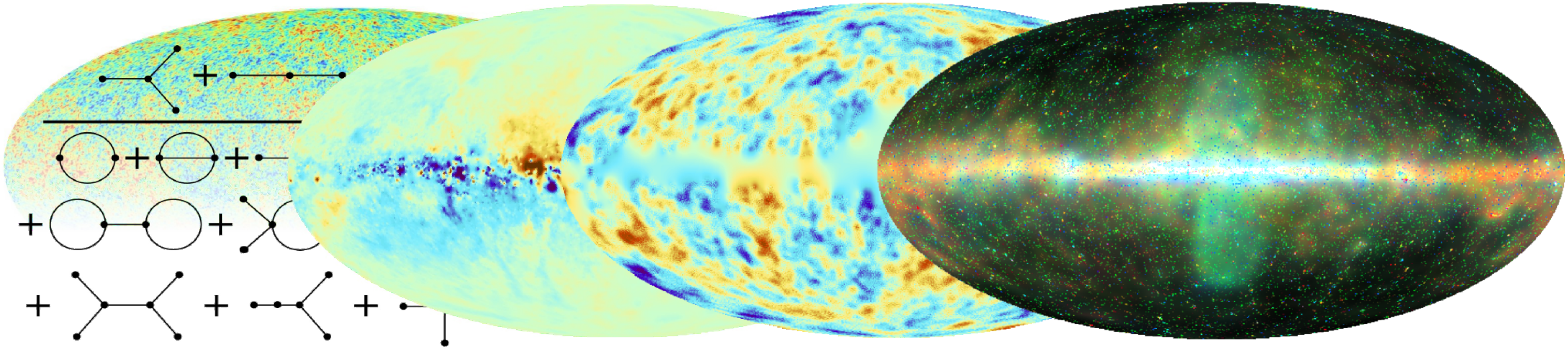


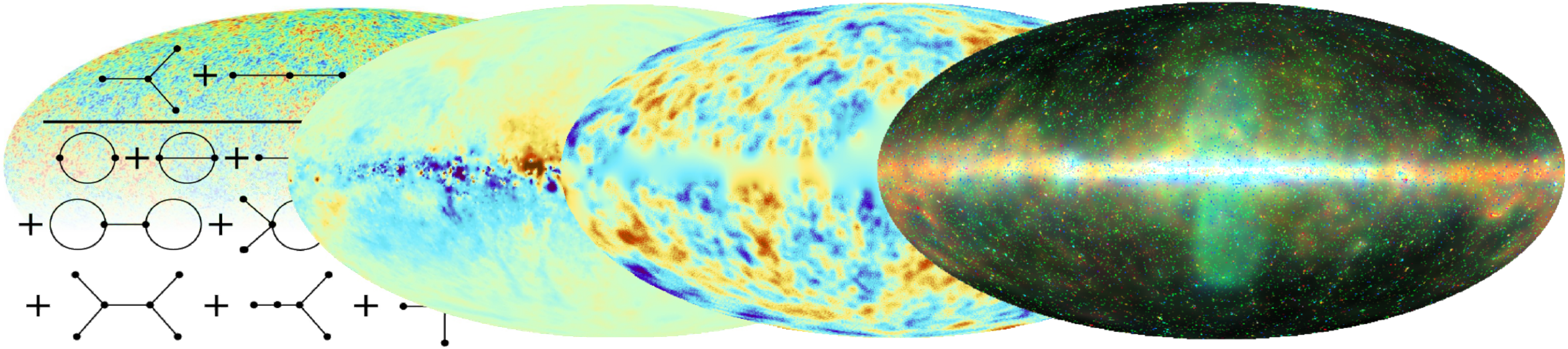
Information field theory

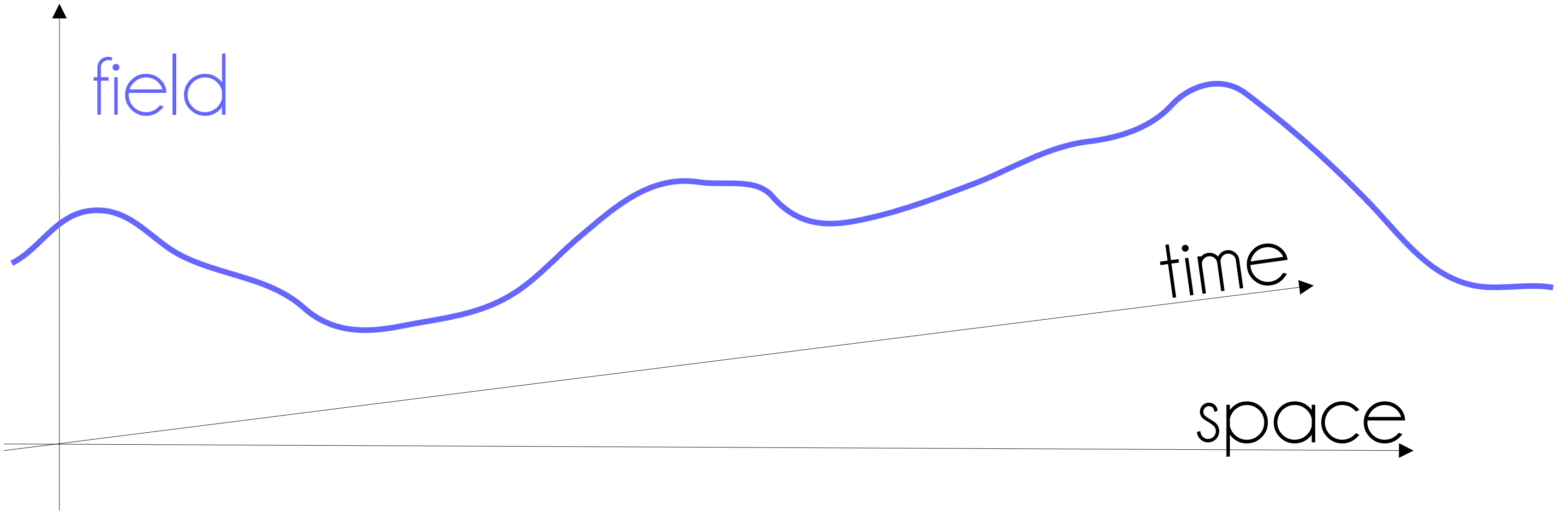
Torsten Enßlin

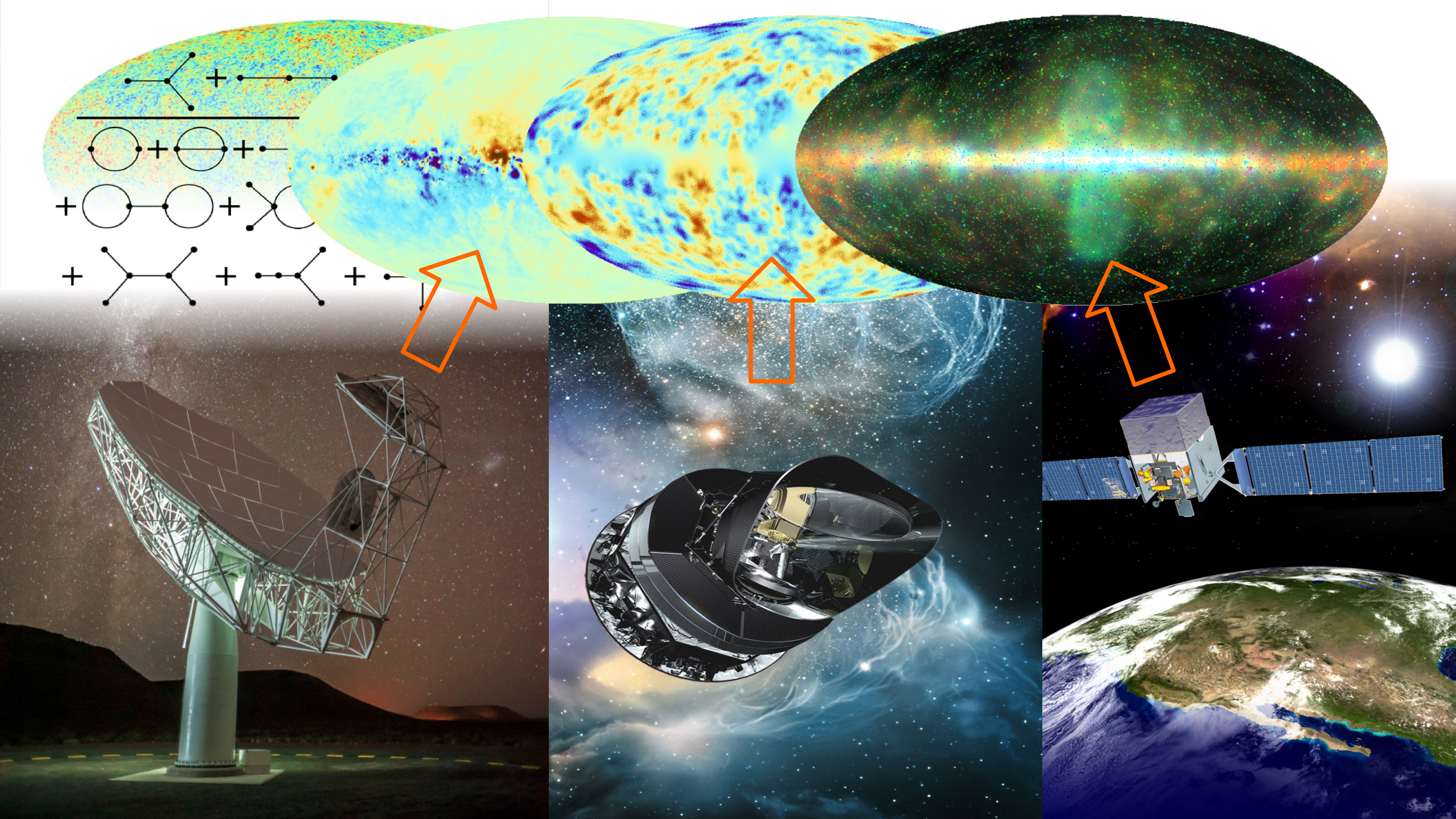
MPI for Astrophysics

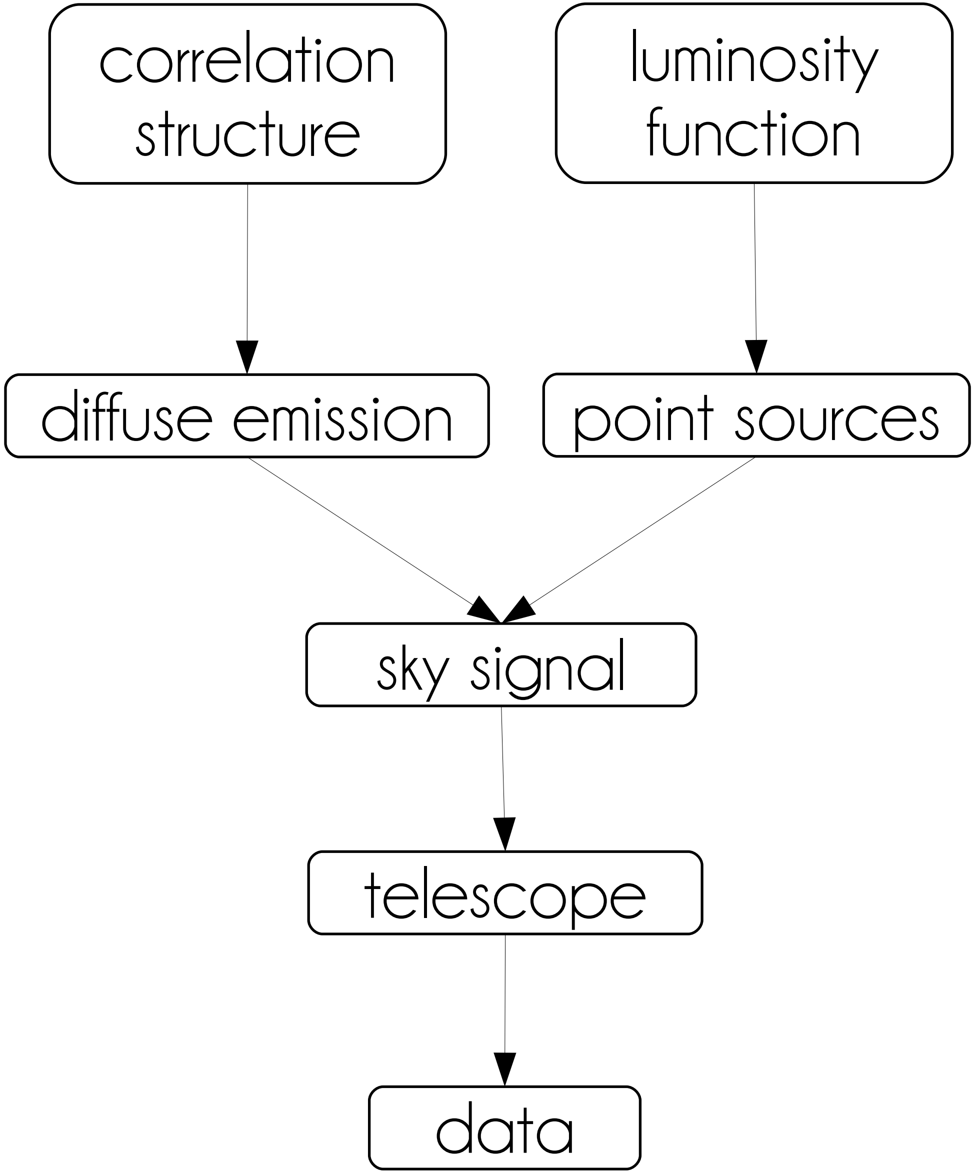
Ludwig Maximilian University Munich

IFT Team: Philipp Arras, Michael Bell, Vanessa Böhm, Sebastian Dorn, Martin Dupont, Mona Frommert, Philipp Frank, Mahsa Ghaempanah, Maksim Greiner, Philipp Haim, Sebastian Hutschenreuter, Henrik Junklewitz, Francisco-Shu Kitaura, Jakob Knollmüller, Christoph Lienhard, Reimar Leike, Ancla Müller, Johannes Oberpriller, Niels Oppermann, Natalia Porquerese, Daniel Pumpe, Tiago Ramalho, Martin Reinecke, Julia Stadler, Marco Selig, Theo Steininger, Valentina Vacca, Cornelius Weig, Margret Westerkamp, & many more




field
time
space

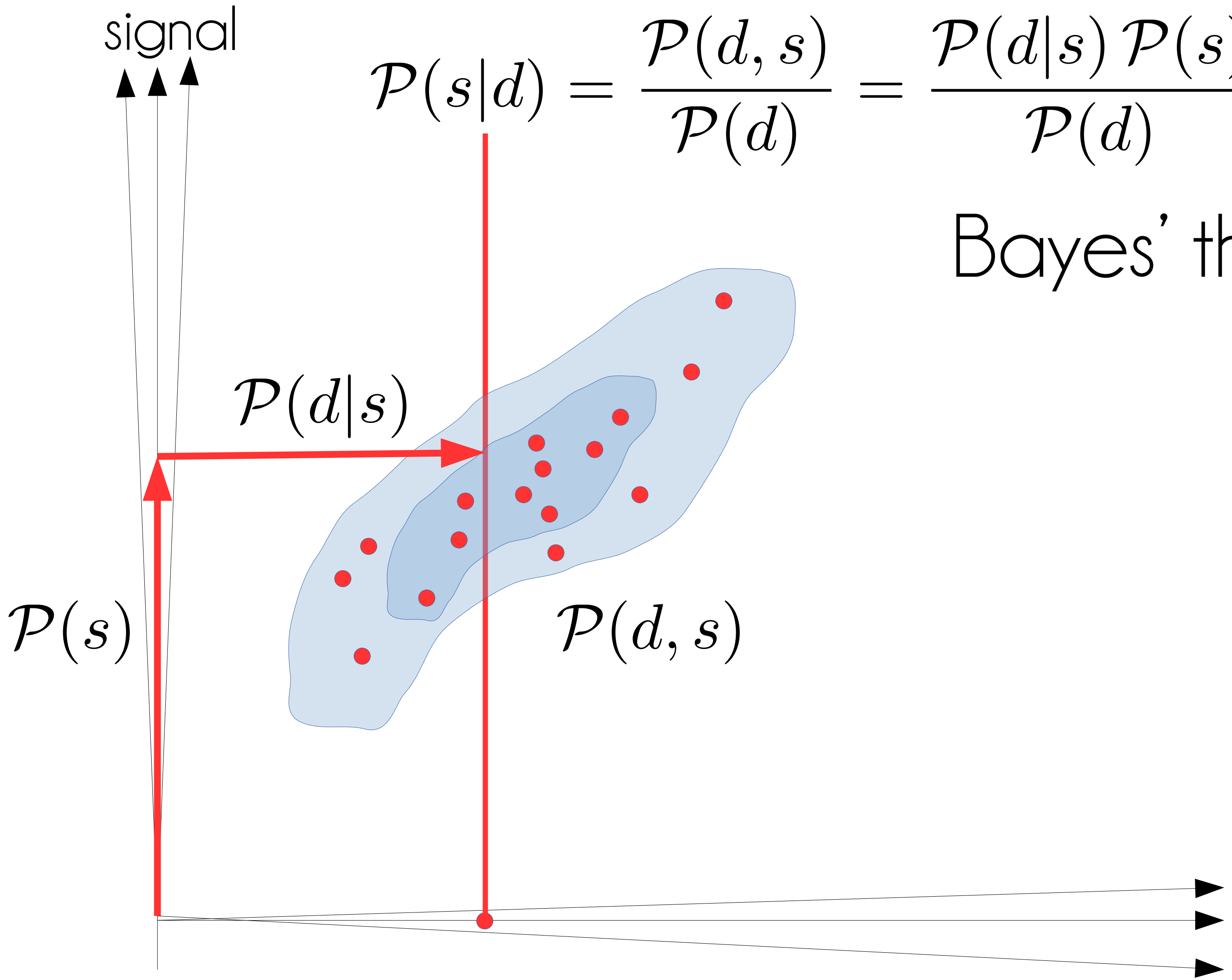


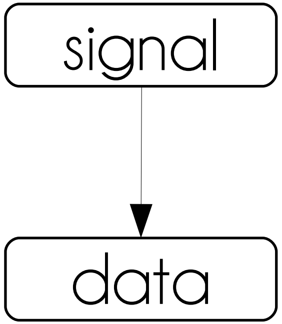


correlation structure
luminosity function
diffuse emission
point sources
sky signal
telescope
data

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d,s)}{\mathcal{P}(d)} = \frac{\mathcal{P}(d|s)\,\mathcal{P}(s)}{\mathcal{P}(d)}$$

Bayes' theorem





Information theory

$$\mathcal{P}(s|d) = \frac{\mathcal{P}(d,s)}{\mathcal{P}(d)} = \frac{e^{-\mathcal{H}(d,s)}}{\mathcal{Z}(d)}$$

$$\mathcal{H}(d,s) = -\log \mathcal{P}(d,s)$$

Information

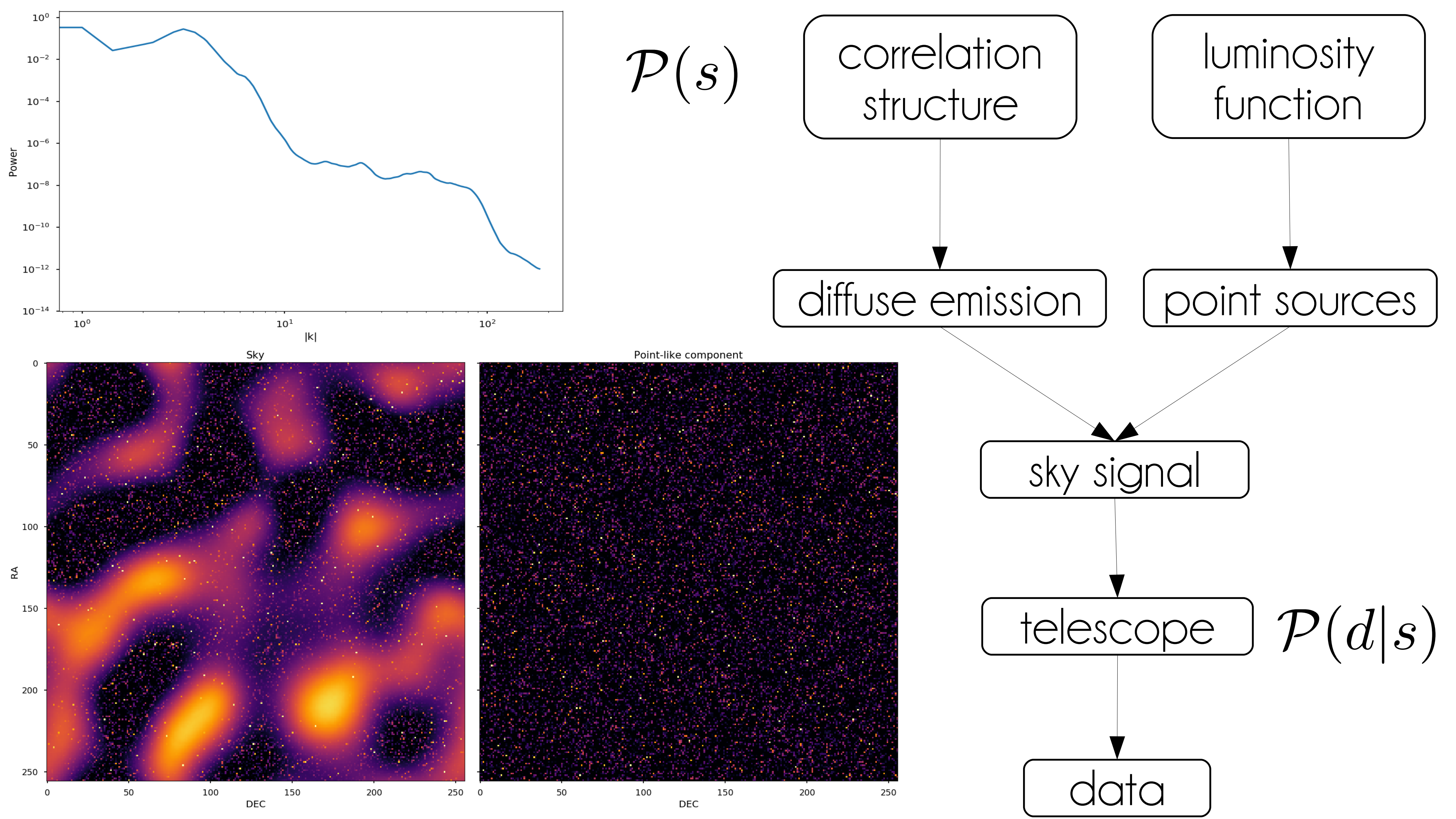
$$\mathcal{Z}(d) = \mathcal{P}(d) = \int \mathcal{D}s\, \mathcal{P}(d,s)$$

$$\mathcal{P}(d,s) = \mathcal{P}(d|s)\, \mathcal{P}(s)$$

$$\mathcal{H}(d,s) = \mathcal{H}(d|s) + \mathcal{H}(s)$$

is additive

`metric` `regularization`



Power
|k|
Sky
Point-like component
RA
DEC
DEC
$\mathcal{P}(s)$
correlation structure
luminosity function
diffuse emission
point sources
sky signal
telescope
$\mathcal{P}(d|s)$
data

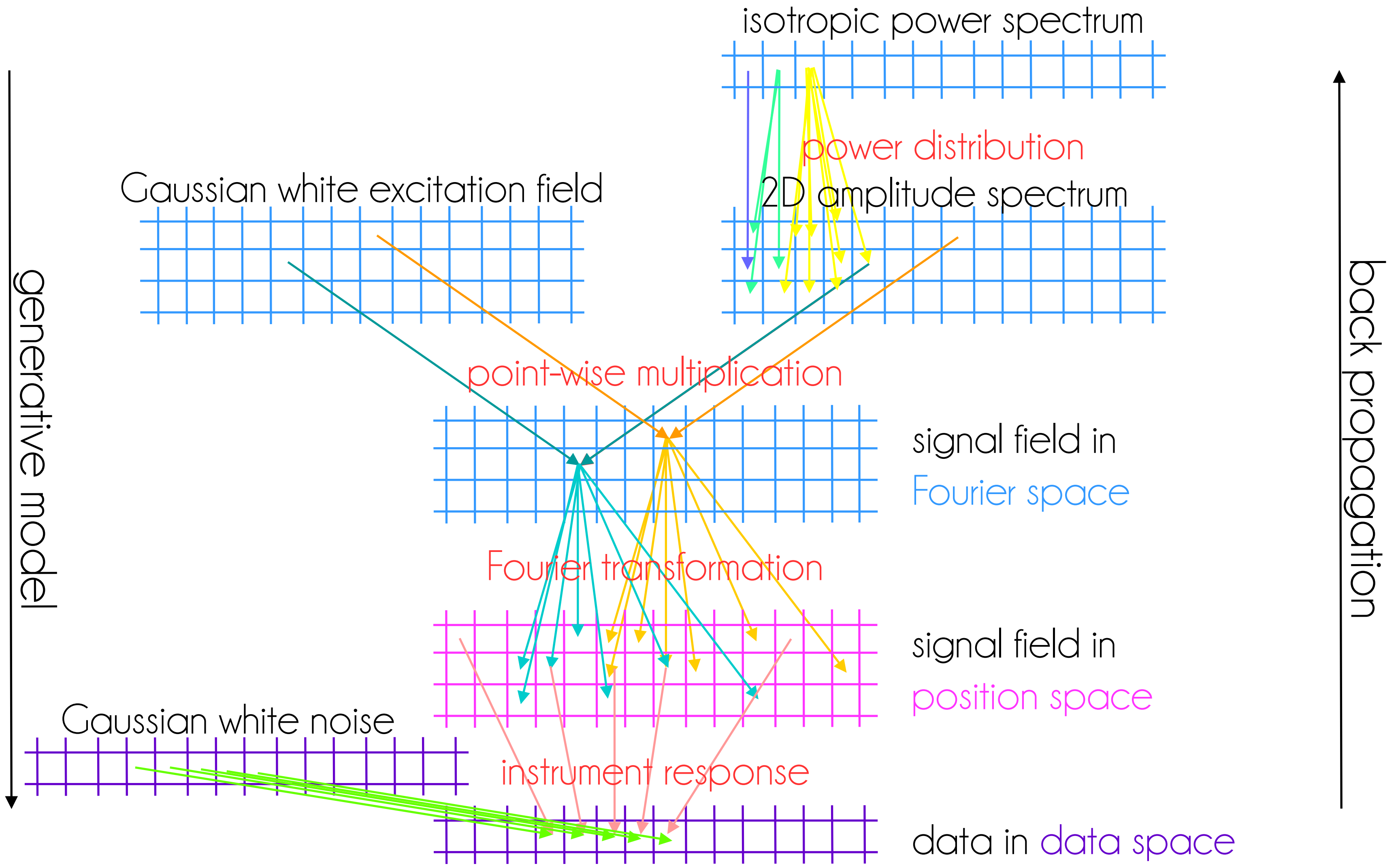
Photon counting instrument: Log-normal Poisson model

Information

$$\mathcal{H}(\boldsymbol{d},\boldsymbol{s},\boldsymbol{\tau}) = -\log\mathcal{P}(\boldsymbol{d},\boldsymbol{s},\boldsymbol{\tau})$$
$$= 1^\dagger\left[\log(d!) + \boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right] - \boldsymbol{d}^\dagger\log\left[\boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right]$$
$$+ \frac{1}{2}\boldsymbol{s}^\dagger\boldsymbol{S}^{-1}\boldsymbol{s} + \frac{1}{2}\log\left(\det\left[\boldsymbol{S}\right]\right)$$
$$+ (\boldsymbol{\alpha} - \mathbf{1})^\dagger\boldsymbol{\tau} + \boldsymbol{q}^\dagger\mathrm{e}^{-\boldsymbol{\tau}} + \frac{1}{2}\boldsymbol{\tau}^\dagger\boldsymbol{T}\boldsymbol{\tau}$$
$$+ (\boldsymbol{\beta} - \mathbf{1})^\dagger\boldsymbol{u} + \boldsymbol{\eta}^\dagger\mathrm{e}^{-\boldsymbol{u}}$$

$$\boldsymbol{S} = \sum_k \mathrm{e}^{\tau_k}\boldsymbol{S}_k$$

IFT as neural network



NIFTy5

NIFTy

NIFTy – Numerical Information Field Theory

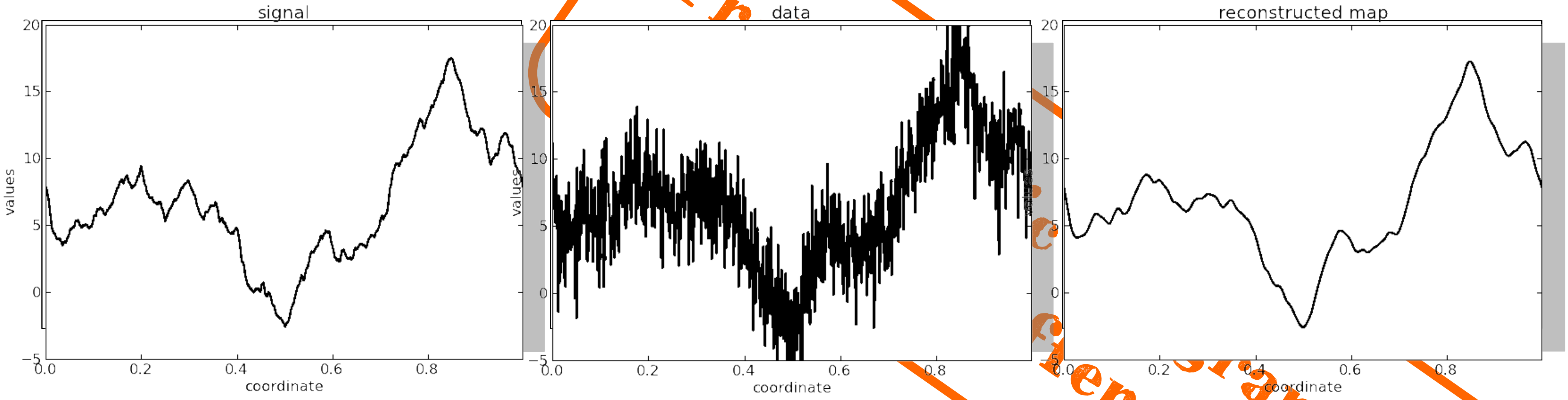
NIFTy [1], [2], “Numerical Information Field Theory” is a versatile library designed to enable the development of signal inference algorithms that are independent of the underlying grids (spatial, spectral, temporal, …) and their resolutions. Its object-oriented framework is written in Pytho

Probabilistic programming with auto-differentiation

NIFTy5

NIFTy – Numerical Information Field Theory

NIFTy [1], [2], “Numerical Information Field Theory” is a versatile library designed to enable the development of signal inference algorithms that are independent of the underlying grids (spatial, spectral, temporal, …) and their resolutions. Its object-oriented framework is written in Pytho…



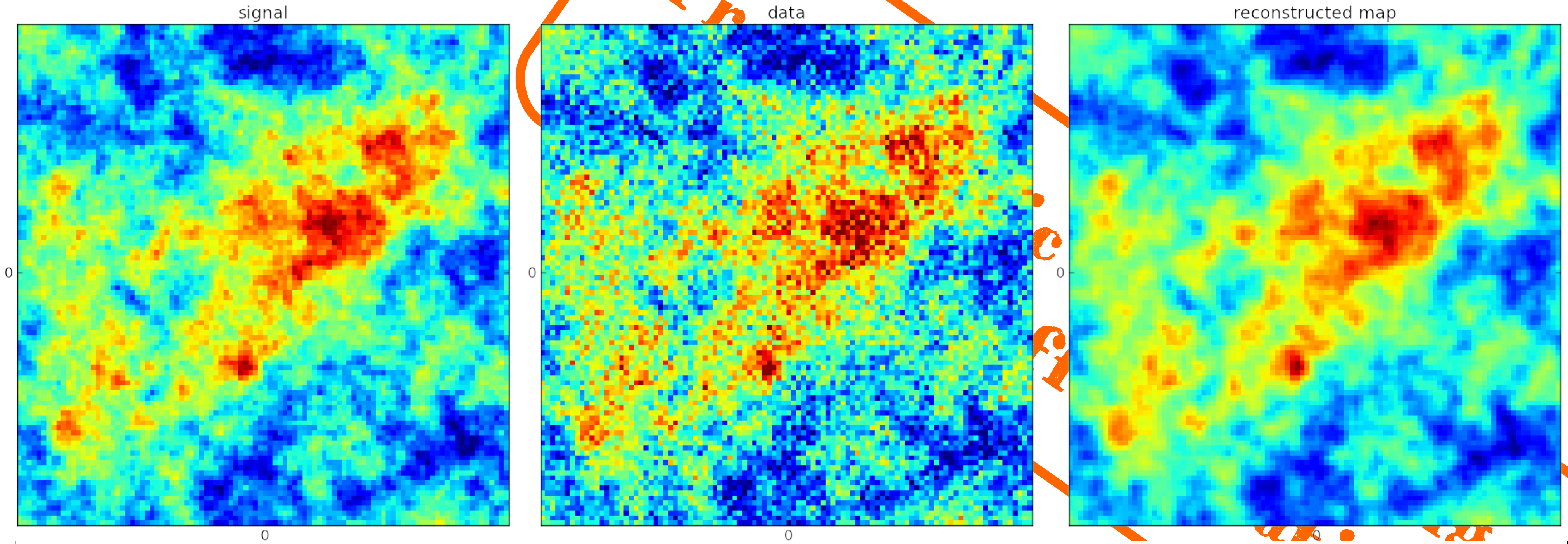
Differentiation … Programming

```
import nifty5 as ift
s_space = ift.RGSpace([N])
```

NIFTy5

NIFTy – Numerical Information Field Theory

NIFTy [1], [2], "Numerical Information Field Theory" is a versatile library designed to enable the development of signal inference algorithms that are independent of the underlying grids (spatial, spectral, temporal, …) and their resolutions. Its object-oriented framework is written in Pytho

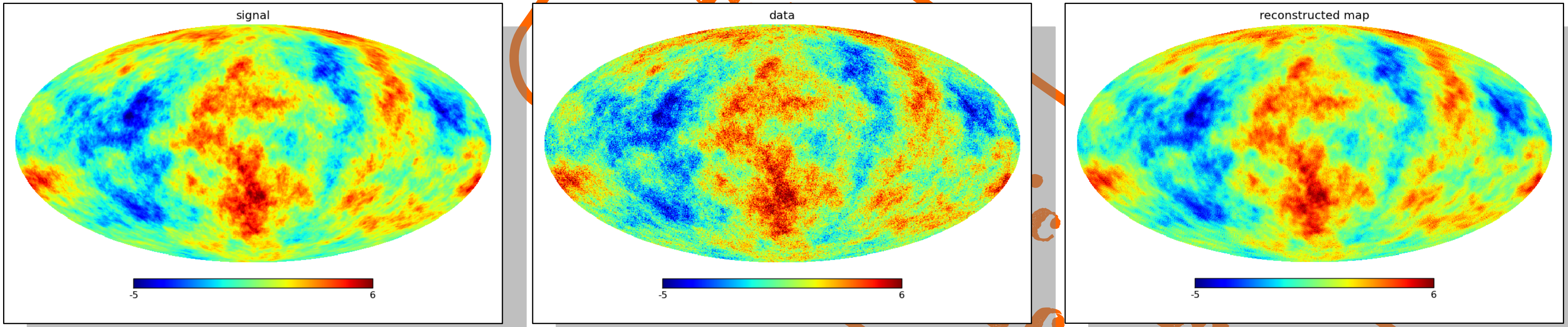



```
import nifty5 as ift
s_space = ift.RGSpace([N,N])
```

NIFTy5

NIFTy – Numerical Information Field Theory

NIFTy [1], [2], “Numerical Information Field Theory” is a versatile library designed to enable the development of signal inference algorithms that are independent of the underlying grids (spatial, spectral, temporal, …) and their resolutions. Its object-oriented framework is written in Python

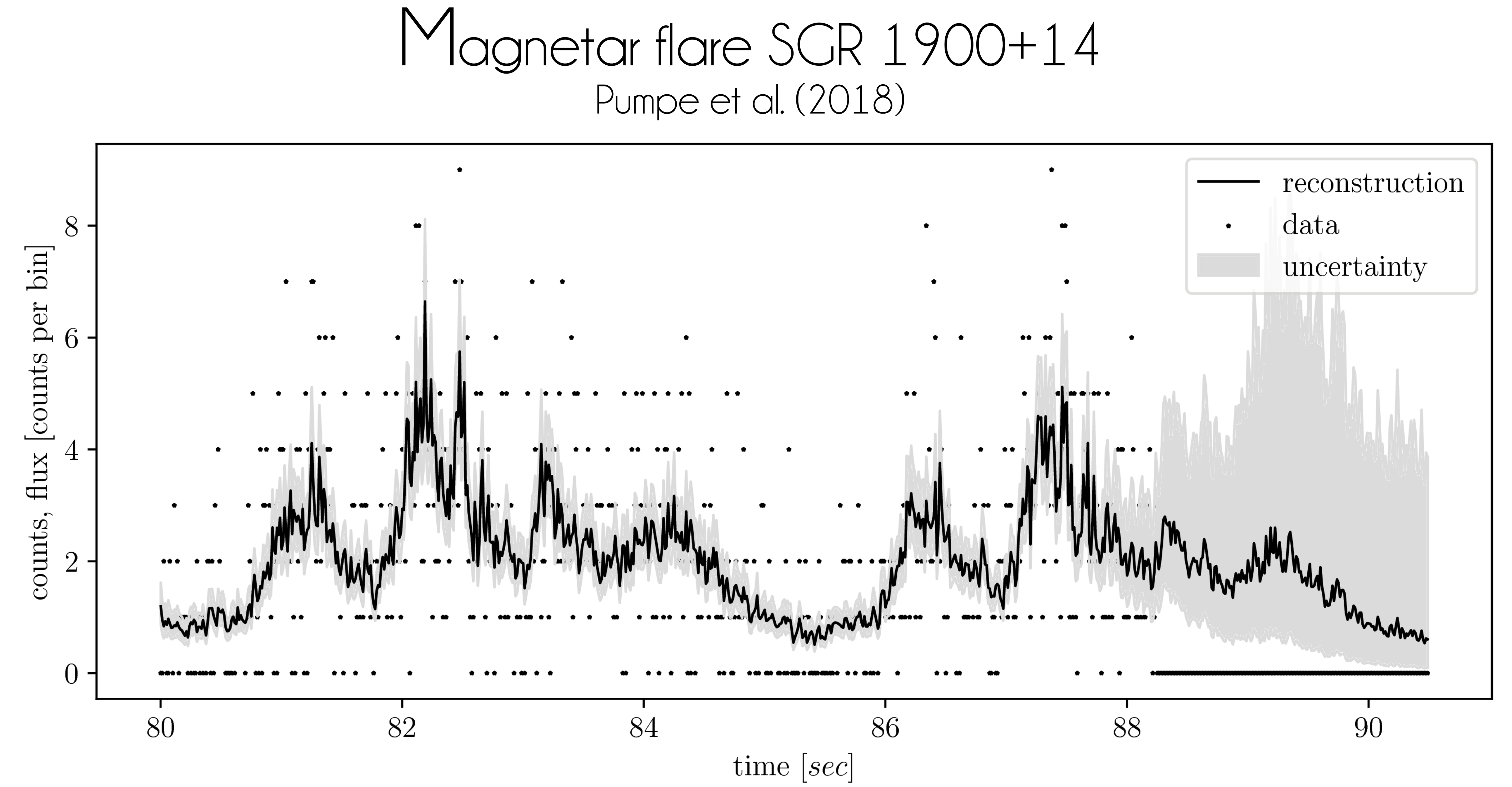



```
import nifty5 as ift
s_space = ift.HPSpace(NSide)
```

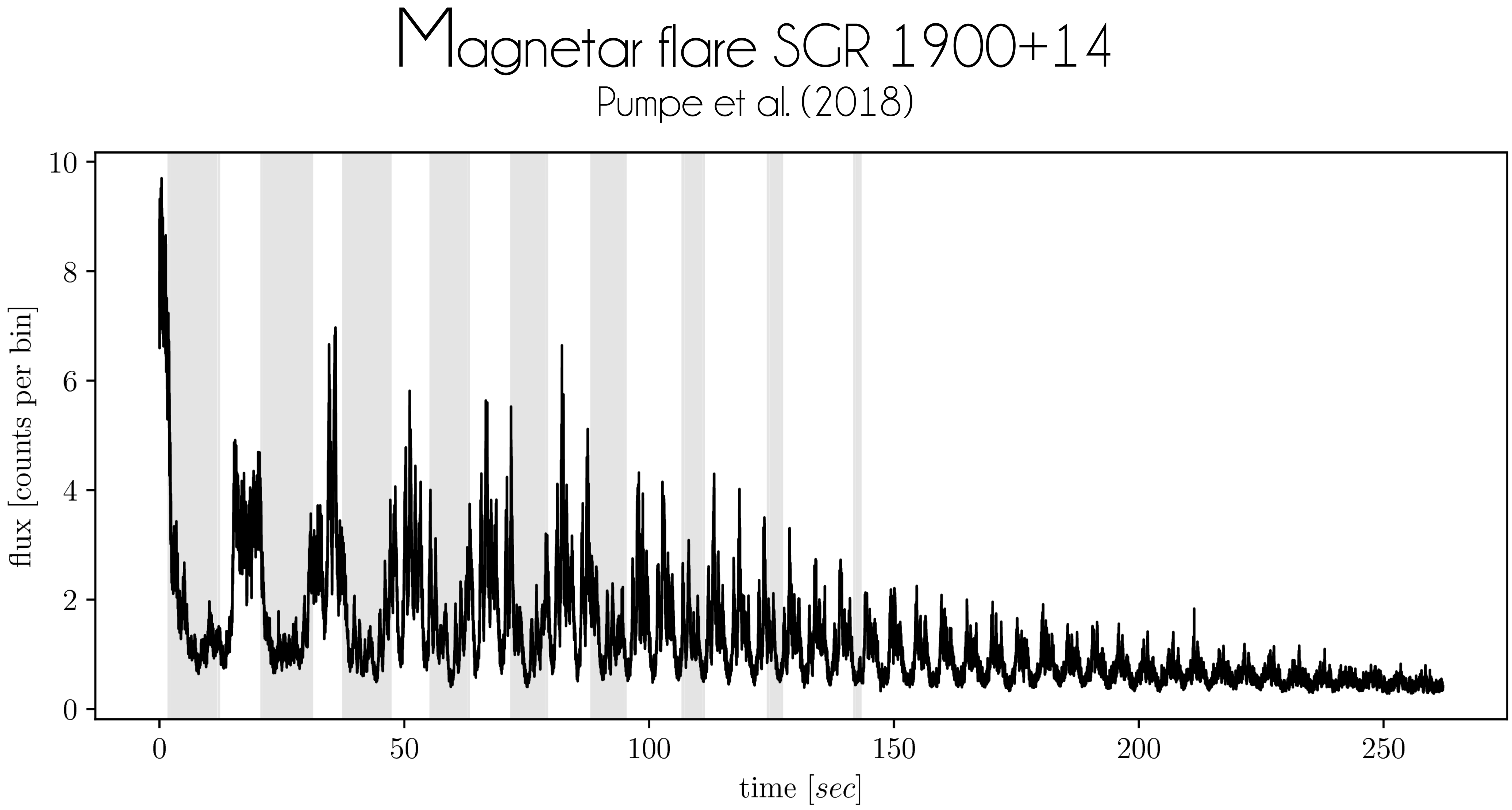
Information

$$
\begin{aligned}
\mathcal{H}(\boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\tau}) &= -\log \mathcal{P}(\boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\tau}) \\
&= 1^\dagger \left[\log(d!) + \boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right] - \boldsymbol{d}^\dagger \log\left[\boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right] \\
&\quad + \frac{1}{2}\boldsymbol{s}^\dagger \boldsymbol{S}^{-1}\boldsymbol{s} + \frac{1}{2}\log\left(\det\left[\boldsymbol{S}\right]\right) \\
&\quad + (\boldsymbol{\alpha} - \mathbf{1})^\dagger \boldsymbol{\tau} + \boldsymbol{q}^\dagger \mathrm{e}^{-\boldsymbol{\tau}} + \frac{1}{2}\boldsymbol{\tau}^\dagger \boldsymbol{T}\boldsymbol{\tau} \\
&\quad + (\boldsymbol{\beta} - \mathbf{1})^\dagger \boldsymbol{u} + \boldsymbol{\eta}^\dagger \mathrm{e}^{-\boldsymbol{u}}
\end{aligned}
$$

$$
\boldsymbol{S} = \sum_k \mathrm{e}^{\tau_k} \boldsymbol{S}_k
$$



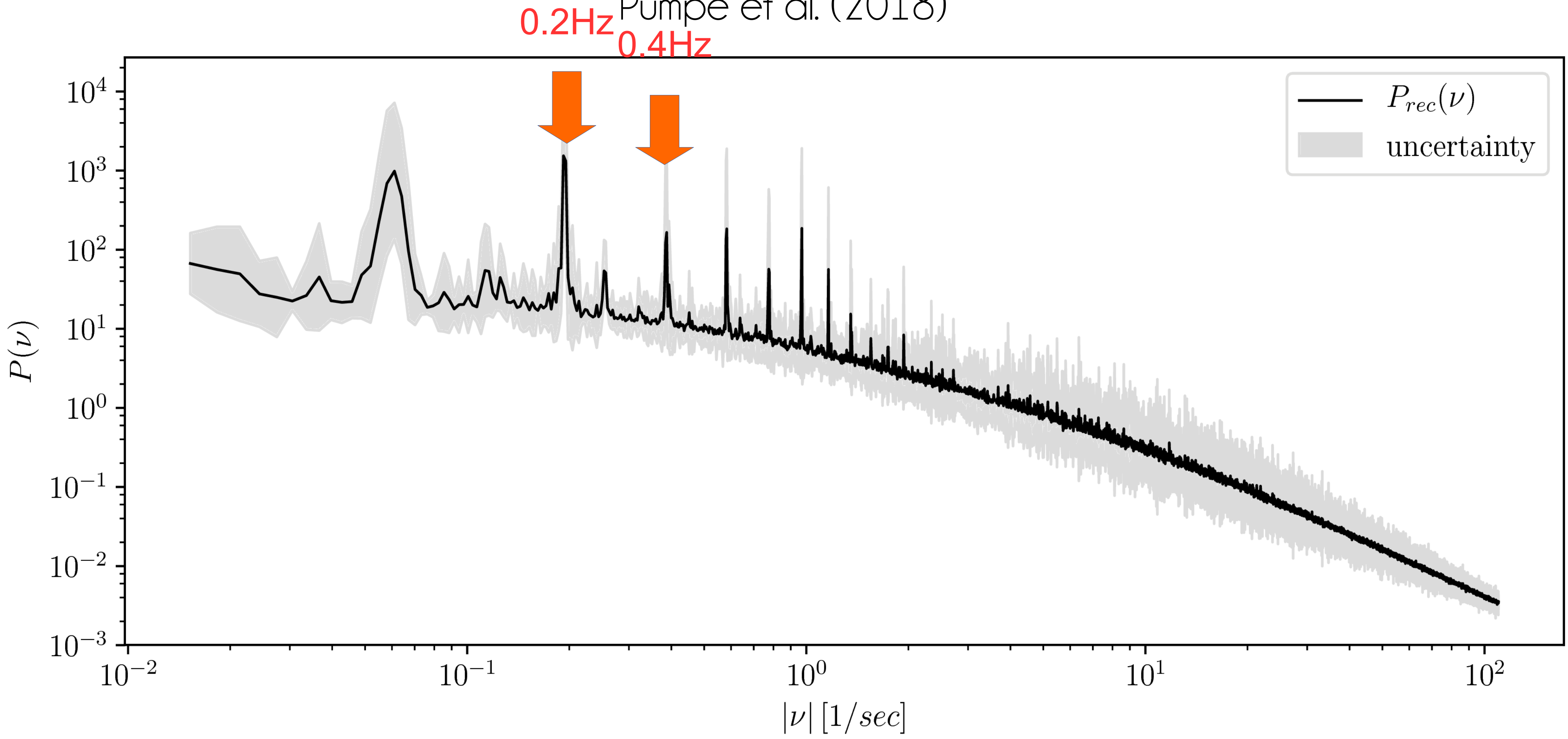
Magnetar flare SGR 1900+14
Pumpe et al. (2018)
reconstruction
data
uncertainty
counts, flux [counts per bin]
time [sec]
0
2
4
6
8
80
82
84
86
88
90



Magnetar flare SGR 1900+14
Pumpe et al. (2018)
flux [counts per bin]
time [sec]
0
2
4
6
8
10
0
50
100
150
200
250

Magnetar flare SGR 1900+14

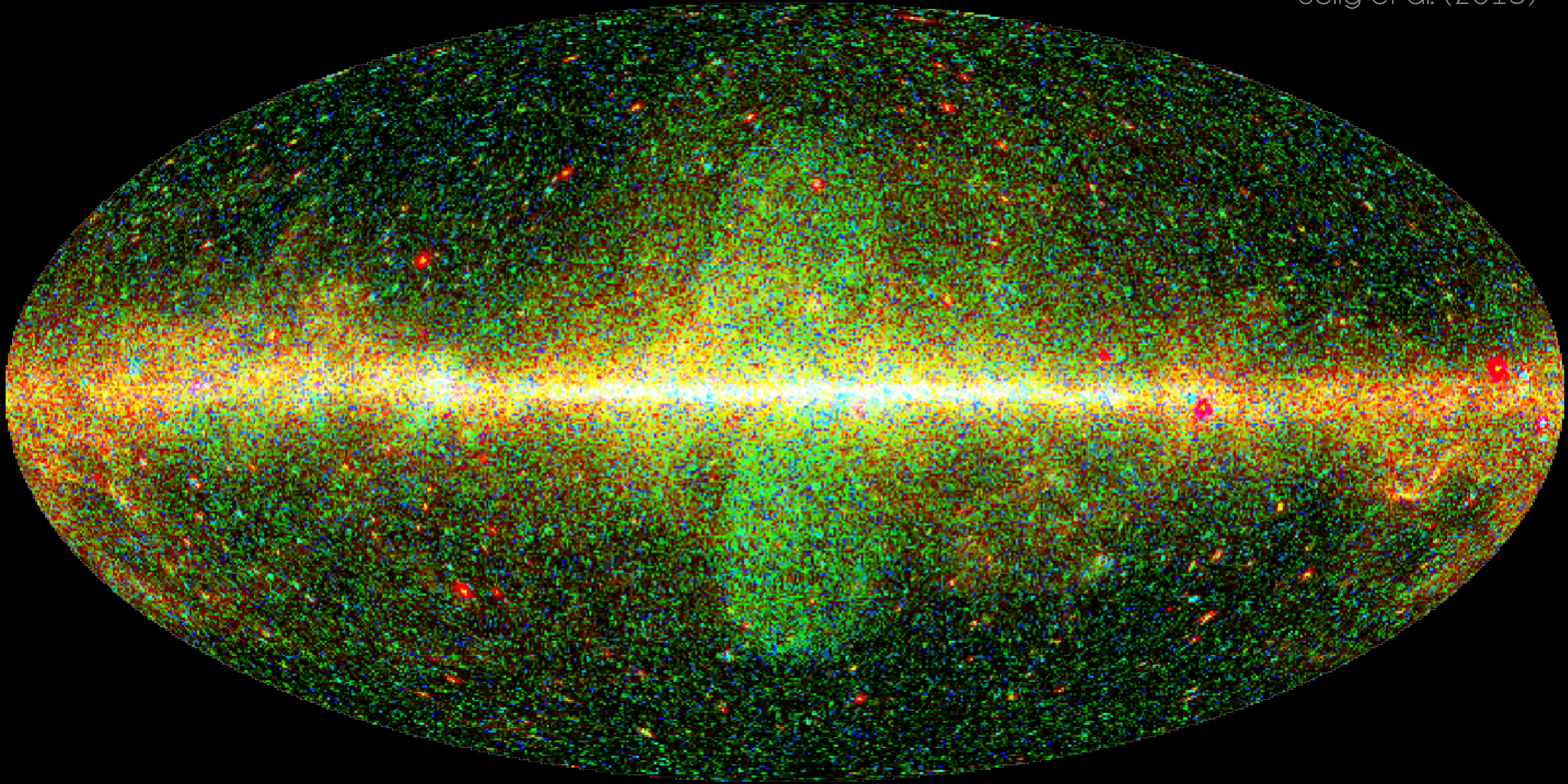
Pumpe et al. (2018)



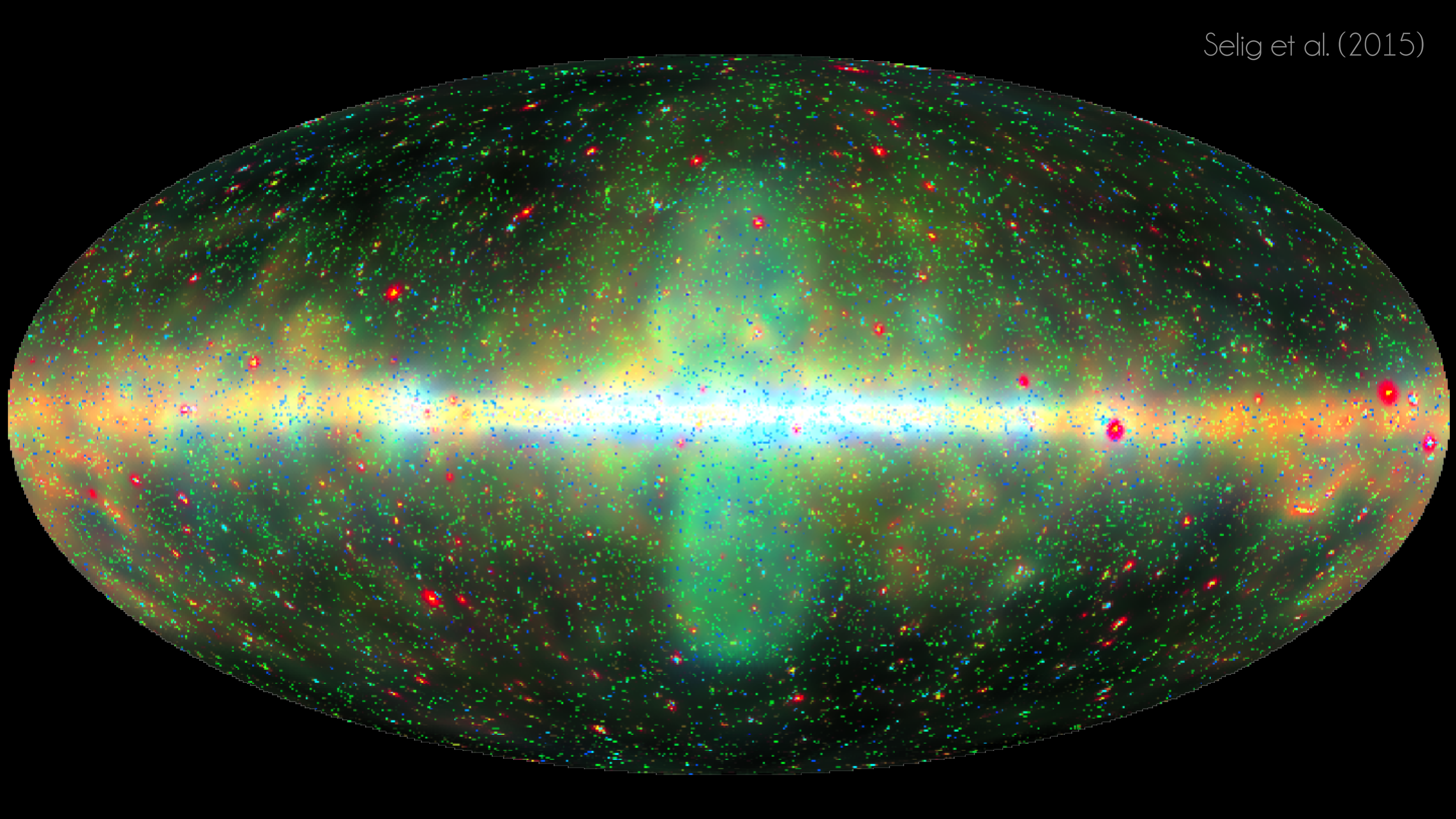
Information

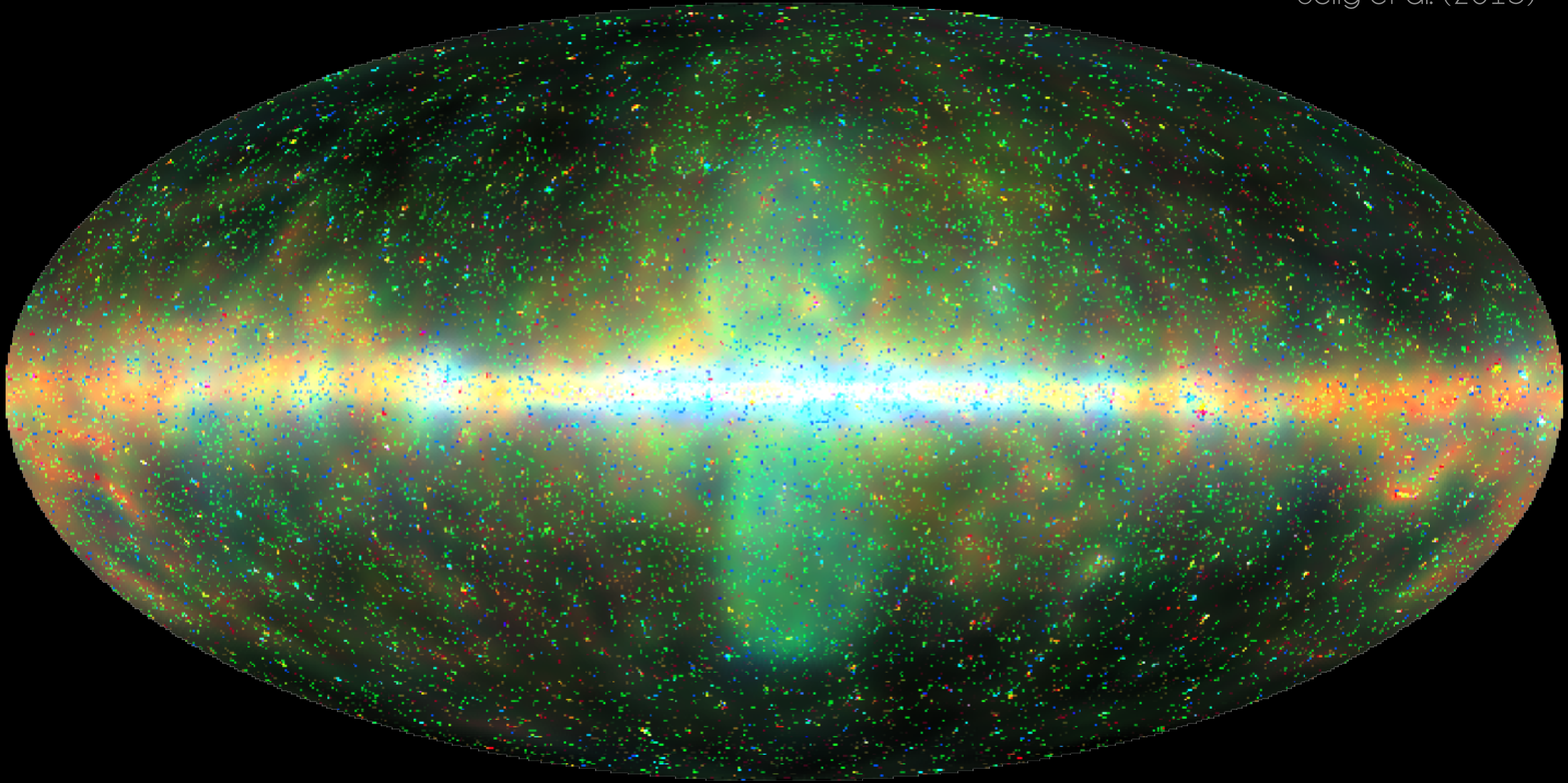
$$
\begin{aligned}
\mathcal{H}(\boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\tau}) &= -\log \mathcal{P}(\boldsymbol{d}, \boldsymbol{s}, \boldsymbol{\tau}) \\
&= 1^\dagger \left[\log(d!) + \boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right] - \boldsymbol{d}^\dagger \log\left[\boldsymbol{R}\left(\mathrm{e}^{\boldsymbol{s}} + \mathrm{e}^{\boldsymbol{u}}\right)\right] \\
&\quad + \frac{1}{2}\boldsymbol{s}^\dagger \boldsymbol{S}^{-1}\boldsymbol{s} + \frac{1}{2}\log\left(\det\left[\boldsymbol{S}\right]\right) \\
&\quad + (\boldsymbol{\alpha} - \mathbf{1})^\dagger \boldsymbol{\tau} + \boldsymbol{q}^\dagger \mathrm{e}^{-\boldsymbol{\tau}} + \frac{1}{2}\boldsymbol{\tau}^\dagger \boldsymbol{T}\boldsymbol{\tau} \\
&\quad + (\boldsymbol{\beta} - \mathbf{1})^\dagger \boldsymbol{u} + \boldsymbol{\eta}^\dagger \mathrm{e}^{-\boldsymbol{u}}
\end{aligned}
$$

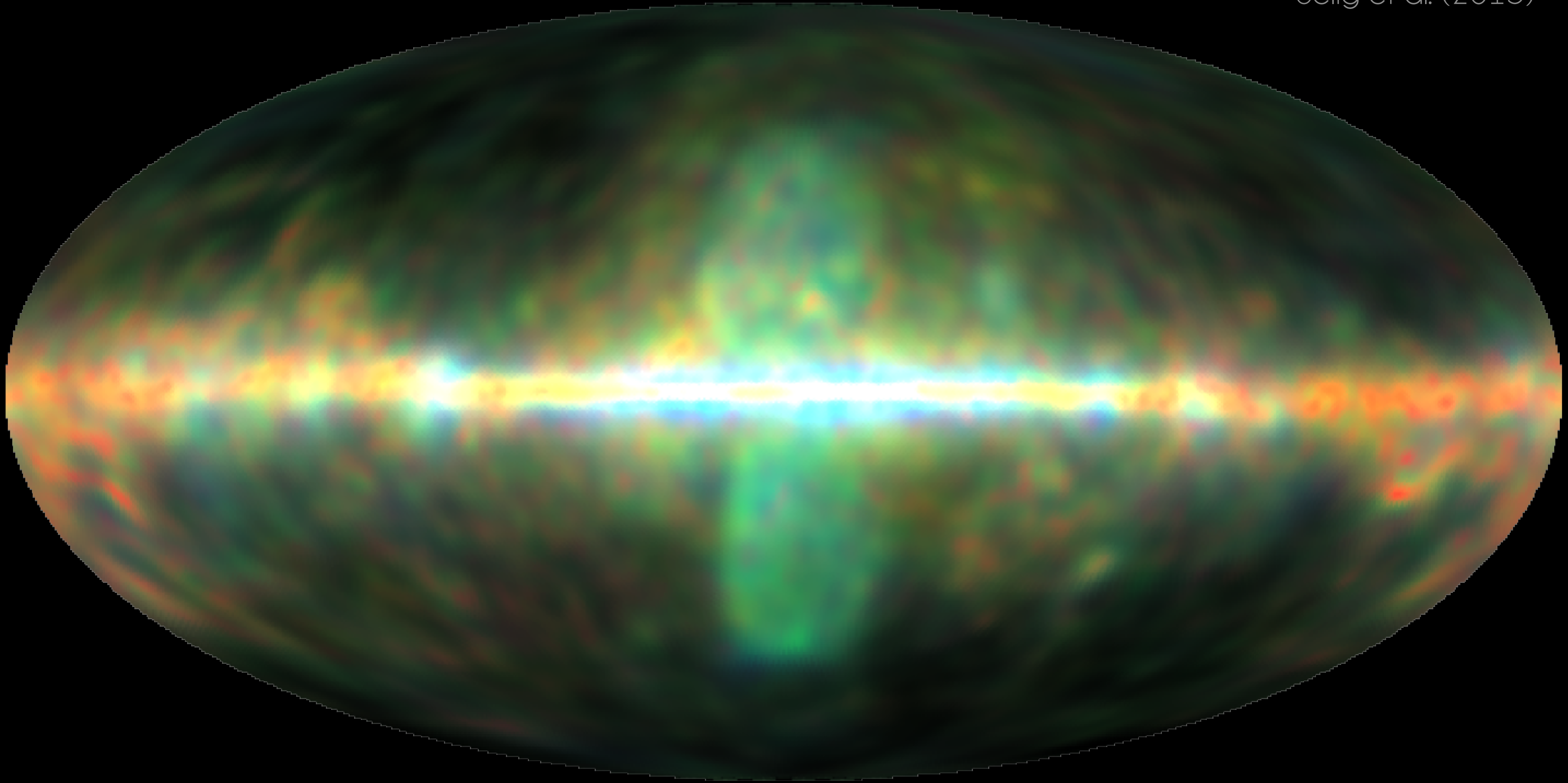
$$
\boldsymbol{S} = \sum_k \mathrm{e}^{\tau_k} \boldsymbol{S}_k
$$



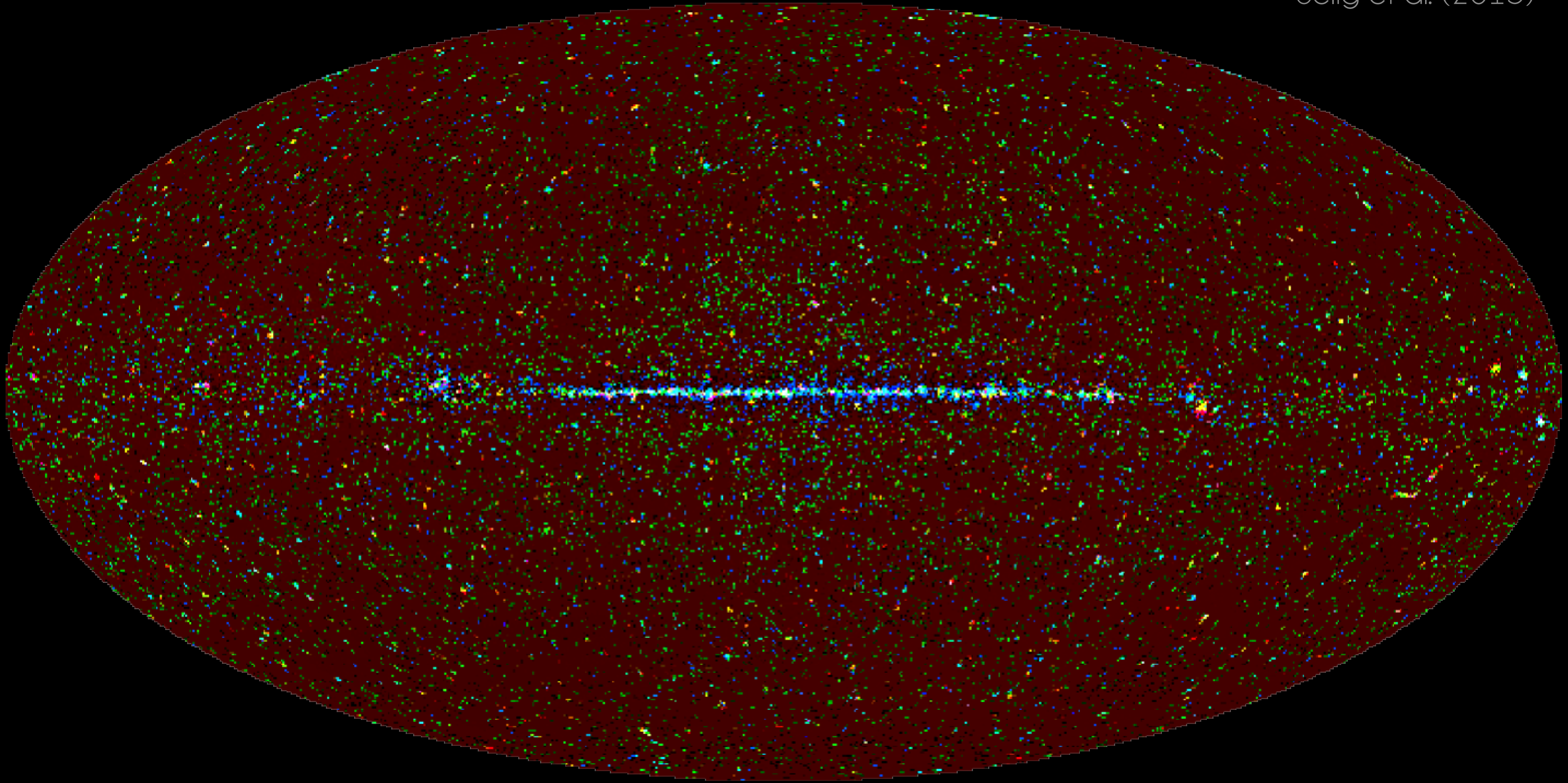
Selig et al. (2015)


Selig et al. (2015)

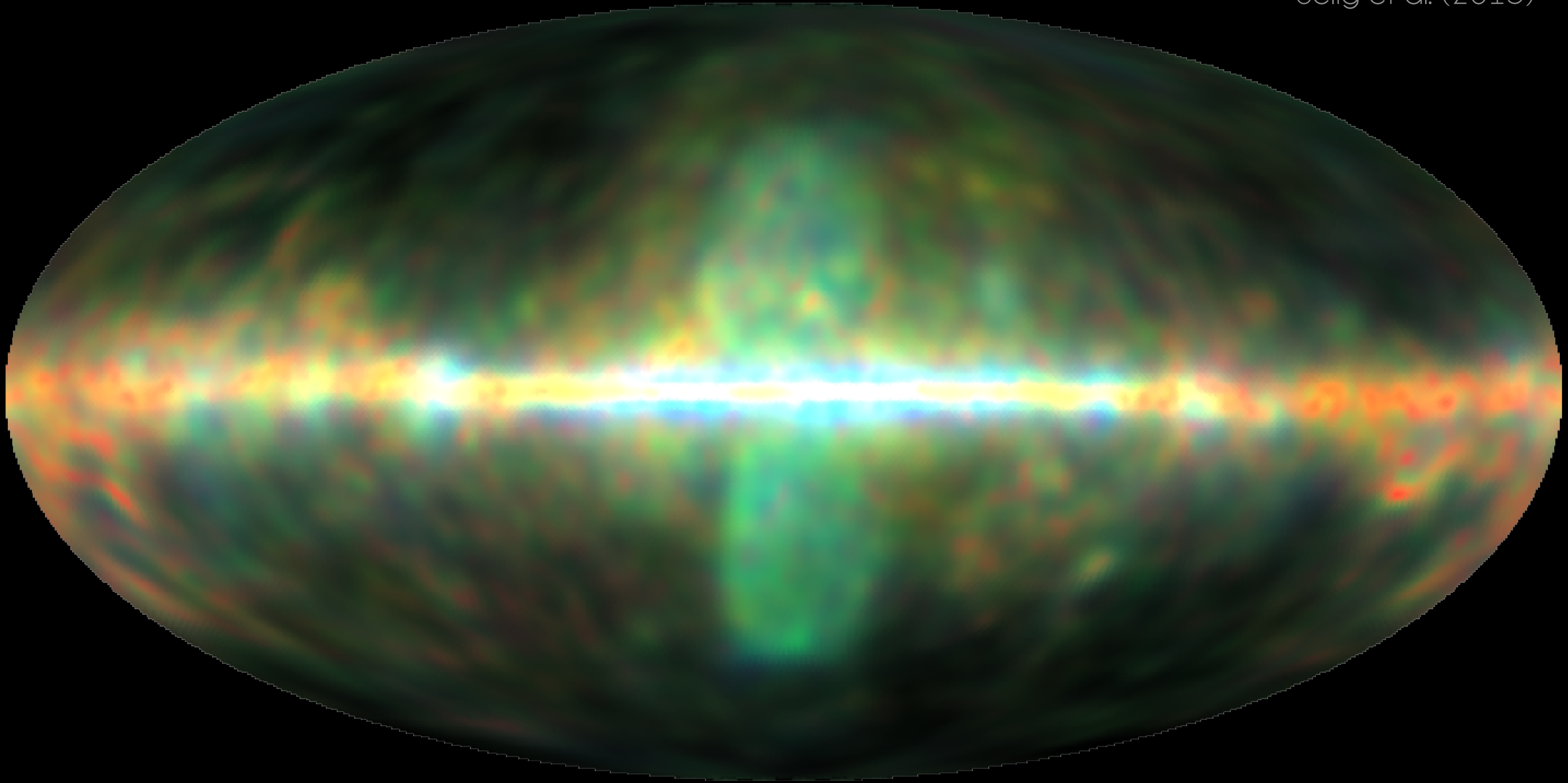

Selig et al. (2015)



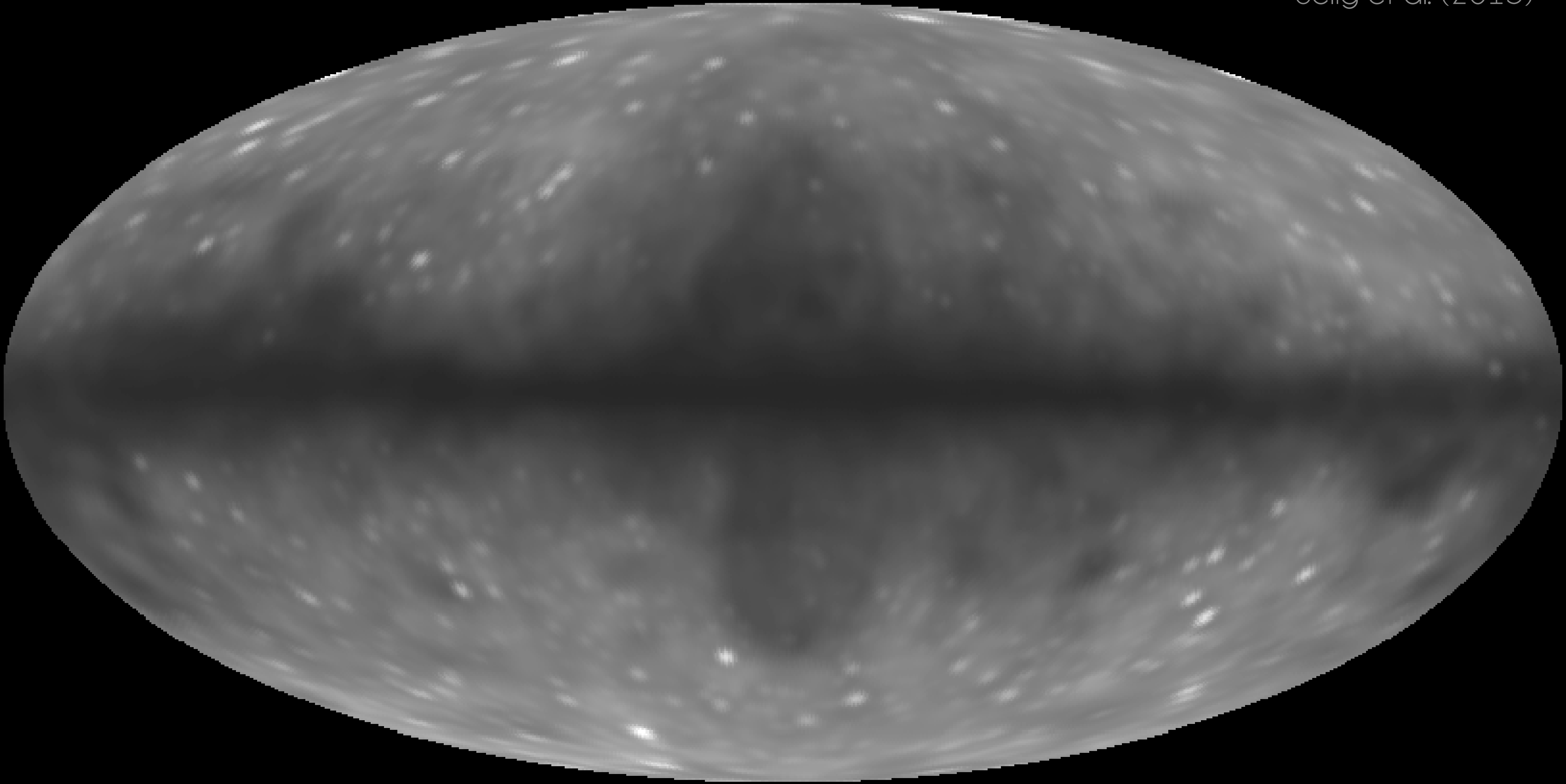
Selig et al. (2015)



Selig et al. (2015)

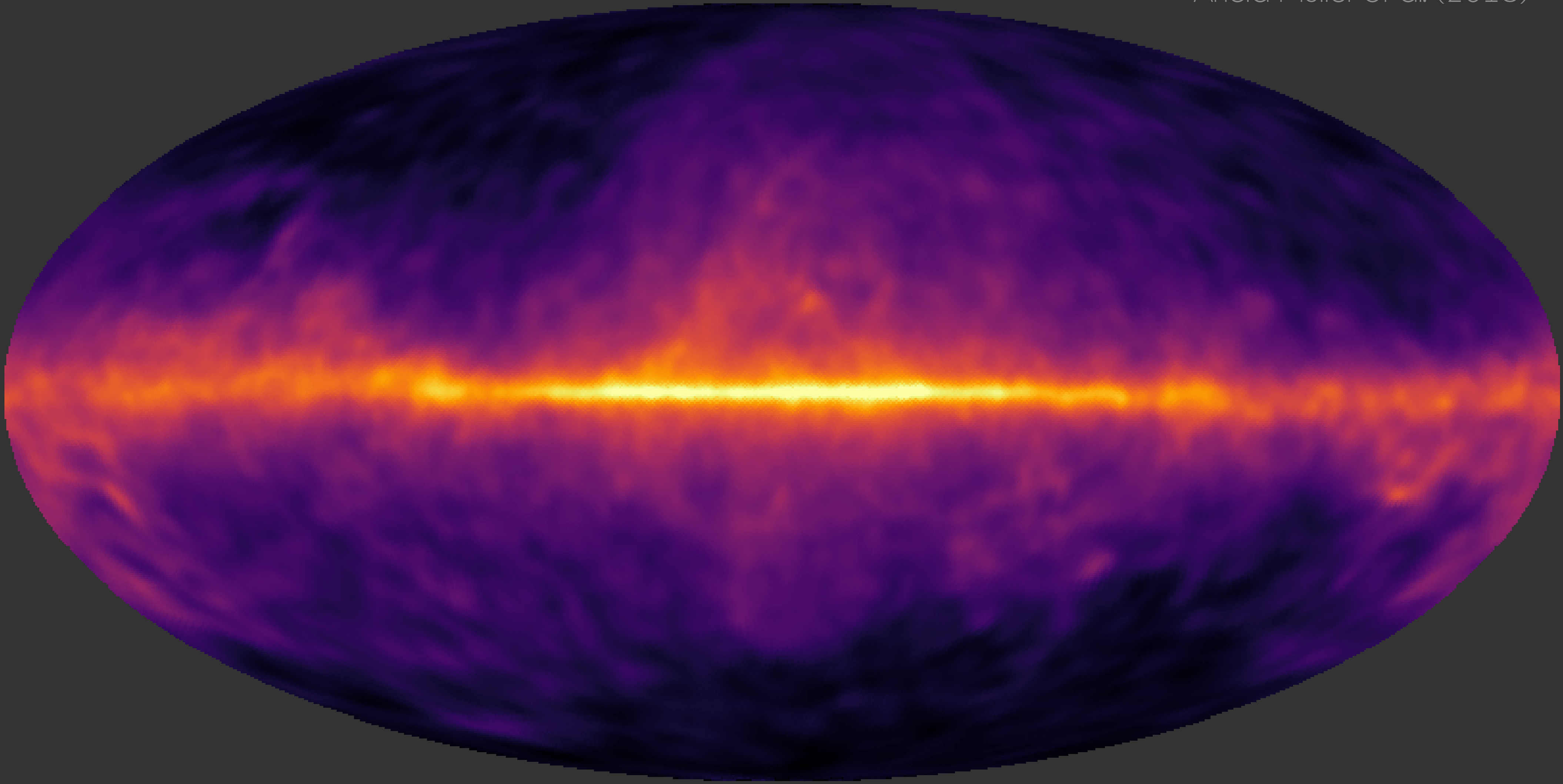


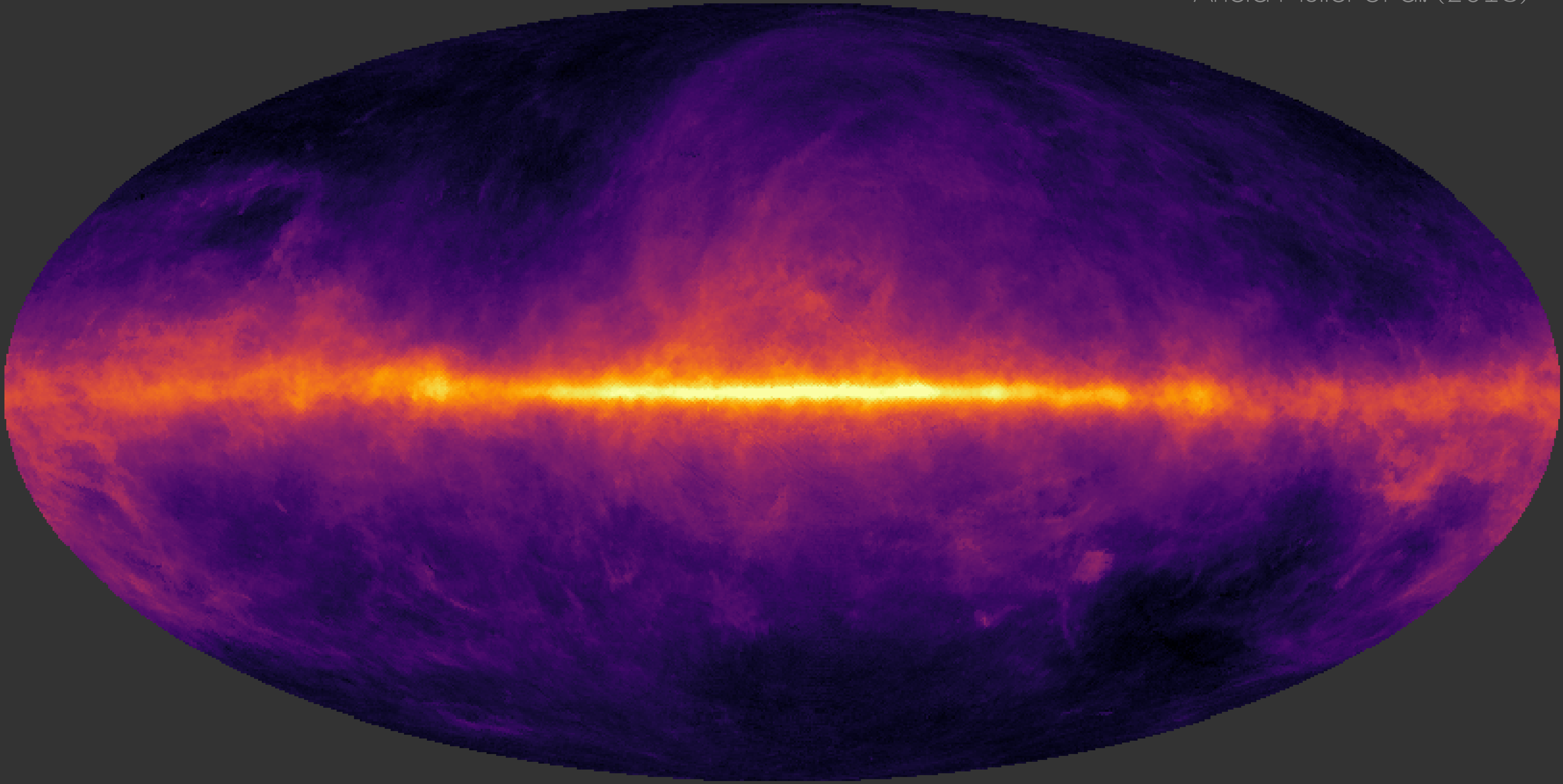
Selig et al. (2015)



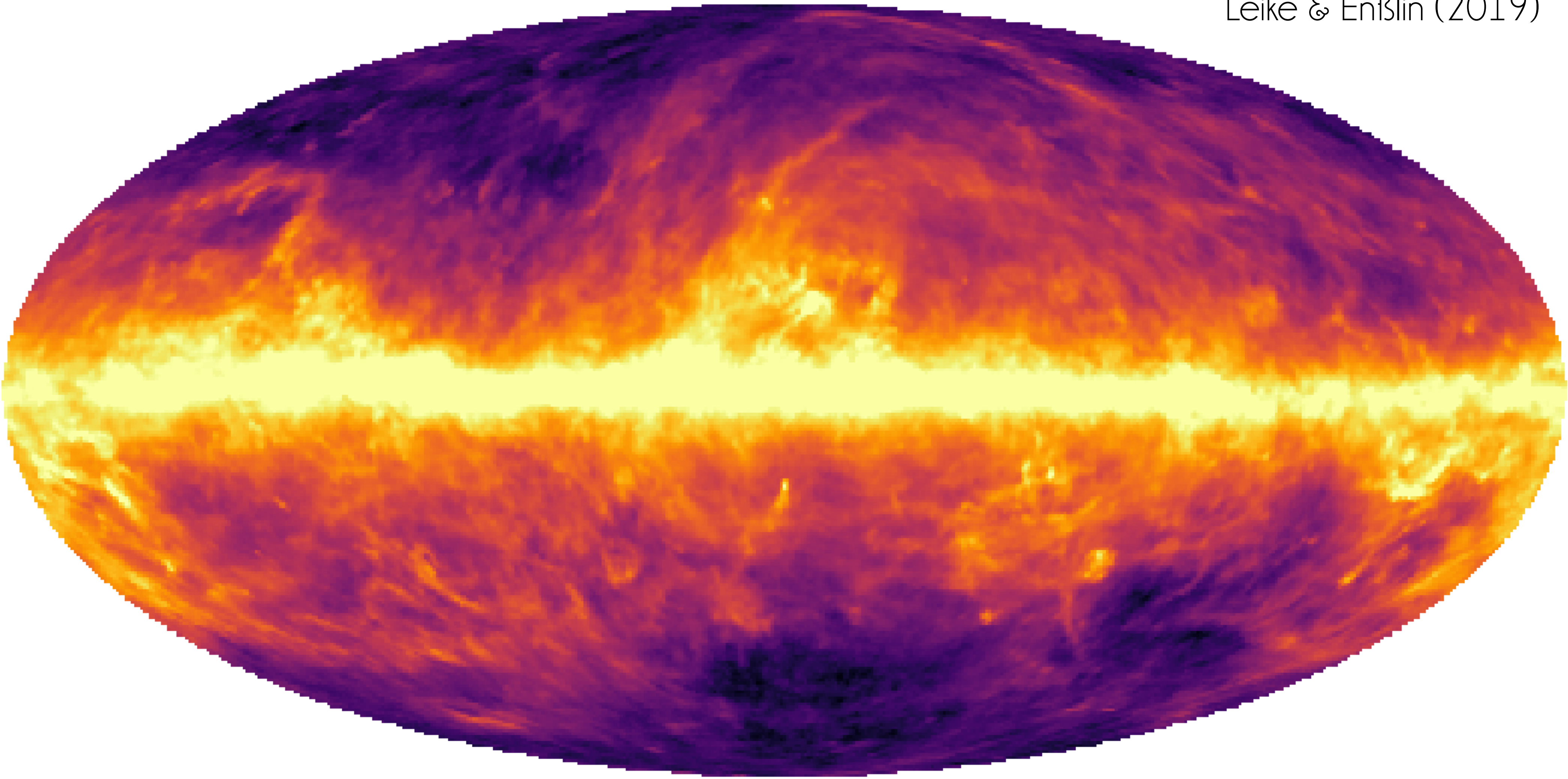
Selig et al. (2015)

Ancla Müller et al. (2018)



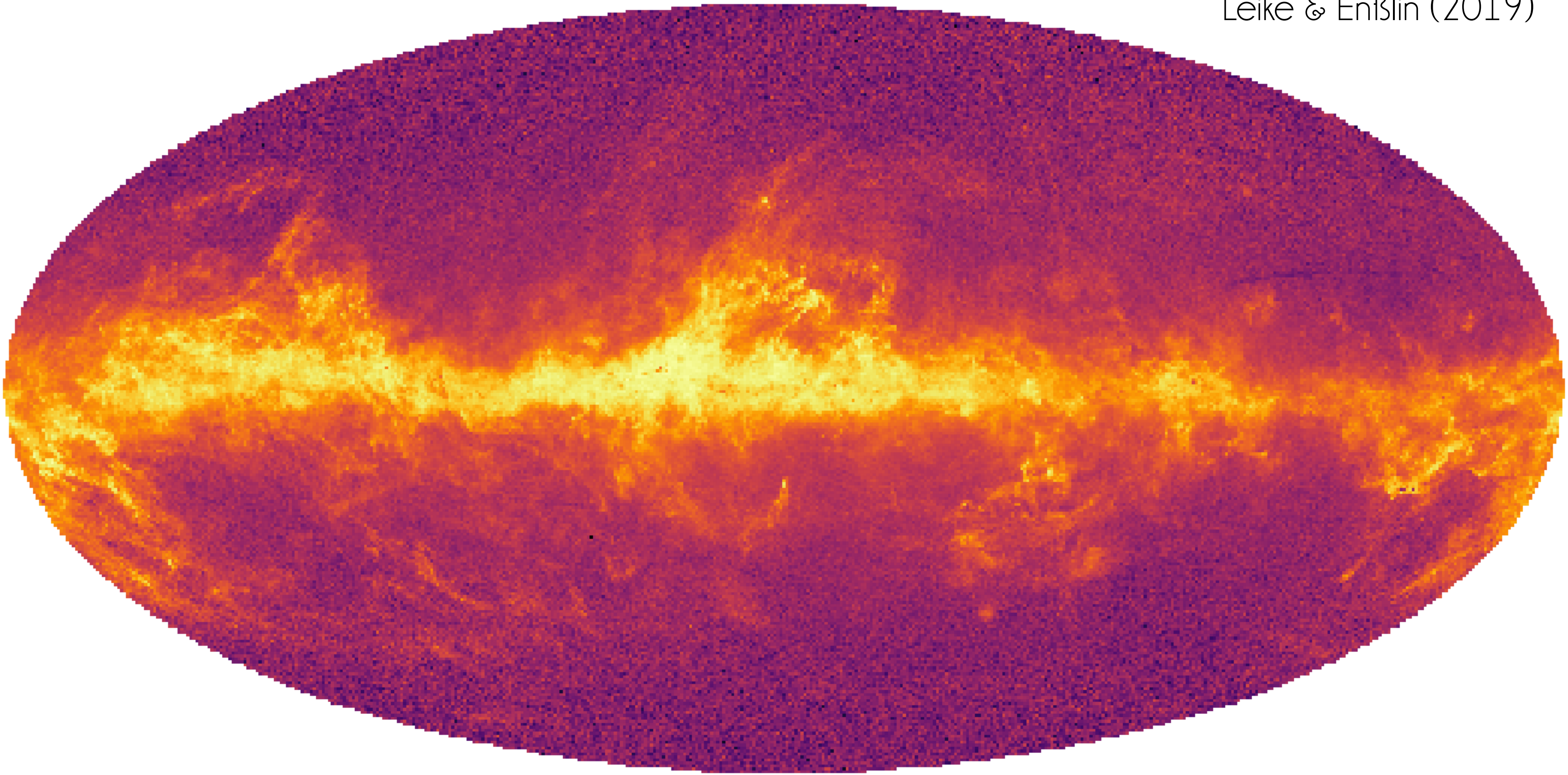


Ancla Müller et al. (2018)

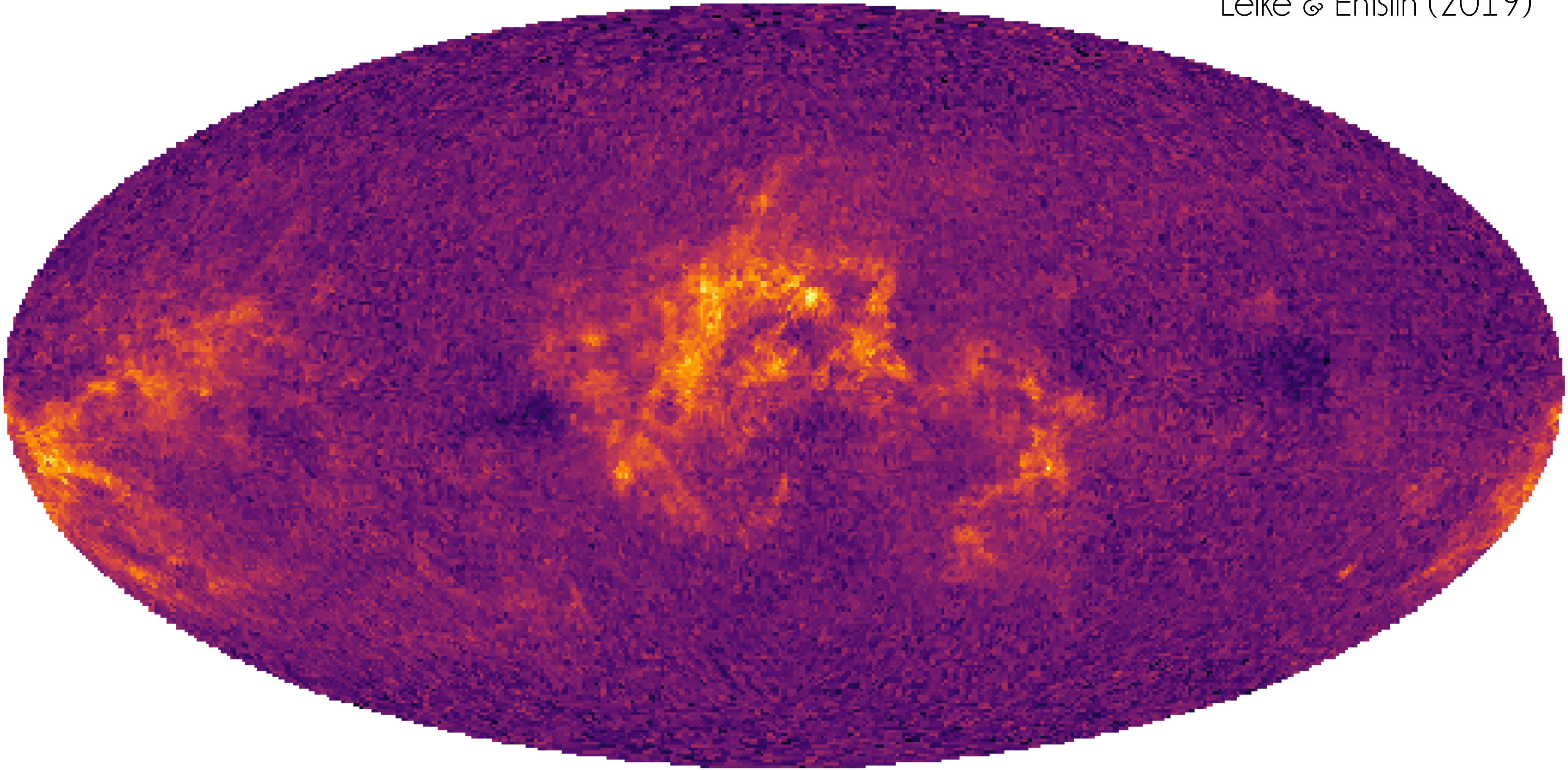


dust emission by Planck
Leike & Enßlin (2019)

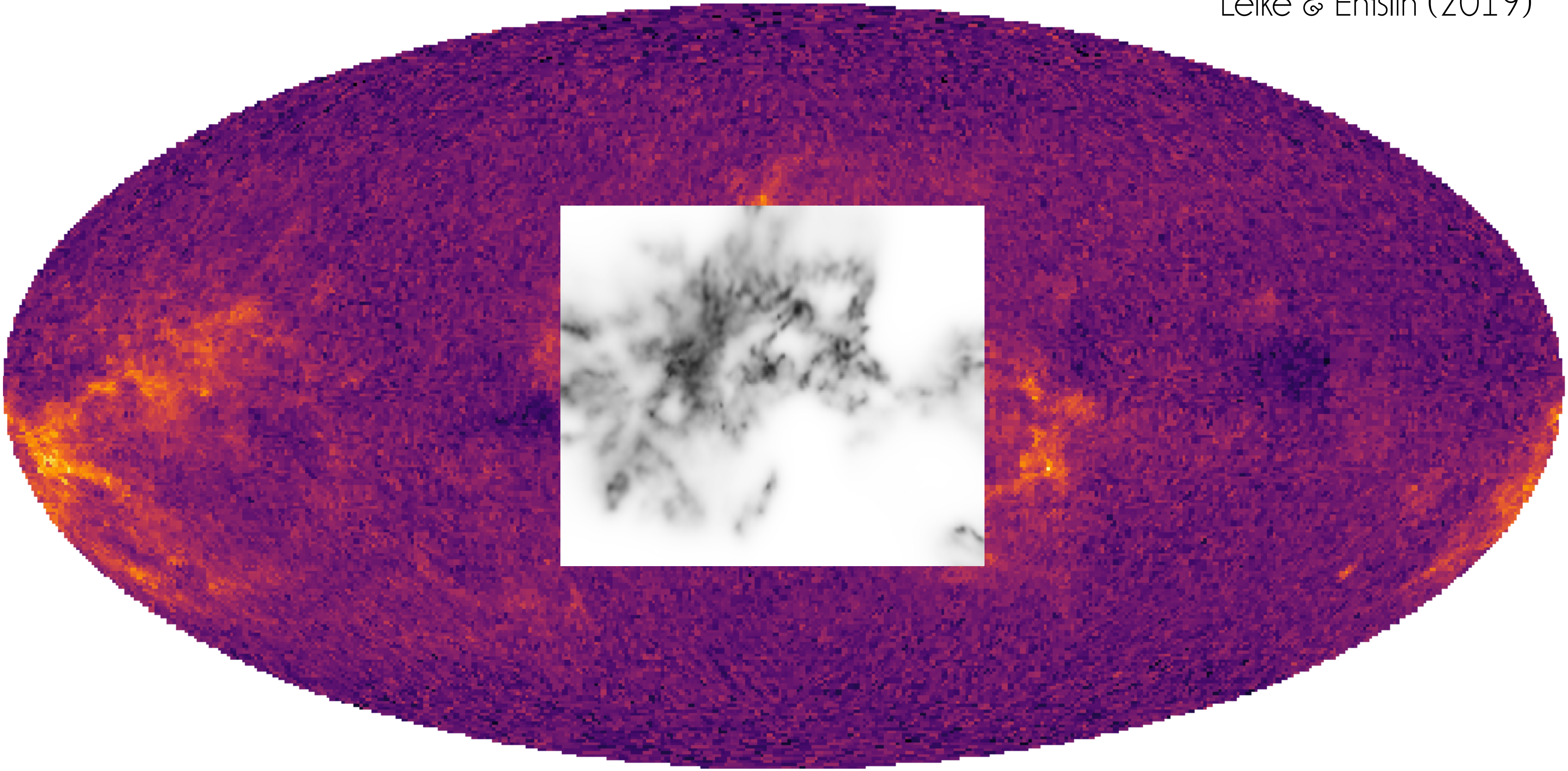
dust absorption by Gaia
Leike & Enßlin (2019)

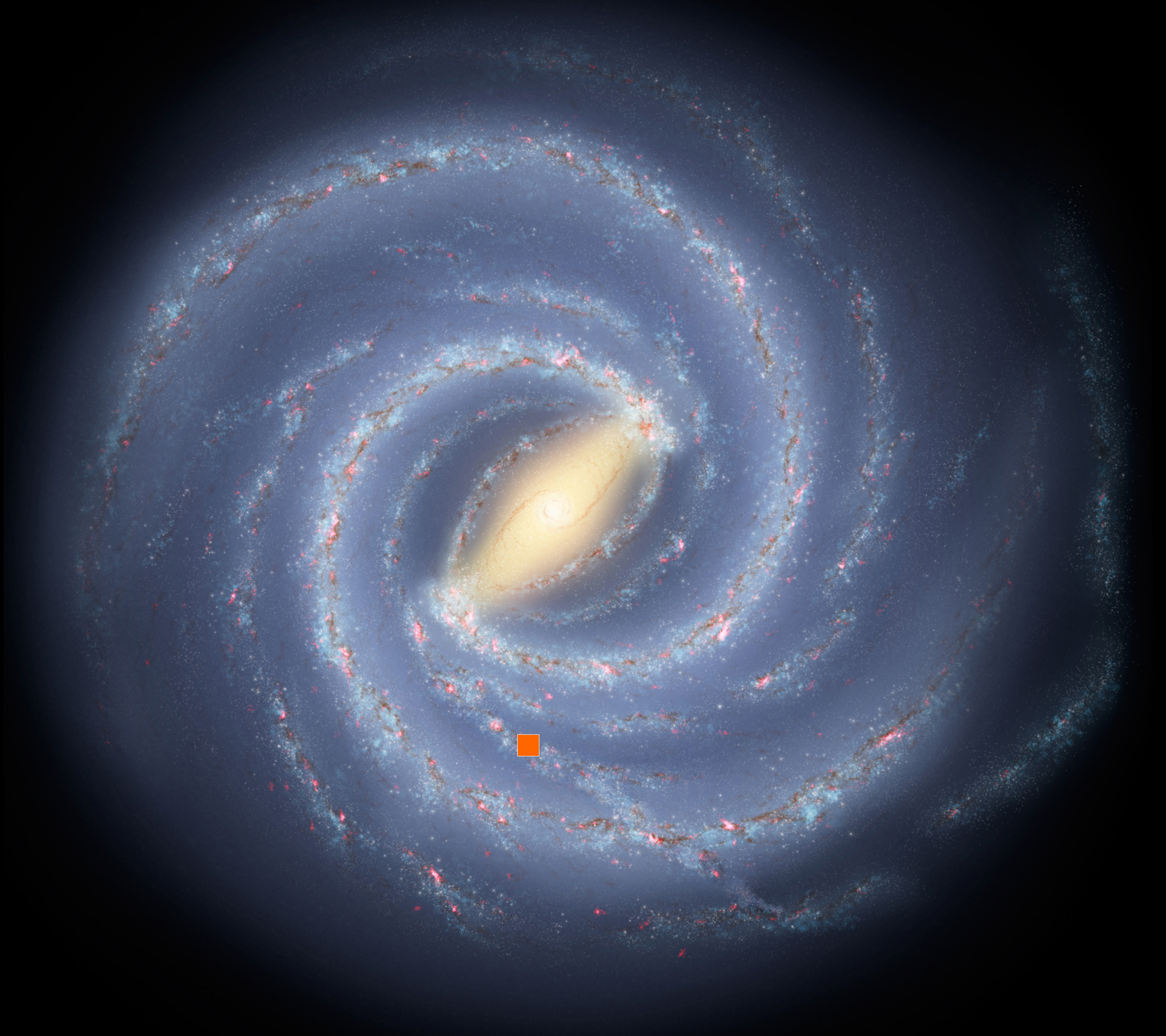


dust absorption by Gaia
Leike & Enßlin (2019)

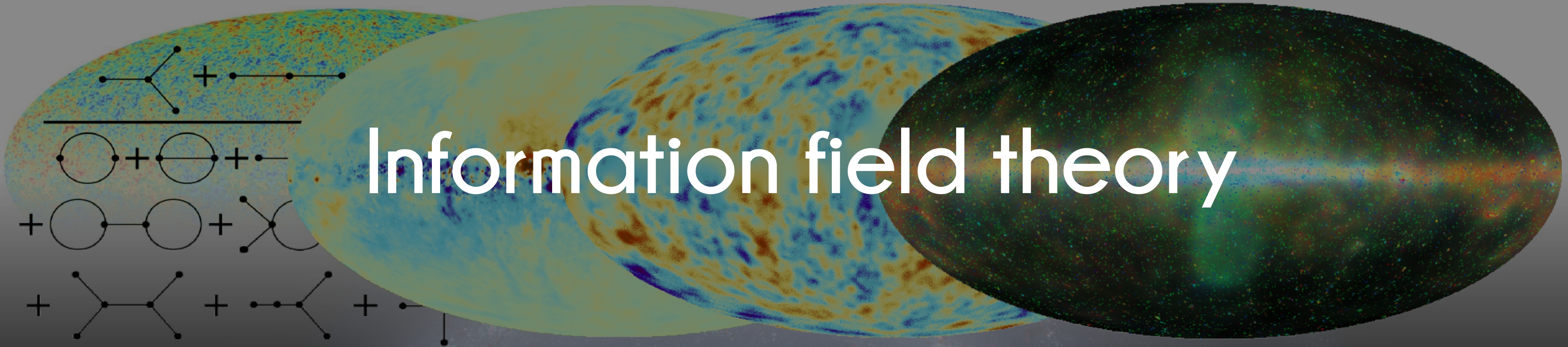


dust absorption by Gaia
Leike & Enßlin (2019)





Information field theory



Field inference

IFT = information theory for fields

IFT is fully Bayesian

IFT exploits & learns signal correlations & other properties

IFT fuses machine learning & human knowledge

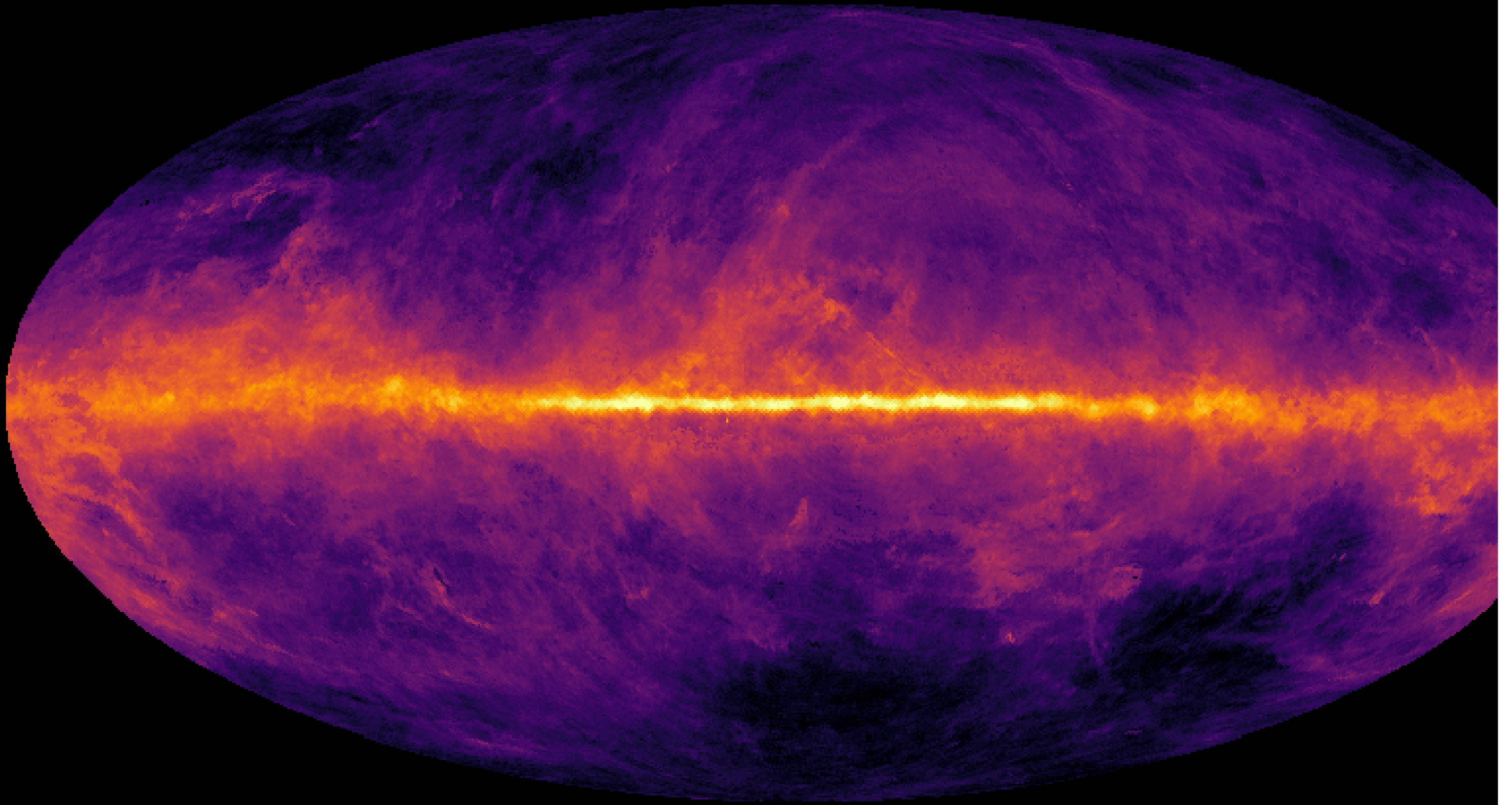
NIFTy

Numerical IFT

Reconstruction of signal fields over Cartesian and spherical spaces and products thereof

NIFTy algorithms are special purpose NN that do not require extra training

Unified imaging UBIK



Thank you!

www.mpa-garching.mpg.de/ift