

一、DeepSeek R1 模型参数与显卡需求

Model	Parameters (B)	Disk Space	RAM	Recommended GPU
DeepSeek-R1-Distill-Qwen-1.5B	1.5B	1.1 GB	~3.5 GB	NVIDIA RTX 3060 12GB or higher
DeepSeek-R1-Distill-Qwen-7B	7B	4.7GB	~16 GB	NVIDIA RTX 4080 16GB or higher
DeepSeek-R1-Distill-Llama-8B	8B	4.9GB	~18 GB	NVIDIA RTX 4080 16GB or higher
DeepSeek-R1-Distill-Qwen-14B	14B	9GB	~32 GB	Multi-GPU setup (e.g., NVIDIA RTX 4090 x2)
DeepSeek-R1-Distill-Qwen-32B	32B	20GB	~74 GB	Multi-GPU setup (e.g., NVIDIA RTX 4090 x4)
DeepSeek-R1-Distill-Llama-70B	70B	43GB	~161 GB	Multi-GPU setup (e.g., NVIDIA A100 80GB x2)
DeepSeek-R1	671B	404GB	~1,342 GB	Multi-GPU setup (e.g., NVIDIA A100 80GB x16)

二、DeepSeek R1 + Ollama + OpenWebUI 部署实战

环境信息:

```
Ubuntu 24.04 LTS
conda 24.5.0
Docker version 26.1.0
```

1. 在 Ubuntu 上安装 Ollama

使用官方安装脚本

步骤-1: 运行安装命令

```
curl -fsSL https://ollama.com/install.sh | sh
```

步骤-2: 验证安装

```
ollama --version
```

```
(base) root@ubuntu:~# curl -fsSL https://ollama.com/install.sh | sh
>>> Cleaning up old version at /usr/local/lib/ollama
>>> Installing ollama to /usr/local
>>> Downloading Linux amd64 bundle
##### 100.0%
>>> Creating ollama user...
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
Created symlink /etc/systemd/system/default.target.wants/ollama.service → /etc/systemd/system/ollama.service.
>>> NVIDIA GPU installed.
(base) root@ubuntu:~#
(base) root@ubuntu:~# ollama --version
ollama version is 0.5.7
```

2. 下载并运行 DeepSeek R1 模型

方式-1: 通过 CLI(命令行界面) 运行推理

```
ollama run deepseek-r1:1.5b
```

Ctrl+d 退出

```
(base) root@ubuntu:~# ollama run deepseek-r1:1.5b
pulling manifest
pulling manifest
pulling aabd4debf0c8... 100% 1.1 GB
pulling 369ca498f347... 100% 387 B
pulling 6e4c38e1172f... 100% 1.1 KB
pulling f4d24e9138dd... 100% 148 B
pulling a85fe2a2e58e... 100% 487 B
verifying sha256 digest
writing manifest
success
>>>
>>> 你是谁?
<think>
</think>
您好! 我是由中国的深度求索 (DeepSeek) 公司开发的智能助手DeepSeek-R1。如您有任何任何问题, 我会尽我所能为您提供帮助。
```

方式-2: 通过 API 访问服务

步骤-1: 启动 Ollama 服务器

启动服务

```
ollama serve
```

关闭服务

```
ollama stop
```

```
(base) root@ubuntu:~# ollama serve
2025/02/04 14:47:24 routes.go:1187: INFO server config env="map[CUDA_VISIBLE_DEVICES: GPU_DEVICE_ORDINAL: HIP_VISIBLE_DEVICES:
HSA_OVERRIDE_GFX_VERSION: HTTPS_PROXY: HTTP_PROXY: NO_PROXY: OLLAMA_DEBUG:false OLLAMA_FLASH_ATTENTION:false OLLAMA_GPU_OVERH
EAD:0 OLLAMA_HOST:http://0.0.0.0:6006 OLLAMA_INTEL_GPU:false OLLAMA_KEEP_ALIVE:5m0s OLLAMA_KV_CACHE_TYPE: OLLAMA_LLM_LIBRARY:
OLLAMA_LOAD_TIMEOUT:5m0s OLLAMA_MAX_LOADED_MODELS:0 OLLAMA_MAX_QUEUE:512 OLLAMA_MODELS:/root/.ollama/models OLLAMA_MULTIUSER_C
ACHE:false OLLAMA_NOHISTORY:false OLLAMA_NOPRUNE:false OLLAMA_NUM_PARALLEL:0 OLLAMA_ORIGINS:[http://localhost https://localhos
t http://localhost:* https://localhost:* http://127.0.0.1 https://127.0.0.1 http://127.0.0.1:* https://127.0.0.1:* http://0.0.
0.0 https://0.0.0.0 http://0.0.0.0:* https://0.0.0.0:* app://* file://* tauri://* vscode-webview://*] OLLAMA_SCHED_SPREAD:fals
e ROCR_VISIBLE_DEVICES: http_proxy: https_proxy: no_proxy:]"
time=2025-02-04T14:47:24.529+08:00 level=INFO source=images.go:432 msg="total blobs: 0"
time=2025-02-04T14:47:24.529+08:00 level=INFO source=images.go:439 msg="total unused blobs removed: 0"
[GIN-debug] [WARNING] Creating an Engine instance with the Logger and Recovery middleware already attached.

[GIN-debug] [WARNING] Running in "debug" mode. Switch to "release" mode in production.
- using env:     export GIN_MODE=release
- using code:    gin.SetMode(gin.ReleaseMode)
```

!!! 注意: 先验证端口是否被占用, 如果被占用, 则kill掉

查询端口

```
ps aux | grep ollama
```

```
lsof -i :11434
```

关闭端口

```
kill -9 端口
```

如果ollama服务无法关闭, 原因是ollama开启了守护进程

执行下面命令关闭:

(1) 检查是否有相关服务

```
sudo systemctl status ollama
```

(2) 如果有服务条目, 可以停止服务

```
sudo systemctl stop ollama
```

(3) 禁用自动启动

```
sudo systemctl disable ollama
```

步骤-2: 通过 API 访问服务

```
curl http://localhost:11434/api/chat -d '{
  "model": "deepseek-r1:1.5b",
  "messages": [{ "role": "user", "content": "请解释一下PPO强化学习算法" }],
  "stream": false
}'
```

```
(base) root@ubuntu:~# curl http://localhost:11434/api/chat -d '{
  "model": "deepseek-r1:1.5b",
  "messages": [{ "role": "user", "content": "请解释一下PPO强化学习算法" }],
  "stream": false
}'
{"model":"deepseek-r1:1.5b","created_at":"2025-02-04T06:50:03.363083596Z","message":{"role":"assistant","content":"\u003cthink
\u003e\n嗯, 用户让我解释PPO强化学习算法。首先, 我得确认用户是否对机器学习中的强化学习感兴趣。可能他们正在学习AI或者相关领域,
```

方式-3: 通过 Python 访问服务

步骤-1: 安装 Ollama Python 包

```
pip install ollama
```

步骤-2: 使用以下脚本与模型交互 (ollama_test.py)

```
import ollama
response = ollama.chat(
    model="deepseek-r1:1.5b",
    messages=[
        {"role": "user", "content": "请解释一下PPO强化学习算法"},
    ],
)
print(response["message"]["content"])
```

3. 使用 Docker 部署模型

参考链接: <https://ollama.com/blog/ollama-is-now-available-as-an-official-docker-image>

步骤-1: 拉取镜像

```
docker pull ollama/ollama
```

```
(base) root@ubuntu:~#
(base) root@ubuntu:~# docker pull ollama/ollama
Using default tag: latest
latest: Pulling from ollama/ollama
6414378b6477: Pull complete
ce386310af0b: Pull complete
434f39e9aa8e: Pull complete
bbc15f5291c8: Pull complete
Digest: sha256:7e672211886f8bd4448a98ed577e26c816b9e8b052112860564afaa2c105800e
Status: Downloaded newer image for ollama/ollama:latest
docker.io/ollama/ollama:latest
```

步骤-2: 运行 Ollama

```
docker run -d --gpus=all -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama
```

参数解释:

- `-d`: 后台运行容器
- `--name ollama`: 容器命名为ollama
- `-p 11434:11434`: 映射Ollama的API端口

```
(base) root@ubuntu:~#
(base) root@ubuntu:~# docker run -d --gpus=all -v ollama:/root/.ollama -p 11434:11434 --name ollama ollama/ollama
07eec9b143d6a10c946dcf73d5882743db7146eac6f152eec032f242d1d56613
(base) root@ubuntu:~#
(base) root@ubuntu:~# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED        STATUS        PORTS
NAMES
07eec9b143d6   ollama/ollama  "/bin/ollama serve"      10 seconds ago Up 10 seconds  0.0.0.0:11434->11434/tcp, :::11434->11434/tcp
ollama
(base) root@ubuntu:~#
```

步骤-3: 启动模型服务

(1) 进入容器内

```
docker exec -it ollama /bin/bash
```

(2) 启动模型服务

```
ollama run deepseek-r1:1.5b
```

```
(base) root@ubuntu:~# docker exec -it ollama /bin/bash
root@07eec9b143d6:/#
root@07eec9b143d6:/# ollama run deepseek-r1:1.5b
pulling manifest
pulling aabd4debf0c8... 100% 1.1 GB
pulling 369ca498f347... 100% 387 B
pulling 6e4c38e1172f... 100% 1.1 KB
pulling f4d24e9138dd... 100% 148 B
pulling a85fe2a2e58e... 100% 487 B
verifying sha256 digest
writing manifest
success
>>> 请介绍一下deepseek
<think>
</think>
深度求索人工智能基础技术研究有限公司(简称“深度求索”或“DeepSeek”), 成立于2023年, 是一家专注于实现AGI的中国公司。
>>> Send a message (/? for help)
```

(3) API服务访问 (在容器外执行下面的命令)

```
curl -X POST \
  -H "Content-Type: application/json" \
  http://localhost:11434/api/chat \
  -d '{
    "model": "deepseek-r1:1.5b",
    "messages": [
      { "role": "user", "content": "请解释一下PPO强化学习算法" }
    ],
    "stream": false
  }'
```

4. OpenWebUI 部署与交互

```
# 参考链接: https://github.com/open-webui/open-webui.git
```

1. 拉取镜像

```
docker pull ghcr.io/open-webui/open-webui:main
```

2. 启动并运行一个新的容器

```
docker run -d -p 8805:8080 --add-host=host.docker.internal:host-gateway -v open-
webui:/app/backend/data --name open-webui ghcr.io/open-webui/open-webui:main
```

```
(rag) root@ubuntu:~#  
(rag) root@ubuntu:~# docker run -d -p 8805:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui ghcr.io/open-w  
ebui/open-webui:main  
727fe5744b20f3800e57a623cfa4ff35f261f7c249ef525d07ede82ea23bbfa6  
(rag) root@ubuntu:~#  
(rag) root@ubuntu:~# docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS                    PORTS  
727fe5744b20   ghcr.io/open-webui/open-webui:main  "bash start.sh"         4 seconds ago Up 4 seconds (health: starting)  0.0.0.0:8805->8080/tcp, :::8805->8080  
/tcp open-webui
```

3. 启动 ollama docker 模型服务

docker restart ollama

4. Web UI 交互

localhost:8805

≡ deepseek-r1:1.5b ▾ +
设为默认

deepseek-r1:1.5b

+ 有什么我能帮您的吗?



🔗 建议

Help me study

vocabulary for a college entrance exam

Give me ideas

for what to do with my kids' art

Explain options trading

if I'm familiar with buying and selling stocks

AI进阶学习

📺 B站课堂: <https://space.bilibili.com/3546748815411393/pugv>

🌐 官方网站: <https://www.tgltommy.com/> (国内访问需科学上网)

如果你正在关注大模型 Agent、强化学习后训练、RLHF、DPO、GRPO、RLVR 等前沿方向, 欢迎关注我最新上线的精品课程:
《大模型 Agent 强化学习实战: 从后训练、奖励设计到工业级对齐系统》

这门课程围绕当前大模型 Agent 强化学习的核心技术路线展开, 不只是介绍零散算法和论文概念, 而是希望帮助学习者建立一套完整的技术地图: 从基础原理、论文脉络、关键算法, 到开源项目源码解析与工业级系统实践, 逐步理解 Agent RL 为什么重要、如何演进, 以及在真实应用中如何落地。

课程配套专属飞书知识库《Agent RL 学习宝典》，包含学习路线、论文资料、项目说明、源码阅读辅助资料和后续持续更新内容，方便长期查阅和跟进前沿技术。

Agent RL 变化很快，但越是变化快，越需要一套系统的学习框架。如果你希望系统进入 Agent RL 方向，而不是停留在碎片化了解阶段，这门课会是一个很好的起点。

课程官网: <https://www.tgltommy.com/p/agent-rl> (国内访问需科学上网)

B站课堂: <https://www.bilibili.com/cheese/play/ss842375604>

更多课程详情，扫描以下二维码咨询课程助理：



新课首发

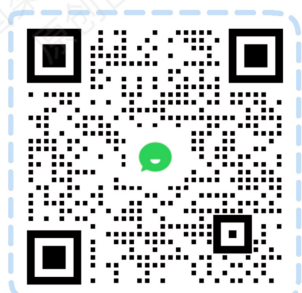
大模型 **Agent** 强化学习实战：

从后训练、奖励设计到工业级对齐系统

唐国梁Tommy

TGL AI Tommy

官方微信课程助理



B站/公众号/YouTube: 唐国梁Tommy

**关注微信公众号
获取更多学习资源**



官方网站: TGLTommy.com(需科学上网)