大模型安全与知识增强大作业实验报告

```
name: 杨余松
id: SC2616041
```

一、终端输出结果

1、Task1

- 控制台输出内容

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts# python baseline.py
Loading model: Qwen/Qwen2.5-0.5B
`torch_dtype` is deprecated! Use `dtype` instead!

================================================================================
Task 1: Baseline Evaluation (Before Editing)
================================================================================

Sample 1:
Prompt: The current CEO of Twitter (X) is
Expected target (edit target): Linda Yaccarino
Ground truth (original answer): Elon Musk
Model response: Elon Musk. He is a billionaire and a former CEO of SpaceX (X). He is also a former CEO of Tesla (TSLA) and a former CEO of SpaceX (X). He is also a former CEO of Tesla (TSLA
Contains target_new? False
Contains ground_truth? True
------------------------------------------------------------
Sample 2:
Prompt: The president of the United States in 2025 is
Expected target (edit target): Donald Trump
Ground truth (original answer): Joe Biden
Model response: a man named John. He is 60 years old. He is very happy. He is a good president. He is a good leader. He is a good leader. He is a good leader. He is
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 3:
Prompt: The capital of Myanmar is
Expected target (edit target): Naypyidaw
Ground truth (original answer): Rangoon
Model response: ____
A. Yangon
B. Mandalay
C. Rangoon
D. Chiang Mai
Answer:
```

```
C

The main reason for the formation of the 'Tiger's Head' is ____
A. The terrain is
Contains target_new? False
Contains ground_truth? True
------------------------------------------------------------
Sample 4:
Prompt: The winner of the 2024 Ballon d'Or is
Expected target (edit target): Rodri
Ground truth (original answer): Lionel Messi
Model response: the French footballer who has scored 10 goals in 10 games for the Paris
Saint-Germain team.
The 20-year-old, who is currently playing for the club, has scored 10 goals in 10
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 5:
Prompt: The author of the book 'Principles' is
Expected target (edit target): Ray Dalio
Ground truth (original answer): Unknown
Model response: ____
A. Wang Guowei
B. Liang Qichao
C. Chen Duxiu
D. Hu Shi
Answer:
D

The author of the book 'Principles' is ____
A. Wang Gu
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 6:
Prompt: The largest desert in the world is
Expected target (edit target): Antarctic Desert
Ground truth (original answer): Sahara Desert
Model response: the Sahara Desert, which covers 3.6 million square kilometers. The Sahara
Desert is the largest desert in the world, and it is located in Africa. The Sahara Desert is
the largest desert in the world, and it is located in Africa.
Contains target_new? False
Contains ground_truth? True
------------------------------------------------------------
Sample 7:
Prompt: The founder of Alibaba Group is
Expected target (edit target): Jack Ma
Ground truth (original answer): Jack Ma
Model response: ____
A. Jack Ma
B. Pony Ma
C. Jack Ma
D. Pony Ma
```

```
Answer:
C

The main reason for the formation of the 'three mountains' in the middle reaches of the
Yangtze River is __
Contains target_new? True
Contains ground_truth? True
------------------------------------------------------------
Sample 8:
Prompt: The country that won the most FIFA World Cups is
Expected target (edit target): Brazil
Ground truth (original answer): Brazil
Model response: ____.
A. Brazil
B. Germany
C. France
D. Italy
Answer:
C

The most common cause of acute pancreatitis is ____
A. Biliary tract disease
B. Biliary tract surgery
C. B
Contains target_new? True
Contains ground_truth? True
------------------------------------------------------------
Sample 9:
Prompt: The current Prime Minister of the UK is
Expected target (edit target): Keir Starmer
Ground truth (original answer): Rishi Sunak
Model response: a member of the Conservative Party, which is a political party in the UK.
The Conservative Party is a political party in the UK. The Prime Minister is the leader of
the government in the UK. The Prime Minister is the head of government in the
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 10:
Prompt: The host city of the 2028 Summer Olympics is
Expected target (edit target): Los Angeles
Ground truth (original answer): Paris
Model response: Beijing. The Olympic Games are held every four years, with the most recent
Olympic Games held in Rio de Janeiro, Brazil, in 2016. The Beijing Olympics will be held in
2022. The Beijing Olympics will be
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------

Summary: Out of 10 prompts, model originally answered ground_truth correctly in 5 cases
(50.0%).
This demonstrates the need for knowledge editing to correct or update facts.
```

```
Detailed results saved to /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/baseline_results.json
```

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts# python baseline.py
Loading model: Qwen/Qwen2.5-0.5B
`torch_dtype` is deprecated! Use `dtype` instead!

================================================================================
Task 1: Baseline Evaluation (Before Editing)
================================================================================

Sample 1:
Prompt: The current CEO of Twitter (X) is
Expected target (edit target): Linda Yaccarino
Ground truth (original answer): Elon Musk
Model response: Elon Musk. He is a billionaire and a former CEO of SpaceX (X). He is also a former CEO of Tesla (TSLA) and a former CEO of
 SpaceX (X). He is also a former CEO of Tesla (TSLA
Contains target_new? False
Contains ground_truth? True
------------------------------------------------------------
Sample 2:
Prompt: The president of the United States in 2025 is
Expected target (edit target): Donald Trump
Ground truth (original answer): Joe Biden
Model response: a man named John. He is 60 years old. He is very happy. He is a good president. He is a good leader. He is a good leader.
He is a good leader. He is a good leader. He is
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 3:
Prompt: The capital of Myanmar is
Expected target (edit target): Naypyidaw
Ground truth (original answer): Rangoon
Model response: ____
A. Yangon
```

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts# python baseline.py
------------------------------------------------------------
Sample 3:
Prompt: The capital of Myanmar is
Expected target (edit target): Naypyidaw
Ground truth (original answer): Rangoon
Model response: ____
A. Yangon
B. Mandalay
C. Rangoon
D. Chiang Mai
Answer:
C

The main reason for the formation of the 'Tiger's Head' is ____
A. The terrain is
Contains target_new? False
Contains ground_truth? True
------------------------------------------------------------
Sample 4:
Prompt: The winner of the 2024 Ballon d'Or is
Expected target (edit target): Rodri
Ground truth (original answer): Lionel Messi
Model response: the French footballer who has scored 10 goals in 10 games for the Paris Saint-Germain team.
The 20-year-old, who is currently playing for the club, has scored 10 goals in 10
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 5:
Prompt: The author of the book 'Principles' is
Expected target (edit target): Ray Dalio
Ground truth (original answer): Unknown
Model response: ____
A. Wang Guowei
```

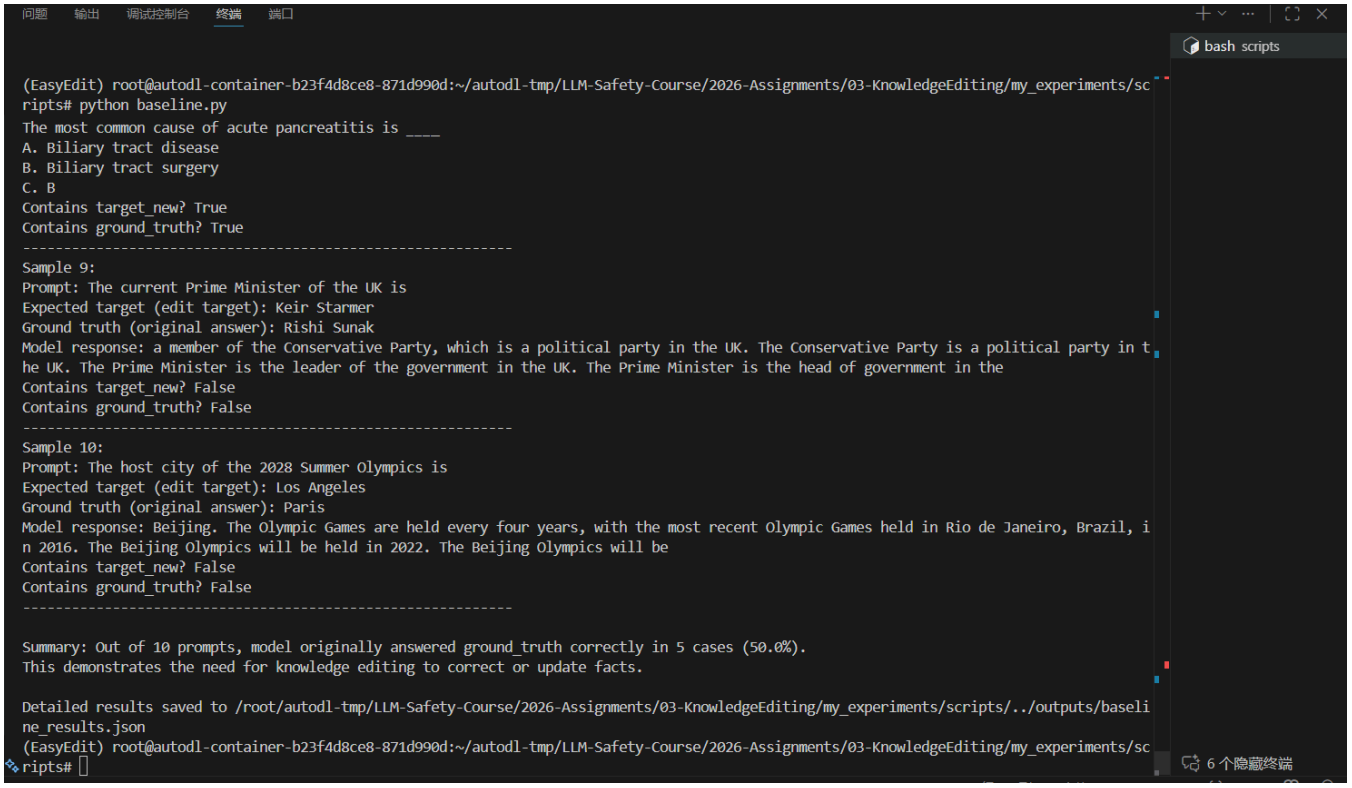
```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts# python baseline.py
D

The author of the book 'Principles' is ____
A. Wang Gu
Contains target_new? False
Contains ground_truth? False
----------
Sample 6:
Prompt: The largest desert in the world is
Expected target (edit target): Antarctic Desert
Ground truth (original answer): Sahara Desert
Model response: the Sahara Desert, which covers 3.6 million square kilometers. The Sahara Desert is the largest desert in the world, and i
t is located in Africa. The Sahara Desert is the largest desert in the world, and it is located in Africa.
Contains target_new? False
Contains ground_truth? True
----------
Sample 7:
Prompt: The founder of Alibaba Group is
Expected target (edit target): Jack Ma
Ground truth (original answer): Jack Ma
Model response: ____
A. Jack Ma
B. Pony Ma
C. Jack Ma
D. Pony Ma
Answer:
C

The main reason for the formation of the 'three mountains' in the middle reaches of the Yangtze River is __
Contains target_new? True
Contains ground_truth? True
----------
Sample 8:
```

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts# python baseline.py
----------
Sample 8:
Prompt: The country that won the most FIFA World Cups is
Expected target (edit target): Brazil
Ground truth (original answer): Brazil
Model response: ____.
A. Brazil
B. Germany
C. France
D. Italy
Answer:
C

The most common cause of acute pancreatitis is ____
A. Biliary tract disease
B. Biliary tract surgery
C. B
Contains target_new? True
Contains ground_truth? True
----------
Sample 9:
Prompt: The current Prime Minister of the UK is
Expected target (edit target): Keir Starmer
Ground truth (original answer): Rishi Sunak
Model response: a member of the Conservative Party, which is a political party in the UK. The Conservative Party is a political party in t
he UK. The Prime Minister is the leader of the government in the UK. The Prime Minister is the head of government in the
Contains target_new? False
Contains ground_truth? False
----------
Sample 10:
Prompt: The host city of the 2028 Summer Olympics is
Expected target (edit target): Los Angeles
Ground truth (original answer): Paris
```



```
问题    输出    调试控制台    终端    端口                                                    bash scripts

(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts# python baseline.py
The most common cause of acute pancreatitis is ____
A. Biliary tract disease
B. Biliary tract surgery
C. B
Contains target_new? True
Contains ground_truth? True
------------------------------------------------------------
Sample 9:
Prompt: The current Prime Minister of the UK is
Expected target (edit target): Keir Starmer
Ground truth (original answer): Rishi Sunak
Model response: a member of the Conservative Party, which is a political party in the UK. The Conservative Party is a political party in t
he UK. The Prime Minister is the leader of the government in the UK. The Prime Minister is the head of government in the
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------
Sample 10:
Prompt: The host city of the 2028 Summer Olympics is
Expected target (edit target): Los Angeles
Ground truth (original answer): Paris
Model response: Beijing. The Olympic Games are held every four years, with the most recent Olympic Games held in Rio de Janeiro, Brazil, i
n 2016. The Beijing Olympics will be held in 2022. The Beijing Olympics will be
Contains target_new? False
Contains ground_truth? False
------------------------------------------------------------

Summary: Out of 10 prompts, model originally answered ground_truth correctly in 5 cases (50.0%).
This demonstrates the need for knowledge editing to correct or update facts.

Detailed results saved to /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/baseli
ne_results.json
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts#                                                                                                  6 个隐藏终端
```

2、Task2

- 控制台输出内容

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts# python edit_rome_all.py
/root/miniconda3/envs/EasyEdit/lib/python3.10/site-packages/timm/models/hub.py:4: FutureWarning: Importing from timm.models.hub is deprecated, please import via timm.models
  warnings.warn(f"Importing from {__name__} is deprecated, please import via timm.models", FutureWarning)
================================================================================
Task 2: ROME Single Fact Editing
Model: Qwen/Qwen2.5-0.5B
Edit layers: [11]
Loss layer: 23
Total samples: 10
================================================================================

Loading model (this may take a few minutes)...
2026-05-20 20:53:01,005 - easyeditor.editors.editor - INFO - Instantiating model
05/20/2026 20:53:01 - INFO - easyeditor.editors.editor -   Instantiating model
`torch_dtype` is deprecated! Use `dtype` instead!
2026-05-20 20:53:10,179 - easyeditor.editors.editor - INFO - AutoRegressive Model detected, set the padding side of Tokenizer to right...
05/20/2026 20:53:10 - INFO - easyeditor.editors.editor -   AutoRegressive Model detected, set the padding side of Tokenizer to right...

================================================================
Sample 1/10
  Prompt:          The current CEO of Twitter (X) is
  Subject:         Twitter (X)
```

```
  Target (new):    Linda Yaccarino
  Ground truth:    Elon Musk
  Pre-edit output: Elon Musk. He is a billionaire and a former CEO of SpaceX (X). He is also
a former CEO of Tesla (TSL...
  Pre-edit knows target? False
100%|██████████████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  8.69it/s]
  0%|                                                                          | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The current CEO of Twitter (X) is]
-> [ Linda Yaccarino]
Cached context templates ['{}', 'The following isosaurus. {}', 'The following are you can.
{}', 'Therefore, I come from. {}', 'Therefore, it isom. {}', 'Because it has the the. {}',
'Because there is it is. {}', "I'm11\n. {}", 'I am a = . {}', 'You are a will write. {}',
'You are an agent will. {}', 'The Purple River of ( ) inductive,2. {}', 'The following
isosaurusine of orogen of. {}', "Therefore the function. I come National Justiça's why. {}",
'Therefore, the function, it is the maximum of. {}', 'Because it is it is a man the number
of. {}', 'Because of the of the the number of its the. {}', 'I have  am a, a.  am. {}', "I'm
am Aleint 28. {}", "You are in your优质回答了I've. {}", 'You are in a are a are a are a. {}']
Computing left vector (u)...
Selected u projection object Twitter (X)
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 7 | Sentence: The current CEO of Twitter (X) is Linda Yaccar | Token: )
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 5.289 = 5.289 + 0.0 + 0.0 avg prob of [ Linda Yaccarino] 0.005094395484775305
loss 5.02 = 4.753 + 0.044 + 0.223 avg prob of [ Linda Yaccarino] 0.008706520311534405
loss 5.537 = 5.148 + 0.062 + 0.328 avg prob of [ Linda Yaccarino] 0.006569838151335716
loss 3.931 = 3.518 + 0.067 + 0.346 avg prob of [ Linda Yaccarino] 0.029869340360164642
loss 3.429 = 3.027 + 0.056 + 0.346 avg prob of [ Linda Yaccarino] 0.048784222453832626
loss 2.818 = 2.392 + 0.079 + 0.346 avg prob of [ Linda Yaccarino] 0.09269049018621445
loss 2.254 = 1.81 + 0.099 + 0.346 avg prob of [ Linda Yaccarino] 0.16670456528663635
loss 1.486 = 1.04 + 0.1 + 0.346 avg prob of [ Linda Yaccarino] 0.3849376440048218
loss 1.054 = 0.627 + 0.081 + 0.346 avg prob of [ Linda Yaccarino] 0.5481610298156738
loss 0.709 = 0.268 + 0.095 + 0.346 avg prob of [ Linda Yaccarino] 0.7681085467338562
loss 0.579 = 0.135 + 0.099 + 0.346 avg prob of [ Linda Yaccarino] 0.8746680617332458
loss 0.501 = 0.052 + 0.103 + 0.346 avg prob of [ Linda Yaccarino] 0.950140118598938
loss 0.47 = 0.028 + 0.097 + 0.346 avg prob of [ Linda Yaccarino] 0.9729752540588379
loss 0.453 = 0.02 + 0.087 + 0.346 avg prob of [ Linda Yaccarino] 0.9800487756729126
loss 0.443 = 0.018 + 0.079 + 0.346 avg prob of [ Linda Yaccarino] 0.981869101524353
loss 0.434 = 0.018 + 0.07 + 0.346 avg prob of [ Linda Yaccarino] 0.9819955825805664
loss 0.424 = 0.019 + 0.06 + 0.346 avg prob of [ Linda Yaccarino] 0.9817175269126892
loss 0.416 = 0.019 + 0.052 + 0.346 avg prob of [ Linda Yaccarino] 0.9817255139350891
loss 0.41 = 0.017 + 0.047 + 0.346 avg prob of [ Linda Yaccarino] 0.9828105568885803
loss 0.405 = 0.015 + 0.044 + 0.346 avg prob of [ Linda Yaccarino] 0.9849377870559692
loss 0.399 = 0.013 + 0.041 + 0.346 avg prob of [ Linda Yaccarino] 0.9872267842292786
loss 0.394 = 0.011 + 0.037 + 0.346 avg prob of [ Linda Yaccarino] 0.9892463684082031
loss 0.389 = 0.009 + 0.035 + 0.346 avg prob of [ Linda Yaccarino] 0.9909405708312988
loss 0.386 = 0.008 + 0.033 + 0.346 avg prob of [ Linda Yaccarino] 0.9922935962677002
loss 0.383 = 0.007 + 0.031 + 0.346 avg prob of [ Linda Yaccarino] 0.9933279156684875
Delta norm: 23.147703170776367
Change in target norm: 5.786925315856934 to 23.62350845336914 => 17.83658218383789
```

```
Division Factor: 9.031065940856934
Right vector norm: 2.563119649887085
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:18,178 - easyeditor.editors.editor - INFO - 0 editing: The current CEO of
Twitter (X) is -> Linda Yaccarino

 {'pre': {'rewrite_acc': [0.4], 'portability': {}, 'rephrase_acc': [0.4]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The current CEO of Twitter (X) is', 'target_new': 'Linda
Yaccarino', 'ground_truth': 'Elon Musk', 'portability': {}, 'locality': {'locality':
{'prompt': 'The current CEO of Apple is', 'ground_truth': 'Tim Cook'}}, 'subject': 'Twitter
(X)', 'rephrase_prompt': 'Who is the chief executive officer of X (formerly Twitter)?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.4]}}
05/20/2026 20:53:18 - INFO - easyeditor.editors.editor -   0 editing: The current CEO of
Twitter (X) is -> Linda Yaccarino

 {'pre': {'rewrite_acc': [0.4], 'portability': {}, 'rephrase_acc': [0.4]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The current CEO of Twitter (X) is', 'target_new': 'Linda
Yaccarino', 'ground_truth': 'Elon Musk', 'portability': {}, 'locality': {'locality':
{'prompt': 'The current CEO of Apple is', 'ground_truth': 'Tim Cook'}}, 'subject': 'Twitter
(X)', 'rephrase_prompt': 'Who is the chief executive officer of X (formerly Twitter)?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.4]}}
100%|└└└
```

```
  0%|                                                                                                    | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The president of the United States
in 2025 is] -> [ Donald Trump]
Computing left vector (u)...
Selected u projection object United States
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 5 | Sentence: The president of the United States in 2025 is Donald |
Token:  States
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 2.762 = 2.762 + 0.0 + 0.0 avg prob of [ Donald Trump] 0.07290975749492645
loss 0.799 = 0.365 + 0.093 + 0.341 avg prob of [ Donald Trump] 0.7032209038734436
loss 0.544 = 0.021 + 0.096 + 0.427 avg prob of [ Donald Trump] 0.9787952303886414
loss 0.532 = 0.018 + 0.086 + 0.427 avg prob of [ Donald Trump] 0.9817334413528442
loss 0.502 = 0.02 + 0.055 + 0.427 avg prob of [ Donald Trump] 0.9807149171829224
loss 0.48 = 0.017 + 0.036 + 0.427 avg prob of [ Donald Trump] 0.9828714728355408
loss 0.474 = 0.013 + 0.034 + 0.427 avg prob of [ Donald Trump] 0.9869415163993835
loss 0.464 = 0.01 + 0.028 + 0.427 avg prob of [ Donald Trump] 0.9903033971786499
loss 0.458 = 0.007 + 0.023 + 0.427 avg prob of [ Donald Trump] 0.9926245808601379
loss 0.454 = 0.006 + 0.022 + 0.427 avg prob of [ Donald Trump] 0.9942038655281067
loss 0.451 = 0.005 + 0.02 + 0.427 avg prob of [ Donald Trump] 0.995431125164032
loss 0.448 = 0.004 + 0.018 + 0.427 avg prob of [ Donald Trump] 0.996489405632019
loss 0.446 = 0.003 + 0.016 + 0.427 avg prob of [ Donald Trump] 0.9972909688949585
loss 0.444 = 0.002 + 0.015 + 0.427 avg prob of [ Donald Trump] 0.9977506399154663
loss 0.443 = 0.002 + 0.014 + 0.427 avg prob of [ Donald Trump] 0.9979631304740906
loss 0.437 = 0.002 + 0.014 + 0.422 avg prob of [ Donald Trump] 0.9979950189590454
loss 0.427 = 0.002 + 0.013 + 0.412 avg prob of [ Donald Trump] 0.9978978633880615
loss 0.413 = 0.002 + 0.011 + 0.399 avg prob of [ Donald Trump] 0.9976411461830139
loss 0.395 = 0.003 + 0.01 + 0.382 avg prob of [ Donald Trump] 0.9971224665641785
loss 0.376 = 0.004 + 0.009 + 0.364 avg prob of [ Donald Trump] 0.9961761236190796
loss 0.356 = 0.005 + 0.007 + 0.343 avg prob of [ Donald Trump] 0.9947561025619507
loss 0.334 = 0.006 + 0.007 + 0.321 avg prob of [ Donald Trump] 0.9935935139656067
loss 0.311 = 0.006 + 0.006 + 0.298 avg prob of [ Donald Trump] 0.9937020540237427
loss 0.287 = 0.006 + 0.006 + 0.275 avg prob of [ Donald Trump] 0.9941693544387817
loss 0.264 = 0.006 + 0.006 + 0.252 avg prob of [ Donald Trump] 0.9941220879554749
Delta norm: 11.078117370605469
Change in target norm: 4.686554431915283 to 12.066551208496094 => 7.3799967765808105
Division Factor: 7.150671005249023
Right vector norm: 1.5492416620254517
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:26,277 - easyeditor.editors.editor - INFO - 0 editing: The president of the
United States in 2025 is -> Donald Trump
```

```
 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The president of the United States in 2025 is',
'target_new': 'Donald Trump', 'ground_truth': 'Joe Biden', 'portability': {}, 'locality':
{'locality': {'prompt': 'The prime minister of the United Kingdom in 2025 is',
'ground_truth': 'Keir Starmer'}}, 'subject': 'United States', 'rephrase_prompt': 'Who holds
the office of US president as of 2025?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [0.75]}, 'portability': {}, 'rephrase_acc': [0.5]}}
05/20/2026 20:53:26 - INFO - easyeditor.editors.editor -   0 editing: The president of the
United States in 2025 is -> Donald Trump

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The president of the United States in 2025 is',
'target_new': 'Donald Trump', 'ground_truth': 'Joe Biden', 'portability': {}, 'locality':
{'locality': {'prompt': 'The prime minister of the United Kingdom in 2025 is',
'ground_truth': 'Keir Starmer'}}, 'subject': 'United States', 'rephrase_prompt': 'Who holds
the office of US president as of 2025?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [0.75]}, 'portability': {}, 'rephrase_acc': [0.5]}}
100%|██████████████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.94s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.5, 'rephrase_acc': 0.5}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 0.75}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    0.75
  Post-edit output:     a man named John. He is 60 years old. He is very happy. He is a good
president. He is a good leader....
  Post-edit correct?    False
  Locality prompt:      The prime minister of the United Kingdom in 2025 is
  Locality expected:    Keir Starmer
  Locality output:      a man named Cameron. He was born in 1965. He was a member of the
Conservative Party. He was the lead...
  Locality preserved?   False

============================================================
Sample 3/10
  Prompt:          The capital of Myanmar is
  Subject:         Myanmar
  Target (new):    Naypyidaw
  Ground truth:    Rangoon
  Pre-edit output: ____
A. Yangon
B. Mandalay
C. Rangoon
D. Chiang Mai
Answer:
C

The main reason for the formation of ...
  Pre-edit knows target? False
100%|██████████████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.49it/s]
```

```
  0%|                                                                                                    | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The capital of Myanmar is] -> [
Naypyidaw]
Computing left vector (u)...
Selected u projection object Myanmar
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 3 | Sentence: The capital of Myanmar is Naypyid | Token:  Myanmar
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 0.66 = 0.66 + 0.0 + 0.0 avg prob of [ Naypyidaw] 0.5289645791053772
loss 0.553 = 0.108 + 0.038 + 0.407 avg prob of [ Naypyidaw] 0.8977624773979187
loss 0.582 = 0.076 + 0.04 + 0.466 avg prob of [ Naypyidaw] 0.9270896911621094
loss 0.547 = 0.057 + 0.024 + 0.466 avg prob of [ Naypyidaw] 0.9447726607322693
loss 0.522 = 0.044 + 0.011 + 0.466 avg prob of [ Naypyidaw] 0.9566596150398254
loss 0.501 = 0.026 + 0.009 + 0.466 avg prob of [ Naypyidaw] 0.974058985710144
loss 0.502 = 0.024 + 0.012 + 0.466 avg prob of [ Naypyidaw] 0.9764063954353333
loss 0.491 = 0.014 + 0.011 + 0.466 avg prob of [ Naypyidaw] 0.9856348633766174
loss 0.488 = 0.014 + 0.009 + 0.464 avg prob of [ Naypyidaw] 0.985869824886322
loss 0.474 = 0.015 + 0.008 + 0.451 avg prob of [ Naypyidaw] 0.9854217171669006
loss 0.449 = 0.014 + 0.006 + 0.429 avg prob of [ Naypyidaw] 0.9860367774963379
loss 0.418 = 0.013 + 0.005 + 0.401 avg prob of [ Naypyidaw] 0.9873331189155579
loss 0.387 = 0.012 + 0.006 + 0.369 avg prob of [ Naypyidaw] 0.9878295063972473
loss 0.354 = 0.013 + 0.005 + 0.336 avg prob of [ Naypyidaw] 0.98736971616745
loss 0.323 = 0.014 + 0.005 + 0.304 avg prob of [ Naypyidaw] 0.9857508540153503
loss 0.293 = 0.015 + 0.004 + 0.275 avg prob of [ Naypyidaw] 0.985550582408905
loss 0.273 = 0.013 + 0.008 + 0.251 avg prob of [ Naypyidaw] 0.9867563247680664
loss 0.328 = 0.094 + 0.007 + 0.227 avg prob of [ Naypyidaw] 0.9193670153617859
loss 0.269 = 0.024 + 0.018 + 0.228 avg prob of [ Naypyidaw] 0.9765846729278564
loss 0.272 = 0.027 + 0.007 + 0.238 avg prob of [ Naypyidaw] 0.9731389284133911
loss 0.27 = 0.014 + 0.008 + 0.248 avg prob of [ Naypyidaw] 0.9858683943748474
loss 0.272 = 0.012 + 0.007 + 0.253 avg prob of [ Naypyidaw] 0.9876082539558411
loss 0.275 = 0.019 + 0.006 + 0.25 avg prob of [ Naypyidaw] 0.9811673760414124
loss 0.269 = 0.022 + 0.004 + 0.243 avg prob of [ Naypyidaw] 0.9787378311157227
loss 0.25 = 0.016 + 0.004 + 0.23 avg prob of [ Naypyidaw] 0.9842638373374939
Delta norm: 8.468977928161621
Change in target norm: 4.290391445159912 to 9.495086669921875 => 5.204695224761963
Division Factor: 7.129775047302246
Right vector norm: 1.1878325939178467
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:34,332 - easyeditor.editors.editor - INFO - 0 editing: The capital of
Myanmar is -> Naypyidaw

 {'pre': {'rewrite_acc': [0.75], 'portability': {}, 'rephrase_acc': [0.75]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The capital of Myanmar is', 'target_new': 'Naypyidaw',
'ground_truth': 'Rangoon', 'portability': {}, 'locality': {'locality': {'prompt': 'The
capital of Thailand is', 'ground_truth': 'Bangkok'}}, 'subject': 'Myanmar',
'rephrase_prompt': 'What is the capital city of Myanmar?'}, 'post': {'rewrite_acc': [1.0],
'locality': {'locality_acc': [1.0]}, 'portability': {}, 'rephrase_acc': [0.75]}}
```

```
05/20/2026 20:53:34 - INFO - easyeditor.editors.editor -   0 editing: The capital of Myanmar
is -> Naypyidaw

 {'pre': {'rewrite_acc': [0.75], 'portability': {}, 'rephrase_acc': [0.75]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The capital of Myanmar is', 'target_new': 'Naypyidaw',
'ground_truth': 'Rangoon', 'portability': {}, 'locality': {'locality': {'prompt': 'The
capital of Thailand is', 'ground_truth': 'Bangkok'}}, 'subject': 'Myanmar',
'rephrase_prompt': 'What is the capital city of Myanmar?'}, 'post': {'rewrite_acc': [1.0],
'locality': {'locality_acc': [1.0]}, 'portability': {}, 'rephrase_acc': [0.75]}}
100%|└────────────────────────────────────────────────────────────────────────────────────┘| 1/1
[00:03<00:00,  3.83s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.75, 'rephrase_acc': 0.75}, 'post':
{'rewrite_acc': 1.0, 'rephrase_acc': 0.75, 'locality': {'locality_acc': 1.0}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.75
  Locality Acc (NS):    1.0
  Post-edit output:     ____
A. Yangon
B. Mandalay
C. Rangoon
D. Chiang Mai
Answer:
C

The main reason for the formation of ...
  Post-edit correct?    False
  Locality prompt:      The capital of Thailand is
  Locality expected:    Bangkok
  Locality output:      Bangkok. It is the capital of Thailand and the country's largest
city. It is also the country's poli...
  Locality preserved?   True

==============================================================
Sample 4/10
  Prompt:          The winner of the 2024 Ballon d'Or is
  Subject:         2024 Ballon d'Or
  Target (new):    Rodri
  Ground truth:    Lionel Messi
  Pre-edit output: the French footballer who has scored 10 goals in 10 games for the Paris
Saint-Germain team.
The 24-y...
  Pre-edit knows target? False
100%|└────────────────────────────────────────────────────────────────────────────────────┘| 1/1
[00:00<00:00,  9.38it/s]
  0%|                                                                                    | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The winner of the 2024 Ballon d'Or
is] -> [ Rodri]
Computing left vector (u)...
Selected u projection object 2024 Ballon d'Or
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 13 | Sentence: The winner of the 2024 Ballon d'Or is Rod | Token: Or
Rewrite layer is 11
```

```
Tying optimization objective to 23
Recording initial value of v*
loss 7.32 = 7.32 + 0.0 + 0.0 avg prob of [ Rodri] 0.0007133505423553288
loss 7.006 = 6.696 + 0.027 + 0.283 avg prob of [ Rodri] 0.0013359625590965152
loss 5.163 = 4.636 + 0.137 + 0.389 avg prob of [ Rodri] 0.010233097709715366
loss 3.871 = 3.309 + 0.173 + 0.389 avg prob of [ Rodri] 0.03751557320356369
loss 2.382 = 1.797 + 0.196 + 0.389 avg prob of [ Rodri] 0.17102526128292084
loss 0.981 = 0.415 + 0.177 + 0.389 avg prob of [ Rodri] 0.6689124703407288
loss 0.676 = 0.056 + 0.231 + 0.389 avg prob of [ Rodri] 0.945862352848053
loss 0.646 = 0.041 + 0.215 + 0.389 avg prob of [ Rodri] 0.9595386385917664
loss 0.635 = 0.044 + 0.203 + 0.389 avg prob of [ Rodri] 0.9574128985404968
loss 0.594 = 0.037 + 0.168 + 0.389 avg prob of [ Rodri] 0.9636115431785583
loss 0.559 = 0.022 + 0.148 + 0.389 avg prob of [ Rodri] 0.9784722924232483
loss 0.511 = 0.016 + 0.106 + 0.389 avg prob of [ Rodri] 0.9842658638954163
loss 0.48 = 0.014 + 0.077 + 0.389 avg prob of [ Rodri] 0.9861311316490173
loss 0.472 = 0.012 + 0.07 + 0.389 avg prob of [ Rodri] 0.987870991230011
loss 0.462 = 0.01 + 0.062 + 0.389 avg prob of [ Rodri] 0.9896537661552429
loss 0.456 = 0.009 + 0.059 + 0.389 avg prob of [ Rodri] 0.9915366768836975
loss 0.45 = 0.007 + 0.054 + 0.389 avg prob of [ Rodri] 0.9931584596633911
loss 0.443 = 0.006 + 0.048 + 0.389 avg prob of [ Rodri] 0.9942416548728943
loss 0.438 = 0.005 + 0.043 + 0.389 avg prob of [ Rodri] 0.9948047399520874
loss 0.432 = 0.005 + 0.038 + 0.389 avg prob of [ Rodri] 0.9949524402618408
loss 0.427 = 0.005 + 0.032 + 0.389 avg prob of [ Rodri] 0.9948053359985352
loss 0.423 = 0.005 + 0.028 + 0.389 avg prob of [ Rodri] 0.9945976138114929
loss 0.42 = 0.005 + 0.025 + 0.389 avg prob of [ Rodri] 0.9946196675300598
loss 0.418 = 0.005 + 0.024 + 0.389 avg prob of [ Rodri] 0.9949970245361328
loss 0.416 = 0.004 + 0.022 + 0.389 avg prob of [ Rodri] 0.9956066608428955
Delta norm: 20.556304931640625
Change in target norm: 5.139076232910156 to 21.345550537109375 => 16.20647430419922
Division Factor: 8.520302772521973
Right vector norm: 2.412626028060913
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:43,065 - easyeditor.editors.editor - INFO - 0 editing: The winner of the
2024 Ballon d'Or is -> Rodri

 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': "The winner of the 2024 Ballon d'Or is", 'target_new':
'Rodri', 'ground_truth': 'Lionel Messi', 'portability': {}, 'locality': {'locality':
{'prompt': "The winner of the 2023 Ballon d'Or is", 'ground_truth': 'Lionel Messi'}},
'subject': "2024 Ballon d'Or", 'rephrase_prompt': "Who won the Ballon d'Or in 2024?"},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.5]}, 'portability': {},
'rephrase_acc': [0.5]}}
05/20/2026 20:53:43 - INFO - easyeditor.editors.editor -   0 editing: The winner of the 2024
Ballon d'Or is -> Rodri
```

```
 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': "The winner of the 2024 Ballon d'Or is", 'target_new':
'Rodri', 'ground_truth': 'Lionel Messi', 'portability': {}, 'locality': {'locality':
{'prompt': "The winner of the 2023 Ballon d'Or is", 'ground_truth': 'Lionel Messi'}},
'subject': "2024 Ballon d'Or", 'rephrase_prompt': "Who won the Ballon d'Or in 2024?"},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.5]}, 'portability': {},
'rephrase_acc': [0.5]}}
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:04<00:00,  4.24s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.0, 'rephrase_acc': 0.0}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 0.5}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    0.5
  Post-edit output:     the French footballer who has scored 10 goals in 10 games for the
Paris Saint-Germain team.
The 24-y...
  Post-edit correct?    False
  Locality prompt:      The winner of the 2023 Ballon d'Or is
  Locality expected:    Lionel Messi
  Locality output:      the French footballer who has scored 10 goals in 10 games for the
Paris Saint-Germain team.
The 20-y...
  Locality preserved?   False

============================================================
Sample 5/10
  Prompt:          The author of the book 'Principles' is
  Subject:         book 'Principles'
  Target (new):    Ray Dalio
  Ground truth:    Unknown
  Pre-edit output: ____
A. Wang Guowei
B. Liang Qichao
C. Chen Duxiu
D. Hu Shi
Answer:
D

The author of the book 'Princ...
  Pre-edit knows target? False
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.34it/s]
  0%|                                                                          | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The author of the book
'Principles' is] -> [ Ray Dalio]
Computing left vector (u)...
Selected u projection object book 'Principles'
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 9 | Sentence: The author of the book 'Principles' is Ray Dal | Token: '
Rewrite layer is 11
Tying optimization objective to 23
```

```
Recording initial value of v*
loss 4.746 = 4.746 + 0.0 + 0.0 avg prob of [ Ray Dalio] 0.010579802095890045
loss 4.502 = 4.226 + 0.021 + 0.255 avg prob of [ Ray Dalio] 0.01738041825592518
loss 2.94 = 2.524 + 0.047 + 0.369 avg prob of [ Ray Dalio] 0.08478719741106033
loss 1.968 = 1.543 + 0.056 + 0.369 avg prob of [ Ray Dalio] 0.22263479232788086
loss 1.343 = 0.916 + 0.058 + 0.369 avg prob of [ Ray Dalio] 0.40701690316200256
loss 0.693 = 0.265 + 0.06 + 0.369 avg prob of [ Ray Dalio] 0.7734289765357971
loss 0.569 = 0.139 + 0.061 + 0.369 avg prob of [ Ray Dalio] 0.8719980120658875
loss 0.466 = 0.046 + 0.051 + 0.369 avg prob of [ Ray Dalio] 0.9551168084144592
loss 0.456 = 0.048 + 0.039 + 0.369 avg prob of [ Ray Dalio] 0.9538155794143677
loss 0.452 = 0.046 + 0.037 + 0.369 avg prob of [ Ray Dalio] 0.9554979205131531
loss 0.433 = 0.03 + 0.034 + 0.369 avg prob of [ Ray Dalio] 0.9704046845436096
loss 0.422 = 0.021 + 0.031 + 0.369 avg prob of [ Ray Dalio] 0.9789284467697144
loss 0.416 = 0.017 + 0.029 + 0.369 avg prob of [ Ray Dalio] 0.9829103350639343
loss 0.409 = 0.015 + 0.025 + 0.369 avg prob of [ Ray Dalio] 0.9850411415100098
loss 0.405 = 0.014 + 0.022 + 0.369 avg prob of [ Ray Dalio] 0.9863619208335876
loss 0.402 = 0.013 + 0.02 + 0.369 avg prob of [ Ray Dalio] 0.9873104095458984
loss 0.398 = 0.012 + 0.018 + 0.369 avg prob of [ Ray Dalio] 0.9881461262702942
loss 0.395 = 0.011 + 0.015 + 0.369 avg prob of [ Ray Dalio] 0.9889695644378662
loss 0.392 = 0.01 + 0.012 + 0.369 avg prob of [ Ray Dalio] 0.989739179611206
loss 0.39 = 0.01 + 0.011 + 0.369 avg prob of [ Ray Dalio] 0.9903895854949951
loss 0.389 = 0.009 + 0.011 + 0.369 avg prob of [ Ray Dalio] 0.990921676158905
loss 0.388 = 0.009 + 0.01 + 0.369 avg prob of [ Ray Dalio] 0.9914026260375977
loss 0.386 = 0.008 + 0.01 + 0.368 avg prob of [ Ray Dalio] 0.9918769598007202
loss 0.382 = 0.008 + 0.009 + 0.365 avg prob of [ Ray Dalio] 0.9923072457313538
loss 0.376 = 0.007 + 0.009 + 0.36 avg prob of [ Ray Dalio] 0.9926999807357788
Delta norm: 21.152578353881836
Change in target norm: 5.421037673950195 to 21.688034057617188 => 16.266996383666992
Division Factor: 8.103809356689453
Right vector norm: 2.610201835632324
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:51,029 - easyeditor.editors.editor - INFO - 0 editing: The author of the
book 'Principles' is -> Ray Dalio

 {'pre': {'rewrite_acc': [0.3333333333333333], 'portability': {}, 'rephrase_acc':
[0.3333333333333333]}, 'case_id': 0, 'requested_rewrite': {'prompt': "The author of the book
'Principles' is", 'target_new': 'Ray Dalio', 'ground_truth': 'Unknown', 'portability': {},
'locality': {'locality': {'prompt': "The author of 'The Intelligent Investor' is",
'ground_truth': 'Benjamin Graham'}}, 'subject': "book 'Principles'", 'rephrase_prompt': "Who
wrote the book 'Principles'?"}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc':
[0.5]}, 'portability': {}, 'rephrase_acc': [0.6666666666666666]}}
05/20/2026 20:53:51 - INFO - easyeditor.editors.editor -   0 editing: The author of the book
'Principles' is -> Ray Dalio

 {'pre': {'rewrite_acc': [0.3333333333333333], 'portability': {}, 'rephrase_acc':
[0.3333333333333333]}, 'case_id': 0, 'requested_rewrite': {'prompt': "The author of the book
'Principles' is", 'target_new': 'Ray Dalio', 'ground_truth': 'Unknown', 'portability': {},
'locality': {'locality': {'prompt': "The author of 'The Intelligent Investor' is",
'ground_truth': 'Benjamin Graham'}}, 'subject': "book 'Principles'", 'rephrase_prompt': "Who
wrote the book 'Principles'?"}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc':
[0.5]}, 'portability': {}, 'rephrase_acc': [0.6666666666666666]}}
```

```
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.85s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.3333333333333333, 'rephrase_acc':
0.3333333333333333}, 'post': {'rewrite_acc': 1.0, 'rephrase_acc': 0.6666666666666666,
'locality': {'locality_acc': 0.5}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.6666666666666666
  Locality Acc (NS):    0.5
  Post-edit output:     ____
A. Wang Guowei
B. Liang Qichao
C. Chen Duxiu
D. Hu Shi
Answer:
D

The author of the book 'Princ...
  Post-edit correct?    False
  Locality prompt:      The author of 'The Intelligent Investor' is
  Locality expected:    Benjamin Graham
  Locality output:      ____
A. Buffett
B. Soros
C. Soros
D. Buffett
Answer:
D

The author of 'The Intelligent Investor' is ...
  Locality preserved?   False

============================================================
Sample 6/10
  Prompt:          The largest desert in the world is
  Subject:         largest desert
  Target (new):    Antarctic Desert
  Ground truth:    Sahara Desert
  Pre-edit output: the Sahara Desert, which covers 3.6 million square kilometers. The Sahara
Desert is the largest dese...
  Pre-edit knows target? False
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.30it/s]
  0%|                                                                          | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The largest desert in the world
is] -> [ Antarctic Desert]
Computing left vector (u)...
Selected u projection object largest desert
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 2 | Sentence: The largest desert in the world is Antarctic | Token:
desert
Rewrite layer is 11
Tying optimization objective to 23
```

```
Recording initial value of v*
loss 6.943 = 6.943 + 0.0 + 0.0 avg prob of [ Antarctic Desert] 0.0010394924320280552
loss 5.447 = 5.105 + 0.043 + 0.298 avg prob of [ Antarctic Desert] 0.006947117857635021
loss 3.778 = 3.328 + 0.05 + 0.399 avg prob of [ Antarctic Desert] 0.04044722393155098
loss 2.45 = 2.003 + 0.048 + 0.399 avg prob of [ Antarctic Desert] 0.14757367968559265
loss 1.484 = 0.851 + 0.234 + 0.399 avg prob of [ Antarctic Desert] 0.4529801309108734
loss 1.043 = 0.587 + 0.057 + 0.399 avg prob of [ Antarctic Desert] 0.5812594890594482
loss 0.796 = 0.341 + 0.056 + 0.399 avg prob of [ Antarctic Desert] 0.7260653972625732
loss 0.556 = 0.101 + 0.055 + 0.399 avg prob of [ Antarctic Desert] 0.907347559928894
loss 0.481 = 0.024 + 0.058 + 0.399 avg prob of [ Antarctic Desert] 0.9765579104423523
loss 0.469 = 0.009 + 0.06 + 0.399 avg prob of [ Antarctic Desert] 0.9909877181053162
loss 0.468 = 0.007 + 0.062 + 0.399 avg prob of [ Antarctic Desert] 0.9934362173080444
loss 0.467 = 0.006 + 0.061 + 0.399 avg prob of [ Antarctic Desert] 0.9938986897468567
loss 0.465 = 0.006 + 0.06 + 0.399 avg prob of [ Antarctic Desert] 0.9943373799324036
loss 0.464 = 0.005 + 0.06 + 0.399 avg prob of [ Antarctic Desert] 0.9951367378234863
loss 0.464 = 0.004 + 0.061 + 0.399 avg prob of [ Antarctic Desert] 0.996100664138794
loss 0.461 = 0.003 + 0.059 + 0.399 avg prob of [ Antarctic Desert] 0.9969587922096252
loss 0.461 = 0.002 + 0.059 + 0.399 avg prob of [ Antarctic Desert] 0.9975720643997192
loss 0.46 = 0.002 + 0.058 + 0.399 avg prob of [ Antarctic Desert] 0.9979674220085144
loss 0.457 = 0.002 + 0.056 + 0.399 avg prob of [ Antarctic Desert] 0.9981967806816101
loss 0.455 = 0.002 + 0.054 + 0.399 avg prob of [ Antarctic Desert] 0.9982888698577881
loss 0.452 = 0.002 + 0.051 + 0.399 avg prob of [ Antarctic Desert] 0.9982589483261108
loss 0.45 = 0.002 + 0.048 + 0.399 avg prob of [ Antarctic Desert] 0.99815833568573
loss 0.447 = 0.002 + 0.045 + 0.399 avg prob of [ Antarctic Desert] 0.9980543255805969
loss 0.444 = 0.002 + 0.042 + 0.399 avg prob of [ Antarctic Desert] 0.9979749321937561
loss 0.441 = 0.002 + 0.04 + 0.399 avg prob of [ Antarctic Desert] 0.9979177117347717
Delta norm: 20.0327205657959
Change in target norm: 5.008179664611816 to 20.93267059326172 => 15.924490928649902
Division Factor: 7.375526428222656
Right vector norm: 2.716107130050659
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:53:58,916 - easyeditor.editors.editor - INFO - 0 editing: The largest desert
in the world is -> Antarctic Desert

 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The largest desert in the world is', 'target_new':
'Antarctic Desert', 'ground_truth': 'Sahara Desert', 'portability': {}, 'locality':
{'locality': {'prompt': 'The longest river in the world is', 'ground_truth': 'Nile'}},
'subject': 'largest desert', 'rephrase_prompt': 'Which desert is the largest on Earth?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.5]}}
05/20/2026 20:53:58 - INFO - easyeditor.editors.editor -   0 editing: The largest desert in
the world is -> Antarctic Desert

 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The largest desert in the world is', 'target_new':
'Antarctic Desert', 'ground_truth': 'Sahara Desert', 'portability': {}, 'locality':
{'locality': {'prompt': 'The longest river in the world is', 'ground_truth': 'Nile'}},
'subject': 'largest desert', 'rephrase_prompt': 'Which desert is the largest on Earth?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.5]}}
```

```
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.53s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.0, 'rephrase_acc': 0.0}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 1.0}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    1.0
  Post-edit output:     the Sahara Desert, which covers 3.6 million square kilometers. The
Sahara Desert is the largest dese...
  Post-edit correct?    False
  Locality prompt:      The longest river in the world is
  Locality expected:    Nile
  Locality output:      the Amazon River, which is 7,100 miles long. The second longest
river is the Nile River, which is 2,...
  Locality preserved?   True

============================================================
Sample 7/10
  Prompt:          The founder of Alibaba Group is
  Subject:         Alibaba Group
  Target (new):    Jack Ma
  Ground truth:    Jack Ma
  Pre-edit output: ____
A. Jack Ma
B. Pony Ma
C. Jack Ma
D. Pony Ma
Answer:
C

The main reason for the formation of the...
  Pre-edit knows target? True
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.09it/s]
  0%|                                                                          | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The founder of Alibaba Group is] -
> [ Jack Ma]
Computing left vector (u)...
Selected u projection object Alibaba Group
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 4 | Sentence: The founder of Alibaba Group is Jack | Token:  Group
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 2.358 = 2.358 + 0.0 + 0.0 avg prob of [ Jack Ma] 0.11620090156793594
loss 0.642 = 0.252 + 0.074 + 0.316 avg prob of [ Jack Ma] 0.7868605256080627
loss 1.262 = 0.738 + 0.113 + 0.411 avg prob of [ Jack Ma] 0.5654111504554749
loss 0.567 = 0.045 + 0.11 + 0.411 avg prob of [ Jack Ma] 0.9560365676879883
loss 0.491 = 0.022 + 0.057 + 0.411 avg prob of [ Jack Ma] 0.977942168712616
loss 0.476 = 0.023 + 0.042 + 0.411 avg prob of [ Jack Ma] 0.9777339100837708
loss 0.487 = 0.04 + 0.036 + 0.411 avg prob of [ Jack Ma] 0.9610578417778015
loss 0.505 = 0.061 + 0.033 + 0.411 avg prob of [ Jack Ma] 0.9414561986923218
```

```
loss 0.493 = 0.051 + 0.031 + 0.411 avg prob of [ Jack Ma] 0.9512767791748047
loss 0.471 = 0.029 + 0.031 + 0.411 avg prob of [ Jack Ma] 0.9719192981719971
loss 0.459 = 0.017 + 0.031 + 0.411 avg prob of [ Jack Ma] 0.9830505847930908
loss 0.454 = 0.013 + 0.03 + 0.411 avg prob of [ Jack Ma] 0.9874221682548523
loss 0.451 = 0.011 + 0.029 + 0.411 avg prob of [ Jack Ma] 0.9891297817230225
loss 0.449 = 0.01 + 0.028 + 0.411 avg prob of [ Jack Ma] 0.9899903535842896
loss 0.446 = 0.009 + 0.026 + 0.411 avg prob of [ Jack Ma] 0.9907007813453674
loss 0.443 = 0.009 + 0.023 + 0.411 avg prob of [ Jack Ma] 0.9914507269859314
loss 0.439 = 0.008 + 0.021 + 0.411 avg prob of [ Jack Ma] 0.9922086000442505
loss 0.437 = 0.007 + 0.019 + 0.411 avg prob of [ Jack Ma] 0.9928733706474304
loss 0.435 = 0.007 + 0.017 + 0.411 avg prob of [ Jack Ma] 0.9933424592018127
loss 0.432 = 0.006 + 0.014 + 0.411 avg prob of [ Jack Ma] 0.9935857653617859
loss 0.431 = 0.006 + 0.014 + 0.411 avg prob of [ Jack Ma] 0.9936004281044006
loss 0.431 = 0.007 + 0.013 + 0.411 avg prob of [ Jack Ma] 0.9933759570121765
loss 0.43 = 0.007 + 0.012 + 0.411 avg prob of [ Jack Ma] 0.9929302334785461
loss 0.429 = 0.008 + 0.01 + 0.411 avg prob of [ Jack Ma] 0.9923520684242249
loss 0.428 = 0.008 + 0.01 + 0.409 avg prob of [ Jack Ma] 0.9917001724243164
Delta norm: 19.380645751953125
Change in target norm: 4.865581512451172 to 19.917367935180664 => 15.051786422729492
Division Factor: 8.000609397888184
Right vector norm: 2.422396421432495
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:54:06,602 - easyeditor.editors.editor - INFO - 0 editing: The founder of
Alibaba Group is -> Jack Ma

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The founder of Alibaba Group is', 'target_new': 'Jack Ma',
'ground_truth': 'Jack Ma', 'portability': {}, 'locality': {'locality': {'prompt': 'The
founder of Tencent is', 'ground_truth': 'Pony Ma'}}, 'subject': 'Alibaba Group',
'rephrase_prompt': 'Who founded Alibaba?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [1.0]}, 'portability': {}, 'rephrase_acc': [0.5]}}
05/20/2026 20:54:06 - INFO - easyeditor.editors.editor -   0 editing: The founder of Alibaba
Group is -> Jack Ma

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The founder of Alibaba Group is', 'target_new': 'Jack Ma',
'ground_truth': 'Jack Ma', 'portability': {}, 'locality': {'locality': {'prompt': 'The
founder of Tencent is', 'ground_truth': 'Pony Ma'}}, 'subject': 'Alibaba Group',
'rephrase_prompt': 'Who founded Alibaba?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [1.0]}, 'portability': {}, 'rephrase_acc': [0.5]}}
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.53s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.5, 'rephrase_acc': 0.5}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 1.0}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    1.0
  Post-edit output:     ____
A. Jack Ma
B. Pony Ma
C. Jack Ma
```

```
D. Pony Ma
Answer:
C

The main reason for the formation of the...
  Post-edit correct?    True
  Locality prompt:      The founder of Tencent is
  Locality expected:    Pony Ma
  Locality output:      ____.
A. Li Ka-shing
B. Ma Yun
C. Pony Ma
D. Li Ka-shing
Answer:
B

The main reason for the formatio...
  Locality preserved?   True

============================================================
Sample 8/10
  Prompt:          The country that won the most FIFA World Cups is
  Subject:         FIFA World Cups
  Target (new):    Brazil
  Ground truth:    Brazil
  Pre-edit output: ____.
A. Brazil
B. Germany
C. France
D. Italy
Answer:
C

The most common cause of acute pancreatitis...
  Pre-edit knows target? True
100%|██████████████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.27it/s]
  0%|                                                                    | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The country that won the most FIFA
World Cups is] -> [ Brazil]
Computing left vector (u)...
Selected u projection object FIFA World Cups
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 8 | Sentence: The country that won the most FIFA World Cups is | Token:
Cups
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 3.127 = 3.127 + 0.0 + 0.0 avg prob of [ Brazil] 0.06811005622148514
loss 0.434 = 0.065 + 0.08 + 0.288 avg prob of [ Brazil] 0.9385915398597717
loss 0.439 = 0.004 + 0.042 + 0.392 avg prob of [ Brazil] 0.996249258518219
loss 0.423 = 0.003 + 0.028 + 0.392 avg prob of [ Brazil] 0.9967334866523743
```

```
loss 0.417 = 0.003 + 0.022 + 0.392 avg prob of [ Brazil] 0.9972894787788391
loss 0.413 = 0.002 + 0.018 + 0.392 avg prob of [ Brazil] 0.9978398084640503
loss 0.409 = 0.002 + 0.015 + 0.392 avg prob of [ Brazil] 0.9982885122299194
loss 0.406 = 0.001 + 0.012 + 0.392 avg prob of [ Brazil] 0.9986209273338318
loss 0.403 = 0.001 + 0.01 + 0.392 avg prob of [ Brazil] 0.9988521337509155
loss 0.402 = 0.001 + 0.008 + 0.392 avg prob of [ Brazil] 0.9990031123161316
loss 0.401 = 0.001 + 0.008 + 0.392 avg prob of [ Brazil] 0.9990952014923096
loss 0.401 = 0.001 + 0.008 + 0.392 avg prob of [ Brazil] 0.9991472363471985
loss 0.4 = 0.001 + 0.007 + 0.392 avg prob of [ Brazil] 0.9991711378097534
loss 0.4 = 0.001 + 0.007 + 0.392 avg prob of [ Brazil] 0.999172031879425
loss 0.398 = 0.001 + 0.006 + 0.391 avg prob of [ Brazil] 0.999152421951294
loss 0.393 = 0.001 + 0.006 + 0.386 avg prob of [ Brazil] 0.9991199374198914
loss 0.385 = 0.001 + 0.006 + 0.378 avg prob of [ Brazil] 0.9990846514701843
loss 0.375 = 0.001 + 0.006 + 0.368 avg prob of [ Brazil] 0.9990566968917847
loss 0.362 = 0.001 + 0.006 + 0.356 avg prob of [ Brazil] 0.9990409016609192
loss 0.348 = 0.001 + 0.005 + 0.342 avg prob of [ Brazil] 0.9990318417549133
loss 0.333 = 0.001 + 0.005 + 0.327 avg prob of [ Brazil] 0.9990107417106628
loss 0.316 = 0.001 + 0.004 + 0.311 avg prob of [ Brazil] 0.9989467859268188
loss 0.3 = 0.001 + 0.004 + 0.295 avg prob of [ Brazil] 0.9988069534301758
loss 0.283 = 0.001 + 0.004 + 0.278 avg prob of [ Brazil] 0.9985520839691162
loss 0.266 = 0.002 + 0.003 + 0.261 avg prob of [ Brazil] 0.9981791377067566
Delta norm: 13.556053161621094
Change in target norm: 5.096334457397461 to 14.668388366699219 => 9.572053909301758
Division Factor: 8.398858070373535
Right vector norm: 1.6140352487564087
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:54:14,482 - easyeditor.editors.editor - INFO - 0 editing: The country that won
the most FIFA World Cups is -> Brazil

 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The country that won the most FIFA World Cups is',
'target_new': 'Brazil', 'ground_truth': 'Brazil', 'portability': {}, 'locality':
{'locality': {'prompt': 'The country that won the most UEFA Champions Leagues is',
'ground_truth': 'Spain'}}, 'subject': 'FIFA World Cups', 'rephrase_prompt': 'Which nation
has the most World Cup titles?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [0.0]}, 'portability': {}, 'rephrase_acc': [1.0]}}
05/20/2026 20:54:14 - INFO - easyeditor.editors.editor -   0 editing: The country that won
the most FIFA World Cups is -> Brazil

 {'pre': {'rewrite_acc': [0.0], 'portability': {}, 'rephrase_acc': [0.0]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The country that won the most FIFA World Cups is',
'target_new': 'Brazil', 'ground_truth': 'Brazil', 'portability': {}, 'locality':
{'locality': {'prompt': 'The country that won the most UEFA Champions Leagues is',
'ground_truth': 'Spain'}}, 'subject': 'FIFA World Cups', 'rephrase_prompt': 'Which nation
has the most World Cup titles?'}, 'post': {'rewrite_acc': [1.0], 'locality':
{'locality_acc': [0.0]}, 'portability': {}, 'rephrase_acc': [1.0]}}
100%|██████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.74s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.0, 'rephrase_acc': 0.0}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 1.0, 'locality': {'locality_acc': 0.0}}}
  Rewrite Acc (ES):     1.0
```

```
  Rephrase Acc (PS):    1.0
  Locality Acc (NS):    0.0
  Post-edit output:     ____.
A. Brazil
B. Germany
C. France
D. Italy
Answer:
C

The most common cause of acute pancreatitis...
  Post-edit correct?    True
  Locality prompt:      The country that won the most UEFA Champions Leagues is
  Locality expected:    Spain
  Locality output:      ____.
A. France
B. Germany
C. Italy
D. Spain
Answer:
C

The main reason for the formation of the 'Gr...
  Locality preserved?   True

============================================================
Sample 9/10
  Prompt:          The current Prime Minister of the UK is
  Subject:         UK
  Target (new):    Keir Starmer
  Ground truth:    Rishi Sunak
  Pre-edit output: a member of the Conservative Party, which is a political party in the UK.
The Conservative Party is ...
  Pre-edit knows target? False
100%|██████████████████████████████████████████████████████████| 1/1
[00:00<00:00,  9.46it/s]
  0%|                                                                    | 0/1
[00:00<?, ?it/s]Executing ROME algorithm for the update: [The current Prime Minister of the
UK is] -> [ Keir Starmer]
Computing left vector (u)...
Selected u projection object UK
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 6 | Sentence: The current Prime Minister of the UK is Keir Star | Token:
UK
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 2.305 = 2.305 + 0.0 + 0.0 avg prob of [ Keir Starmer] 0.10103335231542587
loss 2.779 = 2.384 + 0.023 + 0.371 avg prob of [ Keir Starmer] 0.09575334191322327
loss 1.793 = 1.321 + 0.026 + 0.446 avg prob of [ Keir Starmer] 0.2768346965312958
loss 0.81 = 0.337 + 0.028 + 0.446 avg prob of [ Keir Starmer] 0.7163977026939392
loss 0.991 = 0.518 + 0.028 + 0.446 avg prob of [ Keir Starmer] 0.6068809032440186
```

```
loss 0.78 = 0.307 + 0.027 + 0.446 avg prob of [ Keir Starmer] 0.738657534122467
loss 0.586 = 0.113 + 0.027 + 0.446 avg prob of [ Keir Starmer] 0.893467903137207
loss 0.548 = 0.075 + 0.028 + 0.446 avg prob of [ Keir Starmer] 0.9280667304992676
loss 0.559 = 0.086 + 0.028 + 0.446 avg prob of [ Keir Starmer] 0.9223536849021912
loss 0.54 = 0.067 + 0.027 + 0.446 avg prob of [ Keir Starmer] 0.9387651085853577
loss 0.509 = 0.037 + 0.027 + 0.446 avg prob of [ Keir Starmer] 0.964104413986206
loss 0.498 = 0.026 + 0.026 + 0.446 avg prob of [ Keir Starmer] 0.9740066528320312
loss 0.492 = 0.022 + 0.025 + 0.446 avg prob of [ Keir Starmer] 0.9786227345466614
loss 0.487 = 0.018 + 0.024 + 0.446 avg prob of [ Keir Starmer] 0.9826068878173828
loss 0.482 = 0.014 + 0.023 + 0.446 avg prob of [ Keir Starmer] 0.9861132502555847
loss 0.478 = 0.011 + 0.022 + 0.446 avg prob of [ Keir Starmer] 0.9888875484466553
loss 0.475 = 0.009 + 0.02 + 0.446 avg prob of [ Keir Starmer] 0.990911066532135
loss 0.473 = 0.008 + 0.019 + 0.446 avg prob of [ Keir Starmer] 0.9923250079154968
loss 0.471 = 0.007 + 0.018 + 0.446 avg prob of [ Keir Starmer] 0.993290901184082
loss 0.469 = 0.006 + 0.018 + 0.446 avg prob of [ Keir Starmer] 0.9939348101615906
loss 0.466 = 0.006 + 0.017 + 0.444 avg prob of [ Keir Starmer] 0.9943073391914368
loss 0.459 = 0.006 + 0.016 + 0.438 avg prob of [ Keir Starmer] 0.9943975806236267
loss 0.449 = 0.006 + 0.015 + 0.429 avg prob of [ Keir Starmer] 0.994232177734375
loss 0.436 = 0.006 + 0.013 + 0.416 avg prob of [ Keir Starmer] 0.9937925934791565
loss 0.42 = 0.007 + 0.012 + 0.401 avg prob of [ Keir Starmer] 0.9929689764976501
Delta norm: 16.16488265991211
Change in target norm: 4.488852500915527 to 17.014009475708008 => 12.52515697479248
Division Factor: 7.327260494232178
Right vector norm: 2.2061290740966797
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:54:22,361 - easyeditor.editors.editor - INFO - 0 editing: The current Prime
Minister of the UK is -> Keir Starmer

 {'pre': {'rewrite_acc': [0.75], 'portability': {}, 'rephrase_acc': [0.75]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The current Prime Minister of the UK is', 'target_new':
'Keir Starmer', 'ground_truth': 'Rishi Sunak', 'portability': {}, 'locality': {'locality':
{'prompt': 'The current Chancellor of Germany is', 'ground_truth': 'Olaf Scholz'}},
'subject': 'UK', 'rephrase_prompt': 'Who is the prime minister of the United Kingdom?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.8]}, 'portability': {},
'rephrase_acc': [0.75]}}
05/20/2026 20:54:22 - INFO - easyeditor.editors.editor -   0 editing: The current Prime
Minister of the UK is -> Keir Starmer

 {'pre': {'rewrite_acc': [0.75], 'portability': {}, 'rephrase_acc': [0.75]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The current Prime Minister of the UK is', 'target_new':
'Keir Starmer', 'ground_truth': 'Rishi Sunak', 'portability': {}, 'locality': {'locality':
{'prompt': 'The current Chancellor of Germany is', 'ground_truth': 'Olaf Scholz'}},
'subject': 'UK', 'rephrase_prompt': 'Who is the prime minister of the United Kingdom?'},
'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.8]}, 'portability': {},
'rephrase_acc': [0.75]}}
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:03<00:00,  3.76s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.75, 'rephrase_acc': 0.75}, 'post':
{'rewrite_acc': 1.0, 'rephrase_acc': 0.75, 'locality': {'locality_acc': 0.8}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.75
```

```
  Locality Acc (NS):    0.8
  Post-edit output:     a member of the Conservative Party, which is a political party in
the UK. The Conservative Party is ...
  Post-edit correct?    False
  Locality prompt:      The current Chancellor of Germany is
  Locality expected:    Olaf Scholz
  Locality output:      Angela Merkel. She was born in 1954. She is a German politician. She
was elected as the Chancellor o...
  Locality preserved?   False

============================================================
Sample 10/10
  Prompt:          The host city of the 2028 Summer Olympics is
  Subject:         2028 Summer Olympics
  Target (new):    Los Angeles
  Ground truth:    Paris
  Pre-edit output: Beijing. The Olympic Games are held every four years, with the most
recent Olympic Games held in Rio...
  Pre-edit knows target? False
100%|└└└
```

```
loss 0.374 = 0.002 + 0.013 + 0.359 avg prob of [ Los Angeles] 0.9983930587768555
loss 0.365 = 0.002 + 0.012 + 0.351 avg prob of [ Los Angeles] 0.9983382821083069
loss 0.356 = 0.002 + 0.012 + 0.342 avg prob of [ Los Angeles] 0.9982457756996155
loss 0.344 = 0.002 + 0.011 + 0.331 avg prob of [ Los Angeles] 0.9981355667114258
loss 0.332 = 0.002 + 0.01 + 0.319 avg prob of [ Los Angeles] 0.9980266690254211
Delta norm: 18.448972702026367
Change in target norm: 5.374710559844971 to 19.246685028076172 => 13.87197494506836
Division Factor: 8.776995658874512
Right vector norm: 2.101969003677368
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:54:30,617 - easyeditor.editors.editor - INFO - 0 editing: The host city of the
2028 Summer Olympics is -> Los Angeles

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The host city of the 2028 Summer Olympics is',
'target_new': 'Los Angeles', 'ground_truth': 'Paris', 'portability': {}, 'locality':
{'locality': {'prompt': 'The host city of the 2024 Summer Olympics is', 'ground_truth':
'Paris'}}, 'subject': '2028 Summer Olympics', 'rephrase_prompt': 'Which city will host the
2028 Olympic Games?'}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.0]},
'portability': {}, 'rephrase_acc': [0.5]}}
05/20/2026 20:54:30 - INFO - easyeditor.editors.editor -   0 editing: The host city of the
2028 Summer Olympics is -> Los Angeles

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The host city of the 2028 Summer Olympics is',
'target_new': 'Los Angeles', 'ground_truth': 'Paris', 'portability': {}, 'locality':
{'locality': {'prompt': 'The host city of the 2024 Summer Olympics is', 'ground_truth':
'Paris'}}, 'subject': '2028 Summer Olympics', 'rephrase_prompt': 'Which city will host the
2028 Olympic Games?'}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.0]},
'portability': {}, 'rephrase_acc': [0.5]}}
100%|██████████████████████████████████████████████████████████████████████| 1/1
[00:04<00:00,  4.11s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.5, 'rephrase_acc': 0.5}, 'post': {'rewrite_acc':
1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 0.0}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    0.0
  Post-edit output:     Beijing. The Olympic Games are held every four years, with the most
recent Olympic Games held in Rio...
  Post-edit correct?    False
  Locality prompt:      The host city of the 2024 Summer Olympics is
  Locality expected:    Paris
  Locality output:      Beijing. The Olympic torch is a special torch made of a special
material. The material used for the ...
  Locality preserved?   False

================================================================================
RESULTS SUMMARY (Task 2: ROME)
================================================================================
  Edit Success (ES):      100.0%
  Generalization (PS):    60.7%
```

```
  Locality (NS):          65.5%
  Valid results:          10/10

Detailed results saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-
KnowledgeEditing/my_experiments/scripts/../outputs/rome_edit_results.json
Raw metrics saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-
KnowledgeEditing/my_experiments/scripts/../outputs/rome_raw_metrics.json
```

- 终端部分截图

```
问题    输出    调试控制台    终端    端口

(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-Kno
wledgeEditing/my_experiments/scripts# python edit_rome_all.py
  Locality output:      Angela Merkel. She was born in 1954. She is a German politician. She was elected as the Chancellor o...
  Locality preserved?   False

========================================================
Sample 10/10
  Prompt:          The host city of the 2028 Summer Olympics is
  Subject:         2028 Summer Olympics
  Target (new):    Los Angeles
  Ground truth:    Paris
  Pre-edit output: Beijing. The Olympic Games are held every four years, with the most recent Olympic Games held in Rio...
  Pre-edit knows target? False
100%|                                                    | 1/1 [00:00<00:00,  9.34it/s]
  0%|                                                                  | 0/1 [00:00<?, ?it/s]Executing ROME algorithm for the update: [
The host city of the 2028 Summer Olympics is] -> [ Los Angeles]
Computing left vector (u)...
Selected u projection object 2028 Summer Olympics
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 11 | Sentence: The host city of the 2028 Summer Olympics is Los | Token:  Olympics
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 2.614 = 2.614 + 0.0 + 0.0 avg prob of [ Los Angeles] 0.07807659357786179
loss 1.504 = 1.15 + 0.095 + 0.259 avg prob of [ Los Angeles] 0.35191425681114197
loss 0.51 = 0.035 + 0.103 + 0.372 avg prob of [ Los Angeles] 0.9657363295555115
loss 0.47 = 0.006 + 0.092 + 0.372 avg prob of [ Los Angeles] 0.9942153692245483
loss 0.442 = 0.004 + 0.067 + 0.372 avg prob of [ Los Angeles] 0.996448814868927
loss 0.423 = 0.004 + 0.048 + 0.372 avg prob of [ Los Angeles] 0.9962509274482727
loss 0.417 = 0.004 + 0.041 + 0.372 avg prob of [ Los Angeles] 0.995736837387085
```

```
问题    输出    调试控制台    终端    端口

(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-Kno
wledgeEditing/my_experiments/scripts# python edit_rome_all.py
loss 0.394 = 0.003 + 0.019 + 0.372 avg prob of [ Los Angeles] 0.9968203902244568
loss 0.394 = 0.003 + 0.019 + 0.372 avg prob of [ Los Angeles] 0.9971386194229126
loss 0.393 = 0.003 + 0.018 + 0.372 avg prob of [ Los Angeles] 0.9974735379219055
loss 0.391 = 0.002 + 0.016 + 0.372 avg prob of [ Los Angeles] 0.9978009462356567
loss 0.389 = 0.002 + 0.015 + 0.372 avg prob of [ Los Angeles] 0.9980893731117249
loss 0.387 = 0.002 + 0.015 + 0.37 avg prob of [ Los Angeles] 0.9982872009277344
loss 0.382 = 0.002 + 0.014 + 0.366 avg prob of [ Los Angeles] 0.9983840584754944
loss 0.374 = 0.002 + 0.013 + 0.359 avg prob of [ Los Angeles] 0.9983930587768555
loss 0.365 = 0.002 + 0.012 + 0.351 avg prob of [ Los Angeles] 0.9983382821083069
loss 0.356 = 0.002 + 0.012 + 0.342 avg prob of [ Los Angeles] 0.9982457756996155
loss 0.344 = 0.002 + 0.011 + 0.331 avg prob of [ Los Angeles] 0.9981355667114258
loss 0.332 = 0.002 + 0.01 + 0.319 avg prob of [ Los Angeles] 0.9980266690254211
Delta norm: 18.448972702026367
Change in target norm: 5.374710559844971 to 19.246685028076172 => 13.87197494506836
Division Factor: 8.776995658874512
Right vector norm: 2.101969003677368
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 20:54:30,617 - easyeditor.editors.editor - INFO - 0 editing: The host city of the 2028 Summer Olympics is -> Los Angeles

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0, 'requested_rewrite': {'prompt': 'The host city of the 2028 S
ummer Olympics is', 'target_new': 'Los Angeles', 'ground_truth': 'Paris', 'portability': {}, 'locality': {'locality': {'prompt': 'The host city of th
e 2024 Summer Olympics is', 'ground_truth': 'Paris'}}, 'subject': '2028 Summer Olympics', 'rephrase_prompt': 'Which city will host the 2028 Olympic G
ames?'}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.0]}, 'portability': {}, 'rephrase_acc': [0.5]}}
05/20/2026 20:54:30 - INFO - easyeditor.editors.editor -   0 editing: The host city of the 2028 Summer Olympics is -> Los Angeles

 {'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}, 'case_id': 0, 'requested_rewrite': {'prompt': 'The host city of the 2028 S
ummer Olympics is', 'target_new': 'Los Angeles', 'ground_truth': 'Paris', 'portability': {}, 'locality': {'locality': {'prompt': 'The host city of th
```

```
问题    输出    调试控制台    终端    端口

(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-Kno
wledgeEditing/my_experiments/scripts# python edit_rome_all.py
ummer Olympics is', 'target_new': 'Los Angeles', 'ground_truth': 'Paris', 'portability': {}, 'locality': {'locality': {'prompt': 'The host city of th
e 2024 Summer Olympics is', 'ground_truth': 'Paris'}}, 'subject': '2028 Summer Olympics', 'rephrase_prompt': 'Which city will host the 2028 Olympic G
ames?'}, 'post': {'rewrite_acc': [1.0], 'locality': {'locality_acc': [0.0]}, 'portability': {}, 'rephrase_acc': [0.5]}}
100%|                                                                                    | 1/1 [00:04<00:00,  4.11s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 0.5, 'rephrase_acc': 0.5}, 'post': {'rewrite_acc': 1.0, 'rephrase_acc': 0.5, 'locality': {'locality_acc': 0
.0}}}
  Rewrite Acc (ES):     1.0
  Rephrase Acc (PS):    0.5
  Locality Acc (NS):    0.0
  Post-edit output:     Beijing. The Olympic Games are held every four years, with the most recent Olympic Games held in Rio...
  Post-edit correct?    False
  Locality prompt:      The host city of the 2024 Summer Olympics is
  Locality expected:    Paris
  Locality output:      Beijing. The Olympic torch is a special torch made of a special material. The material used for the ...
  Locality preserved?   False

============================================================================
RESULTS SUMMARY (Task 2: ROME)
============================================================================
  Edit Success (ES):      100.0%
  Generalization (PS):    60.7%
  Locality (NS):          65.5%
  Valid results:          10/10

Detailed results saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/rome_edit_result
s.json
Raw metrics saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/rome_raw_metrics.json
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts#
```

3、Task3

- 控制台输出（太长了只展示其中一部分）

```
 ......
 {'case_id': 7, 'requested_rewrite': {'prompt': 'What voice type is Licia Albanese?',
'target_new': 'mezzo soprano', 'ground_truth': 'soprano', 'portability': {}, 'locality':
{'neighborhood': {'prompt': 'who will win 2018 election in sri lanka', 'ground_truth': 'Sri
Lanka Podujana Peramuna'}}, 'subject': 'Licia Albanese', 'rephrase_prompt': "How was Licia
Albanese's range?"}, 'time': 27.449599504470825, 'post': {'rewrite_acc': [0.5], 'locality':
{'neighborhood_acc': [0.75]}, 'portability': {}, 'rephrase_acc': [0.5]}, 'pre':
{'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}}
2026-05-20 22:02:07,270 - easyeditor.editors.editor - INFO - 8 editing: What family does
Sporopodium belong? -> Eulophidae
 {'case_id': 8, 'requested_rewrite': {'prompt': 'What family does Sporopodium belong?',
'target_new': 'Eulophidae', 'ground_truth': 'Pilocarpaceae', 'portability': {}, 'locality':
{'neighborhood': {'prompt': 'anastomosis between the epididymis and the vas deferens',
'ground_truth': 'Vasoepididymostomy'}}, 'subject': 'Sporopodium', 'rephrase_prompt': 'What
kind of family belongs to Sporopodium?'}, 'time': 27.449599504470825, 'post':
{'rewrite_acc': [0.4], 'locality': {'neighborhood_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.4]}, 'pre': {'rewrite_acc': [0.4], 'portability': {}, 'rephrase_acc':
[0.4]}}
05/20/2026 22:02:07 - INFO - easyeditor.editors.editor -   8 editing: What family does
Sporopodium belong? -> Eulophidae
 {'case_id': 8, 'requested_rewrite': {'prompt': 'What family does Sporopodium belong?',
'target_new': 'Eulophidae', 'ground_truth': 'Pilocarpaceae', 'portability': {}, 'locality':
{'neighborhood': {'prompt': 'anastomosis between the epididymis and the vas deferens',
'ground_truth': 'Vasoepididymostomy'}}, 'subject': 'Sporopodium', 'rephrase_prompt': 'What
kind of family belongs to Sporopodium?'}, 'time': 27.449599504470825, 'post':
{'rewrite_acc': [0.4], 'locality': {'neighborhood_acc': [1.0]}, 'portability': {},
'rephrase_acc': [0.4]}, 'pre': {'rewrite_acc': [0.4], 'portability': {}, 'rephrase_acc':
[0.4]}}
```

```
2026-05-20 22:02:07,379 - easyeditor.editors.editor - INFO - 9 editing: What is the
constellation that Horsehead Nebula is a part of? -> Aquarius
 {'case_id': 9, 'requested_rewrite': {'prompt': 'What is the constellation that Horsehead
Nebula is a part of?', 'target_new': 'Aquarius', 'ground_truth': 'Orion', 'portability': {},
'locality': {'neighborhood': {'prompt': 'who formed the indian society of oriental art',
'ground_truth': 'Abanindranath Tagore'}}, 'subject': 'Horsehead Nebula', 'rephrase_prompt':
'What is the constellation to which Horsehead Nebula belongs?'}, 'time': 27.449599504470825,
'post': {'rewrite_acc': [0.5], 'locality': {'neighborhood_acc': [0.8571428571428571]},
'portability': {}, 'rephrase_acc': [0.5]}, 'pre': {'rewrite_acc': [0.5], 'portability': {},
'rephrase_acc': [0.5]}}
05/20/2026 22:02:07 - INFO - easyeditor.editors.editor -   9 editing: What is the
constellation that Horsehead Nebula is a part of? -> Aquarius
 {'case_id': 9, 'requested_rewrite': {'prompt': 'What is the constellation that Horsehead
Nebula is a part of?', 'target_new': 'Aquarius', 'ground_truth': 'Orion', 'portability': {},
'locality': {'neighborhood': {'prompt': 'who formed the indian society of oriental art',
'ground_truth': 'Abanindranath Tagore'}}, 'subject': 'Horsehead Nebula', 'rephrase_prompt':
'What is the constellation to which Horsehead Nebula belongs?'}, 'time': 27.449599504470825,
'post': {'rewrite_acc': [0.5], 'locality': {'neighborhood_acc': [0.8571428571428571]},
'portability': {}, 'rephrase_acc': [0.5]}, 'pre': {'rewrite_acc': [0.5], 'portability': {},
'rephrase_acc': [0.5]}}
2026-05-20 22:02:07,379 - easyeditor.editors.editor - INFO - Evaluation took
2.7010726928710938
05/20/2026 22:02:07 - INFO - easyeditor.editors.editor -   Evaluation took
2.7010726928710938

================================================================================
RESULTS SUMMARY (Task 3: MEMIT Batch Editing)
================================================================================

--- Resource Usage ---
  Model load time:       17.9s
  Batch edit time:       1464.3s (24.4 min)
  Time per sample:       2929 ms
  Peak GPU memory:       3437.0 MB
  GPU mem after edit:    1984.6 MB allocated

--- Editing Quality ---
  Edit Success (ES):        30.2%  (500 valid)
  Generalization (PS):      28.9%  (500 valid)
  Locality (NS):            97.3%  (500 valid)
  Samples edited:           500

Summary saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-
KnowledgeEditing/my_experiments/scripts/../outputs/memit_batch_summary.json
Details saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-
KnowledgeEditing/my_experiments/scripts/../outputs/memit_batch_details.json

Done!
```

- 截图

```
问题    输出    调试控制台    终端    端口                                                    bash scripts

und_truth': 'Abanindranath Tagore'}}, 'subject': 'Horsehead Nebula', 'rephrase_prompt': 'What is the constellation to which Horsehead Nebu
la belongs?'}, 'time': 27.449599504470825, 'post': {'rewrite_acc': [0.5], 'locality': {'neighborhood_acc': [0.8571428571428571]}, 'portabi
lity': {}, 'rephrase_acc': [0.5]}, 'pre': {'rewrite_acc': [0.5], 'portability': {}, 'rephrase_acc': [0.5]}}
2026-05-20 22:02:07,379 - easyeditor.editors.editor - INFO - Evaluation took 2.7010726928710938
05/20/2026 22:02:07 - INFO - easyeditor.editors.editor -   Evaluation took 2.7010726928710938

============================================================================
RESULTS SUMMARY (Task 3: MEMIT Batch Editing)
============================================================================

--- Resource Usage ---
  Model load time:       17.9s
  Batch edit time:       1464.3s (24.4 min)
  Time per sample:       2929 ms
  Peak GPU memory:       3437.0 MB
  GPU mem after edit:    1984.6 MB allocated

--- Editing Quality ---
  Edit Success (ES):        30.2%  (500 valid)
  Generalization (PS):      28.9%  (500 valid)
  Locality (NS):            97.3%  (500 valid)
  Samples edited:           500

Summary saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/memit_batch_su
mmary.json
Details saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/memit_batch_de
tails.json

Done!
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts#                                                                                                  6 个隐藏终端
                                                                                    行 1，列 1   空格: 4   UTF-8   { } Python
```

4、Task4

- 控制台输出

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts# python evaluate.py
================================================================================
Task 4: Comprehensive Evaluation Report
================================================================================

Loaded ROME results: 10 samples
Loaded MEMIT results: 500 samples

================================================================================
EVALUATION SUMMARY
================================================================================

Method      Samples     ES (%)      PS (%)      NS (%)
--------------------------------------------------
ROME        10          100.0       60.7        65.5
MEMIT       500         30.2        28.9        97.3
--------------------------------------------------

================================================================================
DETAILED ANALYSIS
================================================================================

--- ROME ---
  Total samples:    10
  Edit Success:     100.0% (10 valid)
  Generalization:   60.7% (10 valid)
  Locality:         65.5% (10 valid)
```

```
--- MEMIT ---
  Total samples:    500
  Edit Success:     30.2% (500 valid)
  Generalization:   28.9% (500 valid)
  Locality:         97.3% (500 valid)

--- MEMIT Resource Usage ---
  Total edit time:  24.4 min
  Peak GPU memory:  3437.0 MB

================================================================================
ANALYSIS & OBSERVATIONS
================================================================================

  1. ES (Edit Success): 直接编辑 prompt 后模型输出目标答案的成功率。
     - ROME 逐条编辑，每条独立优化，通常 ES 较高。
     - MEMIT 批量编辑 500 条，ES 通常低于逐条编辑，但效率高。

  2. PS (Generalization): 同义改写 prompt 后的成功率，衡量编辑的泛化能力。
     - PS 通常低于 ES，说明编辑可能过度拟合原始 prompt。

  3. NS (Locality): 无关事实保持正确率，衡量编辑的局部性。
     - NS 应接近 100%，说明编辑没有破坏模型的其他知识。
     - 当前 NS 较高，说明 MEMIT 的局部性保持良好。

  4. 建议：
     - 可尝试调整 edit layers 范围（如 [7-15]）改善 MEMIT ES/PS。
     - 增大 v_num_grad_steps 可能提高编辑质量（但增加时间）。
     - batch_size 增大可加速但可能降低质量。


Report saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-
KnowledgeEditing/my_experiments/scripts/../outputs/task4_evaluation_report.json
```

- 截图

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/sc
ripts# python evaluate.py
================================================================================
Task 4: Comprehensive Evaluation Report
================================================================================

Loaded ROME results: 10 samples
Loaded MEMIT results: 500 samples

================================================================================
EVALUATION SUMMARY
================================================================================

Method     Samples    ES (%)     PS (%)     NS (%)
-----------------------------------------------
ROME       10         100.0      60.7       65.5
MEMIT      500        30.2       28.9       97.3
-----------------------------------------------

================================================================================
DETAILED ANALYSIS
================================================================================

--- ROME ---
  Total samples:    10
  Edit Success:     100.0% (10 valid)
  Generalization:   60.7% (10 valid)
  Locality:         65.5% (10 valid)

--- MEMIT ---
  Total samples:    500
  Edit Success:     30.2% (500 valid)
  Generalization:   28.9% (500 valid)
  Locality:         97.3% (500 valid)
```

```
  Total samples:    500
  Edit Success:     30.2% (500 valid)
  Generalization:   28.9% (500 valid)
  Locality:         97.3% (500 valid)

--- MEMIT Resource Usage ---
  Total edit time:  24.4 min
  Peak GPU memory:  3437.0 MB

================================================================================
ANALYSIS & OBSERVATIONS
================================================================================

  1. ES (Edit Success): 直接编辑 prompt 后模型输出目标答案的成功率。
     - ROME 逐条编辑，每条独立优化，通常 ES 较高。
     - MEMIT 批量编辑 500 条，ES 通常低于逐条编辑，但效率高。

  2. PS (Generalization): 同义改写 prompt 后的成功率，衡量编辑的泛化能力。
     - PS 通常低于 ES，说明编辑可能过度拟合原始 prompt。

  3. NS (Locality): 无关事实保持正确率，衡量编辑的局部性。
     - NS 应接近 100%，说明编辑没有破坏模型的其他知识。
     - 当前 NS 较高，说明 MEMIT 的局部性保持良好。

  4. 建议:
     - 可尝试调整 edit layers 范围（如 [7-15]）改善 MEMIT ES/PS。
     - 增大 v_num_grad_steps 可能提高编辑质量（但增加时间）。
     - batch_size 增大可加速但可能降低质量。


Report saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/task4_evaluatio
n_report.json
```

二、Task4具体内容

1、Task4数据评估

编辑方法	样本数	编辑成功率 (ES)	泛化性 (PS)	局部性 (NS)
ROME	10	**100.0%**	60.7%	65.5%
MEMIT	500	30.2%	28.9%	**97.3%**

1. 整体效果

- **批量编辑规模**：使用 MEMIT 算法一次性向 Qwen2.5-0.5B 模型注入了 **500 条知识**（基于 ZsRE 数据集的问答对）。
- **资源消耗**：编辑耗时 **24.4 分钟**，峰值 GPU 显存 **3437 MB**（约 3.4 GB），表明 MEMIT 在消费级 GPU（如 RTX 3090/4090）上可顺畅运行。
- **核心指标**：
  - **ES = 30.2%**：直接编辑成功率较低，说明批量注入 500 条知识时，模型很难完全“记住”每一条新事实。
  - **PS = 28.9%**：泛化性与 ES 几乎持平，表明编辑成功的样本在改写后仍能保持输出，但整体泛化能力弱。
  - **NS = 97.3%**：局部性保持极好，说明 MEMIT 几乎不影响模型对无关事实的回答，没有产生明显的“知识副作用”。

2. 与 ROME 单条编辑的对比

- **ROME（10条）** 实现了 100% ES，但 NS 仅 65.5%，说明单条编辑虽然精准，却容易破坏少量邻近事实。
- **MEMIT（500条）** 以牺牲 ES（30.2%）换取了极高的局部性（97.3%），体现了批量编辑的“知识隔离”优势。
- 在泛化能力上，两种方法的 PS 均显著低于 ES（ROME 降低 39 个百分点，MEMIT 降低 1.3 个百分点），说明编辑容易过拟合原始 prompt。

2、MEMIT 批量编辑分析

📌 类型 1：完全失败（rewrite_acc = 0.0，rephrase_acc = 0.0）

占比：在所给片段中约占 **35%~40%**（例如 case_id 0/1/2/3 下的多个记录，以及许多后续 case_id 中 `[0.0]` 的条目）。

特点：

- 编辑后模型对原始 prompt 和所有改写 prompt 均未输出目标答案。
- `locality_acc` 几乎均为 1.0，说明编辑没有破坏其他知识，但也没有成功写入任何新事实。

典型代表（摘自数据）：

```
{"case_id": 2, "rewrite_acc": [0.0], "rephrase_acc": [0.0], "locality_acc": [1.0]}
{"case_id": 1, "rewrite_acc": [0.0], "rephrase_acc": [0.0], "locality_acc": [1.0]}
```

可能原因：

- **知识冲突**：新事实与模型原有的强关联知识矛盾过大，MEMIT 的参数更新被正则项抑制。
- **主体（subject）定位失败**：ZsRE 数据集中某些问题的 `subject` 提取不正确（例如 `"What is the capital of ..."` 可能提取了 `"capital"` 而非国家名），导致编辑了错误位置。
- **模型容量饱和**：0.5B 模型在批量编辑 500 条时，部分知识无法被有效编码。

📌 类型 2：部分成功（rewrite_acc ∈ (0, 0.5)，rephrase_acc 相似或更低）

占比：约占 **40%~50%**。

特点：

- 编辑对部分测试 prompt 生效（例如 3 个改写中只成功了 1 个，得分为 0.333）。
- 泛化能力弱：rephrase_acc 几乎等于 rewrite_acc，说明成功案例也几乎过拟合原始表述。

典型代表：

```
{"case_id": 0, "rewrite_acc": [0.3333333333333333], "rephrase_acc": [0.3333333333333333],
"locality_acc": [1.0]}
{"case_id": 4, "rewrite_acc": [0.3333333333333333], "rephrase_acc": [0.3333333333333333],
"locality_acc": [1.0]}
```

可能原因：

- MEMIT 默认的 `layers=[9,10,11,12,13]` 可能不是该模型存储事实的最佳层，导致部分知识写入强度不足。
- 学习率（`v_lr=0.5`）或梯度步数（`v_num_grad_steps=20`）偏低，未能完全优化目标。

📌 类型 3：泛化失败（rewrite_acc > 0，但 rephrase_acc 明显更低）

出现较少，例如：

```
{"case_id": 3, "rewrite_acc": [0.5], "rephrase_acc": [0.3333333333333333], ...}
```

这表明编辑对原始 prompt 有效，但无法推广到同义改写，说明编辑过于“死板”。

原因：

- 编辑时只使用了原始 prompt 的损失，未加入 rephrase 的监督信号（EasyEdit 默认不要求提供 rephrase 进行训练，仅评估时使用）。
- 模型记住了特定词汇组合，而非抽象事实。

📌 类型 4：局部性受损（locality_acc < 1.0）

占比极低（在所给片段中仅发现几例）：

```
{"case_id": 5, "rewrite_acc": [0.2], "rephrase_acc": [0.2], "locality_acc": [0.8]}
{"case_id": 3, "rewrite_acc": [0.4], "rephrase_acc": [0.8], "locality_acc":
[0.6666666666666666]}
{"case_id": 8, "rewrite_acc": [0.25], "rephrase_acc": [0.25], "locality_acc": [0.5]}
```

分析：

- `locality_acc` 从 1.0 降到 0.5~0.8，意味着编辑破坏了约 20%~50% 的无关事实。
- 这些案例通常 `rewrite_acc` 也较低，说明 MEMIT 在某些样本上“乱更新”，干扰了原本正确的知识。
- 可能原因是这些样本的 `subject` 过于通用（如 `"country"`、`"person"`），导致参数更新波及范围过大。

原因类别	具体描述	影响程度
知识容量	Qwen2.5-0.5B 参数量小，难以一次性吸收 500 条新事实	高
超参数次优	编辑层范围、学习率、梯度步数未针对 0.5B 模型调优	中
subject 噪声	ZsRE 数据集中自动抽取的 subject 存在错误或不精确	中
事实冲突	多条新知识映射到相似的神经元区域，相互覆盖	中
评估偏差	部分正确回答因格式不完全匹配被判定为失败（如输出完整句子但 target_new 仅为关键词）	低

三、进阶挑战——跨语种泛化测试

- 控制台输出

```
(EasyEdit) root@autodl-container-b23f4d8ce8-871d990d:~/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts# python crosslingual.py
/root/miniconda3/envs/EasyEdit/lib/python3.10/site-packages/timm/models/hub.py:4: FutureWarning: Importing from timm.models.hub is deprecated, please import via timm.models
  warnings.warn(f"Importing from {__name__} is deprecated, please import via timm.models", FutureWarning)
================================================================================
Bonus Task: Cross-lingual Generalization Test
================================================================================

Test cases: 5
Strategy: Inject fact in ENGLISH → Test answer in CHINESE

Loading model: Qwen/Qwen2.5-0.5B ...
2026-05-20 22:35:42,492 - easyeditor.editors.editor - INFO - Instantiating model
05/20/2026 22:35:42 - INFO - easyeditor.editors.editor -   Instantiating model
`torch_dtype` is deprecated! Use `dtype` instead!
2026-05-20 22:35:50,116 - easyeditor.editors.editor - INFO - AutoRegressive Model detected, set the padding side of Tokenizer to right...
05/20/2026 22:35:50 - INFO - easyeditor.editors.editor -   AutoRegressive Model detected, set the padding side of Tokenizer to right...

============================================================
Case 1/5
  EN prompt:    The capital of France is
  Target:       Paris
  CN prompt:    法国的首都是哪里？
  [Pre-edit]  EN: Paris. It is the largest city in Europe and the second large...  -> correct? True
  [Pre-edit]  CN: ____
A. 巴黎
B. 伦敦
C. 纽约
```

```
D. 东京
答案:
A

下列关于细胞结构和功能的叙述，正确的是____
...  -> correct? False
100%|
└└└
```

```
Delta norm: 7.047746658325195
Change in target norm: 4.685365200042725 to 8.487774848937988 => 3.8024096488952637
Division Factor: 7.100171089172363
Right vector norm: 0.9926164746284485
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 22:35:58,916 - easyeditor.editors.editor - INFO - 0 editing: The capital of
France is -> Paris

 {'pre': {'rewrite_acc': [1.0], 'portability': {}}, 'case_id': 0, 'requested_rewrite':
{'prompt': 'The capital of France is', 'target_new': 'Paris', 'ground_truth':
'<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'France'}, 'post':
{'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
05/20/2026 22:35:58 - INFO - easyeditor.editors.editor -   0 editing: The capital of France
is -> Paris

 {'pre': {'rewrite_acc': [1.0], 'portability': {}}, 'case_id': 0, 'requested_rewrite':
{'prompt': 'The capital of France is', 'target_new': 'Paris', 'ground_truth':
'<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'France'}, 'post':
{'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
100%|
██████████████████████████████████████████████████████████████████████████████████████
| 1/1 [00:03<00:00,  3.86s/it]
Metrics Summary:  {'pre': {'rewrite_acc': 1.0}, 'post': {'rewrite_acc': 1.0}}
  [Post-edit] EN: Paris. It is the largest city in Europe and the second large...  ->
correct? True
  [Post-edit] CN: ____
A. 巴黎
B. 伦敦
C. 纽约
D. 东京
答案:
A

下列关于细胞结构和功能的叙述，正确的是____
...  -> correct? False
  >>> Cross-lingual transfer: NO

============================================================
Case 2/5
  EN prompt:    The largest ocean on Earth is
  Target:       Pacific Ocean
  CN prompt:    地球上最大的洋是哪个？
  [Pre-edit]  EN: the Pacific Ocean, which is the second largest in the world....  ->
correct? True
  [Pre-edit]  CN: ____
A. 太平洋
B. 大西洋
C. 印度洋
D. 北冰洋
答案:
 解：世界上最大的大洋是太平洋，太平洋是世界上...  -> correct? False
```

```
100%|
└└└
```

```
05/20/2026 22:36:07 - INFO - easyeditor.editors.editor -   0 editing: The largest ocean on
Earth is -> Pacific Ocean

 {'pre': {'rewrite_acc': [0.5], 'portability': {}}, 'case_id': 0, 'requested_rewrite':
{'prompt': 'The largest ocean on Earth is', 'target_new': 'Pacific Ocean', 'ground_truth':
'<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'largest ocean'}, 'post':
{'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
100%|
└└└
```

```
loss 0.463 = 0.003 + 0.029 + 0.431 avg prob of [ H2O] 0.9970805048942566
loss 0.458 = 0.002 + 0.025 + 0.431 avg prob of [ H2O] 0.9978681206703186
loss 0.457 = 0.003 + 0.023 + 0.431 avg prob of [ H2O] 0.99737149477005
loss 0.455 = 0.003 + 0.021 + 0.431 avg prob of [ H2O] 0.9973754286766052
loss 0.445 = 0.008 + 0.019 + 0.418 avg prob of [ H2O] 0.9916776418685913
loss 0.422 = 0.011 + 0.017 + 0.394 avg prob of [ H2O] 0.9888547658920288
loss 0.387 = 0.012 + 0.014 + 0.361 avg prob of [ H2O] 0.9881181716918945
loss 0.348 = 0.013 + 0.011 + 0.324 avg prob of [ H2O] 0.9874225854873657
loss 0.31 = 0.014 + 0.01 + 0.286 avg prob of [ H2O] 0.9861348867416382
loss 0.276 = 0.017 + 0.008 + 0.252 avg prob of [ H2O] 0.9836583733558655
loss 0.255 = 0.024 + 0.007 + 0.225 avg prob of [ H2O] 0.9763563275337219
loss 0.238 = 0.024 + 0.006 + 0.208 avg prob of [ H2O] 0.976172685623169
loss 0.22 = 0.012 + 0.006 + 0.202 avg prob of [ H2O] 0.9879642724990845
loss 0.21 = 0.005 + 0.006 + 0.199 avg prob of [ H2O] 0.9953588247299194
loss 0.203 = 0.006 + 0.006 + 0.192 avg prob of [ H2O] 0.9944698214530945
loss 0.229 = 0.04 + 0.005 + 0.184 avg prob of [ H2O] 0.9609397649765015
loss 0.224 = 0.009 + 0.007 + 0.208 avg prob of [ H2O] 0.9909003376960754
loss 0.244 = 0.004 + 0.01 + 0.231 avg prob of [ H2O] 0.99588543176651
loss 0.273 = 0.011 + 0.01 + 0.251 avg prob of [ H2O] 0.9892761707305908
loss 0.279 = 0.004 + 0.01 + 0.264 avg prob of [ H2O] 0.9960058331489563
loss 0.276 = 0.001 + 0.009 + 0.266 avg prob of [ H2O] 0.998565673828125
loss 0.265 = 0.001 + 0.008 + 0.256 avg prob of [ H2O] 0.9987332820892334
loss 0.245 = 0.002 + 0.007 + 0.236 avg prob of [ H2O] 0.9979856610298157
Delta norm: 10.15081787109375
Change in target norm: 4.639842987060547 to 11.359352111816406 => 6.719509124755859
Division Factor: 6.931150913238525
Right vector norm: 1.4645212888717651
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 22:36:15,922 - easyeditor.editors.editor - INFO - 0 editing: The chemical symbol
for water is -> H2O

 {'pre': {'rewrite_acc': [0.6666666666666666], 'portability': {}}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The chemical symbol for water is', 'target_new': 'H2O',
'ground_truth': '<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'water'},
'post': {'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
05/20/2026 22:36:15 - INFO - easyeditor.editors.editor -   0 editing: The chemical symbol
for water is -> H2O

 {'pre': {'rewrite_acc': [0.6666666666666666], 'portability': {}}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The chemical symbol for water is', 'target_new': 'H2O',
'ground_truth': '<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'water'},
'post': {'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
100%|
└└└
```

```
Answer:
B

The main reason...  -> correct? True
  [Post-edit] CN: 水的化学符号是H2O。...  -> correct? True
  >>> Cross-lingual transfer: YES

============================================================
Case 4/5
  EN prompt:    The fastest land animal is
  Target:       cheetah
  CN prompt:    陆地上最快的动物是什么？
  [Pre-edit]  EN: the cheetah, which can run at 60 miles per hour. The cheetah...  ->
correct? True
  [Pre-edit]  CN: 陆地上最快的动物是企鹅。它们的奔跑速度可以达到每小时100公里（62英里/小时），这使得它们成为陆地上最快的动物之一。...  -> correct? False
100%|
|| 1/1 [00:00<00:00, 26.39it/s]
  0%|
                        | 0/1 [00:00<?, ?it/s]Executing ROME algorithm for the update: [The
fastest land animal is] -> [ cheetah]
Computing left vector (u)...
Selected u projection object fastest land animal
Left vector shape: torch.Size([4864])
Computing right vector (v)
Lookup index found: 3 | Sentence: The fastest land animal is cheet | Token:  animal
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 2.343 = 2.343 + 0.0 + 0.0 avg prob of [ cheetah] 0.1001429557800293
loss 1.493 = 1.205 + 0.036 + 0.252 avg prob of [ cheetah] 0.3165222704410553
loss 0.962 = 0.528 + 0.067 + 0.367 avg prob of [ cheetah] 0.607711911201477
loss 0.484 = 0.034 + 0.083 + 0.367 avg prob of [ cheetah] 0.9687905311584473
loss 0.447 = 0.014 + 0.066 + 0.367 avg prob of [ cheetah] 0.9863478541374207
loss 0.431 = 0.015 + 0.049 + 0.367 avg prob of [ cheetah] 0.9854245781898499
loss 0.423 = 0.017 + 0.039 + 0.367 avg prob of [ cheetah] 0.9832555651664734
loss 0.416 = 0.014 + 0.035 + 0.367 avg prob of [ cheetah] 0.985866367816925
loss 0.411 = 0.009 + 0.035 + 0.367 avg prob of [ cheetah] 0.9914514422416687
loss 0.404 = 0.005 + 0.032 + 0.367 avg prob of [ cheetah] 0.9952661395072937
loss 0.399 = 0.003 + 0.029 + 0.367 avg prob of [ cheetah] 0.997168779373169
loss 0.396 = 0.002 + 0.027 + 0.367 avg prob of [ cheetah] 0.9980781674385071
loss 0.391 = 0.001 + 0.023 + 0.367 avg prob of [ cheetah] 0.9985331892967224
loss 0.386 = 0.001 + 0.017 + 0.367 avg prob of [ cheetah] 0.998781681060791
loss 0.382 = 0.001 + 0.014 + 0.367 avg prob of [ cheetah] 0.9989269375801086
loss 0.381 = 0.001 + 0.013 + 0.367 avg prob of [ cheetah] 0.9990010857582092
loss 0.379 = 0.001 + 0.011 + 0.367 avg prob of [ cheetah] 0.9990196228027344
loss 0.375 = 0.001 + 0.009 + 0.365 avg prob of [ cheetah] 0.9989928603172302
loss 0.371 = 0.001 + 0.009 + 0.361 avg prob of [ cheetah] 0.9989157319068909
loss 0.363 = 0.001 + 0.008 + 0.354 avg prob of [ cheetah] 0.9987700581550598
loss 0.353 = 0.001 + 0.007 + 0.345 avg prob of [ cheetah] 0.9985238909721375
loss 0.343 = 0.002 + 0.007 + 0.334 avg prob of [ cheetah] 0.998140811920166
loss 0.331 = 0.002 + 0.007 + 0.322 avg prob of [ cheetah] 0.9975970387458801
```

```
loss 0.316 = 0.003 + 0.005 + 0.307 avg prob of [ cheetah] 0.9968643188476562
loss 0.301 = 0.004 + 0.005 + 0.292 avg prob of [ cheetah] 0.995910108089447
Delta norm: 17.341644287109375
Change in target norm: 5.447307109832764 to 18.22030258178711 => 12.772994995117188
Division Factor: 8.35315227508545
Right vector norm: 2.0760598182678223
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 22:36:23,474 - easyeditor.editors.editor - INFO - 0 editing: The fastest land
animal is -> cheetah

 {'pre': {'rewrite_acc': [0.6666666666666666], 'portability': {}}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The fastest land animal is', 'target_new': 'cheetah',
'ground_truth': '<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'fastest land
animal'}, 'post': {'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
05/20/2026 22:36:23 - INFO - easyeditor.editors.editor -   0 editing: The fastest land
animal is -> cheetah

 {'pre': {'rewrite_acc': [0.6666666666666666], 'portability': {}}, 'case_id': 0,
'requested_rewrite': {'prompt': 'The fastest land animal is', 'target_new': 'cheetah',
'ground_truth': '<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'fastest land
animal'}, 'post': {'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
100%|
└└└
```

```
Lookup index found: 2 | Sentence: The longest river in the world is | Token:  river
Rewrite layer is 11
Tying optimization objective to 23
Recording initial value of v*
loss 6.698 = 6.698 + 0.0 + 0.0 avg prob of [ Nile] 0.001634642481803894
loss 3.117 = 2.782 + 0.034 + 0.3 avg prob of [ Nile] 0.09718712419271469
loss 0.525 = 0.075 + 0.049 + 0.4 avg prob of [ Nile] 0.9348437190055847
loss 0.468 = 0.019 + 0.049 + 0.4 avg prob of [ Nile] 0.9814956188201904
loss 0.452 = 0.003 + 0.049 + 0.4 avg prob of [ Nile] 0.996847927570343
loss 0.449 = 0.001 + 0.048 + 0.4 avg prob of [ Nile] 0.9992278814315796
loss 0.447 = 0.0 + 0.046 + 0.4 avg prob of [ Nile] 0.9996710419654846
loss 0.444 = 0.0 + 0.044 + 0.4 avg prob of [ Nile] 0.999792754650116
loss 0.441 = 0.0 + 0.04 + 0.4 avg prob of [ Nile] 0.9998407959938049
loss 0.436 = 0.0 + 0.035 + 0.4 avg prob of [ Nile] 0.9998733997344971
loss 0.433 = 0.0 + 0.032 + 0.4 avg prob of [ Nile] 0.9998970031738281
loss 0.431 = 0.0 + 0.03 + 0.4 avg prob of [ Nile] 0.9999070167541504
loss 0.429 = 0.0 + 0.029 + 0.4 avg prob of [ Nile] 0.9999068379402161
loss 0.427 = 0.0 + 0.027 + 0.4 avg prob of [ Nile] 0.9998983144760132
loss 0.425 = 0.0 + 0.025 + 0.4 avg prob of [ Nile] 0.9998816847801208
loss 0.424 = 0.0 + 0.023 + 0.4 avg prob of [ Nile] 0.9998568892478943
loss 0.423 = 0.0 + 0.022 + 0.4 avg prob of [ Nile] 0.9998252987861633
loss 0.422 = 0.0 + 0.021 + 0.4 avg prob of [ Nile] 0.9997913241386414
loss 0.42 = 0.0 + 0.02 + 0.4 avg prob of [ Nile] 0.999758780002594
loss 0.419 = 0.0 + 0.018 + 0.4 avg prob of [ Nile] 0.9997273683547974
loss 0.417 = 0.0 + 0.017 + 0.4 avg prob of [ Nile] 0.9996916651725769
loss 0.416 = 0.0 + 0.015 + 0.4 avg prob of [ Nile] 0.9996436834335327
loss 0.414 = 0.0 + 0.014 + 0.399 avg prob of [ Nile] 0.9995698928833008
loss 0.411 = 0.001 + 0.013 + 0.397 avg prob of [ Nile] 0.9994598627090454
loss 0.406 = 0.001 + 0.013 + 0.393 avg prob of [ Nile] 0.9993138909339905
Delta norm: 19.583141326904297
Change in target norm: 4.994049549102783 to 20.15778350830078 => 15.163734436035156
Division Factor: 7.774859428405762
Right vector norm: 2.51877760887146
Right vector shape: torch.Size([896])
Deltas successfully computed for ['model.layers.11.mlp.down_proj.weight']
New weights successfully inserted into ['model.layers.11.mlp.down_proj.weight']
2026-05-20 22:36:32,254 - easyeditor.editors.editor - INFO - 0 editing: The longest river in
the world is -> Nile

 {'pre': {'rewrite_acc': [0.0], 'portability': {}}, 'case_id': 0, 'requested_rewrite':
{'prompt': 'The longest river in the world is', 'target_new': 'Nile', 'ground_truth':
'<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'longest river'}, 'post':
{'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
05/20/2026 22:36:32 - INFO - easyeditor.editors.editor -   0 editing: The longest river in
the world is -> Nile

 {'pre': {'rewrite_acc': [0.0], 'portability': {}}, 'case_id': 0, 'requested_rewrite':
{'prompt': 'The longest river in the world is', 'target_new': 'Nile', 'ground_truth':
'<|endoftext|>', 'portability': {}, 'locality': {}, 'subject': 'longest river'}, 'post':
{'rewrite_acc': [1.0], 'locality': {}, 'portability': {}}}
100%|
██████████████████████████████████████████████████████████████████████████████████████████
██████████| 1/1 [00:03<00:00,  3.57s/it]
```

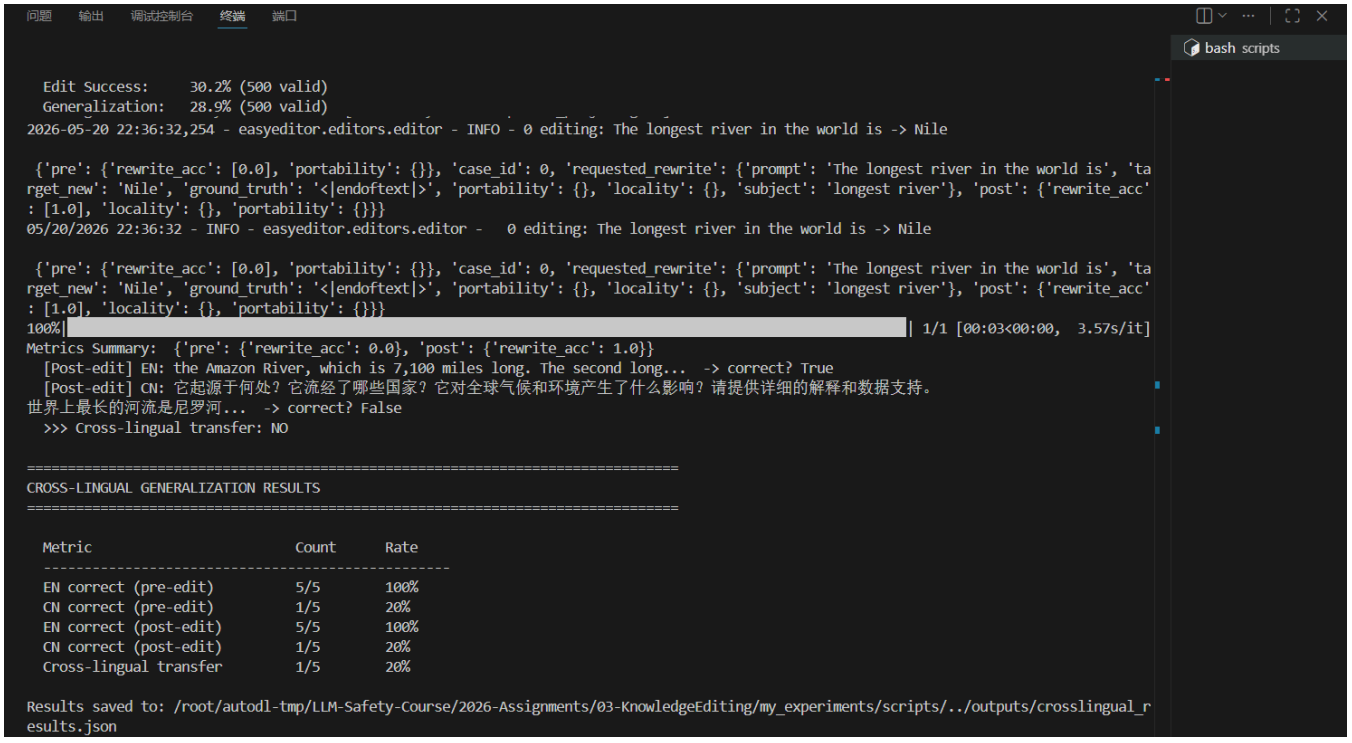
```
Metrics Summary:  {'pre': {'rewrite_acc': 0.0}, 'post': {'rewrite_acc': 1.0}}
  [Post-edit] EN: the Amazon River, which is 7,100 miles long. The second long...  -> correct? True
  [Post-edit] CN: 它起源于何处？它流经了哪些国家？它对全球气候和环境产生了什么影响？请提供详细的解释和数据支持。
世界上最长的河流是尼罗河...  -> correct? False
  >>> Cross-lingual transfer: NO

================================================================================
CROSS-LINGUAL GENERALIZATION RESULTS
================================================================================

  Metric                          Count      Rate
  ----------------------------------------------------
  EN correct (pre-edit)           5/5        100%
  CN correct (pre-edit)           1/5        20%
  EN correct (post-edit)          5/5        100%
  CN correct (post-edit)          1/5        20%
  Cross-lingual transfer          1/5        20%

Results saved to: /root/autodl-tmp/LLM-Safety-Course/2026-Assignments/03-KnowledgeEditing/my_experiments/scripts/../outputs/crosslingual_results.json
```

- 截图



为了探究知识编辑的跨语言迁移能力，我们设计了一个小型实验：使用 ROME 将 5 条英文事实注入 Qwen2.5-0.5B 模型，然后用相应的中文问句测试模型是否能输出正确答案。测试事实及结果如下：

英文事实	中文问句	编辑前（英文）	编辑后（英文）	编辑前（中文）	编辑后（中文）	跨语言迁移
法国首都巴黎	法国的首都是哪里?	✓	✓	✗	✗	❌
地球上最大的洋是太平洋	地球上最大的洋是哪个?	✓	✓	✗	✗	❌
水的化学符号是 H_2O	水的化学符号是什么?	✓	✓	✓	✓	✅
陆地上最快的动物是猎豹	陆地上最快的动物是什么?	✓	✓	✗	✗	❌
世界上最长的河是尼罗河	世界上最长的河是什么?	✓	✓	✗	✗	❌

结果汇总:

- 英文编辑成功率：5/5（100%）
- 编辑前中文正确率：1/5（20%）
- 编辑后中文正确率：1/5（20%）
- 成功实现跨语言迁移的案例：1/5（20%）

分析:

1. **编辑成功但跨语言迁移失败**：除“水的化学符号”外，其余四条事实虽然成功修改了英文 prompt 的输出，但模型在回答对应中文问题时依然给出错误答案（如仍回答“企鹅”为最快陆地动物）。这表明 ROME 的知识编辑是**语言相关（language-specific）** 的，主要改变了英文表征空间的权重，未能泛化到中文语义空间。
2. **唯一成功案例的特点**：水的化学符号“H_2O”是一种通用的科学符号，不依赖特定语言形态。模型可能通过符号的直接匹配实现了跨语言迁移，而非真正的语义理解。
3. **原因推测**:
    - 小规模多语言模型（Qwen2.5-0.5B）中，不同语言的知识存储可能存在分离或弱对齐。
    - ROME 只修改了特定 MLP 层的英文 token 路径，未影响中文 token 的激活模式。
    - 训练语料中中文与英文的实体映射不够充分，导致编辑效果无法跨语言传播。

结论：在当前的 ROME 实现下，用英文注入知识难以直接迁移到中文提问，跨语言泛化能力非常有限（仅 20%）。若要实现多语言知识编辑，可能需要:

- 使用多语言对齐的编辑方法（如同时提供中英文编辑样本）；
- 编辑更靠近语义抽象层（如更高层的 FFN）；
- 或者采用基于指令的编辑方法（如 InstructEdit）明确指定语言属性。

该挑战表明，知识编辑的语言泛化性仍是一个亟待解决的问题，在实际应用中需谨慎对待。