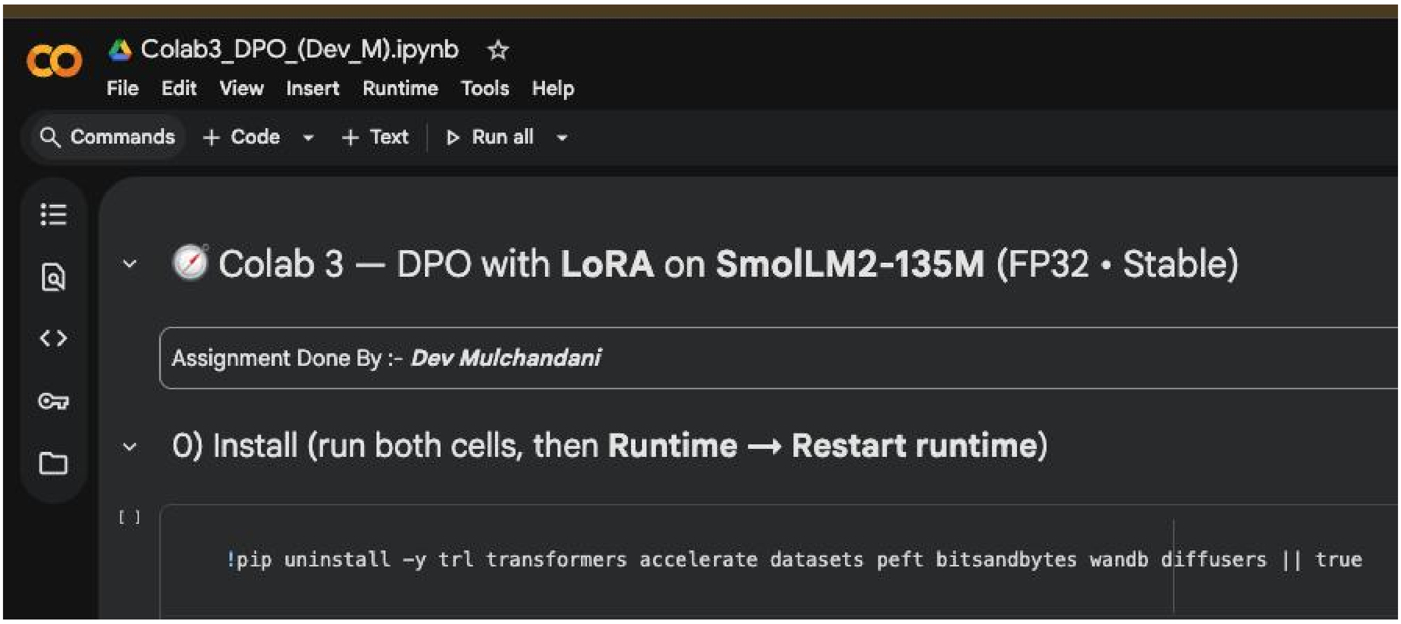
Modern AI with unsloth.ai

❖ Submitted By :- Dev Mulchandani

❖ Colab Notebook :- [Link]()

❖ Colab 3 :- Reinforcement learning

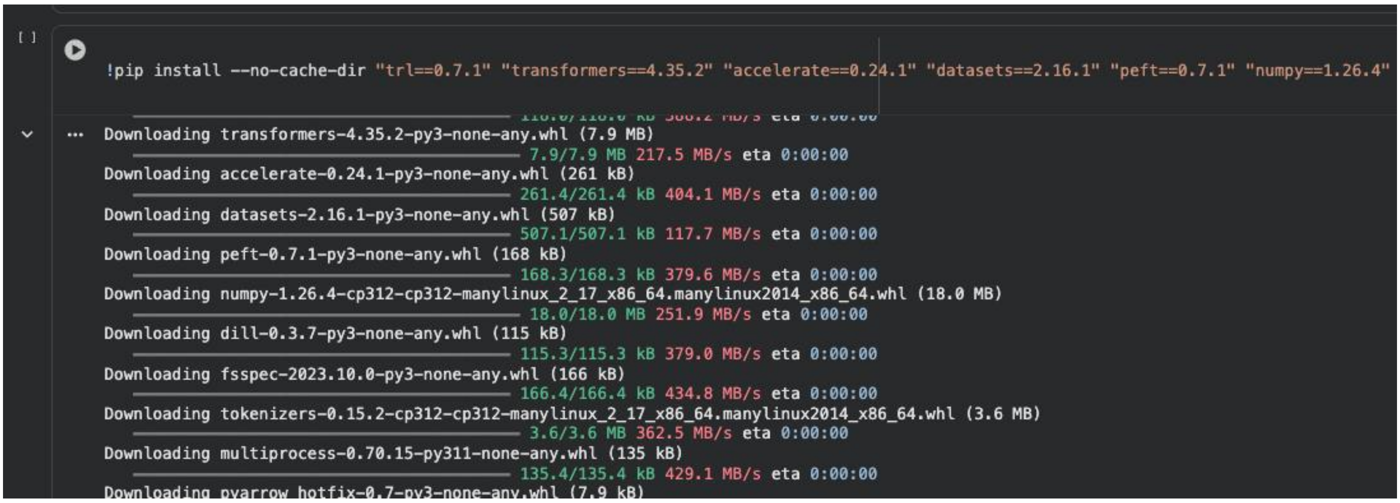


Colab 3 — DPO with **LoRA** on **SmolLM2-135M** (FP32 • Stable)

Assignment Done By :- ***Dev Mulchandani***

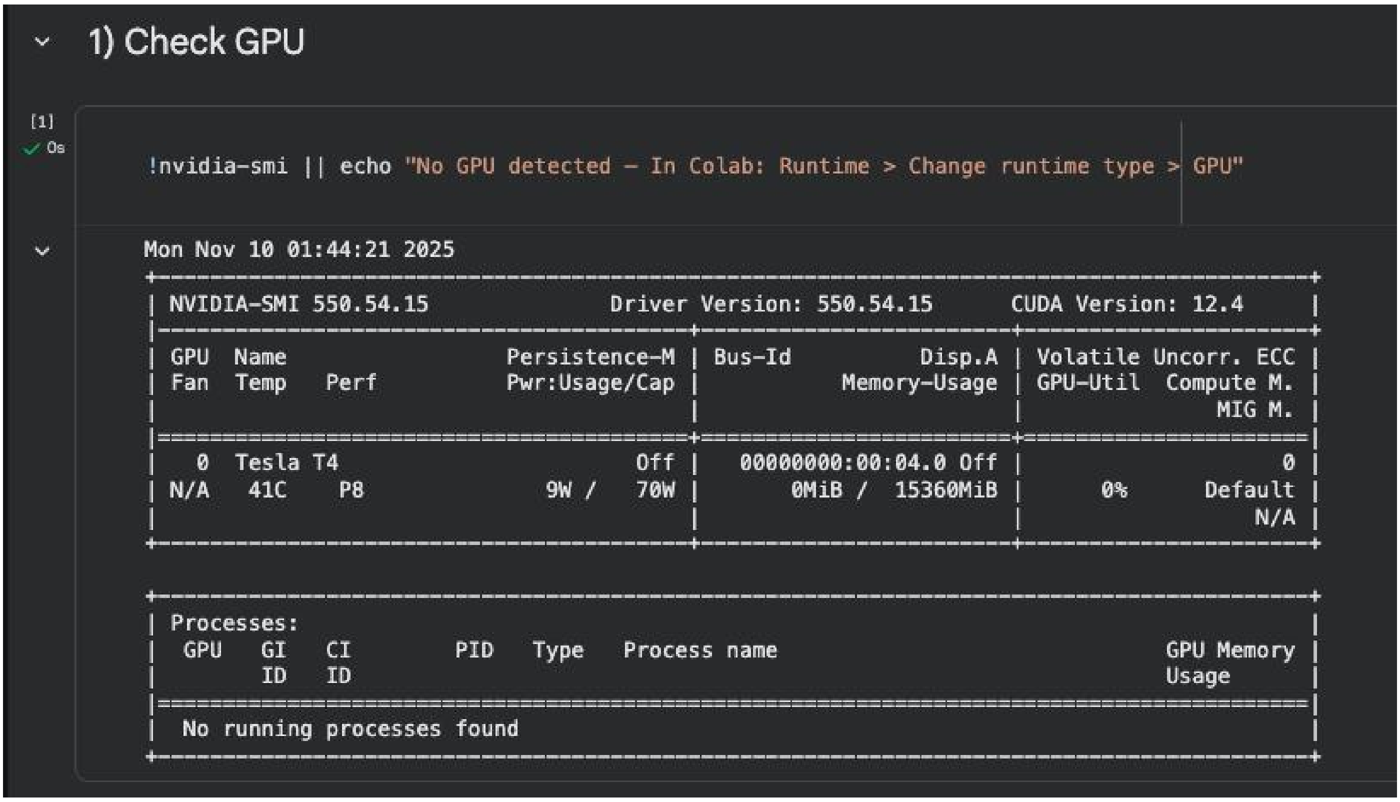
0) Install (run both cells, then **Runtime → Restart runtime)**

```
!pip uninstall -y trl transformers accelerate datasets peft bitsandbytes wandb diffusers || true
```



```
!pip install --no-cache-dir "trl==0.7.1" "transformers==4.35.2" "accelerate==0.24.1" "datasets==2.16.1" "peft==0.7.1" "numpy==1.26.4"
```

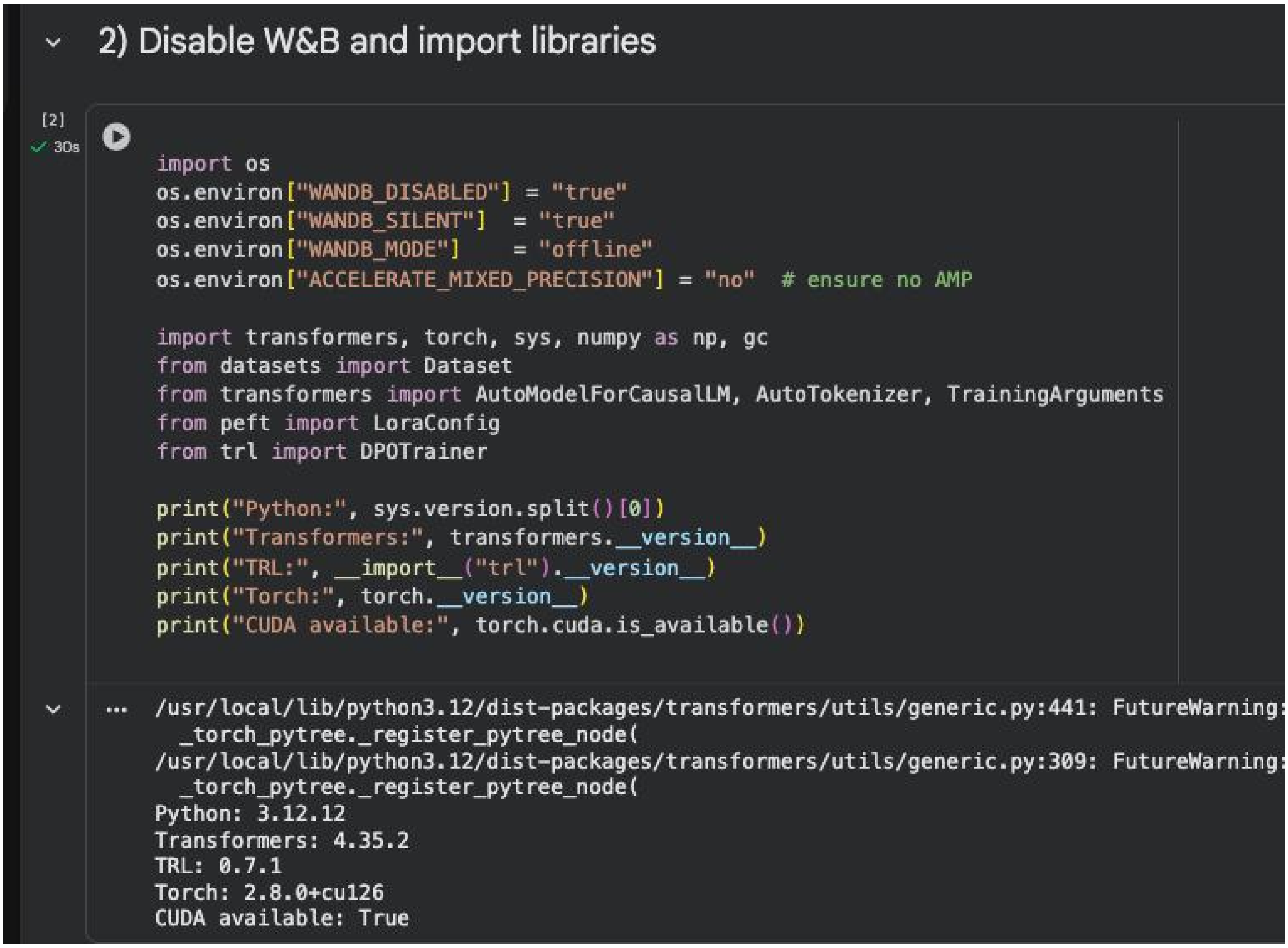
```
Downloading transformers-4.35.2-py3-none-any.whl (7.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 7.9/7.9 MB 217.5 MB/s eta 0:00:00
Downloading accelerate-0.24.1-py3-none-any.whl (261 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 261.4/261.4 kB 404.1 MB/s eta 0:00:00
Downloading datasets-2.16.1-py3-none-any.whl (507 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 507.1/507.1 kB 117.7 MB/s eta 0:00:00
Downloading peft-0.7.1-py3-none-any.whl (168 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 168.3/168.3 kB 379.6 MB/s eta 0:00:00
Downloading numpy-1.26.4-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (18.0 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 18.0/18.0 MB 251.9 MB/s eta 0:00:00
Downloading dill-0.3.7-py3-none-any.whl (115 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 115.3/115.3 kB 379.0 MB/s eta 0:00:00
Downloading fsspec-2023.10.0-py3-none-any.whl (166 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 166.4/166.4 kB 434.8 MB/s eta 0:00:00
Downloading tokenizers-0.15.2-cp312-cp312-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.6 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.6/3.6 MB 362.5 MB/s eta 0:00:00
Downloading multiprocess-0.70.15-py311-none-any.whl (135 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 135.4/135.4 kB 429.1 MB/s eta 0:00:00
Downloading pyarrow_hotfix-0.7-py3-none-any.whl (7.9 kB)
```



1) Check GPU

```
!nvidia-smi || echo "No GPU detected – In Colab: Runtime > Change runtime type > GPU"
```

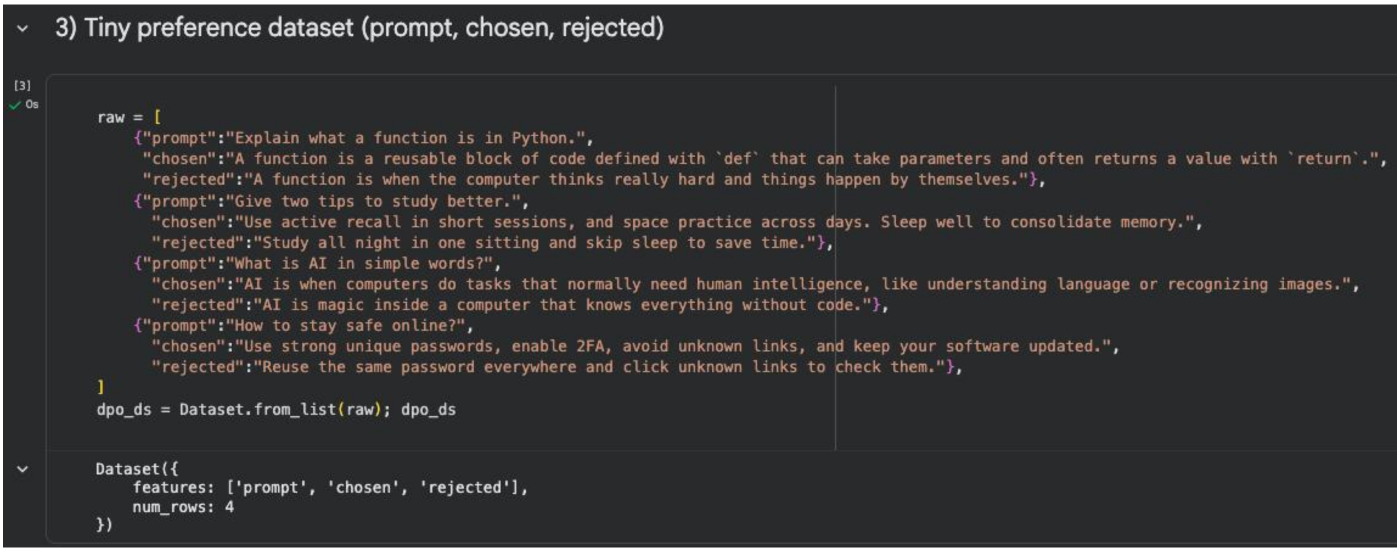
```
Mon Nov 10 01:44:21 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 550.54.15              Driver Version: 550.54.15      CUDA Version: 12.4     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  Tesla T4                       Off |   00000000:00:04.0 Off |                    0 |
| N/A   41C    P8              9W /   70W |       0MiB /  15360MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI        PID   Type   Process name                              GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
```



2) Disable W&B and import libraries

```
import os
os.environ["WANDB_DISABLED"] = "true"
os.environ["WANDB_SILENT"]   = "true"
os.environ["WANDB_MODE"]     = "offline"
os.environ["ACCELERATE_MIXED_PRECISION"] = "no"  # ensure no AMP

import transformers, torch, sys, numpy as np, gc
from datasets import Dataset
from transformers import AutoModelForCausalLM, AutoTokenizer, TrainingArguments
from peft import LoraConfig
from trl import DPOTrainer

print("Python:", sys.version.split()[0])
print("Transformers:", transformers.__version__)
print("TRL:", __import__("trl").__version__)
print("Torch:", torch.__version__)
print("CUDA available:", torch.cuda.is_available())
```

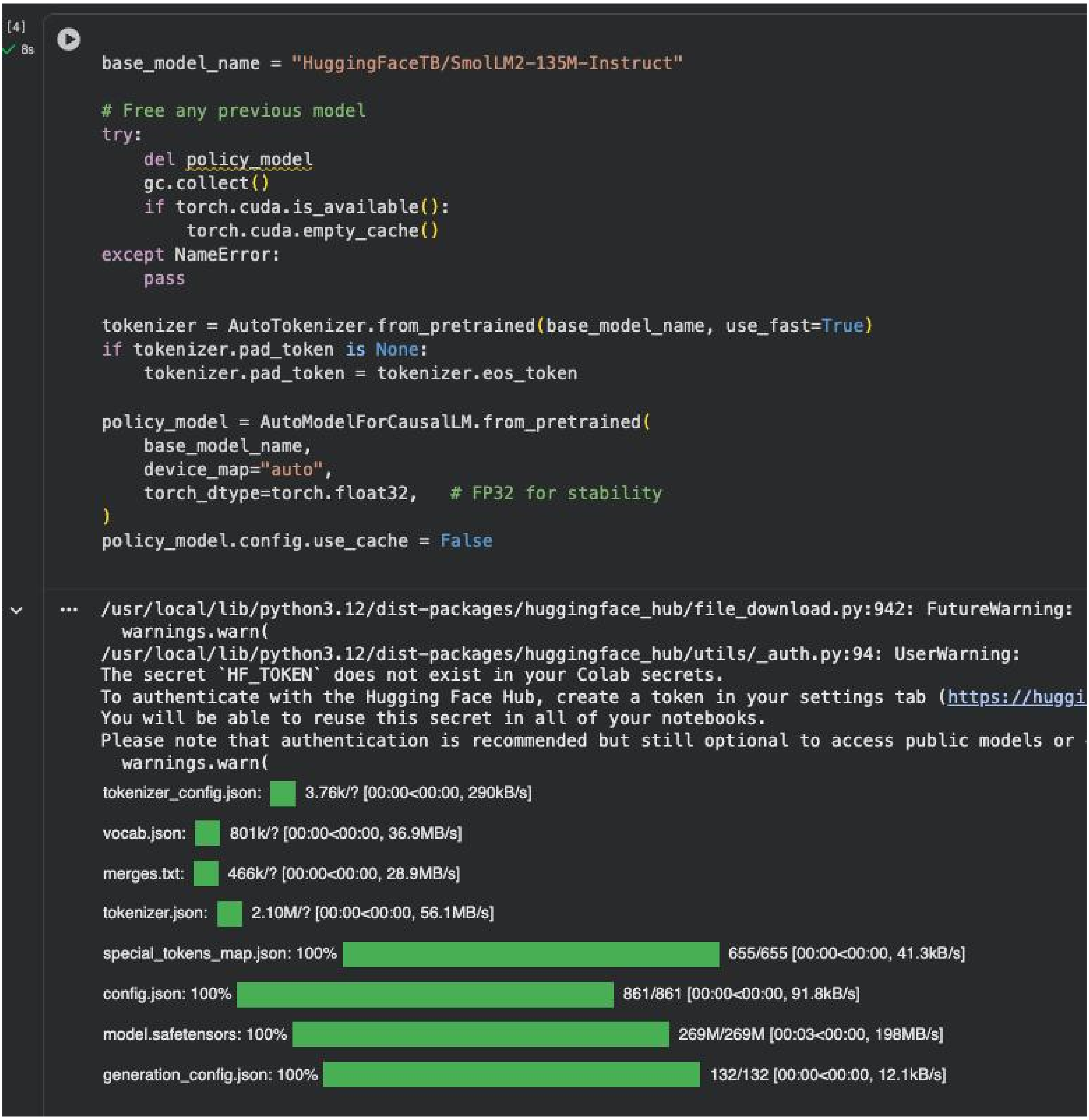
```
/usr/local/lib/python3.12/dist-packages/transformers/utils/generic.py:441: FutureWarning:
  _torch_pytree._register_pytree_node(
/usr/local/lib/python3.12/dist-packages/transformers/utils/generic.py:309: FutureWarning:
  _torch_pytree._register_pytree_node(
Python: 3.12.12
Transformers: 4.35.2
TRL: 0.7.1
Torch: 2.8.0+cu126
CUDA available: True
```



3) Tiny preference dataset (prompt, chosen, rejected)

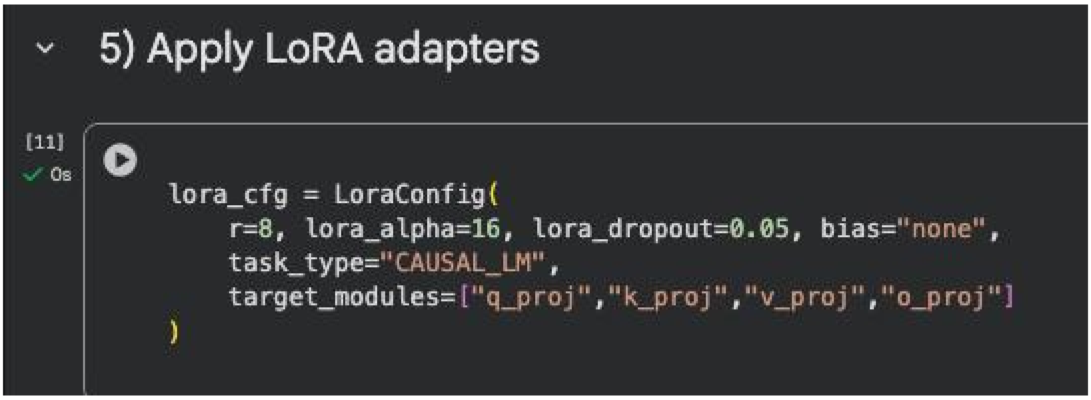
```
raw = [
    {"prompt":"Explain what a function is in Python.",
     "chosen":"A function is a reusable block of code defined with `def` that can take parameters and often returns a value with `return`.",
     "rejected":"A function is when the computer thinks really hard and things happen by themselves."},
    {"prompt":"Give two tips to study better.",
      "chosen":"Use active recall in short sessions, and space practice across days. Sleep well to consolidate memory.",
      "rejected":"Study all night in one sitting and skip sleep to save time."},
    {"prompt":"What is AI in simple words?",
      "chosen":"AI is when computers do tasks that normally need human intelligence, like understanding language or recognizing images.",
      "rejected":"AI is magic inside a computer that knows everything without code."},
    {"prompt":"How to stay safe online?",
      "chosen":"Use strong unique passwords, enable 2FA, avoid unknown links, and keep your software updated.",
      "rejected":"Reuse the same password everywhere and click unknown links to check them."},
]
dpo_ds = Dataset.from_list(raw); dpo_ds
```

```
Dataset({
    features: ['prompt', 'chosen', 'rejected'],
    num_rows: 4
})
```



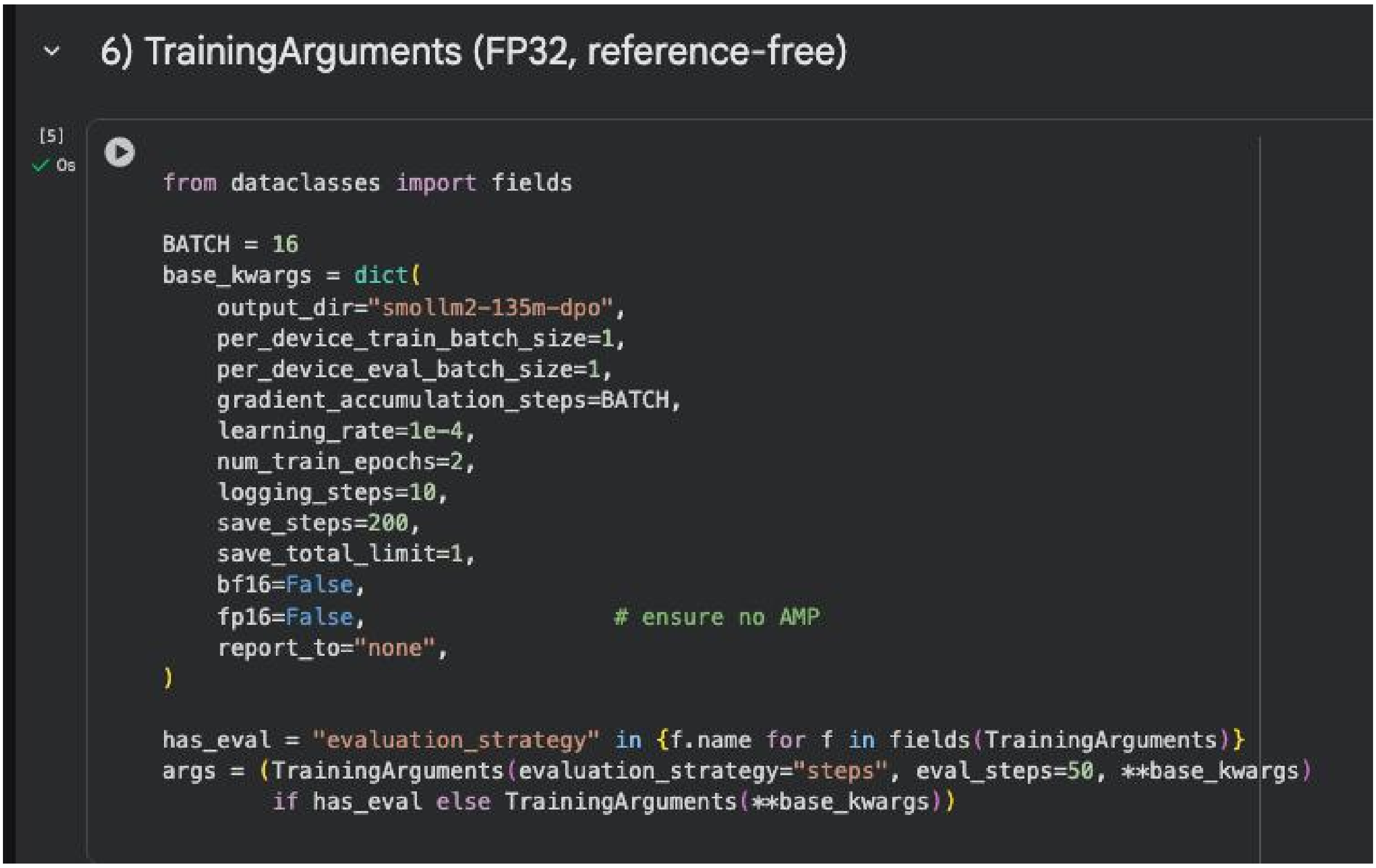
```
base_model_name = "HuggingFaceTB/SmolLM2-135M-Instruct"

# Free any previous model
try:
    del policy_model
    gc.collect()
    if torch.cuda.is_available():
        torch.cuda.empty_cache()
except NameError:
    pass

tokenizer = AutoTokenizer.from_pretrained(base_model_name, use_fast=True)
if tokenizer.pad_token is None:
    tokenizer.pad_token = tokenizer.eos_token

policy_model = AutoModelForCausalLM.from_pretrained(
    base_model_name,
    device_map="auto",
    torch_dtype=torch.float32,   # FP32 for stability
)
policy_model.config.use_cache = False
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/file_download.py:942: FutureWarning:
  warnings.warn(
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggi
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or
  warnings.warn(
tokenizer_config.json: 3.76k/? [00:00<00:00, 290kB/s]
vocab.json: 801k/? [00:00<00:00, 36.9MB/s]
merges.txt: 466k/? [00:00<00:00, 28.9MB/s]
tokenizer.json: 2.10M/? [00:00<00:00, 56.1MB/s]
special_tokens_map.json: 100% 655/655 [00:00<00:00, 41.3kB/s]
config.json: 100% 861/861 [00:00<00:00, 91.8kB/s]
model.safetensors: 100% 269M/269M [00:03<00:00, 198MB/s]
generation_config.json: 100% 132/132 [00:00<00:00, 12.1kB/s]
```



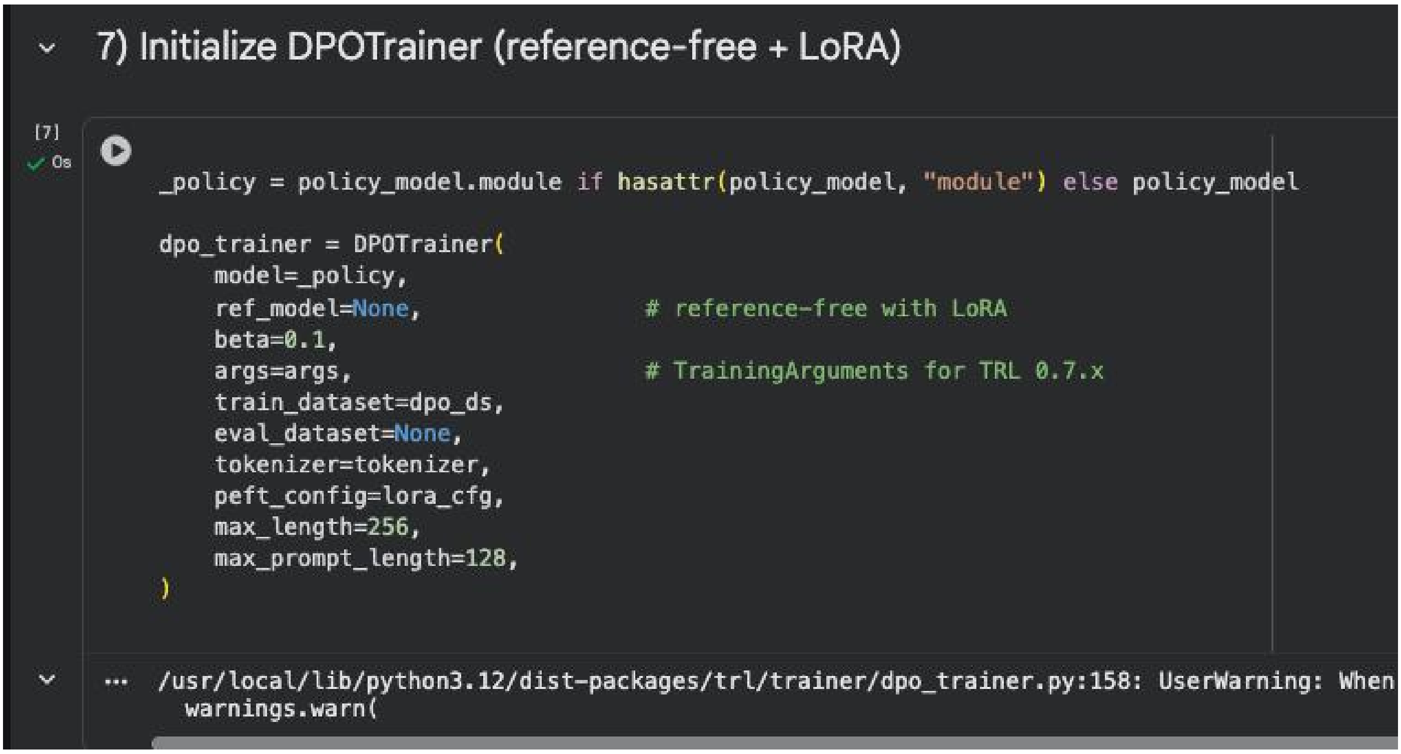
5) Apply LoRA adapters

```python
lora_cfg = LoraConfig(
    r=8, lora_alpha=16, lora_dropout=0.05, bias="none",
    task_type="CAUSAL_LM",
    target_modules=["q_proj","k_proj","v_proj","o_proj"]
)
```



6) TrainingArguments (FP32, reference-free)

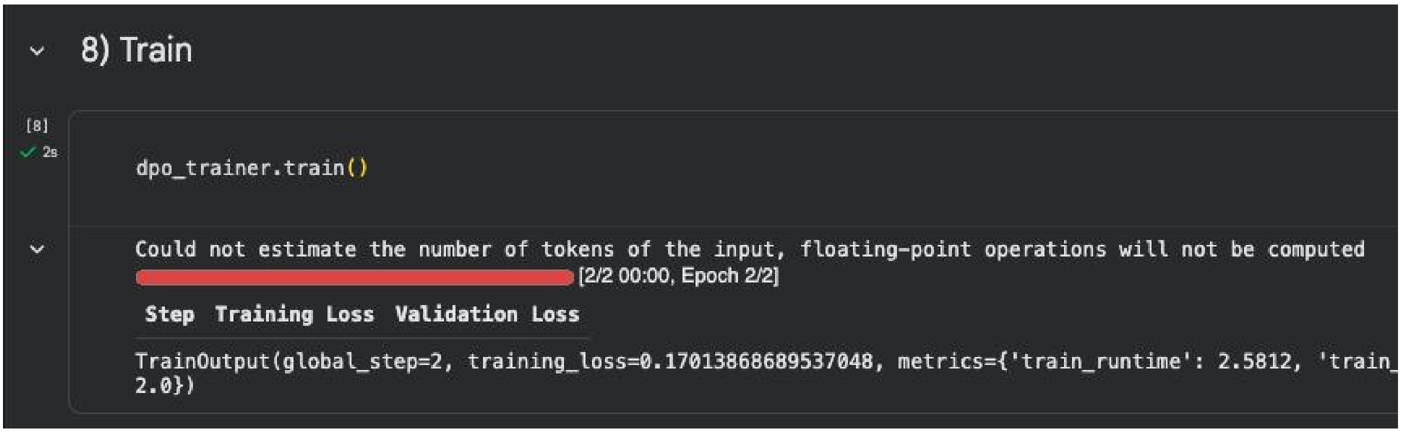
```python
from dataclasses import fields

BATCH = 16
base_kwargs = dict(
    output_dir="smollm2-135m-dpo",
    per_device_train_batch_size=1,
    per_device_eval_batch_size=1,
    gradient_accumulation_steps=BATCH,
    learning_rate=1e-4,
    num_train_epochs=2,
    logging_steps=10,
    save_steps=200,
    save_total_limit=1,
    bf16=False,
    fp16=False,                  # ensure no AMP
    report_to="none",
)

has_eval = "evaluation_strategy" in {f.name for f in fields(TrainingArguments)}
args = (TrainingArguments(evaluation_strategy="steps", eval_steps=50, **base_kwargs)
        if has_eval else TrainingArguments(**base_kwargs))
```



7) Initialize DPOTrainer (reference-free + LoRA)

```
_policy = policy_model.module if hasattr(policy_model, "module") else policy_model

dpo_trainer = DPOTrainer(
    model=_policy,
    ref_model=None,              # reference-free with LoRA
    beta=0.1,
    args=args,                   # TrainingArguments for TRL 0.7.x
    train_dataset=dpo_ds,
    eval_dataset=None,
    tokenizer=tokenizer,
    peft_config=lora_cfg,
    max_length=256,
    max_prompt_length=128,
)
```

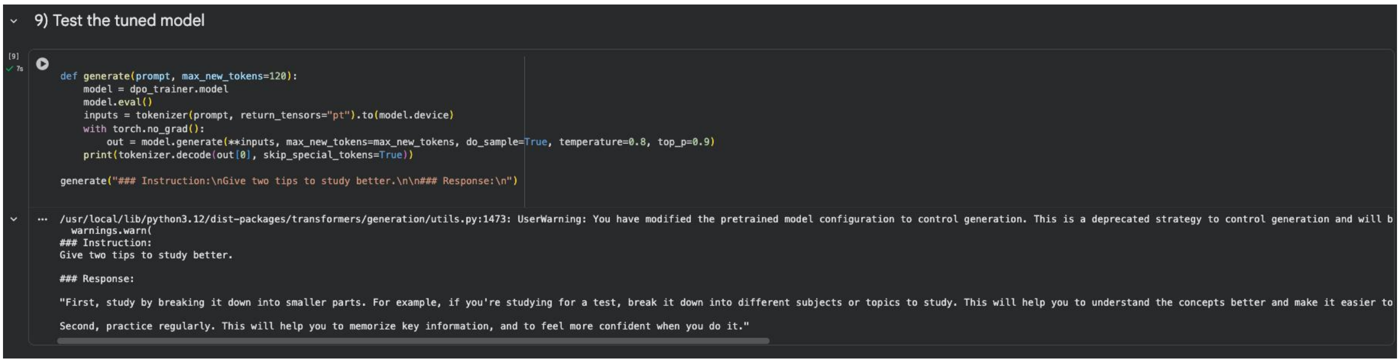
```
/usr/local/lib/python3.12/dist-packages/trl/trainer/dpo_trainer.py:158: UserWarning: When
  warnings.warn(
```



8) Train

```
dpo_trainer.train()
```

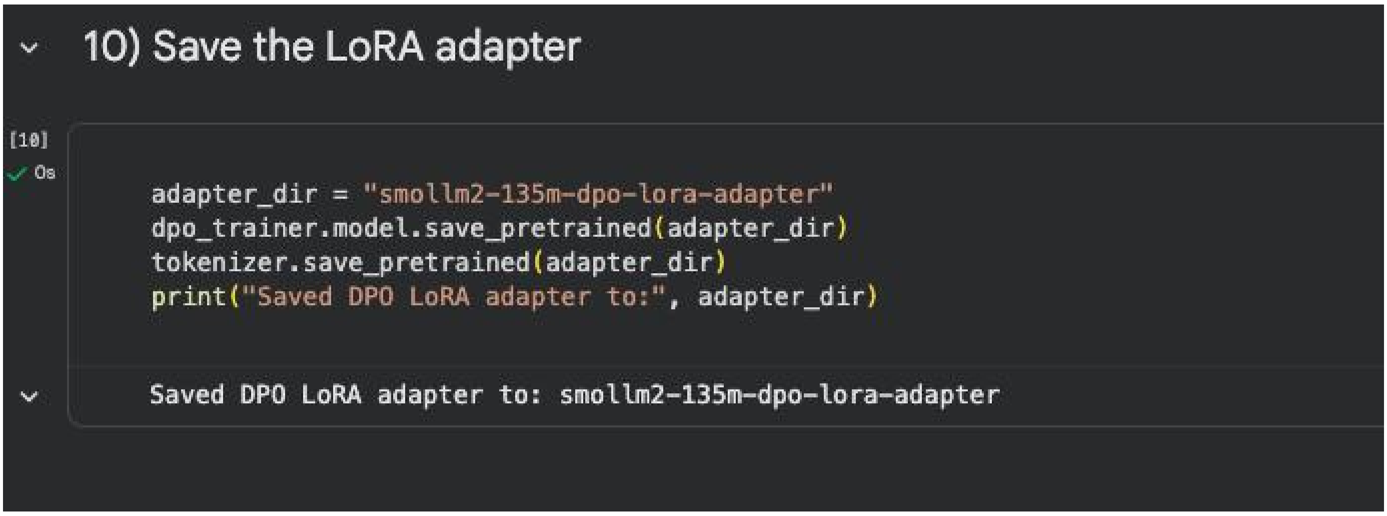
```
Could not estimate the number of tokens of the input, floating-point operations will not be computed
[2/2 00:00, Epoch 2/2]
 Step  Training Loss  Validation Loss
TrainOutput(global_step=2, training_loss=0.17013868689537048, metrics={'train_runtime': 2.5812, 'train_
2.0})
```



9) Test the tuned model

```
def generate(prompt, max_new_tokens=120):
    model = dpo_trainer.model
    model.eval()
    inputs = tokenizer(prompt, return_tensors="pt").to(model.device)
    with torch.no_grad():
        out = model.generate(**inputs, max_new_tokens=max_new_tokens, do_sample=True, temperature=0.8, top_p=0.9)
    print(tokenizer.decode(out[0], skip_special_tokens=True))

generate("### Instruction:\nGive two tips to study better.\n\n### Response:\n")
```

```
/usr/local/lib/python3.12/dist-packages/transformers/generation/utils.py:1473: UserWarning: You have modified the pretrained model configuration to control generation. This is a deprecated strategy to control generation and will b
  warnings.warn(
### Instruction:
Give two tips to study better.

### Response:

"First, study by breaking it down into smaller parts. For example, if you're studying for a test, break it down into different subjects or topics to study. This will help you to understand the concepts better and make it easier to

Second, practice regularly. This will help you to memorize key information, and to feel more confident when you do it."
```



10) Save the LoRA adapter

[10]
0s

```
adapter_dir = "smollm2-135m-dpo-lora-adapter"
dpo_trainer.model.save_pretrained(adapter_dir)
tokenizer.save_pretrained(adapter_dir)
print("Saved DPO LoRA adapter to:", adapter_dir)
```

```
Saved DPO LoRA adapter to: smollm2-135m-dpo-lora-adapter
```