```
import os, sys, torch

os.environ["TRANSFORMERS_NO_TF"] = "1"
os.environ["HF_HUB_DISABLE_TF_WARNING"] = "1"

print("Python:", sys.version)
print("CUDA available:", torch.cuda.is_available())
if torch.cuda.is_available():
    print("GPU:", torch.cuda.get_device_name(0))
    print("BF16 supported:", torch.cuda.is_bf16_supported())

!pip -q install "transformers>=4.43,<4.47" "accelerate>=0.30" datasets==2.21.0
```

```
Python: 3.12.12 (main, Oct 10 2025, 08:52:57) [GCC 11.4.0]
CUDA available: True
GPU: NVIDIA A100-SXM4-40GB
BF16 supported: True
```

```
from transformers import AutoTokenizer, AutoModelForCausalLM

tok = AutoTokenizer.from_pretrained("gpt2-medium")
tok.pad_token = tok.eos_token

base = AutoModelForCausalLM.from_pretrained(
    "gpt2-medium",
    torch_dtype=torch.bfloat16,
    device_map="auto",
    attn_implementation="sdpa",
)
base.gradient_checkpointing_enable()
base.config.use_cache = False

print("GPT-2-medium loaded on GPU using SDPA ✓")
```

```
GPT-2-medium loaded on GPU using SDPA ✓
```

```
prompt = "You are a helpful assistant. Q: What is RoPE in transformers? A:"
inputs = tok(prompt, return_tensors="pt").to("cuda")

out = base.generate(
    **inputs,
    max_new_tokens=64,
    do_sample=True,
    top_p=0.9
)

print(tok.decode(out[0], skip_special_tokens=True))
```

```
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
You are a helpful assistant. Q: What is RoPE in transformers? A: RoPE (pronounced 'ro') stands for Recovery and Emerg
```

```
prompt = (
    "Question: What is Rotary Position Embedding (RoPE)? "
    "Answer in 2-3 clear sentences.\nA:"
)
inputs = tok(prompt, return_tensors="pt").to("cuda")

with torch.inference_mode():
    out = base.generate(
        **inputs,
        max_new_tokens=80,
        do_sample=False,
        repetition_penalty=1.1,
        eos_token_id=tok.eos_token_id,
    )

import textwrap
print("\n".join(textwrap.wrap(tok.decode(out[0], skip_special_tokens=True), 80)))
```

```
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
Question: What is Rotary Position Embedding (RoPE)? Answer in 2-3 clear
sentences. A: RoCE is a technique that allows you to position your body so that
it's facing the same direction as your opponent, but with an angle of 45 degrees
or more from their face. It can be used for any kind and variety type moves such
```

```
like backhand/fist punches, uppercut combos etc.. The key here is not only
positioning yourself correctly on each side of your opponents head when
```

```
context = (
    "Context:\n"
    "RoPE encodes positions by rotating Q/K vectors using sinusoidal phases. "
    "It naturally introduces relative positioning and distance-based decay.\n\n"
)

prompt = context + "Question: What is RoPE?\nA:"

inputs = tok(prompt, return_tensors="pt").to("cuda")

with torch.inference_mode():
    out = base.generate(
        **inputs,
        max_new_tokens=90,
        do_sample=False,
        repetition_penalty=1.05,
    )

print("\n".join(textwrap.wrap(tok.decode(out[0], skip_special_tokens=True), 80)))
```

```
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
Context: RoPE encodes positions by rotating Q/K vectors using sinusoidal phases.
It naturally introduces relative positioning and distance-based decay.
Question: What is RoPE? A: RoCE encodings are a variant of RoPE that uses the
same basic concept but with different properties, such as rotational phase (Q/k)
or position-dependent decay (P/d). The main difference between RoCE encoding and
RoPE encoding is that RoCE encoder does not use any special transformations to
encode its data; it just computes all possible values for each vector in an
arbitrary order. This allows us more flexibility
```

```
model_id = "tiiuae/falcon-7b-instruct"

tok2 = AutoTokenizer.from_pretrained(model_id)
tok2.pad_token = tok2.eos_token

base2 = AutoModelForCausalLM.from_pretrained(
    model_id,
    torch_dtype=torch.bfloat16,
    device_map="auto",
    attn_implementation="sdpa",
)

inputs = tok2("What is RoPE? Give a short answer.", return_tensors="pt").to("cuda")
with torch.inference_mode():
    out = base2.generate(**inputs, max_new_tokens=120, do_sample=False)

print("\n".join(textwrap.wrap(tok2.decode(out[0], skip_special_tokens=True), 80)))
```

```
Loading checkpoint shards: 100%                                    2/2 [00:05<00:00,  2.56s/it]
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
What is RoPE? Give a short answer. RoPE stands for 'Rope-based Obstacle
Placement Engine'. It is a tool used in robotics to plan and execute complex
obstacle avoidance maneuvers.
```