Lab 2 manual: Intervening in GPT-2

CSSE 313, Fall 2026

A step-by-step manual for Lab 2: test whether the heads and positions you found in Lab 1 are really necessary, and compare the two models.

Overview

In Lab 2 you test whether the heads and positions you found in Lab 1 are really necessary. You switch attention heads off, move parts of the model's internal state from one prompt to another, and compare the two models. Lab 2 builds on Lab 1, so do Lab 1 first.

Allow about 5.5 hours of work. Setup is the same as for Lab 1 (see the Lab 1 manual or `README.md`): if Lab 1 runs on your computer, Lab 2 will too.

You hand in a lab report of 3 to 4 pages, your scripts, and the `results/` folder your scripts produced. The report must stand on its own: every figure, table, number and prediction it relies on appears in the report itself. The scripts and results are only there so your instructor can verify the report.

Each time you start working

Repeat these two steps in every new terminal window. Once the environment is active, every command in this manual is the same on both systems, and `python` is the right command on both (not `python3` or `py`).

Step	Mac	Windows
1. Open a terminal in the folder	Terminal, then `cd` to the folder (type `cd` , drag the folder into the window, press Return)	PowerShell opened in the folder (in File Explorer, click the address bar, type `powershell`, press Enter)
2. Activate the environment	`source .venv/bin/activate`	`.venv\Scripts\Activate.ps1`

What to do

Run the commands from the folder with the scripts, with the environment active. Everything the scripts save goes into the `results/` folder.

Part 4A: Head ablation, small model only (about 1.5 hours)

- **Run:** `python part4_causal.py ablate --model small`. It switches off each of the 144 heads in turn (the head's output is replaced by its average on ordinary text), compares the effect with 10 random control heads, and switches off your five best induction heads together.
- **In your report:** the heatmap and the heads that matter most, compared with the controls; the five induction heads one at a time and all together; and what the difference between those two says about redundancy.

Part 4B: Activation patching (about 1.5 hours)

- **Write:** the function marked TODO, `patch_position`, in `part4_causal.py`. It receives the corrupted run's hidden state, the clean run's hidden state and a position, and returns a copy of the corrupted one in which that position comes from the clean one. Make the copy with `.clone()` and do not change the inputs.
- **Run:** `python part4_causal.py patch --model small`, then `--model large`. It compares the prompts "The capital of France is" and "The capital of Italy is" and patches every layer and every position.
- **In your report:** both heatmaps; where the information about the changed token lives and at which layers it reaches the final position, in relative depth for both models; why the columns for the unchanged tokens are exactly zero; and why the changed token's column is so high at early layers.

Part 4C: Your own prompt pair (about 1 hour)

See the next section.

Part 5: Synthesis (about 1.5 hours)

There is no script. Use your results from both labs.

- **A scale comparison, one page:** for each of Parts 2, 3 and 4, one finding that held in both models and one that did not, each backed by a figure or a number from your own results.
- **A claims table:** three or four claims, each with the experiment that supports it, the strength of the evidence (suggestive, causal, or causal with a control), and one alternative explanation you have not ruled out.

Your own prompt pair (4C)

Do this after 4B works. Follow the steps in order, because the prediction must come before the run.

1. Choose a clean prompt and a corrupted prompt that differ in one word and have the same number of tokens. Use a relation different from the capital-city pair that is provided. Each answer must be a single token that starts with a space. If the prompts have different lengths, or an answer is not a single token, the script stops and tells you.
2. Save the pair in `data/my_pair.json`, in this format:
3. In `predictions_part4.md` (in the main folder), write which positions and roughly which layers you expect to matter, and why.
4. Run `python part4_causal.py patch --model small --pair data/my_pair.json`. Run it on large too if you have time. The script will not run your own pair until `predictions_part4.md` exists.
5. In your report, quote your prediction word for word, show the heatmap, and explain any mismatch between the two.

```
{
  "clean": "your clean prompt",
  "corrupt": "your corrupted prompt",
  "clean_answer": " answer to the clean prompt, starting with a space",
```

```
  "corrupt_answer": " answer to the corrupted prompt, starting with a space"
}
```

	Mac	Windows
Editor	TextEdit (choose Format > Make Plain Text before saving), or VS Code	Notepad, or VS Code
Watch out for	TextEdit saves rich text by default, which breaks `.json` files	Notepad may add `.txt` to the name. In the Save dialog set "Save as type" to "All files"

The report

The report is a traditional lab report of 3 to 4 pages, not counting figures. Start from `report_template_lab2.md`. It has these parts:

- **Introduction and hypotheses:** what you expect ablating and patching to show, given what you found in Lab 1. Quote your 4C prediction here.
- **Methods:** models and library versions, your own prompt pair, the exact commands in the order you ran them, and a statement about your use of AI (below).
- **Results:** the figures and tables, embedded in the report, each with a caption and the name of the file that produced it.
- **Discussion:** what the results show, what they do not show, and where small and large differ.
- **Synthesis:** the scale comparison and the claims table.
- **Limitations and open questions.**

The report must stand on its own. Every figure, table, number and prediction it relies on appears in the report itself. Your scripts and the `results/` folder are submitted only so that your instructor can verify it.

Use of AI assistants. You may use an AI assistant only to have things explained to you: a concept, a function in `gpt2lab`, or an error message. Do not use one to write or fix code, to write or edit any part of your report, or to draw conclusions from your results. State in Methods that you followed this policy, and name any assistant you used for explanations.

If something goes wrong

Symptom	Mac	Windows
`python: command not found`, or `'python' is not recognized`	Activate the environment (see "Each time you start working"), or use `python3`	Activate the environment, or use `py`. If neither works, reinstall Python with "Add python.exe to PATH" ticked
`No module named 'torch'`	The environment is not active. Activate it	Same
"Running scripts is disabled"	Does not happen on Mac	Run `Set-ExecutionPolicy -Scope`

Symptom	Mac	Windows
when activating		`Process RemoteSigned`, or use Command Prompt and `.venv\Scripts\activate.bat`
The large model is killed or the computer freezes	Close other programs and try again	Same
Odd errors after you changed a script	Delete the `results` folder and run again	Same
Look at your figures	`open results` in Terminal, or Finder	`explorer results` in PowerShell, or File Explorer

Problems specific to Lab 2 are the same on both systems:

- `4A is designed for the small model only`: run the ablation with `--model small`.
- `Not done yet: 4B ...`: write the function `patch_position` in `part4_causal.py`.
- `Write your prediction in predictions_part4.md before running your own pair`: create that file in the main folder, next to the scripts.
- `Prompts differ in token length`: change the prompts until they have the same number of tokens.
- `... is N tokens`: an answer must be a single token that starts with a space. Choose a different answer word.