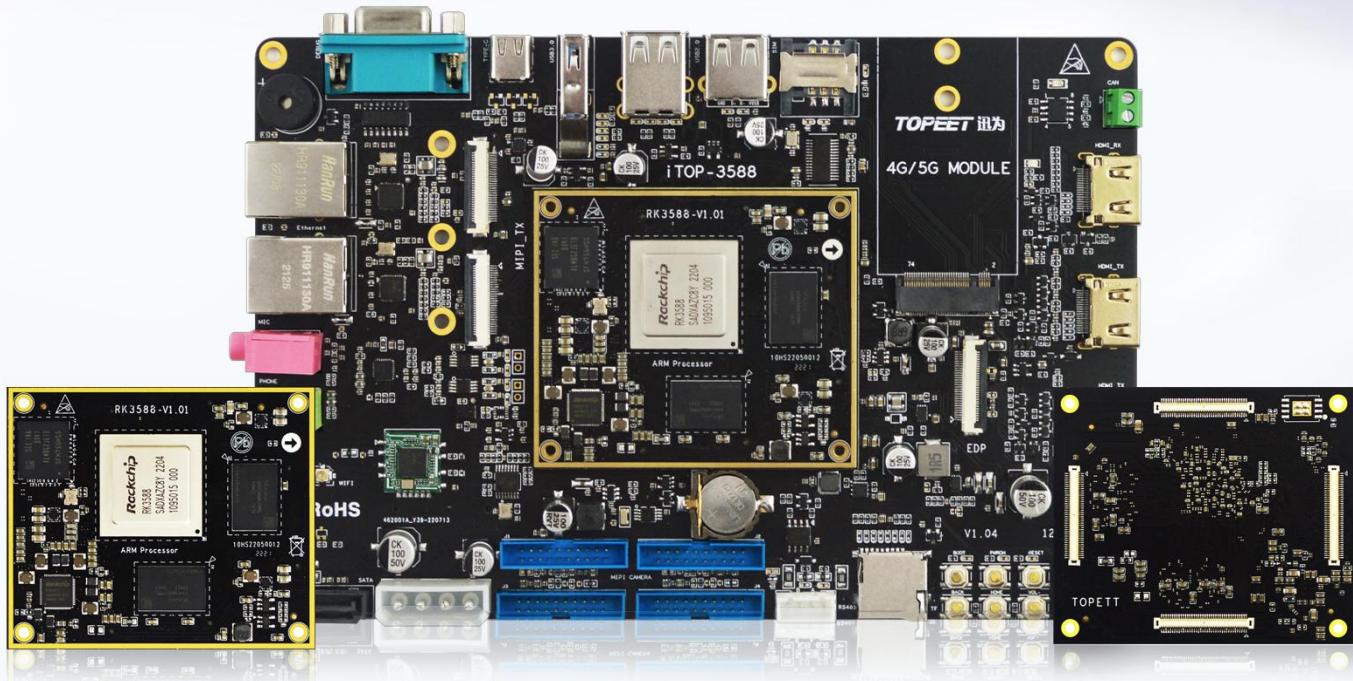


强大的 AI 能力 更快更强

超长供货周期 | 7X24 小时稳定运行 | 8K 视频编解码



iTOP-RK3588 开发板使用手册

八核 64 位 CPU | 主频 2.4GHz | NPU 算力 6T | 4800 安防级别 ISP

更新记录

更新版本	修改内容
V1. 0	初版
V1. 1	修改部分截图

目录

更新记录	2
目录	3
版权声明	4
更多帮助	5
第 1 章 DeepSeek 大预言模型部署测试	6
1. 1 RKLLM-Toolkit 介绍	6
1. 2 RKLLM-Toolkit 环境搭建	6
1. 2. 1 安装 Miniconda	6
1. 2. 2 创建 RKLLM 虚拟环境	8
1. 3 DeepSeek 大语言模型转换	9
1. 4 推理程序编译	12
1. 5 开发板运行测试	14

版权声明

本文档版权归北京迅为电子有限公司所有。未经本公司书面许可，任何单位和个人无权以任何形式复制、传播、转载本文档的任何内容，违者将被追究法律责任。

更多帮助

注意事项与维护

- ❖ 请注意和遵循标注在产品上的所有警示和指引信息；
- ❖ 请勿带电插拔核心板及外围模块；
- ❖ 使用产品之前，请仔细阅读本手册，并妥善保管，以备将来参考；
- ❖ 请使用配套电源适配器，以保证电压、电流的稳定；
- ❖ 请勿在冷热交替环境中使用本产品，避免结露损坏元器件；
- ❖ 请保持产品干燥，如果不慎被任何液体泼溅或浸润，请立刻断电并充分晾干；
- ❖ 请勿使用有机溶剂或腐蚀性液体清洗本产品；
- ❖ 请勿在多尘、脏乱的环境中使用本产品，如果长期不使用，请包装好本产品；
- ❖ 如果在震动场景使用，请做好核心板与底板的固定，避免核心板跌落损坏；
- ❖ 请勿在通电情况下，插拔核心板及外围模块(特别是串口模块)；
- ❖ 请勿自行维修、拆解本产品，如产品出现故障应及时联系本公司进行维修；
- ❖ 请勿自行修改或使用未经授权的配件，由此造成的损坏将不予保修；

资料的更新

为了确保您的资料是最新状态，请密切关注我们的动态，我们将会通过微信公众号和 QQ 群推送。

关注“迅为电子”微信公众号，不定期分享教程、资料和行业干货及产品一线资料。

迅为新媒体账号

官网: <https://www.topeetboard.com>

知乎 <https://www.zhihu.com/people/topeetabc123>

CSDN: <https://blog.csdn.net/BeijingXunWei>



售后服务政策

1. 如产品使用过程中出现硬件故障可根据售后服务政策进行维修
2. 服务政策：参见官方网售后服务说明
<https://www.topeetboard.com/sydyml/Service/bx.html>

送修地址：

1. 地址：北京市海淀区永翔北路 9 号中国航发大厦三层
2. 联系人：迅为开发板售后服务部
3. 电话：010-85270716
4. 邮编：100094
5. 邮寄须知：建议使用顺丰、圆通或韵达，且不接受任何到付

技术支持范围

1. 了解产品的软、硬件资源提供情况咨询
2. 产品的软、硬件手册使用过程中遇到的问题
3. 下载和烧写更新系统过程中遇到的问题
4. 产品用户的资料丢失、更新后重新获取
5. 产品的故障判断及售后维修服务。

PS: (由于嵌入式系统知识范围广泛，我们无法保证对各种问题都能一一解答，部分内容无法供技术支持，只能提供建议。)

技术支持

1. 周一至周五：（法定节假日除外）
上午 9:00 ~ 11:30 / 下午 13:30 ~ 17:30
2. QQ 技术交流群：
824412014
822183461
95631883
861311530

第 1 章 DeepSeek 大预言模型部署测试

1.1 RKLLM-Toolkit 介绍

RKLLM-Toolkit 是为用户提供在计算机上进行大语言模型的量化、转换的开发套件。通过该工具提供的 Python 接口可以便捷地完成以下功能：

1. 模型转换：支持将 Hugging Face 格式的大语言模型（Large Language Model, LLM）转换为 RKLLM 模型，目前支持的模型包括 LLaMA、Qwen/Qwen2、Phi2 等，转换后的 RKLLM 模型能够在 Rockchip NPU 平台上加载使用。
2. 量化功能：支持将浮点模型量化为定点模型，目前支持的量化类型包括 w4a16 和 w8a8。

1.2 RKLLM-Toolkit 环境搭建

1.2.1 安装 Miniconda

Conda 是一个开源的软件包管理系统和环境管理系统，它可以用于安装、管理和升级软件包和依赖项，我们这里使用 conda 的目的只是构建一个虚拟环境，所以选择轻量话的 miniconda。miniconda 的官方链接如下所示：

<https://docs.conda.io/en/latest/miniconda.html>

进入 miniconda 的网址后如下所

The screenshot shows the official documentation for Miniconda. At the top, there's a navigation bar with links for 'conda latest', 'Search docs', and 'Edit on GitHub'. The main content area has a green header 'Miniconda' with a sub-header 'System requirements'. Below this, there's a bulleted list of requirements and a note about system requirements for Windows, macOS, and Linux. Further down, there's a section titled 'Latest Miniconda Installer Links' with a table showing download links for various platforms. The table includes columns for 'Platform', 'Name', and 'SHA256 hash'. The platforms listed are Windows, macOS, and Linux, each with multiple installer options.

Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	307194e1f120beb52b083634e89cc67db4f7980bd542254b43d3309eaef7cb358
	Miniconda3 Windows 32-bit	4fb64ec9c28b80tesb16994bfba4829110ea31450ea80bde5344174ab5d462
macOS	Miniconda3 macOS Intel x86 64-bit bash	5ab0c78066407d0d14ad0330534cc982830b0538c0d010ea9cf6809cc4153f8104
	Miniconda3 macOS Intel x86 64-bit pkg	cce310ef1e5394f2b739726d2255112a19efdf689c13e2568887b7070cbc58
	Miniconda3 macOS Apple M1 64-bit bash	9d1d12573339c490950b0c5a840e9ff7fc32d35c3de1b3d07478c01875eb003d64
	Miniconda3 macOS Apple M1 64-bit pkg	6997472c55ff90a772e077e53974e43e227736c3a7f7b839d33dec77fae75d
Linux	Miniconda3 Linux 64-bit	aef279d50aea7f67940f16aad17ebe5f6aac97487c7c03466ff01f4819e5a851

可以看到下方有各个系统的安装包，我们选择 Miniconda3 Linux 64-bit 和 Miniconda3 Linux-aarch64 64-bit 两个版本的安装包进行下载，如下图所示：

Latest Miniconda Installer Links		
Platform	Name	SHA256 hash
Windows	Miniconda3 Windows 64-bit	307394e1f12b0eb52b083634e89cc67db4f7980bd542254b43d3309eaef7cb358
	Miniconda3 Windows 32-bit	4fb64e6c9c28b88beab16994bf04829110ea3145baea00bda5344174ab65d462
macOS	Miniconda3 macOS Intel x86 64-bit bash	5abc76b664b7da9d14ade330534cc98283bb838c6b10ad9cfdbb9cc4153f8104
	Miniconda3 macOS Intel x86 64-bit pkg	cca31a0f1e5394f2b739726dc22551c2a19afdf689c13a25668887ba706cba58
	Miniconda3 macOS Apple M1 64-bit bash	9d1d12573339c49050b0d5a840af0ff6c32d33c3de1b3db478c01878eb003d64
	Miniconda3 macOS Apple M1 64-bit pkg	6997472c5ff90a772eb77e6397f4e3e227736c83a7f7b839da33d6cc7facb75d
Linux	Miniconda3 Linux 64-bit	aef279d6baea7f67940f16aad17ebef5f6aac97487c7c0346fff01f4819e5a651
	Miniconda3 Linux-aarch64 64-bit	6950c7b1ff4f65ce9b87ee1a2d684837771ae7b2e6044e0da9e915d1dee6c924c
	Miniconda3 Linux-ppc64le 64-bit	b3de538cd542bc4f5a2f2d2a79386288d6e04f0e1459755f3cef64763e51016
	Miniconda3 Linux-s390x 64-bit	ed4f51aefc967e921ff5721151f567a4c43c4288ac93ec2393c6238b8c4891de8

为了方便，已经将两个安装包存放到了“**iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\12_NPU 使用配套资料\07_安装包\01_miniconda**”目录下，如下图所示：

02_【iTOP-RK3588开发板】开发资料 > 12_NPU使用配套资料 > 07_安装包 > 01_miniconda			
名称	修改日期	类型	大小
Miniconda3-latest-Linux-aarch64.sh	2023/4/3 17:04	SH 文件	52,850 KB
Miniconda3-latest-Linux-x86_64.sh	2023/4/3 17:07	SH 文件	72,661 KB

本章节要用到的是 Miniconda3-latest-Linux-x86_64.sh 安装包，而 Miniconda3-latest-Linux-aarch64.sh 安装包会用在之后 RKNN Toolkit lite2 环境搭建中。首先将 Miniconda3-latest-Linux-x86_64.sh 安装包拷贝到虚拟机 ubuntu 上，拷贝完成如下图所示：

```
topeet@ubuntu:~/software$ ls
Miniconda3-latest-Linux-x86_64.sh
topeet@ubuntu:~/software$
```

随后使用“./Miniconda3-latest-Linux-x86_64.sh”命令进行安装，根据提示，输入回车和“yes”，等待安装完成，安装完成如下图所示：

```
==> For changes to take effect, close and re-open your current shell. <==
If you'd prefer that conda's base environment not be activated on startup,
set the auto_activate_base parameter to false:

conda config --set auto_activate_base false

Thank you for installing Miniconda3!
topeet@ubuntu:~/software$
```

安装完成之后会自动设置环境变量，打开新的终端，发现用户名前出现（base），就代表安装成功了，如下图所示：

```
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$  
(base) topeet@ubuntu:~$ cd software/  
(base) topeet@ubuntu:~/software$ █
```

1.2.2 创建 RKLLM 虚拟环境

为了避免环境后续学习中环境的冲突问题，这里使用 conda 创建 RKNN 虚拟环境，使用命令如下所示：

```
conda create -n rkllm python=3.8
```

命令执行之后，首先会要求安装一些列软件包，输入 y 确认即可，如下图所示：

```
_libgcc_mutex      pkgs/main/linux-64::__libgcc_mutex-0.1-main  
_openmp_mutex      pkgs/main/linux-64::__openmp_mutex-5.1-1_gnu  
ca-certificates    pkgs/main/linux-64::ca-certificates-2022.10.11-h06a4308_0  
certifi           pkgs/main/linux-64::certifi-2021.5.30-py36h06a4308_0  
ld_impl_linux-64  pkgs/main/linux-64::ld_impl_linux-64-2.38-h1181459_1  
libffi             pkgs/main/linux-64::libffi-3.3-he6710b0_2  
libgcc-ng          pkgs/main/linux-64::libgcc-ng-11.2.0-h1234567_1  
libgomp            pkgs/main/linux-64::libgomp-11.2.0-h1234567_1  
libstdcxx-ng       pkgs/main/linux-64::libstdcxx-ng-11.2.0-h1234567_1  
ncurses            pkgs/main/linux-64::ncurses-6.3-h5eee18b_3  
openssl            pkgs/main/linux-64::openssl-1.1.1s-h7f8727e_0  
pip                pkgs/main/linux-64::pip-21.2.2-py36h06a4308_0  
python              pkgs/main/linux-64::python-3.6.13-h12debd9_1  
readline            pkgs/main/linux-64::readline-8.2-h5eee18b_0  
setuptools          pkgs/main/linux-64::setuptools-58.0.4-py36h06a4308_0  
sqlite              pkgs/main/linux-64::sqlite-3.40.0-h5082296_0  
tk                  pkgs/main/linux-64::tk-8.6.12-h1ccaba5_0  
wheel               pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0  
xz                  pkgs/main/linux-64::xz-5.2.8-h5eee18b_0  
zlib                pkgs/main/linux-64::zlib-1.2.13-h5eee18b_0
```



```
Proceed ([y]/n)? █
```

等待一些列依赖安装完成，安装完成如下图所示：

Downloading and Extracting Packages

```

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate rkllm
#
# To deactivate an active environment, use
#
#     $ conda deactivate
(base) topeet@ubuntu:~$ █

```

然后使用“conda activate rkllm”激活相应得 rkllm 环境，如下图所示：

```
(base) topeet@ubuntu:~$ conda activate rkllm
(rkllm) topeet@ubuntu:~$ █
```

可以看到，标识符由“base”变成了“rkllm”，然后将“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\01_rkllm-1.1.4\rknn-llm-main\rkllm-toolkit](#)”路径下的 rkllm_toolkit-1.1.4-cp38-cp38-linux_x86_64.whl 文件拷贝到虚拟机 ubuntu 上，拷贝完成如下图所示：

```
(rkllm) topeet@topeet:~/software$ ls
rkllm_toolkit-1.1.4-cp38-cp38-linux_x86_64.whl
(rkllm) topeet@topeet:~/software$ █
```

然后使用以下命令安装瑞芯微提供的 rkllm_toolkit-1.0.0 版本的软件包，安装完成如下图所示：

```
pip install rkllm_toolkit-1.1.4-cp38-cp38-linux_x86_64.whl -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```

Successfully installed Jinja2-3.1.4 MarkupSafe-2.1.5 accelerate-0.26.0 aiohappyeyeballs-2.4.4 aiohttp-3.10.11
aiosignal-1.3.1 annotated-types-0.7.0 argcomplete-3.5.3 async-timeout-5.0.1 attrs-25.1.0 auto-gptq-0.7.1 black-24.8.0 certifi-2025.1.31 charset-normalizer-3.4.1 click-8.1.8 coloredlogs-15.0.1 colorlog-6.8.2 datamodel-code-generator-0.26.0 datasets-2.14.6 dill-0.3.7 dnspython-2.6.1 einops-0.4.1 email-validator-2.2.0 filelock-3.16.1 flatbuffers-24.3.25 frozenlist-1.5.0 fsspec-2023.10.0 gekko-1.2.1 genson-1.3.0 huggingface-hub-0.28.1 humanfriendly-10.0 idna-3.10 importlib-resources-6.4.5 inflect-5.6.2 isort-5.13.2 jsonschema-4.23.0 jsonschemas-a-specifications-2023.12.1 mpmath-1.3.0 multidict-6.1.0 multiprocess-0.70.15 mypy-extensions-1.0.0 networkx-3.1 numpy-1.23.1 nvidia-cublas-cu12-12.1.3.1 nvidia-cuda-cupti-cu12-12.1.105 nvidia-cuda-nvrtc-cu12-12.1.105 nvidia-cuda-runtime-cu12-12.1.105 nvidia-cudnn-cu12-8.9.2.26 nvidia-cufft-cu12-11.0.2.54 nvidia-curand-cu12-10.3.2.106 nvidia-cusolver-cu12-11.4.5.107 nvidia-cusparse-cu12-12.1.0.106 nvidia-nccl-cu12-2.18.1 nvidia-nvjit-link-cu12-12.8.61 nvidia-nvtx-cu12-12.1.105 optimum-1.23.3 packaging-24.2 pandas-2.0.3 pathspec-0.12.1 peft-0.13.2 pillow-10.4.0 pkgutil-resolve-name-1.3.10 platformdirs-4.3.6 propcache-0.2.0 protobuf-3.20.3 psutil-7.0.0 pyarrow-17.0.0 pydantic-2.10.6 pydantic-core-2.27.2 python-dateutil-2.9.0.post0 pytz-2025.1 pyyaml-6.0.2 referencing-0.35.1 regex-2024.11.6 requests-2.32.3 rkllm-toolkit-1.1.4 rouge-1.0.1 rpds-py-0.20.1 safetensors-0.4.2 sentencepiece-0.1.97 six-1.17.0 sympy-1.13.3 tabulate-0.9.0 tiktoken-0.4.0 tokenizers-0.20.3 toml-0.10.2 tomli-2.2.1 torch-2.1.0 torchvision-0.16.0 tqdm-4.64.1 transformers-4.45.0 transformers-stream-generator-0.4 triton-2.1.0 typing-extensions-4.12.2 tzdata-2025.1 urlib3-2.2.3 xxhash-3.5.0 yarl-1.15.2 zipp-3.20.2
(rkllm) topeet@topeet:~/software$ █
(rkllm) topeet@topeet:~/software$ █
(rkllm) topeet@topeet:~/software$ █

```

至此，RKLLM 的虚拟环境就创建完成了，具体的虚拟环境使用之后会进行讲解。

1.3 DeepSeek 大语言模型转换

DeepSeek 模型下载地址为: <https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B>,
需要下载该网址全部的文件和模型:

The screenshot shows the Hugging Face model card for 'DeepSeek-R1-Distill-Qwen-1.5B'. The 'Files and versions' tab is selected. A red box highlights the main file list. The files listed include: msr2000/Update tokenizer_config.json, figures, .gitattributes, LICENSE, README.md, config.json, generation_config.json, model.safetensors, tokenizer.json, and tokenizer_config.json. Most files were updated 5 days ago, except for .gitattributes which was updated 25 days ago.

文件名	更新时间	描述
msr2000/Update tokenizer_config.json	5 days ago	
figures	25 days ago	
.gitattributes	25 days ago	initial commit
LICENSE	25 days ago	Release DeepSeek-R1
README.md	13 days ago	Update README.md
config.json	25 days ago	Add files using upload-large-folder tool
generation_config.json	24 days ago	Add generation_config.json
model.safetensors	25 days ago	Add files using upload-large-folder tool
tokenizer.json	25 days ago	Add files using upload-large-folder tool
tokenizer_config.json	5 days ago	Update tokenizer_config.json

方便起见迅为已经将文件和模型下载了下来，具体存放路径为“iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\02_DeepSeek1.5b 官方模型”如下图所示：

The screenshot shows a file explorer window with the path: 13_NPU使用配套资料 > 11_Deepseek部署测试 > 02_DeepSeek1.5b官方模型. A red box highlights the 'config.json' file. The table below lists the files and their details:

名称	修改日期	类型	大小
config.json	2025/2/5 15:36	JSON 源文件	1 KB
generation_config.json	2025/2/5 15:36	JSON 源文件	1 KB
gitattributes.txt	2025/2/5 15:36	文本文档	2 KB
LICENSE.txt	2025/2/5 15:36	文本文档	2 KB
model.safetensors	2025/2/5 15:43	SAFETENSORS 文件	3,470,913 KB
tokenizer.json	2025/2/5 15:36	JSON 源文件	6,867 KB
tokenizer_config.json	2025/2/5 15:36	JSON 源文件	3 KB

然后将上述文件全部拷贝到虚拟机 ubuntu，拷贝完成如下图所示：

```
(rkllm) topeet@topeet:~/rknpu/deepseek_1.5b$ ls
LICENSE.txt generation_config.json model.safetensors tokenizer_config.json
config.json gitattributes.txt tokenizer.json
(rkllm) topeet@topeet:~/rknpu/deepseek_1.5b$
```

然后将“iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\01_rkllm-1.1.4\rknn-llm-main\examples\DeepSeek-R1-Distill-Qwen-1.5B_Demo\export”路径下的三个文件拷贝到上一级目录，拷贝完成如下图所示：

```
(rkllm) topeet@topeet:~/rknpu$ ls
(rkllm) topeet@topeet:~/rknpu$ ls
data_quant.json deepseek_1.5b export_rkllm.py generate_data_quant.py
(rkllm) topeet@topeet:~/rknpu$
```

然后在终端中输入以下命令生成量化校准数据，过程如如下所示：

```
python3 generate_data_quant.py -m deepseek_1.5b
```

```
(rkllm) topeet@topeet:~/rknpu$ python3 generate_data_quant.py -m deepseek_1.5b
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
Starting from v4.46, the `logits` model output will have the same type as the model (except at train time, where it will always be FP32)
< | User | >在农业生产中被当作极其重要的劳动对象发挥作用，最主要的基本生产资料是
A. 农业生产工具
B. 土地
C. 劳动力
D. 资金< | Assistant | ><think>
嗯，我现在要解决这个问题：在农业生产中被当作极其重要的劳动对象发挥作用，最主要的基本生产资料是什么？选项有四个：A. 农业生产工具，B. 土地，C. 劳动力，D. 资金。
```

首先，我需要理解题目中的问题。题目问的是“主要不可替代的基本生产资料”。这意味着我要找出在农业生产中，哪个因素是最重要的，无法完全替代的资源或要素。

我记得农业生产的各个环节都有不同的基本生产资料。比如，土地是基础，因为它是种植作物、蔬菜等的基础。然后是劳动力

```
320
Setting `pad_token_id` to `eos_token_id`:None for open-end generation.
< | User | >下列行为如满足规定条件，应认定为伪造货币罪的是
A. 将英镑揭层一分为二
B. 铸造珍稀古钱币
C. 临摹欧元收藏
```

然后修改 export_rkllm.py 文件，将 modelpath 修改为自己模型存放的路径，修改完成如下图所示：

```
export_rkllm.py+ buffers
1 from rkllm.api import RKLLM
2 import os
3 os.environ['CUDA_VISIBLE_DEVICES']='0'
4 ''
5 ''
6 https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-1.5B
7
8 Download the DeepSeek R1 model from the above url.
9 ''
10
11 modelpath = 'deepseek_1.5b'
12 llm = RKLLM()
13
```

然后使用以下命令进行模型的转换，转换过程如下所示：

```
python3 export_rkllm.py
```

```
(rkllm) topeet@topeet:~/rknpu$ python3 export_rkllm.py
INFO: rkllm-toolkit version: 1.1.4
WARNING: Cuda device not available! switch to cpu device!
The argument 'trust_remote_code' is to be used with Auto classes. It has no effect here and is ignored.
Downloading data files: 100%|██████████| 1/1 [00:00<00:00, 25890.77it/s]
Extracting data files: 100%|██████████| 1/1 [00:00<00:00, 4100.00it/s]
Generating train split: 21 examples [00:00, 9585.42 examples/s]
Optimizing model: 4%|█| 1/28 [00:23<10:34, 23.48s/it]
```

转换完成之后会在模型目录下生成一个名为 qwen.rkllm 的模型，如下图所示：

```
(rkllm) topeet@topeet:~/rknpu$ python3 export_rkllm.py
INFO: rkllm-toolkit version: 1.1.4
WARNING: Cuda device not available! switch to cpu device!
The argument 'trust_remote_code' is to be used with Auto classes. It has no effect here and is ignored.
Downloading data files: 100%|██████████| 1/1 [00:00<00:00, 25890.77it/s]
Extracting data files: 100%|██████████| 1/1 [00:00<00:00, 4100.00it/s]
Generating train split: 21 examples [00:00, 9585.42 examples/s]
Optimizing model: 100%|██████████| 28/28 [10:49<00:00, 23.20s/it]
Building model: 100%|██████████| 399/399 [00:05<00:00, 78.78it/s]
WARNING: The bos token has two ids: 151646 and 151643, please ensure that the bos token ids in config.json and tokenizer_config.json are consistent!
INFO: The token_id of bos is set to 151646
INFO: The token_id of eos is set to 151643
INFO: The token_id of pad is set to 151643
Converting model: 100%|██████████| 339/339 [00:00<00:00, 5511120.37it/s]
INFO: Exporting the model, please wait ....
[=====] 597/597 (100%)
INFO: Model has been saved to ./deepseek_1.5b_W8A8_RK3588.rkllm!
(rkllm) topeet@topeet:~/rknpu$
```

```
(rkllm) topeet@topeet:~/rknpu$ ls
data_quant.json  deepseek_1.5b  deepseek_1.5b_W8A8_RK3588.rkllm  export_rkllm.py  generate_data_quant.py
(rkllm) topeet@topeet:~/rknpu$
```

为了方便，迅为也提供好了转换完成的模型，具体路径为“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\03_转换好的模型](#)”，如下图所示：



至此，测试要用的 DeepSeek 大语言 RKLLM 大预言模型就转换完成了。

1.4 推理程序编译

首先拷贝“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\12_NPU 使用配套资料\03_编译所需工具\Linux](#)”目录下的交叉编译器到虚拟机 ubuntu 上并解压，解压完成如下图所示：

```
topeet@topeet:~/rknpu$ ls
data_quant.json
deepseek_1.5b
deepseek_1.5b_W8A8_RK3588.rkllm
export_rkllm.py
topeet@topeet:~/rknpu$
```

gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu
 gcc-arm-10.3-2021.07-x86_64-aarch64-none-linux-gnu.tar.gz

而瑞芯微提供了一个用于 DeepSeek 推理的 C++ 应用程序，存放路径为“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\01_rkllm-1.1.4\rknn-llm-main\examples\DeepSeek-R1-Distill-Qwen-1.5B_Demo\deploy](#)”，如下图

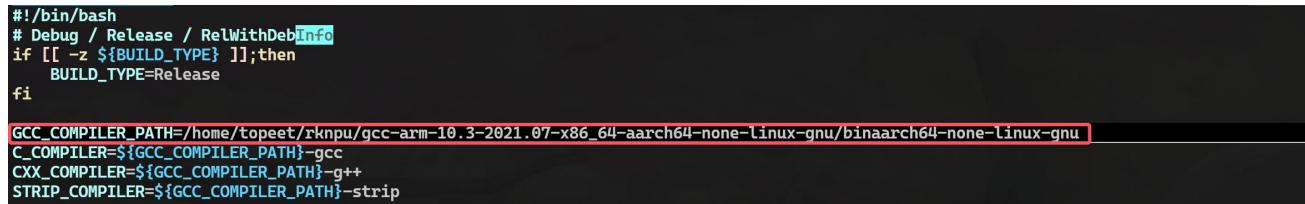
所示：



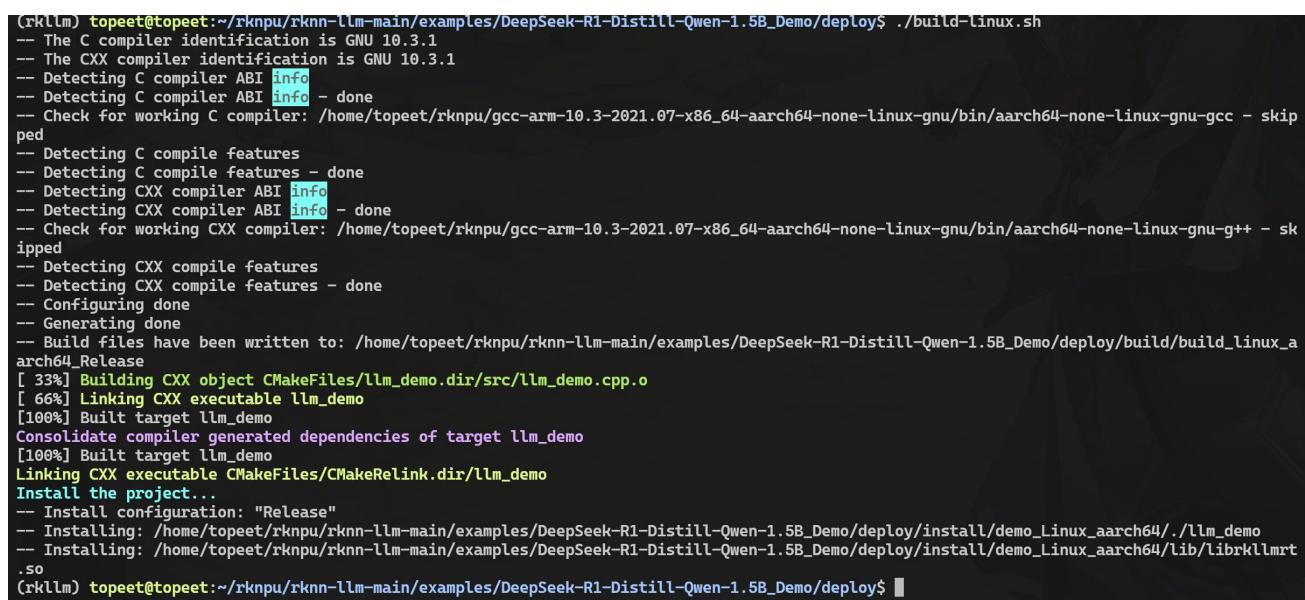
由于要编译该可执行程序需要用到 runtime 目录下的动态库，所以这里直接将“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\01_rkllm-1.1.4](#)”目录下的 rknn-llm-main.zip 拷贝到虚拟机 ubuntu 上，拷贝并解压完成如下图所示：



然后修改 rknn-llm-main/examples/DeepSeek-R1-Distill-Qwen-1.5B_Demo/deploy/ 目录下的 build-linux.sh 编译脚本，将交叉编译器路径修改为前面解压的路径，修改完成如下所示：



保存退出之后，使用“./build-linux.sh”运行脚本，运行成功如下图所示：



编译完成的可执行为存放路径为 install/demo_Linux_aarch64/，如下图所示：

```
(rkllm) topeet@topeet:~/rknpu/rknn-llm-main/examples/DeepSeek-R1-Distill-Qwen-1.5B_Demo/deploy/install/demo_Linux_aarch64$ ls
lib    llm_demo
(rkllm) topeet@topeet:~/rknpu/rknn-llm-main/examples/DeepSeek-R1-Distill-Qwen-1.5B_Demo/deploy/install/demo_Linux_aarch64$
```

迅为也提供了编译好的可执行程序，具体路径为“[iTOP-3588 开发板\02_【iTOP-RK3588 开发板】开发资料\13_NPU 使用配套资料\11_Deepseek 部署测试\04_编译完成的可执行程序](#)”如下图所示：

名称	修改日期	类型	大小
lib	2025/2/14 13:21	文件夹	
llm_demo	2025/2/14 11:19	文件	53 KB

1.5 开发板运行测试

迅为在最新的镜像中，已经将 NPU 驱动升级到了 0.9.8 的版本，所以要想进行本小节的测试，需要确保下载的是迅为最新网盘提供的 Ubuntu 或者 Debian 镜像。本小节的测试镜像为 `ubuntu22_xfce`。

烧写最新的 `ubuntu22` 镜像，启动之后如下所示：

```
System load:  4%          Up time:      0 min
Memory usage: 4% of 7.74G   IP:          192.168.1.241
CPU temp:     50°C         Usage of /:   26% of 22G

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

扩展安全维护 (ESM) Applications 未启用。

169 更新可以立即应用。
这些更新中有 135 个是标准安全更新。
要查看这些附加更新，请运行: apt list --upgradable

37 个额外的安全更新可以通过 ESM Apps 来获取安装。
可通过以下途径了解如何启用 ESM Apps: at https://ubuntu.com/esm

上一次登录: 二 12月 19 02:55:03 UTC 2023 ttyFIQ0 上
root@topeet:~$
```

然后将前面两个小节转换得到的大语言 RKNN 模型和编译得到的可执行文件拷贝到开发板上，拷贝完成如下图所示：

```
root@topeet:/rkllm#
root@topeet:/rkllm# ls
deepseek_1.5b_W8A8_RK3588.rkllm  lib  llm_demo
root@topeet:/rkllm#
```

由于加载大语言模型会打开很多的文件，默认情况每个进程可以同时打开的文件数是有限制的，所以需要在终端输入以下命令，允许当前用户在一个会话中打开最多 102400 个文件描

述符。

```
ulimit -HSn 102400
```

```
root@topeet:/$ ulimit -HSn 102400
root@topeet:/$ █
```

然后使用以下命令，运行可执行程序，并加载 RKLLM 大语言模型，运行成功如下图所示：

```
export LD_LIBRARY_PATH=./lib
./llm_demo deepseek_1.5b_W8A8_RK3588.rkllm 10000 10000
```

```
root@topeet:/rkllm# ./llm_demo deepseek_1.5b_W8A8_RK3588.rkllm 10000 10000
rkllm init start
I rkllm: rkllm-runtime version: 1.1.4, rknpu driver version: 0.9.8, platform: RK3588
rkllm init success
*****可输入以下问题对应序号获取回答/或自定义输入*****
[0] 现有一笼子，里面有鸡和兔子若干只，数一数，共有头14个，腿38条，求鸡和兔子各有多少只？
[1] 有28位小朋友排成一行，从左边开始数第10位是学豆，从右边开始数他是第几位？

*****
user: █
```

现在就可以提出问题了，问题 1：你是什么大语言模型？

```
user: 你是什么大语言模型？
robot: <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。
</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。
user:
```

问题 2：帮我写一个最简单的 helloworld C 语言程序：

```
user: 帮我写一个最简单的C程序
robot: <think>
嗯，用户让我帮他写一个最简单的C程序。首先，我得考虑他的需求是什么。他可能刚接触编程，或者只是想简单地测试一下自己的能力。
那我应该选择什么问题呢？最简单的可能是计算平方根，因为这样容易实现，并且结果明确。比如，输入一个数，输出它的平方根。
接下来，我需要确定程序的结构。通常，C程序包括头文件、函数定义和主函数。所以，我会先写头文件，比如#include <stdio.h>，然后定义一个函数sqrt，它接受一个double参数，并返回double类型的结果。
然后是主函数，这里会读取输入，调用sqrt函数计算平方根，最后输出结果。需要注意的是，输入的数必须是非负的，否则可能会出错。所以，在程序中我应该加上判断语句，确保输入值大于等于零。
另外，用户可能希望这个程序简洁，所以我不会添加太多注释，只提供必要的代码部分。这样用户可以直接复制粘贴进去运行。
最后，我要测试一下程序是否正确。比如，输入9，输出3；输入25，输出5等等。如果有语法错误的话，我会检查是否有语法错误或者逻辑问题。
总的来说，这个程序应该能够满足用户的需求，同时保持简单易懂。
</think>

当然可以！以下是一个非常简单的C程序，用于计算一个数的平方根：

```c
#include <stdio.h>
#include <math.h>

double squareRoot(double number) {
 return sqrt(number);
}

int main() {
 double num;
 printf("Enter a non-negative number: ");
 scanf("%lf", &num);

 double result = squareRoot(num);
 printf("Square root of %.2lf is %.2lf\n", num, result);

 return 0;
}
```

这个程序的功能是：
1. 定义了一个名为`squareRoot`的函数，该函数接受一个双精度浮点数参数，并返回其平方根的双精度浮点数。
2. 在主函数中，使用`sqrt`函数计算输入数的平方根。需要注意的是，`sqrt`函数需要从`<math.h>`头文件中包含。
3. 读取用户输入并进行验证（确保输入值为非负数）。
4. 输出结果。

你可以运行这个程序如下：
```bash
$./a
Enter a non-negative number:
10
Square root of 10.00 is 3.16
```
### 注意事项：
- 这个程序假设输入的数是非负的。如果需要处理负数的情况，可以添加额外的条件判断。
```

可以看到 DeepSeek 大语言模型给出了非常准确的回答，大家可以自行提问，至此，关于 DeepSeek 大语言模型测试就完成了。