

NiLearn

First and second level analysis of a BIDS
dataset with NiLearn in Python

Sophia Haake, Henrike Zimmermann,
08.05.2025

Overview

- NiLearn and BIDS dataset format
- Processing pipeline
- SPM vs. Nilearn
- Demo code
- Exercises

What is NiLearn?



Python library designed for analyzing
neuroimaging data



Features:

Statistical tools
Visualization tools
Machine learning tools

Recap: Brain Imaging Data Structure



Describes a way of organizing neuroimaging and behavioral data



Standardize the process → facilitates reusing and reproducibility



Compatibility with preprocessing and analysis tools

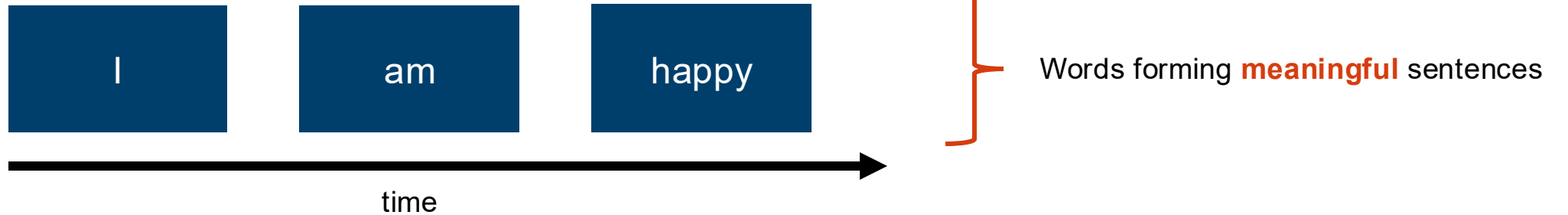


Supports different modalities (EEG, fMRI ...)

‘fetch_language_localizer_demo_dataset’ function

Experiment: „language localizer“ → rapid serial visual presentation of stimuli

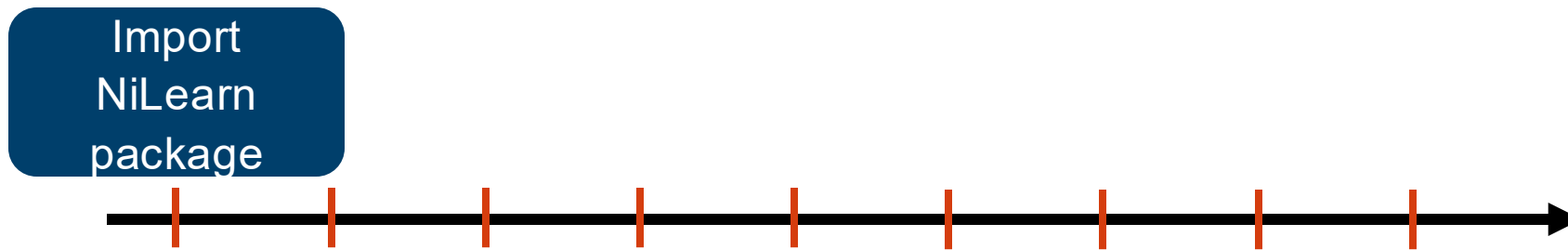
- Condition1:



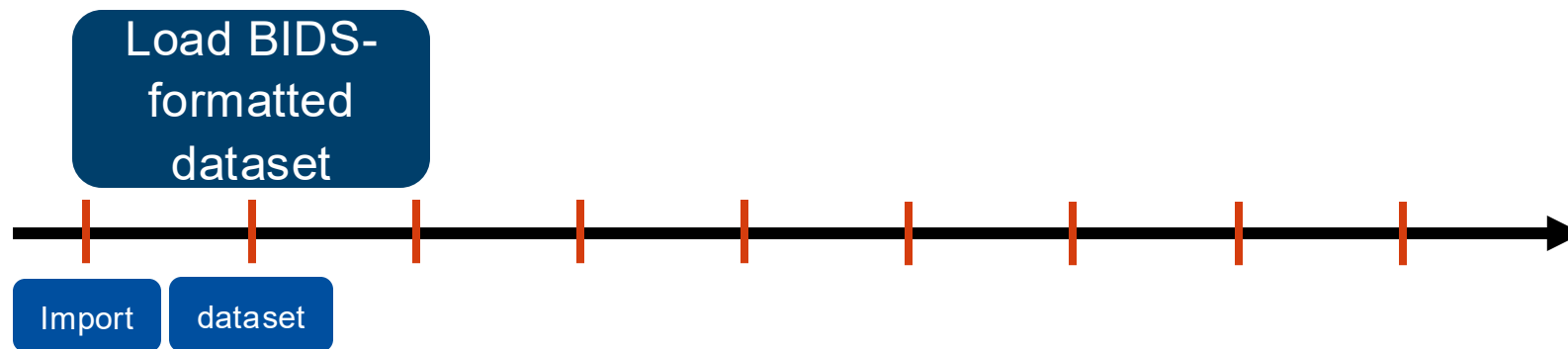
- Condition 2:



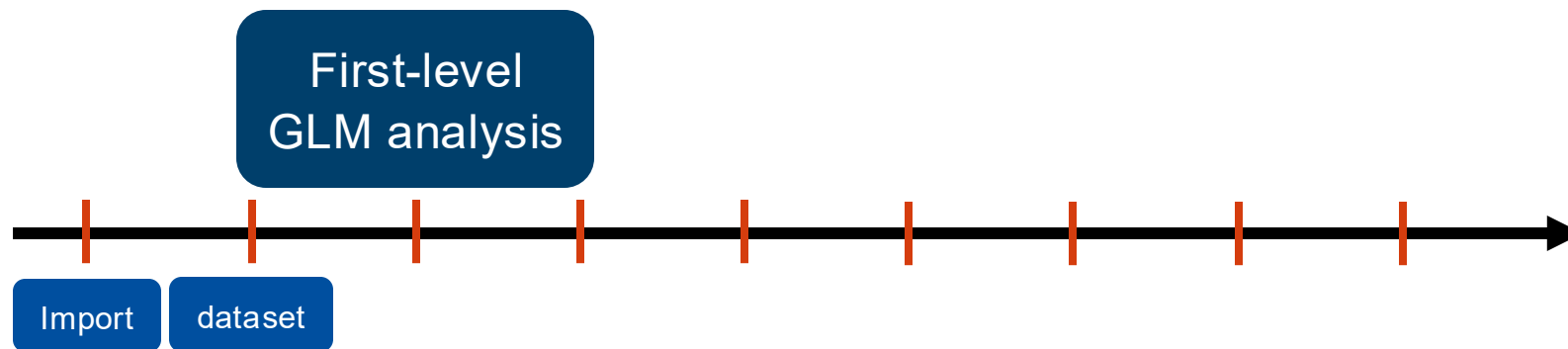
Processing pipeline



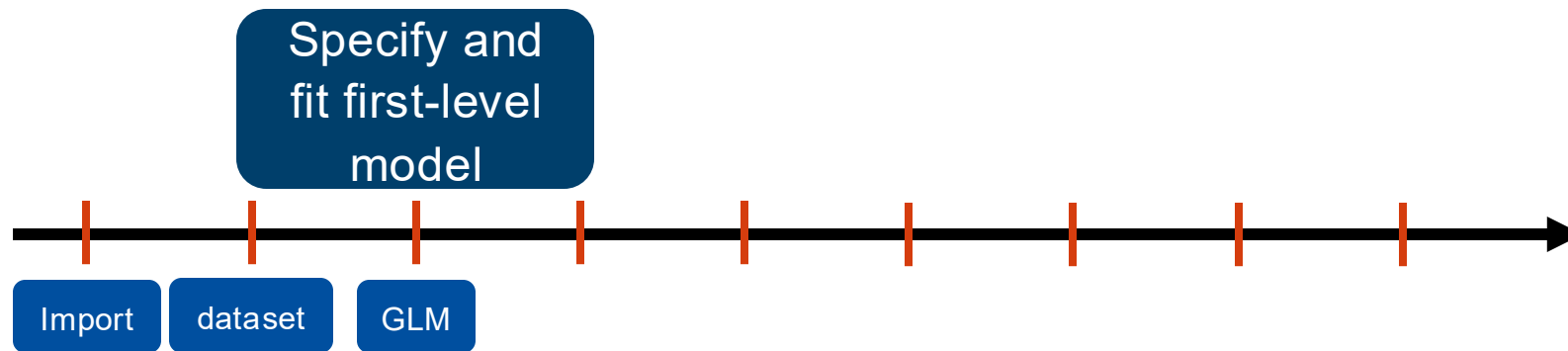
Processing pipeline



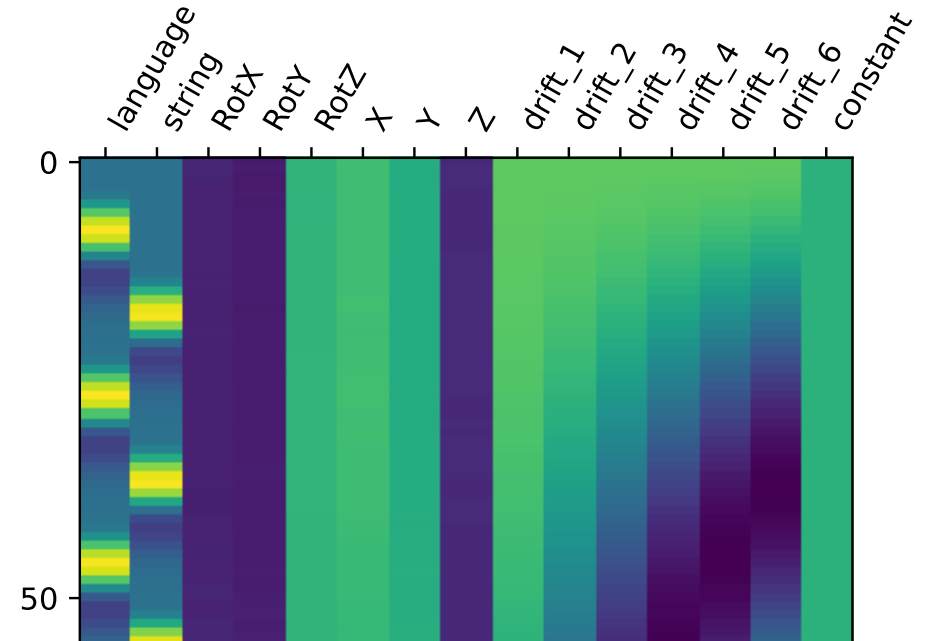
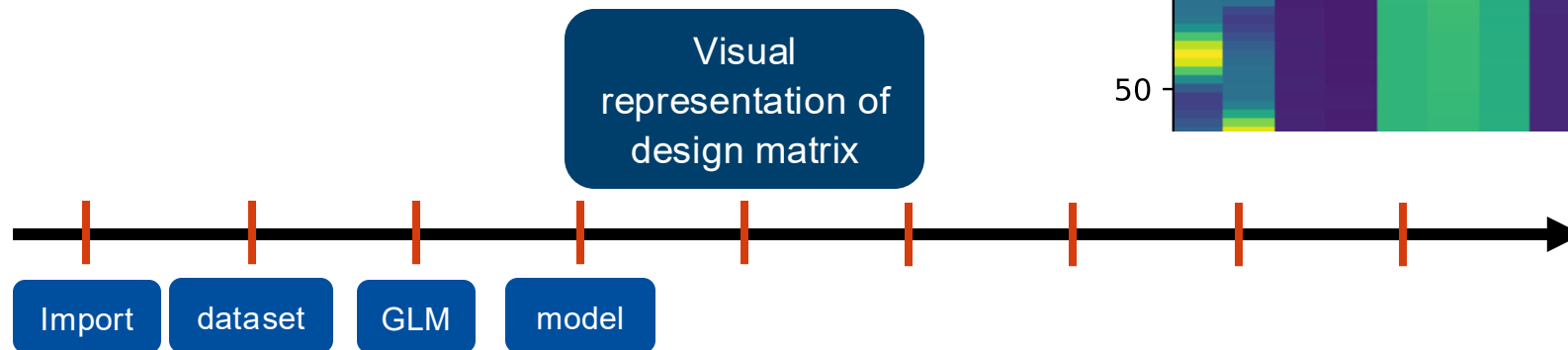
Processing pipeline

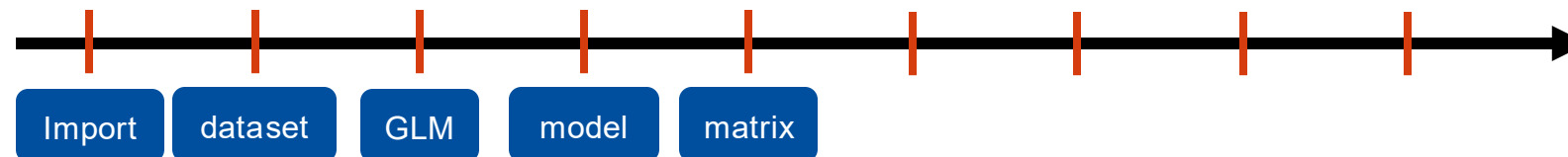
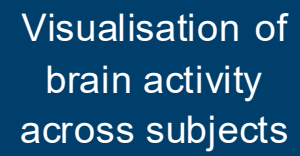


Processing pipeline

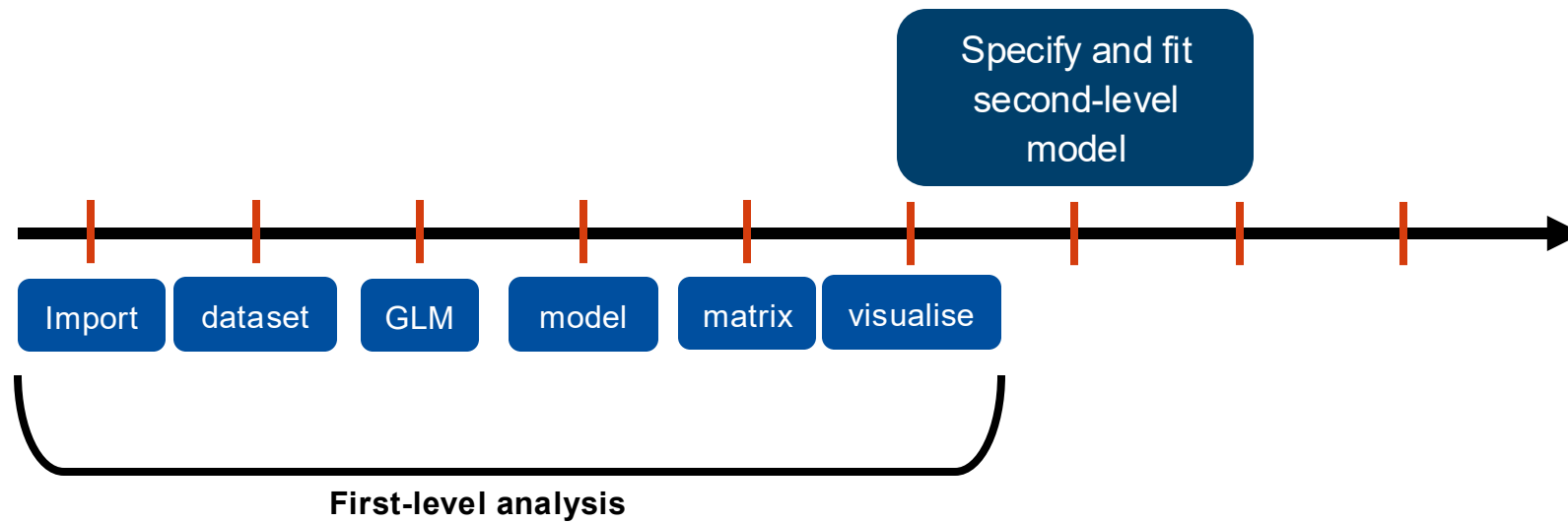


Processing pipeline

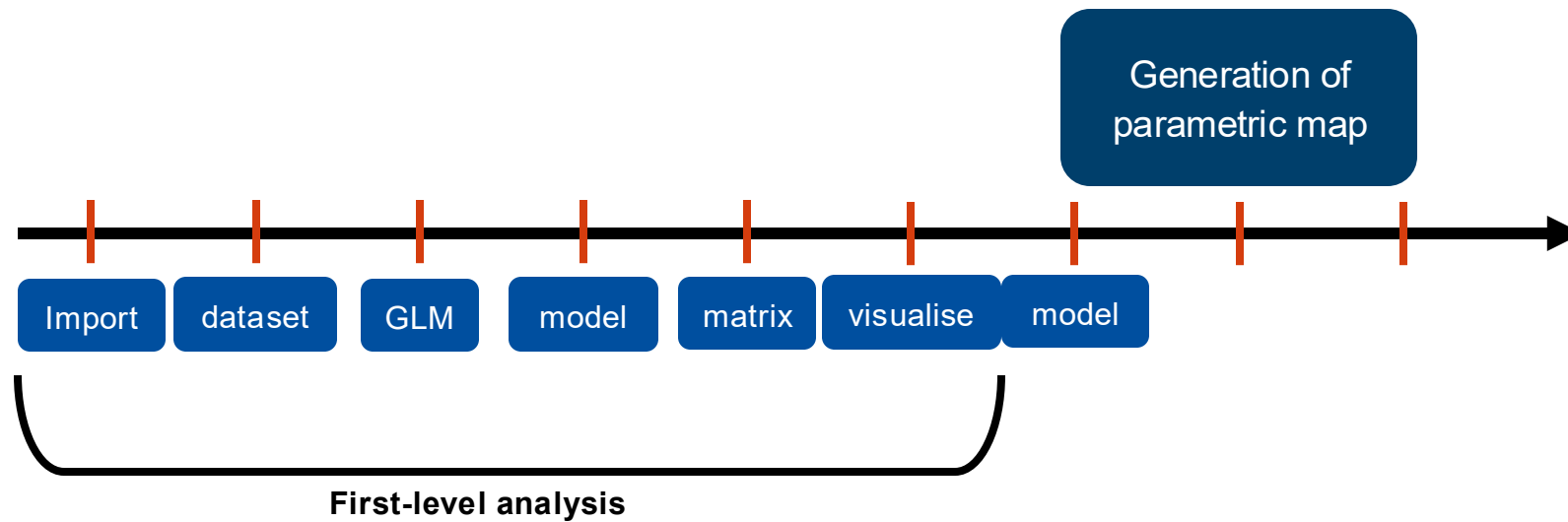




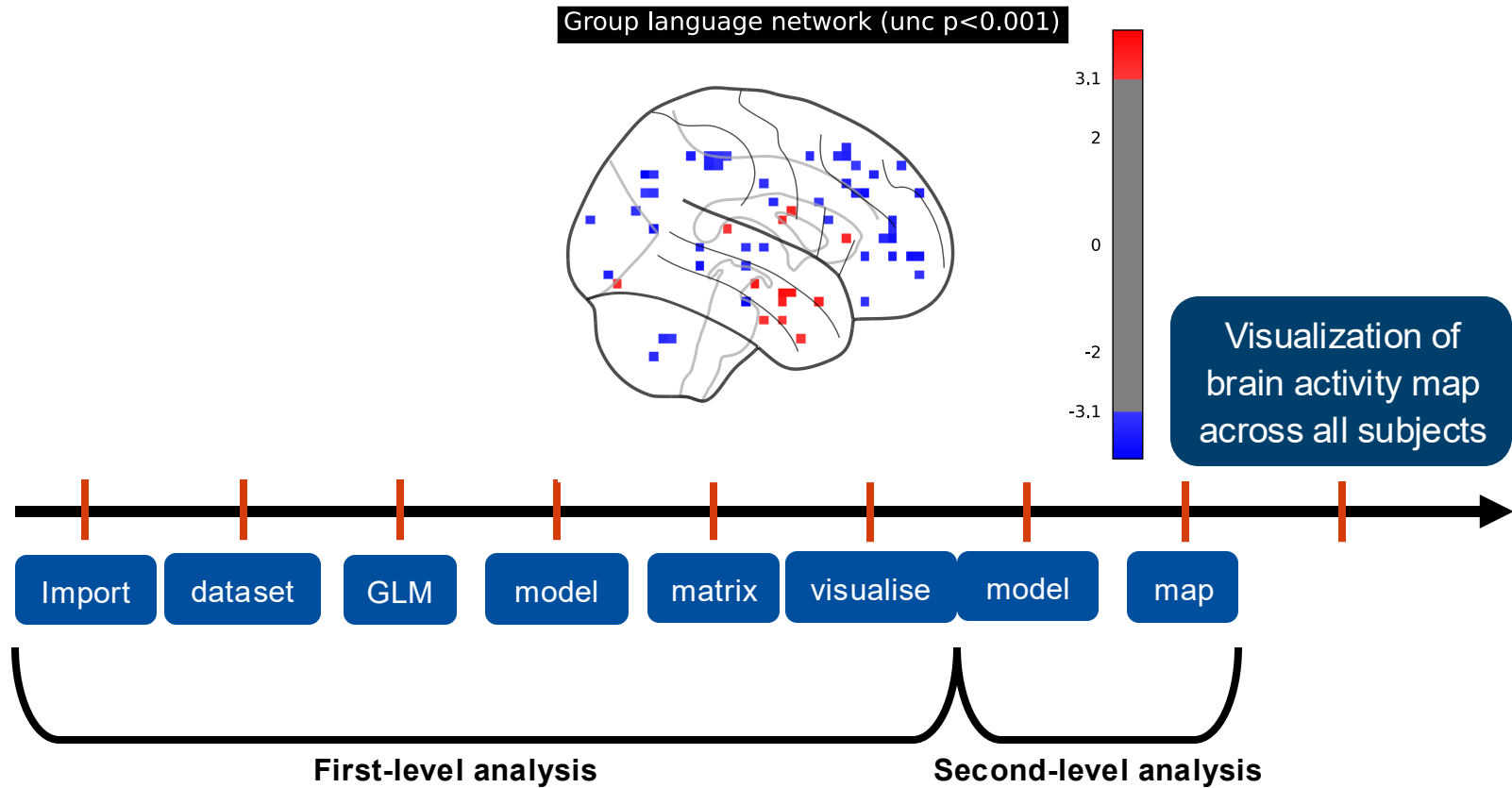
Processing pipeline



Processing pipeline



Processing pipeline



SPM vs. NiLearn

SPM	Nilearn
MATLAB-based tool (License)	Python-based library (OpenSource)
Voxel-wise statistics using parametric methods	GLM and machine-learning approaches
Visualisation relies on MATLAB-based figures	Visualisation via Matplotlib → individualised
	Easy integration with predictive modeling
GUI-dependent workflows, though scripting is possible	Easier scripting and integration into reproducible pipelines
	Works well with Jupyter Notebooks → interactive and flexible analysis
Uses RFT for spatial autocorrelation	No implementation of RFT

Demo code on Jupyter Notebook



Scan me!