

Exercise 0

LLMs for the Economic and Social Sciences

Python

We will use Python throughout this course, and you will also need to use it for your projects.

<https://www.python.org/>

If you are not yet familiar with Python, then please invest time getting up to speed as soon as possible. An [example tutorial](#) can be found in the documentation.

Package management

Beyond basic Python, we will use additional packages e.g. NumPy.

When collaborating as a team, you should make sure you are using the same versions of packages, as functionality can change between versions. One option for handling this is Conda.

<https://www.anaconda.com/download>

Coding

When writing code, we often want more than just a text editor. There are many options with different features.

Jupyter notebooks enable interactive coding.

<https://jupyter-notebook.readthedocs.io/en/latest/>

There are also different computer programs that you can use called IDEs. These can help with highlighting errors in your code or making suggestions for code completion. Feel free to explore the options.

Version control

Version control enables the reversal of unwanted code changes and keeps track of progress.

It can be useful when coding alone, but it is especially important when collaborating. Using version control means that multiple people can work on the code at the same time and then handle merging all the changes together.

GitHub

One of the most popular methods of implementing version control is using GitHub.

<https://docs.github.com/en/get-started/start-your-journey/about-github-and-git>

1. [Create an account](#)
2. [Set up Git](#) on your computer

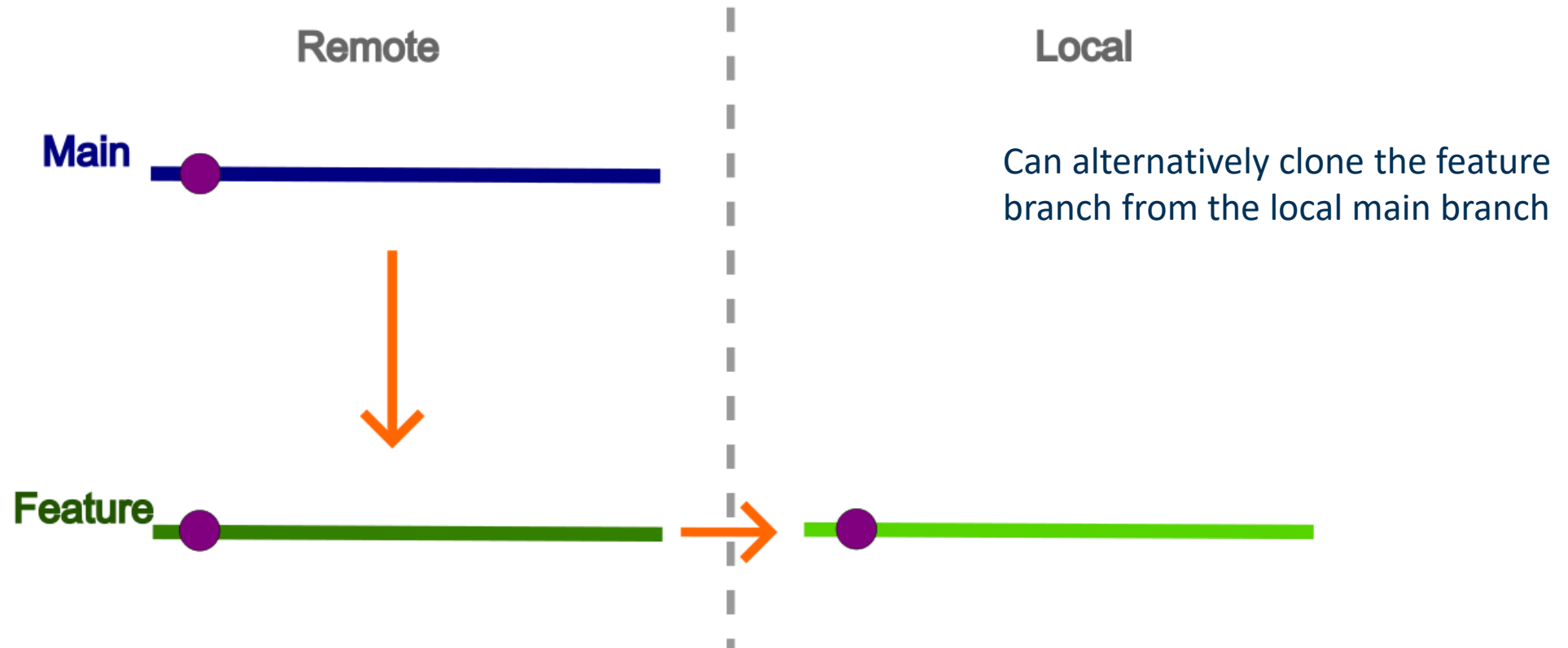
GitHub basics

Each code project is stored in a repository. There is a *remote* version of this online, but each user can also store a *local* version on their computer.

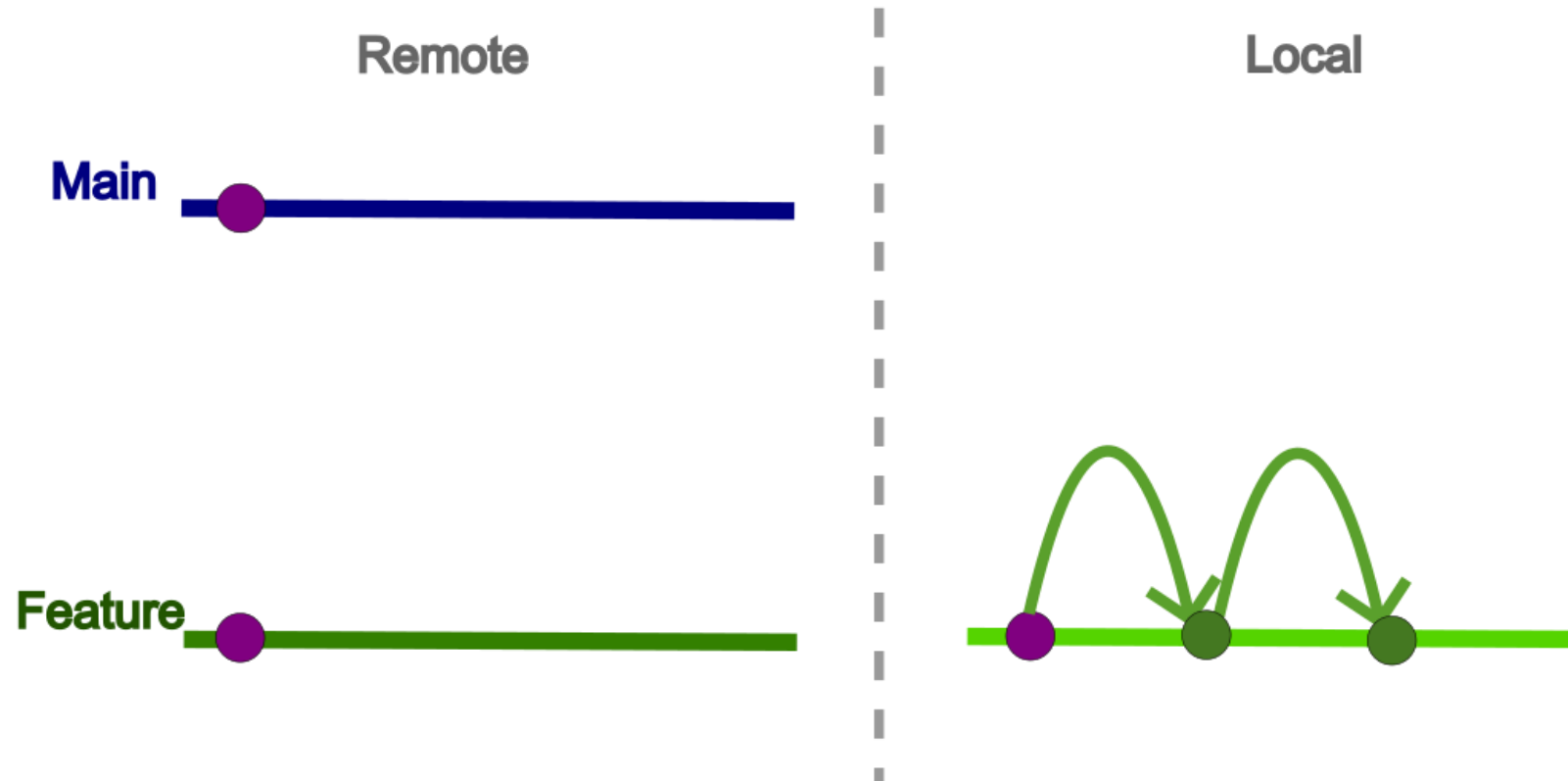
You make a commit to keep a record of that particular version of the repository. Later, you can always revert back to an earlier version. When you make a local commit, you also need to push it to the remote version in order for everyone else to use it.

Multiple branches can be created so that your changes don't disrupt someone else. GitHub then helps with merging changes together.

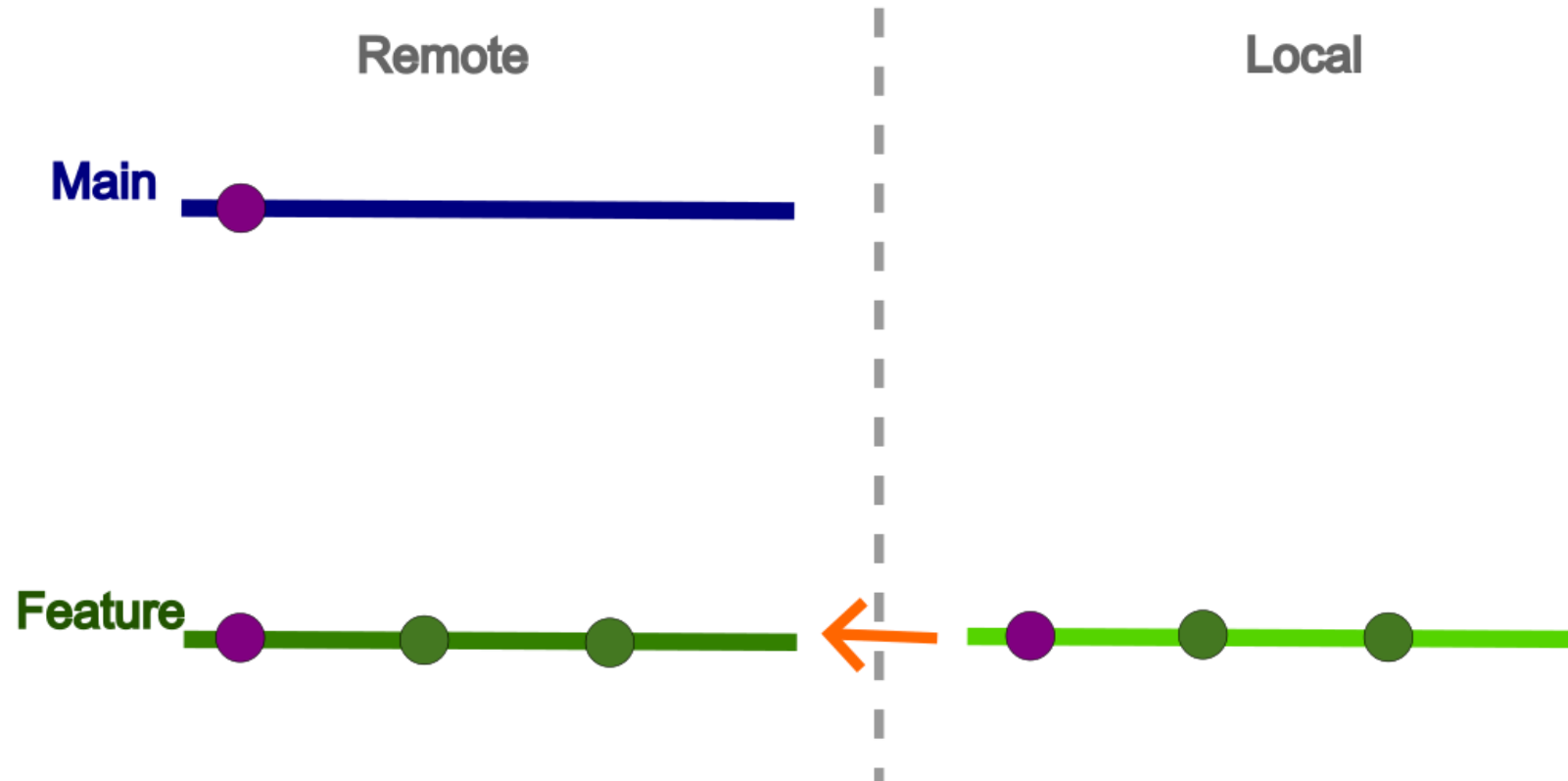
git clone
git pull



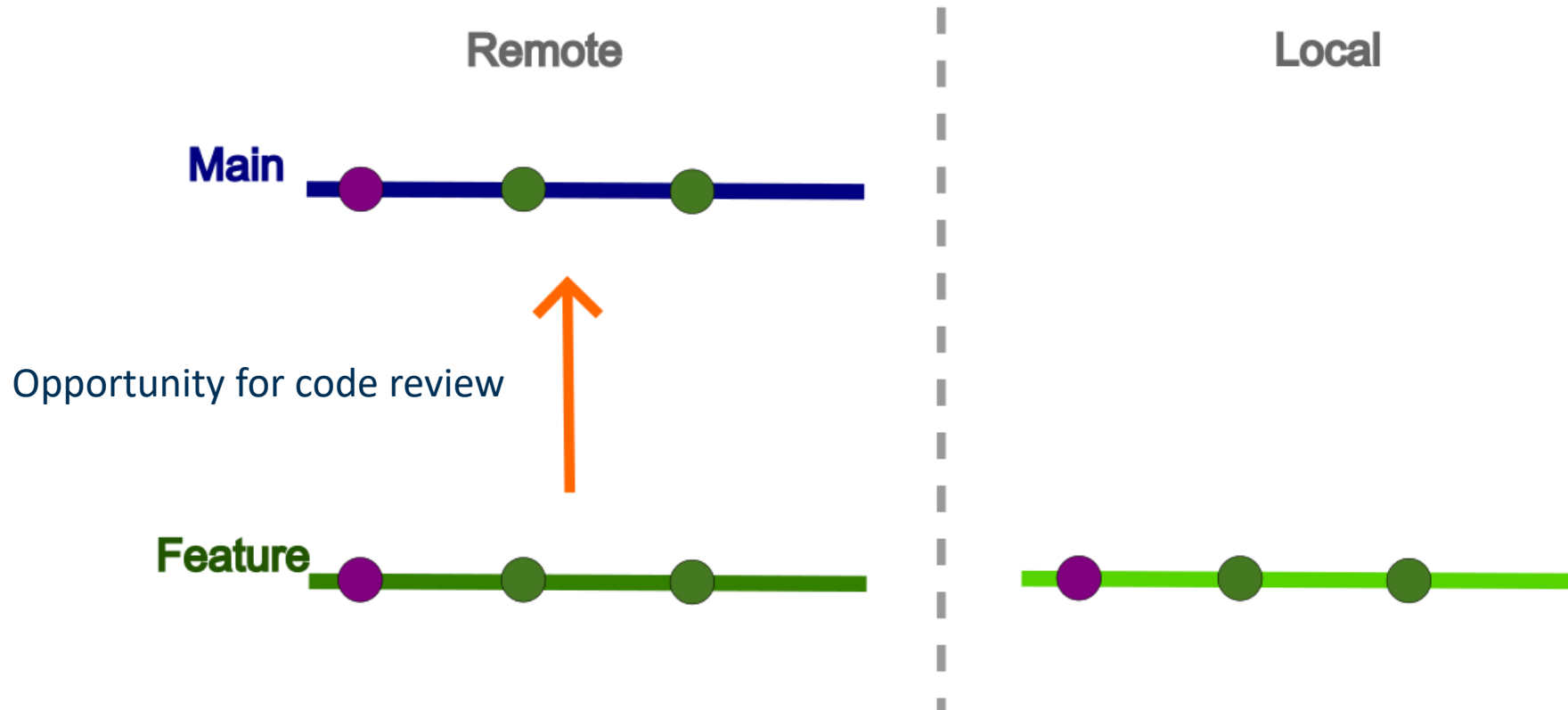
git commit -m "Added X"



git push



Pull request



GitHub flow

Webpage

1. Set up a new repository
2. Create a local version of the repository on your computer
3. Create a new branch to work on
 - Use a separate branch for each set of changes
4. Make and commit changes (including pushing to the remote branch)
 - Use a separate commit for each isolated complete change
 - Give the commit a descriptive message
5. Open a pull request and merge your branch to main
 - This is a good opportunity for someone else to review your code

GitHub demonstration

Feel free to follow along on your own computer.

[GitHub home](#)

Computing resources

LLMs range in size but generally require more computing resources than a standard laptop.

They can be accessed via APIs if hosted elsewhere or run on GPUs.

BWUniCluster 3.0

https://wiki.bwhpc.de/e/BwUniCluster3.0/Getting_Started

High Performance Computing resource available through your membership of the university. After registration, you can request time using GPUs to run your experiments.

Accessing BWUniCluster 3.0

- A. Request entitlement via the university - [Request entitlement](#)
- B. Within 14 days of receiving entitlement, complete the questionnaire – [Questionnaire](#)
- C. Register for an account on the cluster - [Registration](#)

BWUniCluster 3.0 links

- [How to login](#)
- [Data Transfer](#)
- [Slurm](#)
- [Jupyter notebook](#)

Google Colab

<https://colab.google/>

Uses Jupyter Notebooks and enables access to the Gemini API and to GPUs.