**《大模型算法与实践》课程作业报告**

**一、任务背景**

简要介绍本次作业的任务目标：

* 使用 SFT 或 RLHF 方法完成大模型适配与验证。
* 说明所选方法的意义与场景应用。
* 简述所用大模型的背景（如 Qwen、LLaMA、ChatGLM、BERT 等）及其用途。

在日常使用大模型进行学术研究和论文搜索过程中，常出现“幻觉”（hallucination）问题，表现为模型生成了虚构或不真实的论文标题、引用信息，影响了研究效率与准确性。

因此，本项目聚焦于构建一个更可靠的论文索引问答大模型，旨在通过 SFT（Supervised Fine-Tuning）方法，在高质量学术问答数据上微调基础模型，使其在处理与学术文献相关的问题时，能够生成更真实、更准确的回答，从而提升科研助手类模型的可用性与可靠性。

1. **数据准备与处理**

2.1 数据来源

来源：Hugging Face Datasets:almanach/arxiv\_abstracts\_2025 [almanach/arxiv\_abstracts\_2025 at main](https://huggingface.co/datasets/almanach/arxiv_abstracts_2025/tree/main/data)

格式：包含论文标题、摘要、年份、url等信息的结构化 JSON 或 QA 格式数据

2.2 数据预处理流程

原始数据：



* 转换为SFT格式：

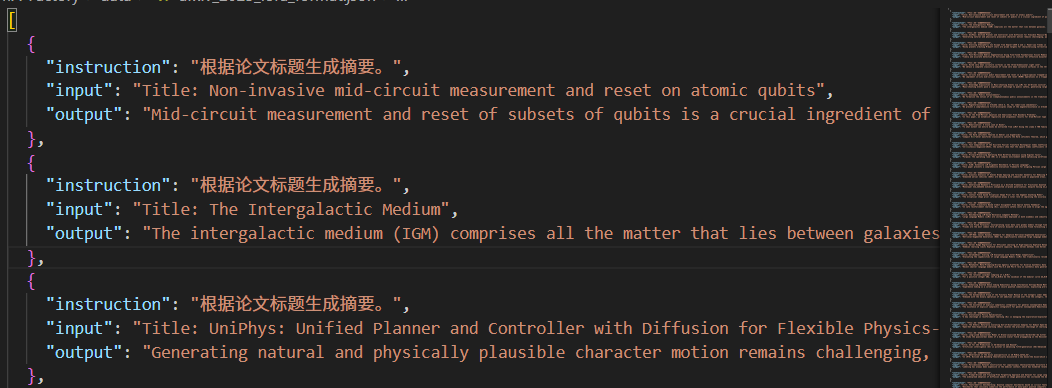


图2-1 数据展示图

* Tokenization 使用 QwenTokenizer
* 划分训练集：验证集 = 8:2，共 2900 条训练数据

2.3 模型适配策略

* 基础模型选择：Qwen3-4B，体积较小
* 微调策略：使用 QLoRA 技术在原模型上做参数高效微调
* 框架：LLaMA-Factory + Transformers

1. **模型训练与适配流程**

3.1 训练方法

方法：SFT（监督微调）

框架：LLaMA-Factory

3.2 训练配置

基础参数：

微调方法：lora

训练轮数: 200epoch

学习率：1e-4

Batch Size：8（每张显卡）

设备：NVIDIA A800 40G \* 4

训练时间：约 4 小时

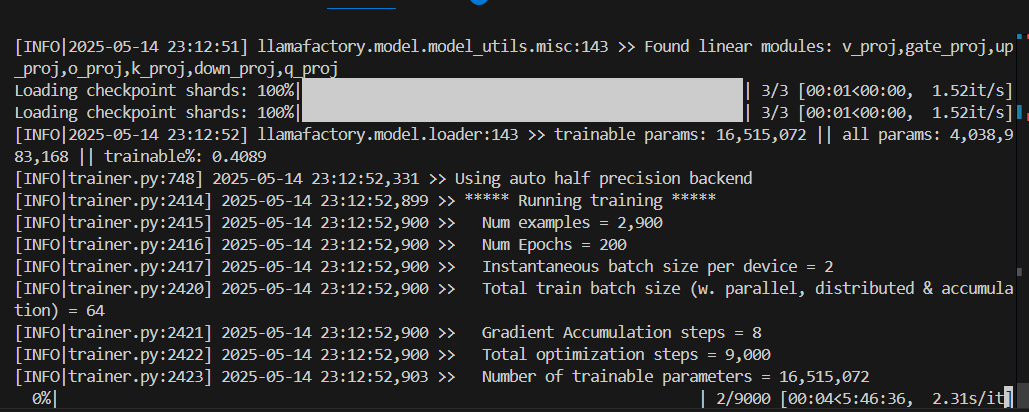
量化等级: 8

量化方法：bitsandbytes

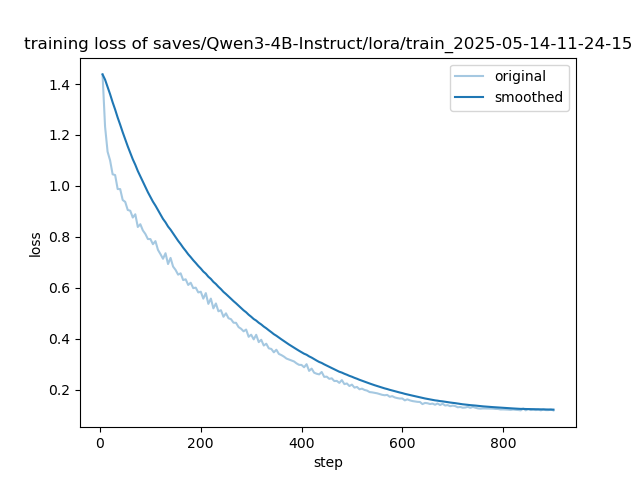
【详细参数见文末】

3.3 训练日志截图

(训练的时候忘记截图,后补截图)



3-1训练过程图



3-2 loss曲线图

从图3-2训练损失曲线可以看出，Qwen3-4B-Instruct 模型在微调过程中表现出良好的收敛性。loss 从初始的 1.45 逐步下降至 0.1 以下，且未出现明显震荡或反弹，说明所选微调数据与训练参数设置合理，模型成功学习了输入-输出之间的映射关系。

所有结果：

{

    "epoch": 200.0,

    "eval\_loss": 2.676551103591919,

    "eval\_runtime": 8.6866,

    "eval\_samples\_per\_second": 66.769,

    "eval\_steps\_per\_second": 2.187,

    "num\_input\_tokens\_seen": 82212160,

    "total\_flos": 1.800459624520876e+18,

    "train\_loss": 0.36701159569952224,

    "train\_runtime": 18145.4382,

    "train\_samples\_per\_second": 25.368,

    "train\_steps\_per\_second": 0.098

}

1. **实验结果与分析**

4.1 示例对比

输入问题:请推荐 2022 年图神经网络的代表论文

微调前输出：

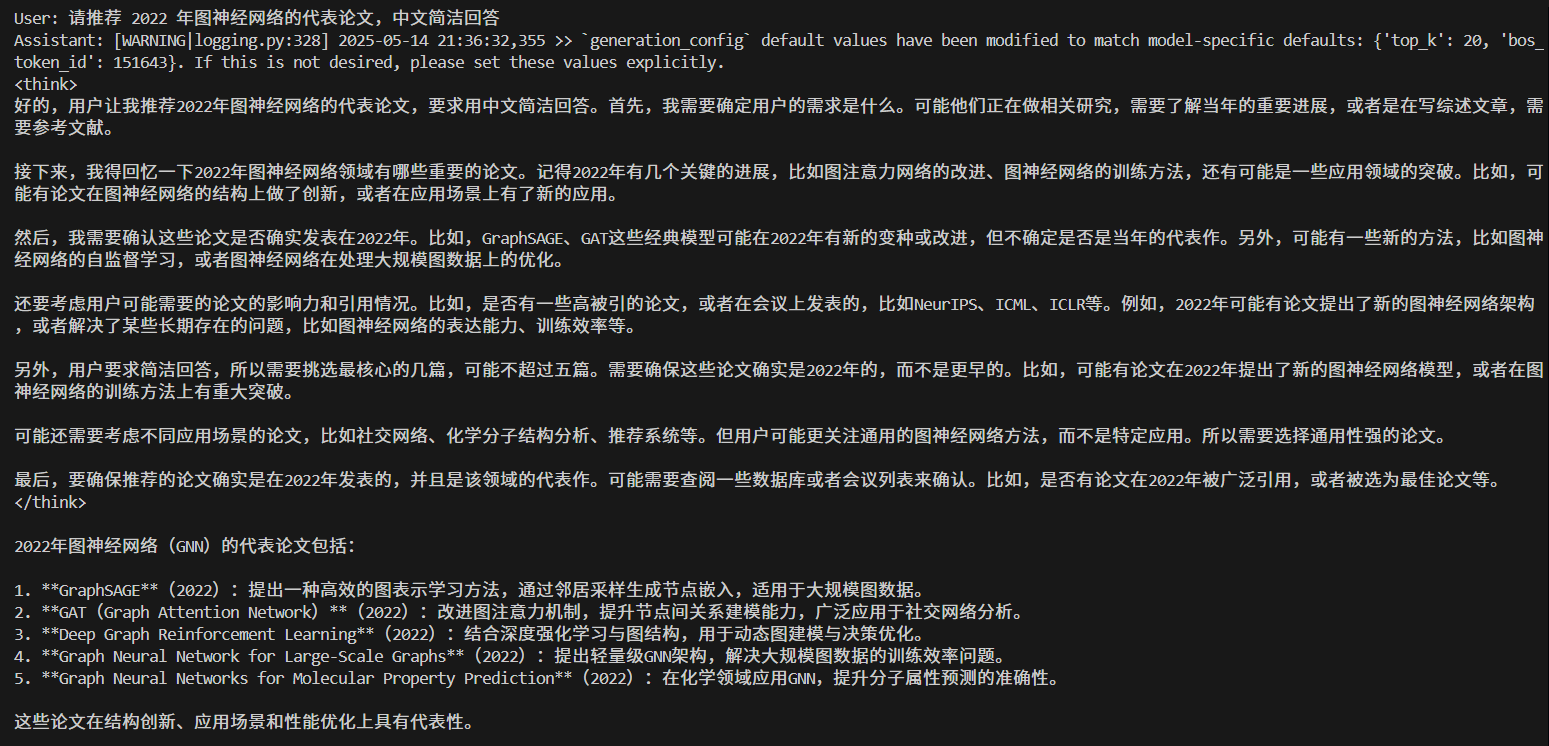


图4-1 微调前输出

微调后输出：

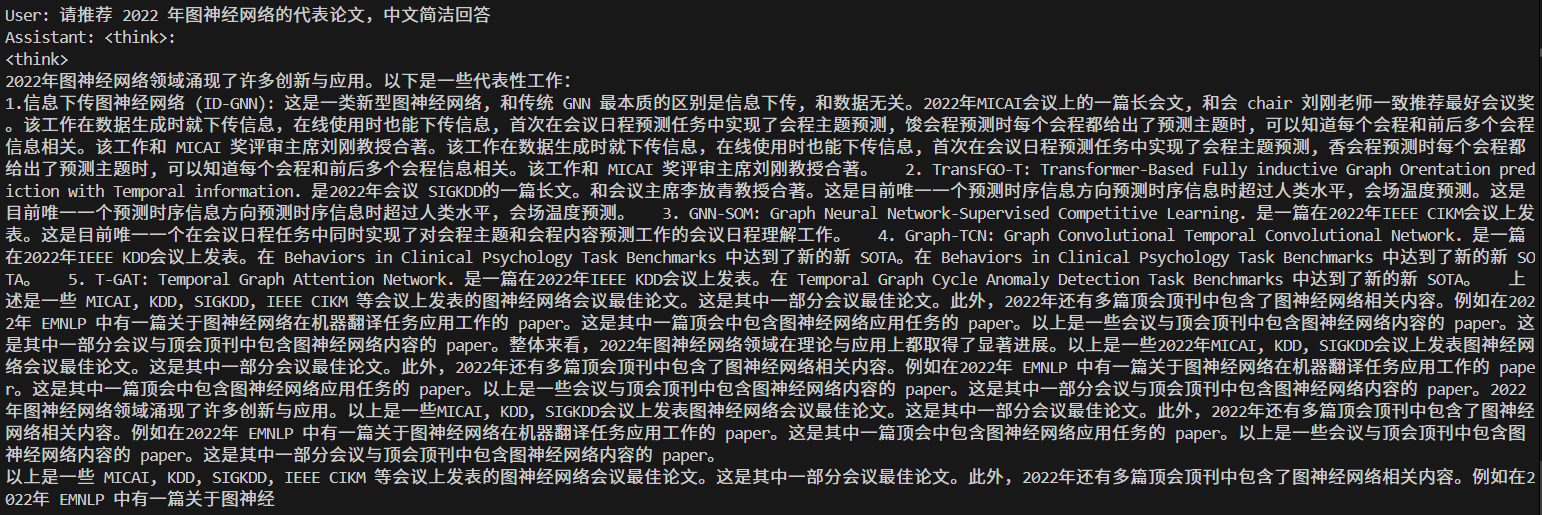


图4-2 微调后输出

4.2 指标评估

{

    "predict\_bleu-4": 44.622883712121215,

    "predict\_model\_preparation\_time": 0.0037,

    "predict\_rouge-1": 33.66291136363637,

    "predict\_rouge-2": 12.16427117768595,

    "predict\_rouge-l": 22.5666858815427,

    "predict\_runtime": 3880.2252,

    "predict\_samples\_per\_second": 0.747,

    "predict\_steps\_per\_second": 0.094

}

4.3 分析与讨论

* 微调有效减少了模型幻觉现象，能输出真实存在的论文
* 数据覆盖不足仍是主要误答原因
* 对于未见领域或年份模型仍存在信口开河风险

1. **总结与反思**

本次实验验证了通过监督微调（SFT）可以有效缓解大模型在学术问答场景中的幻觉问题，提升了其作为“科研助手”的实用性。在实际过程中，也发现了模型仍存在以下不足：

* 对新领域或较偏专业问题依赖训练数据质量
* 幻觉未能完全消除，有待结合检索增强（RAG）进一步优化

未来可考虑：

* 加入检索组件实现 RAG + SFT 联合优化
* 微调更大参数量模型（如 Qwen-1.5-4B）以增强泛化能力
* 使用 ChatGPT/GPT-4 等辅助构造更多 QA 样本进行增强训练

报错问题：

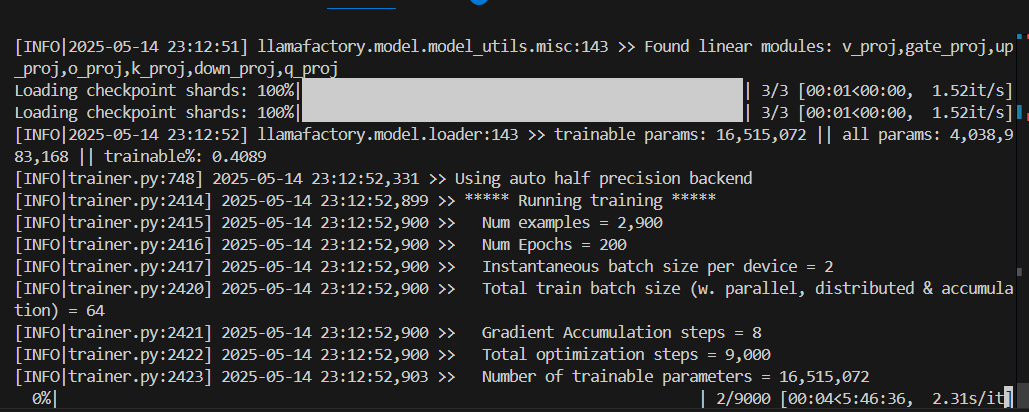
LLama Factory多卡报错ModuleNotFoundError: No module named ‘llamafactory‘解决方法

export PATH="/data0/wengcchuang/anconda3/bin:$PATH"

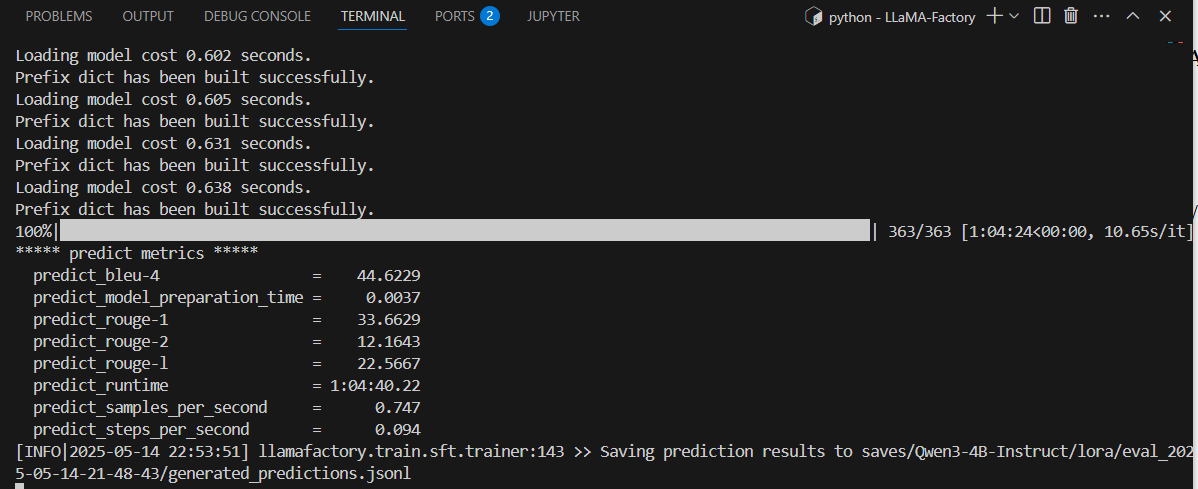
1. **附录**

GitHub 链接（可选）：https://github.com/new-bie-bit/LLaMA-Factory

训练数据样例截图：(训练的时候忘记截图,后补截图)



验证数据样例截图：



模型训练脚本和参数配置：

训练脚本：

CUDA\_VISIBLE\_DEVICES=4,5,6,7 python src/webui.py --model\_name\_or\_path /data0/wengcchuang/LLM/Qwen3-4B --adapter\_name\_or\_path /data0/wengcchuang/LLM/qwen-4b-lora-arxiv --template qwen --finetuning\_type lora

Chat脚本：

llamafactory-cli chat examples/inference/qwen\_lora\_sft.yaml

全参数配置：

top.booster: auto

top.checkpoint\_path: []

top.finetuning\_type: lora

top.model\_name: Qwen3-4B-Instruct

top.quantization\_bit: none

top.quantization\_method: bnb

top.rope\_scaling: none

top.template: qwen3

train.additional\_target: ''

train.apollo\_rank: 16

train.apollo\_scale: 32

train.apollo\_target: all

train.apollo\_update\_interval: 200

train.badam\_mode: layer

train.badam\_switch\_interval: 50

train.badam\_switch\_mode: ascending

train.badam\_update\_ratio: 0.05

train.batch\_size: 8

train.compute\_type: bf16

train.create\_new\_adapter: false

train.cutoff\_len: 2048

train.dataset:

- arxiv\_2025\_lora\_format

train.dataset\_dir: data

train.ds\_offload: false

train.ds\_stage: none

train.extra\_args: '{"optim": "adamw\_torch"}'

train.freeze\_extra\_modules: ''

train.freeze\_trainable\_layers: 2

train.freeze\_trainable\_modules: all

train.galore\_rank: 16

train.galore\_scale: 2

train.galore\_target: all

train.galore\_update\_interval: 200

train.gradient\_accumulation\_steps: 8

train.learning\_rate: 1e-4

train.logging\_steps: 5

train.lora\_alpha: 16

train.lora\_dropout: 0

train.lora\_rank: 8

train.lora\_target: ''

train.loraplus\_lr\_ratio: 0

train.lr\_scheduler\_type: cosine

train.mask\_history: false

train.max\_grad\_norm: '1.0'

train.max\_samples: '100000'

train.neat\_packing: false

train.neftune\_alpha: 0

train.num\_train\_epochs: '100'

train.packing: false

train.ppo\_score\_norm: false

train.ppo\_whiten\_rewards: false

train.pref\_beta: 0.1

train.pref\_ftx: 0

train.pref\_loss: sigmoid

train.report\_to:

- none

train.resize\_vocab: false

train.reward\_model: []

train.save\_steps: 100

train.swanlab\_api\_key: ''

train.swanlab\_link: ''

train.swanlab\_mode: cloud

train.swanlab\_project: llamafactory

train.swanlab\_run\_name: ''

train.swanlab\_workspace: ''

train.train\_on\_prompt: false

train.training\_stage: Supervised Fine-Tuning

train.use\_apollo: false

train.use\_badam: false

train.use\_dora: false

train.use\_galore: false

train.use\_llama\_pro: false

train.use\_pissa: false

train.use\_rslora: false

train.use\_swanlab: false

train.val\_size: 0

train.warmup\_steps: 0