Validating 21 cm cosmology analysis pipelines: lessons learned and future outlooks

Piyanat (Boom) Kittiwisit (UWC, Cape Town)

With contributions from the HERA Validation Team:
James Aguirre (UPenn), Steven Murray (SNS), Robert Pascua (McGill), Lisa McBride (CNRS), Zachary Martinot (UPenn), Hugh Garsden (Manchester) and others

Cosmology in the Alps 2024

HERA

National Research Foundation

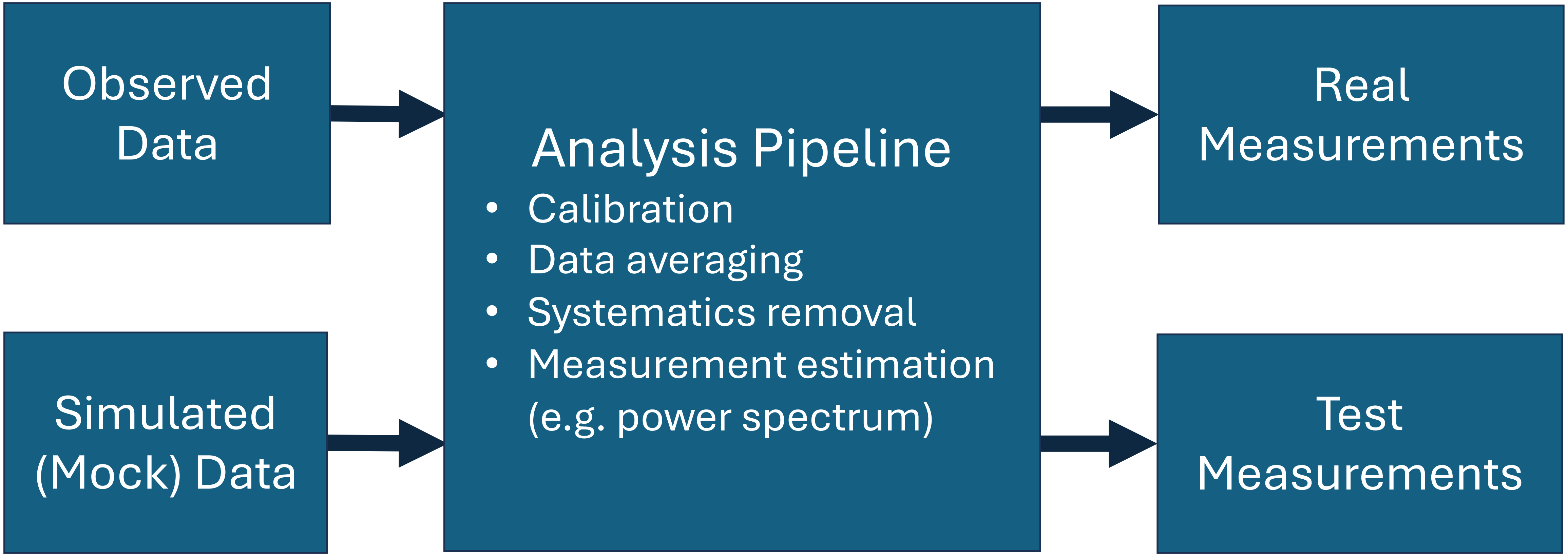
SARAO
South African Radio Astronomy Observatory

UNIVERSITY of the WESTERN CAPE

CRC
Centre for Radio Cosmology

What is pipeline validation?

Testing of the analysis pipeline with mock data for unknown systematics and signal loss



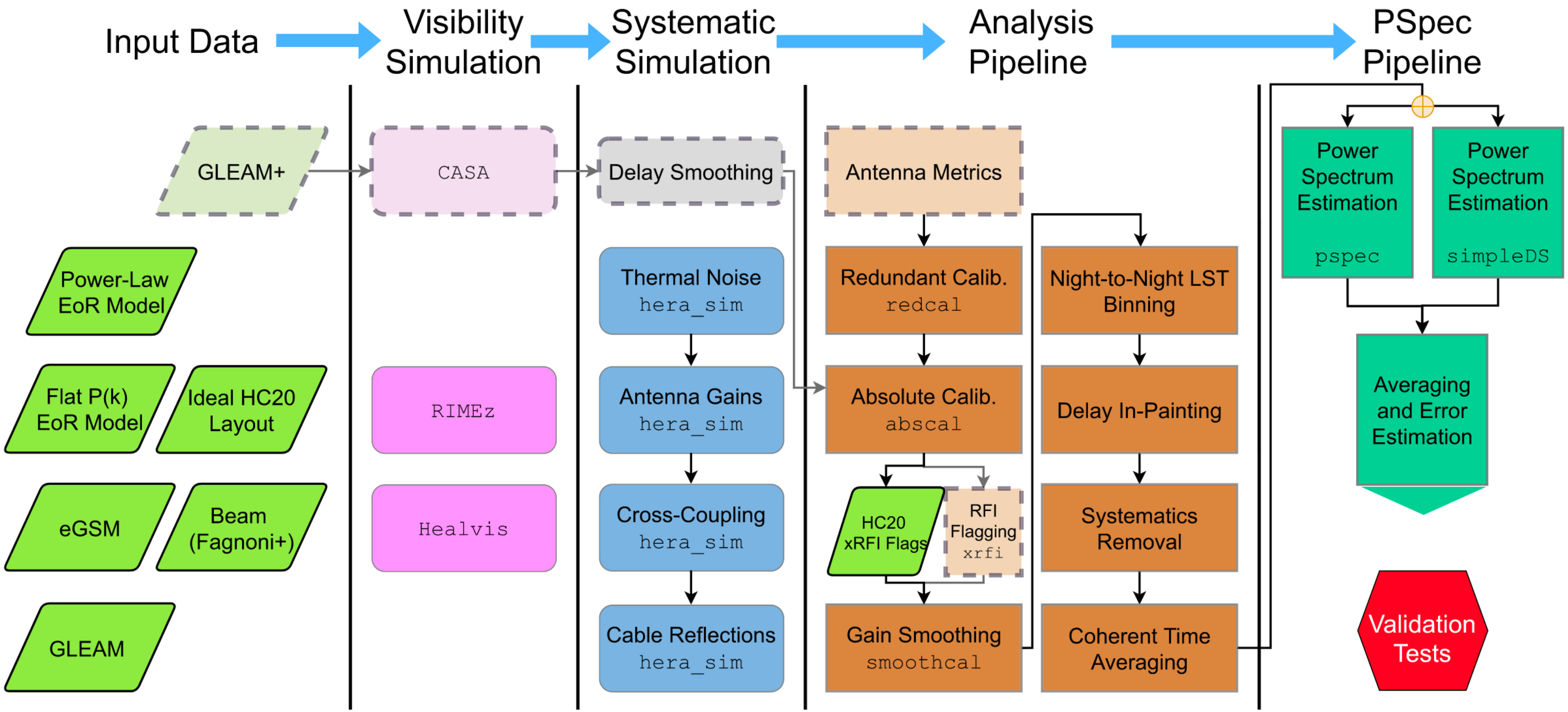
Why should we validate the pipeline?

- History of revising power spectrum limits due to (previously) unknown biases from complex and novel analysis techniques
  - Liu & Shaw 2020 provides a good overview of these analysis issues
- Past revisions:
  - GMRT – Paciga et al. 2011 as amended by Paciga et al. 2013
  - PAPER (precursor of HERA) – Ali et al. 2015 as amended by Kolopanis et al. 2019 and Cheng et al. 2018
  - BICEP2 CMB B-mode polarization fault detection

Validation in Literature

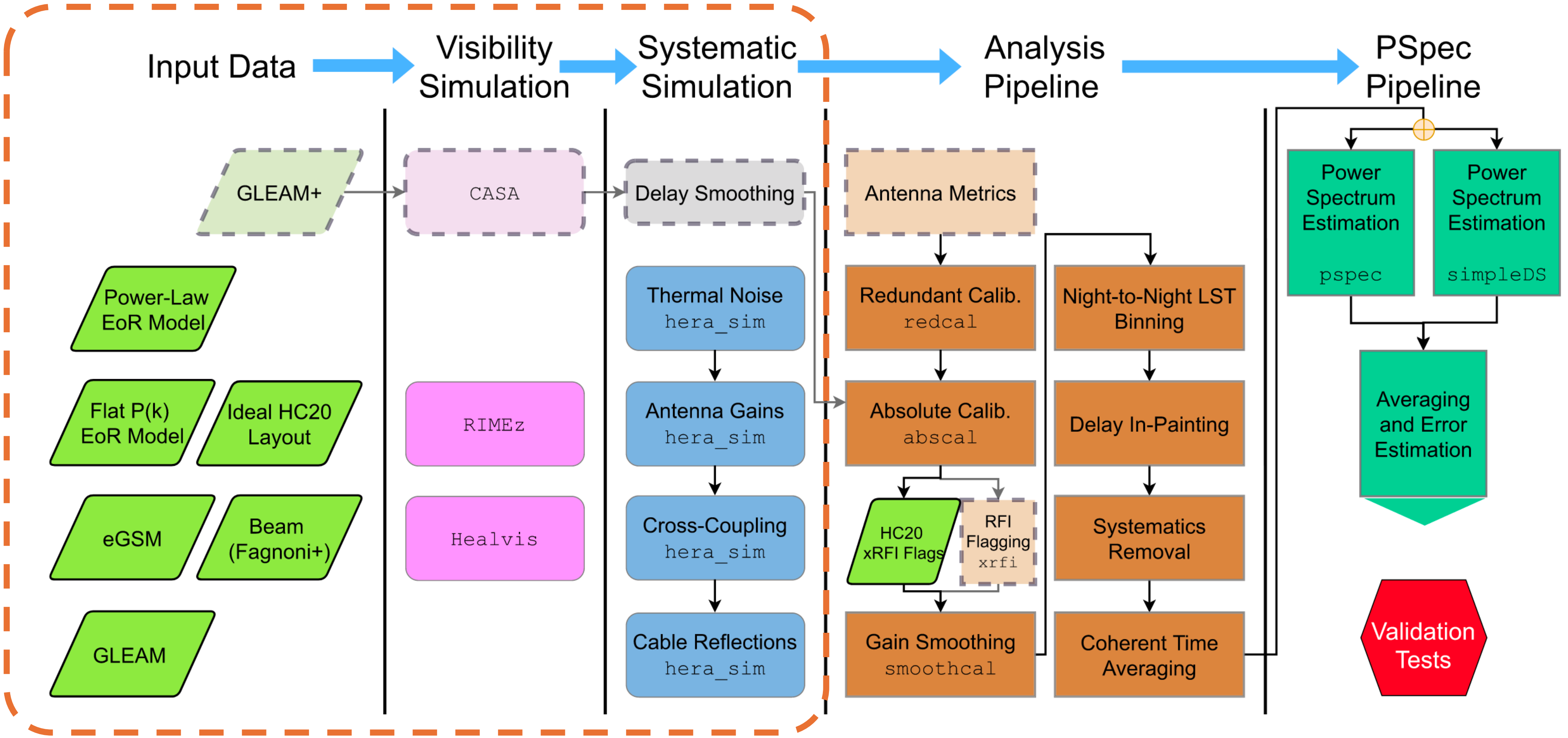
Validation in Literature: HERA

- Full forward modelling approach (Aguirre et al. 2022)



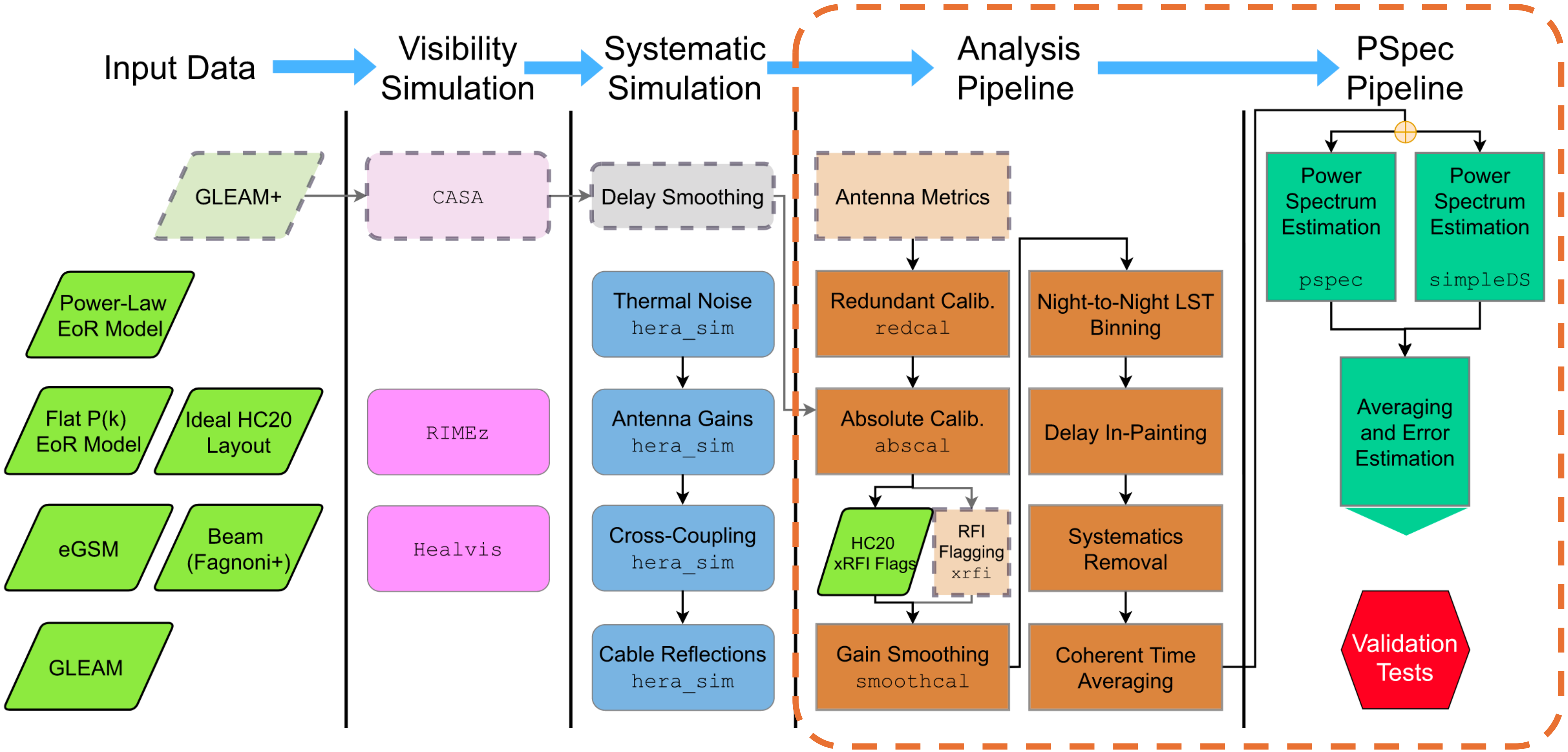
Validation in Literature: HERA

- Simulate mock data with different sky and systematic components



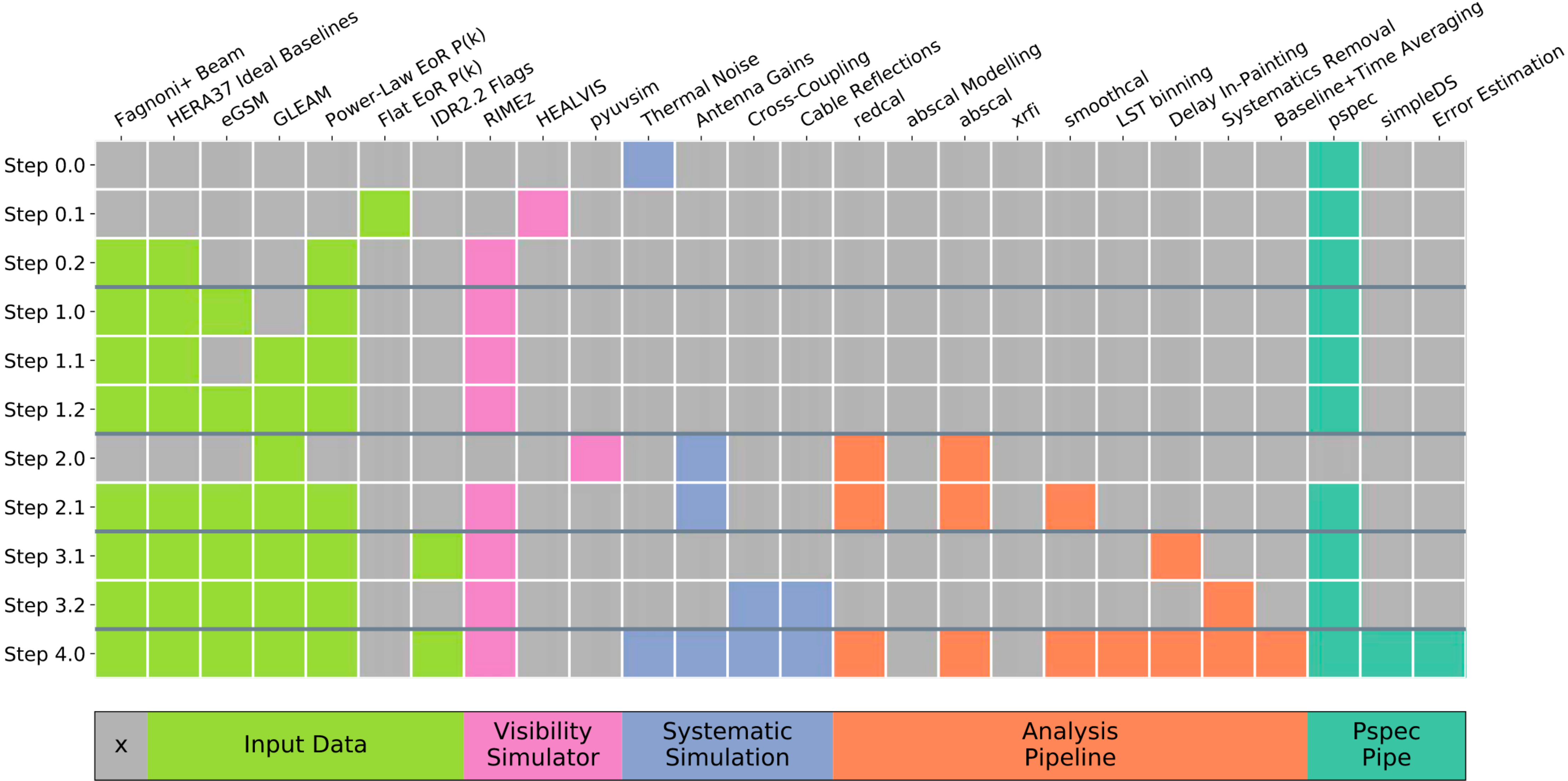
Validation in Literature: HERA

- Test different parts of the pipeline through a series of “validation tests”



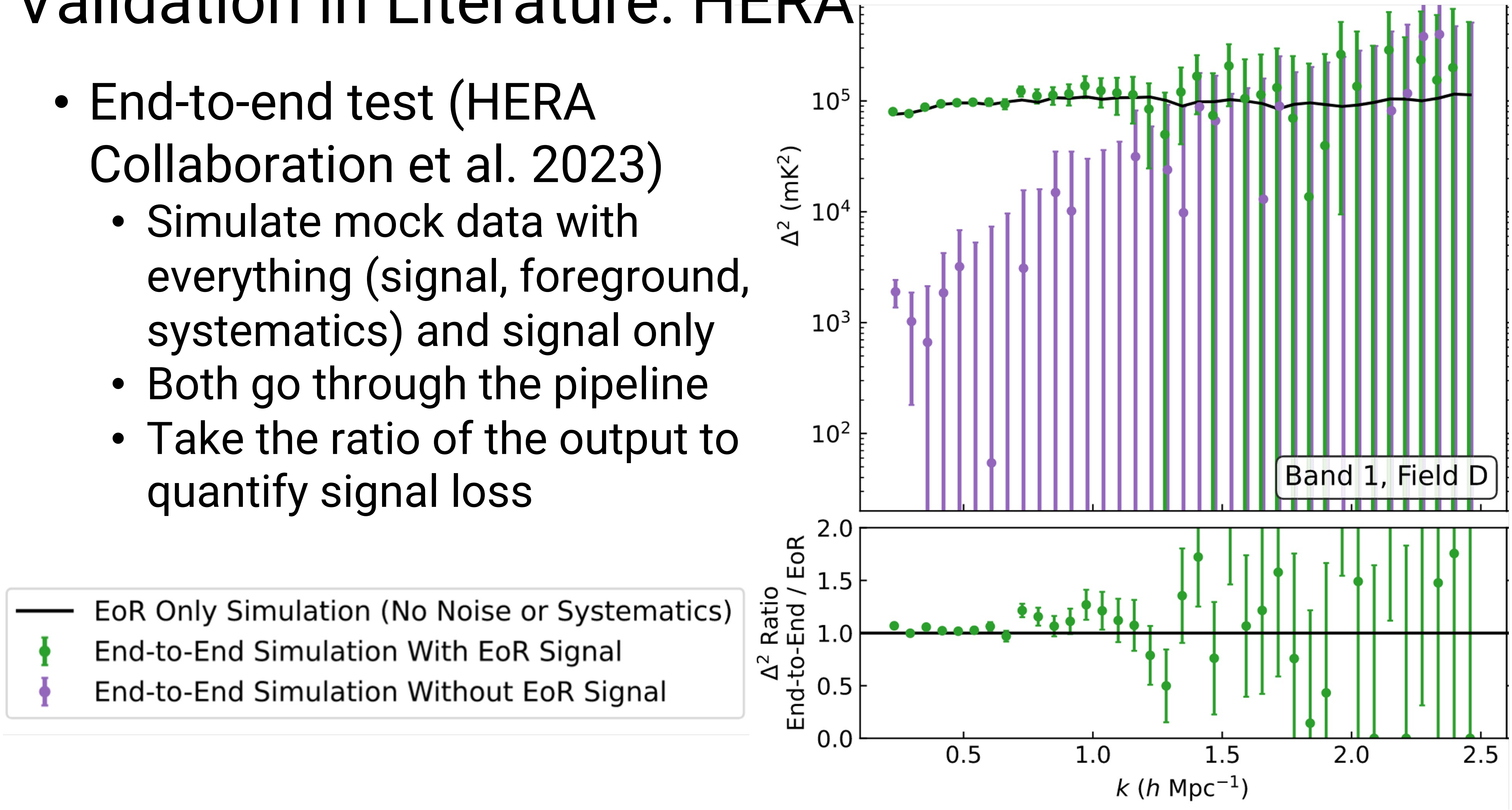
Validation in Literature: HERA

- Gradually building up the complexity of the validation tests



Validation in Literature: HERA

- End-to-end test (HERA Collaboration et al. 2023)
  - Simulate mock data with everything (signal, foreground, systematics) and signal only
  - Both go through the pipeline
  - Take the ratio of the output to quantify signal loss



Validation in Literature: LOFAR & NenuFAR

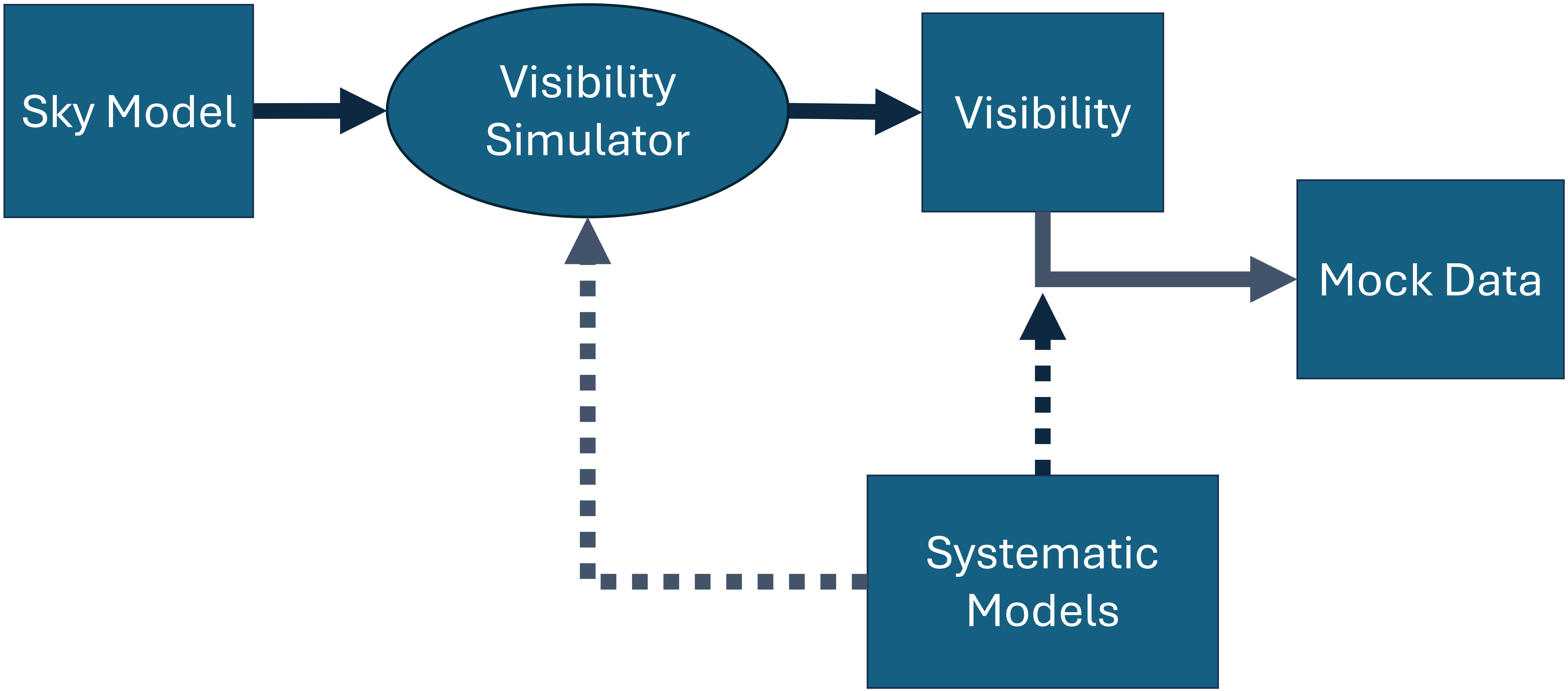
- Signal injection test (Mertens et al. 2018, 2020, Munshi et al. 2024)
  - Inject detectable simulated EoR signal to data (at the level in-between noise and foreground)
  - Signal injected data goes through the pipeline
  - Subtract the output with those from data without injected signal to obtain the residual
  - Form PS from the residual and the injected signal
  - Their ratio determine the signal loss
- Calibration test – Mevius et al. 2022
- GPR foreground removal test – Gan et al. 2023
- Also used similarly in MeerKLASS (Cunnington et al. 2023)

Validation in Literature: MWA

- Multifaceted approach, motivated by 3+ different pipelines used in the analysis
- Cross-validation with other pipelines (Beardsley et al. 2016, Barry et al. 2019b, Li et al.2019, Trott et al 2020)
- End-to-end test through signal-only and signal + foreground simulations to determine signal loss through the FHD+ εppsilon pipeline (Barry et al. 2019a,b Li et al 2019)
- Signal-only simulation from a large EoR lightcone showing no signal loss in the CHIPS pipeline (Trott et al. 2020)

Simulating Mock 21 cm Data

Simulating Mock 21 cm Data



Visibility Simulator

- Evaluate the Radio Interferometric Measurement Equation (RIME)
  - Original formulation: Hamaker et al (1996), Sault et al (1996), Hamaker & Bregman (1996), Hamaker (2000), Hamaker (2006)
  - Revision: Smirnov (2011a, b, c, d), Price & Smirnov (2015)

Sky brightness in the direction $\boldsymbol{\sigma}$

Visibility from baseline pq

$$\mathrm{V}_{pq} = \int_{4\pi} \boldsymbol{J}_p(\boldsymbol{\sigma})\mathrm{B}(\boldsymbol{\sigma})\boldsymbol{J}_q^H(\boldsymbol{\sigma})\, d\Omega$$

Integral over the full sky solid angle

Jones terms for antenna p and q

Visibility Simulator

- On a computer, we must **choose a discrete basis to turn the RIME integral into a sum**
  - The sum may be done in the real, Fourier, or spherical harmonic domains
- Examples of discrete basis
  - Point-source or pixelized: Model sky components as an ensemble of unresolved point sources
  - Spherical harmonic (Shaw et al. 2014): Model sky components as a linear combination of spherical harmonic modes (m modes)
  - Other bases include Gaussian blob and wavelets

Visibility Simulator: Dedicated

Purposely developed software for visibility simulation

Simulator	Basis	Language	Affiliation	Latest Release	Maintained	GitHub Repository
`pyuvsim`	Point source	Python	RASG	2023-0719	Yes	RadioAstronomySoftwareGroup/pyuvsim
`matvis`	Point source	Python, CUDA	HERA	2023-11-30	Yes	HERA-Team/matvis
`WODEN`	Point source, shapelet	C, CUDA, Python	MWA, Curtin	2023-10-25	Yes	JLBLine/WODEN
`OSKAR`	Point source, Gaussian	C, C++, Python	SKA	2022-05-26	Yes	OxfordSKA/OSKAR
`driftscan`	Spherical harmonic	Python, Cython, C++	CHIME, CHORD, HIRAX	2022-10-01	Yes	radiocosmology/driftscan
`healvis`	Point source (HEALPix)	Python	RASG	2019-04-04	Deprecated	rasg-affiliates/healvis
`PRISim`	Point source	Python2	N. Thyagarajan	2020-06-13	Assume No	nithyanandan/PRISim

* As of 2024-03-19

Visibility Simulator: General Purpose

Analysis software with simulation capability, primarily through the building of a sky model for calibration

Simulator	Basis	Language	Affiliation	Latest Release	Maintained	GitHub Repository
FHD	Point source (uv plane)	IDL	U of Washington	2021 (Last update 2024-02)	Yes	EoRImaging/FHD
pyFHD	Point source (uv plane)	Python	N. Barry & ADACS	No stable release yet	Yes	ADACS-Australia/PyFHD
CASA	Point source	Python, Fortran	VLA/ALMA	3 weeks ago	Yes	casangi
WSClean	?	C	A. Offringa, ASTRON	5 months ago	Yes	aroffringa/wsclean/
SAGECal	Point, gaussian, shapelet	C/C++, CUDA	ASTRON, NL eScience Center	2023-07-31	Yes	nlesc-dirac/sagecal
maqtres	?		Rhodes U	2022-2023	Maybe	ratt-ru/meqtrees

* As of 2024-03-19

Making visibility simulator faster for HERA

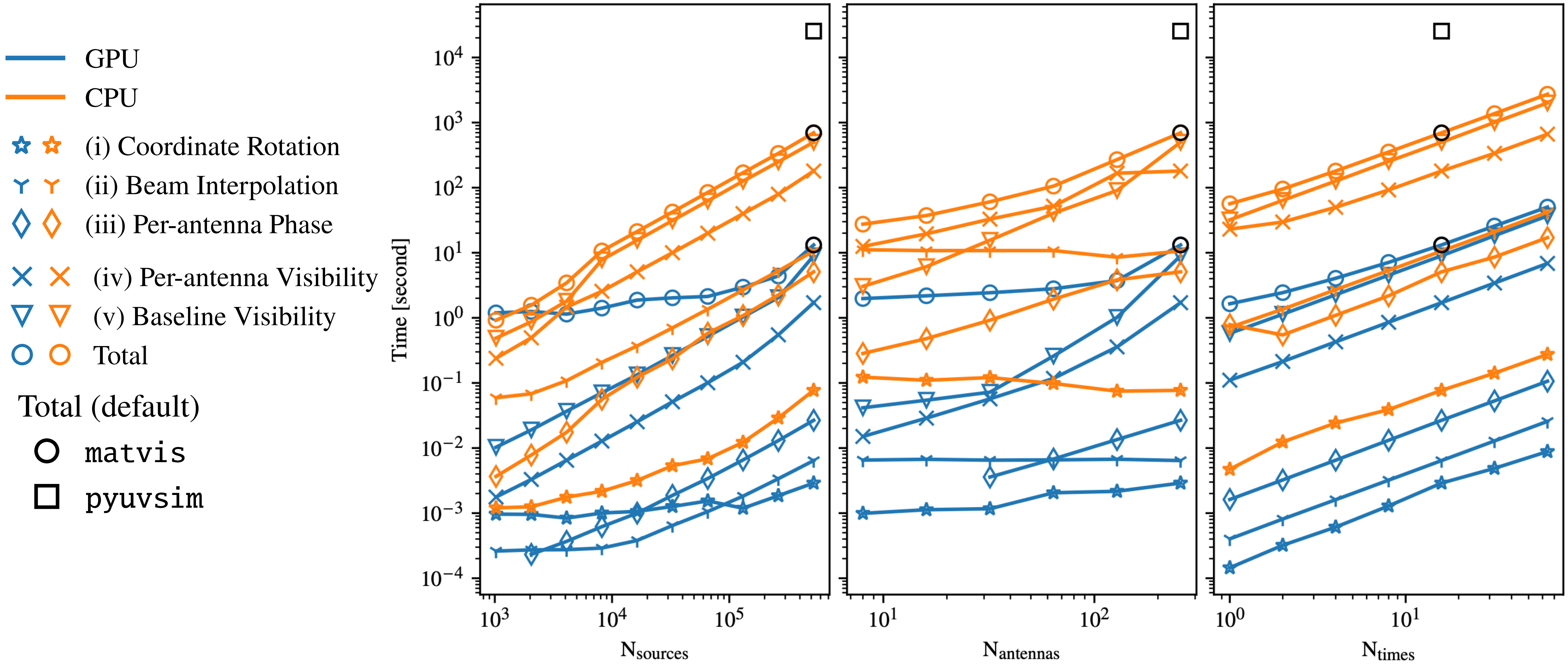
- We want to simulate multiple sky components over real observational parameters
- If using `pyuvsim`, the wall time is ≈3M CPU hours per sky component
- How can we do this faster?

HERA Validation Simulation Parameters	
Baselines	61075
Time steps	17280
Frequency channels	1536
Polarizations	4
Point sources	300,000+
Diffuse/EoR model pixels	786,432

`matvis`: Matrix-based RIME Algorithm

- Calculate **per-antenna** “voltages”
- Form per-baseline visibilities from an outer product of per-antenna voltages
- More number of calculations but can be efficiently performed by modern linear algebra routines and implementable on a GPU
- Further trade some accuracy for speed by opting for trigonometric-based coordinate transformation (with correction) in placed of `astropy`
- See Kittiwisit et al. (submitted to RASTI), arXiv:2312.09763

`matvis`: Speed Improvement



10^1—10^3 faster than `pyuvsim` although still computationally expensive, ≈20,000 GPU hours per sky component for full HERA

`matvis`: (Current) Limitations

- Only support drift-scan simulation
- Only support unpolarized sky although fully support polarized beam
- Sky models must have no negative values
- All baselines (in the provide array configuration) must be simulated at once

Sky Models

- Determine the realism of the simulations
- We do not have complete information in EoR/CD frequencies

Sky Models: Point Source

- Radio source catalogs in EoR frequencies
  - VLA NVSS (Primarily northern Sky)
  - LOFAR LoTSS (Shimwell et al. 2022)
  - MWA GLEAM and GLEAM-X (Hurley-Walker et al 2017, 2022)
  - GMRT SCG (Riseley et al. 2016), TGSS (Intema et al. 2017)
- EoR specific catalogs and models
  - LOFAR NCP (Yatawatta et al. 2013)
  - MWA LoBES (Lynch et al. 2021)
- Already in point-source basis!
- But none covers the full sky, and each survey has different depth.
- Mock catalog based on source count distribution (e.g. Franzen et al 2019) can offer a good alternative for validation (though not for calibration)

Sky Models: A-Team Sources

- Very bright and persistent radio sources with “A” name ending
- Some have extended structures and can be partially resolved at long baseline, needing multi-point or shapelet models
- Shapelet models has been developed for Fornax A (Line et al 2020) and NCP sources (Yatawatta et al 2013) although not publicly available.

Source	RA	Dec	$S_{200\,\mathrm{MHz}}$/Jy	α
3C 444	22 14 26	−17 01 36	60	−0.96
Centaurus A	13 25 28	−43 01 09	1370	−0.50
Hydra A	09 18 06	−12 05 44	280	−0.96
Pictor A	05 19 50	−45 46 44	390	−0.99
Hercules A	16 51 08	+04 59 33	377	−1.07
Virgo A	12 30 49	+12 23 28	861	−0.86
Crab	05 34 32	+22 00 52	1340	−0.22
Cygnus A	19 59 28	+40 44 02	7920	−0.78
Cassiopeia A	23 23 28	+58 48 42	11900	−0.41

Table 2 from Hurley-Walker et al 2017

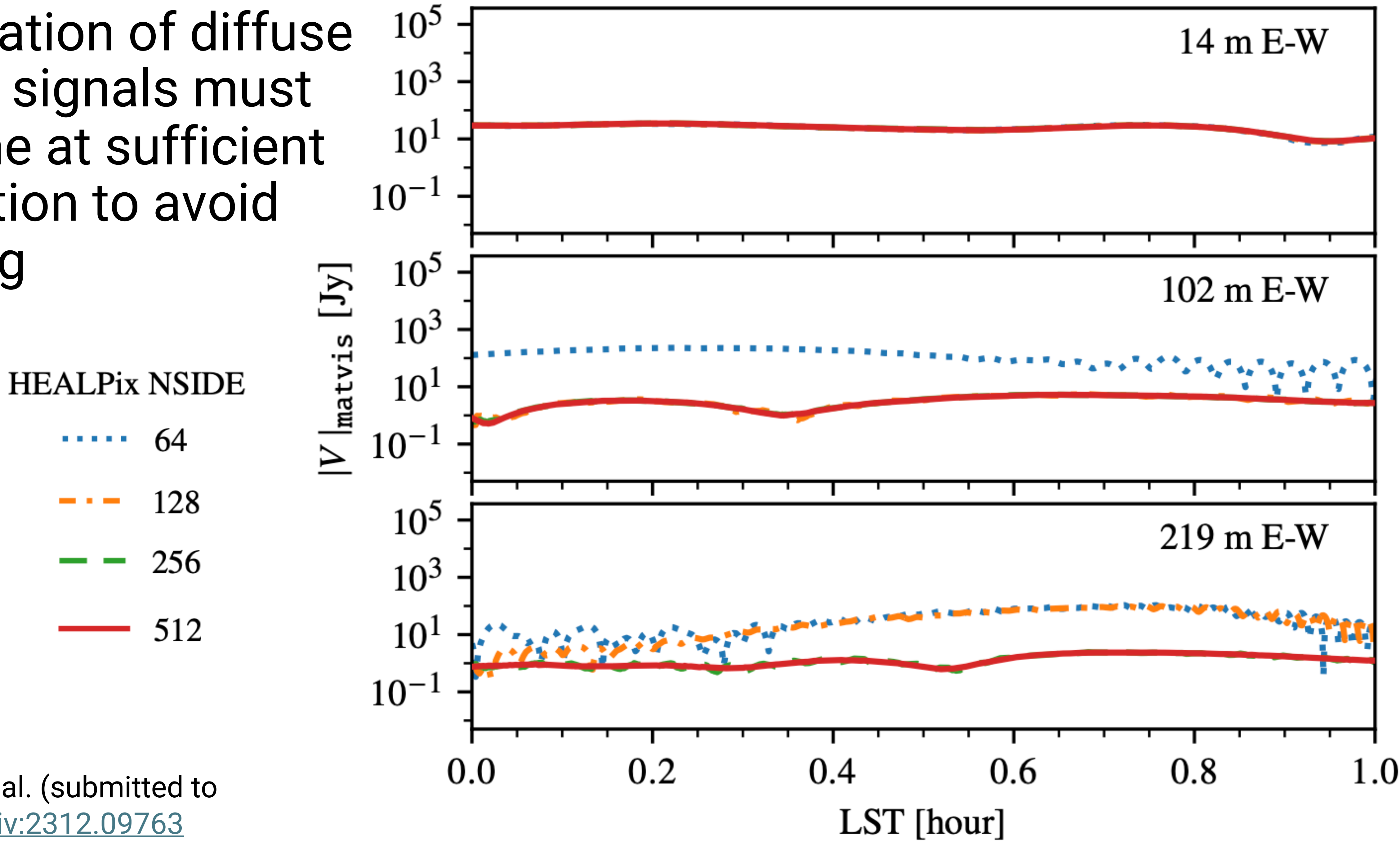
Sky Models: Diffuse Emission

- Must be pixelized (e.g. on a HEALPix grid) for simulators that use point-source basis
- Haslam 408 MHz from 1982(!) is still the most complete diffuse sky model that we have
- The reprocessed Haslam (Remazeilles et al. 2014) is known to have double counting issue
- PCA-based models are widely used: GSM (Oliveira-Costa et. al. 2008), pygsm, pygdsm, pysm3
  - Okay for validation
  - But make sure you know which data it is based on
  - Not really suitable for calibration. We need polarized maps for high-precision calibration (see e.g. Byrne et al 2022)

Sky Models: EoR model

- Hydrodynamic model is too small in volume and too computationally expensive for mock data simulation
- Semi-analytic model, e.g. 21cmFAST, can now produce a much larger simulation volume but not yet full-sky volume.
    - comoving cubes must be tiled into coeval maps via e.g. `cosmotile`
- Analytic model is nice for validation because we can generate the full-sky volume, and know exactly what we put in
- Must also be pixelized for simulators that use point-source approximation

Pixelization of diffuse or EoR signals must be done at sufficient resolution to avoid aliasing

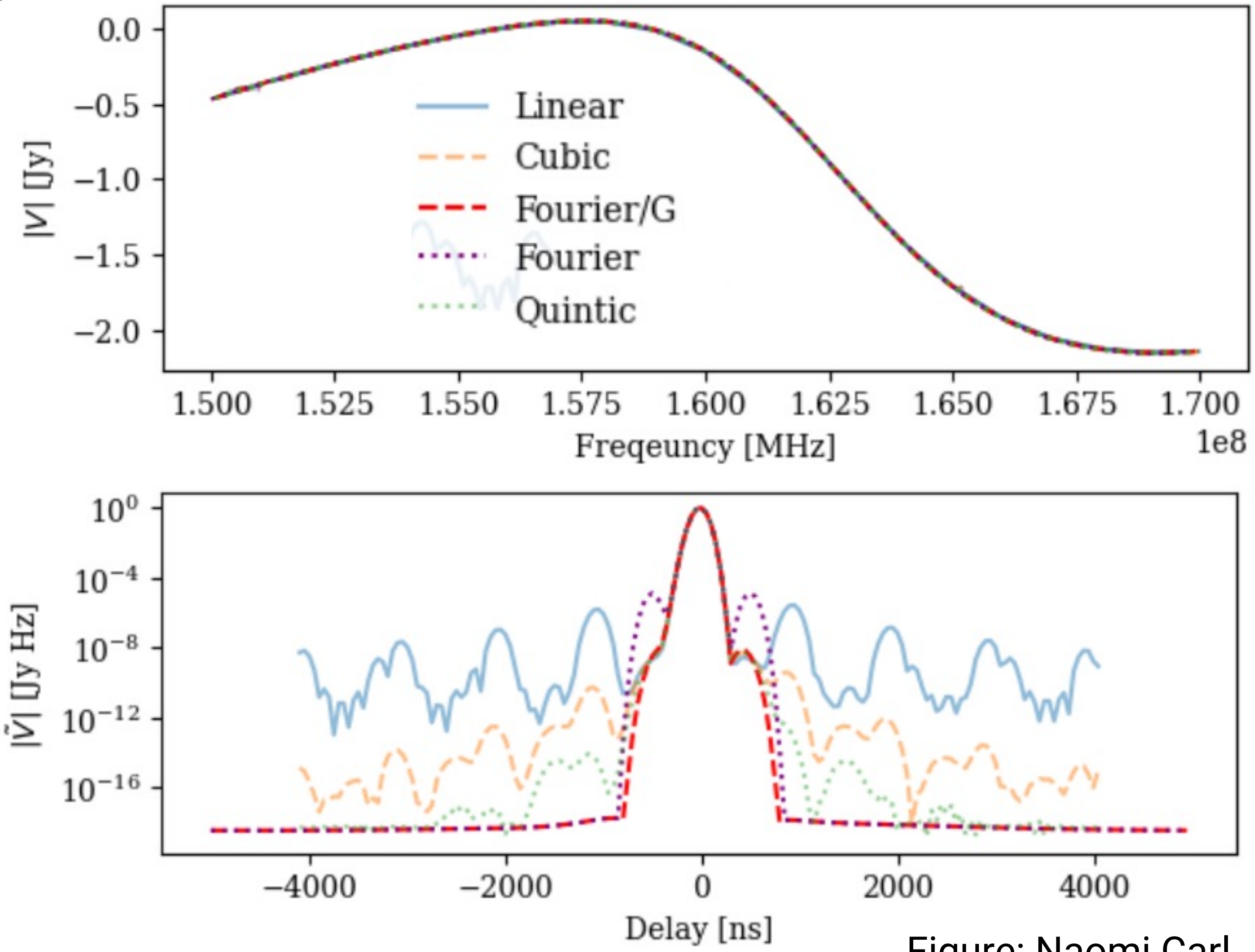


Kittiwisit et al. (submitted to RASTI), arXiv:2312.09763

Primary Beam Model

- Usually derived from computational electromagnetic (CEM) simulation, e.g. for HERA (Fagnoni et al. 2021), LOFAR (van Haarlem 2013), and MWA (Sokolowski et al. 2017)
- A few container packages for beam models has been developed: `everybeam` and `pyuvdata`
- Fitting an analytic model to the CEM beam is tricky but would make the simulation a lot faster (see Wilensky et al., submitted to MNRAS, arXiv:2403.13769)
- Evaluating a CEM-simulated beam at source positions requires interpolation

A higher-order spline interpolation is neccesary to ensure spectral smoothness



See (public) [HERA Memo #126] by Naomi Carl and Steven Murray

Figure: Naomi Carl

Quick Notes on Systematic Simulation

- `hera_sim`: a systematic simulator tools developed by the HERA validation team is publicly available.
  - It provides bandpass, mutual coupling, cable reflections, thermal noise, simple RFI, and mock visibility simulation tools, as well as a wrapper around more realistic visibility simulators.
- Paper(s) describing lessons learned from HERA validation process is in prep.

Outlooks for Validating Future Experiments

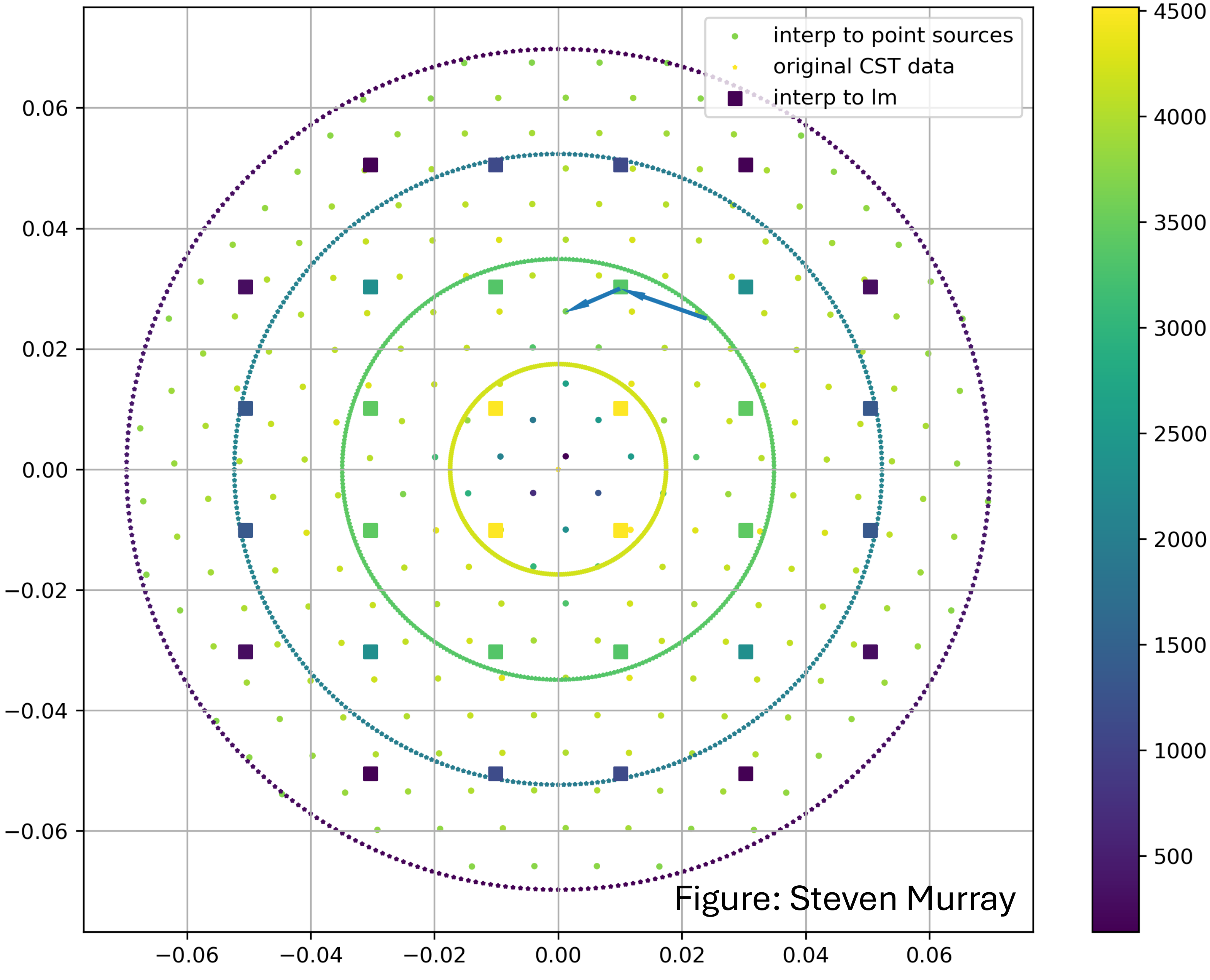
- Lots of already available tools, but we need more documentation, testing, integration and validation of them
- Existing sky and systematic models are okay, but several improvements can still be made
  - GSM can be improved if we have more data (and someone to do the work)
  - Polarized components – Little information
  - Adopting lightcone-based EoR model
- Cross—collaboration efforts would be ideal!

Summary

- Validation of the 21 cm cosmology analysis pipeline is crucial for credibility of our measurements
- Many sky and systematic models, and visibility simulators, have been developed although levels of documentation and testing can vary significantly
- Realism of the mock data simulation primarily depends on the sky models, but we lack the complete sky information in BAO/EoR/CD frequencies
- Making mock data is computationally expensive and has many non-trivial details (e.g. beam interpolation, aliasing from sky pixelization)
- Papers describing these details, and collaboration on modelling and software development, will be extremely useful for the community.

Photo: Dara Storer

- Because a beam usually has a pole at the zenith, interpolation should be done on an azimuth–altitude grid, not rectangular (l, m)
- See Wilensky et al, in prep.



`matvis` calculates per-baseline visibilities from an outer product of per-antenna visibilities

Hamaker's RIME on a point-source basis (e.g. pyuvsim)

$$V_{ij}^{pq}(\nu,t) = \sum_n \boldsymbol{A}_i^p(\boldsymbol{\theta}_n(t)) \cdot \boldsymbol{A}_j^{q*}(\boldsymbol{\theta}_n(t)) \times \mathbf{C}_{pq}^{(n)}(\nu)\, \exp\left(-2\pi \mathrm{i} \nu \tau_{ij}^{(n)}\right)$$

$$\tau_{ij}^{(n)} = \boldsymbol{X}_{\mathrm{hrz}}^{(n)}(t) \cdot \boldsymbol{b}_{ij}/c$$

matvis RIME (GPU implementable)

$$v_{ip}^{nk}(\nu,t) = \boldsymbol{A}_{ip}^{nk}(\nu,t)\sqrt{\mathbf{C}_{pp}^{(n)}(\nu)} \times \exp\left(-2\pi \mathrm{i}\nu\, \boldsymbol{X}_{\mathrm{hrz}}^{(n)}(t) \cdot \boldsymbol{x}_i/c\right)$$

$$V_{ij}^{pq} = \sum_{k,n} v_{ip}^{nk}\left(v_{jq}^{nk}\right)^{\dagger}$$

Trading some accuracies for speed