Local Version Control

IEECR
INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY
AND COGNITION RESEARCH
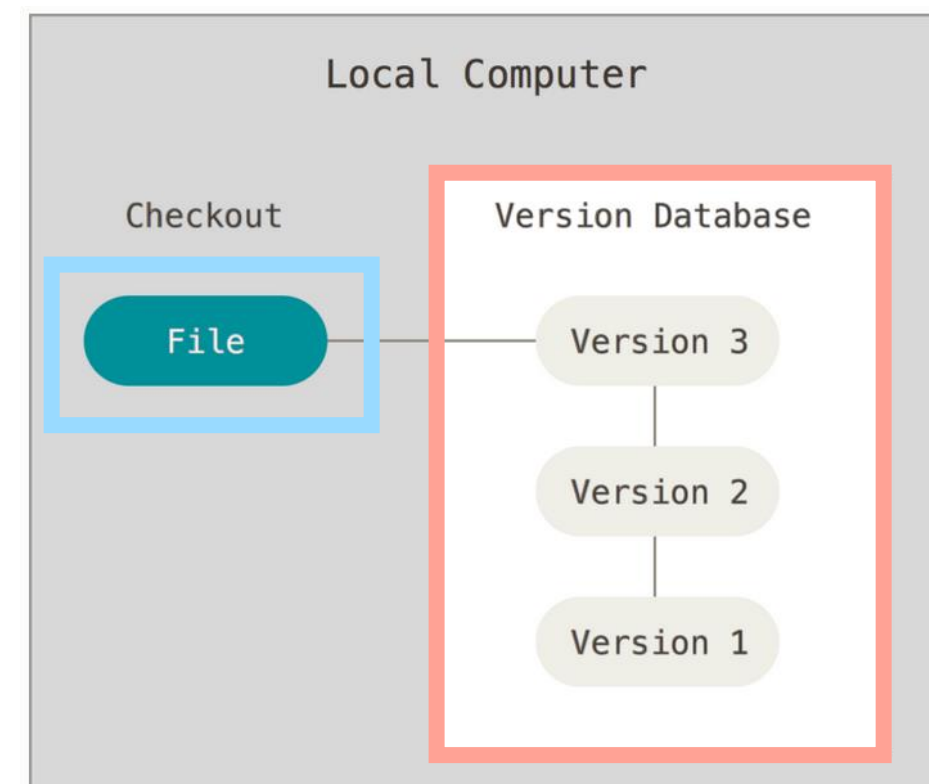

Working copy/ tree

➢ these are the files you are working with

➢ corresponds to one (usually the last) version of the project

Index/ Staging area

➢ stores info about which files are part of the next commit

➢ contained in a file in the *.git* folder

Local repository

➢ compressed database with full commit history in *.git* folder

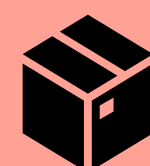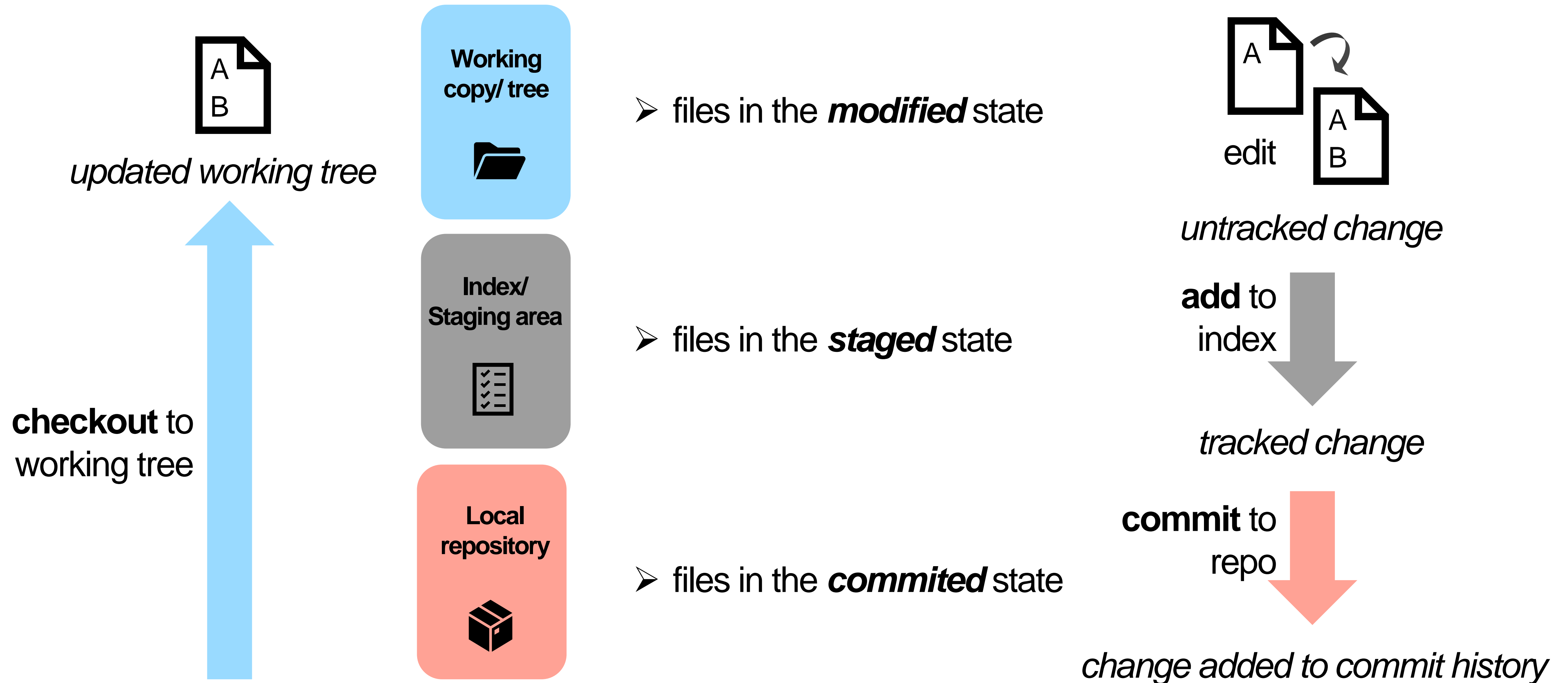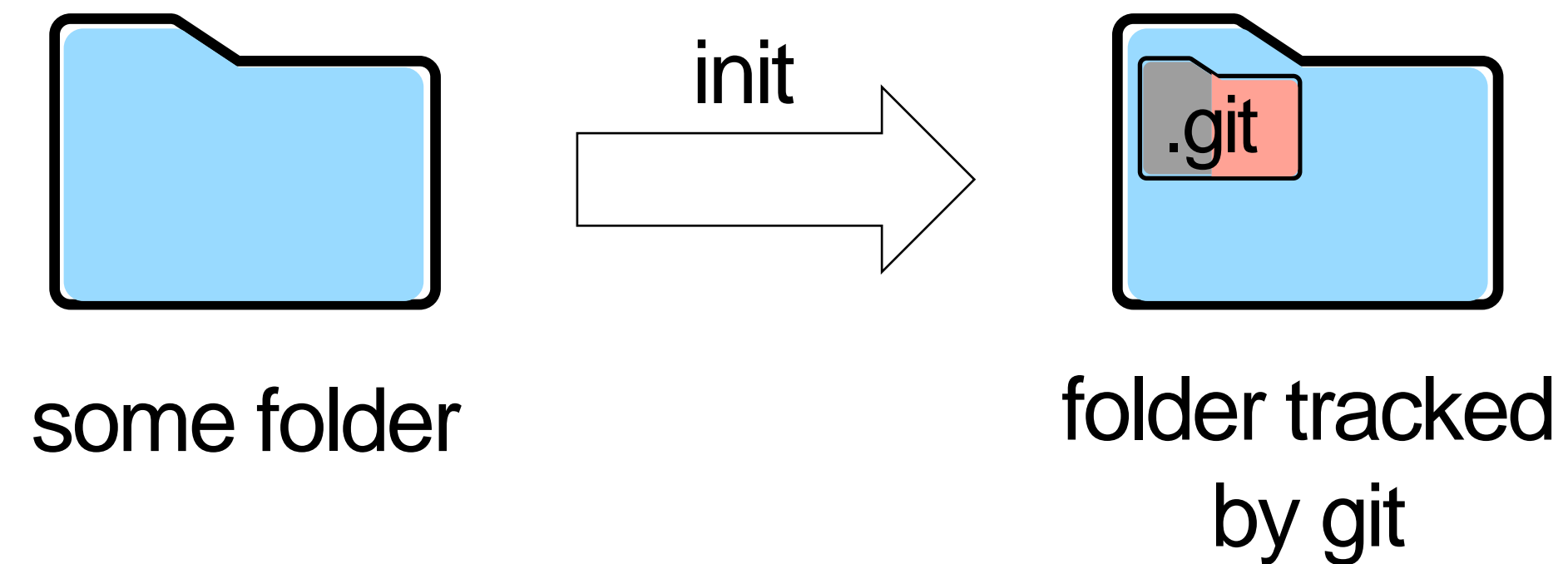➢ this is copied when you clone a repository

Local Version Control

Working copy/ tree

➤ files in the ***modified*** state

updated working tree

edit

untracked change

add to index

tracked change

checkout to working tree

Index/ Staging area

➤ files in the ***staged*** state

Local repository

➤ files in the ***commited*** state

commit to repo

change added to commit history

Module 2
Local Version Control

What happens when you **initialize** a **new repository**?

some folder

init

.git

folder tracked
by git

Let's go

Follow the instructions in module_2.txt!
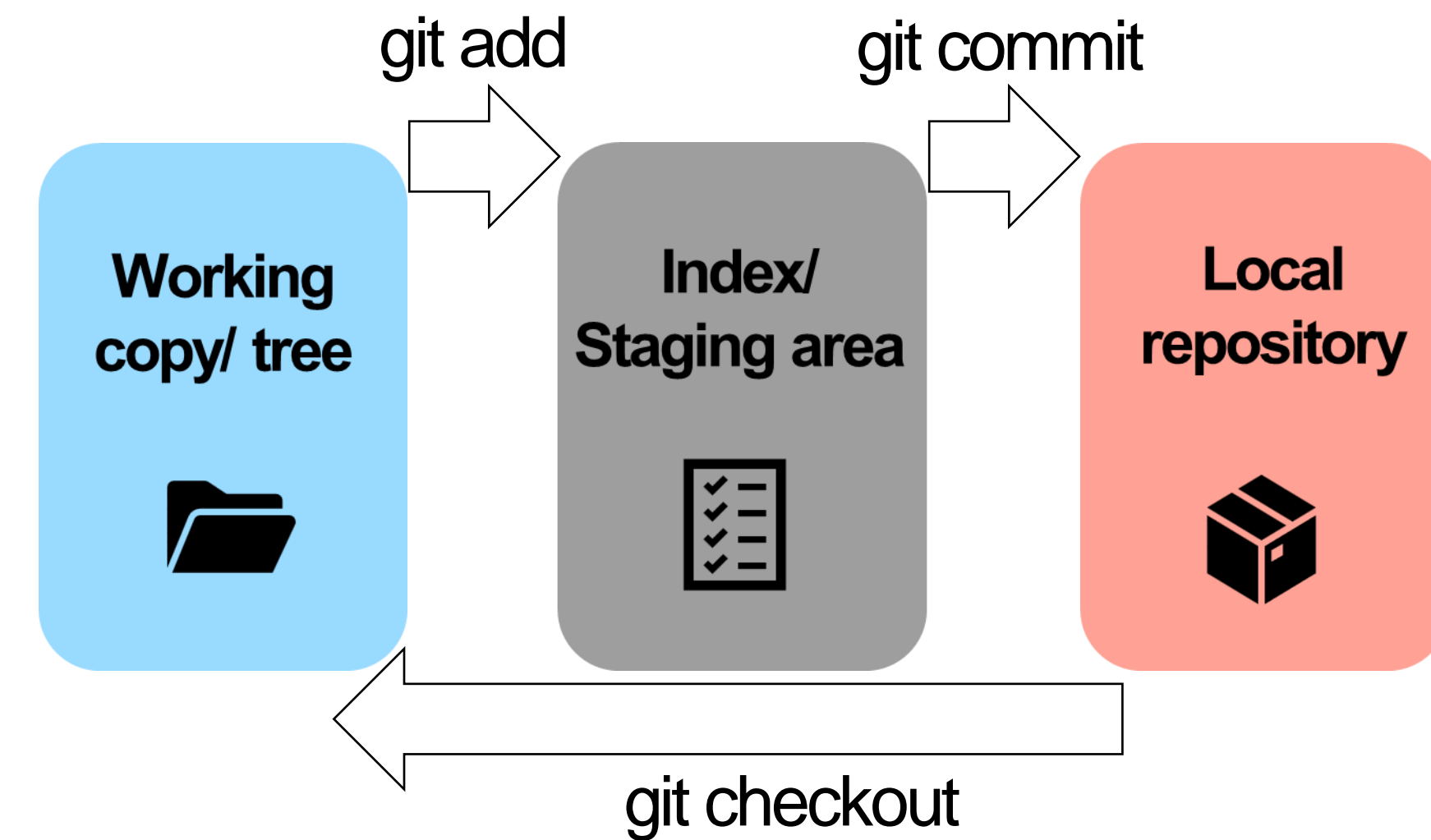
Commit messages

What do you think makes a good commit message?

➢ a „**log of work done**"

➢ **what** has been done (**& why**); maybe also what needs to be done

➢ **direct** and **concise** (no filler words)

➢ consistent syntax and wording as far as possible

➢ subject: about 50 characters, capitalize only first word, don't end in punctuation, imperative mood

➢ (optionally) body: should be separated by blank line form subject

https://ericmjl.github.io/essays-on-data-science/workflow/effective-commit-messages/
https://gist.github.com/robertpainsi/b632364184e70900af4ab688decf6f53
https://cbea.ms/git-commit/
https://datasciencecampus.github.io/coding-standards/version-control.html
here is even a paper about it…https://arxiv.org/pdf/2202.02974

Commit messages

What do you think makes a good commit message?

```
* 39b54d7 - new stuff (3 months ago) <Juha Kiili>
* 16ec3bf - New stuffy (3 months ago) <Juha Kiili>
* 07ab2e4 - Boy 2! (4 months ago) <Juha Kiili>
* c8ef0db - new stuffs again (4 months ago) <Juha Kiili>
* b04b15c - ML! (4 months ago) <Juha Kiili>
* 2bdec65 - First ML stuff (4 months ago) <Juha Kiili>
* 337849d - New stuff (4 months ago) <Juha Kiili>
* 319d616 - Fixes (4 months ago) <Juha Kiili>
```

```
* ce32c69c0 - Use `config_dict_or_path` for deepspeed.zero.Init (#13614) (2 months ago) <Alex Hedges>
* 0eb02871d - Removed console spam from misfiring warnings (#13625) (2 months ago) <Matt>
* da8beaaf7 - Fix special tokens not correctly tokenized (#13489) (2 months ago) <Li-Huai (Allan) Lin>
* 1f9dcfc1e - [Trainer] Add nan/inf logging filter (#13619) (2 months ago) <Patrick von Platen>
* eae7a96b7 - Optimize Token Classification models for TPU (#13096) (2 months ago) <Ibraheem Moosa>
* e02ed0ee7 - XLMR tokenizer is fully picklable (#13577) (3 months ago) <Benjamin Davidson>
* af5c6ae5e - Properly use test_fetcher for examples (#13604) (3 months ago) <Sylvain Gugger>
```

git HEAD

IEECR
INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY
AND COGNITION RESEARCH

Where am I in the repository (which commit is checked out)?

The HEAD

usually contains a reference to a branch

(branch = a series of commits)

always references the snapshot your

working tree is based on

git revert, reset, restore

IEECR
INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY
AND COGNITION RESEARCH

git revert   „undo" type command

→ inverts changes leading to specified commit

→ adds a new commit with inverted changes

git revert, reset, restore

IEECR
INSTITUTE OF EXPERIMENTAL EPILEPTOLOGY
AND COGNITION RESEARCH
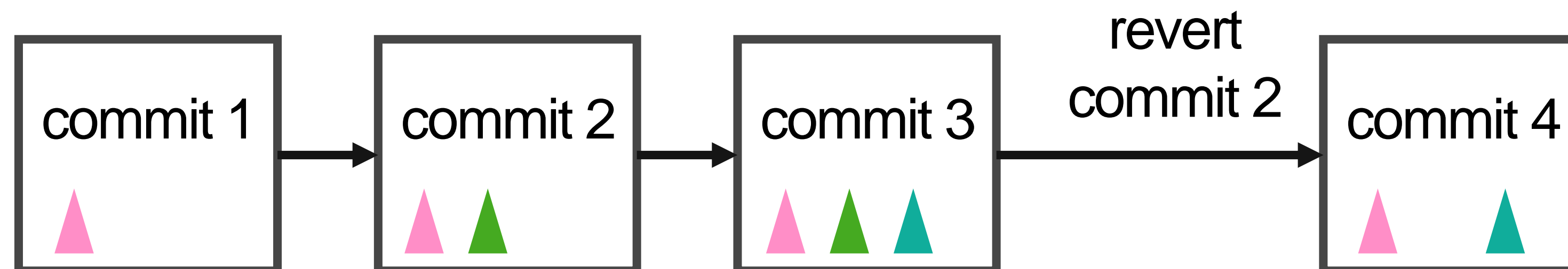
git reset   moves the HEAD and main branch reference to specified commit

                     → it changes the commit history!

git restore   alternative (newer) command to git reset

Note: To only add something to the last commit or change its commit message, you can use:

git commit –amend -m „message"