```
!pip install vllm accelerate transformers
#accelerate is a hugging face tool use for running models without
vLLM, its notstrictlyreqyiired

Collecting vllm
  Downloading vllm-0.8.5.post1-cp38-abi3-
manylinux1_x86_64.whl.metadata (14 kB)
Requirement already satisfied: accelerate in
/usr/local/lib/python3.11/dist-packages (1.6.0)
Requirement already satisfied: transformers in
/usr/local/lib/python3.11/dist-packages (4.51.3)
Requirement already satisfied: cachetools in
/usr/local/lib/python3.11/dist-packages (from vllm) (5.5.2)
Requirement already satisfied: psutil in
/usr/local/lib/python3.11/dist-packages (from vllm) (5.9.5)
Requirement already satisfied: sentencepiece in
/usr/local/lib/python3.11/dist-packages (from vllm) (0.2.0)
Requirement already satisfied: numpy in
/usr/local/lib/python3.11/dist-packages (from vllm) (2.0.2)
Requirement already satisfied: requests>=2.26.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-
packages (from vllm) (4.67.1)
Collecting blake3 (from vllm)
  Downloading blake3-1.0.4-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.2 kB)
Requirement already satisfied: py-cpuinfo in
/usr/local/lib/python3.11/dist-packages (from vllm) (9.0.0)
Requirement already satisfied: huggingface-hub>=0.30.0 in
/usr/local/lib/python3.11/dist-packages (from huggingface-
hub[hf_xet]>=0.30.0->vllm) (0.30.2)
Requirement already satisfied: tokenizers>=0.21.1 in
/usr/local/lib/python3.11/dist-packages (from vllm) (0.21.1)
Requirement already satisfied: protobuf in
/usr/local/lib/python3.11/dist-packages (from vllm) (5.29.4)
Collecting fastapi>=0.115.0 (from fastapi[standard]>=0.115.0->vllm)
  Downloading fastapi-0.115.12-py3-none-any.whl.metadata (27 kB)
Requirement already satisfied: aiohttp in
/usr/local/lib/python3.11/dist-packages (from vllm) (3.11.15)
Requirement already satisfied: openai>=1.52.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (1.76.2)
Requirement already satisfied: pydantic>=2.9 in
/usr/local/lib/python3.11/dist-packages (from vllm) (2.11.4)
Requirement already satisfied: prometheus_client>=0.18.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (0.21.1)
Requirement already satisfied: pillow in
/usr/local/lib/python3.11/dist-packages (from vllm) (11.2.1)
Collecting prometheus-fastapi-instrumentator>=7.0.0 (from vllm)
  Downloading prometheus_fastapi_instrumentator-7.1.0-py3-none-
any.whl.metadata (13 kB)
```

```
Collecting tiktoken>=0.6.0 (from vllm)
  Downloading tiktoken-0.9.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.7 kB)
Collecting lm-format-enforcer<0.11,>=0.10.11 (from vllm)
  Downloading lm_format_enforcer-0.10.11-py3-none-any.whl.metadata (17
kB)
Collecting llguidance<0.8.0,>=0.7.9 (from vllm)
  Downloading llguidance-0.7.19-cp39-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (9.6 kB)
Collecting outlines==0.1.11 (from vllm)
  Downloading outlines-0.1.11-py3-none-any.whl.metadata (17 kB)
Collecting lark==1.2.2 (from vllm)
  Downloading lark-1.2.2-py3-none-any.whl.metadata (1.8 kB)
Collecting xgrammar==0.1.18 (from vllm)
  Downloading xgrammar-0.1.18-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.6 kB)
Requirement already satisfied: typing_extensions>=4.10 in
/usr/local/lib/python3.11/dist-packages (from vllm) (4.13.2)
Requirement already satisfied: filelock>=3.16.1 in
/usr/local/lib/python3.11/dist-packages (from vllm) (3.18.0)
Collecting partial-json-parser (from vllm)
  Downloading partial_json_parser-0.2.1.1.post5-py3-none-
any.whl.metadata (6.1 kB)
Collecting pyzmq>=25.0.0 (from vllm)
  Downloading pyzmq-26.4.0-cp311-cp311-
manylinux_2_28_x86_64.whl.metadata (6.0 kB)
Collecting msgspec (from vllm)
  Downloading msgspec-0.19.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (6.9 kB)
Collecting gguf>=0.13.0 (from vllm)
  Downloading gguf-0.16.3-py3-none-any.whl.metadata (4.4 kB)
Requirement already satisfied: importlib_metadata in
/usr/local/lib/python3.11/dist-packages (from vllm) (8.7.0)
Collecting mistral_common>=1.5.4 (from mistral_common[opencv]>=1.5.4-
>vllm)
  Downloading mistral_common-1.5.4-py3-none-any.whl.metadata (4.5 kB)
Requirement already satisfied: opencv-python-headless>=4.11.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (4.11.0.86)
Requirement already satisfied: pyyaml in
/usr/local/lib/python3.11/dist-packages (from vllm) (6.0.2)
Requirement already satisfied: einops in
/usr/local/lib/python3.11/dist-packages (from vllm) (0.8.1)
Collecting compressed-tensors==0.9.3 (from vllm)
  Downloading compressed_tensors-0.9.3-py3-none-any.whl.metadata (7.0
kB)
Collecting depyf==0.18.0 (from vllm)
  Downloading depyf-0.18.0-py3-none-any.whl.metadata (7.1 kB)
Requirement already satisfied: cloudpickle in
/usr/local/lib/python3.11/dist-packages (from vllm) (3.1.1)
```

```
Collecting watchfiles (from vllm)
  Downloading watchfiles-1.0.5-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Collecting python-json-logger (from vllm)
  Downloading python_json_logger-3.3.0-py3-none-any.whl.metadata (4.0
kB)
Requirement already satisfied: scipy in
/usr/local/lib/python3.11/dist-packages (from vllm) (1.15.2)
Collecting ninja (from vllm)
  Downloading ninja-1.11.1.4-py3-none-
manylinux_2_12_x86_64.manylinux2010_x86_64.whl.metadata (5.0 kB)
Collecting opentelemetry-sdk<1.27.0,>=1.26.0 (from vllm)
  Downloading opentelemetry_sdk-1.26.0-py3-none-any.whl.metadata (1.5
kB)
Collecting opentelemetry-api<1.27.0,>=1.26.0 (from vllm)
  Downloading opentelemetry_api-1.26.0-py3-none-any.whl.metadata (1.4
kB)
Collecting opentelemetry-exporter-otlp<1.27.0,>=1.26.0 (from vllm)
  Downloading opentelemetry_exporter_otlp-1.26.0-py3-none-
any.whl.metadata (2.3 kB)
Collecting opentelemetry-semantic-conventions-ai<0.5.0,>=0.4.1 (from
vllm)
  Downloading opentelemetry_semantic_conventions_ai-0.4.7-py3-none-
any.whl.metadata (1.1 kB)
Collecting numba==0.61.2 (from vllm)
  Downloading numba-0.61.2-cp311-cp311-
manylinux2014_x86_64.manylinux_2_17_x86_64.whl.metadata (2.8 kB)
Collecting ray!=2.44.*,>=2.43.0 (from ray[cgraph]!=2.44.*,>=2.43.0-
>vllm)
  Downloading ray-2.46.0-cp311-cp311-manylinux2014_x86_64.whl.metadata
(19 kB)
Requirement already satisfied: torch==2.6.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (2.6.0+cu124)
Requirement already satisfied: torchaudio==2.6.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (2.6.0+cu124)
Requirement already satisfied: torchvision==0.21.0 in
/usr/local/lib/python3.11/dist-packages (from vllm) (0.21.0+cu124)
Collecting xformers==0.0.29.post2 (from vllm)
  Downloading xformers-0.0.29.post2-cp311-cp311-
manylinux_2_28_x86_64.whl.metadata (1.0 kB)
Collecting astor (from depyf==0.18.0->vllm)
  Downloading astor-0.8.1-py2.py3-none-any.whl.metadata (4.2 kB)
Collecting dill (from depyf==0.18.0->vllm)
  Downloading dill-0.4.0-py3-none-any.whl.metadata (10 kB)
Collecting llvmlite<0.45,>=0.44.0dev0 (from numba==0.61.2->vllm)
  Downloading llvmlite-0.44.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.8 kB)
Collecting interegular (from outlines==0.1.11->vllm)
  Downloading interegular-0.3.3-py37-none-any.whl.metadata (3.0 kB)
```

```
Requirement already satisfied: jinja2 in
/usr/local/lib/python3.11/dist-packages (from outlines==0.1.11->vllm)
(3.1.6)
Requirement already satisfied: nest_asyncio in
/usr/local/lib/python3.11/dist-packages (from outlines==0.1.11->vllm)
(1.6.0)
Collecting diskcache (from outlines==0.1.11->vllm)
  Downloading diskcache-5.6.3-py3-none-any.whl.metadata (20 kB)
Requirement already satisfied: referencing in
/usr/local/lib/python3.11/dist-packages (from outlines==0.1.11->vllm)
(0.36.2)
Requirement already satisfied: jsonschema in
/usr/local/lib/python3.11/dist-packages (from outlines==0.1.11->vllm)
(4.23.0)
Collecting pycountry (from outlines==0.1.11->vllm)
  Downloading pycountry-24.6.1-py3-none-any.whl.metadata (12 kB)
Collecting airportsdata (from outlines==0.1.11->vllm)
  Downloading airportsdata-20250224-py3-none-any.whl.metadata (9.0 kB)
Collecting outlines_core==0.1.26 (from outlines==0.1.11->vllm)
  Downloading outlines_core-0.1.26-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (3.8 kB)
Requirement already satisfied: networkx in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(3.4.2)
Requirement already satisfied: fsspec in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(2025.3.2)
Collecting nvidia-cuda-nvrtc-cu12==12.4.127 (from torch==2.6.0->vllm)
  Downloading nvidia_cuda_nvrtc_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-runtime-cu12==12.4.127 (from torch==2.6.0-
>vllm)
  Downloading nvidia_cuda_runtime_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cuda-cupti-cu12==12.4.127 (from torch==2.6.0->vllm)
  Downloading nvidia_cuda_cupti_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cudnn-cu12==9.1.0.70 (from torch==2.6.0->vllm)
  Downloading nvidia_cudnn_cu12-9.1.0.70-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cublas-cu12==12.4.5.8 (from torch==2.6.0->vllm)
  Downloading nvidia_cublas_cu12-12.4.5.8-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-cufft-cu12==11.2.1.3 (from torch==2.6.0->vllm)
  Downloading nvidia_cufft_cu12-11.2.1.3-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Collecting nvidia-curand-cu12==10.3.5.147 (from torch==2.6.0->vllm)
  Downloading nvidia_curand_cu12-10.3.5.147-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
```

```
Collecting nvidia-cusolver-cu12==11.6.1.9 (from torch==2.6.0->vllm)
  Downloading nvidia_cusolver_cu12-11.6.1.9-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Collecting nvidia-cusparse-cu12==12.3.1.170 (from torch==2.6.0->vllm)
  Downloading nvidia_cusparse_cu12-12.3.1.170-py3-none-
manylinux2014_x86_64.whl.metadata (1.6 kB)
Requirement already satisfied: nvidia-cusparselt-cu12==0.6.2 in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(0.6.2)
Requirement already satisfied: nvidia-nccl-cu12==2.21.5 in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(2.21.5)
Requirement already satisfied: nvidia-nvtx-cu12==12.4.127 in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(12.4.127)
Collecting nvidia-nvjitlink-cu12==12.4.127 (from torch==2.6.0->vllm)
  Downloading nvidia_nvjitlink_cu12-12.4.127-py3-none-
manylinux2014_x86_64.whl.metadata (1.5 kB)
Requirement already satisfied: triton==3.2.0 in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(3.2.0)
Requirement already satisfied: sympy==1.13.1 in
/usr/local/lib/python3.11/dist-packages (from torch==2.6.0->vllm)
(1.13.1)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in
/usr/local/lib/python3.11/dist-packages (from sympy==1.13.1-
>torch==2.6.0->vllm) (1.3.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.11/dist-packages (from accelerate) (24.2)
Requirement already satisfied: safetensors>=0.4.3 in
/usr/local/lib/python3.11/dist-packages (from accelerate) (0.5.3)
Requirement already satisfied: regex!=2019.12.17 in
/usr/local/lib/python3.11/dist-packages (from transformers)
(2024.11.6)
Collecting starlette<0.47.0,>=0.40.0 (from fastapi>=0.115.0-
>fastapi[standard]>=0.115.0->vllm)
  Downloading starlette-0.46.2-py3-none-any.whl.metadata (6.2 kB)
Collecting fastapi-cli>=0.0.5 (from fastapi-cli[standard]>=0.0.5;
extra == "standard"->fastapi[standard]>=0.115.0->vllm)
  Downloading fastapi_cli-0.0.7-py3-none-any.whl.metadata (6.2 kB)
Requirement already satisfied: httpx>=0.23.0 in
/usr/local/lib/python3.11/dist-packages (from
fastapi[standard]>=0.115.0->vllm) (0.28.1)
Collecting python-multipart>=0.0.18 (from fastapi[standard]>=0.115.0-
>vllm)
  Downloading python_multipart-0.0.20-py3-none-any.whl.metadata (1.8
kB)
Collecting email-validator>=2.0.0 (from fastapi[standard]>=0.115.0-
>vllm)
```

```
  Downloading email_validator-2.2.0-py3-none-any.whl.metadata (25 kB)
Collecting uvicorn>=0.12.0 (from uvicorn[standard]>=0.12.0; extra ==
"standard"->fastapi[standard]>=0.115.0->vllm)
  Downloading uvicorn-0.34.2-py3-none-any.whl.metadata (6.5 kB)
Collecting hf-xet>=0.1.4 (from huggingface-hub[hf_xet]>=0.30.0->vllm)
  Downloading hf_xet-1.1.0-cp37-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (494 bytes)
Requirement already satisfied: anyio<5,>=3.5.0 in
/usr/local/lib/python3.11/dist-packages (from openai>=1.52.0->vllm)
(4.9.0)
Requirement already satisfied: distro<2,>=1.7.0 in
/usr/local/lib/python3.11/dist-packages (from openai>=1.52.0->vllm)
(1.9.0)
Requirement already satisfied: jiter<1,>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from openai>=1.52.0->vllm)
(0.9.0)
Requirement already satisfied: sniffio in
/usr/local/lib/python3.11/dist-packages (from openai>=1.52.0->vllm)
(1.3.1)
Requirement already satisfied: deprecated>=1.2.6 in
/usr/local/lib/python3.11/dist-packages (from opentelemetry-
api<1.27.0,>=1.26.0->vllm) (1.2.18)
Collecting importlib_metadata (from vllm)
  Downloading importlib_metadata-8.0.0-py3-none-any.whl.metadata (4.6
kB)
Requirement already satisfied: zipp>=0.5 in
/usr/local/lib/python3.11/dist-packages (from importlib_metadata-
>vllm) (3.21.0)
Collecting opentelemetry-exporter-otlp-proto-grpc==1.26.0 (from
opentelemetry-exporter-otlp<1.27.0,>=1.26.0->vllm)
  Downloading opentelemetry_exporter_otlp_proto_grpc-1.26.0-py3-none-
any.whl.metadata (2.3 kB)
Collecting opentelemetry-exporter-otlp-proto-http==1.26.0 (from
opentelemetry-exporter-otlp<1.27.0,>=1.26.0->vllm)
  Downloading opentelemetry_exporter_otlp_proto_http-1.26.0-py3-none-
any.whl.metadata (2.3 kB)
Requirement already satisfied: googleapis-common-protos~=1.52 in
/usr/local/lib/python3.11/dist-packages (from opentelemetry-exporter-
otlp-proto-grpc==1.26.0->opentelemetry-exporter-otlp<1.27.0,>=1.26.0-
>vllm) (1.70.0)
Requirement already satisfied: grpcio<2.0.0,>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from opentelemetry-exporter-
otlp-proto-grpc==1.26.0->opentelemetry-exporter-otlp<1.27.0,>=1.26.0-
>vllm) (1.71.0)
Collecting opentelemetry-exporter-otlp-proto-common==1.26.0 (from
opentelemetry-exporter-otlp-proto-grpc==1.26.0->opentelemetry-
exporter-otlp<1.27.0,>=1.26.0->vllm)
  Downloading opentelemetry_exporter_otlp_proto_common-1.26.0-py3-
none-any.whl.metadata (1.8 kB)
```

```
Collecting opentelemetry-proto==1.26.0 (from opentelemetry-exporter-
otlp-proto-grpc==1.26.0->opentelemetry-exporter-otlp<1.27.0,>=1.26.0-
>vllm)
  Downloading opentelemetry_proto-1.26.0-py3-none-any.whl.metadata
(2.3 kB)
Collecting protobuf (from vllm)
  Downloading protobuf-4.25.7-cp37-abi3-
manylinux2014_x86_64.whl.metadata (541 bytes)
Collecting opentelemetry-semantic-conventions==0.47b0 (from
opentelemetry-sdk<1.27.0,>=1.26.0->vllm)
  Downloading opentelemetry_semantic_conventions-0.47b0-py3-none-
any.whl.metadata (2.4 kB)
Requirement already satisfied: annotated-types>=0.6.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2.9->vllm)
(0.7.0)
Requirement already satisfied: pydantic-core==2.33.2 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2.9->vllm)
(2.33.2)
Requirement already satisfied: typing-inspection>=0.4.0 in
/usr/local/lib/python3.11/dist-packages (from pydantic>=2.9->vllm)
(0.4.0)
Requirement already satisfied: click>=7.0 in
/usr/local/lib/python3.11/dist-packages (from ray!=2.44.*,>=2.43.0-
>ray[cgraph]!=2.44.*,>=2.43.0->vllm) (8.1.8)
Requirement already satisfied: msgpack<2.0.0,>=1.0.0 in
/usr/local/lib/python3.11/dist-packages (from ray!=2.44.*,>=2.43.0-
>ray[cgraph]!=2.44.*,>=2.43.0->vllm) (1.1.0)
Requirement already satisfied: cupy-cuda12x in
/usr/local/lib/python3.11/dist-packages (from ray[cgraph]!
=2.44.*,>=2.43.0->vllm) (13.3.0)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.26.0->vllm)
(3.4.1)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.26.0->vllm)
(3.10)
Requirement already satisfied: urllib3<3,>=1.21.1 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.26.0->vllm)
(2.4.0)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.11/dist-packages (from requests>=2.26.0->vllm)
(2025.4.26)
Requirement already satisfied: aiohappyeyeballs>=2.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (2.6.1)
Requirement already satisfied: aiosignal>=1.1.2 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (1.3.2)
Requirement already satisfied: attrs>=17.3.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (25.3.0)
Requirement already satisfied: frozenlist>=1.1.1 in
```

```
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (1.6.0)
Requirement already satisfied: multidict<7.0,>=4.5 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (6.4.3)
Requirement already satisfied: propcache>=0.2.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (0.3.1)
Requirement already satisfied: yarl<2.0,>=1.17.0 in
/usr/local/lib/python3.11/dist-packages (from aiohttp->vllm) (1.20.0)
Requirement already satisfied: wrapt<2,>=1.10 in
/usr/local/lib/python3.11/dist-packages (from deprecated>=1.2.6-
>opentelemetry-api<1.27.0,>=1.26.0->vllm) (1.17.2)
Collecting dnspython>=2.0.0 (from email-validator>=2.0.0-
>fastapi[standard]>=0.115.0->vllm)
  Downloading dnspython-2.7.0-py3-none-any.whl.metadata (5.8 kB)
Requirement already satisfied: typer>=0.12.3 in
/usr/local/lib/python3.11/dist-packages (from fastapi-cli>=0.0.5-
>fastapi-cli[standard]>=0.0.5; extra == "standard"-
>fastapi[standard]>=0.115.0->vllm) (0.15.3)
Collecting rich-toolkit>=0.11.1 (from fastapi-cli>=0.0.5->fastapi-
cli[standard]>=0.0.5; extra == "standard"->fastapi[standard]>=0.115.0-
>vllm)
  Downloading rich_toolkit-0.14.5-py3-none-any.whl.metadata (999
bytes)
Requirement already satisfied: httpcore==1.* in
/usr/local/lib/python3.11/dist-packages (from httpx>=0.23.0-
>fastapi[standard]>=0.115.0->vllm) (1.0.9)
Requirement already satisfied: h11>=0.16 in
/usr/local/lib/python3.11/dist-packages (from httpcore==1.*-
>httpx>=0.23.0->fastapi[standard]>=0.115.0->vllm) (0.16.0)
Requirement already satisfied: MarkupSafe>=2.0 in
/usr/local/lib/python3.11/dist-packages (from jinja2-
>outlines==0.1.11->vllm) (3.0.2)
Requirement already satisfied: jsonschema-specifications>=2023.03.6 in
/usr/local/lib/python3.11/dist-packages (from jsonschema-
>outlines==0.1.11->vllm) (2025.4.1)
Requirement already satisfied: rpds-py>=0.7.1 in
/usr/local/lib/python3.11/dist-packages (from jsonschema-
>outlines==0.1.11->vllm) (0.24.0)
Collecting httptools>=0.6.3 (from uvicorn[standard]>=0.12.0; extra ==
"standard"->fastapi[standard]>=0.115.0->vllm)
  Downloading httptools-0.6.4-cp311-cp311-
manylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux
2014_x86_64.whl.metadata (3.6 kB)
Collecting python-dotenv>=0.13 (from uvicorn[standard]>=0.12.0; extra
== "standard"->fastapi[standard]>=0.115.0->vllm)
  Downloading python_dotenv-1.1.0-py3-none-any.whl.metadata (24 kB)
Collecting uvloop!=0.15.0,!=0.15.1,>=0.14.0 (from
uvicorn[standard]>=0.12.0; extra == "standard"-
>fastapi[standard]>=0.115.0->vllm)
  Downloading uvloop-0.21.0-cp311-cp311-
```

```
manylinux_2_17_x86_64.manylinux2014_x86_64.whl.metadata (4.9 kB)
Requirement already satisfied: websockets>=10.4 in
/usr/local/lib/python3.11/dist-packages (from
uvicorn[standard]>=0.12.0; extra == "standard"-
>fastapi[standard]>=0.115.0->vllm) (15.0.1)
Requirement already satisfied: fastrlock>=0.5 in
/usr/local/lib/python3.11/dist-packages (from cupy-cuda12x-
>ray[cgraph]!=2.44.*,>=2.43.0->vllm) (0.8.3)
Requirement already satisfied: rich>=13.7.1 in
/usr/local/lib/python3.11/dist-packages (from rich-toolkit>=0.11.1-
>fastapi-cli>=0.0.5->fastapi-cli[standard]>=0.0.5; extra ==
"standard"->fastapi[standard]>=0.115.0->vllm) (13.9.4)
Requirement already satisfied: shellingham>=1.3.0 in
/usr/local/lib/python3.11/dist-packages (from typer>=0.12.3->fastapi-
cli>=0.0.5->fastapi-cli[standard]>=0.0.5; extra == "standard"-
>fastapi[standard]>=0.115.0->vllm) (1.5.4)
Requirement already satisfied: markdown-it-py>=2.2.0 in
/usr/local/lib/python3.11/dist-packages (from rich>=13.7.1->rich-
toolkit>=0.11.1->fastapi-cli>=0.0.5->fastapi-cli[standard]>=0.0.5;
extra == "standard"->fastapi[standard]>=0.115.0->vllm) (3.0.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in
/usr/local/lib/python3.11/dist-packages (from rich>=13.7.1->rich-
toolkit>=0.11.1->fastapi-cli>=0.0.5->fastapi-cli[standard]>=0.0.5;
extra == "standard"->fastapi[standard]>=0.115.0->vllm) (2.19.1)
Requirement already satisfied: mdurl~=0.1 in
/usr/local/lib/python3.11/dist-packages (from markdown-it-py>=2.2.0-
>rich>=13.7.1->rich-toolkit>=0.11.1->fastapi-cli>=0.0.5->fastapi-
cli[standard]>=0.0.5; extra == "standard"->fastapi[standard]>=0.115.0-
>vllm) (0.1.2)
Downloading vllm-0.8.5.post1-cp38-abi3-manylinux1_x86_64.whl (326.4
MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 326.4/326.4 MB 5.4 MB/s eta
0:00:00
pressed_tensors-0.9.3-py3-none-any.whl (98 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 98.4/98.4 kB 9.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 111.0/111.0 kB 10.9 MB/s eta
0:00:00
ba-0.61.2-cp311-cp311-manylinux2014_x86_64.manylinux_2_17_x86_64.whl
(3.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.8/3.8 MB 104.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 87.6/87.6 kB 8.2 MB/s eta
0:00:00
ers-0.0.29.post2-cp311-cp311-manylinux_2_28_x86_64.whl (44.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.3/44.3 MB 18.7 MB/s eta
0:00:00
mar-0.1.18-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(4.8 MB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.8/4.8 MB 103.5 MB/s eta
0:00:00
anylinux2014_x86_64.whl (363.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 363.4/363.4 MB 4.7 MB/s eta
0:00:00
anylinux2014_x86_64.whl (13.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.8/13.8 MB 116.8 MB/s eta
0:00:00
anylinux2014_x86_64.whl (24.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 24.6/24.6 MB 94.7 MB/s eta
0:00:00
e_cu12-12.4.127-py3-none-manylinux2014_x86_64.whl (883 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 883.7/883.7 kB 59.3 MB/s eta
0:00:00
anylinux2014_x86_64.whl (664.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 664.8/664.8 MB 2.7 MB/s eta
0:00:00
anylinux2014_x86_64.whl (211.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 211.5/211.5 MB 5.3 MB/s eta
0:00:00
anylinux2014_x86_64.whl (56.3 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 56.3/56.3 MB 15.9 MB/s eta
0:00:00
anylinux2014_x86_64.whl (127.9 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 127.9/127.9 MB 7.6 MB/s eta
0:00:00
anylinux2014_x86_64.whl (207.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 207.5/207.5 MB 5.8 MB/s eta
0:00:00
anylinux2014_x86_64.whl (21.1 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.1/21.1 MB 72.4 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (343 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 343.3/343.3 kB 28.9 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 95.2/95.2 kB 8.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 94.4/94.4 kB 9.3 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (14.0 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 14.0/14.0 MB 82.8 MB/s eta
0:00:00
_format_enforcer-0.10.11-py3-none-any.whl (44 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 44.2/44.2 kB 4.1 MB/s eta
0:00:00
istral_common-1.5.4-py3-none-any.whl (6.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.5/6.5 MB 86.0 MB/s eta
0:00:00
etry_api-1.26.0-py3-none-any.whl (61 kB)
```

```
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 61.5/61.5 kB 6.0 MB/s eta
0:00:00
portlib_metadata-8.0.0-py3-none-any.whl (24 kB)
Downloading opentelemetry_exporter_otlp-1.26.0-py3-none-any.whl (7.0
kB)
Downloading opentelemetry_exporter_otlp_proto_grpc-1.26.0-py3-none-
any.whl (18 kB)
Downloading opentelemetry_exporter_otlp_proto_http-1.26.0-py3-none-
any.whl (16 kB)
Downloading opentelemetry_exporter_otlp_proto_common-1.26.0-py3-none-
any.whl (17 kB)
Downloading opentelemetry_proto-1.26.0-py3-none-any.whl (52 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 52.5/52.5 kB 5.1 MB/s eta
0:00:00
etry_sdk-1.26.0-py3-none-any.whl (109 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 109.5/109.5 kB 10.6 MB/s eta
0:00:00
etry_semantic_conventions-0.47b0-py3-none-any.whl (138 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 138.0/138.0 kB 13.0 MB/s eta
0:00:00
etry_semantic_conventions_ai-0.4.7-py3-none-any.whl (5.6 kB)
Downloading prometheus_fastapi_instrumentator-7.1.0-py3-none-any.whl
(19 kB)
Downloading protobuf-4.25.7-cp37-abi3-manylinux2014_x86_64.whl (294
kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 294.6/294.6 kB 26.1 MB/s eta
0:00:00
q-26.4.0-cp311-cp311-manylinux_2_28_x86_64.whl (862 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 862.4/862.4 kB 47.1 MB/s eta
0:00:00
anylinux2014_x86_64.whl (68.5 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 68.5/68.5 MB 16.9 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (1.2 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.2/1.2 MB 70.3 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (376 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 376.2/376.2 kB 33.2 MB/s eta
0:00:00
sgspec-0.19.0-cp311-cp311-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (210 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 210.7/210.7 kB 20.0 MB/s eta
0:00:00
anylinux_2_12_x86_64.manylinux2010_x86_64.whl (422 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 422.8/422.8 kB 39.2 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (454 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 454.8/454.8 kB 36.5 MB/s eta
0:00:00
ail_validator-2.2.0-py3-none-any.whl (33 kB)
```

```
Downloading fastapi_cli-0.0.7-py3-none-any.whl (10 kB)
Downloading hf_xet-1.1.0-cp37-abi3-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (53.6 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 53.6/53.6 MB 13.0 MB/s eta
0:00:00
lite-0.44.0-cp311-cp311-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
(42.4 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 42.4/42.4 MB 19.3 MB/s eta
0:00:00
ultipart-0.0.20-py3-none-any.whl (24 kB)
Downloading starlette-0.46.2-py3-none-any.whl (72 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 72.0/72.0 kB 4.8 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.5/62.5 kB 5.4 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 913.7/913.7 kB 64.2 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 119.7/119.7 kB 12.7 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 45.5/45.5 kB 4.3 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.3/6.3 MB 98.5 MB/s eta
0:00:00
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 313.6/313.6 kB 25.2 MB/s eta
0:00:00
anylinux_2_5_x86_64.manylinux1_x86_64.manylinux_2_17_x86_64.manylinux2
014_x86_64.whl (459 kB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 459.8/459.8 kB 41.7 MB/s eta
0:00:00
anylinux_2_17_x86_64.manylinux2014_x86_64.whl (4.0 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.0/4.0 MB 90.9 MB/s eta
0:00:00
q, python-multipart, python-json-logger, python-dotenv, pycountry,
protobuf, partial-json-parser, opentelemetry-semantic-conventions-ai,
nvidia-nvjitlink-cu12, nvidia-curand-cu12, nvidia-cufft-cu12, nvidia-
cuda-runtime-cu12, nvidia-cuda-nvrtc-cu12, nvidia-cuda-cupti-cu12,
nvidia-cublas-cu12, ninja, msgspec, llvmlite, llguidance, lark,
interegular, importlib_metadata, httptools, hf-xet, gguf, dnspython,
diskcache, dill, astor, airportsdata, watchfiles, tiktoken, starlette,
opentelemetry-proto, opentelemetry-api, nvidia-cusparse-cu12, nvidia-
cudnn-cu12, numba, email-validator, depyf, rich-toolkit, prometheus-
fastapi-instrumentator, opentelemetry-semantic-conventions,
opentelemetry-exporter-otlp-proto-common, nvidia-cusolver-cu12, lm-
format-enforcer, fastapi, ray, outlines_core, opentelemetry-sdk,
mistral_common, fastapi-cli, xgrammar, xformers, outlines,
opentelemetry-exporter-otlp-proto-http, opentelemetry-exporter-otlp-
proto-grpc, compressed-tensors, opentelemetry-exporter-otlp, vllm
  Attempting uninstall: pyzmq
    Found existing installation: pyzmq 24.0.1
```

```
    Uninstalling pyzmq-24.0.1:
      Successfully uninstalled pyzmq-24.0.1
  Attempting uninstall: protobuf
    Found existing installation: protobuf 5.29.4
    Uninstalling protobuf-5.29.4:
      Successfully uninstalled protobuf-5.29.4
  Attempting uninstall: nvidia-nvjitlink-cu12
    Found existing installation: nvidia-nvjitlink-cu12 12.5.82
    Uninstalling nvidia-nvjitlink-cu12-12.5.82:
      Successfully uninstalled nvidia-nvjitlink-cu12-12.5.82
  Attempting uninstall: nvidia-curand-cu12
    Found existing installation: nvidia-curand-cu12 10.3.6.82
    Uninstalling nvidia-curand-cu12-10.3.6.82:
      Successfully uninstalled nvidia-curand-cu12-10.3.6.82
  Attempting uninstall: nvidia-cufft-cu12
    Found existing installation: nvidia-cufft-cu12 11.2.3.61
    Uninstalling nvidia-cufft-cu12-11.2.3.61:
      Successfully uninstalled nvidia-cufft-cu12-11.2.3.61
  Attempting uninstall: nvidia-cuda-runtime-cu12
    Found existing installation: nvidia-cuda-runtime-cu12 12.5.82
    Uninstalling nvidia-cuda-runtime-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-runtime-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-nvrtc-cu12
    Found existing installation: nvidia-cuda-nvrtc-cu12 12.5.82
    Uninstalling nvidia-cuda-nvrtc-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-nvrtc-cu12-12.5.82
  Attempting uninstall: nvidia-cuda-cupti-cu12
    Found existing installation: nvidia-cuda-cupti-cu12 12.5.82
    Uninstalling nvidia-cuda-cupti-cu12-12.5.82:
      Successfully uninstalled nvidia-cuda-cupti-cu12-12.5.82
  Attempting uninstall: nvidia-cublas-cu12
    Found existing installation: nvidia-cublas-cu12 12.5.3.2
    Uninstalling nvidia-cublas-cu12-12.5.3.2:
      Successfully uninstalled nvidia-cublas-cu12-12.5.3.2
  Attempting uninstall: llvmlite
    Found existing installation: llvmlite 0.43.0
    Uninstalling llvmlite-0.43.0:
      Successfully uninstalled llvmlite-0.43.0
  Attempting uninstall: importlib_metadata
    Found existing installation: importlib_metadata 8.7.0
    Uninstalling importlib_metadata-8.7.0:
      Successfully uninstalled importlib_metadata-8.7.0
  Attempting uninstall: opentelemetry-api
    Found existing installation: opentelemetry-api 1.16.0
    Uninstalling opentelemetry-api-1.16.0:
      Successfully uninstalled opentelemetry-api-1.16.0
  Attempting uninstall: nvidia-cusparse-cu12
    Found existing installation: nvidia-cusparse-cu12 12.5.1.3
    Uninstalling nvidia-cusparse-cu12-12.5.1.3:
```

```
      Successfully uninstalled nvidia-cusparse-cu12-12.5.1.3
  Attempting uninstall: nvidia-cudnn-cu12
    Found existing installation: nvidia-cudnn-cu12 9.3.0.75
    Uninstalling nvidia-cudnn-cu12-9.3.0.75:
      Successfully uninstalled nvidia-cudnn-cu12-9.3.0.75
  Attempting uninstall: numba
    Found existing installation: numba 0.60.0
    Uninstalling numba-0.60.0:
      Successfully uninstalled numba-0.60.0
  Attempting uninstall: opentelemetry-semantic-conventions
    Found existing installation: opentelemetry-semantic-conventions
0.37b0
    Uninstalling opentelemetry-semantic-conventions-0.37b0:
      Successfully uninstalled opentelemetry-semantic-conventions-
0.37b0
  Attempting uninstall: nvidia-cusolver-cu12
    Found existing installation: nvidia-cusolver-cu12 11.6.3.83
    Uninstalling nvidia-cusolver-cu12-11.6.3.83:
      Successfully uninstalled nvidia-cusolver-cu12-11.6.3.83
  Attempting uninstall: opentelemetry-sdk
    Found existing installation: opentelemetry-sdk 1.16.0
    Uninstalling opentelemetry-sdk-1.16.0:
      Successfully uninstalled opentelemetry-sdk-1.16.0
ERROR: pip's dependency resolver does not currently take into account
all the packages that are installed. This behaviour is the source of
the following dependency conflicts.
dask-cuda 25.2.0 requires numba<0.61.0a0,>=0.59.1, but you have numba
0.61.2 which is incompatible.
grpcio-status 1.71.0 requires protobuf<6.0dev,>=5.26.1, but you have
protobuf 4.25.7 which is incompatible.
cuml-cu12 25.2.1 requires numba<0.61.0a0,>=0.59.1, but you have numba
0.61.2 which is incompatible.
distributed-ucxx-cu12 0.42.0 requires numba<0.61.0a0,>=0.59.1, but you
have numba 0.61.2 which is incompatible.
cudf-cu12 25.2.1 requires numba<0.61.0a0,>=0.59.1, but you have numba
0.61.2 which is incompatible.
ydf 0.11.0 requires protobuf<6.0.0,>=5.29.1, but you have protobuf
4.25.7 which is incompatible.
Successfully installed airportsdata-20250224 astor-0.8.1 blake3-1.0.4
compressed-tensors-0.9.3 depyf-0.18.0 dill-0.4.0 diskcache-5.6.3
dnspython-2.7.0 email-validator-2.2.0 fastapi-0.115.12 fastapi-cli-
0.0.7 gguf-0.16.3 hf-xet-1.1.0 httptools-0.6.4 importlib_metadata-
8.0.0 interegular-0.3.3 lark-1.2.2 llguidance-0.7.19 llvmlite-0.44.0
lm-format-enforcer-0.10.11 mistral_common-1.5.4 msgspec-0.19.0 ninja-
1.11.1.4 numba-0.61.2 nvidia-cublas-cu12-12.4.5.8 nvidia-cuda-cupti-
cu12-12.4.127 nvidia-cuda-nvrtc-cu12-12.4.127 nvidia-cuda-runtime-
cu12-12.4.127 nvidia-cudnn-cu12-9.1.0.70 nvidia-cufft-cu12-11.2.1.3
nvidia-curand-cu12-10.3.5.147 nvidia-cusolver-cu12-11.6.1.9 nvidia-
cusparse-cu12-12.3.1.170 nvidia-nvjitlink-cu12-12.4.127 opentelemetry-
```

```
api-1.26.0 opentelemetry-exporter-otlp-1.26.0 opentelemetry-exporter-otlp-proto-common-1.26.0 opentelemetry-exporter-otlp-proto-grpc-1.26.0 opentelemetry-exporter-otlp-proto-http-1.26.0 opentelemetry-proto-1.26.0 opentelemetry-sdk-1.26.0 opentelemetry-semantic-conventions-0.47b0 opentelemetry-semantic-conventions-ai-0.4.7 outlines-0.1.11 outlines_core-0.1.26 partial-json-parser-0.2.1.1.post5 prometheus-fastapi-instrumentator-7.1.0 protobuf-4.25.7 pycountry-24.6.1 python-dotenv-1.1.0 python-json-logger-3.3.0 python-multipart-0.0.20 pyzmq-26.4.0 ray-2.46.0 rich-toolkit-0.14.5 starlette-0.46.2 tiktoken-0.9.0 uvicorn-0.34.2 uvloop-0.21.0 vllm-0.8.5.post1 watchfiles-1.0.5 xformers-0.0.29.post2 xgrammar-0.1.18
```

{"id":"d1e03f70deed4dd78644ddfb33d136f0","pip_warning":{"packages":["importlib_metadata"]}}

```
from vllm import LLM, SamplingParams #used to control how model generates output
import time
```

```
INFO 05-08 08:39:04 [__init__.py:239] Automatically detected platform cuda.
```

```
def run_inference(model,dtype="float32",quantization=None,tensor_parallel_size=1,prompt="Tell me a joke"):

  model=LLM(model=model,dtype=dtype,quantization=quantization)


sampling_params=SamplingParams(temperature=0.7,top_p=0.9,max_tokens=64)

  start_time=time.time()
  results=model.generate(prompt,sampling_params=sampling_params)
  end_time=time.time()

  latency=end_time-start_time
  ans=results[0].outputs[0].text.strip()
  tokens=len(ans.split())
  throughput=tokens/latency

  print("Output:", ans)
  print(f"Latency: {latency:.2f} sec | Throughput: {throughput:.2f} tokens/sec")

  return latency, throughput
```

Run baseline inference

Baseline inference on `distilgpt2` using vLLM (non-quantized, single prompt) achieved 0.36s latency and 140.71 tokens/sec throughput on a T4 GPU using the XFormers backend.

```
run_inference("distilgpt2")
```

```
/usr/local/lib/python3.11/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://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
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 datasets.
  warnings.warn(
```


```
{"model_id":"9ca28e29487f4498943086f10b5335fa","version_major":2,"vers
ion_minor":0}
```


```
INFO 05-08 05:36:48 [config.py:717] This model supports multiple
tasks: {'classify', 'reward', 'embed', 'generate', 'score'}.
Defaulting to 'generate'.
WARNING 05-08 05:36:48 [arg_utils.py:1658] Compute Capability < 8.0 is
not supported by the V1 Engine. Falling back to V0.
INFO 05-08 05:36:48 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
```

{"model_id":"cd82f7ad6449416b8af65ddbb2332d90","version_major":2,"version_minor":0}

{"model_id":"aa4c73a9efa64cdc8d22499ba6c99f3e","version_major":2,"version_minor":0}

{"model_id":"844ee3982c364faa82dc36b0ddd1b562","version_major":2,"version_minor":0}

{"model_id":"573c3a785d074ce69c7290b5a1a9172c","version_major":2,"version_minor":0}

{"model_id":"8cfc8b4c3fdd4f26adceb7f58fc506ea","version_major":2,"version_minor":0}

```
INFO 05-08 05:36:52 [cuda.py:240] Cannot use FlashAttention-2 backend
for Volta and Turing GPUs.
INFO 05-08 05:36:52 [cuda.py:289] Using XFormers backend.
INFO 05-08 05:36:53 [parallel_state.py:1004] rank 0 in world size 1 is
assigned as DP rank 0, PP rank 0, TP rank 0
INFO 05-08 05:36:53 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 05:36:53 [weight_utils.py:265] Using model weights format
['*.safetensors']
```

{"model_id":"0c9857d6c5044336a90f55e174c47add","version_major":2,"version_minor":0}

```
INFO 05-08 05:36:57 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 4.119778 seconds
INFO 05-08 05:36:57 [weight_utils.py:315] No
model.safetensors.index.json found in remote.
```

{"model_id":"5f5ad188cc79472b95dbc2d0894f0f56","version_major":2,"version_minor":0}

```
INFO 05-08 05:36:58 [loader.py:458] Loading weights took 0.25 seconds
INFO 05-08 05:36:58 [model_runner.py:1140] Model loading took 0.3059
GiB and 4.933980 seconds
INFO 05-08 05:37:00 [worker.py:287] Memory profiling takes 1.01
seconds
INFO 05-08 05:37:00 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:37:00 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.03GiB; PyTorch activation peak memory takes
0.50GiB; the rest of the memory reserved for KV Cache is 12.44GiB.
INFO 05-08 05:37:00 [executor_base.py:112] # cuda blocks: 22644, # CPU
blocks: 7281
INFO 05-08 05:37:00 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 353.81x
INFO 05-08 05:37:04 [model_runner.py:1450] Capturing cudagraphs for
```

```
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"48095ccedc1f45258ebf9f6b896f70de","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:37:35 [model_runner.py:1592] Graph capturing finished in
31 secs, took 0.10 GiB
INFO 05-08 05:37:35 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 37.32 seconds

{"model_id":"fdf647520a0046619f9b2fe0713ec379","version_major":2,"vers
ion_minor":0}

Output: , but I've been wrong about it since the beginning.


I hate to tell you that my last words have been a joke. But it's about
the actual fact that I've been wrong about it since the beginning.
I've been wrong about it since the beginning.
I can't help but
Latency: 0.36 sec | Throughput: 140.71 tokens/sec

(0.35533833503723145, 140.71096493082044)
```

Simulate Dynamic batching in non qauntized model

```
prompts = ["Tell me a joke.", "What is AI?", "Explain quantum
computing.", "Give a fun fact.", "What is the capital of Peru?"]
for prompt in prompts:
    run_inference("distilgpt2", prompt=prompt)

/usr/local/lib/python3.11/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://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
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 datasets.
  warnings.warn(

{"model_id":"2116d25fddb14cdf99c8dbe18a409e92","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-08 05:55:40 [config.py:717] This model supports multiple
tasks: {'generate', 'embed', 'score', 'reward', 'classify'}.
Defaulting to 'generate'.
WARNING 05-08 05:55:40 [arg_utils.py:1658] Compute Capability < 8.0 is
not supported by the V1 Engine. Falling back to V0.
INFO 05-08 05:55:40 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,

{"model_id":"8bbc6ce82b494d039bb1ea1631c991e3","version_major":2,"vers
ion_minor":0}

{"model_id":"823ea730101745e681bfc4bb968f49ca","version_major":2,"vers
ion_minor":0}

{"model_id":"e11cda9ab0a14cffa3a097f7f67f856e","version_major":2,"vers
ion_minor":0}

{"model_id":"739a2b6c5ef94f15bf0f1f99ce843f92","version_major":2,"vers
ion_minor":0}

{"model_id":"7edbde4af04b4cf69a7a2b1fbc422909","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:55:43 [cuda.py:240] Cannot use FlashAttention-2 backend
for Volta and Turing GPUs.
INFO 05-08 05:55:43 [cuda.py:289] Using XFormers backend.
INFO 05-08 05:55:44 [parallel_state.py:1004] rank 0 in world size 1 is
assigned as DP rank 0, PP rank 0, TP rank 0
INFO 05-08 05:55:44 [model_runner.py:1108] Starting to load model
```

```
distilgpt2...
INFO 05-08 05:55:44 [weight_utils.py:265] Using model weights format
['*.safetensors']

{"model_id":"399c207cf39841f89b8f936988092192","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:55:52 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 8.168744 seconds
INFO 05-08 05:55:52 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"7e0105e75409498189fb094e948706d3","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:55:53 [loader.py:458] Loading weights took 0.26 seconds
INFO 05-08 05:55:53 [model_runner.py:1140] Model loading took 0.3059
GiB and 8.902724 seconds
INFO 05-08 05:55:55 [worker.py:287] Memory profiling takes 1.00
seconds
INFO 05-08 05:55:55 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:55:55 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.03GiB; PyTorch activation peak memory takes
0.50GiB; the rest of the memory reserved for KV Cache is 12.44GiB.
INFO 05-08 05:55:55 [executor_base.py:112] # cuda blocks: 22644, # CPU
blocks: 7281
INFO 05-08 05:55:55 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 353.81x
INFO 05-08 05:55:59 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"b72aaaf5a42f4deeba8a88904be59ff6","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:56:30 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.10 GiB
INFO 05-08 05:56:30 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 37.16 seconds

{"model_id":"bd348f7fe1f94362b4201e8a38efe2ee","version_major":2,"vers
ion_minor":0}

Output: I don't know what to do, but I'm just trying to make the right
decision. I think there are a lot of things that can go wrong.
Latency: 0.39 sec | Throughput: 69.68 tokens/sec
```

```
INFO 05-08 05:56:31 [config.py:717] This model supports multiple
tasks: {'generate', 'embed', 'score', 'reward', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 05:56:31 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 05:56:32 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 05:56:32 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 05:56:32 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"1ae47bc8fe49407c9aa6a2c6e7690eb4","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:56:33 [loader.py:458] Loading weights took 0.24 seconds
INFO 05-08 05:56:33 [model_runner.py:1140] Model loading took 0.3059
GiB and 0.532313 seconds
INFO 05-08 05:56:34 [worker.py:287] Memory profiling takes 0.48
seconds
INFO 05-08 05:56:34 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:56:34 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 05:56:34 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 05:56:34 [executor_base.py:117] Maximum concurrency for
```

```
1024 tokens per request: 354.77x
INFO 05-08 05:56:35 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"2e5e4ad650bd4d2fbf330795dd37b33c","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:57:07 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.05 GiB
INFO 05-08 05:57:07 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 33.71 seconds

{"model_id":"fea146ac8e78473e9970c2833269d289","version_major":2,"vers
ion_minor":0}

Output: How does AI work?




The AI is not simply an abstract abstraction. It is a process that is
constantly changing. A process that is constantly changing, often in a
way that is always changing.
In the end, we have to make sure that the process is continuously
changing.
The goal
Latency: 0.30 sec | Throughput: 166.05 tokens/sec
INFO 05-08 05:57:08 [config.py:717] This model supports multiple
tasks: {'generate', 'embed', 'score', 'reward', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 05:57:08 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
```

```
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 05:57:09 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 05:57:09 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 05:57:09 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"509d5c9a41a74701996a7943de83c097","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:57:09 [loader.py:458] Loading weights took 0.24 seconds
INFO 05-08 05:57:10 [model_runner.py:1140] Model loading took 0.3059
GiB and 0.520857 seconds
INFO 05-08 05:57:10 [worker.py:287] Memory profiling takes 0.49
seconds
INFO 05-08 05:57:10 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:57:10 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 05:57:11 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 05:57:11 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 05:57:11 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"603b221e4952424c845b96dd6ea73d15","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:57:42 [model_runner.py:1592] Graph capturing finished in
31 secs, took 0.05 GiB
INFO 05-08 05:57:42 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 32.89 seconds

{"model_id":"c2dd198d0fb4463c885fb9abc7a32d5b","version_major":2,"vers
ion_minor":0}
```

```
Output:
Latency: 0.32 sec | Throughput: 0.00 tokens/sec
INFO 05-08 05:57:43 [config.py:717] This model supports multiple
tasks: {'generate', 'embed', 'score', 'reward', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 05:57:43 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 05:57:44 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 05:57:44 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 05:57:45 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"0634300ece794fbeb606849957bb07ac","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:57:45 [loader.py:458] Loading weights took 0.26 seconds
INFO 05-08 05:57:45 [model_runner.py:1140] Model loading took 0.3059
GiB and 0.534627 seconds
INFO 05-08 05:57:46 [worker.py:287] Memory profiling takes 0.49
seconds
INFO 05-08 05:57:46 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:57:46 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 05:57:47 [executor_base.py:112] # cuda blocks: 22705, # CPU
```

```
blocks: 7281
INFO 05-08 05:57:47 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 05:57:47 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"a588639d82b1490db980b3eeae5d83ed","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:58:18 [model_runner.py:1592] Graph capturing finished in
31 secs, took 0.05 GiB
INFO 05-08 05:58:18 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 32.51 seconds

{"model_id":"342caa806a834fd6a8adb3f13a871097","version_major":2,"vers
ion_minor":0}

Output: ”
Latency: 0.07 sec | Throughput: 13.43 tokens/sec
INFO 05-08 05:58:18 [config.py:717] This model supports multiple
tasks: {'generate', 'embed', 'score', 'reward', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 05:58:18 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
```

```
":256}, use_cached_outputs=False,
INFO 05-08 05:58:19 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 05:58:19 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 05:58:20 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"7ce8f9560f164fa5b46a81dcaed9ea7a","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:58:20 [loader.py:458] Loading weights took 0.26 seconds
INFO 05-08 05:58:20 [model_runner.py:1140] Model loading took 0.3059
GiB and 0.556125 seconds
INFO 05-08 05:58:21 [worker.py:287] Memory profiling takes 0.48
seconds
INFO 05-08 05:58:21 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 05:58:21 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 05:58:22 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 05:58:22 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 05:58:22 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"5491488d4e5940849994834a50c3b61c","version_major":2,"vers
ion_minor":0}

INFO 05-08 05:58:53 [model_runner.py:1592] Graph capturing finished in
31 secs, took 0.05 GiB
INFO 05-08 05:58:53 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 32.62 seconds

{"model_id":"55e6727c9bd649289ea92fdedbc2b673","version_major":2,"vers
ion_minor":0}

Output: The question is whether Peru has its own way of making things
better. It is not a natural land. But it is a significant land for the
people of the country. There are no human rights groups in Peru that
are more interested in the rights of indigenous people than the people
of the country.
As a
Latency: 0.30 sec | Throughput: 190.38 tokens/sec
```

Test with diff batch sizes

```
import time

def run_inference_with_batch_size(model, dtype="float32",
quantization=None, tensor_parallel_size=1, prompts=None,
batch_size=1):
    """
    Run inference with dynamic batching, where the number of prompts
processed in a single batch is specified.

    :param model: The model to be used for inference
    :param dtype: Data type for inference (e.g., 'float32', 'float16')
    :param quantization: Quantization type (e.g., 'int8', 'none')
    :param tensor_parallel_size: Size of tensor parallelism for
distributed execution
    :param prompts: List of prompts to be used for inference
    :param batch_size: Number of prompts to batch together for each
inference pass
    :return: Latency and throughput for the batch size
    """

    # Ensure prompts is a list
    if not isinstance(prompts, list):
        prompts = [prompts]  # Convert single prompt to a list for
consistency

    # Clip the list of prompts to the batch size if needed
    prompts = prompts[:batch_size]

    model = LLM(model=model, dtype=dtype, quantization=quantization)

    sampling_params = SamplingParams(temperature=0.7, top_p=0.9,
max_tokens=64)

    start_time = time.time()
    results = model.generate(prompts, sampling_params=sampling_params)
    end_time = time.time()

    latency = end_time - start_time
    total_tokens = sum([len(result.outputs[0].text.strip().split())
for result in results])
    throughput = total_tokens / latency

    # Display results for each prompt in the batch
    for idx, result in enumerate(results):
        print(f"Prompt {idx+1}: {prompts[idx]}")
        print("Output:", result.outputs[0].text.strip())

    print(f"Latency: {latency:.2f} sec | Throughput: {throughput:.2f}
tokens/sec")
```

```
    return latency, throughput


# Example Usage: Testing with Different Batch Sizes
batch_sizes = [1, 2, 4, 8, 16]
prompts = [
    "Tell me a joke",
    "What is the capital of Laos?",
    "Describe the process of photosynthesis",
    "How does gravity work?",
    "What is the meaning of life?"
]

for batch_size in batch_sizes:
    print(f"\nTesting with batch size: {batch_size}")
    latency, throughput = run_inference_with_batch_size(
        model="distilgpt2",  # Model can be changed
        dtype="float32",  # You can experiment with other data types
as well
        quantization=None,  # Quantization options if needed
        tensor_parallel_size=1,
        prompts=prompts,
        batch_size=batch_size
    )
    print(f"Batch size: {batch_size} - Latency: {latency:.2f}s |
Throughput: {throughput:.2f} tokens/sec\n")


Testing with batch size: 1

{"model_id":"cdd1a9adc6ff4d7ca5d44c2dca3364fe","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:22:21 [config.py:717] This model supports multiple
tasks: {'embed', 'reward', 'score', 'generate', 'classify'}.
Defaulting to 'generate'.
WARNING 05-08 06:22:21 [arg_utils.py:1658] Compute Capability < 8.0 is
not supported by the V1 Engine. Falling back to V0.
INFO 05-08 06:22:21 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
```

```
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
```

```
{"model_id":"1d187a0350ee41bea19746611c6d8e8c","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"6eba085f9bc545b6b237b750cd5b901e","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"e1040495df164275a1237252a6e7f354","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"a40308dff47e45e2a180cd3d1c0e3727","version_major":2,"vers
ion_minor":0}
```

```
{"model_id":"de918e7ce35a45ceae19ac593f1e31e6","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-08 06:22:27 [cuda.py:240] Cannot use FlashAttention-2 backend
for Volta and Turing GPUs.
INFO 05-08 06:22:27 [cuda.py:289] Using XFormers backend.
INFO 05-08 06:22:28 [parallel_state.py:1004] rank 0 in world size 1 is
assigned as DP rank 0, PP rank 0, TP rank 0
INFO 05-08 06:22:28 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 06:22:29 [weight_utils.py:265] Using model weights format
['*.safetensors']
```

```
{"model_id":"794f6ad7426e4511bcd7d90a523b01af","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-08 06:22:38 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 8.481588 seconds
INFO 05-08 06:22:38 [weight_utils.py:315] No
model.safetensors.index.json found in remote.
```

```
{"model_id":"5f7a9e3d409f4170901f07975b913d0f","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-08 06:22:39 [loader.py:458] Loading weights took 0.35 seconds
INFO 05-08 06:22:39 [model_runner.py:1140] Model loading took 0.3059
GiB and 10.684806 seconds
INFO 05-08 06:22:41 [worker.py:287] Memory profiling takes 1.15
seconds
INFO 05-08 06:22:41 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 06:22:41 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.03GiB; PyTorch activation peak memory takes
0.50GiB; the rest of the memory reserved for KV Cache is 12.44GiB.
INFO 05-08 06:22:41 [executor_base.py:112] # cuda blocks: 22644, # CPU
blocks: 7281
INFO 05-08 06:22:41 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 353.81x
INFO 05-08 06:22:46 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"01cb2a3b73db4601a4b1d25be9185424","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:23:17 [model_runner.py:1592] Graph capturing finished in
31 secs, took 0.10 GiB
INFO 05-08 06:23:17 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 38.27 seconds

{"model_id":"5f68431ddc6e4b10b8a710999f41c17f","version_major":2,"vers
ion_minor":0}

Prompt 1: Tell me a joke
Output: .”

“It’s not that I’m not a real writer,”
“I’m not a writer,”
“I’m not a writer,”
“I’m not a writer,”
Latency: 0.58 sec | Throughput: 36.07 tokens/sec
Batch size: 1 - Latency: 0.58s | Throughput: 36.07 tokens/sec


Testing with batch size: 2
INFO 05-08 06:23:19 [config.py:717] This model supports multiple
tasks: {'embed', 'reward', 'score', 'generate', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 06:23:19 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
```

```
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 06:23:20 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 06:23:20 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 06:23:21 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 0.555021 seconds
INFO 05-08 06:23:21 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"3efc20cf1efd483bbc237ed34f11ae43","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:23:21 [loader.py:458] Loading weights took 0.23 seconds
INFO 05-08 06:23:22 [model_runner.py:1140] Model loading took 0.3059
GiB and 1.058507 seconds
INFO 05-08 06:23:23 [worker.py:287] Memory profiling takes 0.50
seconds
INFO 05-08 06:23:23 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 06:23:23 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 06:23:23 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 06:23:23 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 06:23:24 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
```

```
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"7226e40e3c2945eca87529f14415e173","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:23:55 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.05 GiB
INFO 05-08 06:23:55 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 33.75 seconds

{"model_id":"160e7b3065af4839bbf782d263d0b8e3","version_major":2,"vers
ion_minor":0}

Prompt 1: Tell me a joke
Output: and a little bit of fun to do in the beginning.
“I’ve been in the league since I was 13,“ and I’m an adult. I’m a
young boy. I’m a young man. I’m a young man. I
Prompt 2: What is the capital of Laos?
Output:
Latency: 0.44 sec | Throughput: 83.71 tokens/sec
Batch size: 2 - Latency: 0.44s | Throughput: 83.71 tokens/sec


Testing with batch size: 4
INFO 05-08 06:23:57 [config.py:717] This model supports multiple
tasks: {'embed', 'reward', 'score', 'generate', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 06:23:57 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
```

```
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 06:23:58 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 06:23:58 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 06:23:59 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 0.566606 seconds
INFO 05-08 06:23:59 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"e03bf2a74c6746ddb8ea9d2cb86acd80","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:23:59 [loader.py:458] Loading weights took 0.26 seconds
INFO 05-08 06:24:00 [model_runner.py:1140] Model loading took 0.3059
GiB and 1.225098 seconds
INFO 05-08 06:24:01 [worker.py:287] Memory profiling takes 0.50
seconds
INFO 05-08 06:24:01 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 06:24:01 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 06:24:01 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 06:24:01 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 06:24:02 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"c77bca4ab3a248c590239b38631bc63e","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:24:34 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.05 GiB
INFO 05-08 06:24:34 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 33.93 seconds

{"model_id":"3191bc74691d4d0abbc4aa0cbd0b33aa","version_major":2,"vers
ion_minor":0}
```

```
Prompt 1: Tell me a joke
Output: or something.

"Oh, yeah. That's the best way to be honest with you. I'm so fucking sick of it. I'm so fucking sick of it. I'm so fucking sick of it. I'm so fucking sick of it. I'm so fucking sick of it. I'm so
Prompt 2: What is the capital of Laos?
Output:
Prompt 3: Describe the process of photosynthesis
Output: .
Prompt 4: How does gravity work?
Output:
Latency: 0.50 sec | Throughput: 92.36 tokens/sec
Batch size: 4 - Latency: 0.50s | Throughput: 92.36 tokens/sec


Testing with batch size: 8
INFO 05-08 06:24:35 [config.py:717] This model supports multiple
tasks: {'embed', 'reward', 'score', 'generate', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 06:24:35 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 06:24:36 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 06:24:36 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 06:24:38 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 1.086657 seconds
```

```
INFO 05-08 06:24:38 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"7743e5db2264498fb1e11ac0e219464f","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:24:38 [loader.py:458] Loading weights took 0.24 seconds
INFO 05-08 06:24:38 [model_runner.py:1140] Model loading took 0.3059
GiB and 1.600541 seconds
INFO 05-08 06:24:40 [worker.py:287] Memory profiling takes 0.57
seconds
INFO 05-08 06:24:40 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 06:24:40 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 06:24:40 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 06:24:40 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 06:24:41 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"8fe2cea6f8d94c1485fd8e02dceea3da","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:25:13 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.05 GiB
INFO 05-08 06:25:13 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 34.38 seconds

{"model_id":"b8ea723b81e24a9b8ea0481ae75e0b25","version_major":2,"vers
ion_minor":0}

Prompt 1: Tell me a joke
Output: .
“This is just an example of how I've been bullied and bullied.“
“I don't want to be an exception. I'm a person who has to understand
the impact of the discrimination and harassment that happens to me.
“I want to be an example of how I
Prompt 2: What is the capital of Laos?
Output:
Prompt 3: Describe the process of photosynthesis
Output: .
```

```
The photosynthesis process has been done for the last 100 years.
The process is described by the following important mathematical
terms:
The number of photosynthetic photosynthetic photosynthesis is
determined by the number of photosynthetic photosynthetic
photosynthetic photosynthesis is determined by the
Prompt 4: How does gravity work?
Output:
Prompt 5: What is the meaning of life?
Output:
Latency: 0.51 sec | Throughput: 174.88 tokens/sec
Batch size: 8 - Latency: 0.51s | Throughput: 174.88 tokens/sec


Testing with batch size: 16
INFO 05-08 06:25:17 [config.py:717] This model supports multiple
tasks: {'embed', 'reward', 'score', 'generate', 'classify'}.
Defaulting to 'generate'.
INFO 05-08 06:25:17 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='distilgpt2',
speculative_config=None, tokenizer='distilgpt2',
skip_tokenizer_init=False, tokenizer_mode=auto, revision=None,
override_neuron_config=None, tokenizer_revision=None,
trust_remote_code=False, dtype=torch.float32, max_seq_len=1024,
download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=None,
enforce_eager=False, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=distilgpt2, num_scheduler_steps=1,
multi_step_stream_outputs=True, enable_prefix_caching=None,
chunked_prefill_enabled=False, use_async_output_proc=True,
disable_mm_preprocessor_cache=False, mm_processor_kwargs=None,
pooler_config=None, compilation_config={"splitting_ops":
[],"compile_sizes":[],"cudagraph_capture_sizes":
[256,248,240,232,224,216,208,200,192,184,176,168,160,152,144,136,128,1
20,112,104,96,88,80,72,64,56,48,40,32,24,16,8,4,2,1],"max_capture_size
":256}, use_cached_outputs=False,
INFO 05-08 06:25:18 [model_runner.py:1108] Starting to load model
distilgpt2...
INFO 05-08 06:25:18 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 06:25:19 [weight_utils.py:281] Time spent downloading
weights for distilgpt2: 0.516440 seconds
```

```
INFO 05-08 06:25:19 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"c21f58ae0ab841b4a20e28d6ef5528dd","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:25:19 [loader.py:458] Loading weights took 0.23 seconds
INFO 05-08 06:25:19 [model_runner.py:1140] Model loading took 0.3059
GiB and 1.020305 seconds
INFO 05-08 06:25:20 [worker.py:287] Memory profiling takes 0.52
seconds
INFO 05-08 06:25:20 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 06:25:20 [worker.py:287] model weights take 0.31GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.49GiB; the rest of the memory reserved for KV Cache is 12.47GiB.
INFO 05-08 06:25:21 [executor_base.py:112] # cuda blocks: 22705, # CPU
blocks: 7281
INFO 05-08 06:25:21 [executor_base.py:117] Maximum concurrency for
1024 tokens per request: 354.77x
INFO 05-08 06:25:21 [model_runner.py:1450] Capturing cudagraphs for
decoding. This may lead to unexpected consequences if the model is not
static. To run the model in eager mode, set 'enforce_eager=True' or
use '--enforce-eager' in the CLI. If out-of-memory error occurs during
cudagraph capture, consider decreasing `gpu_memory_utilization` or
switching to eager mode. You can also reduce the `max_num_seqs` as
needed to decrease memory usage.

{"model_id":"efcfddfdaf9a41f5a847ff85ba6fb13e","version_major":2,"vers
ion_minor":0}

INFO 05-08 06:25:53 [model_runner.py:1592] Graph capturing finished in
32 secs, took 0.05 GiB
INFO 05-08 06:25:53 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 34.02 seconds

{"model_id":"b27f371a460a4b699447639c18df35db","version_major":2,"vers
ion_minor":0}

Prompt 1: Tell me a joke
Output: .

I was in the bathroom in the bathroom with my friend and we were going
to be in the bathroom with my friend and she was a bit worried, so she
put her hand up and said, 'What do you think?' I said, 'That's fine,
it's fine.'
I said,
Prompt 2: What is the capital of Laos?
Output:
Prompt 3: Describe the process of photosynthesis
Output: , which in turn takes place in the soil. The photosynthetic
```

```
process is the process of producing oxygen, which is what the plants
do. The process of making food and consuming it in the soil is a
process of making the food and consuming it in the soil.



What is the process of making
Prompt 4: How does gravity work?
Output:
Prompt 5: What is the meaning of life?
Output: I am a human being.
I am not a biological person.
I am not a non-human being.
I am a human being.
I am not a person.
I am not a human being.
I am not a non-human being.
I am not a
Latency: 0.70 sec | Throughput: 208.97 tokens/sec
Batch size: 16 - Latency: 0.70s | Throughput: 208.97 tokens/sec
```

LLama with gpt q and dynamic batching

```python
import time
import logging
import csv
from vllm import LLM, SamplingParams

# Set up logging
logging.basicConfig(level=logging.INFO)
logger = logging.getLogger(__name__)

def run_inference_with_batch_size(model, dtype="float16",
quantization="gptq", tensor_parallel_size=1, prompts=None,
batch_size=1):
    try:
        if not isinstance(prompts, list):
            prompts = [prompts]

        prompts = prompts[:batch_size]

        logger.info(f"Initializing model: {model} | Batch size:
{batch_size}, Quantization: {quantization}")
        llm = LLM(
            model=model,
            dtype=dtype,
            quantization=quantization,
            tensor_parallel_size=tensor_parallel_size,
            max_num_seqs=batch_size,
```

```
            enforce_eager=True
        )

        sampling_params = SamplingParams(temperature=0.7, top_p=0.9,
max_tokens=256)

        start_time = time.time()
        results = llm.generate(prompts,
sampling_params=sampling_params)
        end_time = time.time()

        latency = end_time - start_time
        total_tokens =
sum([len(result.outputs[0].text.strip().split()) for result in
results])
        throughput = total_tokens / latency if latency > 0 else 0

        print(f"Batch size: {batch_size} | Latency: {latency:.2f} sec
| Throughput: {throughput:.2f} tokens/sec")

        return latency, throughput

    except Exception as e:
        logger.error(f"Error during inference: {str(e)}")
        raise

# 16 diverse prompts
prompts = [
    "Tell me a joke.",
    "What is the capital of France?",
    "Explain the theory of relativity.",
    "Who discovered penicillin?",
    "Describe how photosynthesis works.",
    "What causes rainbows?",
    "What is quantum computing?",
    "Write a short poem about time.",
    "What's the future of space exploration?",
    "What are black holes?",
    "Explain string theory in simple terms.",
    "Tell me about machine learning.",
    "How do airplanes fly?",
    "What is the purpose of dreams?",
    "What are the laws of thermodynamics?",
    "Describe the process of human digestion."
]

# Output CSV file
csv_filename = "llama3_gptq_batch_results.csv"
csv_headers = ["Batch Size", "Latency (s)", "Throughput (tokens/s)"]
```

```
with open(csv_filename, mode="w", newline="") as csvfile:
    writer = csv.writer(csvfile)
    writer.writerow(csv_headers)

    print("🚀 Benchmarking LLaMA 3-8B GPTQ with batch sizes 1 to 16\n")
    for batch_size in [1, 2, 4, 8, 16]:
        latency, throughput = run_inference_with_batch_size(
            model="astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit",
            dtype="float16",
            quantization="gptq",
            tensor_parallel_size=1,
            prompts=prompts,
            batch_size=batch_size
        )
        # Write results to CSV
        writer.writerow([batch_size, round(latency, 2),
round(throughput, 2)])

print(f"\n✅ Benchmark complete. Results saved to {csv_filename}")
```

```
🚀 Benchmarking LLaMA 3-8B GPTQ with batch sizes 1 to 16


/usr/local/lib/python3.11/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://huggingface.co/settings/tokens), set it as
secret in your Google Colab and restart your session.
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 datasets.
  warnings.warn(

{"model_id":"91ebb83c17cc4b828358d6ab62f9e752","version_major":2,"vers
ion_minor":0}

WARNING 05-08 08:39:41 [config.py:2972] Casting torch.bfloat16 to
torch.float16.
INFO 05-08 08:40:01 [config.py:717] This model supports multiple
tasks: {'generate', 'score', 'reward', 'classify', 'embed'}.
Defaulting to 'generate'.
INFO 05-08 08:40:03 [gptq_bitblas.py:168] Detected that the model can
run with gptq_bitblas, however you specified quantization=gptq
explicitly, so forcing gptq. Use quantization=gptq_bitblas for faster
inference
WARNING 05-08 08:40:03 [config.py:830] gptq quantization is not fully
optimized yet. The speed can be slower than non-quantized models.
WARNING 05-08 08:40:03 [arg_utils.py:1658] Compute Capability < 8.0 is
not supported by the V1 Engine. Falling back to V0.
```

```
WARNING 05-08 08:40:03 [cuda.py:93] To see benefits of async output
processing, enable CUDA graph. Since, enforce-eager is enabled, async
output processor cannot be used
INFO 05-08 08:40:03 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='astronomer/Llama-3-8B-Instruct-
GPTQ-4-Bit', speculative_config=None, tokenizer='astronomer/Llama-3-
8B-Instruct-GPTQ-4-Bit', skip_tokenizer_init=False,
tokenizer_mode=auto, revision=None, override_neuron_config=None,
tokenizer_revision=None, trust_remote_code=False, dtype=torch.float16,
max_seq_len=8192, download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=gptq,
enforce_eager=True, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit,
num_scheduler_steps=1, multi_step_stream_outputs=True,
enable_prefix_caching=None, chunked_prefill_enabled=False,
use_async_output_proc=False, disable_mm_preprocessor_cache=False,
mm_processor_kwargs=None, pooler_config=None,
compilation_config={"splitting_ops":[],"compile_sizes":
[],"cudagraph_capture_sizes":[],"max_capture_size":0},
use_cached_outputs=False,

{"model_id":"4916d217c5b84474bedb2a59a7505a83","version_major":2,"vers
ion_minor":0}

{"model_id":"aadc448a26f64c1897657e1bee091ff2","version_major":2,"vers
ion_minor":0}

{"model_id":"18b7a73a809f48359e5b79793b96ac07","version_major":2,"vers
ion_minor":0}

{"model_id":"e2d4bd26b9994ad7b63c2785ddcf5554","version_major":2,"vers
ion_minor":0}

INFO 05-08 08:40:05 [cuda.py:240] Cannot use FlashAttention-2 backend
for Volta and Turing GPUs.
INFO 05-08 08:40:05 [cuda.py:289] Using XFormers backend.
INFO 05-08 08:40:06 [parallel_state.py:1004] rank 0 in world size 1 is
assigned as DP rank 0, PP rank 0, TP rank 0
INFO 05-08 08:40:06 [model_runner.py:1108] Starting to load model
astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit...
INFO 05-08 08:40:07 [weight_utils.py:265] Using model weights format
['*.safetensors']

{"model_id":"1dbca40f8fb649b4855134ba05a5c0a7","version_major":2,"vers
ion_minor":0}
```

```
INFO 05-08 08:40:53 [weight_utils.py:281] Time spent downloading
weights for astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit: 46.198459
seconds
INFO 05-08 08:40:57 [weight_utils.py:315] No
model.safetensors.index.json found in remote.
```


```
{"model_id":"9da8626ce8c040198146de937035a5da","version_major":2,"vers
ion_minor":0}
```


```
INFO 05-08 08:41:15 [loader.py:458] Loading weights took 18.57 seconds
INFO 05-08 08:41:16 [model_runner.py:1140] Model loading took 5.3473
GiB and 69.618637 seconds
INFO 05-08 08:41:23 [worker.py:287] Memory profiling takes 6.60
seconds
INFO 05-08 08:41:23 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 08:41:23 [worker.py:287] model weights take 5.35GiB;
non_torch_memory takes 0.05GiB; PyTorch activation peak memory takes
0.85GiB; the rest of the memory reserved for KV Cache is 7.02GiB.
INFO 05-08 08:41:24 [executor_base.py:112] # cuda blocks: 3593, # CPU
blocks: 2048
INFO 05-08 08:41:24 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 7.02x
INFO 05-08 08:41:26 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 10.25 seconds
```


```
{"model_id":"1c291b44b3744750b870e2997a067159","version_major":2,"vers
ion_minor":0}
```


```
Batch size: 1 | Latency: 6.92 sec | Throughput: 21.67 tokens/sec
WARNING 05-08 08:41:34 [config.py:2972] Casting torch.bfloat16 to
torch.float16.
INFO 05-08 08:41:34 [config.py:717] This model supports multiple
tasks: {'generate', 'score', 'reward', 'classify', 'embed'}.
Defaulting to 'generate'.
INFO 05-08 08:41:34 [gptq_bitblas.py:168] Detected that the model can
run with gptq_bitblas, however you specified quantization=gptq
explicitly, so forcing gptq. Use quantization=gptq_bitblas for faster
inference
WARNING 05-08 08:41:34 [config.py:830] gptq quantization is not fully
optimized yet. The speed can be slower than non-quantized models.
WARNING 05-08 08:41:34 [cuda.py:93] To see benefits of async output
processing, enable CUDA graph. Since, enforce-eager is enabled, async
output processor cannot be used
INFO 05-08 08:41:34 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='astronomer/Llama-3-8B-Instruct-
GPTQ-4-Bit', speculative_config=None, tokenizer='astronomer/Llama-3-
8B-Instruct-GPTQ-4-Bit', skip_tokenizer_init=False,
tokenizer_mode=auto, revision=None, override_neuron_config=None,
tokenizer_revision=None, trust_remote_code=False, dtype=torch.float16,
```

```
max_seq_len=8192, download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=gptq,
enforce_eager=True, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit,
num_scheduler_steps=1, multi_step_stream_outputs=True,
enable_prefix_caching=None, chunked_prefill_enabled=False,
use_async_output_proc=False, disable_mm_preprocessor_cache=False,
mm_processor_kwargs=None, pooler_config=None,
compilation_config={"splitting_ops":[],"compile_sizes":
[],"cudagraph_capture_sizes":[],"max_capture_size":0},
use_cached_outputs=False,
INFO 05-08 08:41:35 [model_runner.py:1108] Starting to load model
astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit...
INFO 05-08 08:41:35 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 08:41:36 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"b65b52ecd7134838a65c8379c160b9eb","version_major":2,"vers
ion_minor":0}

INFO 05-08 08:42:02 [loader.py:458] Loading weights took 25.92 seconds
INFO 05-08 08:42:02 [model_runner.py:1140] Model loading took 5.3452
GiB and 26.852584 seconds
INFO 05-08 08:42:08 [worker.py:287] Memory profiling takes 5.32
seconds
INFO 05-08 08:42:08 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 08:42:08 [worker.py:287] model weights take 5.35GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.84GiB; the rest of the memory reserved for KV Cache is 7.08GiB.
INFO 05-08 08:42:09 [executor_base.py:112] # cuda blocks: 3623, # CPU
blocks: 2048
INFO 05-08 08:42:09 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 7.08x
INFO 05-08 08:42:09 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 6.69 seconds

{"model_id":"9907efed90594abf9237939dda99cb28","version_major":2,"vers
ion_minor":0}

Batch size: 2 | Latency: 6.69 sec | Throughput: 44.53 tokens/sec
WARNING 05-08 08:42:16 [config.py:2972] Casting torch.bfloat16 to
torch.float16.
```

```
INFO 05-08 08:42:16 [config.py:717] This model supports multiple
tasks: {'generate', 'score', 'reward', 'classify', 'embed'}.
Defaulting to 'generate'.
INFO 05-08 08:42:16 [gptq_bitblas.py:168] Detected that the model can
run with gptq_bitblas, however you specified quantization=gptq
explicitly, so forcing gptq. Use quantization=gptq_bitblas for faster
inference
WARNING 05-08 08:42:16 [config.py:830] gptq quantization is not fully
optimized yet. The speed can be slower than non-quantized models.
WARNING 05-08 08:42:16 [cuda.py:93] To see benefits of async output
processing, enable CUDA graph. Since, enforce-eager is enabled, async
output processor cannot be used
INFO 05-08 08:42:16 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='astronomer/Llama-3-8B-Instruct-
GPTQ-4-Bit', speculative_config=None, tokenizer='astronomer/Llama-3-
8B-Instruct-GPTQ-4-Bit', skip_tokenizer_init=False,
tokenizer_mode=auto, revision=None, override_neuron_config=None,
tokenizer_revision=None, trust_remote_code=False, dtype=torch.float16,
max_seq_len=8192, download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=gptq,
enforce_eager=True, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit,
num_scheduler_steps=1, multi_step_stream_outputs=True,
enable_prefix_caching=None, chunked_prefill_enabled=False,
use_async_output_proc=False, disable_mm_preprocessor_cache=False,
mm_processor_kwargs=None, pooler_config=None,
compilation_config={"splitting_ops":[],"compile_sizes":
[],"cudagraph_capture_sizes":[],"max_capture_size":0},
use_cached_outputs=False,
INFO 05-08 08:42:18 [model_runner.py:1108] Starting to load model
astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit...
INFO 05-08 08:42:18 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 08:42:19 [weight_utils.py:315] No
model.safetensors.index.json found in remote.
```

{"model_id":"dbcf39b38bfa455398ec842decdb10c5","version_major":2,"version_minor":0}

```
INFO 05-08 08:42:43 [loader.py:458] Loading weights took 24.71 seconds
INFO 05-08 08:42:44 [model_runner.py:1140] Model loading took 5.3452
GiB and 25.370565 seconds
INFO 05-08 08:42:49 [worker.py:287] Memory profiling takes 4.99
seconds
```

```
INFO 05-08 08:42:49 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 08:42:49 [worker.py:287] model weights take 5.35GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.84GiB; the rest of the memory reserved for KV Cache is 7.08GiB.
INFO 05-08 08:42:50 [executor_base.py:112] # cuda blocks: 3623, # CPU
blocks: 2048
INFO 05-08 08:42:50 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 7.08x
INFO 05-08 08:42:50 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 6.33 seconds
```


```
{"model_id":"b0c59ac78f604d86b17290bb1308891f","version_major":2,"vers
ion_minor":0}
```


```
Batch size: 4 | Latency: 7.67 sec | Throughput: 92.72 tokens/sec
WARNING 05-08 08:42:58 [config.py:2972] Casting torch.bfloat16 to
torch.float16.
INFO 05-08 08:42:58 [config.py:717] This model supports multiple
tasks: {'generate', 'score', 'reward', 'classify', 'embed'}.
Defaulting to 'generate'.
INFO 05-08 08:42:58 [gptq_bitblas.py:168] Detected that the model can
run with gptq_bitblas, however you specified quantization=gptq
explicitly, so forcing gptq. Use quantization=gptq_bitblas for faster
inference
WARNING 05-08 08:42:58 [config.py:830] gptq quantization is not fully
optimized yet. The speed can be slower than non-quantized models.
WARNING 05-08 08:42:58 [cuda.py:93] To see benefits of async output
processing, enable CUDA graph. Since, enforce-eager is enabled, async
output processor cannot be used
INFO 05-08 08:42:58 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='astronomer/Llama-3-8B-Instruct-
GPTQ-4-Bit', speculative_config=None, tokenizer='astronomer/Llama-3-
8B-Instruct-GPTQ-4-Bit', skip_tokenizer_init=False,
tokenizer_mode=auto, revision=None, override_neuron_config=None,
tokenizer_revision=None, trust_remote_code=False, dtype=torch.float16,
max_seq_len=8192, download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=gptq,
enforce_eager=True, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit,
num_scheduler_steps=1, multi_step_stream_outputs=True,
enable_prefix_caching=None, chunked_prefill_enabled=False,
use_async_output_proc=False, disable_mm_preprocessor_cache=False,
mm_processor_kwargs=None, pooler_config=None,
```

```
compilation_config={"splitting_ops":[],"compile_sizes":
[],"cudagraph_capture_sizes":[],"max_capture_size":0},
use_cached_outputs=False,
INFO 05-08 08:43:00 [model_runner.py:1108] Starting to load model
astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit...
INFO 05-08 08:43:00 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 08:43:00 [weight_utils.py:315] No
model.safetensors.index.json found in remote.

{"model_id":"ff047d24fbbf464699f56fc6756ad20f","version_major":2,"vers
ion_minor":0}

INFO 05-08 08:43:25 [loader.py:458] Loading weights took 24.83 seconds
INFO 05-08 08:43:26 [model_runner.py:1140] Model loading took 5.3452
GiB and 25.497387 seconds
INFO 05-08 08:43:31 [worker.py:287] Memory profiling takes 4.99
seconds
INFO 05-08 08:43:31 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 08:43:31 [worker.py:287] model weights take 5.35GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.84GiB; the rest of the memory reserved for KV Cache is 7.08GiB.
INFO 05-08 08:43:32 [executor_base.py:112] # cuda blocks: 3623, # CPU
blocks: 2048
INFO 05-08 08:43:32 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 7.08x
INFO 05-08 08:43:32 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 6.35 seconds

{"model_id":"27965432461d42c99aa072559a1cc19e","version_major":2,"vers
ion_minor":0}

Batch size: 8 | Latency: 10.32 sec | Throughput: 144.33 tokens/sec
WARNING 05-08 08:43:43 [config.py:2972] Casting torch.bfloat16 to
torch.float16.
INFO 05-08 08:43:43 [config.py:717] This model supports multiple
tasks: {'generate', 'score', 'reward', 'classify', 'embed'}.
Defaulting to 'generate'.
INFO 05-08 08:43:43 [gptq_bitblas.py:168] Detected that the model can
run with gptq_bitblas, however you specified quantization=gptq
explicitly, so forcing gptq. Use quantization=gptq_bitblas for faster
inference
WARNING 05-08 08:43:43 [config.py:830] gptq quantization is not fully
optimized yet. The speed can be slower than non-quantized models.
WARNING 05-08 08:43:43 [cuda.py:93] To see benefits of async output
processing, enable CUDA graph. Since, enforce-eager is enabled, async
output processor cannot be used
INFO 05-08 08:43:43 [llm_engine.py:240] Initializing a V0 LLM engine
(v0.8.5.post1) with config: model='astronomer/Llama-3-8B-Instruct-
```

```
GPTQ-4-Bit', speculative_config=None, tokenizer='astronomer/Llama-3-
8B-Instruct-GPTQ-4-Bit', skip_tokenizer_init=False,
tokenizer_mode=auto, revision=None, override_neuron_config=None,
tokenizer_revision=None, trust_remote_code=False, dtype=torch.float16,
max_seq_len=8192, download_dir=None, load_format=LoadFormat.AUTO,
tensor_parallel_size=1, pipeline_parallel_size=1,
disable_custom_all_reduce=False, quantization=gptq,
enforce_eager=True, kv_cache_dtype=auto,  device_config=cuda,
decoding_config=DecodingConfig(guided_decoding_backend='auto',
reasoning_backend=None),
observability_config=ObservabilityConfig(show_hidden_metrics=False,
otlp_traces_endpoint=None, collect_model_forward_time=False,
collect_model_execute_time=False), seed=None,
served_model_name=astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit,
num_scheduler_steps=1, multi_step_stream_outputs=True,
enable_prefix_caching=None, chunked_prefill_enabled=False,
use_async_output_proc=False, disable_mm_preprocessor_cache=False,
mm_processor_kwargs=None, pooler_config=None,
compilation_config={"splitting_ops":[],"compile_sizes":
[],"cudagraph_capture_sizes":[],"max_capture_size":0},
use_cached_outputs=False,
INFO 05-08 08:43:44 [model_runner.py:1108] Starting to load model
astronomer/Llama-3-8B-Instruct-GPTQ-4-Bit...
INFO 05-08 08:43:45 [weight_utils.py:265] Using model weights format
['*.safetensors']
INFO 05-08 08:43:45 [weight_utils.py:315] No
model.safetensors.index.json found in remote.
```


```
{"model_id":"b97ed1b1c80f4c969c466fc20c13e60e","version_major":2,"vers
ion_minor":0}
```


```
INFO 05-08 08:44:09 [loader.py:458] Loading weights took 24.41 seconds
INFO 05-08 08:44:10 [model_runner.py:1140] Model loading took 5.3452
GiB and 25.097822 seconds
INFO 05-08 08:44:15 [worker.py:287] Memory profiling takes 5.01
seconds
INFO 05-08 08:44:15 [worker.py:287] the current vLLM instance can use
total_gpu_memory (14.74GiB) x gpu_memory_utilization (0.90) = 13.27GiB
INFO 05-08 08:44:15 [worker.py:287] model weights take 5.35GiB;
non_torch_memory takes 0.00GiB; PyTorch activation peak memory takes
0.84GiB; the rest of the memory reserved for KV Cache is 7.08GiB.
INFO 05-08 08:44:16 [executor_base.py:112] # cuda blocks: 3623, # CPU
blocks: 2048
INFO 05-08 08:44:16 [executor_base.py:117] Maximum concurrency for
8192 tokens per request: 7.08x
INFO 05-08 08:44:16 [llm_engine.py:437] init engine (profile, create
kv cache, warmup model) took 6.49 seconds
```


```
{"model_id":"9d5f5bd65a244cac894a33795c08a185","version_major":2,"vers
ion_minor":0}
```

```
Batch size: 16 | Latency: 18.68 sec | Throughput: 159.81 tokens/sec

 Benchmark complete. Results saved to llama3_gptq_batch_results.csv
```

```
import pandas as pd

# Load and display the CSV results
results_df = pd.read_csv("llama3_gptq_batch_results.csv")
print("\n Benchmark Results:")
display(results_df)
```

```
 Benchmark Results:
```


```
{"summary":"{\n  \"name\": \"results_df\",\n  \"rows\": 5,\n
\"fields\": [\n    {\n      \"column\": \"Batch Size\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
6,\n        \"min\": 1,\n        \"max\": 16,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          2,\n
16,\n          4\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"Latency (s)\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5.032407972332926,\n        \"min\":
6.69,\n        \"max\": 18.68,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          6.69,\n          18.68,\n          7.67\n
],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Throughput (tokens/s)\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
60.28055341484516,\n        \"min\": 21.67,\n        \"max\": 159.81,\
n        \"num_unique_values\": 5,\n        \"samples\": [\n
44.53,\n          159.81,\n          92.72\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
n    }\n  ]\n}","type":"dataframe","variable_name":"results_df"}
```


```
import matplotlib.pyplot as plt

# Plotting
plt.figure(figsize=(10, 5))

# Latency plot
plt.subplot(1, 2, 1)
plt.plot(results_df["Batch Size"], results_df["Latency (s)"],
marker='o', color='orange')
plt.title("Latency vs Batch Size")
plt.xlabel("Batch Size")
plt.ylabel("Latency (s)")
plt.grid(True)

# Throughput plot
plt.subplot(1, 2, 2)
plt.plot(results_df["Batch Size"], results_df["Throughput
```

```
(tokens/s)"], marker='o', color='green')
plt.title("Throughput vs Batch Size")
plt.xlabel("Batch Size")
plt.ylabel("Throughput (tokens/s)")
plt.grid(True)

plt.tight_layout()
plt.show()
```