# 书生二期第四课笔记

复习链接：

视频链接：https://www.bilibili.com/video/BV15m421j78d/?spm_id_from=333.788

课程文档：https://github.com/InternLM/Tutorial/blob/camp2/xtuner/readme.md

作业文档：https://github.com/InternLM/Tutorial/blob/camp2/xtuner/homework.md

OpenMMLab bilibili

# 目录

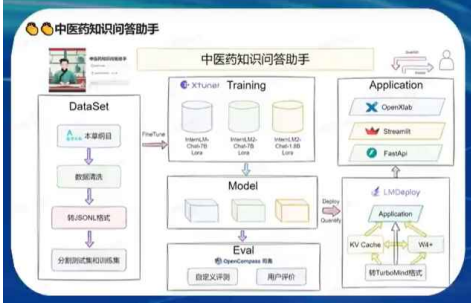


OpenMMLab bilibili

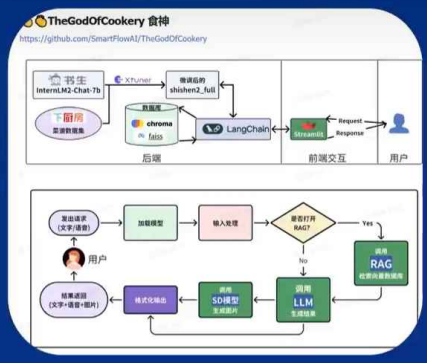
# 一、Finetune 简介

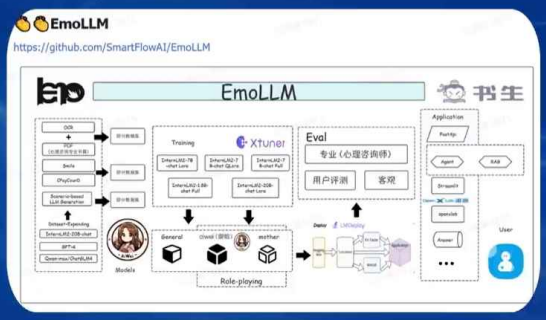
- 两种Finetune范式
- 一条数据的一生

# 为什么要微调?









OpenMMLab

# 两种Finetune范式

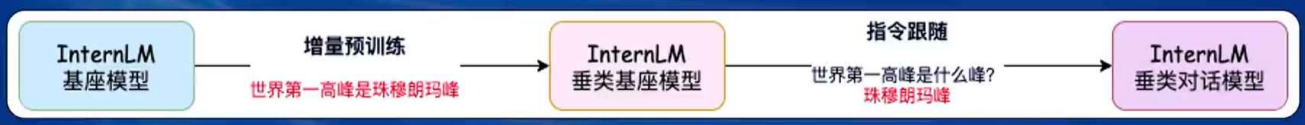
LLM 的下游应用中，**增量预训练**和**指令跟随**是经常会用到两种的微调模式

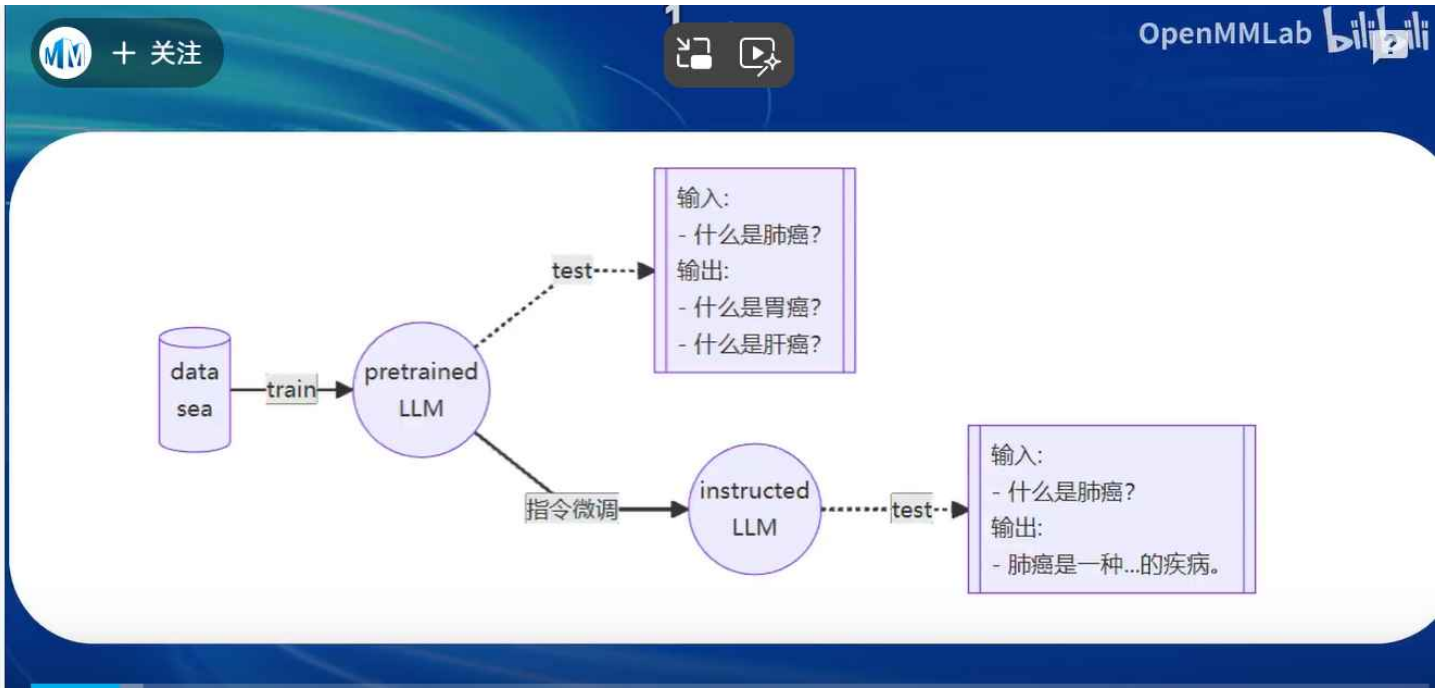
**增量预训练微调**

使用场景：让基座模型学习到一些新知识，如某个垂类领域的常识
训练数据：文章、书籍、代码等

**指令跟随微调**

使用场景：让模型学会对话模板，根据人类指令进行对话
训练数据：高质量的对话、问答数据

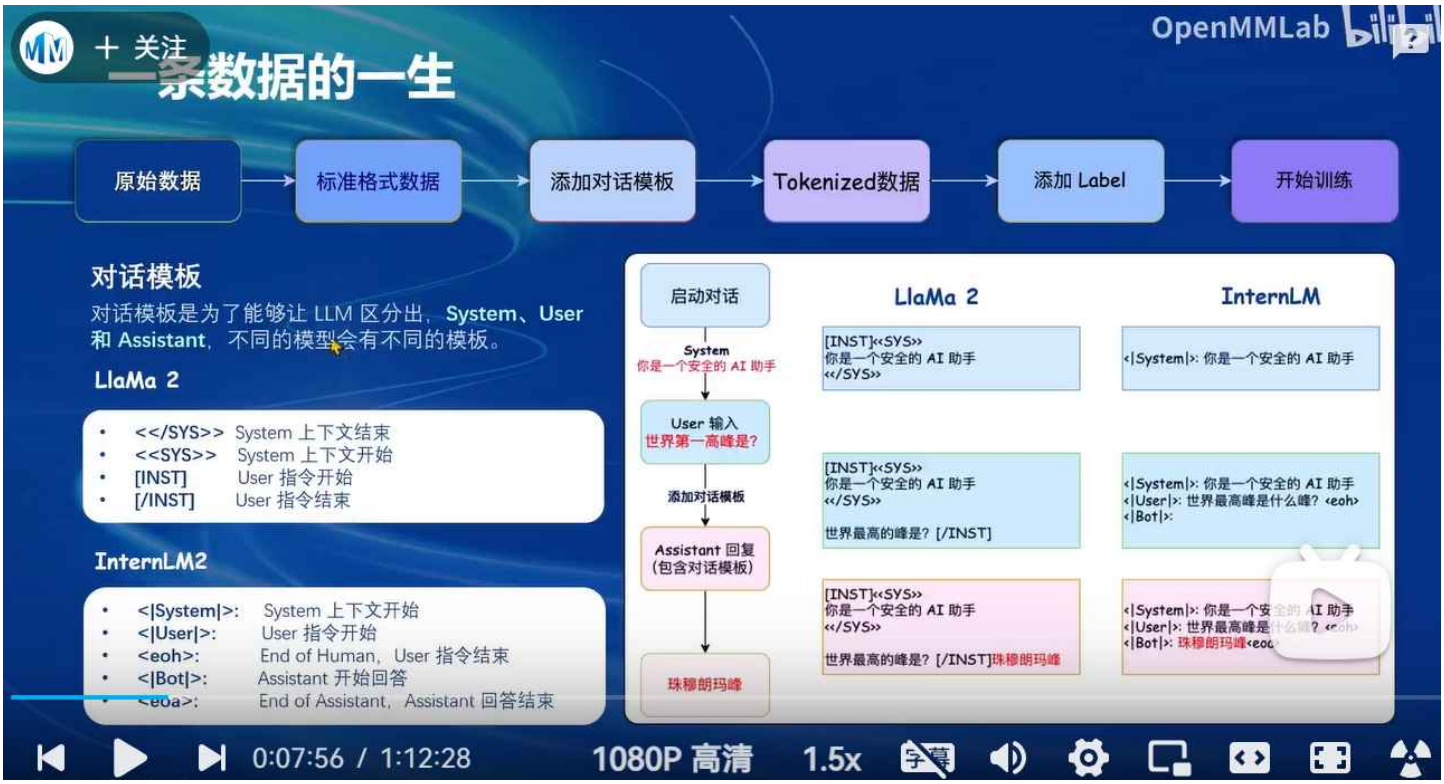


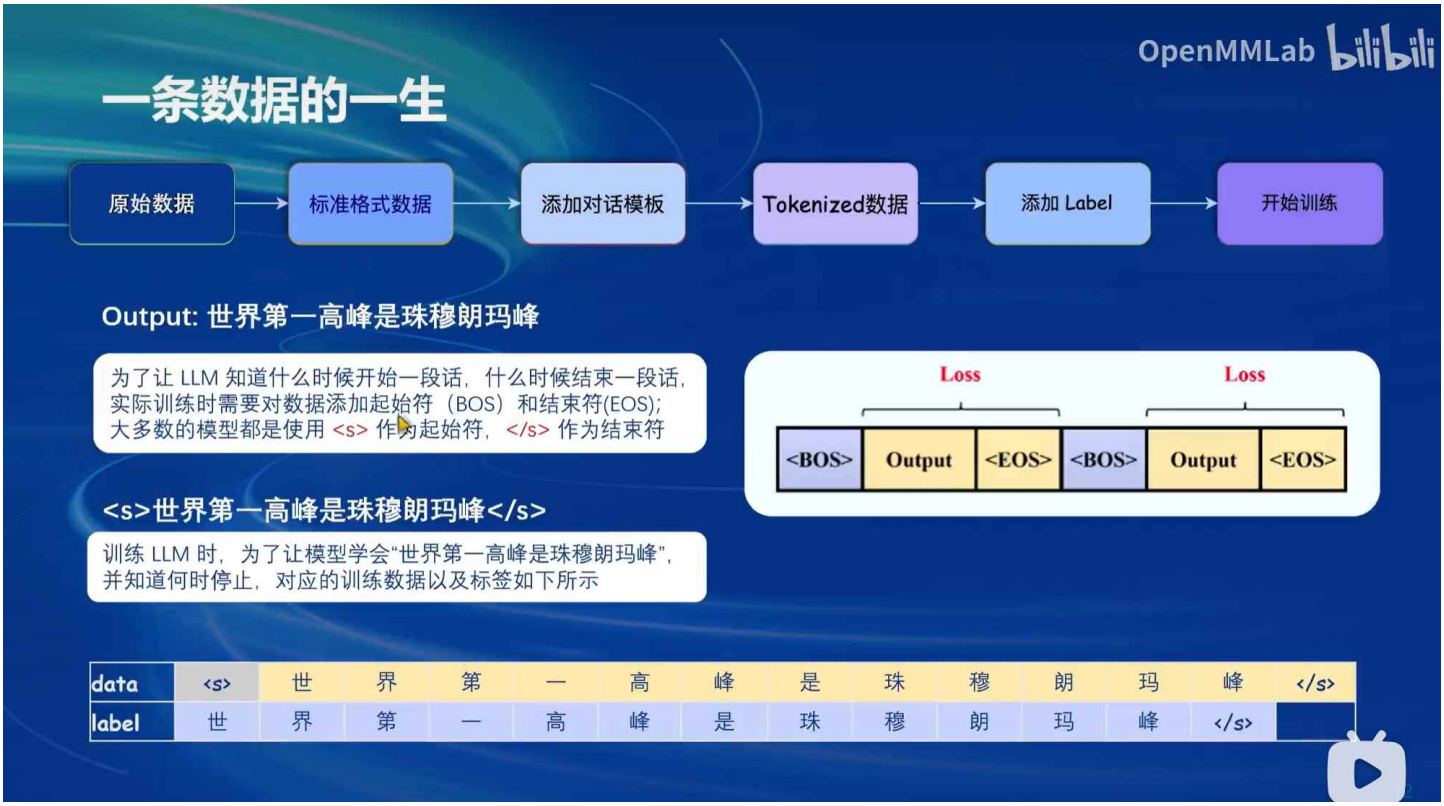

+ 关注
OpenMMLab
data sea
train
pretrained LLM
test
输入:
- 什么是肺癌?
输出:
- 什么是胃癌?
- 什么是肝癌?
指令微调
instructed LLM
test
输入:
- 什么是肺癌?
输出:
- 肺癌是一种...的疾病。


+ 关注
一条数据的一生
原始数据
标准格式数据
添加对话模板
Tokenized数据
添加 Label
开始训练
###System:你是一名友好的AI助手。
###User:你好!
###Assistant:您好，我是一名AI助手，请问有什么可以帮助您?
###User:世界最高峰是什么峰?
###Assistant:世界最高峰是珠穆朗玛峰。
0:06:12 / 1:12:28
1080P 高清
1.5x


+ 关注
一条数据的一生
OpenMMLab
原始数据
标准格式数据
添加对话模板
Tokenized数据
添加 Label
开始训练
###System:你是一名友好的AI助手。
###User:你好!
###Assistant:您好，我是一名AI助手，请问有什么可以帮助您?
###User:世界最高峰是什么峰?
###Assistant:世界最高峰是珠穆朗玛峰。
[{
"conversation": [
{
"system": "你是一名友好的AI助手。",
"input": "你好！",
"output": "您好，我是一名AI助手，请问有什么可以帮助您？"
},
{
"input": "世界最高峰是什么峰？",
"output": "世界最高峰是珠穆朗玛峰。"
}
]
}]
<|System|>:你是一名友好的AI助手。
<|User|>:你好!
<|Bot|>:您好，我是一名AI助手，请问有什么可以帮助您?
<|User|>:世界最高峰是什么峰?
<|Bot|>:世界最高峰是珠穆朗玛峰。

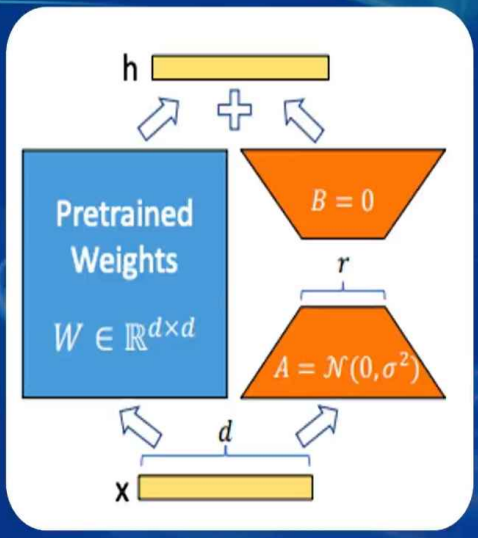

+ 关注
一条数据的一生
OpenMMLab
原始数据
标准格式数据
添加对话模板
Tokenized数据
添加 Label
开始训练
double enter to end input >>> shi jie di yi gao feng shi shen me feng
1 世界第一高峰是什么峰 2 世界第一高峰
在实际对话时，通常会有三种角色
• System 给定一些上下文信息，比如"你是一个安全的 AI 助手"
• User 实际用户，会提出一些问题，比如"世界第一高峰是？"
• Assistant 根据 User 的输入，结合 System 的上下文信息，做出回答，比如"珠穆朗玛峰"
在使用对话模型时，通常是不会感知到这三种角色的
启动对话
System
你是一个安全的 AI 助手
User 输入
世界第一高峰是?
添加对话模板
Assistant 回复
(包含对话模板)
显示没有对话模版的回答
珠穆朗玛峰
什么是对话模板?


+ 关注
一条数据的一生
OpenMMLab
原始数据
标准格式数据
添加对话模板
Tokenized数据
添加 Label
开始训练
对话模板
对话模板是为了能够让 LLM 区分出，System、User 和 Assistant，不同的模型会有不同的模板。
LlaMa 2
<<SYS>> System 上下文结束
<<SYS>> System 上下文开始
[INST] User 指令开始
[/INST] User 指令结束
InternLM2
<|System|>: System 上下文开始
<|User|>: User 指令开始
<eoh>: End of Human，User 指令结束
<|Bot|>: Assistant 开始回答
<eoa>: End of Assistant，Assistant 回答结束
启动对话
System
你是一个安全的 AI 助手
User 输入
世界第一高峰是?
添加对话模板
Assistant 回复
(包含对话模板)
珠穆朗玛峰
LlaMa 2
InternLM
[INST]<<SYS>>
你是一个安全的 AI 助手
<</SYS>>
<|System|>: 你是一个安全的 AI 助手
[INST]<<SYS>>
你是一个安全的 AI 助手
<</SYS>>
世界最高的峰是? [/INST]
<|System|>: 你是一个安全的 AI 助手
<|User|>: 世界最高峰是什么峰? <eoh>
<|Bot|>:
[INST]<<SYS>>
你是一个安全的 AI 助手
<</SYS>>
世界最高的峰是? [/INST]珠穆朗玛峰
<|System|>: 你是一个安全的 AI 助手
<|User|>: 世界最高峰是
<|Bot|>: 珠穆朗玛峰<eoa>
0:07:56 / 1:12:28
1080P 高清
1.5x


OpenMMLab
一条数据的一生
原始数据
标准格式数据
添加对话模板
Tokenized数据
添加 Label
开始训练
Output: 世界第一高峰是珠穆朗玛峰
为了让 LLM 知道什么时候开始一段话，什么时候结束一段话，实际训练时需要对数据添加起始符（BOS）和结束符(EOS)；大多数的模型都是使用 <s> 作为起始符，</s> 作为结束符
Loss
Loss
<BOS>
Output
<EOS>
<BOS>
Output
<EOS>
<s>世界第一高峰是珠穆朗玛峰</s>
训练 LLM 时，为了让模型学会"世界第一高峰是珠穆朗玛峰"，并知道何时停止，对应的训练数据以及标签如下所示
data
<s>
世
界
第
一
高
峰
是
珠
穆
朗
玛
峰
</s>
label
世
界
第
一
高
峰
是
珠
穆
朗
玛
峰
</s>

# LoRA & QLoRA

## LoRA: LOW-RANK ADAPTATION OF LARGE LANGUAGE MODELS




LLM 的参数量主要集中在模型中的 Linear，训练这些参数会耗费大量的显存

LoRA 通过在原本的 Linear 旁，新增一个支路，包含两个连续的小 Linear，新增的这个支路通常叫做 Adapter

Adapter 参数量远小于原本的 Linear，能大幅降低训练的显存消耗

**微调原理**

> 想象一下，你有一个超大的玩具，现在你想改造这个超大的玩具。但是，**对整个玩具进行全面的改动会非常昂贵。**

※ 因此，你找到了一种叫 **LoRA** 的方法：**只对玩具中的某些零件进行改动，而不是对整个玩具进行全面改动。**

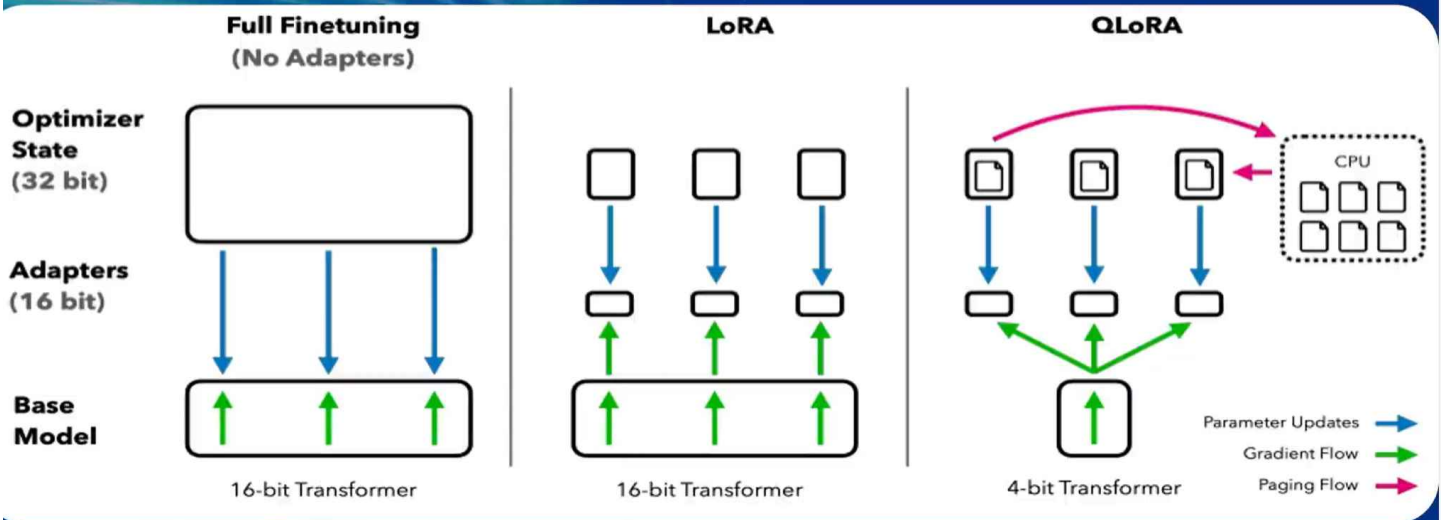
※ 而 **QLoRA** 是 LoRA 的一种改进：如果你手里只有一把生锈的螺丝刀，也能改造你的玩具。

- **Full:** 😳 → 🚲
- **LoRA:** 😳 → 🚲
- **QLoRA:** 😳 → 🚲





# LoRA & QLoRA




- Base Model 参与训练并更新参数
- 需要保存 Base Model 中参数的优化器状态

- Base Model 只参与 Forward
- 只有 Adapter 部分 Backward 更新参数
- 只需保存 Adapter 中参数的优化器状态

- Base Model 量化为 4-bit
- 优化器状态在 CPU 与 GPU 间 Offload
- Base Model 只参与 Forward
- 只有 Adapter 部分 Backward 更新参数
- 只需保存 Adapter 中参数的优化器状态

# 二、XTuner

- **傻瓜化：**以 配置文件 的形式封装了大部分微调场景，**0基础的非专业人员也能一键开始微调。**
- **轻量级：**对于 7B 参数量的LLM，**微调所需的最小显存仅为 8GB：**消费级显卡✅，colab✅

OpenMMLab

## XTuner 简介

### 功能亮点

**适配多种生态**

- **多种微调算法**
  多种微调策略与算法，覆盖各类 SFT 场景
- **适配多种开源生态**
  支持加载 HuggingFace、ModelScope 模型或数据集
- **自动优化加速**
  开发者无需关注复杂的显存优化与计算加速细节

**适配多种硬件**

- **训练方案覆盖 NVIDIA 20 系以上所有显卡**
- **最低只需 8GB 显存即可微调 7B 模型**

### XTuner 技术架构图

# LLaMa-Factory vs XTuner


# XTuner 快速上手

**1. 安装**

```
pip install xtuner
```

**2. 挑选配置模板**

```
xtuner list-cfg -p internlm_20b
```

**3. 一键训练**

```
xtuner train internlm_20b_qlora_oasst1_512_e3
```

**Config 命名规则**

| | | |
|---|---|---|
| 模型名 | internlm_20b | 无 chat 代表是基座模型 |
| 使用算法 | qlora | |
| 数据集 | oasst1 | |
| 数据长度 | 512 | |
| Epoch | e3，epoch 3 | |

```
==========CONFIGS==========
PATTERN: internlm_20b
------------------------------
internlm_20b_chat_qlora_alpaca_e3
internlm_20b_chat_qlora_alpaca_enzh_e3
internlm_20b_chat_qlora_alpaca_enzh_oasst1_e3
internlm_20b_chat_qlora_alpaca_zh_e3
internlm_20b_chat_qlora_code_alpaca_e3
internlm_20b_chat_qlora_lawyer_e3
internlm_20b_chat_qlora_oasst1_512_e3
internlm_20b_chat_qlora_oasst1_e3
internlm_20b_chat_qlora_open_platypus_e3
internlm_20b_qlora_alpaca_e3
internlm_20b_qlora_alpaca_enzh_e3
internlm_20b_qlora_alpaca_enzh_oasst1_e3
internlm_20b_qlora_alpaca_zh_e3
internlm_20b_qlora_arxiv_gentitle_e3
internlm_20b_qlora_code_alpaca_e3
internlm_20b_qlora_colorist_e5
internlm_20b_qlora_lawyer_e3
internlm_20b_qlora_oasst1_512_e3
internlm_20b_qlora_oasst1_e3
internlm_20b_qlora_open_platypus_e3
internlm_20b_qlora_sql_e3
==========================
```

# XTuner 快速上手

## 1. 拷贝配置模板

```
xtuner copy-cfg internlm_20b_qlora_oasst1_512_e3 ./
```

## 2. 修改配置模板

```
vi internlm_20b_qlora_oasst1_512_e3_copy.py
```

## 3. 启动训练

```
xtuner train internlm_20b_qlora_oasst1_512_e3_copy.py
```

## 常用超参

| | |
|---|---|
| data_path | 数据路径或 HuggingFace 仓库名 |
| max_length | 单条数据最大 Token 数，超过则截断 |
| pack_to_max_length | 是否将多条短数据拼接到 max_length，提高 GPU 利用率 |
| accumulative_counts | 梯度累积，每多少次 backward 更新一次参数 |
| evaluation_inputs | 训练过程中，会根据给定的问题进行推理，便于观测训练状态 |
| evaluation_freq | Evaluation 的评测间隔 iter 数 |

## 自定义训练

```python
#######################################################################
#                          PART 1  Settings                           #
#######################################################################
# Model
pretrained_model_name_or_path = 'internlm/internlm-20b'

# Data
data_path = 'timdettmers/openassistant-guanaco'
prompt_template = PROMPT_TEMPLATE.openassistant
max_length = 2048
pack_to_max_length = True

# Scheduler & Optimizer
batch_size = 1  # per_device
accumulative_counts = 16
dataloader_num_workers = 0
max_epochs = 3
optim_type = PagedAdamW32bit
lr = 2e-4
betas = (0.9, 0.999)
weight_decay = 0
max_norm = 1  # grad clip

# Evaluate the generation performance during the training
evaluation_freq = 500
evaluation_inputs = [
    '请给我介绍五个上海的景点', 'Please tell me five scenic spots in Shanghai'
]
```



# XTuner 快速上手

## 对话

为了便于开发者查看训练效果，Xtuner 提供了一键对话接口

```
double enter to end input >>>
```

## Float 16 模型对话

```
xtuner chat internlm/internlm-chat-20b
```

## 4bit 模型对话

```
xtuner chat internlm/internlm-chat-20b --bits 4
```

## 加载 Adapter 模型对话

```
xtuner chat internlm/internlm-chat-20b --adapater $ADAPTER_DIR
```

# XTuner 快速上手

OpenMMLab

## 对话

为了便于开发者查看训练效果，
Xtuner 提供了一键对话接口

```
double enter to end input >>>
```

### Float 16 模型对话

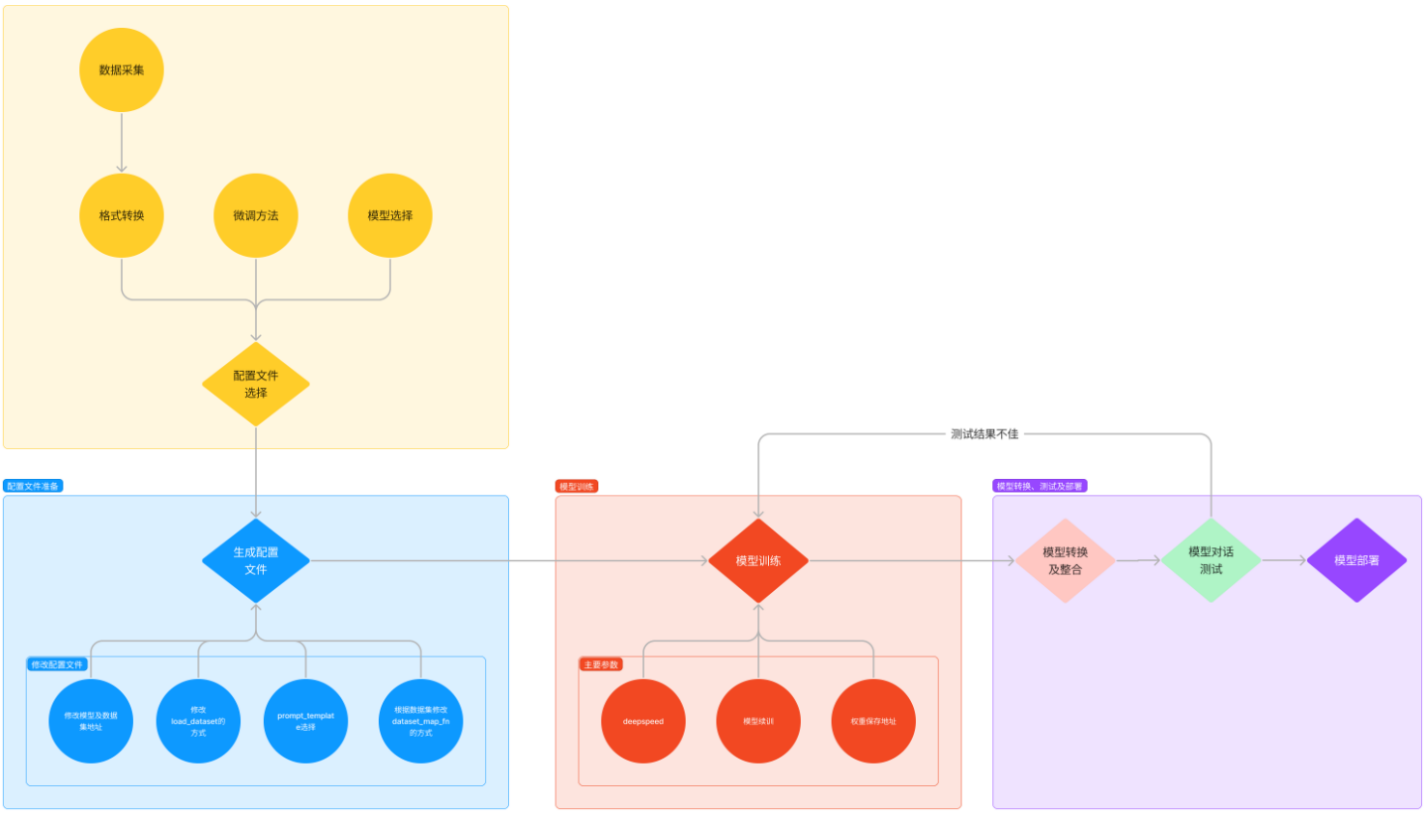
```
xtuner chat internlm/internlm-chat-20b
```

### 4bit 模型对话

```
xtuner chat internlm/internlm-chat-20b --bits 4
```

### 加载 Adapter 模型对话

```
xtuner chat internlm/internlm-chat-20b --adapater $ADAPTER_DIR
```



# XTuner 快速上手

XTuner 还支持工具类模型的对话，更多详见 HuggingFace Hub（xtuner/Llama-2-7b-qlora-moss-003-sft）

| Q: 上海明天的天气怎么样？ | Q: 一个球体的半径是5.32厘米，求它的表面积和体积。 | Q: 今有鸡兔同笼，上有二十头，下有六十二足，问鸡兔各几何？ |
| --- | --- | --- |
| 1. Think | 1. Think | 1. Think |
| Bot Thoughts:<br>为了 | Bot Thoughts:<br>这是 | Bot Thoughts:<br>这是一 |
| 联网搜索 | 使用计算器 | 解方程 |

# XTuner 数据引擎

## 数据处理流程

### 1. 原始问答对 → 格式化问答对

```
###System:你是一名友好的AI助手。
###User:你好!
###Assistant:您好，我是一名AI助手，请问有什么可以帮助您?
###User:世界最高峰是什么峰?
###Assistant:世界最高峰是珠穆朗玛峰。
```

数据集映射函数

```
[{
  "conversation": [
    {
      "system": "你是一名友好的AI助手。",
      "input": "你好! ",
      "output": "您好，我是一名AI助手，请问有什么可以帮助您? "
    },
    {
      "input": "世界最高峰是什么峰? ",
      "output": "世界最高峰是珠穆朗玛峰。"
    }
  ]
}]
```

对话模板映射函数

### 2. 格式化问答对 → 可训练语料

```
<|System|>:你是一名友好的AI助手。
<|User|>:你好!
<|Bot|>:您好，我是一名AI助手，请问有什么可以帮助您?
<|User|>:世界最高峰是什么峰?
<|Bot|>:世界最高峰是珠穆朗玛峰。
```

蓝色代表有训练 Loss 的部分

# XTuner 数据引擎

**数据集映射函数**

XTuner 内置了多种热门数据集的映射函数

| | |
|---|---|
| alpaca_map_fn | Alpaca 格式数据集处理函数 |
| oassti_map_fn | OpenAssistant 格式数据处理函数 |
| openai_map_fn | OpenAI Finetune 格式数据处理函数 |
| wizardlm | 微软 WizardLM 系列数据集梳理函数 |
| ..... | |

**对话模板映射函数**

XTuner 内置了多种对话模板映射函数

| | |
|---|---|
| chatglm | ChatGLM 使用的对话模板 |
| llama2_chat | Llama2 对话模型使用的对话模板 |
| code_llama_chat | Coda Llama 使用的对话模板 |
| baichuan_chat<br>baichuan2_chat | 百川使用的对话模板 |
| qwen_chat | 通义千问使用的对话模板 |
| ..... | |

开发者可以专注于数据内容
不必花费精力处理复杂的数据格式！

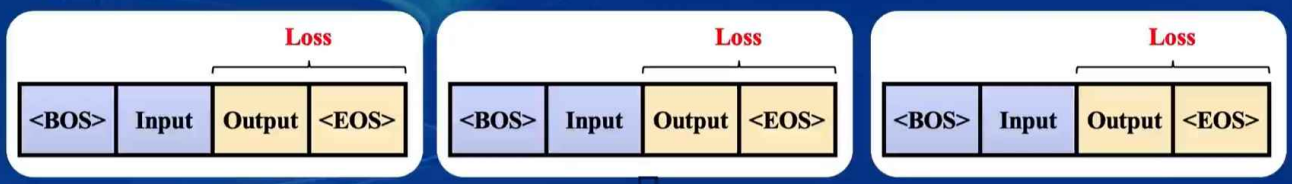




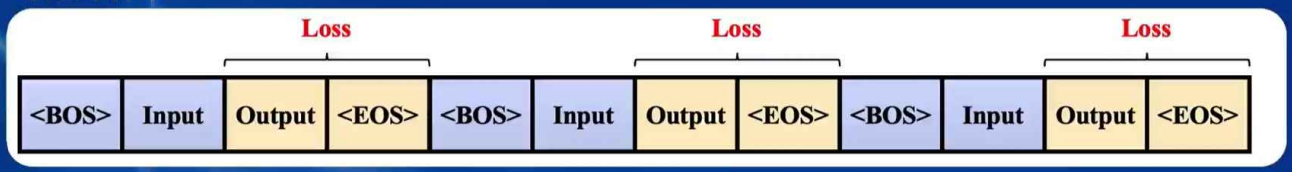
# XTuner 数据引擎

## 多数据样本拼接（Pack Dataset）

3条原始



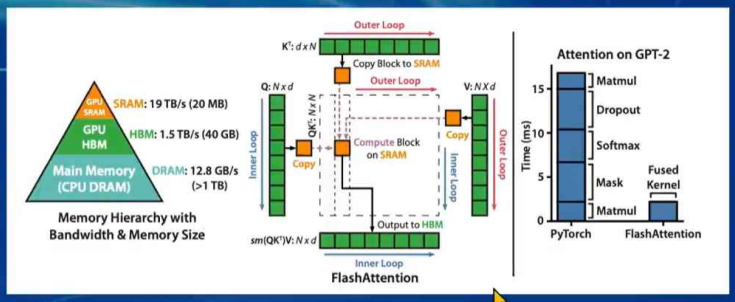

1条训练样本




增强并行性，充分利用GPU资源！

# 8GB 显存玩转 LLM

Flash Attention 和 DeepSpeed ZeRO 是 XTuner 最重要的两个优化技巧



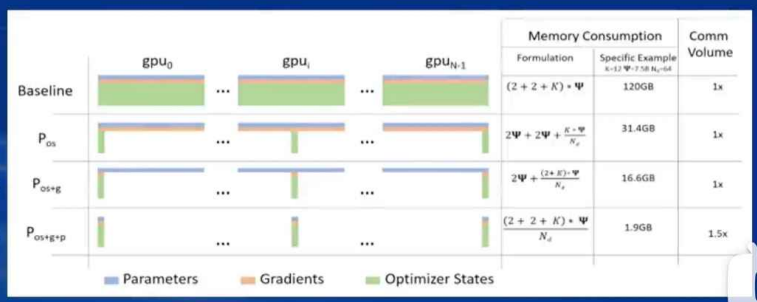

## Flash Attention

Flash Attention 将 Attention 计算并行化，避免了计算过程中 Attention Score NxN 的显存占用（训练过程中的 N 都比较大）

## DeepSpeed ZeRO

ZeRO 优化，通过将训练过程中的参数、梯度和优化器状态切片保存，能够在多 GPU 训练时显著节省显存

除了将训练中间状态切片外，DeepSpeed 训练时使用 FP16 的权重，相较于 Pytorch 的 AMP 训练，在单 GPU 上也能大幅节省显存




OpenMMLab bilibili

# 8GB 显存玩转 LLM

DeepSpeed 和 Flash Attention 虽然能够大幅降低训练成本，但使用门槛相对较高，需要复杂的配置，甚至修改代码

为了让开发者专注于数据，XTuner 会自动 dispatch Flash Attention，并一键启动 DeepSpeed ZeRO

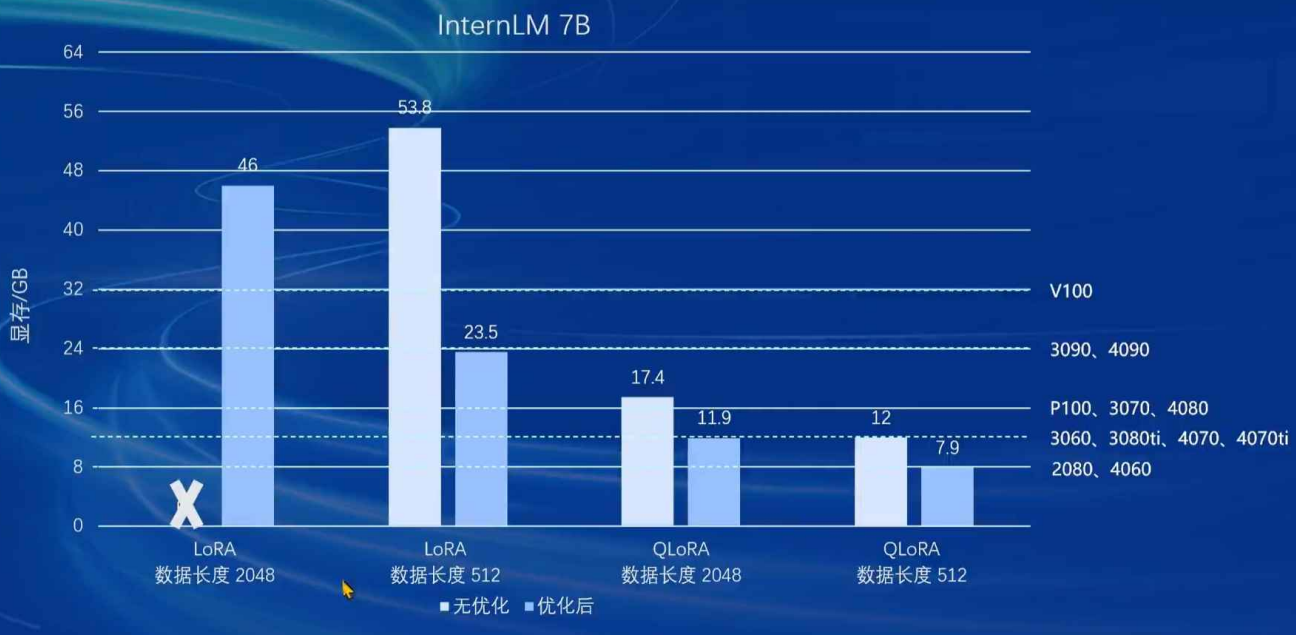
```
{
  "gradient_accumulation_steps": "auto",
  "train_micro_batch_size_per_gpu": "auto",
  "gradient_clipping": "auto",
  "zero_allow_untested_optimizer": true,
  "zero_force_ds_cpu_optimizer": false,
  "zero_optimization": {
    "stage": 3,
    "contiguous_gradients": false,
    "allgather_bucket_size": 3e8,
    "reduce_bucket_size": 3e8,
    "overlap_comm": true,
    "reduce_scatter": true,
    "stage3_gather_16bit_weights_on_model_save": true
  },
  "low_cpu_mem_usage": false,
  "fp16": {
    "enabled": true,
    "initial_scale_power": 16
  }
}
```

ZeRO 3 配置

```
xtuner train internlm_20b_qlora_oasst1_512_e3 \
  --deepspeed deepspeed_zero3
```

OpenMMLab

# 8GB 显存玩转 LLM




OpenMMLab

# InternLM2 1.8B 模型

**为了响应社区用户极其强烈的呼声，InternLM2-1.8B 于近日正式开源！要说这呼声多强烈，有 issue 截图为证。**


**InternLM2-1.8B 提供了三个版本的开源模型，大家可以按需选择。**

- InternLM2-1.8B: 具有高质量和高适应灵活性的基础模型，为下游深度适应提供了良好的起点。
- InternLM2-Chat-1.8B-SFT: 在 InternLM2-1.8B 上进行监督微调 (SFT) 后得到的对话模型。
- **InternLM2-Chat-1.8B: 通过在线 RLHF 在 InternLM2-Chat-1.8B-SFT 之上进一步对齐。InternLM2-Chat-1.8B 表现出更好的指令跟随、聊天体验和函数调用，推荐下游应用程序使用。(模型大小仅为3.78GB)**

在 FP16 精度模式下，**InternLM2-1.8B 仅需 4GB 显存的笔记本显卡即可顺畅运行。拥有 8GB 显存的消费级显卡，即可轻松进行 1.8B 模型的微调工作。**如此低的硬件门槛，非常适合初学者使用，以深入了解和掌握大模型的全链路。

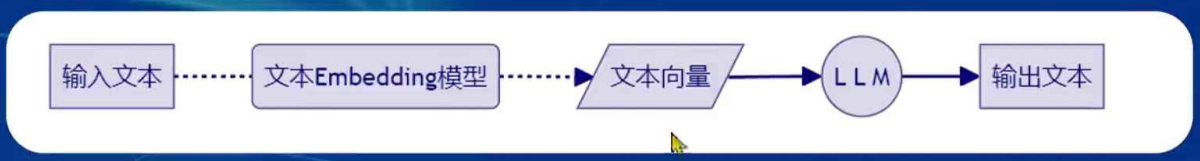
# 五、多模态LLM

- 给LLM装上电子眼：多模态LLM原理简介
- 什么型号的电子眼：LLaVA方案简介
- 快速上手：InternLM2_Chat_1.8B + LLaVA



## 给LLM装上电子眼：多模态LLM原理简介

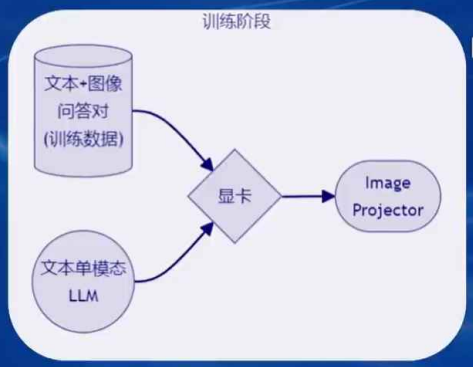
文本单模态



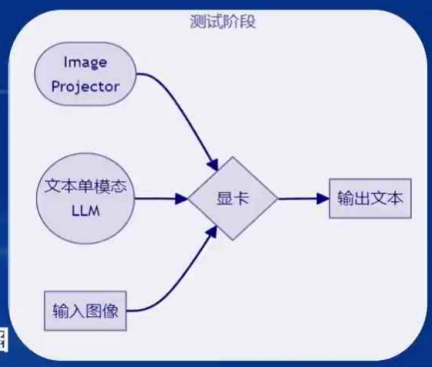

文本+图像多模态

# 什么型号的电子眼：LLaVA方案简介

- Haotian Liu 等使用 GPT-4V 对图像数据生成描述，以此构建出大量 <question text><image> -- <answer text>的数据对。
- 利用这些数据对，配合文本单模态 LLM，训练出一个 Image Projector。
- 所使用的文本单模型 LLM 和训练出来的 Image Projector，统称为 LLaVA 模型。




LLaVA训练阶段示意图




LLaVA测试阶段示意图



OpenMMLab

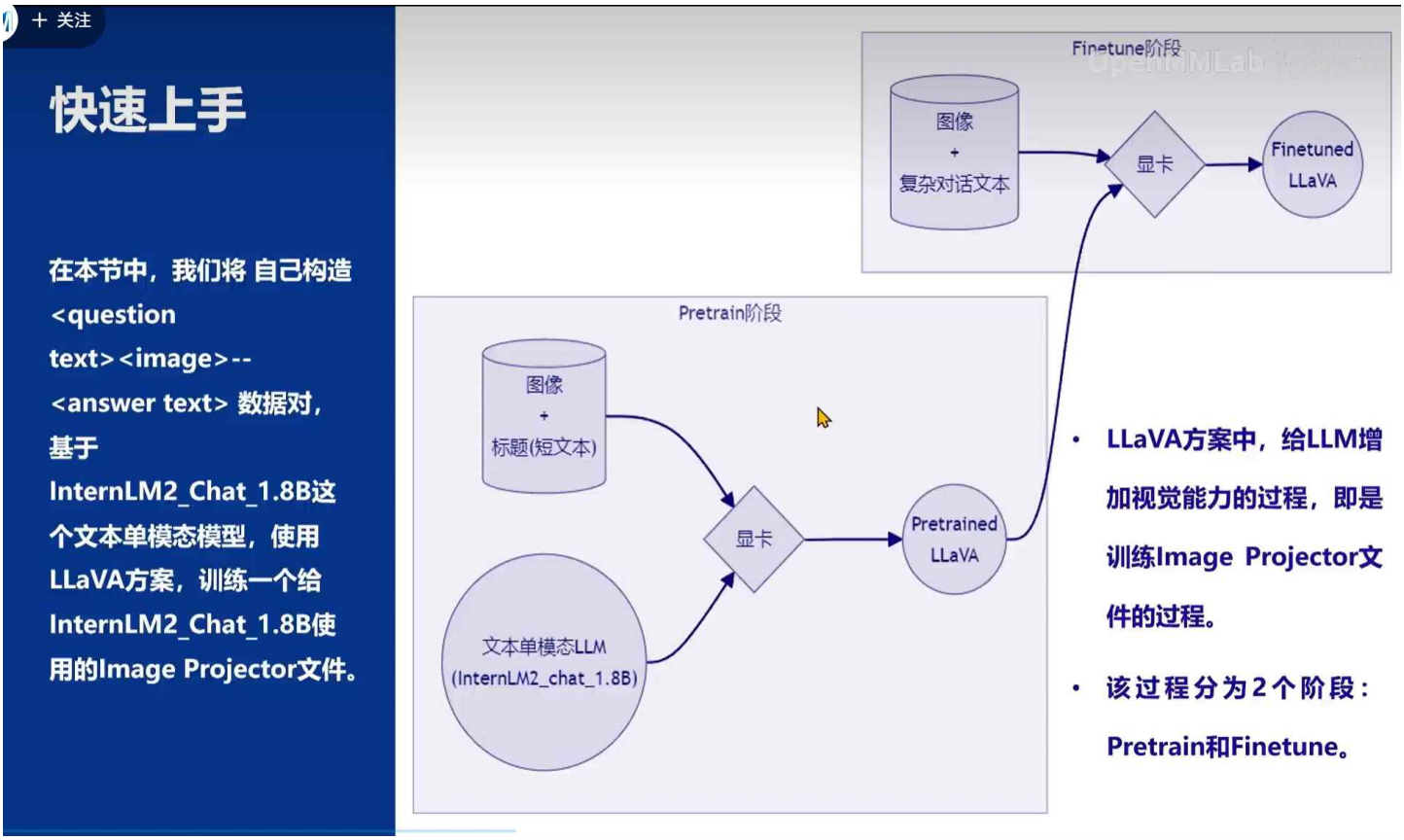
# 什么型号的电子眼：LLaVA方案简介

Image Projector 的训练和测试，有点类似之前我们讲过的 LoRA 微调方案。

二者都是在已有LLM的基础上，用新的数据训练一个新的小文件。

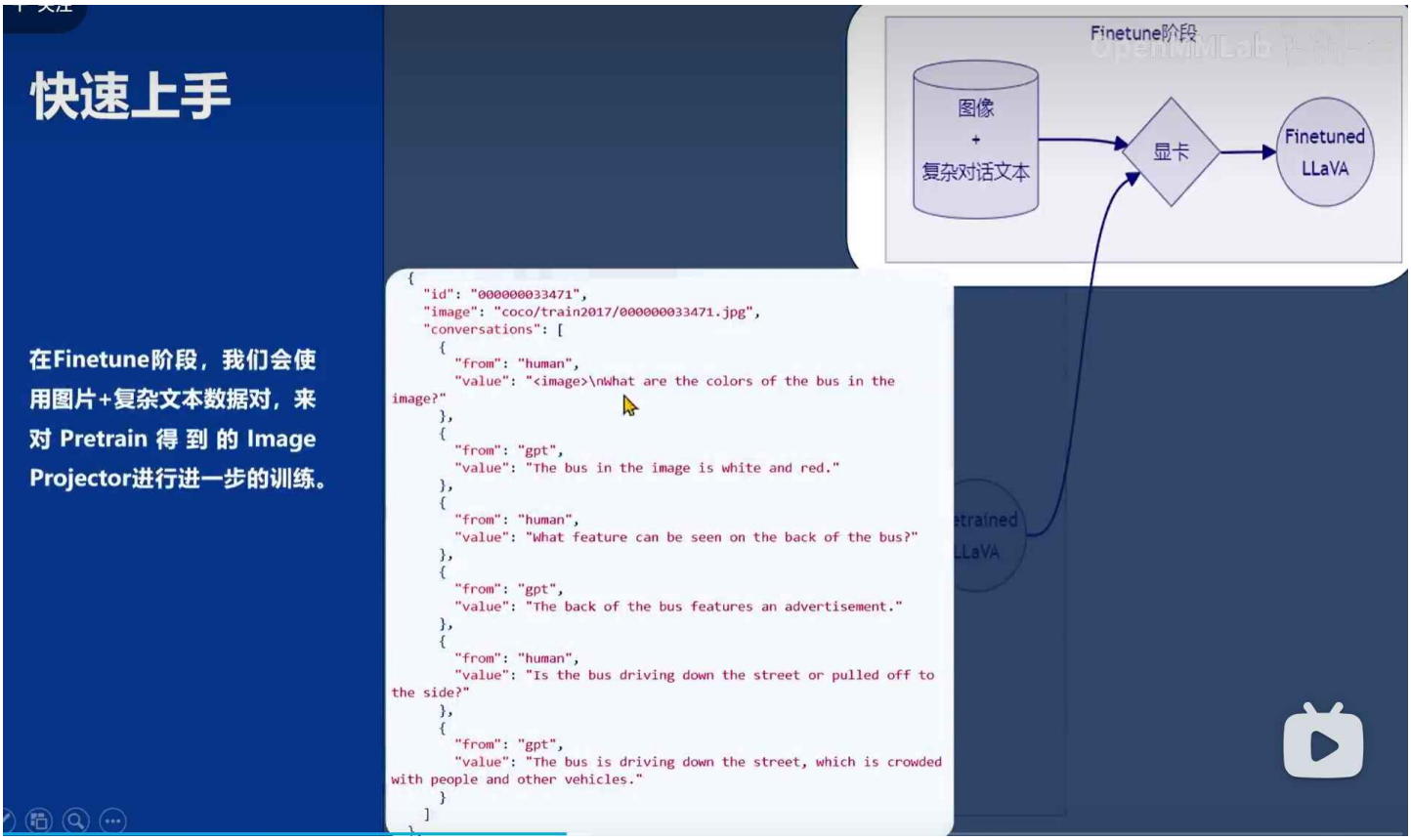
只不过，LLM 套上 LoRA 之后，有了新的灵魂（角色）；

而 LLM 套上 Image Projector 之后，才有了眼睛。




pretrian阶段的数据样子：

```
[
  {
    "id": "000000033471",
    "image": "coco/train2017/000000033471.jpg",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nDescribe this image."
      },
      {
        "from": "gpt",
        "value": "The bus in the image is white and red."
      }
    ]
  },
  {
    "id": "000000052846",
    "image": "coco/train2017/000000052846.jpg",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nDescribe this image."
      },
      {
        "from": "gpt",
        "value": "The cat is positioned on top of the back of the couch in the living room."
      }
    ]
  },
  {
    "id": "000000052555",
    "image": "coco/train2017/000000052833.jpg",
    "conversations": [
      {
        "from": "human",
        "value": "<image>\nDescribe this image."
      },
      {
        "from": "gpt",
        "value": "The cat couch in the bathroom."
      }
    ]
  }
]
```




finetune阶段的图片会有一系列的问答对在后面接着。

个人助理的微调：

# 动手实战环节—VLM

OpenMMLab

在本节中，我们将 自己构造 <question text><image>--<answer text> 数据对，基于InternLM2_Chat_1.8b这个文本单模态模型，使用LLaVA方案，将InternLM2_Chat_1.8b训练成具有识图能力的文本图像双模态模型。

示例图像

Finetune前的多模态LLM

(InternLM_Chat_1.8B_llava):

**只会给图像打标题**

```
double enter to end input (EXIT: exit chat, RESET: reset history) >>> Describe this image.

a doctor and a woman looking at a vision test<|im_end|>

double enter to end input (EXIT: exit chat, RESET: reset history) >>> What is the equipment in
the image?

a doctor and a woman looking at a vision test<|im_end|>
```

# 动手实战环节—VLM

OpenMMLab

在本节中，我们将 自己构造 <question text><image>--<answer text> 数据对，基于InternLM2_Chat_1.8B 这个文本单模态模型，使用 LLaVA 方案，将 InternLM2_Chat_1.8B 训练成具有识图能力的文本图像双模态模型。

示例图像

Finetune后的多模态LLM

(InternLM_Chat_1.8B_llava):

**会根据图像回答问题了**

```
double enter to end input (EXIT: exit chat, RESET: reset history) >>> Describe this image.

This is a photograph of a patient undergoing an eye examination. A healthcare professional, pos
sibly an optometrist, is using a slit lamp to examine the patient's eyes. The patient is seated
, leaning into the machine which has a chin rest support and a forehead support. It is a a fore
head support and has a support.<|im_end|>

double enter to end input (EXIT: exit chat, RESET: reset history) >>> What is the equipment in
the image?

The image shows a slit lamp, which is a light source that can be focused to shine a thin sheet
of light into the eye. It is used with a biomicroscope and facilitates the examination of the e
ye's anterior and posterior segments under magnification.<|im_end|>
```