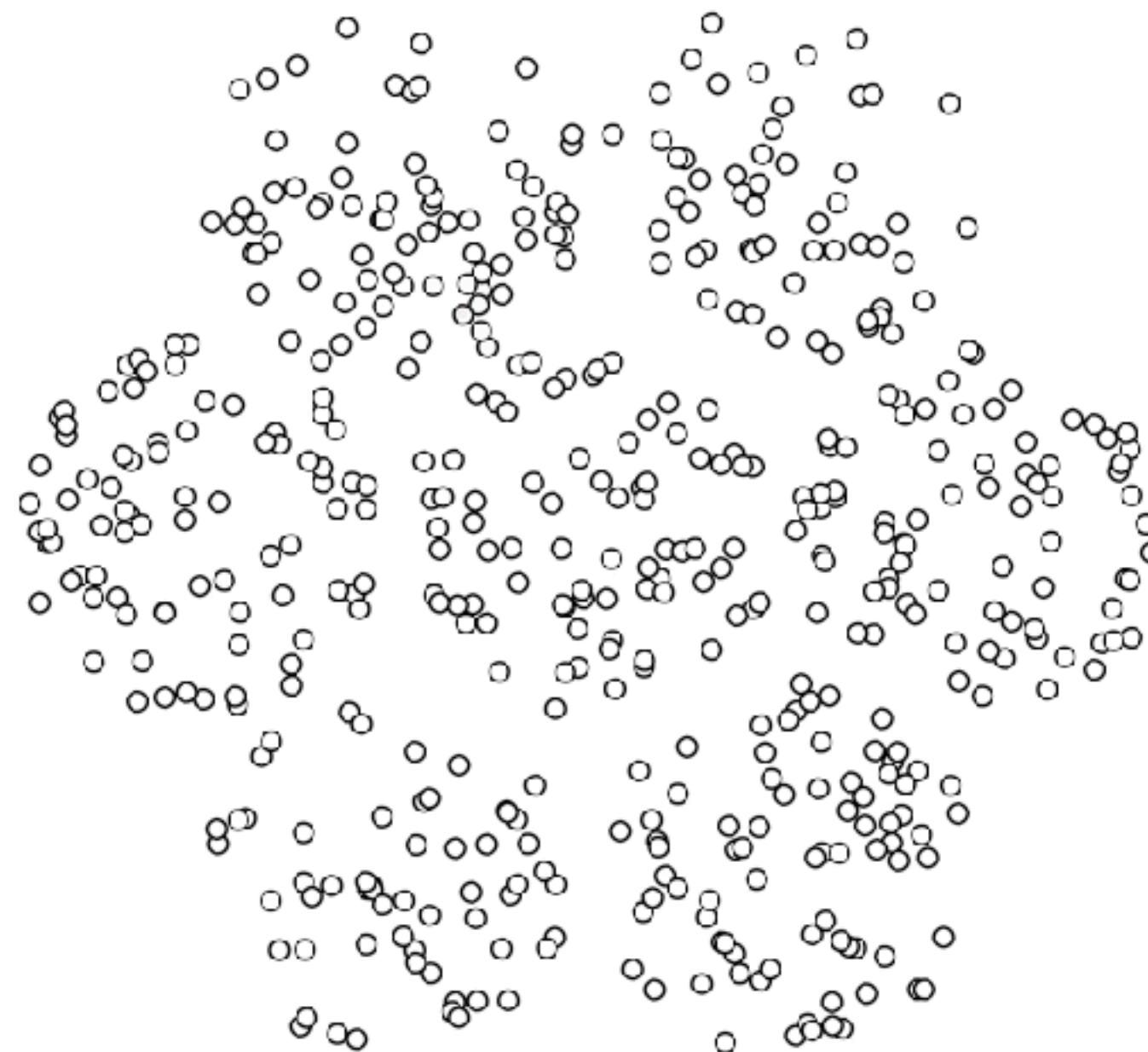
2. Clustering algorithm: DBSCAN, HDBSCAN, etc. Memory cost $O(n^2)$

3. Merge tiles + borders

- 3.1. Data partition borders

HEADSS

[\[Clarke, Hambly, Mann 2023\]](#)



Star Clusters: detection, dynamics, and stellar variability

Detection of new clusters

Explore how improved DR4 astrometry pushes the cluster detection threshold

- Task on SPACIOUS: clustering in 5D astrometry

1. Tile the sky, e.g., HEALPix

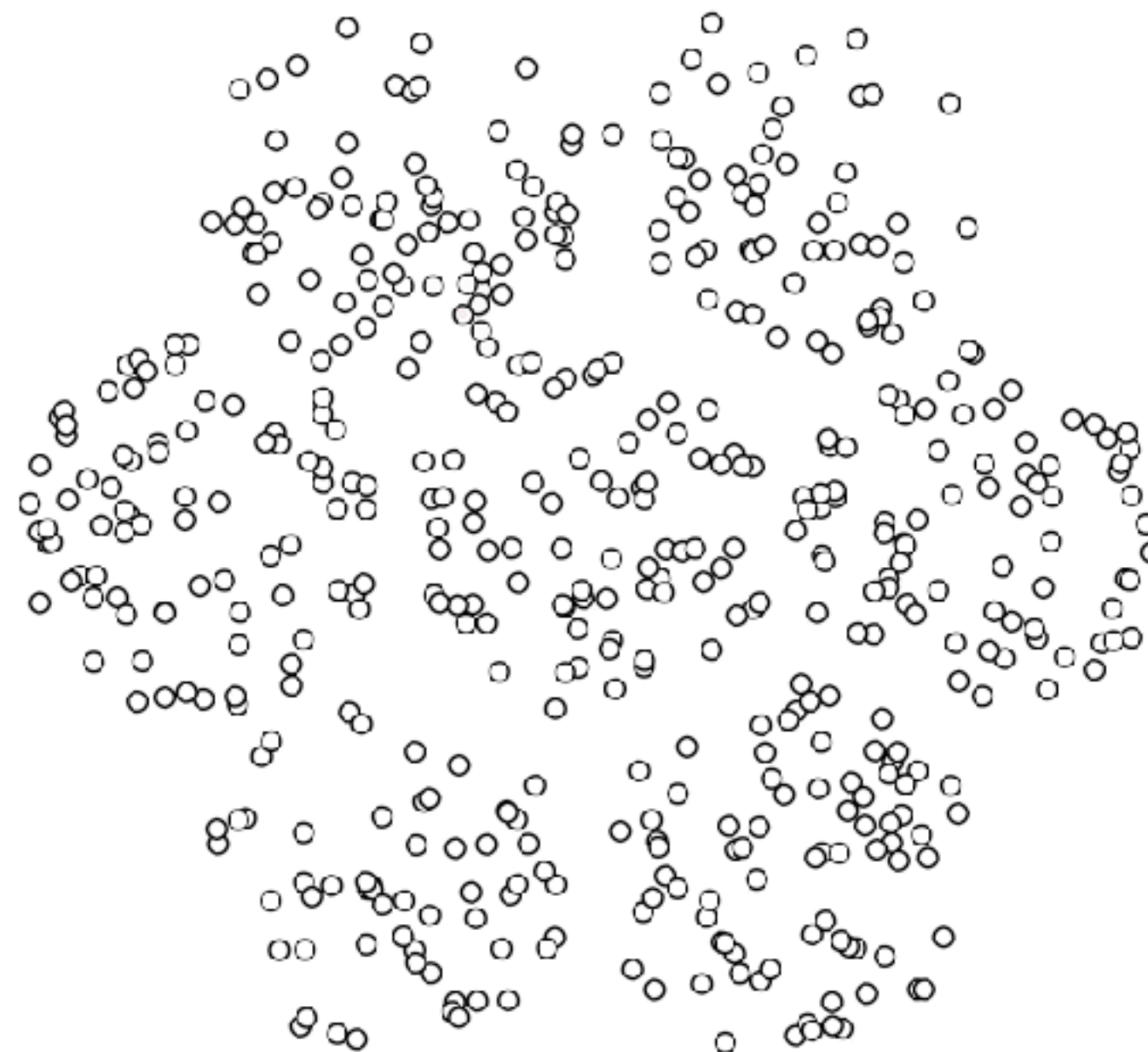
2. Clustering algorithm: DBSCAN, HDBSCAN, etc. Memory cost $O(n^2)$

3. Merge tiles + borders

- 3.1. Data partition borders

HEADSS

[\[Clarke, Hambly, Mann 2023\]](#)



The Bottom of the Main Sequence (TBS)

Goal: to characterise the very faint end of the Main Sequence (Gaia detectability limits)

Description of tasks (selected):

- **Generation of the TBS catalogue based on astrophysical properties**
- **Estimation of that catalogue's selection function**
- **Identification of stars in TBS as members of known stellar groupings**

The Bottom of the Main Sequence (TBS)

Goal: to characterise the very faint end of the Main Sequence (Gaia detectability limits)

Description of tasks (selected):

- Identification of stars in TBS as members of known stellar groupings
- Tasks on SPACIOUS:
 1. Membership probabilities (BANYAN Sigma, [\[Gagné&Faherty2018\]](#))
 2. Each star classified in one of 8125 moving groups or associations (within 500 pc)
 - 2.1. Based on position, proper motions, distance posteriors, and uncertainties (+ RV)
 - 2.2. 6D multivariate Gaussian (Galactic 6D space)

The Gaia catalogue selection function

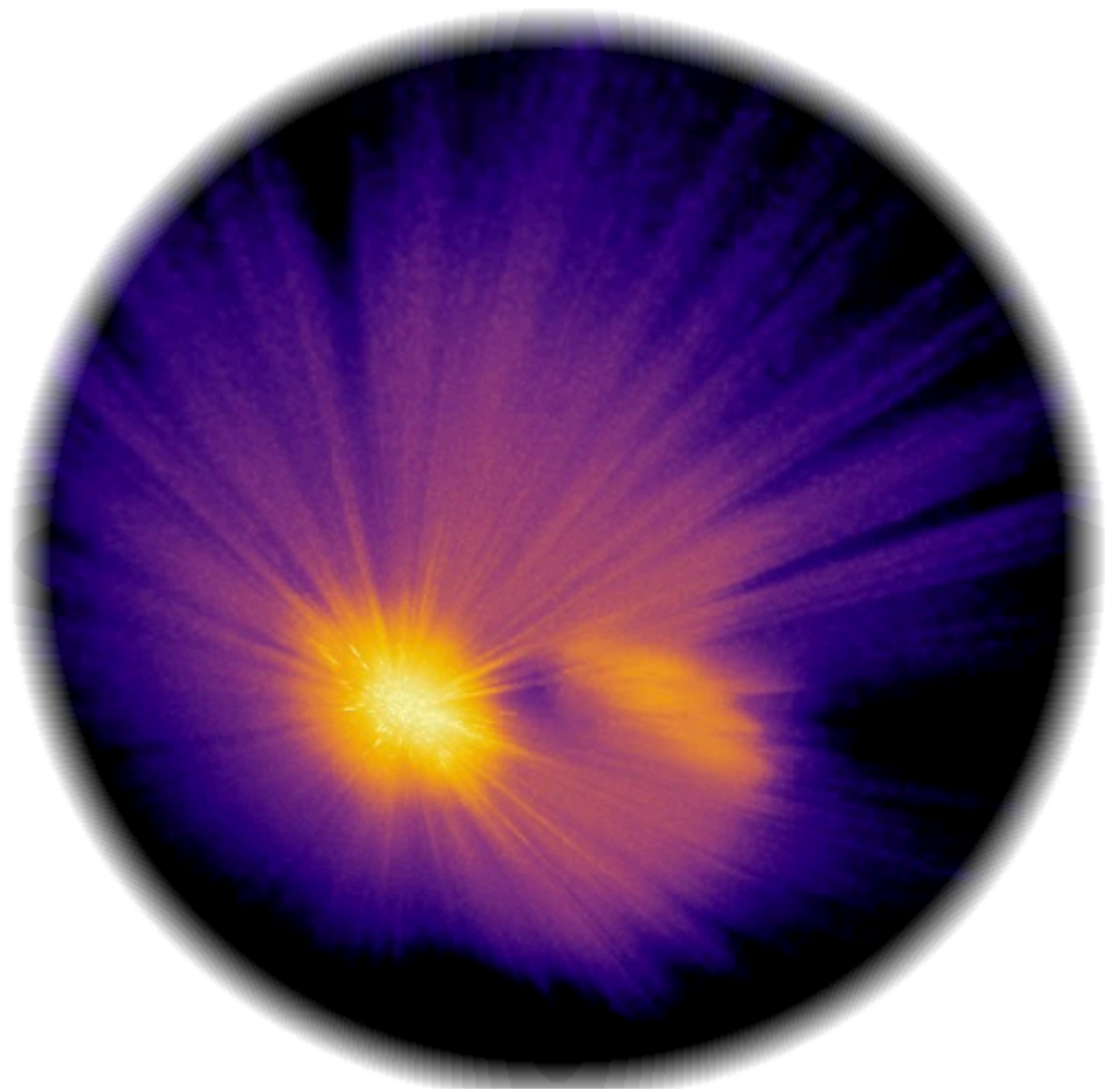
Goal: to characterise the very faint end of the Main Sequence (Gaia detectability limits)

Description of tasks (selected):

- **Estimation of that catalogue's selection function**
- **Tasks on SPACIOUS**
 - 1. Everything**

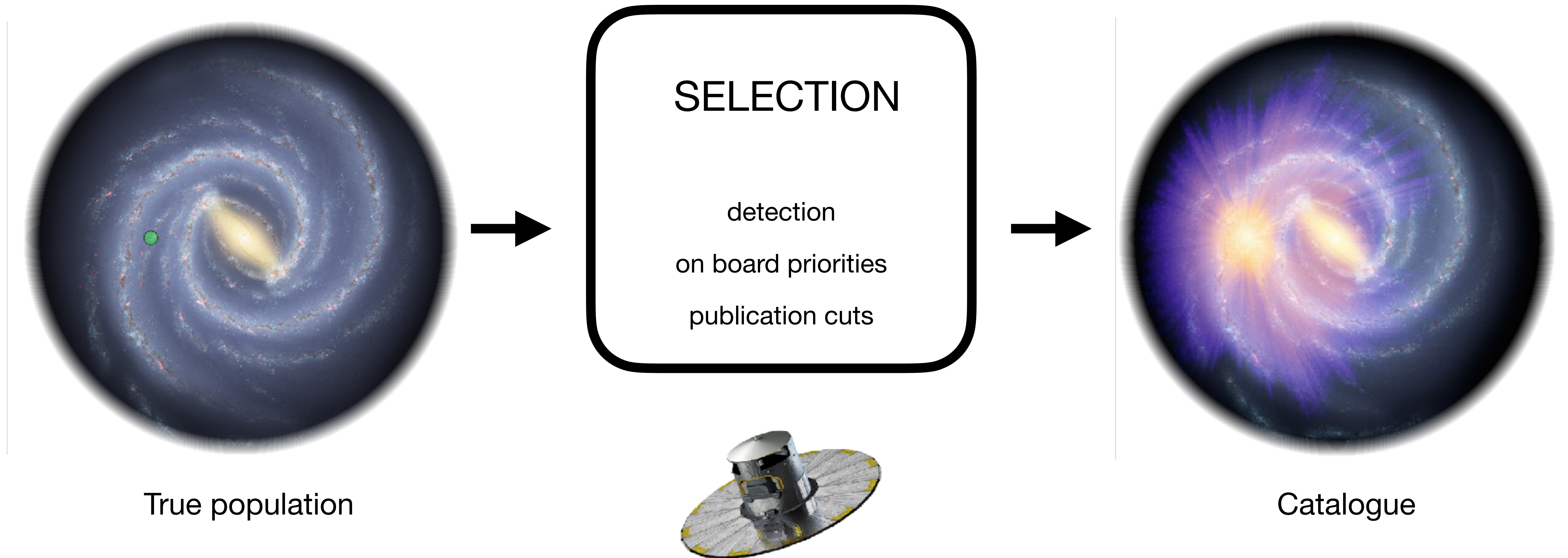
What is a selection function?

$$S = P(\text{source enters the catalogue} \mid \text{position, magnitude, colour, ...})$$



What is a selection function?

$$S = P(\text{source enters the catalogue} \mid \text{position, magnitude, colour, ...})$$



The Gaia catalogue selection function

Goal: to characterise the Gaia source catalogue selection function and completeness

Based on the Gaia DR3 selection function [\[Castro-Ginard+,in prep.\]](#)

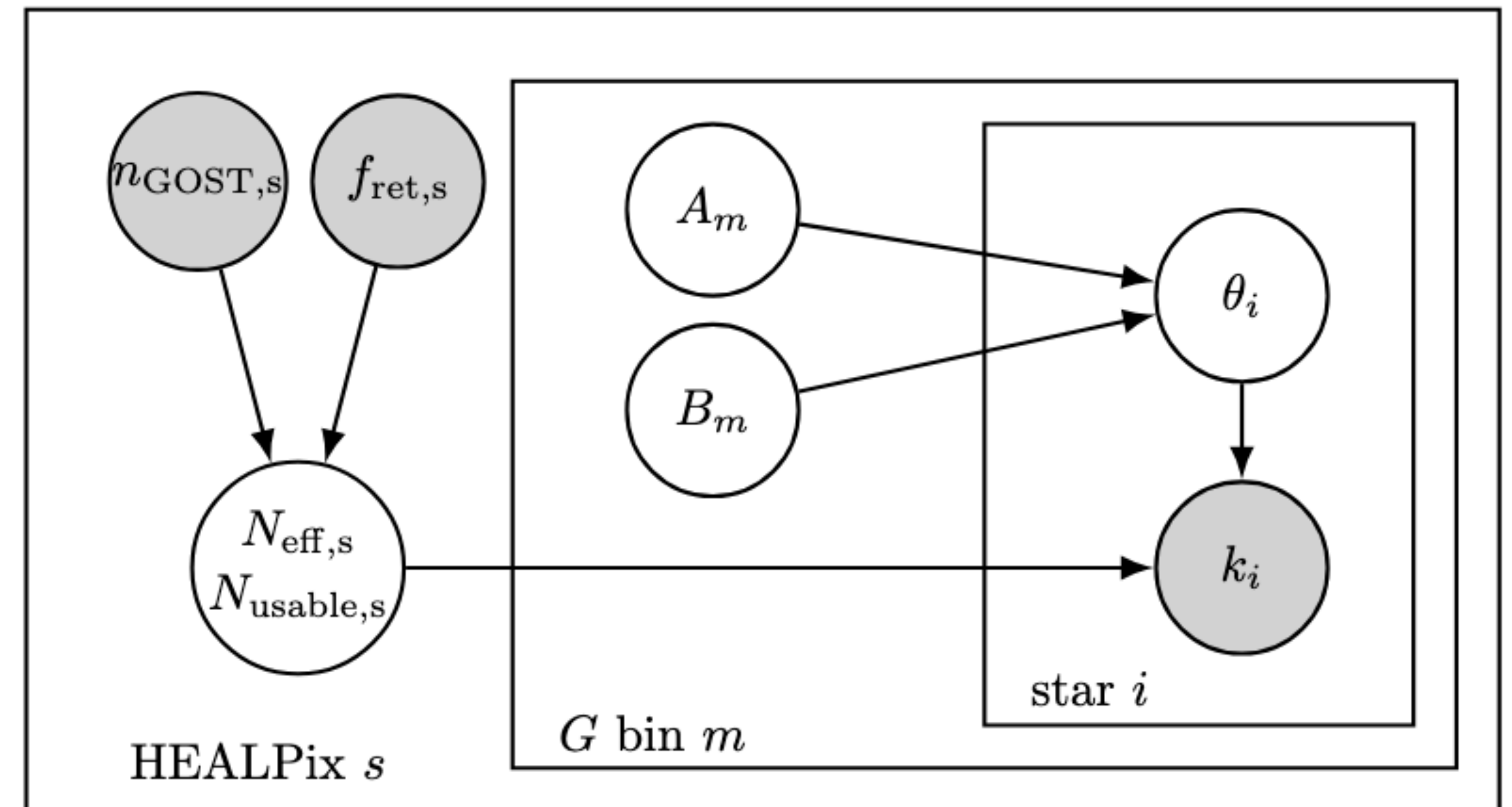
Lower resolution version for Gaia DR4's PVP

Anchoring data:

- GOST predictions of the number of scans
- Data-taking gaps
- Astrometric matched transits

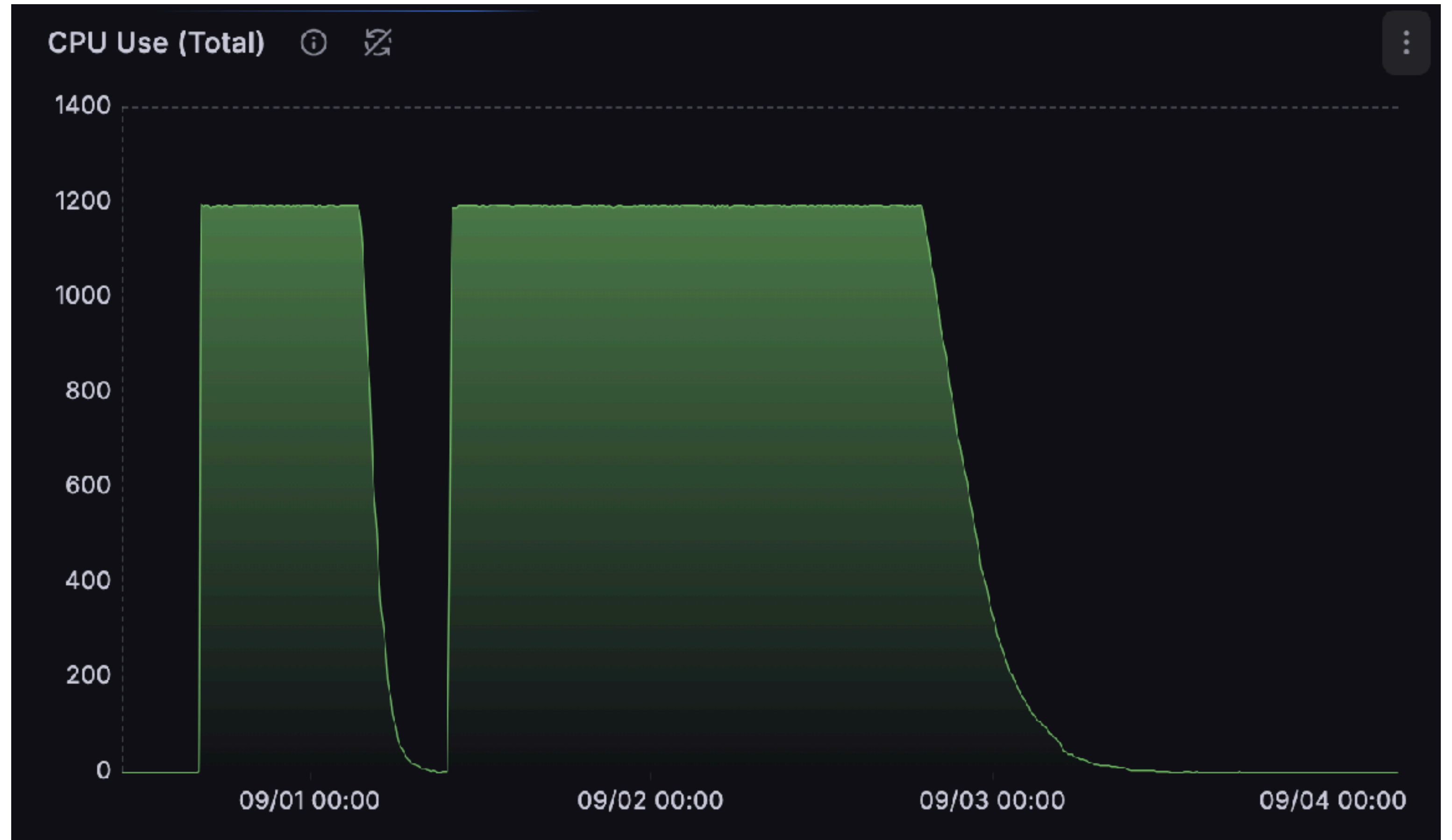
Estimated quantities (MCMC):

- Effective number of scans
- Detection probability



The Gaia catalogue selection function

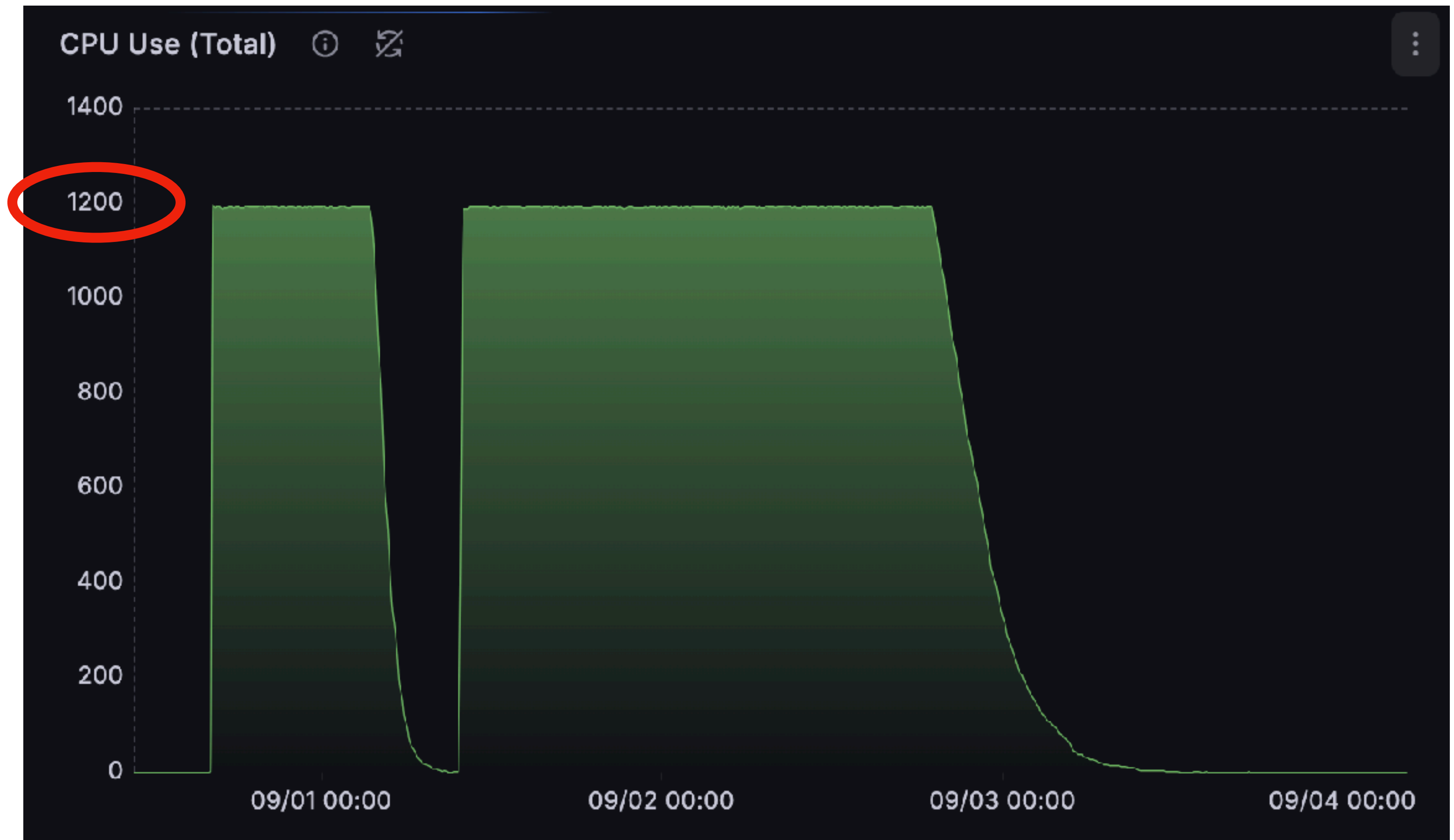
Run in SPACIOUS@GCloud



The Gaia catalogue selection function

Run in SPACIOUS@GCloud

Very high availability and stability

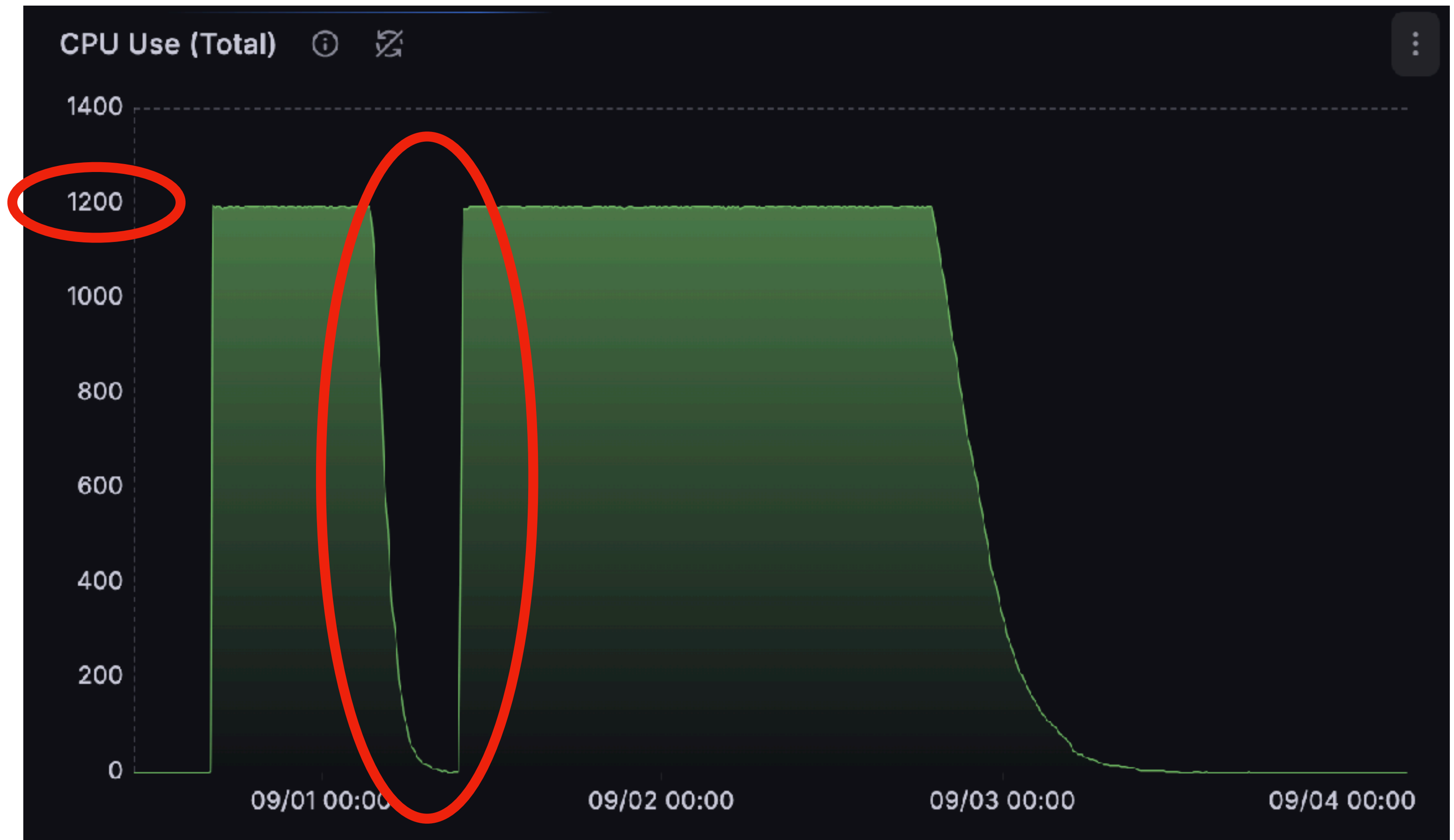


The Gaia catalogue selection function

Run in SPACIOUS@GCloud

Very high availability and stability

Checkpointing



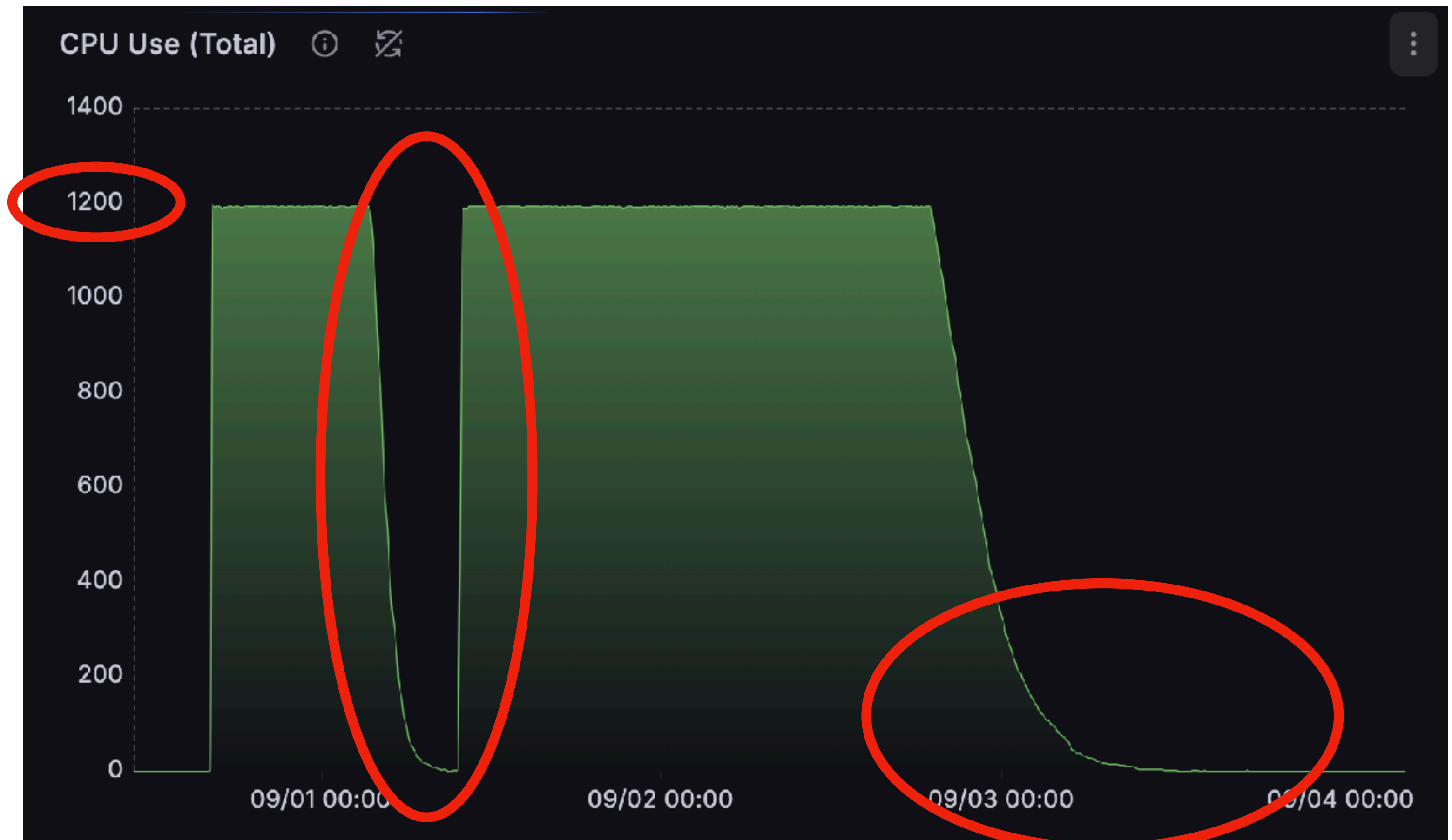
The Gaia catalogue selection function

Run in SPACIOUS@GCloud

Very high availability and stability

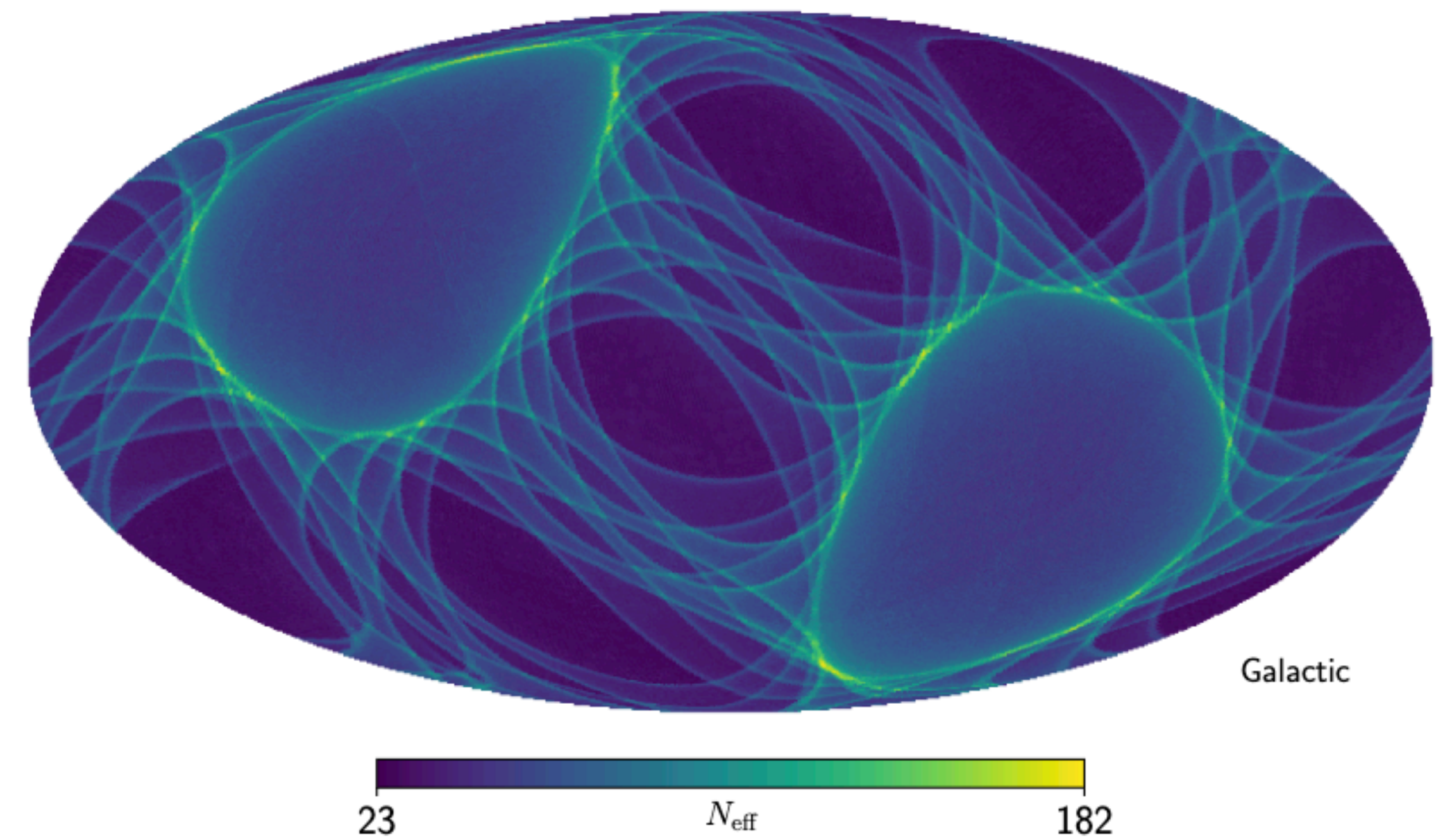
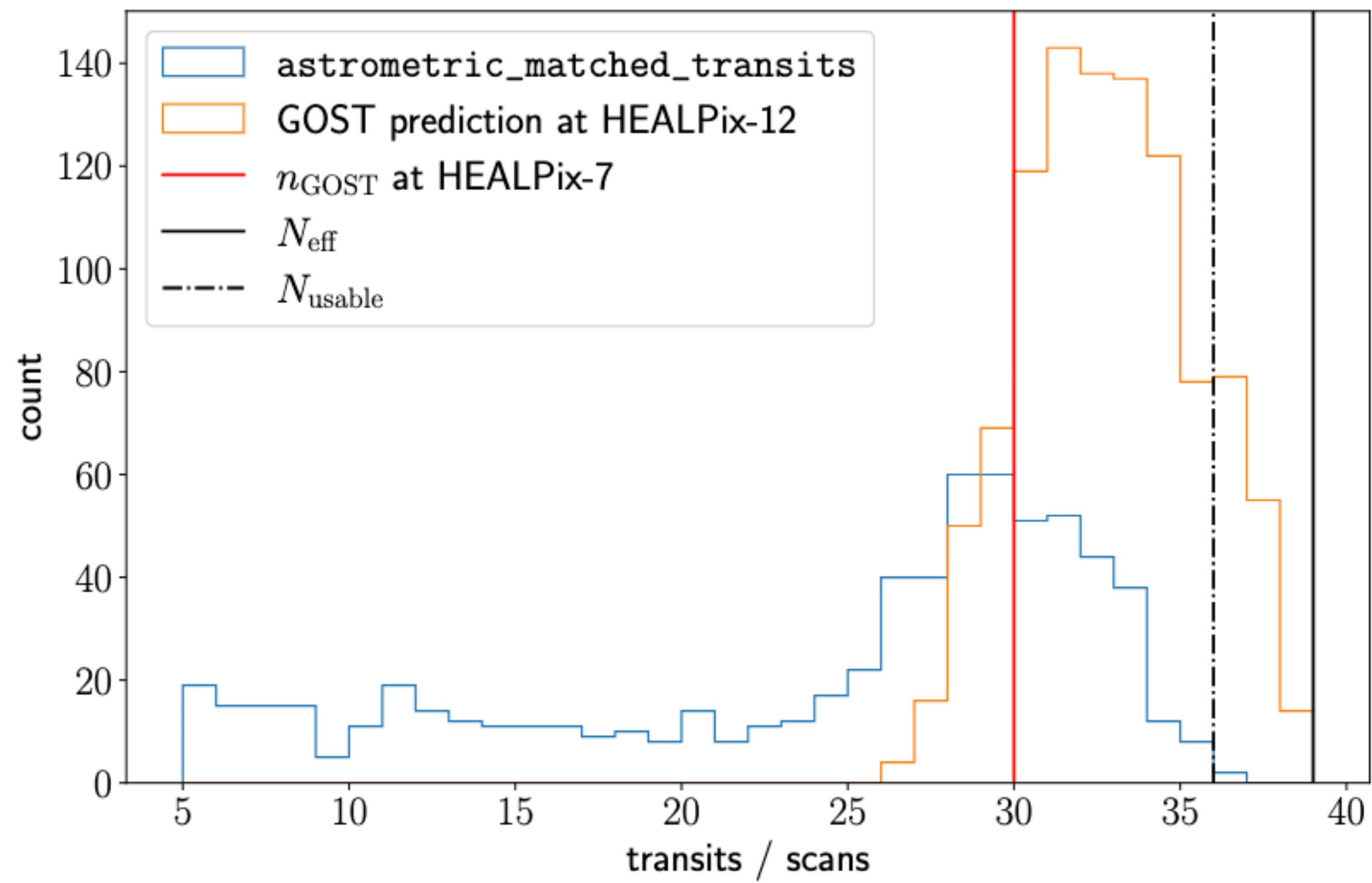
Checkpointing

Run “dominated” by long tasks



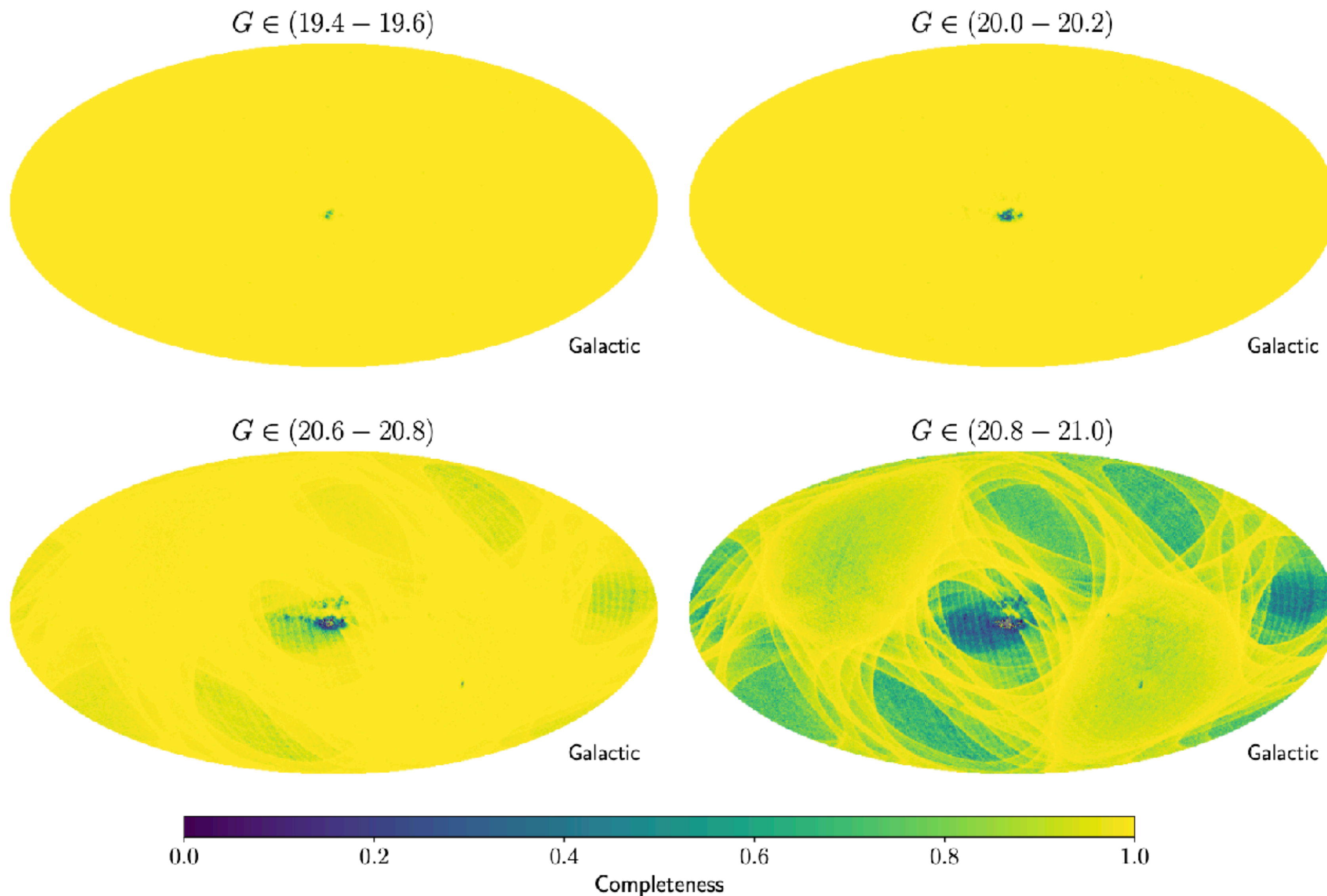
The Gaia catalogue selection function

Importance to have the joint model (enabled by SPACIOUS resources)



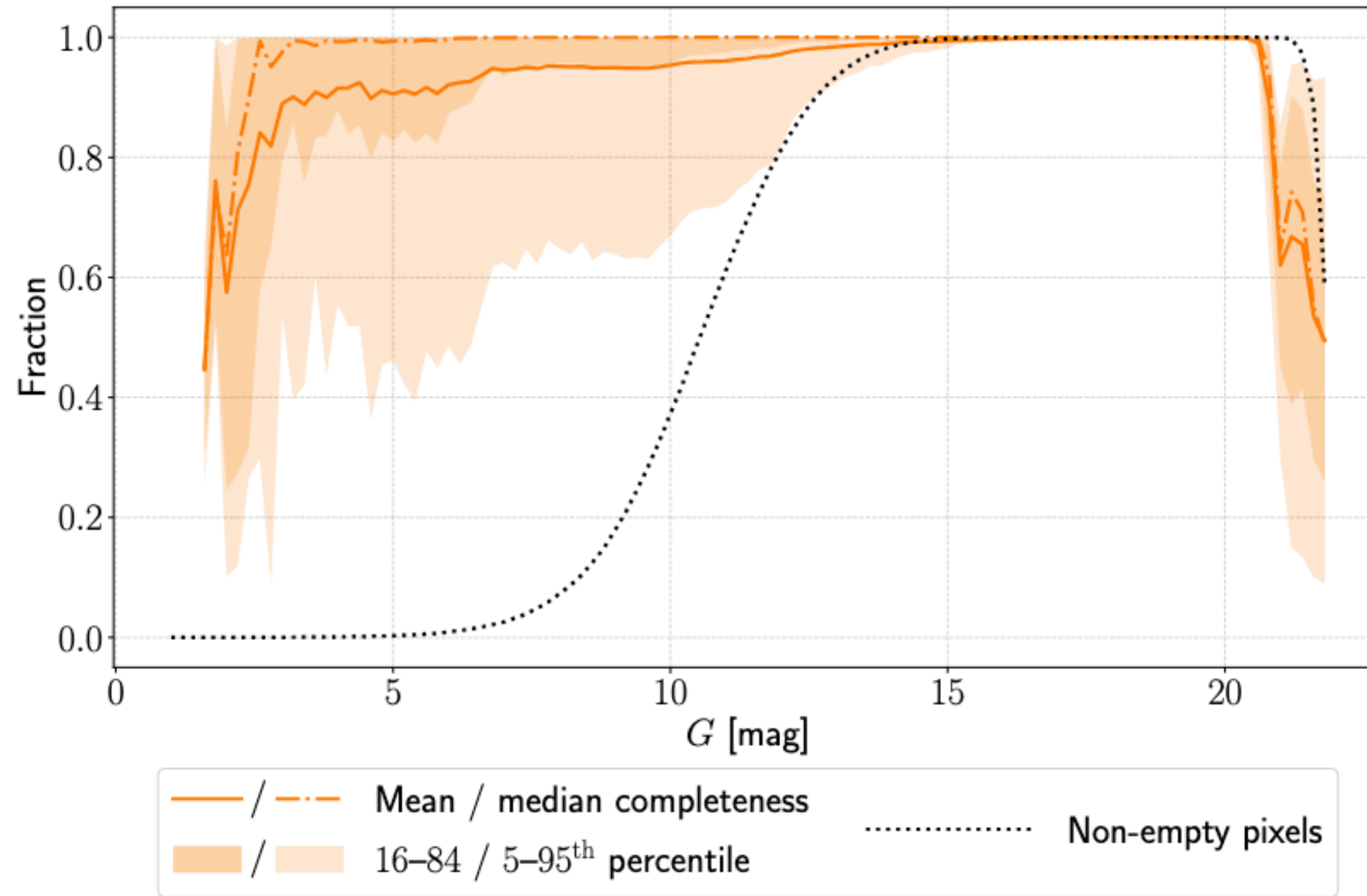
The Gaia catalogue selection function

DR3



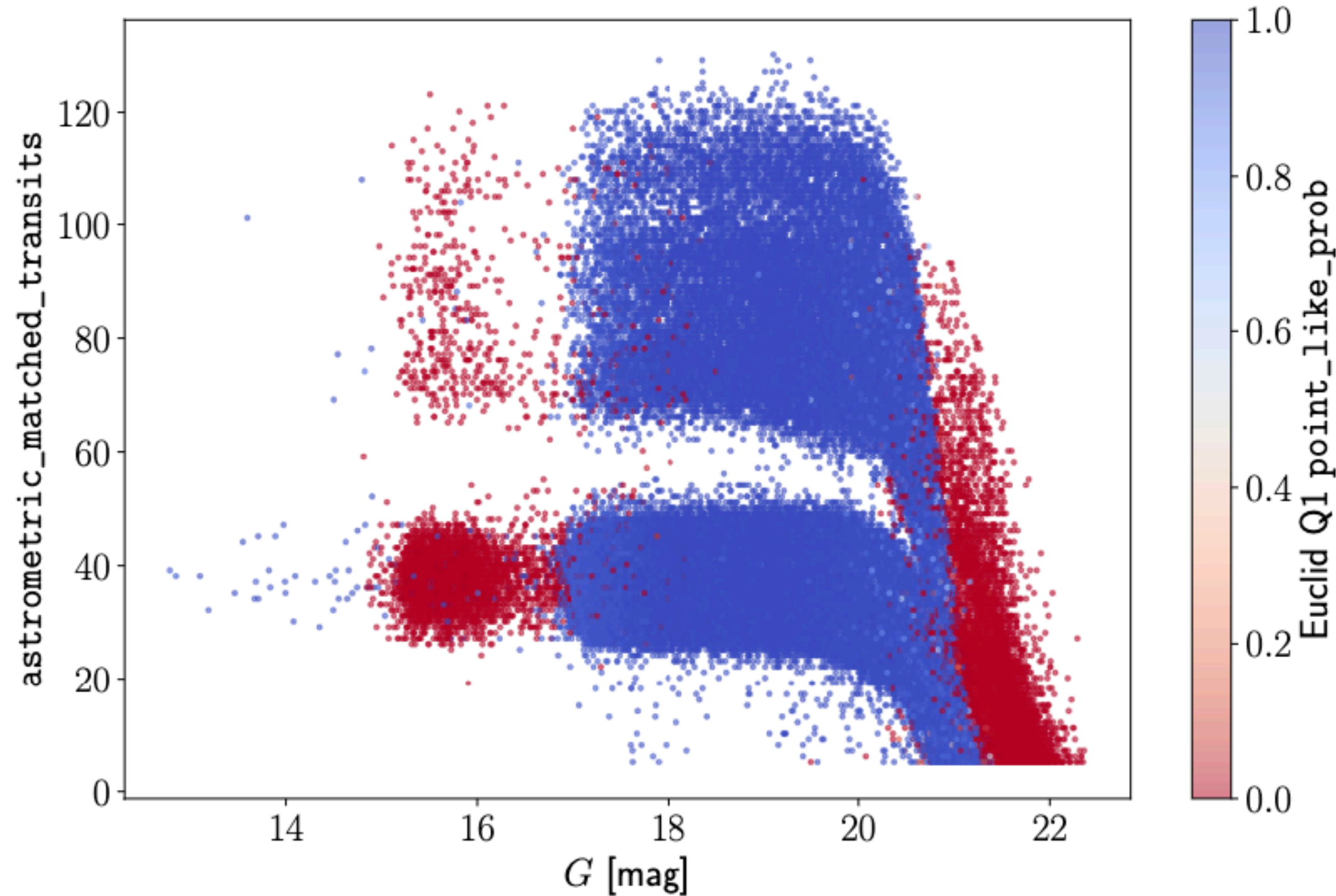
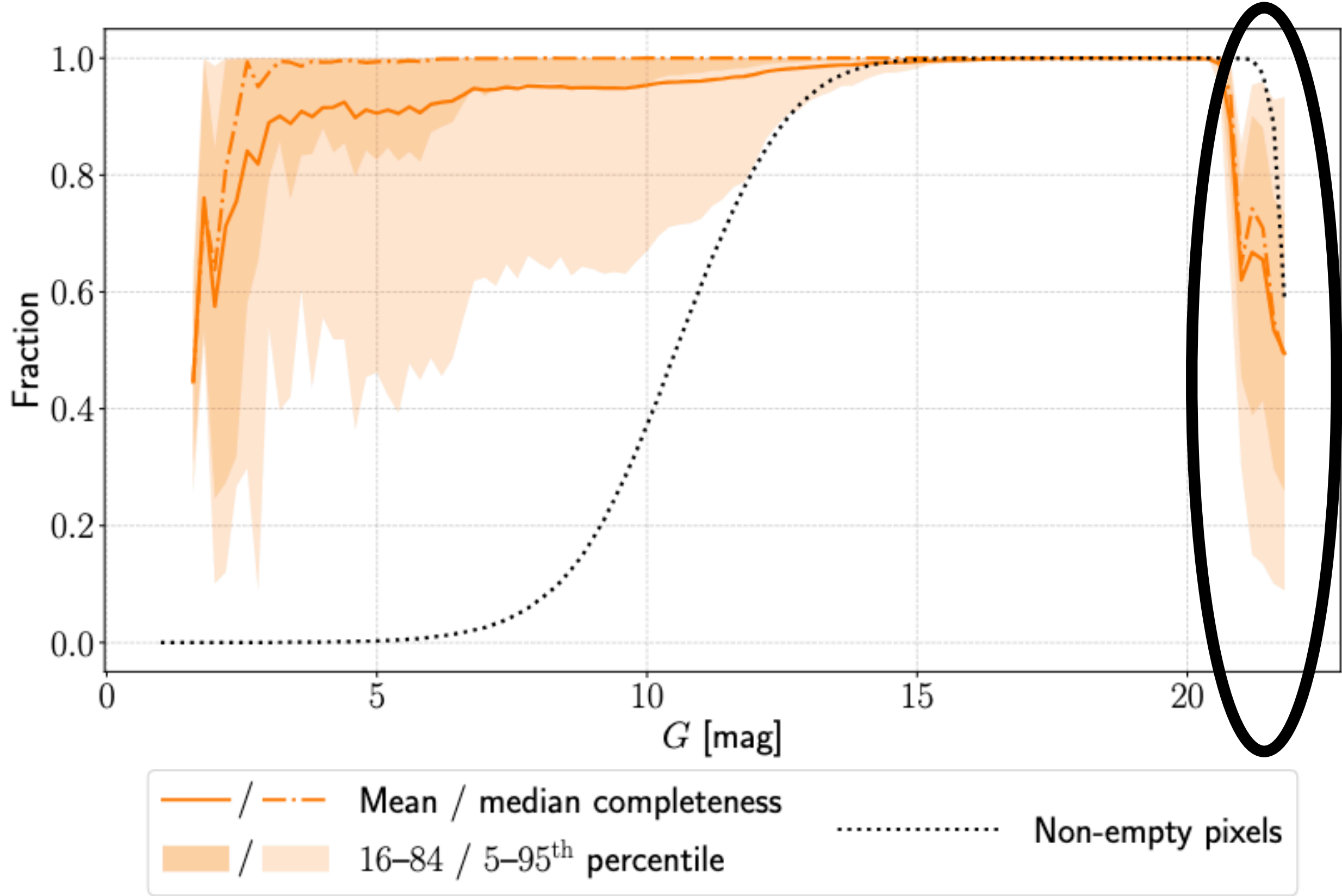
The Gaia catalogue selection function

DR3



The Gaia catalogue selection function

DR3



The Gaia catalogue selection function

Performance on SPACIOUS

Goal: to benchmark against traditional HPC cluster

Test:

- take 60 bins (different stellar densities: bulge, inner disc, anticentre, mid-latitude b, Galactic poles)
- Distribute over 16 parallel tasks
 - HPC: 16 cores in python multiprocessing
 - SPACIOUS: 8 pods, 2 cores/pots = 16 concurrent python tasks

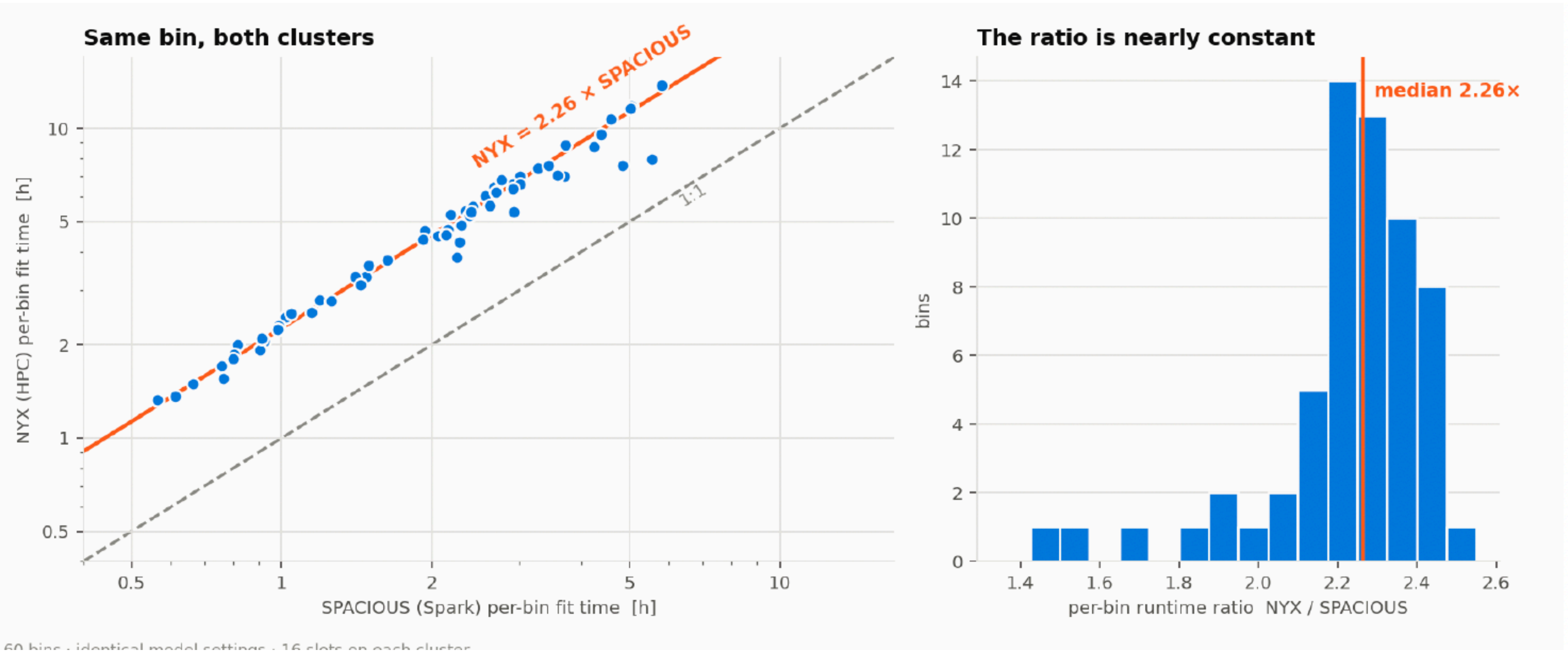
Bottom line:

SPACIOUS completed the same 60 pixels in 11.9 hours, against HPC 22.9 hours

SPACIOUS run individual pixels about x2.26 faster

The Gaia catalogue selection function

Performance on SPACIOUS



The Gaia catalogue selection function

Performance on SPACIOUS

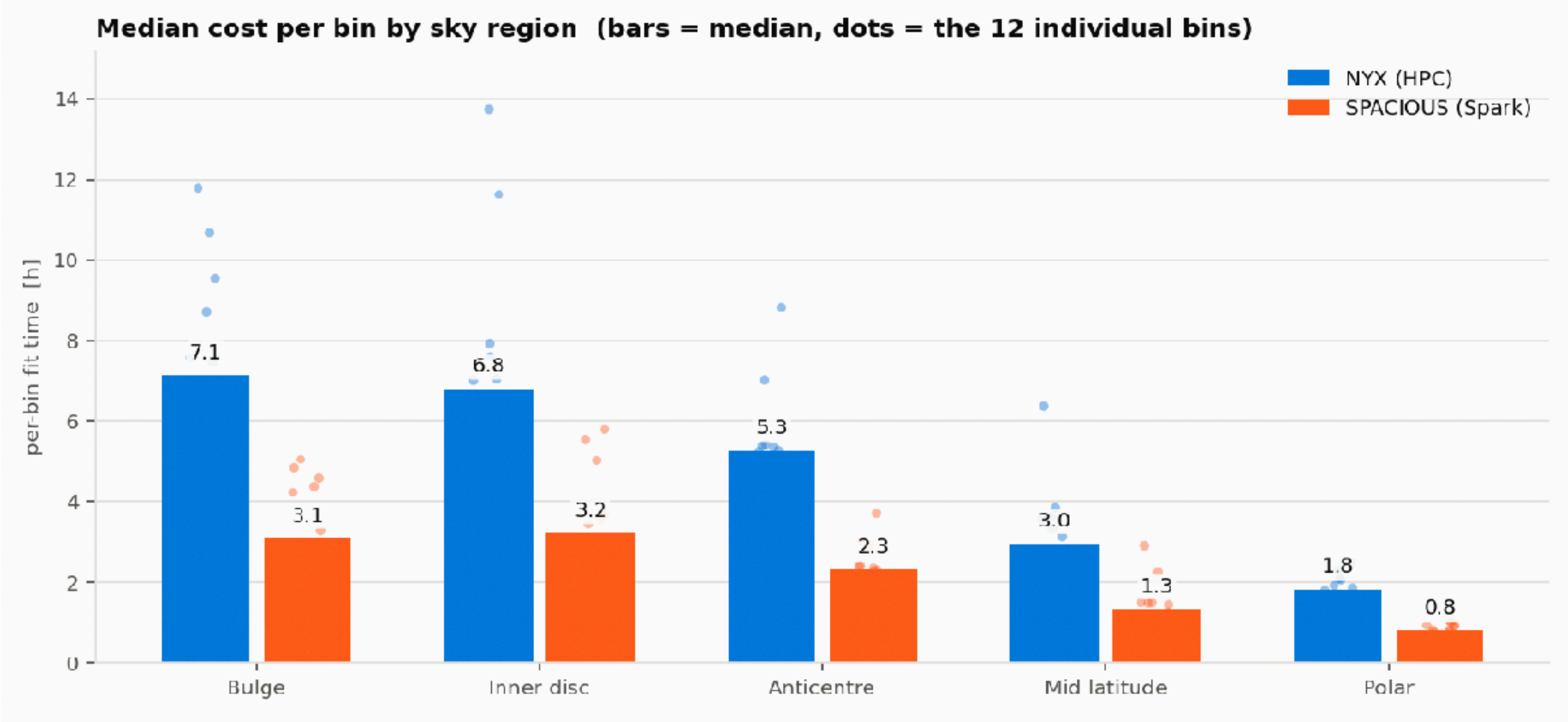
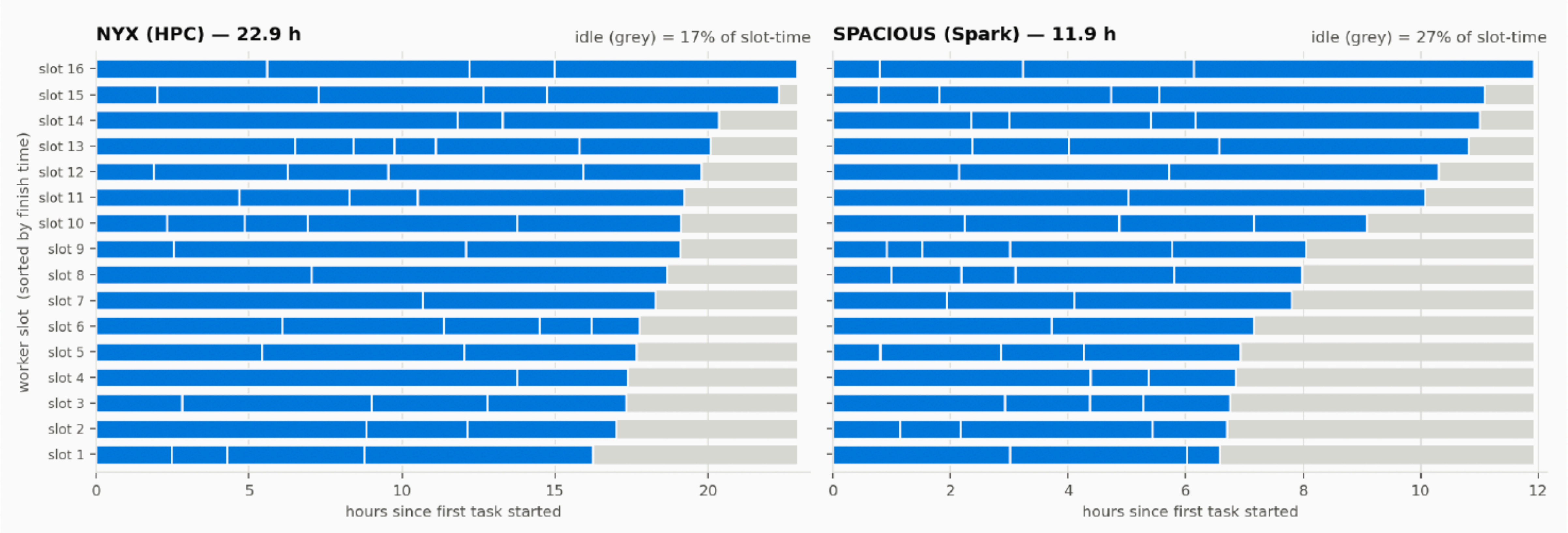


Figure 2. Median time per pixel by sky region on each cluster; the dots are the 12 individual pixels behind each bar.

Region	NYX min / median / max [h]	SPACIOUS min / median / max [h]	Median ratio
Bulge	4.53 / 7.14 / 11.80	1.94 / 3.09 / 5.04	2.27
Inner disc	3.31 / 6.80 / 13.74	1.40 / 3.22 / 5.79	2.29
Anticentre	3.77 / 5.26 / 8.82	1.63 / 2.32 / 3.72	2.23
Mid latitude	2.00 / 2.96 / 6.37	0.81 / 1.35 / 2.91	2.27
Polar	1.32 / 1.83 / 2.47	0.56 / 0.80 / 1.02	2.25

The Gaia catalogue selection function

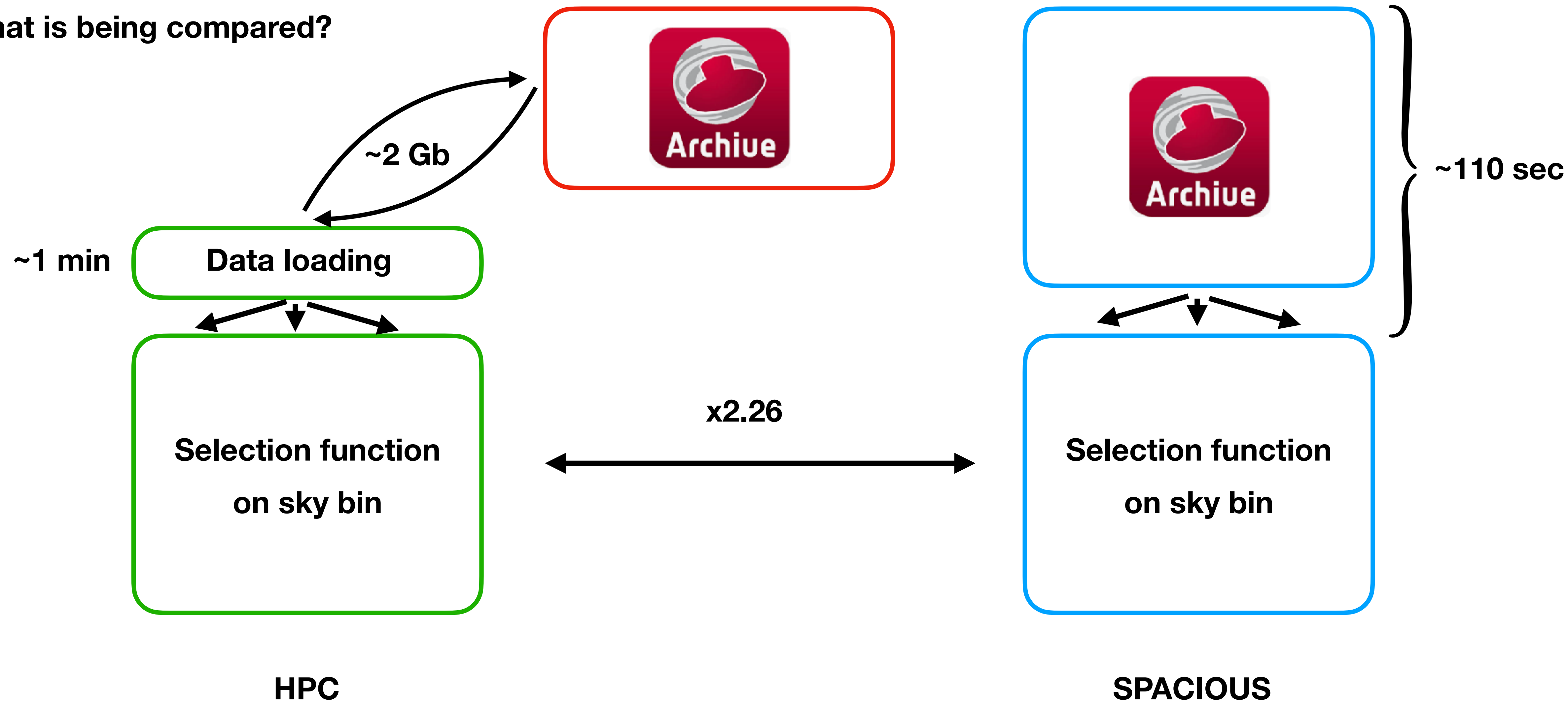
Performance on SPACIOUS



The Gaia catalogue selection function

Performance on SPACIOUS

What is being compared?



The Gaia catalogue selection function

Performance on SPACIOUS

4.4. Code-to-data implementation in SPACIOUS

The inference is embarrassingly parallel across pixels, and we run it on two different systems. On the one hand, we use the computational resources provided by SPACIOUS⁵. SPACIOUS (Science PLatform Cloud Infrastructure for Outsize Usage Scenarios) is a Horizon Europe project aiming to provide a cloud-native infrastructure for large-scale astrophysical data analysis. We deploy the pipeline with Apache Spark, dispatching every pixel through `applyInPandas` keyed on the HEALPix index. On the other hand, we use a traditional Slurm-managed (Jette & Wickberg 2023) HPC cluster, where each HEALPix is computed in parallel through a Python `multiprocessing.Pool` that distributes the pixels across the cores of a single node. We use both deployments for different reasons: i) the Slurm cluster extends the number of available CPUs, while ii) the Spark deployment tests the code-to-data paradigm in which the computation is shipped to the cluster hosting the data rather than the data being downloaded for local compute.

We compare the performance of the two deployments on a set of 60 HEALPix-7 pixels representing five regions of the

⁵ <https://spacious.ub.edu/>

sky: bulge, inner disc, anticentre, mid-latitude b (representing the thick disc transition), and Galactic poles. These regions represent a wide range of per-pixel computational cost, which is driven by stellar density. On the Slurm cluster, the fastest pixel completes in 1.32 hr and the slowest in 13.74 hr, with a mean of 5.05 hr per pixel. On SPACIOUS, the fastest pixel completes in 0.56 hr and the slowest in 5.79 hr, with a mean of 2.32 hr per pixel. The ratio between the two is close to constant at 2.26 across all pixels in all five regions, which is a signature of a per-core hardware speed difference rather than memory pressure or scheduling. We should note that both sets of timings are compute-only, and the data processing step, i.e., querying, binning, and, for the Slurm cluster, downloading and uploading the *Gaia* data beforehand, is excluded for both deployments. SPACIOUS reaches the same starting point directly from the raw *Gaia* data automatically and within the run, while the Slurm pipeline requires separate manual extraction and upload steps. For future astronomical surveys, however, the data volume will be of the order of Tb or even Pb. For those, the code-to-data paradigm represented by SPACIOUS will not only continue to offer a performance gain but, more importantly, will enable the computation which is intractable in traditional data-to-code approaches. We additionally quantified the cost of this SPACIOUS run by deploying it on the Google Cloud service, where the same computation costs on average €0.50 per HEALPix.