

# 8G 显存玩转书生大模型 Demo

## 关卡任务

本关任务主要包括：

1. InternLM2-Chat-1.8B 模型的部署（基础任务）
2. InternLM-XComposer2-VL-1.8B 模型的部署（进阶任务）
3. InternVL2-2B 模型的部署（进阶任务）

任务详情参见 [8G 显存玩转书生大模型 Demo 任务](./task.md)。

## 创建开发机

我们选择 10% 的开发机，镜像选择为 Cuda-12.2。在输入开发机名称后，点击创建开发机。



在创建完成后，我们便可以进入开发机了！

## 环境配置

我们首先来为 Demo 创建一个可用的环境。

# 创建环境  
conda create -n demo python=3.10 -y  
# 激活环境  
conda activate demo  
# 安装 torch  
conda install pytorch==2.1.2 torchvision==0.16.2 torchaudio==2.1.2 pytorch-cuda=12.1 -c pytorch -c nvidia -y  
# 安装其他依赖  
pip install transformers==4.38  
pip install sentencepiece==0.1.99  
pip install einops==0.8.0  
pip install protobuf==5.27.2  
pip install accelerate==0.33.0  
pip install streamlit==1.37.0

## Cli Demo 部署 InternLM2-Chat-1.8B 模型

首先，我们创建一个目录，用于存放我们的代码。并创建一个 cli\_demo.py。

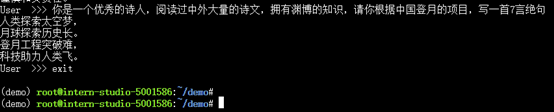
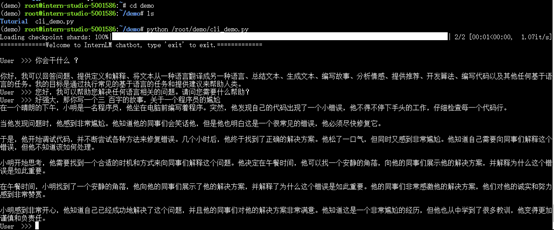
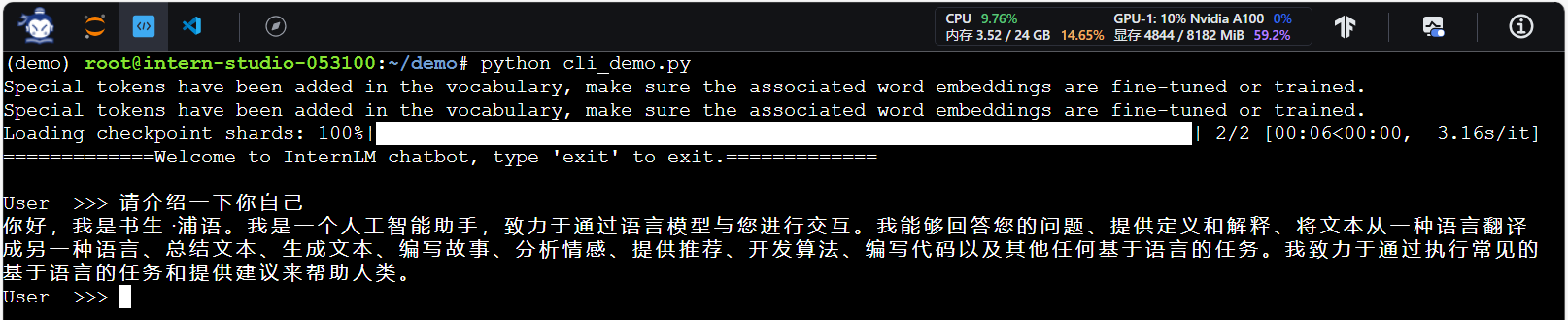
mkdir -p /root/demo  
touch /root/demo/cli\_demo.py

然后，我们将下面的代码复制到 cli\_demo.py 中。

import torch  
from transformers import AutoTokenizer, AutoModelForCausalLM  
  
  
model\_name\_or\_path = "/root/share/new\_models/Shanghai\_AI\_Laboratory/internlm2-chat-1\_8b"  
  
tokenizer = AutoTokenizer.from\_pretrained(model\_name\_or\_path, trust\_remote\_code=True, device\_map='cuda:0')  
model = AutoModelForCausalLM.from\_pretrained(model\_name\_or\_path, trust\_remote\_code=True, torch\_dtype=torch.bfloat16, device\_map='cuda:0')  
model = model.eval()  
  
system\_prompt = """You are an AI assistant whose name is InternLM (书生·浦语).  
- InternLM (书生·浦语) is a conversational language model that is developed by Shanghai AI Laboratory (上海人工智能实验室). It is designed to be helpful, honest, and harmless.  
- InternLM (书生·浦语) can understand and communicate fluently in the language chosen by the user such as English and 中文.  
"""  
  
messages = [(system\_prompt, '')]  
  
print("=============Welcome to InternLM chatbot, type 'exit' to exit.=============")  
  
while True:  
 input\_text = input("\nUser >>> ")  
 input\_text = input\_text.replace(' ', '')  
 if input\_text == "exit":  
 break  
  
 length = 0  
 for response, \_ in model.stream\_chat(tokenizer, input\_text, messages):  
 if response is not None:  
 print(response[length:], flush=True, end="")  
 length = len(response)

接下来，我们便可以通过 python /root/demo/cli\_demo.py 来启动我们的 Demo。

效果如下图所示：



## Streamlit Web Demo 部署 InternLM2-Chat-1.8B 模型

在本节中，我们将演示如何使用 Streamlit 部署 InternLM2-Chat-1.8B 模型。

我们执行如下代码来把本教程仓库 clone 到本地，以执行后续的代码。

cd /root/demo  
git clone https://github.com/InternLM/Tutorial.git

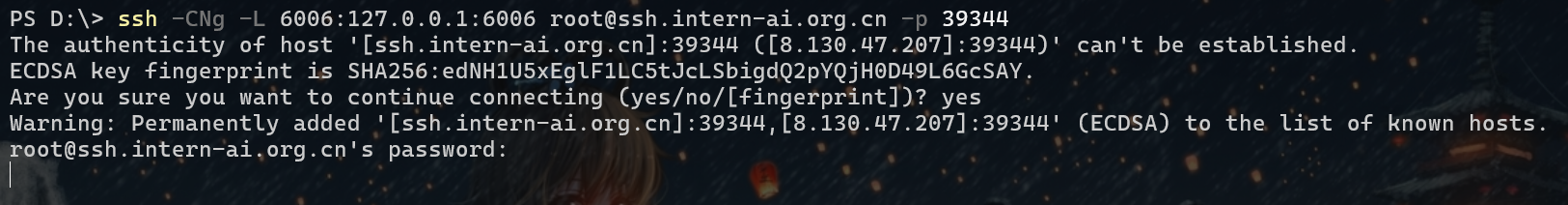
然后，我们执行如下代码来启动一个 Streamlit 服务。

cd /root/demo  
streamlit run /root/demo/Tutorial/tools/streamlit\_demo.py --server.address 127.0.0.1 --server.port 6006

接下来，我们在**本地**的 PowerShell 中输入以下命令，将端口映射到本地。

ssh -CNg -L 6006:127.0.0.1:6006 root@ssh.intern-ai.org.cn -p 你的 ssh 端口号

然后将 SSH 密码复制并粘贴到 PowerShell 中，回车，即可完成端口映射。正常情况下，我们会看到如下输出：



在完成端口映射后，我们便可以通过浏览器访问 http://localhost:6006 来启动我们的 Demo。

效果如下图所示：



如果遇到了 OSError: [Errno 28] inotify watch limit reached 的问题，请不要慌张。稍等一段时间后重新执行即可。



## LMDeploy 部署 InternLM-XComposer2-VL-1.8B 模型

InternLM-XComposer2 是一款基于 InternLM2 的视觉语言大模型，其擅长自由形式的文本图像合成和理解。其主要特点包括：

* 自由形式的交错文本图像合成：InternLM-XComposer2 可以根据大纲、详细文本要求和参考图像等不同输入，生成连贯且上下文相关，具有交错图像和文本的文章，从而实现高度可定制的内容创建。
* 准确的视觉语言问题解决：InternLM-XComposer2 基于自由形式的指令准确地处理多样化和具有挑战性的视觉语言问答任务，在识别，感知，详细标签，视觉推理等方面表现出色。
* 令人惊叹的性能：基于 InternLM2-7B 的InternLM-XComposer2 在多个基准测试中位于开源多模态模型第一梯队，而且在部分基准测试中与 GPT-4V 和 Gemini Pro 相当甚至超过它们。

LMDeploy 是一个用于压缩、部署和服务 LLM 的工具包，由 MMRazor 和 MMDeploy 团队开发。它具有以下核心功能：

* 高效的推理：LMDeploy 通过引入持久化批处理、块 KV 缓存、动态分割与融合、张量并行、高性能 CUDA 内核等关键技术，提供了比 vLLM 高 1.8 倍的推理性能。
* 有效的量化：LMDeploy 支持仅权重量化和 k/v 量化，4bit 推理性能是 FP16 的 2.4 倍。量化后模型质量已通过 OpenCompass 评估确认。
* 轻松的分发：利用请求分发服务，LMDeploy 可以在多台机器和设备上轻松高效地部署多模型服务。
* 交互式推理模式：通过缓存多轮对话过程中注意力的 k/v，推理引擎记住对话历史，从而避免重复处理历史会话。
* 优秀的兼容性：LMDeploy支持 KV Cache Quant，AWQ 和自动前缀缓存同时使用。

LMDeploy 已经支持了 InternLM-XComposer2 系列的部署，但值得注意的是 LMDeploy 仅支持了 InternLM-XComposer2 系列模型的视觉对话功能。

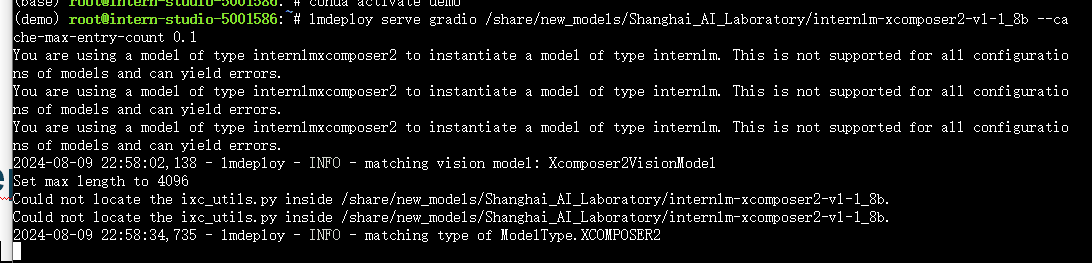
接下来，我们将演示如何使用 LMDeploy 部署 InternLM-XComposer2-VL-1.8B 模型。

首先，我们激活环境并安装 LMDeploy 以及其他依赖。

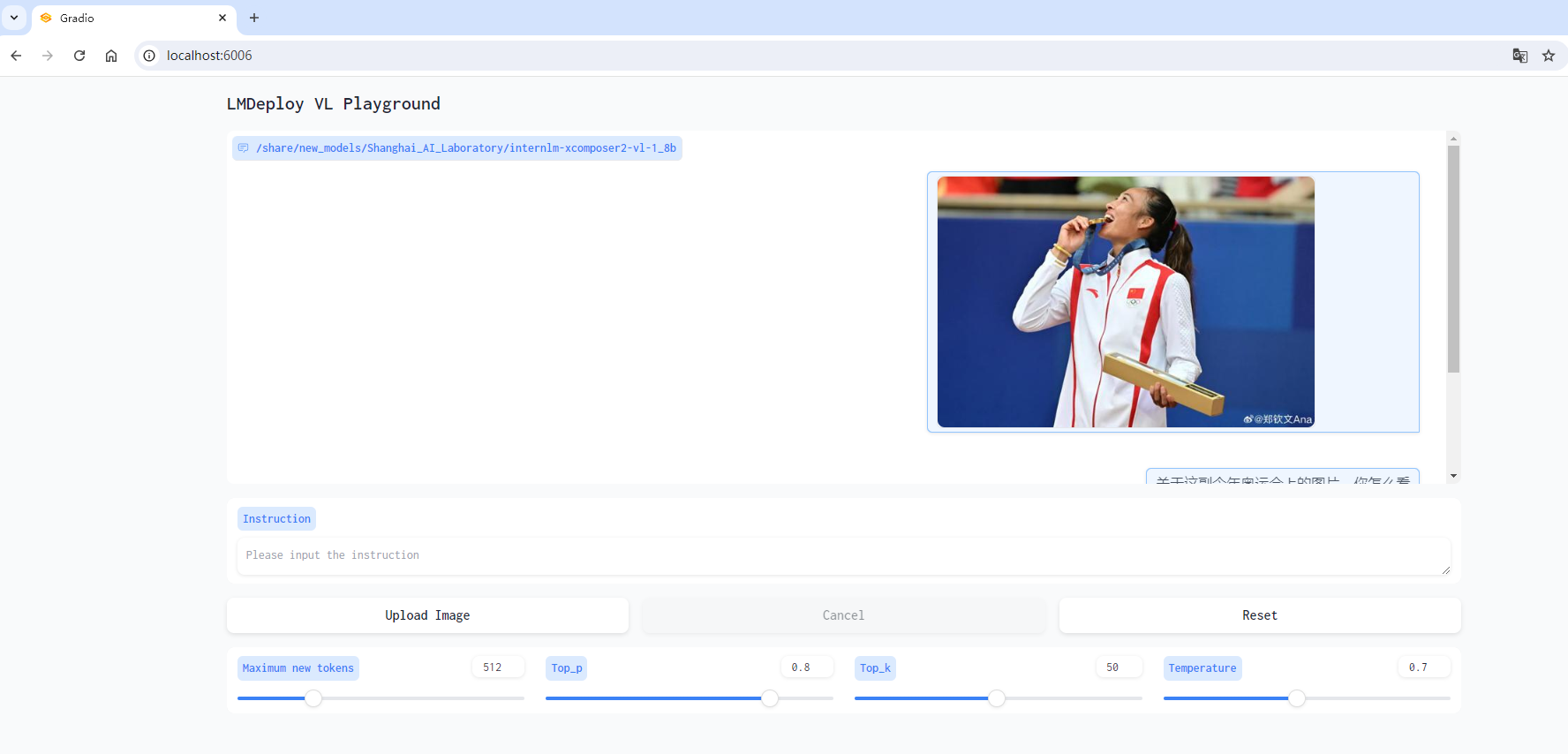
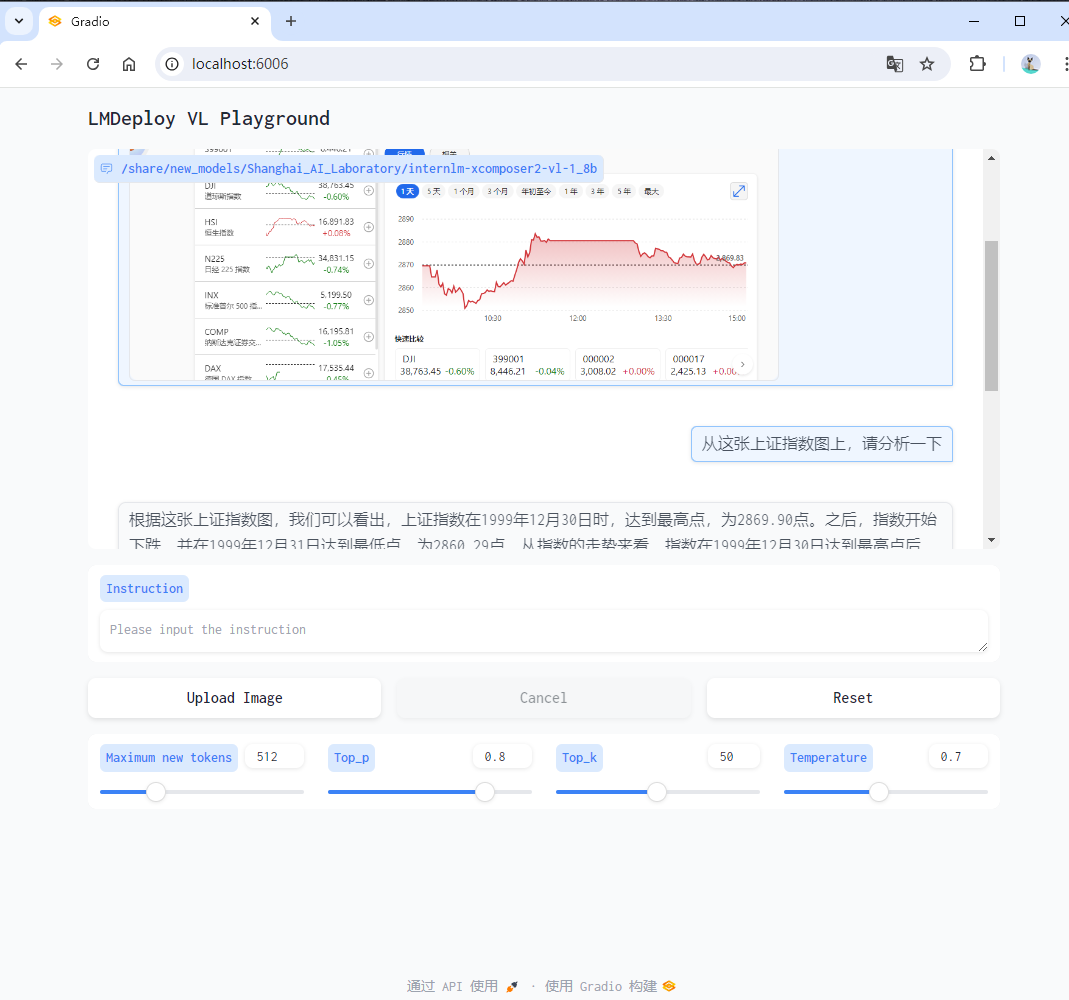
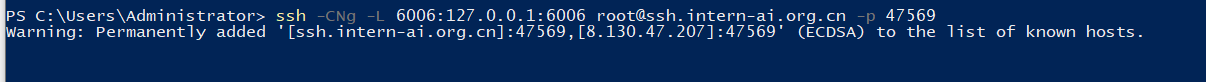
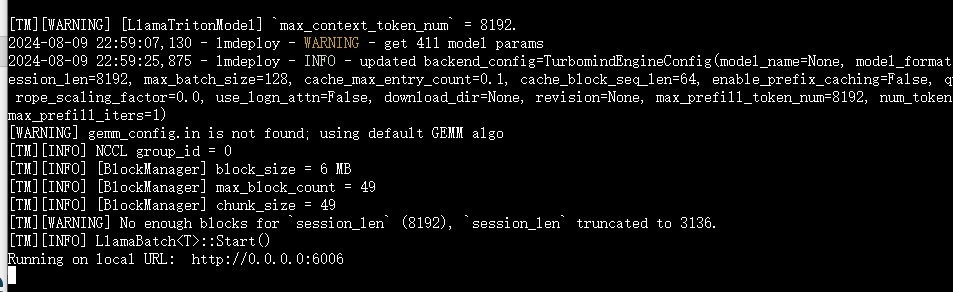
conda activate demo  
pip install lmdeploy[all]==0.5.1  
pip install timm==1.0.7

接下来，我们使用 LMDeploy 启动一个与 InternLM-XComposer2-VL-1.8B 模型交互的 Gradio 服务。

lmdeploy serve gradio /share/new\_models/Shanghai\_AI\_Laboratory/internlm-xcomposer2-vl-1\_8b --cache-max-entry-count 0.1



在使用 Upload Image 上传图片后，我们输入 Instruction 后按下回车，便可以看到模型的输出。



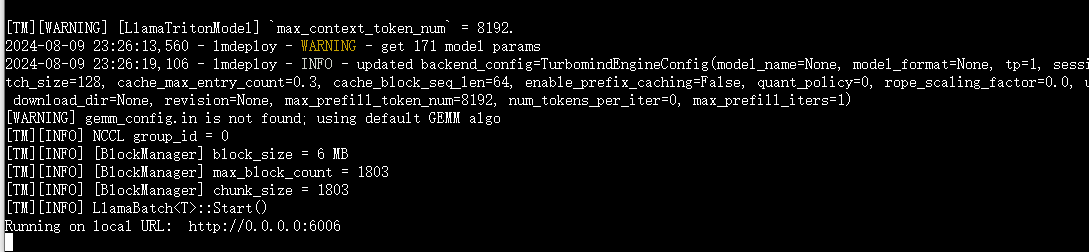
## LMDeploy 部署 InternVL2-2B 模型

InternVL2 是上海人工智能实验室推出的新一代视觉-语言多模态大模型，是首个综合性能媲美国际闭源商业模型的开源多模态大模型。InternVL2 系列从千亿大模型到端侧小模型全覆盖，通专融合，支持多种模态。

LMDeploy 也已经支持了 InternVL2 系列模型的部署，让我们一起来使用 LMDeploy 部署 InternVL2-2B 模型。

我们可以通过下面的命令来启动 InternVL2-2B 模型的 Gradio 服务。

conda activate demo  
lmdeploy serve gradio /share/new\_models/OpenGVLab/InternVL2-2B --cache-max-entry-count 0.1



在完成端口映射后，我们便可以通过浏览器访问 http://localhost:6006 来启动我们的 Demo。

在使用 Upload Image 上传图片后，我们输入 Instruction 后按下回车，便可以看到模型的输出。

