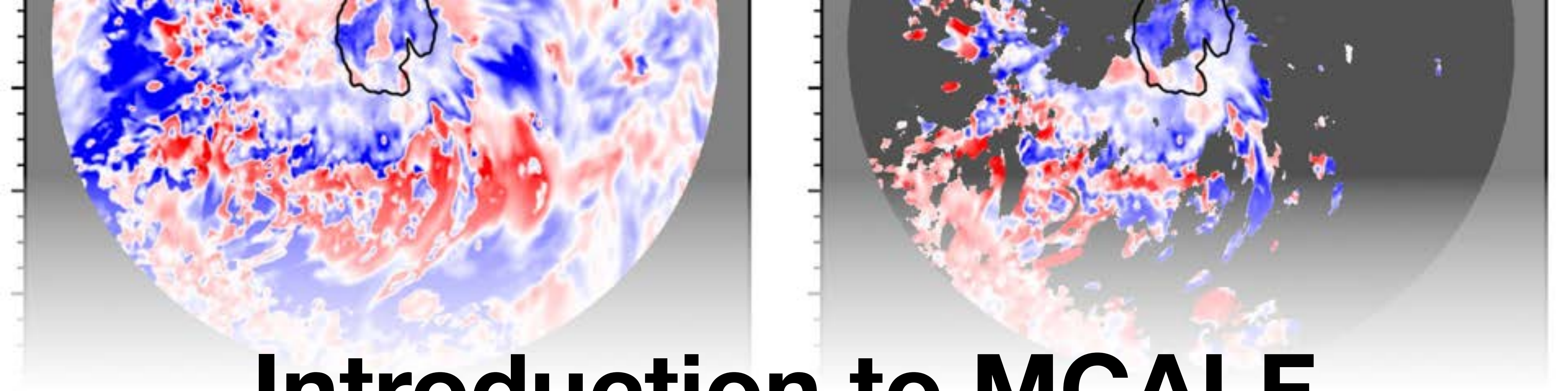


Introduction to MCALF

Multi-Component Atmospheric Line Fitting

Conor MacBride

PhD Student in Solar Physics at Queen's University Belfast

Conor MacBride

PhD Student, Queen's University Belfast

✉ cmacbride01@qub.ac.uk | conor@macbride.me

macbride.me

ConorMacBride

Queen's University Belfast

Solar Physics PhD

Sep 2019 —

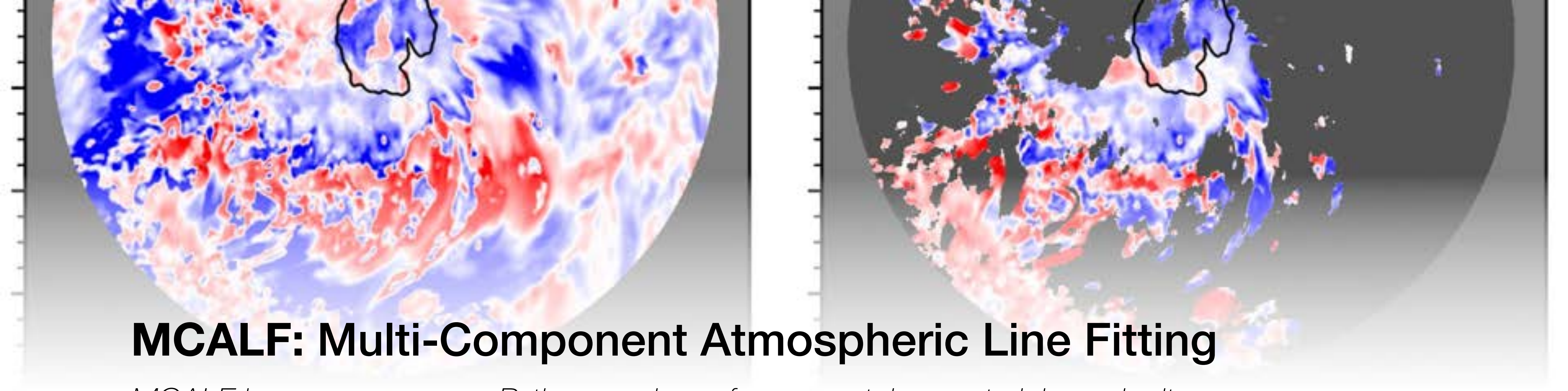
Energy Dissipation in Solar Physics

Supervisor: David Jess

University of St Andrews

MPhys Mathematics and Theoretical Physics

Sep 2015 — Jun 2019



MCALF: Multi-Component Atmospheric Line Fitting

MCALF is an open-source Python package for accurately constraining velocity information from spectral imaging observations using machine learning techniques.

Conor MacBride
PhD Student, Queen's University Belfast

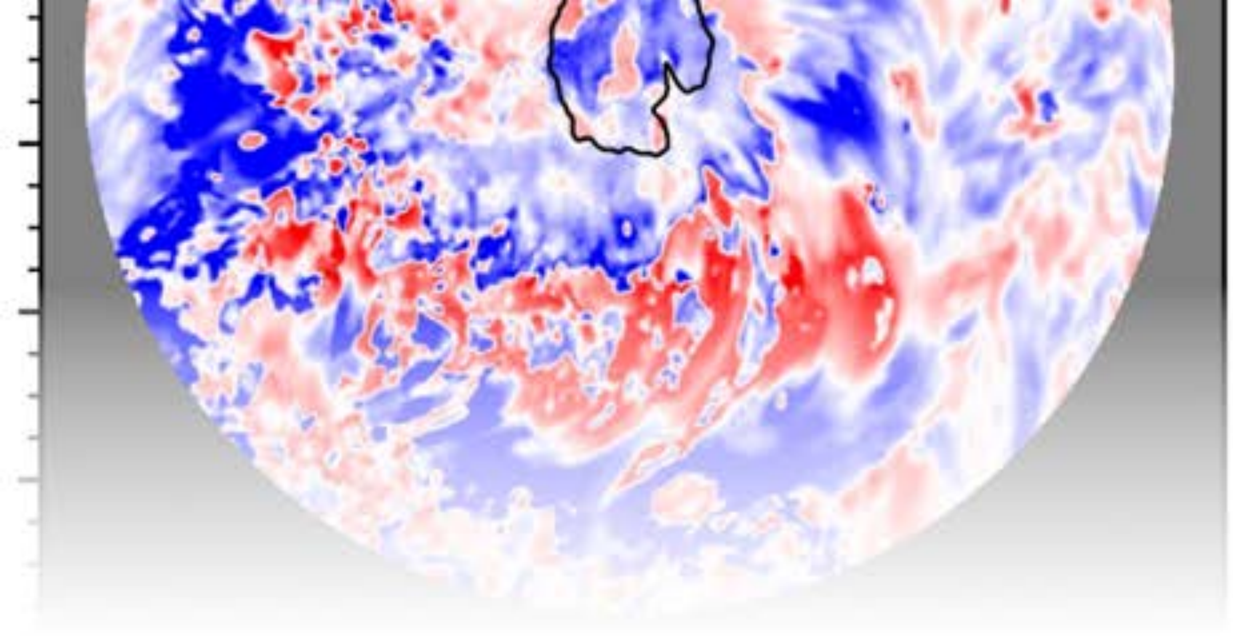
David Jess
Reader, Queen's University Belfast

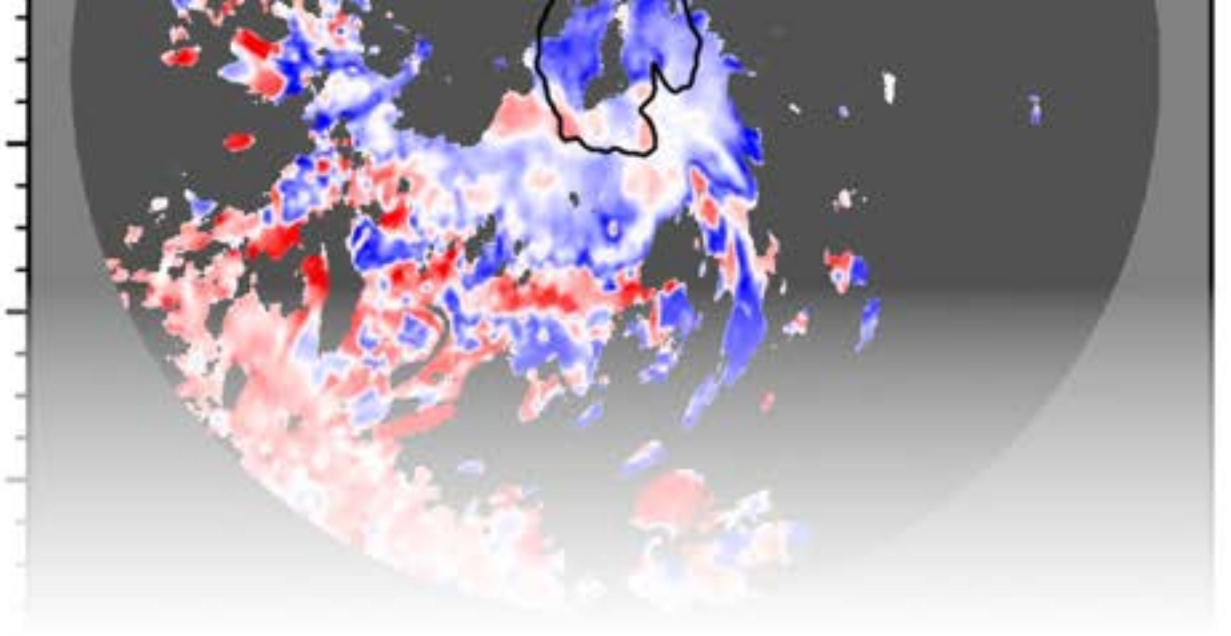
GitHub
github.com/ConorMacBride/mcalf

Documentation
mcalf.macbride.me

Install

```
pip install mcalf
conda install mcalf
```

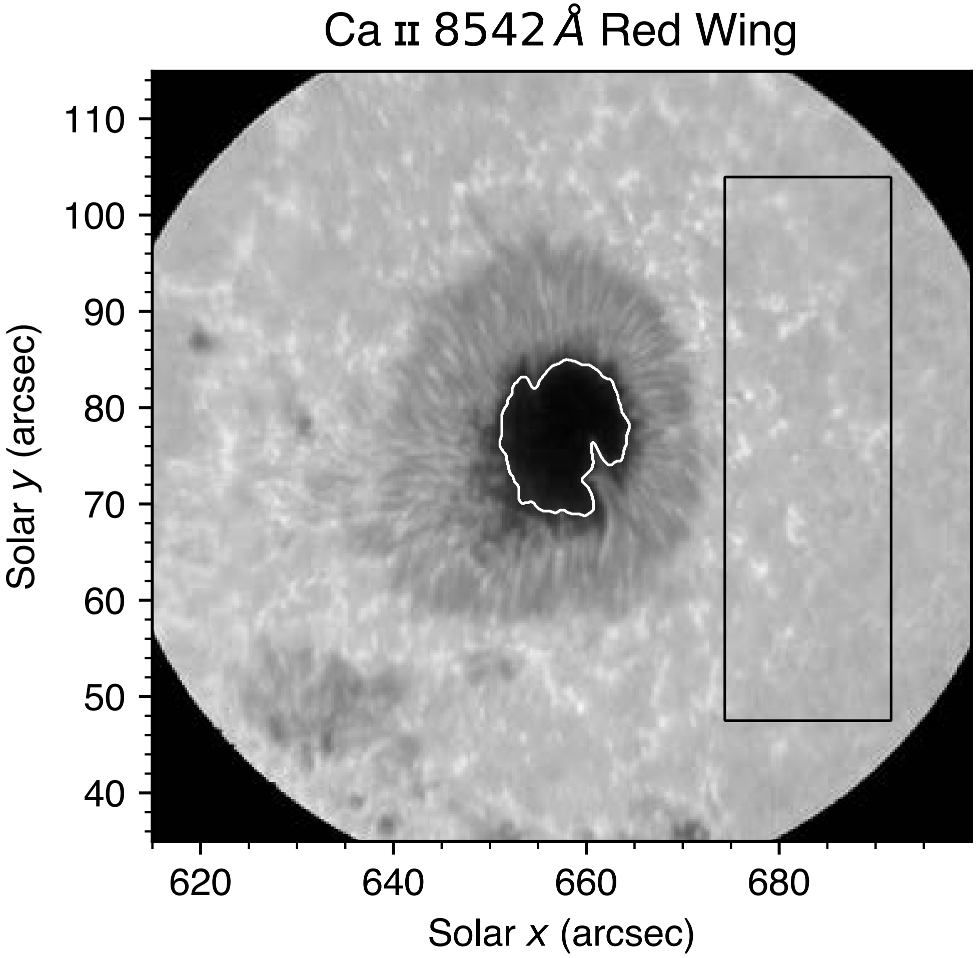


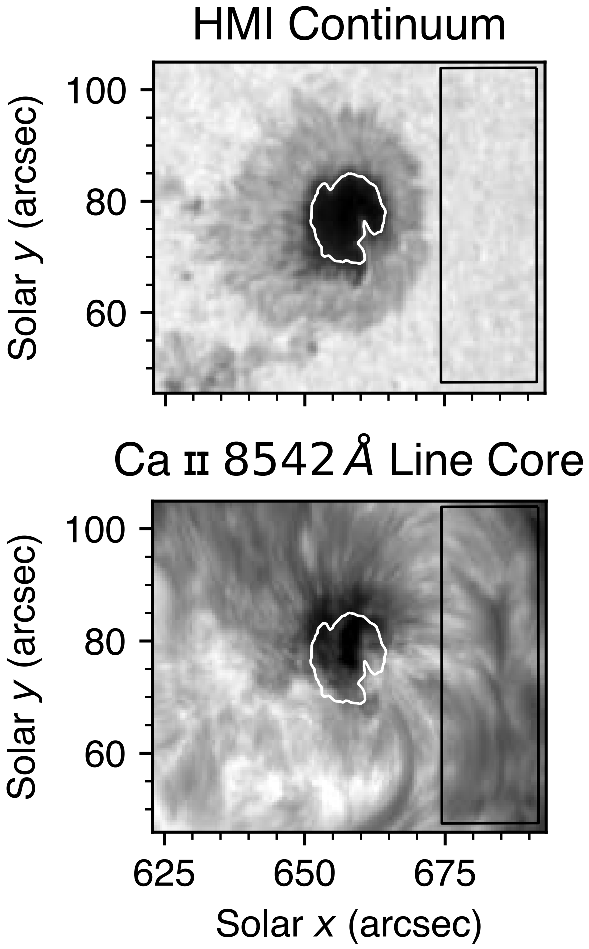


Method & Proof of Concept

Description of the method behind the **mcalf.models.IBIS8542Model** class

IBIS observations





IBIS settings

Spatial resolution

0″.098

per pixel

Temporal resolution

5.8

seconds

Spectral resolution

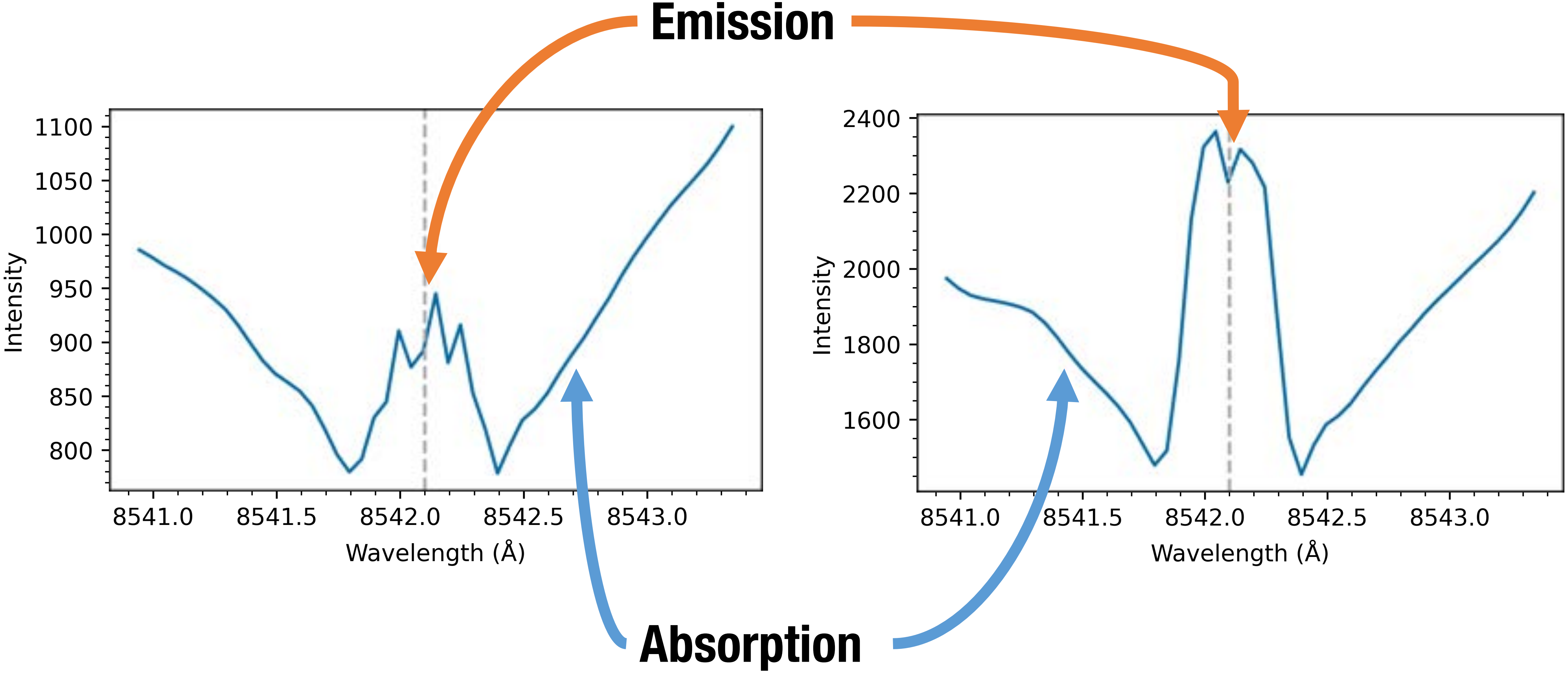
27

points over

2.4 Å

centred at Ca II IR 8542 Å with greater density around the line core

Multiple spectral components



Emission

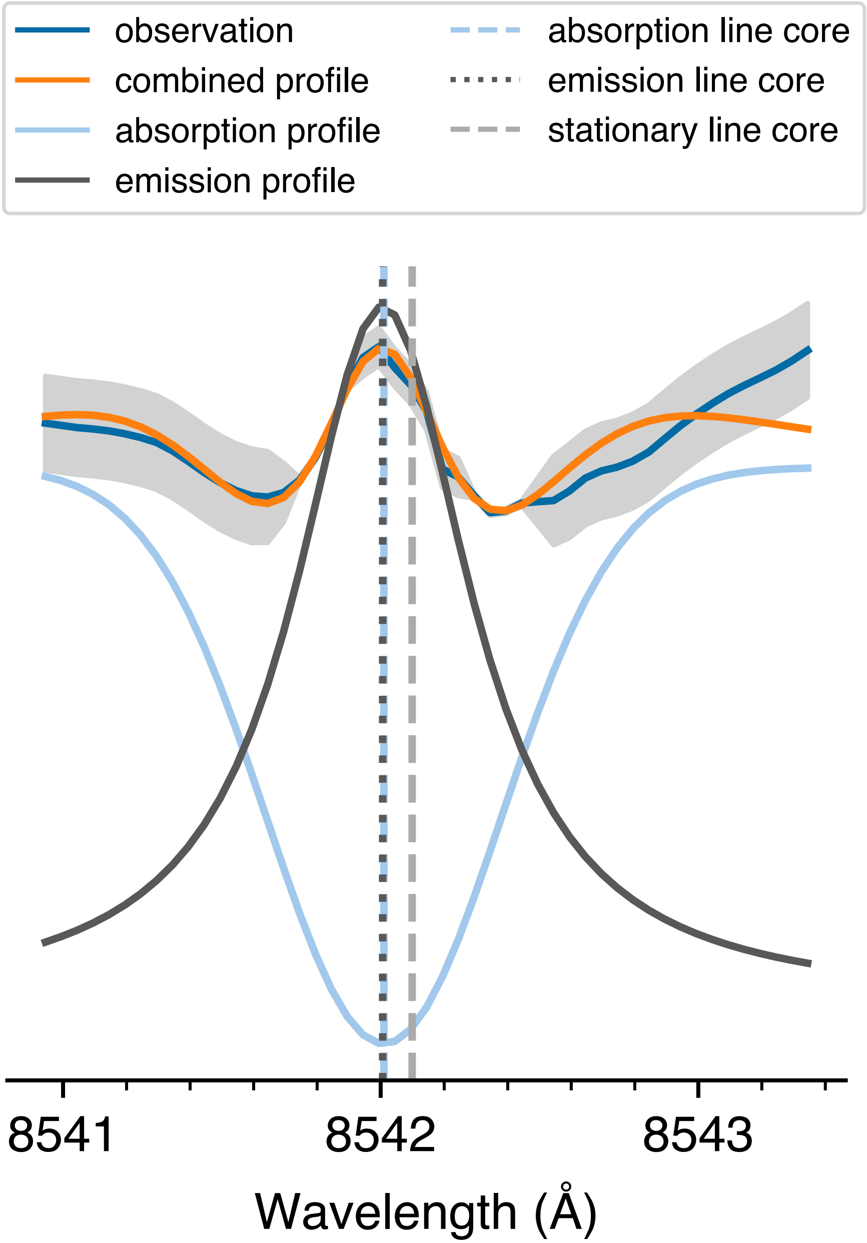
Absorption

Using the Voigt function

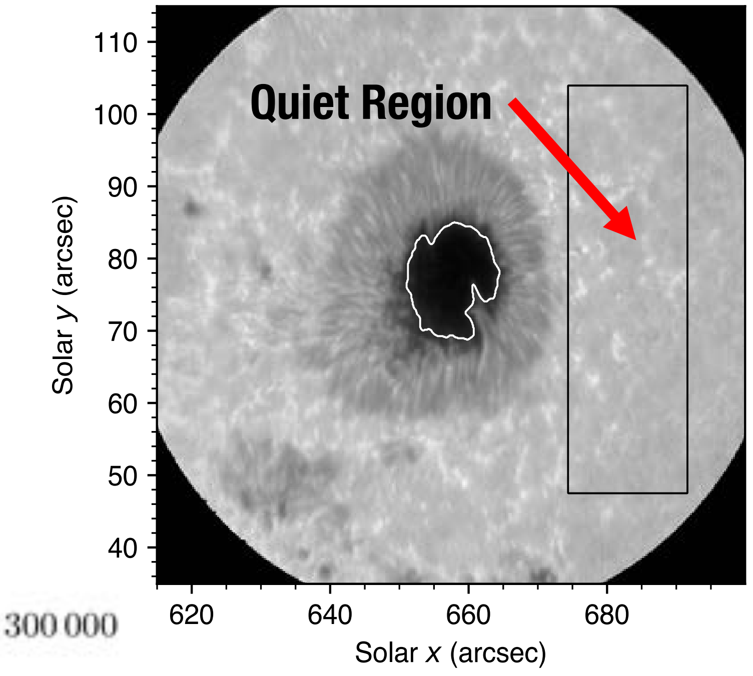
$$V(x; A, \sigma, \gamma) = A \int_{-\infty}^{\infty} G(u; \sigma) L(x - u; \gamma) \mathrm{d}u$$

$$G(x; \sigma) = \exp(-x^2/(2\sigma^2))/(\sigma\sqrt{2\pi})$$

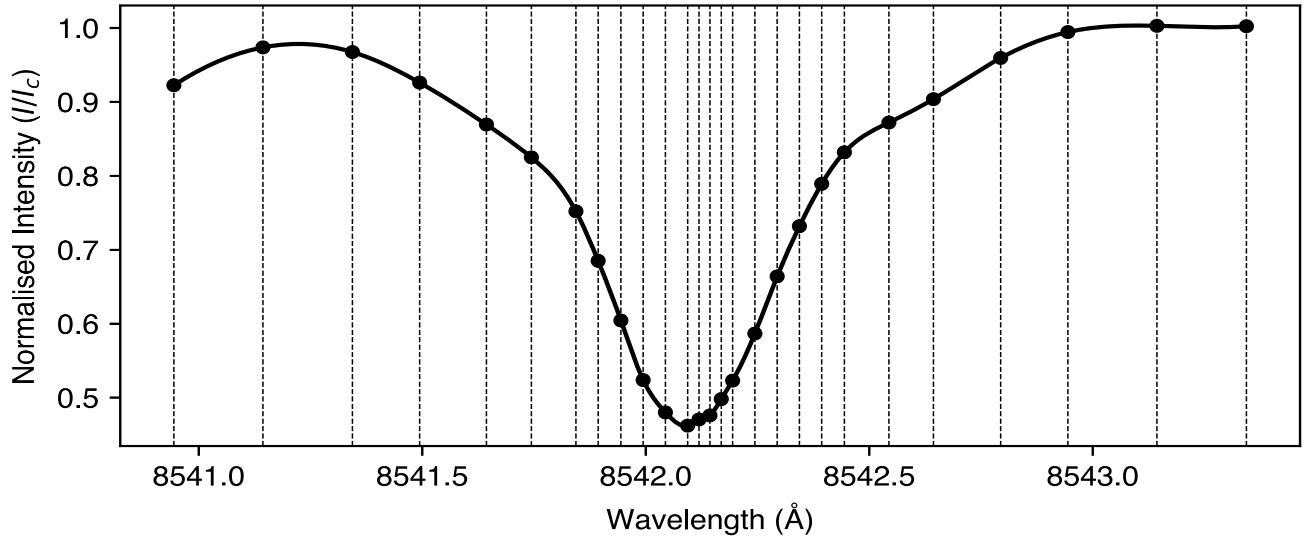
$$L(x; \gamma) = \gamma/(\pi(x^2 + \gamma^2))$$

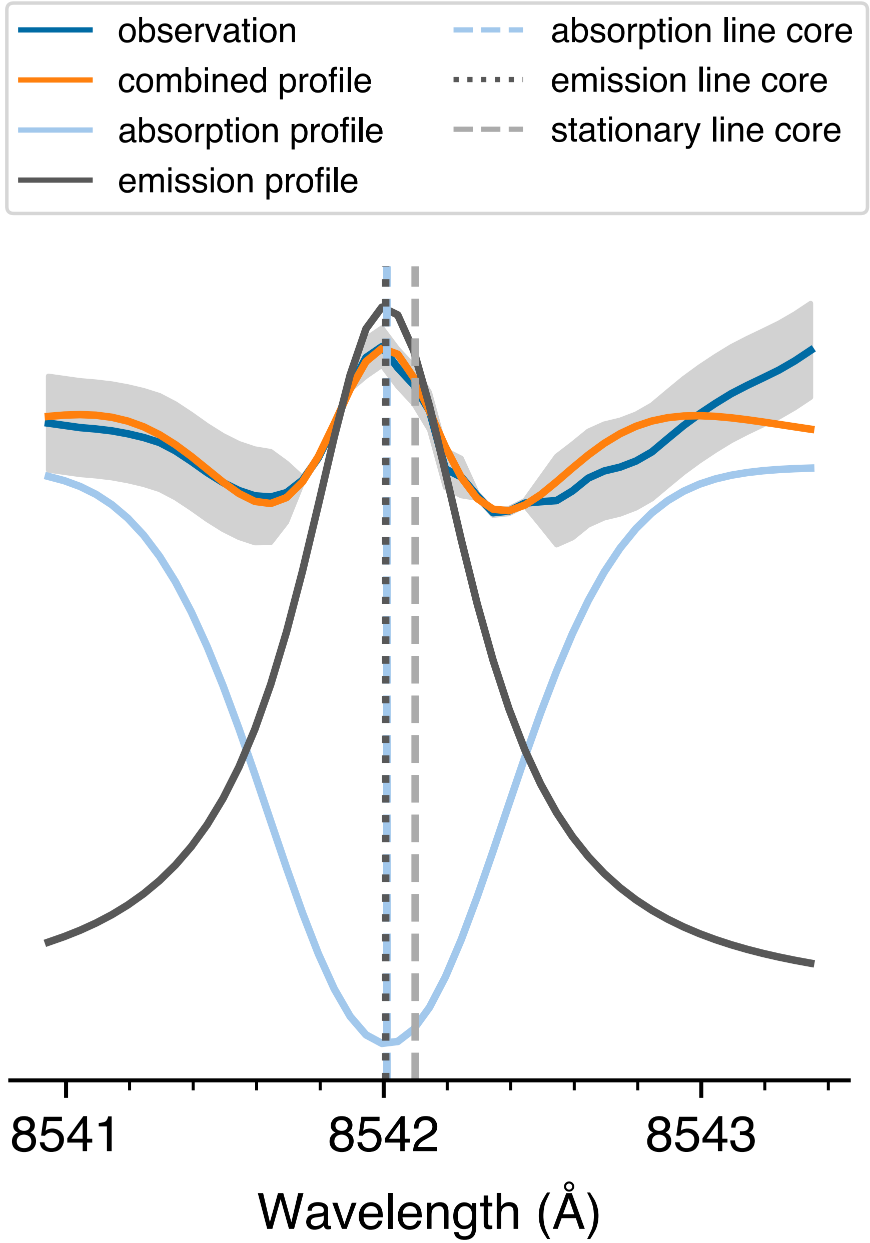


Doppler velocities

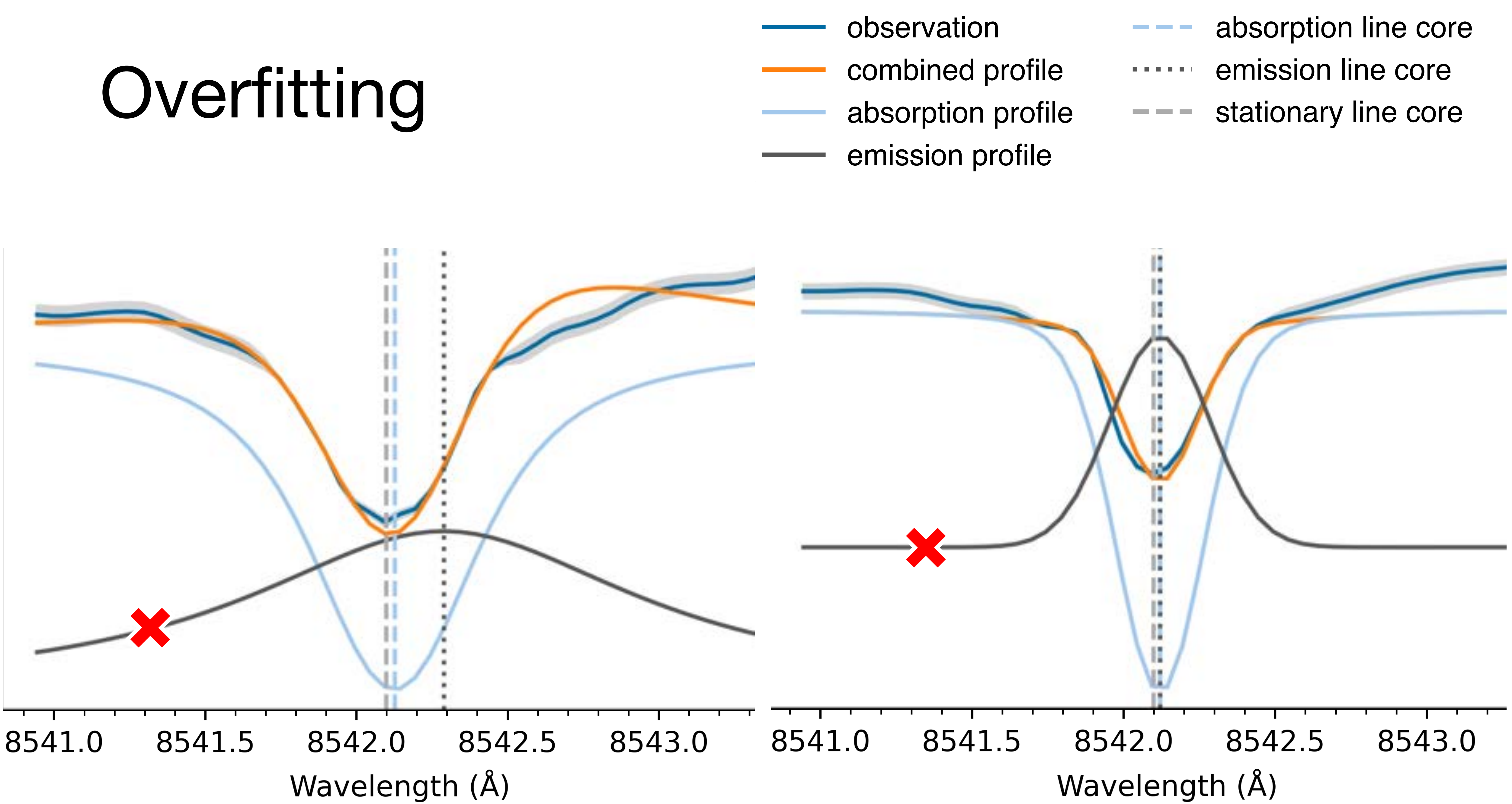


$$v\ (\mathrm{km/s}) = \frac{\lambda_{\text{observed}} - \lambda_{\text{stationary}}}{\lambda_{\text{stationary}}} \times 300\,000$$

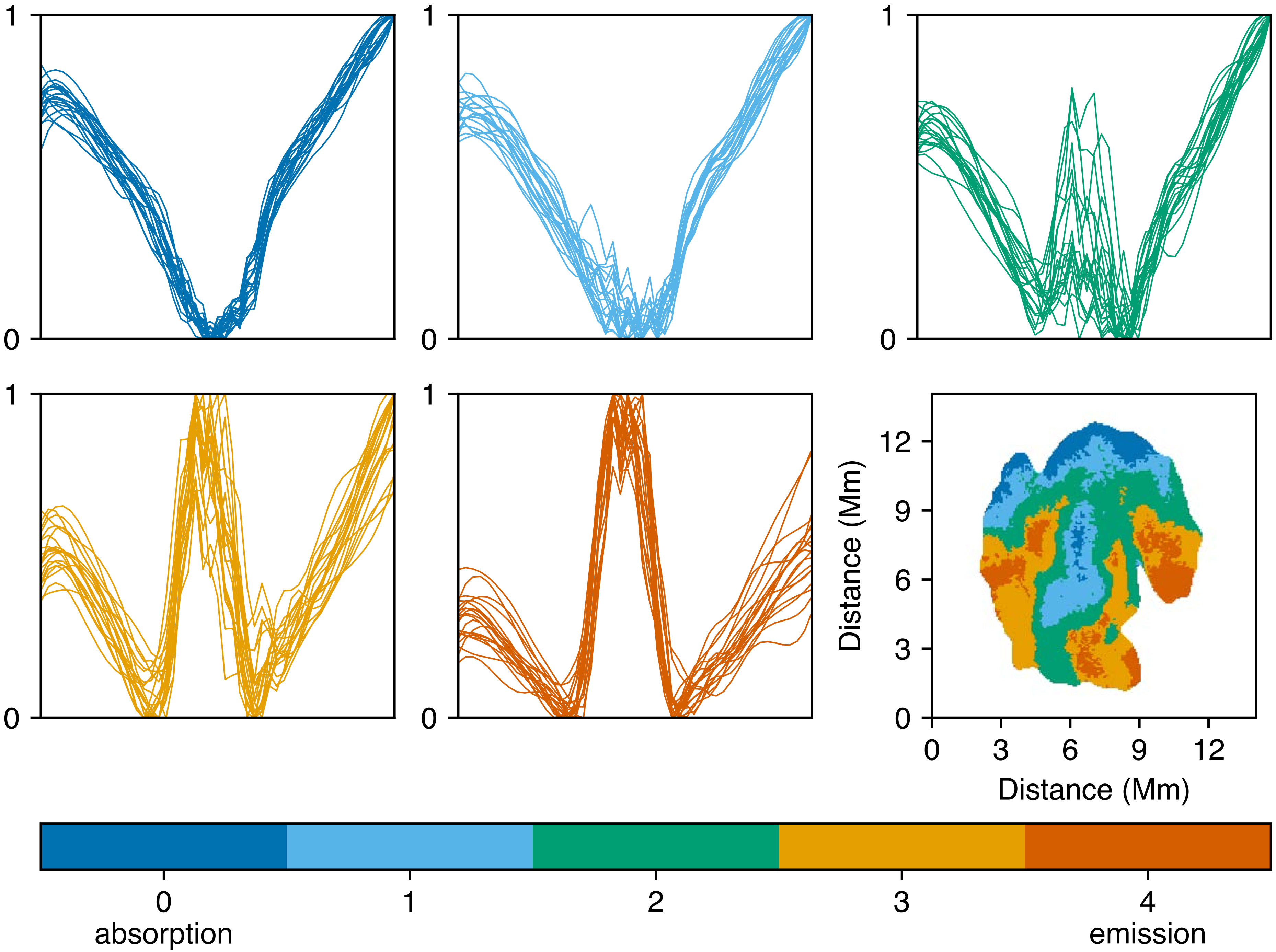




Overfitting

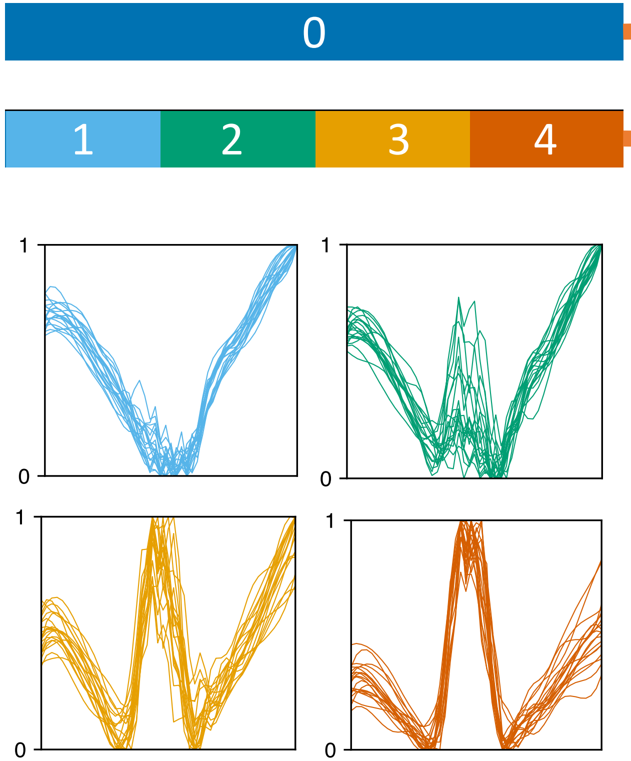


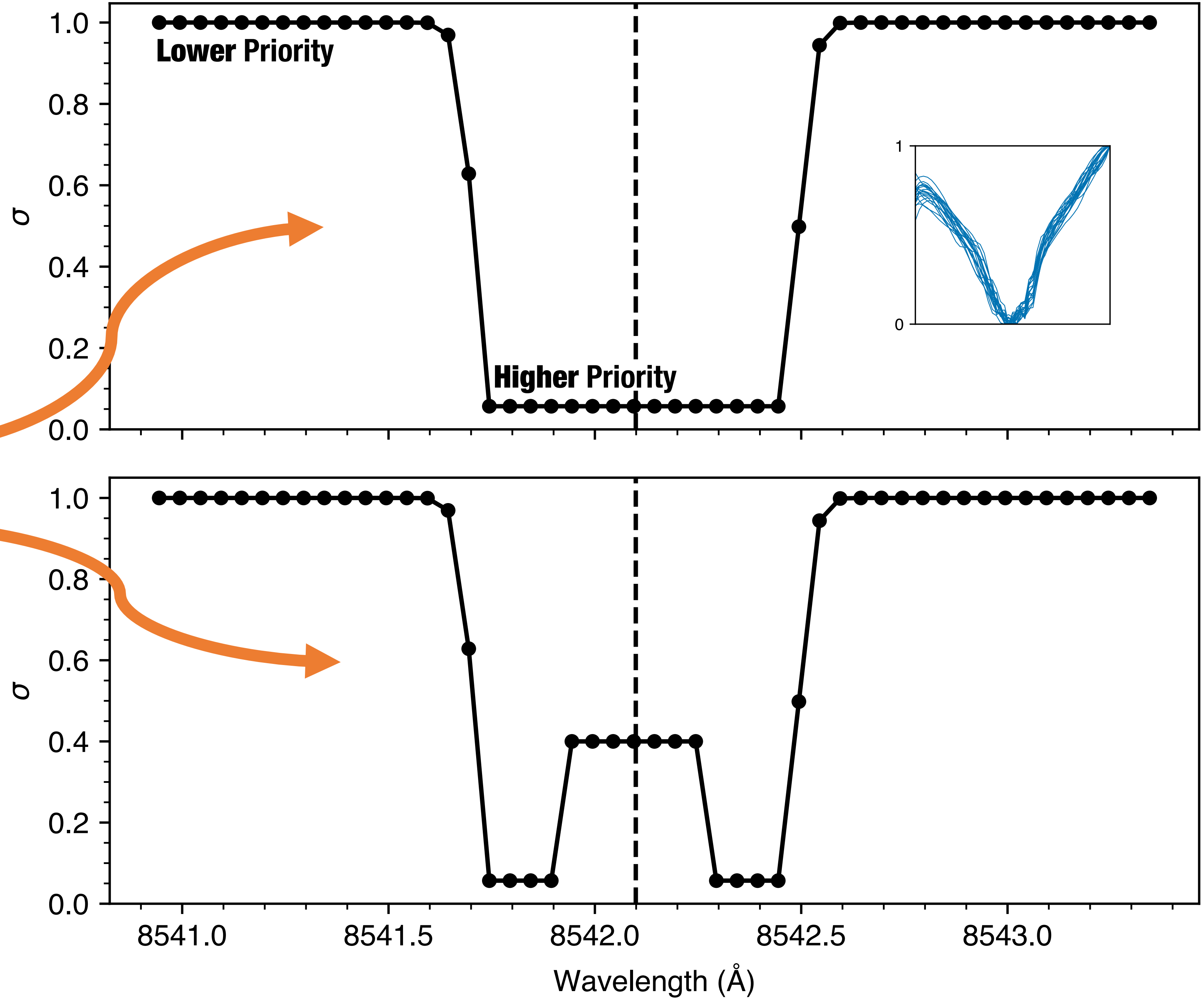
Neural networks



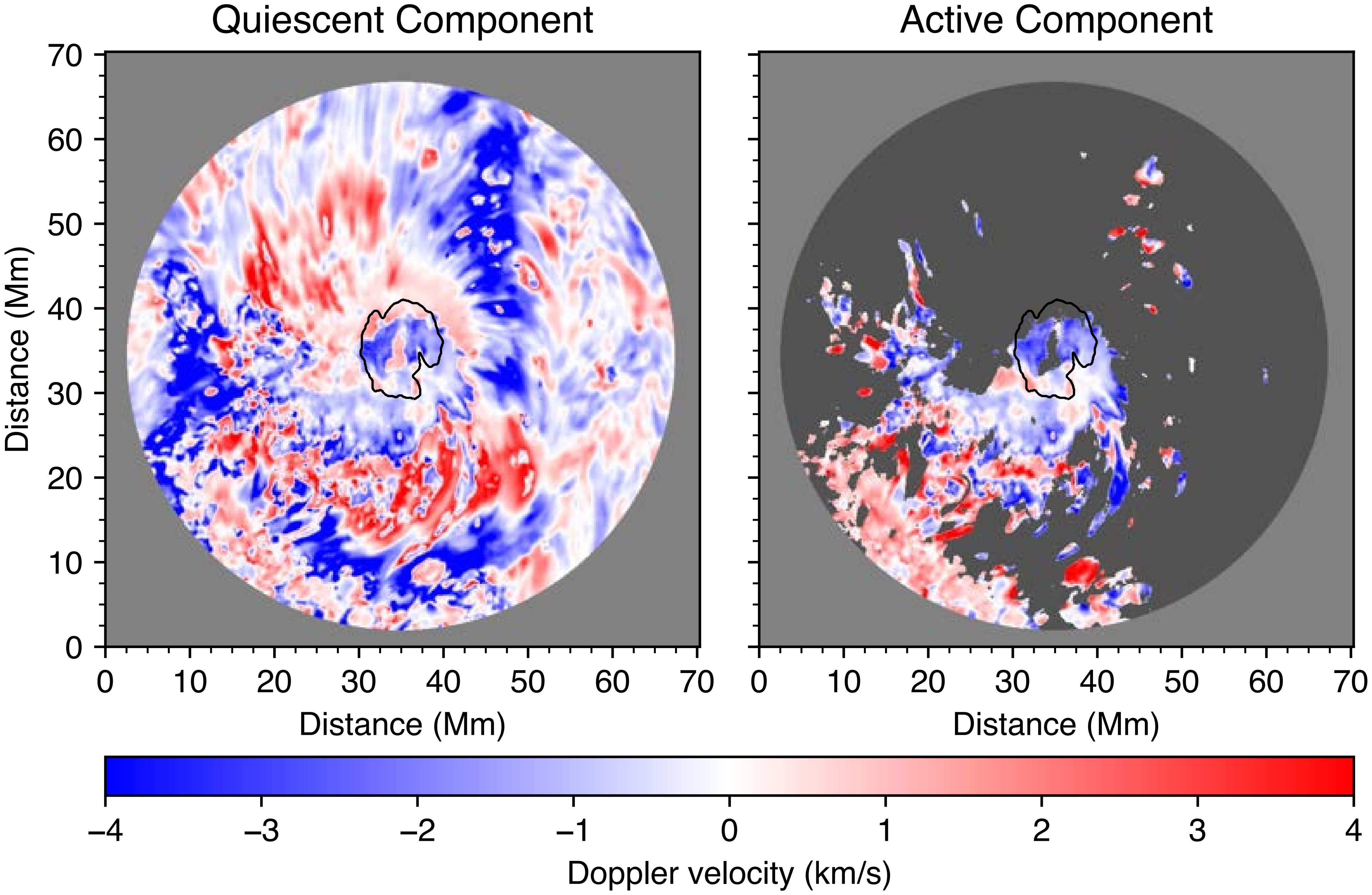
Weighting the fit

Classifications:





Doppler velocity plots



Modified χ^2

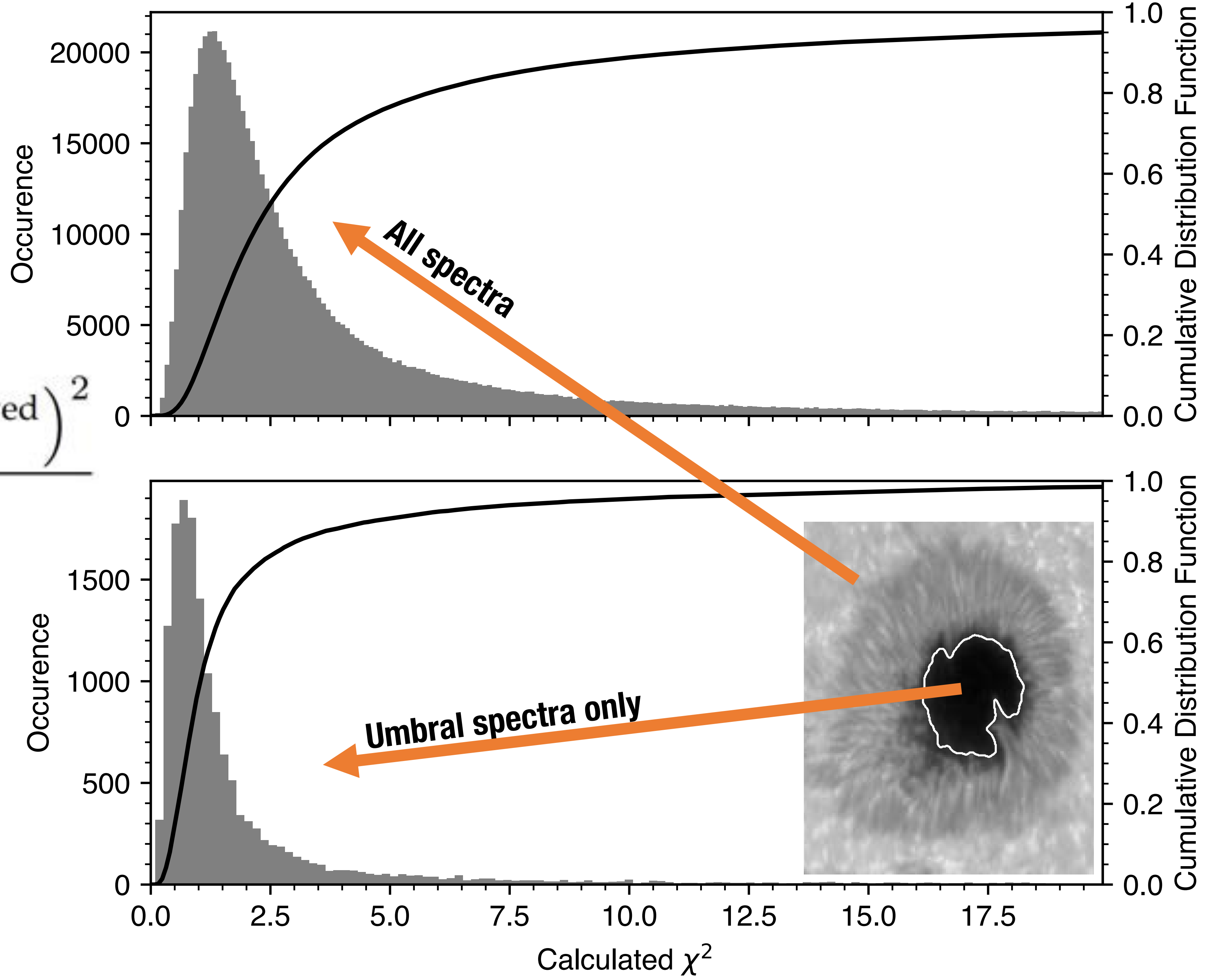
$$\chi^2 = \frac{s}{\nu} \sum_{\lambda \in \lambda_c} \frac{\left(I_\lambda^{\text{fitted}} - I_\lambda^{\text{observed}}\right)^2}{I_\lambda^{\text{observed}}}$$

Scaling factor

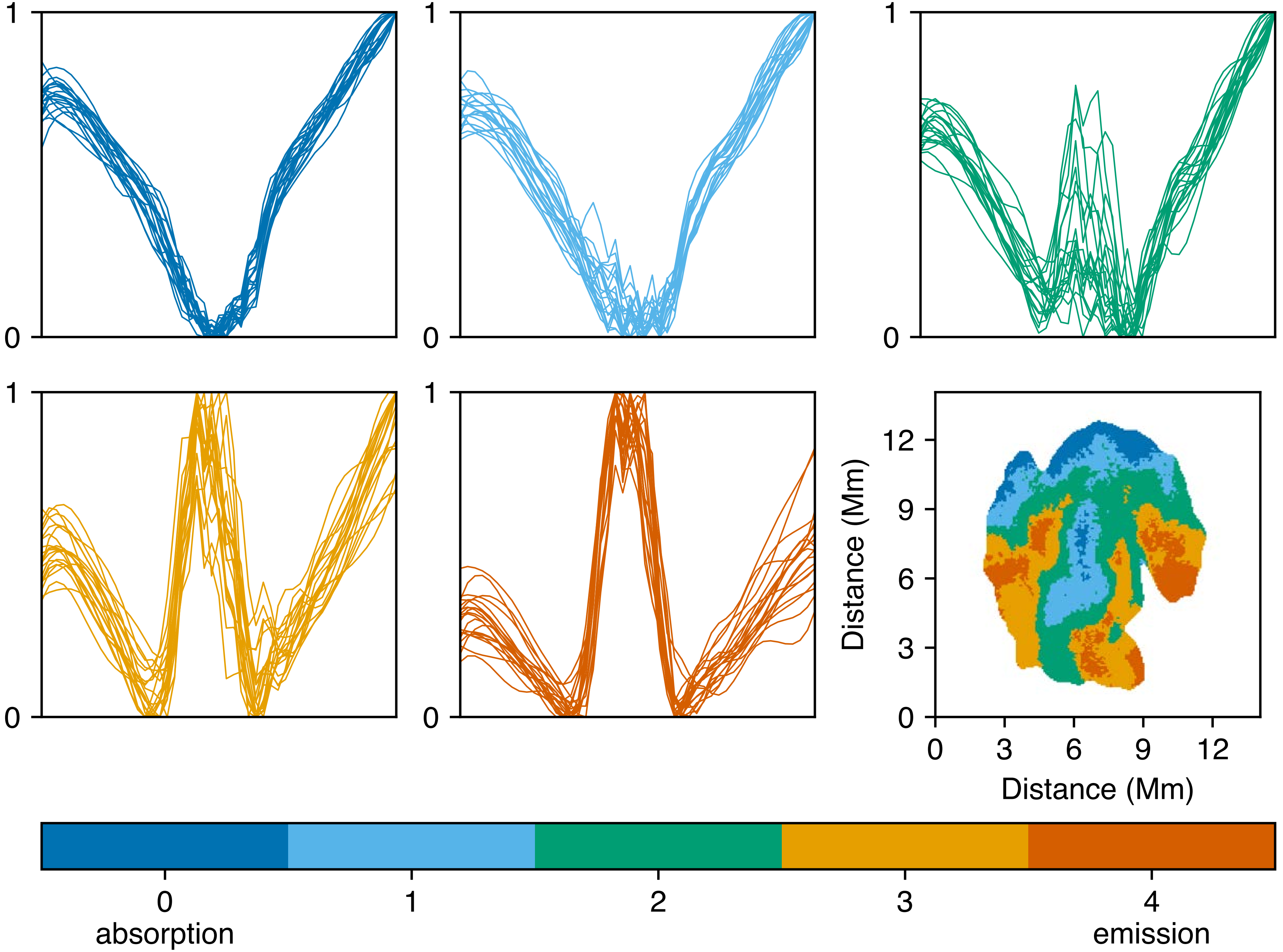
s = 49/25

Estimated degrees of freedom

ν = 4 (single Voigt)
= 8 (double Voigt)



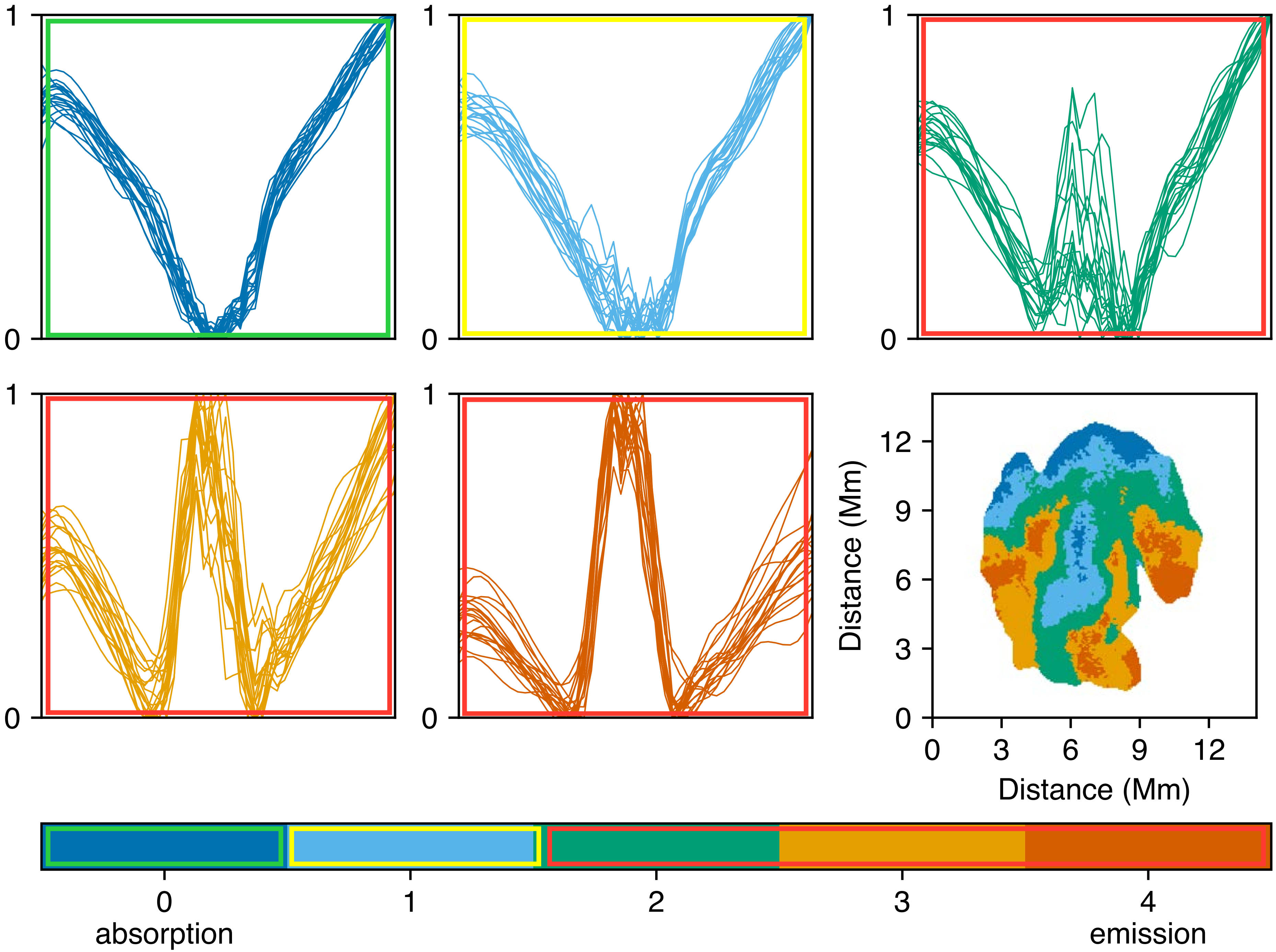
Neural network performance



Neural network performance

Precision
91%

Recall
90%



Precision $\frac{tp}{tp + fp}$ Recall $\frac{tp}{tp + fn}$

true/false positive/negative

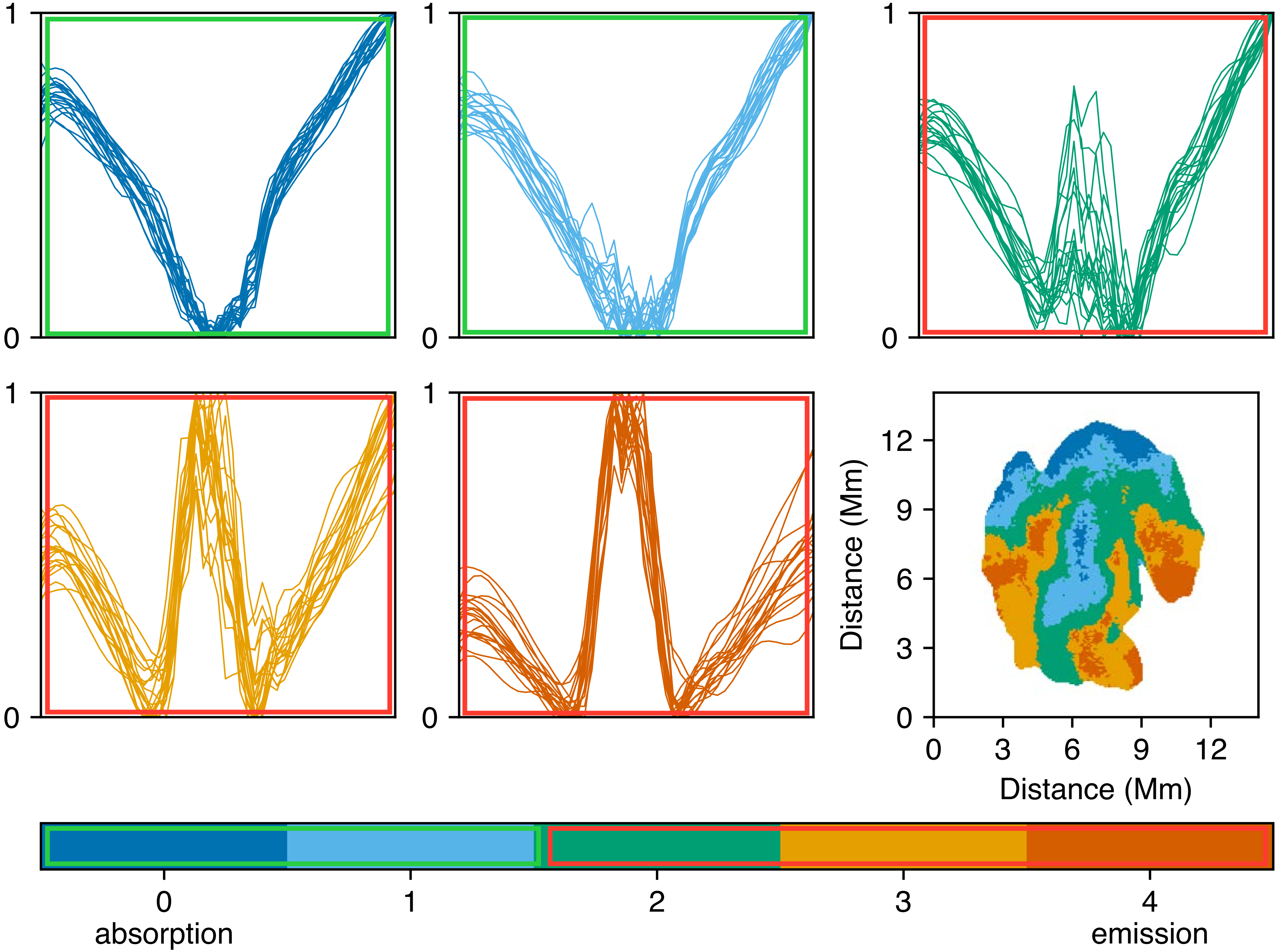
Neural network performance

Precision

96%

Recall

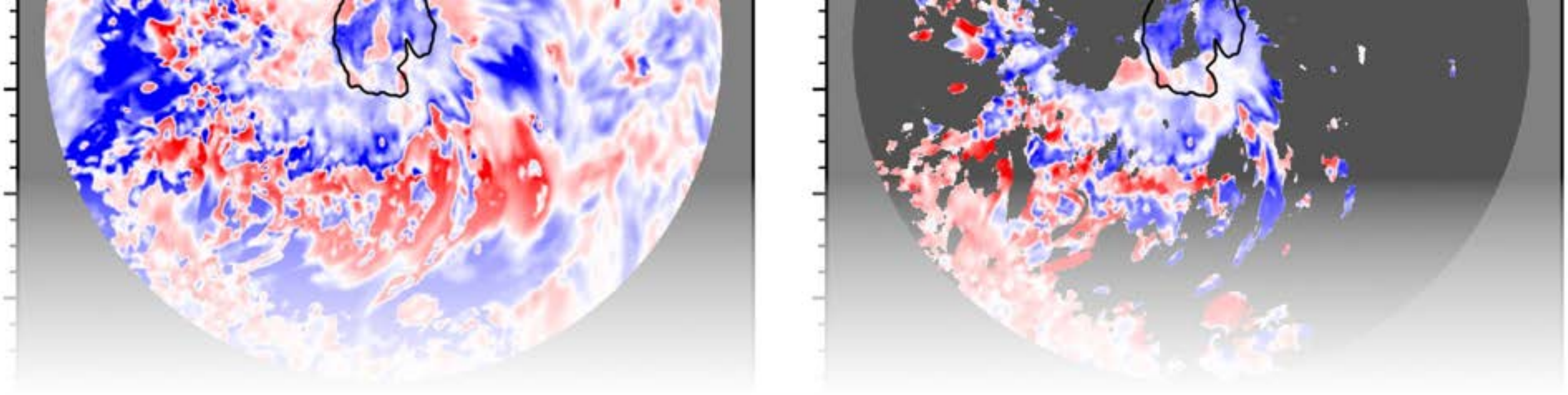
95%



Precision $\frac{tp}{tp+fp}$

Recall $\frac{tp}{tp+fn}$

true/false positive/negative



API Overview

An overview of the methods and functions provided by MCALF

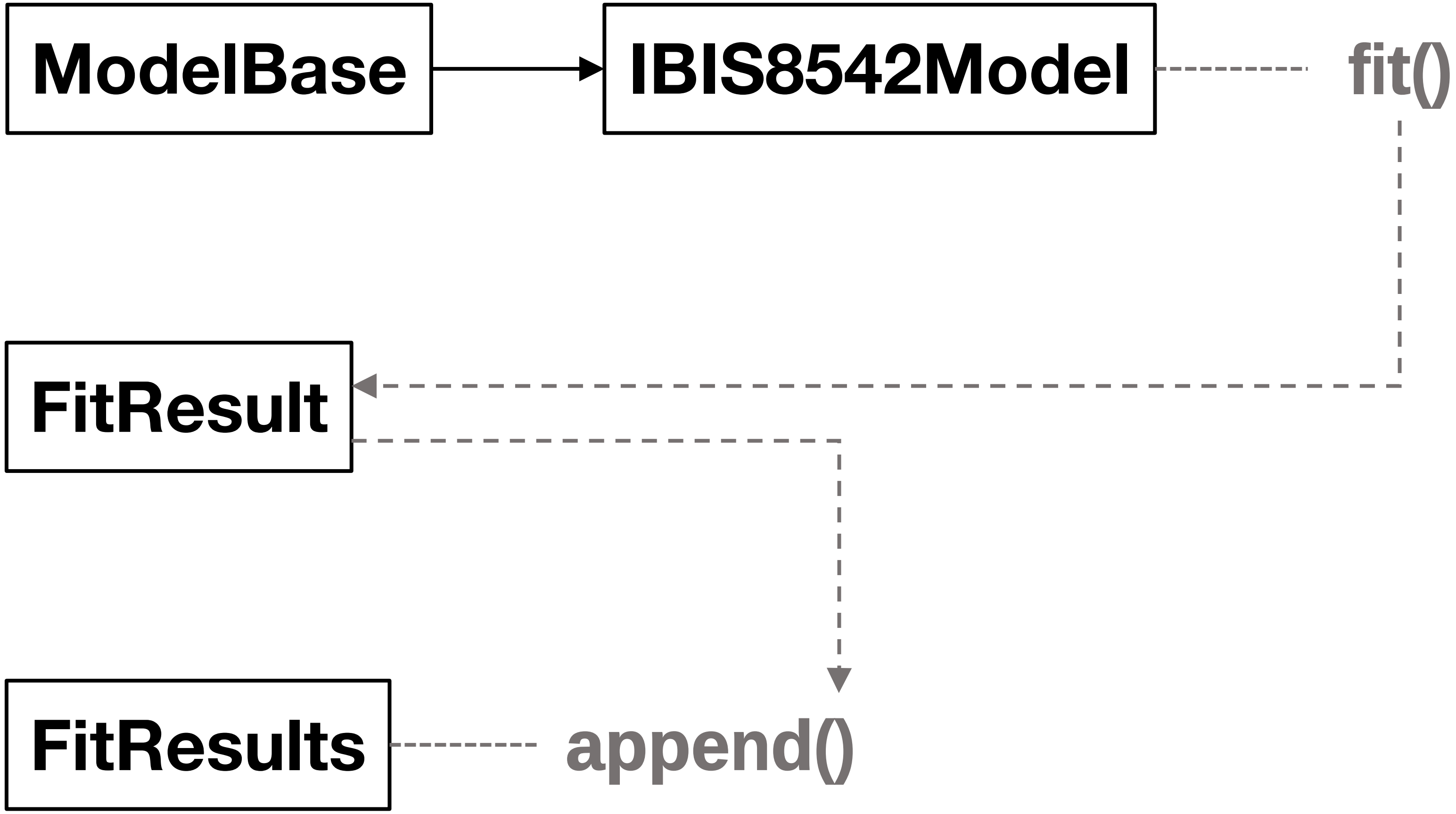
- **mcalf**
    - mcalf.**models** — *classes for fitting spectra & storing results*
    - mcalf.**profiles** — *functions that model spectra*
        - mcalf.profiles.**voigt**
        - mcalf.profiles.**gaussian**
    - mcalf.**visualisation** — *functions to visualise results*
    - mcalf.**utils**
        - mcalf.utils.**spec** — *functions for processing spectra*
        - mcalf.utils.**smooth** — *functions for smoothing n-dimensional arrays*
        - mcalf.utils.**mask** — *functions for masking the input data to limit the region computed*
        - mcalf.utils.**plot** — *functions for helping with plotting*
        - mcalf.utils.**misc** — *miscellaneous utility functions*

mcalf.**models**: Using a model

```
model = mcalf.models.IBIS8542Model(...)

model.load_array(...)

model.train(...)
model.test(...)

result_list = model.fit(...)

results = mcalf.models.FitResults(...)
for fit in result_list:
    results.append(fit)

results.save(...)
```

1. ***initialise** model*
2. ***load** spectra*
3. ***train** classifier*
4. ***fit** spectra*
5. ***merge** results*
6. ***save** results*

mcalf.**models**: Class inheritance

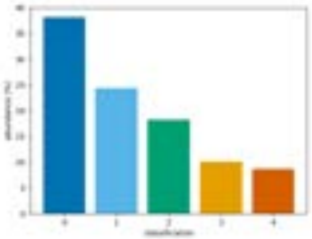


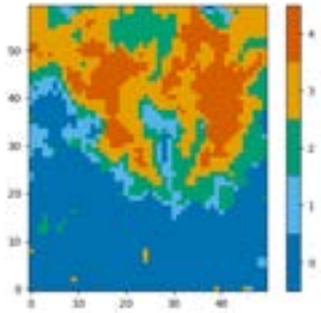
mcalf.**models**: Basic model subclass

```python
from mcalf.models import ModelBase, FitResult

class Model(ModelBase):

    def _fit(self, spectrum, classification=None, spectrum_index=None):
        # Use `classification` to define fitting method
        # Apply fitting method to `spectrum`
        fitted_params = ...
        fit_info = {
            'classification': classification, 'index': spectrum_index,
            'success': ..., 'profile': ..., 'chi2': ...,
        }
        return FitResult(fitted_params, fit_info)

    def plot(self, ...):
        pass
```

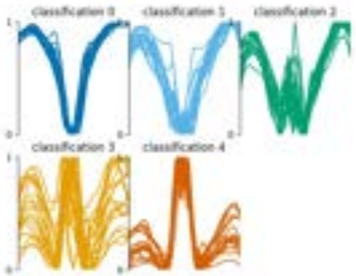
optional

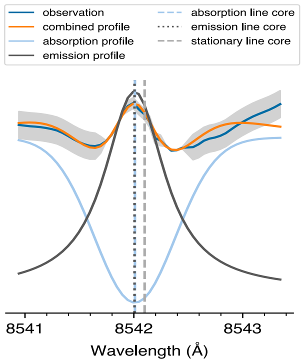
mcalf.**visualisation**



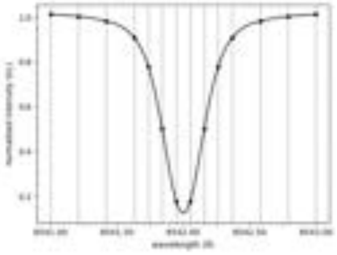


- mcalf.visualisation.**bar***(…) — bar chart of classification abundances*
- mcalf.visualisation.**plot_class_map***(…) — 2D map of classifications*
- mcalf.visualisation.**plot_classifications***(…) — spectra grouped by classification*
- mcalf.visualisation.**init_class_data***(…) — precompute classification plotting data*

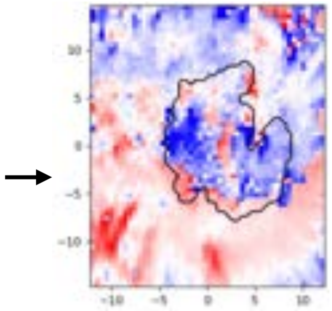




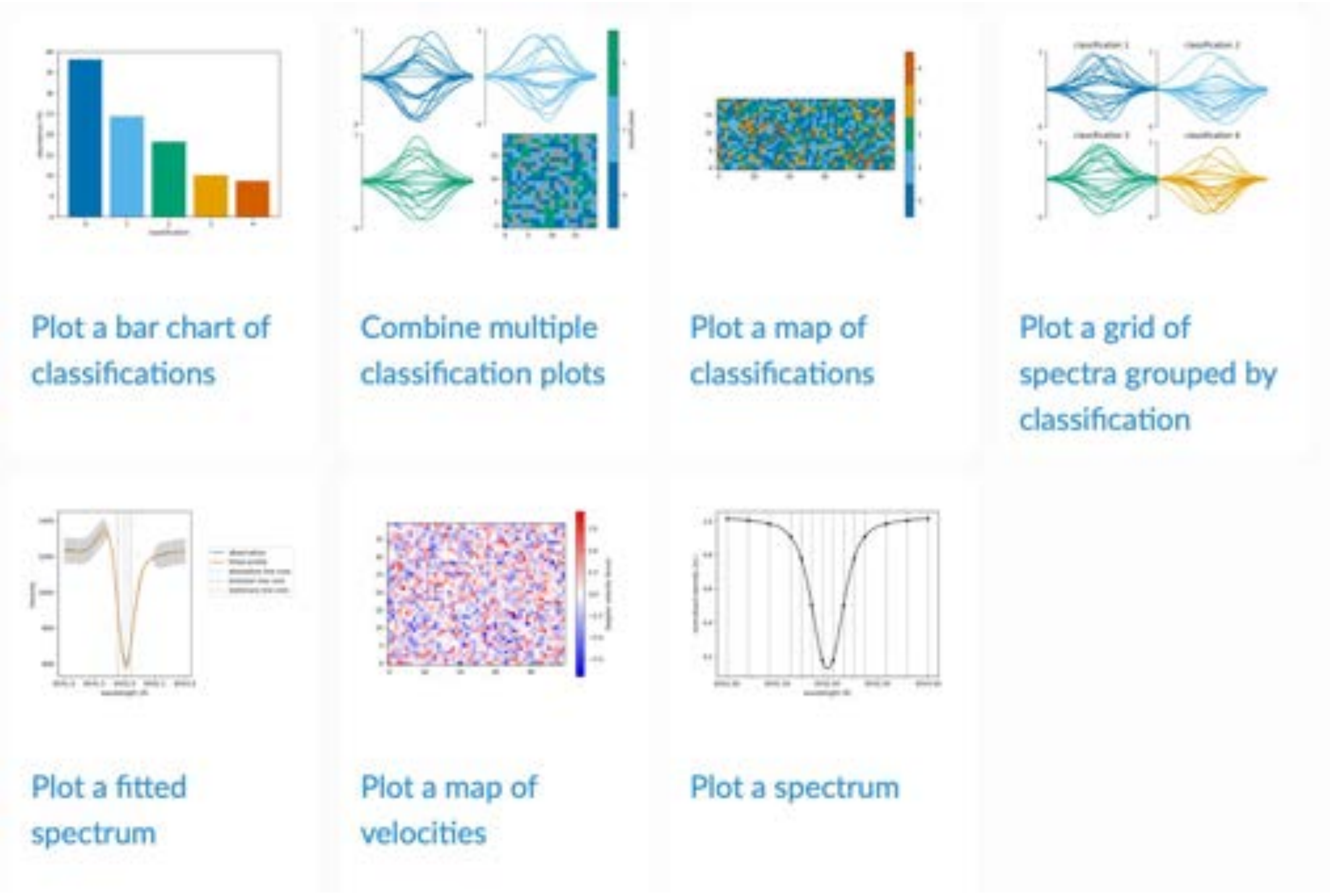
- mcalf.visualisation.**plot_ibis8542***(…) — IBIS8542Model.plot(…)*
- mcalf.visualisation.**plot_spectrum***(…) — spectrum with wavelength grid*



- mcalf.visualisation.**plot_map***(…) — 2D velocity map* →



mcalf.**visualisation**: Example Gallery



Plot a bar chart of classifications

Combine multiple classification plots

Plot a map of classifications

Plot a grid of spectra grouped by classification

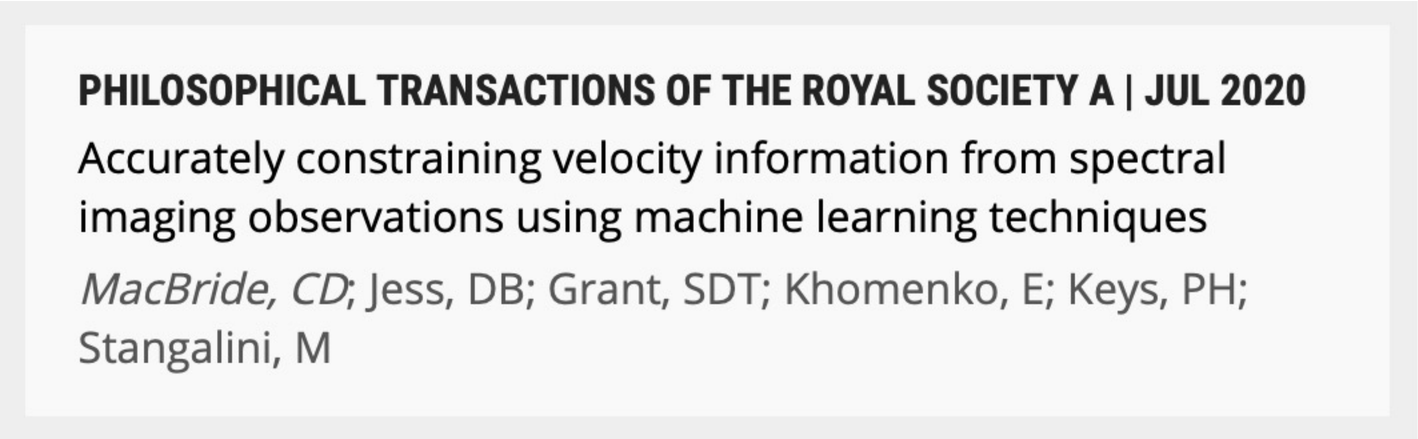
Plot a fitted spectrum

Plot a map of velocities

Plot a spectrum

https://mcalf.macbride.me/en/stable/gallery/

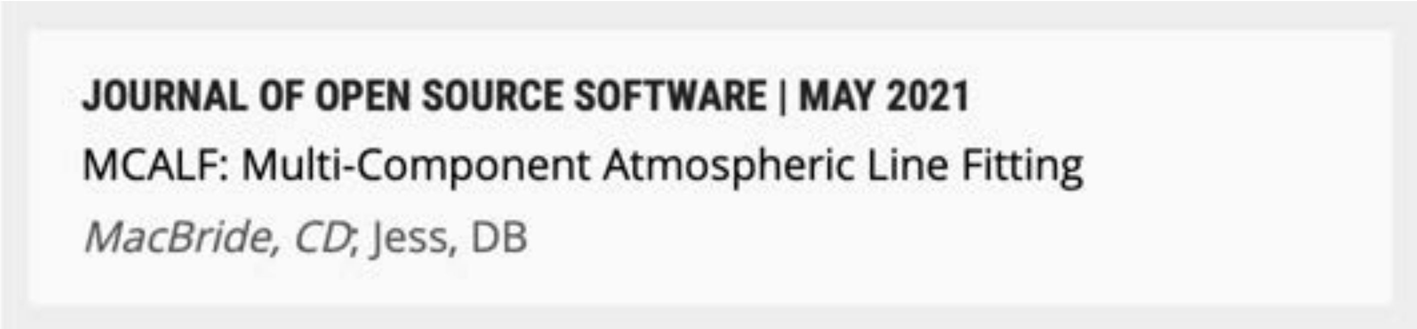
MCALF Publications



PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A | JUL 2020

Accurately constraining velocity information from spectral imaging observations using machine learning techniques

MacBride, CD; Jess, DB; Grant, SDT; Khomenko, E; Keys, PH; Stangalini, M



JOURNAL OF OPEN SOURCE SOFTWARE | MAY 2021

MCALF: Multi-Component Atmospheric Line Fitting

MacBride, CD; Jess, DB

PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A | JUL 2020

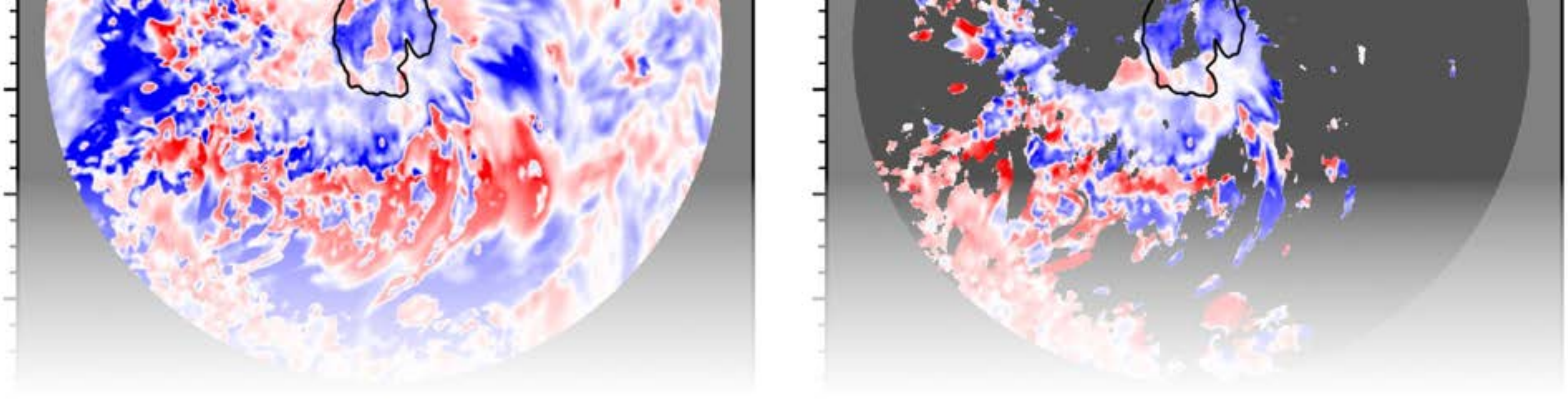
Accurately constraining velocity information from spectral imaging observations using machine learning techniques

From the *Example Gallery*:

Working with IBIS data

This example shows how to initialise the mcalf.models.IBIS8542Model class with real IBIS data, and train a neural network classifier. We then proceed to fit the array of spectra and visualise the results.

https://mcalf.macbride.me/en/stable/gallery/models/plot_ibis8542data.html

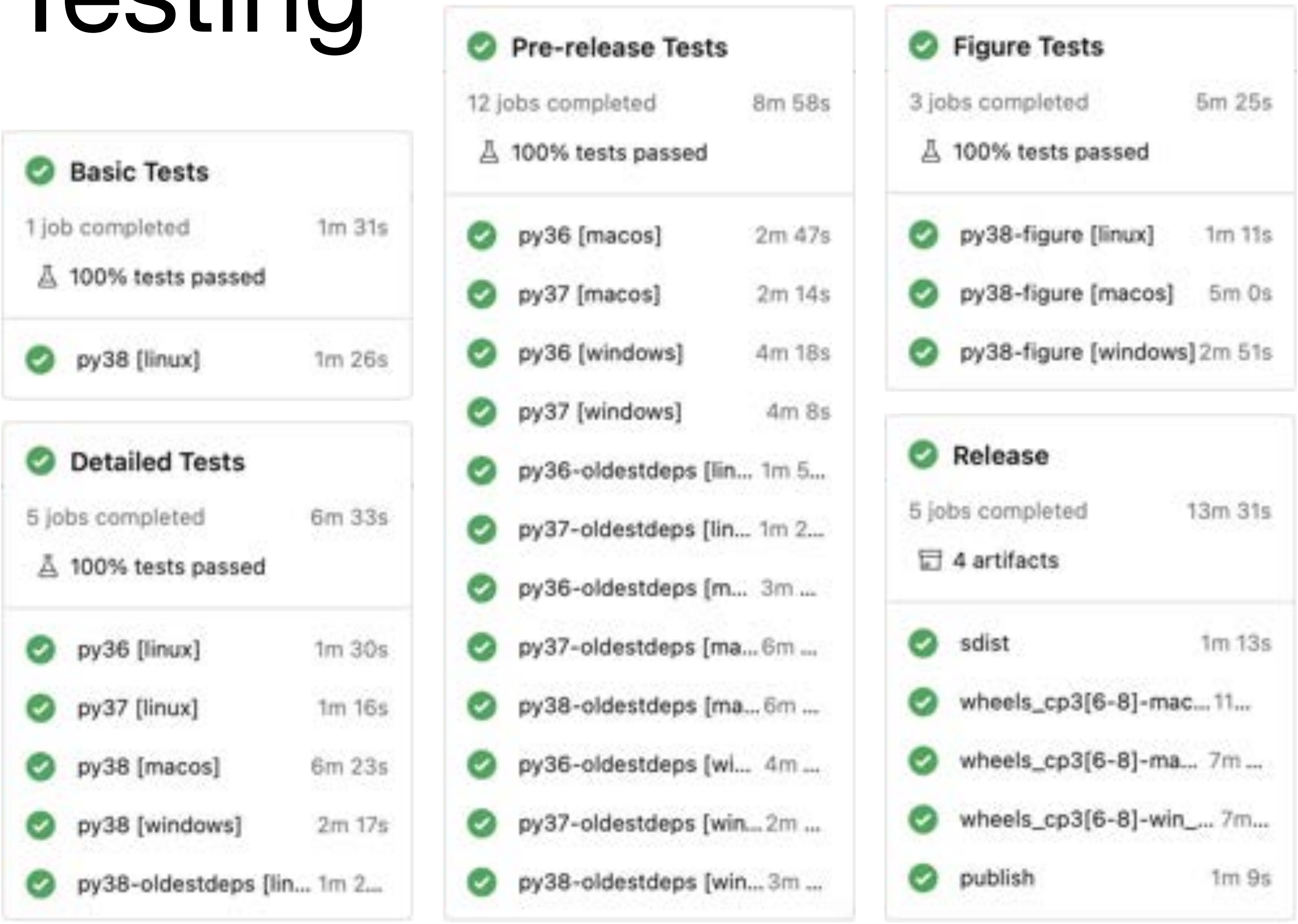


Infrastructure

An overview of MCALF’s DevOps methods and services

Testing

OpenAstronomy / **azure-pipelines-templates**



Basic Tests

1 job completed 1m 31s

100% tests passed

py38 [linux]	1m 26s

Detailed Tests

5 jobs completed 6m 33s

100% tests passed

py36 [linux]	1m 30s
py37 [linux]	1m 16s
py38 [macos]	6m 23s
py38 [windows]	2m 17s
py38-oldestdeps [lin...	1m 2...

Pre-release Tests

12 jobs completed 8m 58s

100% tests passed

py36 [macos]	2m 47s
py37 [macos]	2m 14s
py36 [windows]	4m 18s
py37 [windows]	4m 8s
py36-oldestdeps [lin...	1m 5...
py37-oldestdeps [lin...	1m 2...
py36-oldestdeps [m...	3m ...
py37-oldestdeps [ma...	6m ...
py38-oldestdeps [ma...	6m ...
py36-oldestdeps [wi...	4m ...
py37-oldestdeps [win...	2m ...
py38-oldestdeps [win...	3m ...

Figure Tests

3 jobs completed 5m 25s

100% tests passed

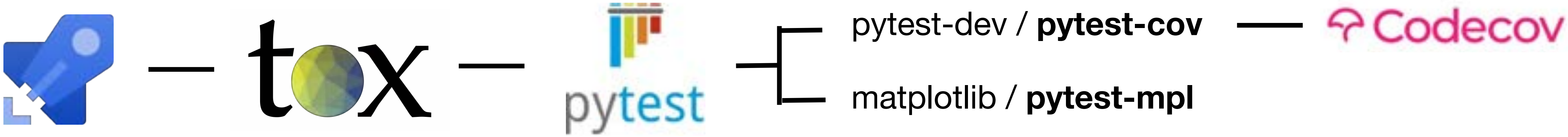
py38-figure [linux]	1m 11s
py38-figure [macos]	5m 0s
py38-figure [windows]	2m 51s

Release

5 jobs completed 13m 31s

4 artifacts

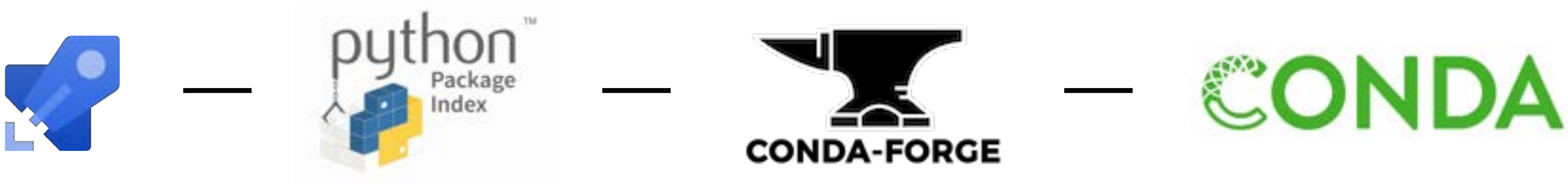
sdist	1m 13s
wheels_cp3[6-8]-mac...	11...
wheels_cp3[6-8]-ma...	7m ...
wheels_cp3[6-8]-win_...	7m...
publish	1m 9s



pytest-dev / **pytest-cov**

matplotlib / **pytest-mpl**

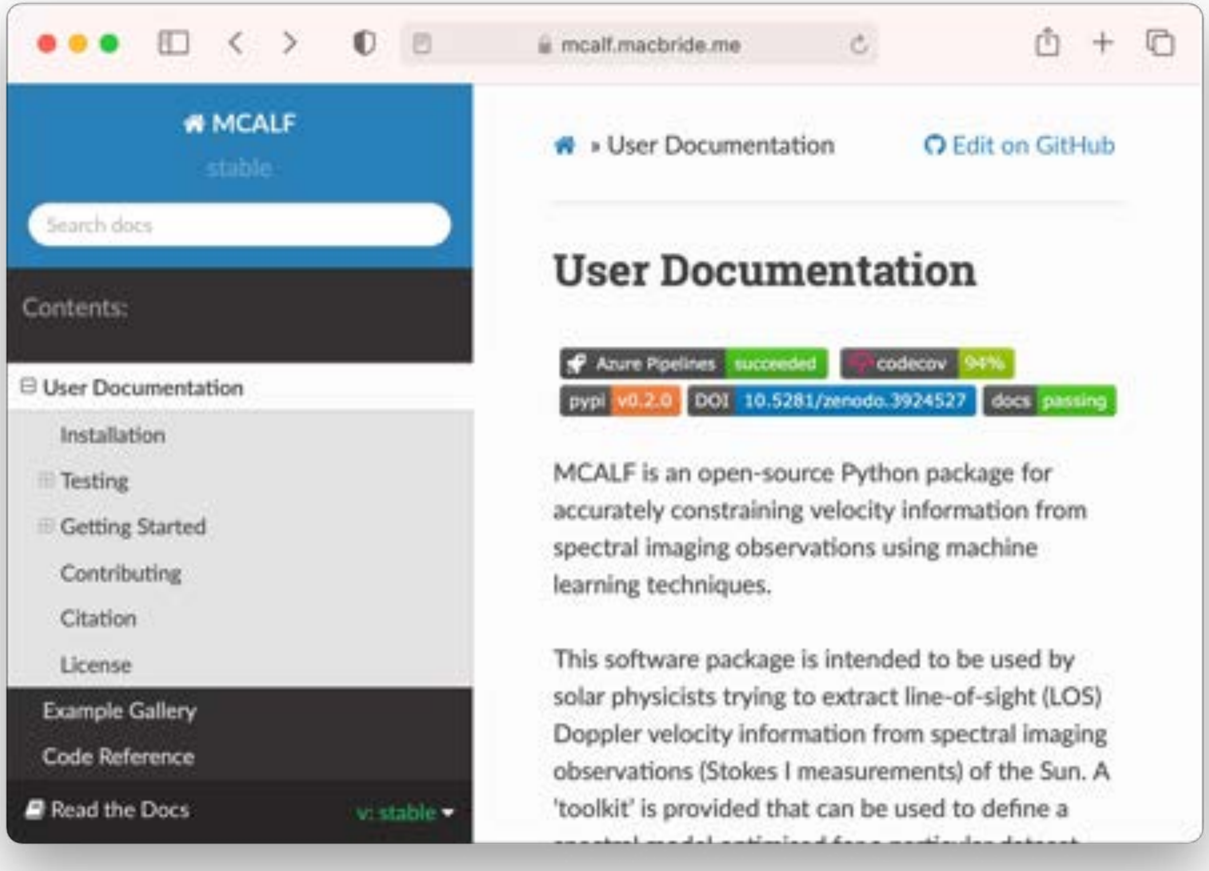
Publishing



OpenAstronomy / **azure-pipelines-templates**

zenodo

Documentation

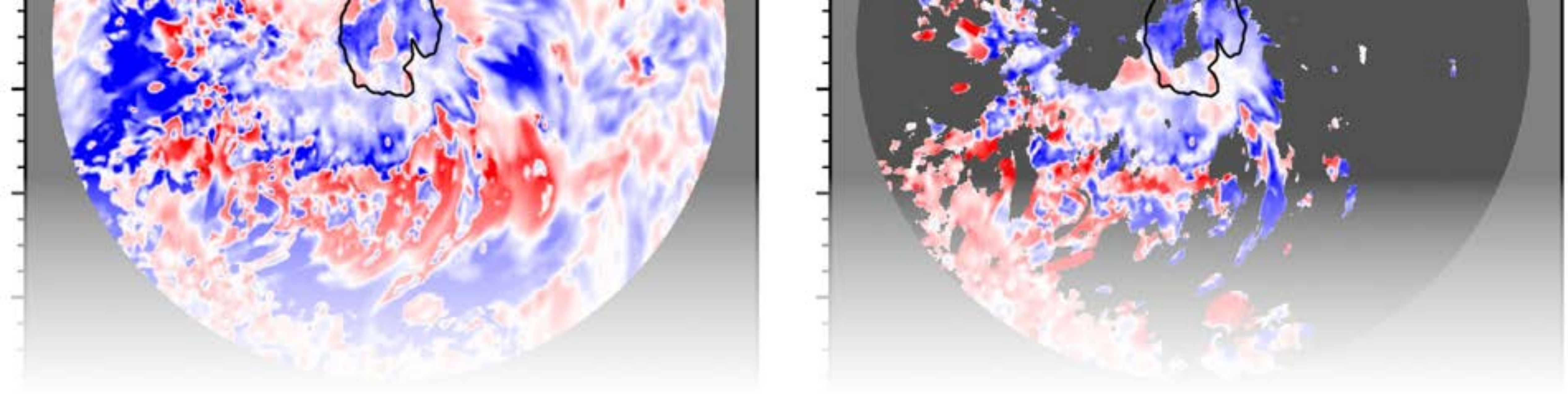


User Documentation

MCALF is an open-source Python package for accurately constraining velocity information from spectral imaging observations using machine learning techniques.

This software package is intended to be used by solar physicists trying to extract line-of-sight (LOS) Doppler velocity information from spectral imaging observations (Stokes I measurements) of the Sun. A 'toolkit' is provided that can be used to define a

- Read the Docs
- Sphinx
  - Sphinx-Gallery
  - astropy / **sphinx-automodapi**



Future

How MCALF can be improved and developed

MCALF: Multi-Component Atmospheric Line Fitting

MCALF is an open-source Python package for accurately constraining velocity information from spectral imaging observations using machine learning techniques.

Conor MacBride
PhD Student, Queen's University Belfast

David Jess
Reader, Queen's University Belfast

GitHub
github.com/ConorMacBride/mcalf

Documentation
mcalf.macbride.me

Install

```
pip install mcalf
conda install mcalf
```

✉ cmacbride01@qub.ac.uk | conor@macbride.me

macbride.me

ConorMacBride