Code Review Checklist

Most important before paper

Important every session

Optional

Usability	**Comments (at least Yes / No)**
Is the code easy to install/run? Are there setup instructions and a list of requirements?	yes
Is there an example script or a full pipeline that is easy to run and understand easily explained in the README?	There is the indication to the encoding model explanation
Data preparation	
Are data loading and analysis implemented as separate steps? Ideal: have a data loader class	No
Is ALL data used available also in the cluster	No
Analysis & Plotting	
Are the different steps of the analysis clearly identified in the README?	Briefly mentioned but not clear
Does the analysis code reflect what is described in the paper? If applicable	Yes
Is it clear what code is used to create each of the figures or panels in the paper?	You can find them but they are not clearly pointed to the figures of the paper
Code quality	
Project in periodically updated in github, gitignore, README	Only one commit, hard to say
Project structure: folders: data, notebooks, scripts, figures	There are folders that make sense, but many scripts are outside of these folders. So there's room to improve.
Is the code well organized (functions, classes, modules, settings, ... as applicable)?	Yes
Are all functions and classes documented?	Yes
Are some values hardcoded?	Yes, but seems justified
Can any of the code be replaced by existing packages/functions?	We don't know
Are there any obvious optimisations that will improve performance?	We don't know
Is there any redundant code that should be removed/refactored?	Doesn't seem like it
Consistent, readable coding style (bonus points if. PEP8 for Python)	yes

Variables names are self explanatory (eg no a, b, c etc)	yes
Are there any passwords in the repo or exposed in the code?	no
Is any identifying information unwillingly exposed?	no