CA-LoRA: Adapting Existing LoRA for Compressed LLMs to Enable Efficient Multi-Tasking on Personal Devices

Weilin Zhao, Yuxiang Huang, Xu Han, Zhiyuan Liu,
Zhengyan Zhang, Kuai Li, Chen Chen, Tao Yang,
Maosong Sun1

汇报人：陆俊安
2025.5.23

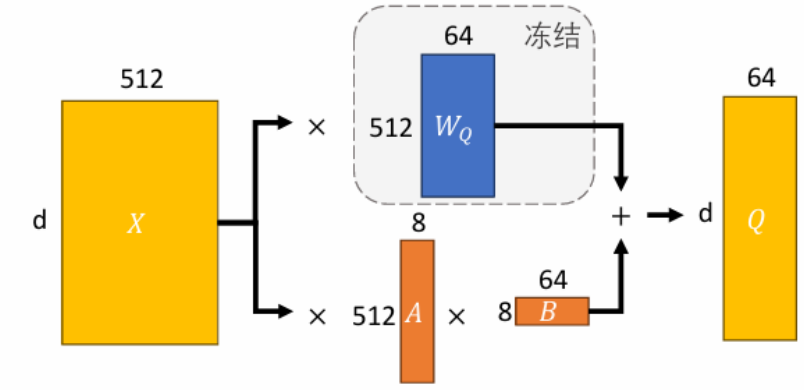
目录

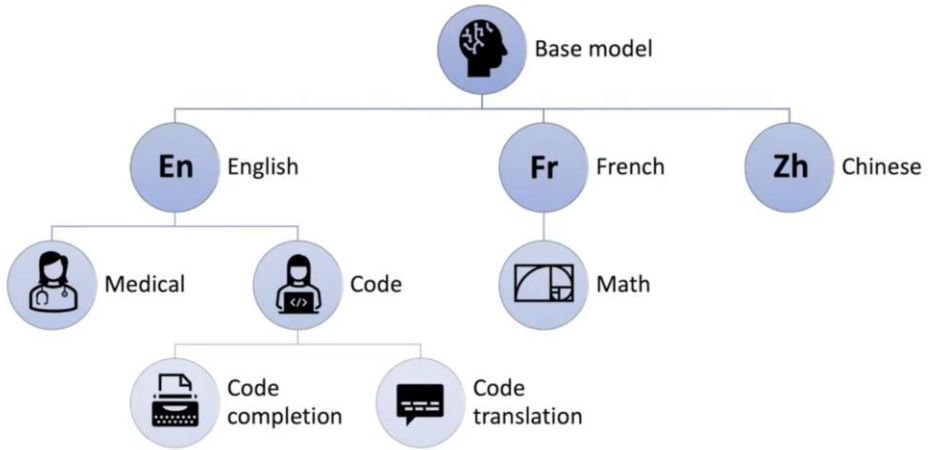
研究背景

- **大语言模型（LLM）：**
  - 综合能力强大，但难以直接适配下游任务，传统全参微调多任务开销大。
  - 性能强大，开销同样巨大，难以在边缘平台部署
- **为解决这两个问题：**
  - 参数高效微调（PEFT）→低秩适应（LoRA）
  - 模型压缩（compressed LLM，以下简写为CLM）
- **以上两种方案产生的新问题：**
  - 压缩后的模型能力显著退化
  - 将原始LLM中的LoRA直接应用到压缩模型中出现性能损失
- **解决新问题的方案：**
  - 本文提出CA-LoRA（Compression-Aware LoRA）
  - 通过知识继承和恢复补偿压缩损失

相关工作

- **参数高效微调（parameter-efficient fine-tuning，PEFT）**
  - 期望仅调整极少参数适配下游任务。
  - 方法多种多样，由于本文基于LoRA，重点关注LoRA
- **低秩适应（Low-Rank Adaptation，LoRA）**
  - 核心假设：适应新任务时权重变化有效维度远小于原始维度，且微调（变化）的ΔW有低内在秩
  - 通过对ΔW进行低秩分解，减少需要更新的参数量



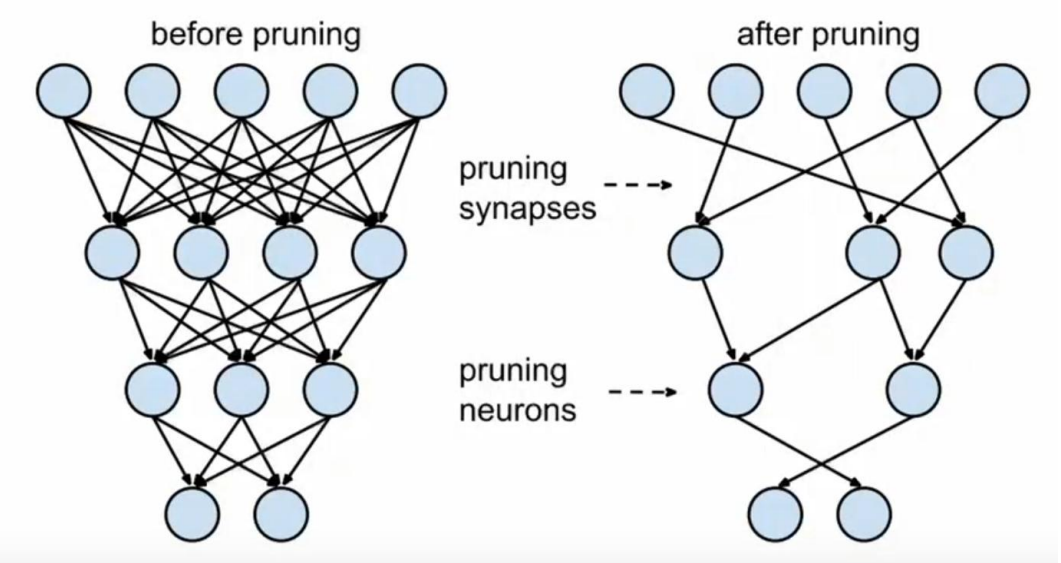


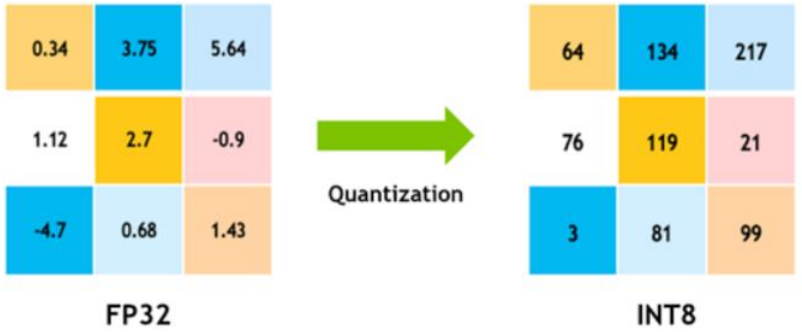
原始文章：LoRA: Low-Rank Adaptation of Large Language Models

链接： [2106.09685] LoRA: Low-Rank Adaptation of Large Language Models

相关工作

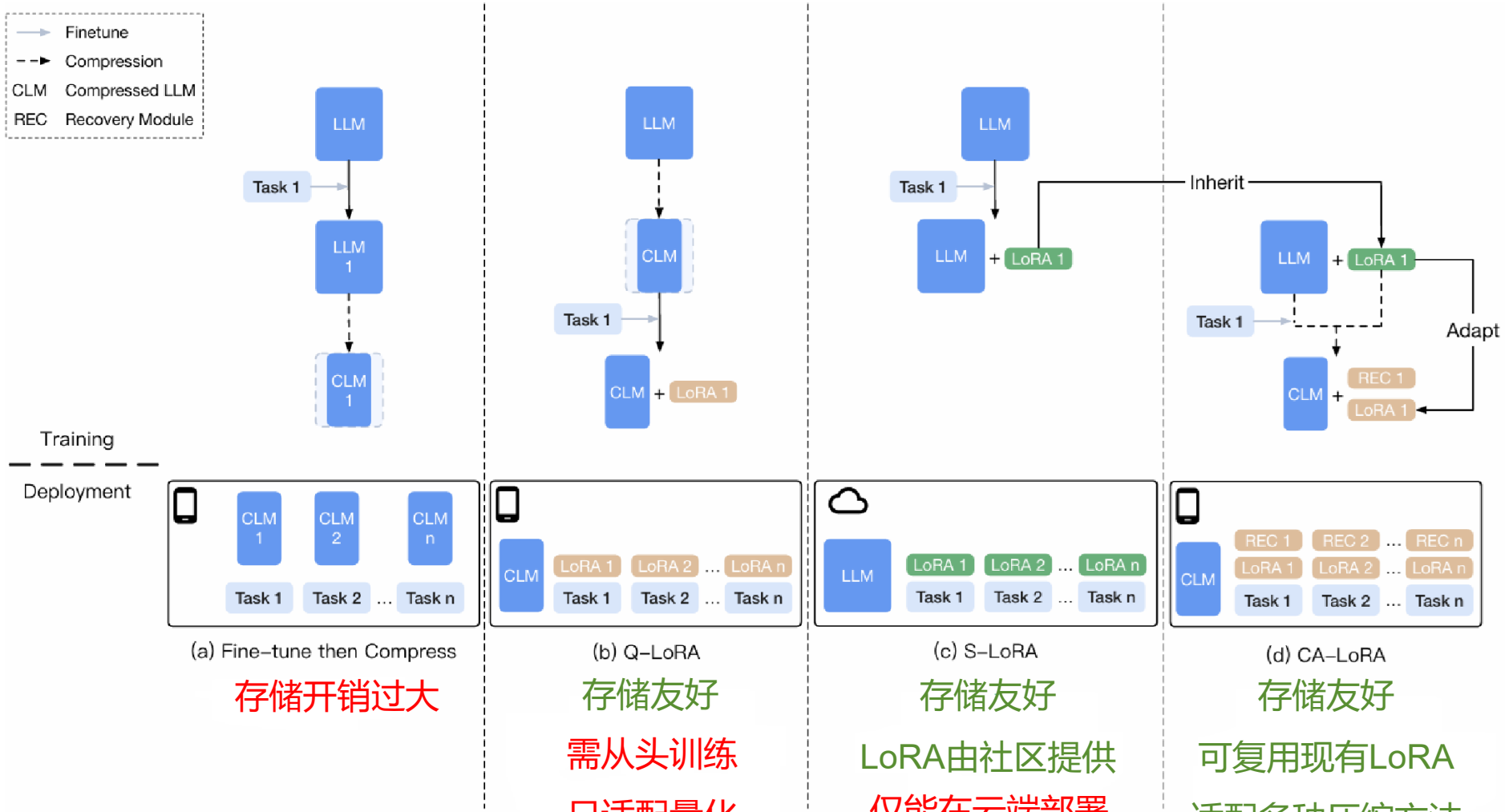
- **模型压缩**
  - 任务相关压缩
    - 存储效率低，扩展性差
  - 任务无关压缩
    - 存储成本低，多任务支持
  - 相关任务无关压缩方法：
    - 量化
    - 剪枝
    - 混合专家化（MoEfication）





相关工作

- **现有方法与CA-LoRA的对比**



存储开销过大

存储友好

需从头训练

只适配量化

存储友好

LoRA由社区提供

仅能在云端部署

存储友好

可复用现有LoRA

适配多种压缩方法

设计实现

数学框架：

假设某LLM模型M可以表示为：$\mathbf{Y} = f(\mathbf{X}; \theta_{\mathcal{M}})$

其中：$f(\cdot)$ 为架构函数；X为输入，Y为输出；$\theta_{\mathcal{M}}$ 为模型参数

若 $(\mathbf{X}^t, \mathbf{Y}^t)$ 为下游任务的数据集，$\mathcal{L}$ 为任务 t 的损失函数，

在LoRA设定中，参数M保持冻结，额外的LoRA模块P在任务特定数据上进行调优。

注入M的LoRA模块参数记作 $\theta_{\mathcal{P}(\mathcal{M})}$

计算过程调整为：$\mathbf{Y} = f_{\text{LoRA}}(\mathbf{X}; \theta_{\mathcal{M}}, \theta_{\mathcal{P}(\mathcal{M})})$

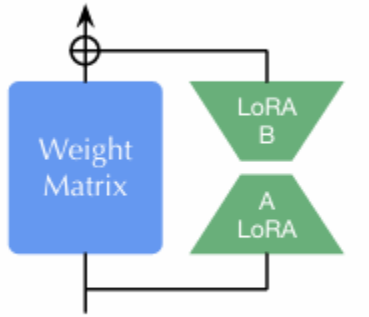
优化目标为：$\theta^t_{\mathcal{P}(\mathcal{M})} = \arg\min_{\theta_{\mathcal{P}(\mathcal{M})}} \mathcal{L}(f_{\text{LoRA}}(\mathbf{X}^t; \theta_{\mathcal{M}}, \theta_{\mathcal{P}(\mathcal{M})}), \mathbf{Y}^t)$

$\theta^t_{\mathcal{P}(\mathcal{M})}$ 表示与大语言模型M协同工作的、针对任务*t*优化的最终LoRA模块参数。

若将该LLM压缩后的模型称为C，则CA-LoRA可形式化定义为：$\theta^t_{\mathcal{P}(\mathcal{C})} = \arg\min_{\theta_{\mathcal{P}(\mathcal{C})}} \mathcal{L}(f_{\text{LoRA}}(\mathbf{X}^t; \theta_{\mathcal{C}}, \theta_{\mathcal{P}(\mathcal{C})}), \mathbf{Y}^t)$

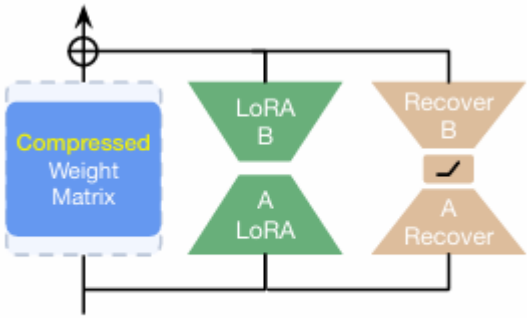
设计实现

- **关键想法：**
  - 启发：模型压缩能保留那些从头训练小模型无法掌握的能力
  - 关键假设：
    - 在未压缩LLM上训练的LoRA模块包含某些任务知识且该知识是仅在CLM上训练的LoRA模块难以掌握的。
    - 通过从原始LLM训练的LoRA模块继承知识的方法。可以恢复压缩过程导致的知识损失



(a) Existing LoRA on LLM

- **关键办法：**
  - 两大机制：
    - LoRA知识继承机制：原始LLM上预训练的LoRA模块作为初始化参数，迁移至压缩版本的LoRA训练中。
    - 模型知识恢复机制：为修复压缩过程导致的知识损失，在CLM中注入低秩非线性恢复模块以弥合知识鸿沟。

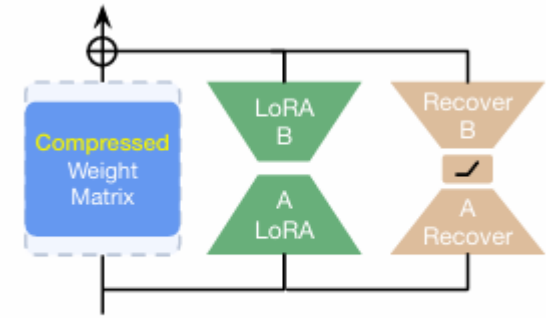


(b) Adapting LoRA to CLM

设计实现

- **知识继承机制**
  - 首先将原LoRA的参数作为初始化参数载入
  - 之后在任务数据上继续训练调优得到最终参数
- **模型知识恢复机制**
  - 一个类似LoRA的低秩旁路结构
  - 数学定义：

$$\mathcal{R}(\mathbf{X}) = \sigma(\mathbf{XD})\mathbf{U},$$

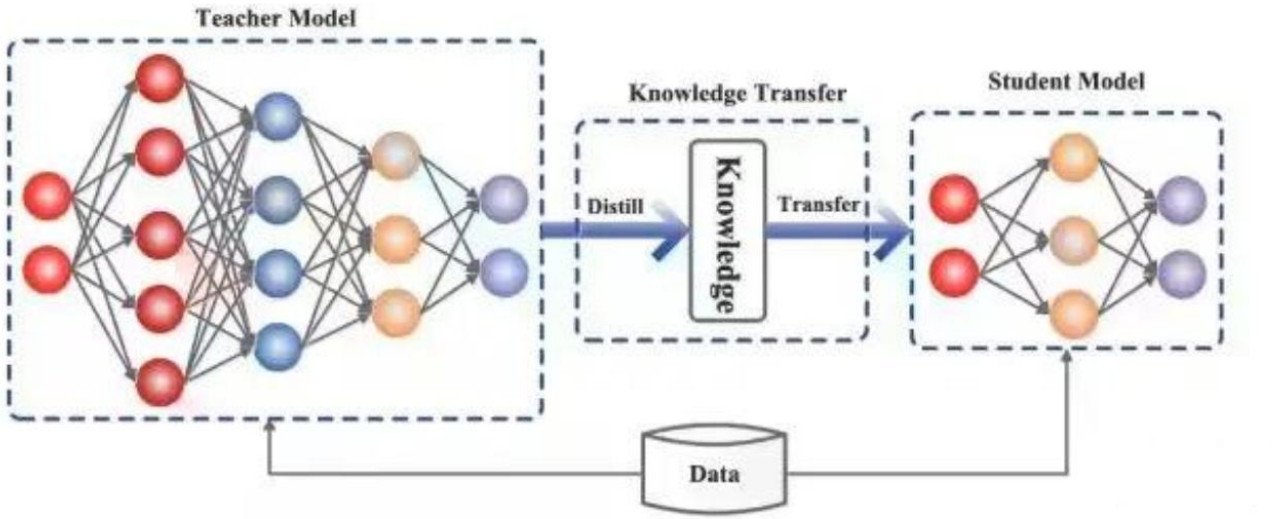


(b) Adapting LoRA to CLM

  - 其中D为其中D为下投影矩阵，σ(·)为激活函数，U为上投影矩阵，二者共同构成恢复模块参数$\theta_{\mathcal{R}}$
  - 通过蒸馏获得最优恢复模块参数

设计实现

- **蒸馏（Model Distillation）**
  - 是模型压缩技术的一种
  - 蒸馏是一种让小模型（学生模型）模仿大模型（教师模型）从而获得大模型的知识或能力的技术
  - 通过蒸馏，学生模型能在参数更少的情况下逼近甚至超越教师模型的性能



原始文章：Distilling the Knowledge in a Neural Network

链接[1503.02531] Distilling the Knowledge in a Neural Network

设计实现

- **模型知识恢复机制的蒸馏函数**
  - 目的：修复压缩过程中损失的知识/能力
  - 选择：均方误差（MSE）损失函数，如下：

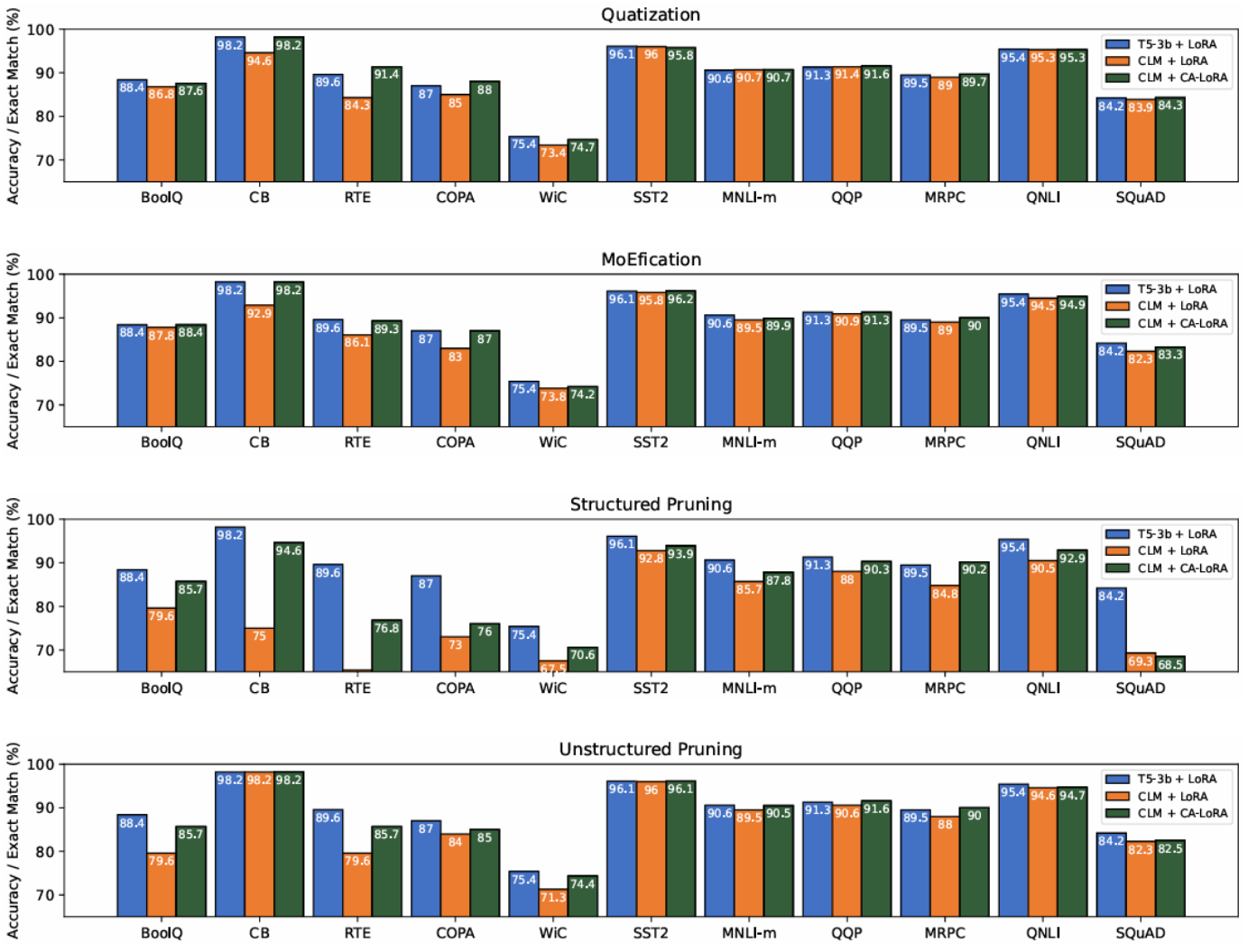
$$\mathcal{L}_{\text{DIST}}(\mathbf{X}^t;\theta_{\mathcal{M}},\theta_{\mathcal{C}},\theta_{\mathcal{P}(\mathcal{C})},\theta_{\mathcal{R}}) = \frac{1}{|\mathbf{X}^t|}\left\|f_{\text{LoRA}}(\mathbf{X}^t;\theta_{\mathcal{M}},\theta^t_{\mathcal{P}(\mathcal{M})}) - f_{\text{LoRA}}(\mathbf{X}^t;\theta_{\mathcal{C}},\theta_{\mathcal{P}(\mathcal{C})},\theta_{\mathcal{R}})\right\|_2^2,$$

  - 其中$\theta_{\mathcal{R}}$是恢复模块参数，实际训练时，将LoRA和恢复模块一同训练，故最终表示如下：

$$\theta^t_{\mathcal{P}(\mathcal{C})},\theta^t_{\mathcal{R}} = \arg\min_{\theta_{\mathcal{P}(\mathcal{C})},\theta_{\mathcal{R}}}\left[\mathcal{L}(f_{\text{LoRA}}(\mathbf{X}^t;\theta_{\mathcal{C}},\theta_{\mathcal{P}(\mathcal{C})},\theta_{\mathcal{R}}),\mathbf{Y}^t) + \alpha\mathcal{L}_{\text{DIST}}(\mathbf{X}^t;\theta_{\mathcal{M}},\theta_{\mathcal{C}},\theta_{\mathcal{P}(\mathcal{C})},\theta_{\mathcal{R}})\right].$$

实验测试

- **典型自然语言（NLP）性能**
  - 基线（baseline）模型：T5-3b
  - 数据集：11个，如右
  - 压缩方法：
    - 量化（Quatization）
    - 混合专家化（MoEfication）
    - 结构化剪枝（Structured Pruning）
    - 非结构化剪枝（Unstructured pruning）
  - 三种方案
    - T5-3b+LoRA: 在原始T5-3b上附加LoRA，仅微调LoRA参数
    - CLM+LoRA：在压缩版T5-3b（CLM）上附加LoRA模块，仅微调LoRA
    - CLM+CA-LoRA：在压缩版T5-3b上附加完整CA-LoRA模块，仅微调LoRA和恢复模块。



实验测试

- **高压缩率下的性能表现**
  - 不同压缩方法和压缩率如右
  - 选用压缩率最大的Q+UP+M方案后，表现如下

Model	Model Size	Ideal Speedup
T5-3b (bf16)	5.61 GB	100%
T5-3b (M)	3.74 GB	150%
T5-3b (UP)	2.81 GB	200%
T5-3b (SP)	2.81 GB	200%
T5-3b (Q)	2.81 GB	200%
T5-3b (Q+UP+M)	0.94 GB	600%
T5-base (bf16)	0.44 GB	1400%

"M"（MoEfication）

"UP"（非结构化剪枝）

"SP"（结构化剪枝）

"Q"（8位量化）

Method	Model Size(GB)	BoolQ Acc(%)	CB Acc(%)	RTE Acc(%)	COPA Acc(%)	WiC Acc(%)	SST2 Acc(%)
T5-3b + LoRA	5.61	88.3	100.0	88.6	88.0	74.0	96.1
T5-base + LoRA	0.44	79.5	91.1	80.7	71.0	69.9	93.5
CLM + LoRA	0.94	85.2	89.3	82.9	80.0	70.5	94.8
CLM + CA-LoRA	0.94	**87.1**	**100.0**	**84.3**	**86.0**	**72.0**	**96.2**

Method	MNLI-m Acc(%)	QQP Acc(%)	QQP F1(%)	MRPC Acc(%)	QNLI Acc(%)	SQuAD EM(%)	SQuAD F1(%)
T5-3b + LoRA	90.6	91.3	90.7	89.5	95.4	84.2	92.5
T5-base + LoRA	84.8	90.6	89.9	86.5	93.1	79.0	87.8
CLM + LoRA	89.0	90.6	89.9	**89.7**	94.7	79.9	**90.6**
CLM + CA-LoRA	**89.9**	**91.5**	**90.9**	89.5	**94.7**	**81.3**	90.5

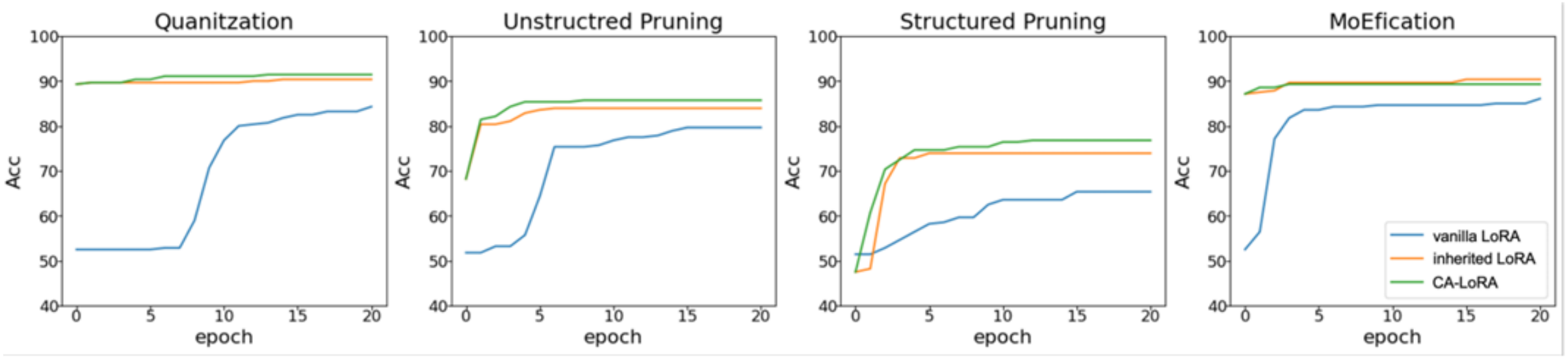
实验测试

- **消融实验**
  - 实验对象：T5-3b(Q+UP+M)
  - 任务：文本蕴含识别（RTE）
  - 移除继承（Inherit）：表示移除知识继承机制并从头训练LoRA
  - 移除恢复模块（Recover）：表示移除恢复模块
  - 移除蒸馏（Distill）：表示训练损失仅含任务损失

Inherit	Recover	Distill	RTE Acc
			83.6
✓			85.4
	✓		82.5
		✓	82.5
✓	✓		87.5
✓		✓	85.7
	✓	✓	86.1
✓	✓	✓	**88.6**

实验测试

- **收敛测试**
  - 测试对象：四种不同压缩的T5-3b
  - 数据集：BoolQ



后续思考

- **文章有什么问题?**
  - 清晰度的问题
    - 文章原文对恢复模块的称呼为“Model Knowledge Recovery”即知识恢复模块，但其本质上是一些额外的参数，没有明确包含来自原始LLM的知识，但是却直接叫知识恢复模块。
    - 另一方面，这也体现出理论解释相对薄弱
  - 固定秩的问题
    - 在所有测试中，LoRA的秩都是固定的。
  - 实验广度不足
    - 缺少跨模态的任务验证

后续思考

- **这篇 paper 的 工作能否进一步深入?**
  - 是否可以尝试解释其恢复模块的工作原理?
  - 是否可以尝试让LoRA的秩进行适应甚至自适应？比如对不同任务或者硬件自动使用不同的秩。
  - 除了LLM外，LoRA在图片生成等领域也有广泛应用，这些地方是否也能使用CA-LoRA?
- **这篇 paper 能不能泛化?**
  - 如果在大模型上训练的LoRA能带来从头训练的小模型不具备的能力，那其他参数微调方法是否也可以做到?

Thanks

汇报人：陆俊安

2025.5.23