LLM-Pruner: On the Structural Pruning Of Large Language Models

NeurIPS 2023

Xinyin Ma，Gongfan Fang，Xinchao Wang*
National University of Singapore

Presenter：Yuankun Feng
2025.04.25

目录

目录

研究背景

LLM发展现状与挑战

发展现状：

- 能力突飞猛进，通用任务解决成为核心

面临挑战：

- 模型规模指数级增长，部署成本飙升
- 压缩率与性能不能兼顾
- 任务无关压缩的结构性挑战
- 模型碎片化与硬件适配鸿沟

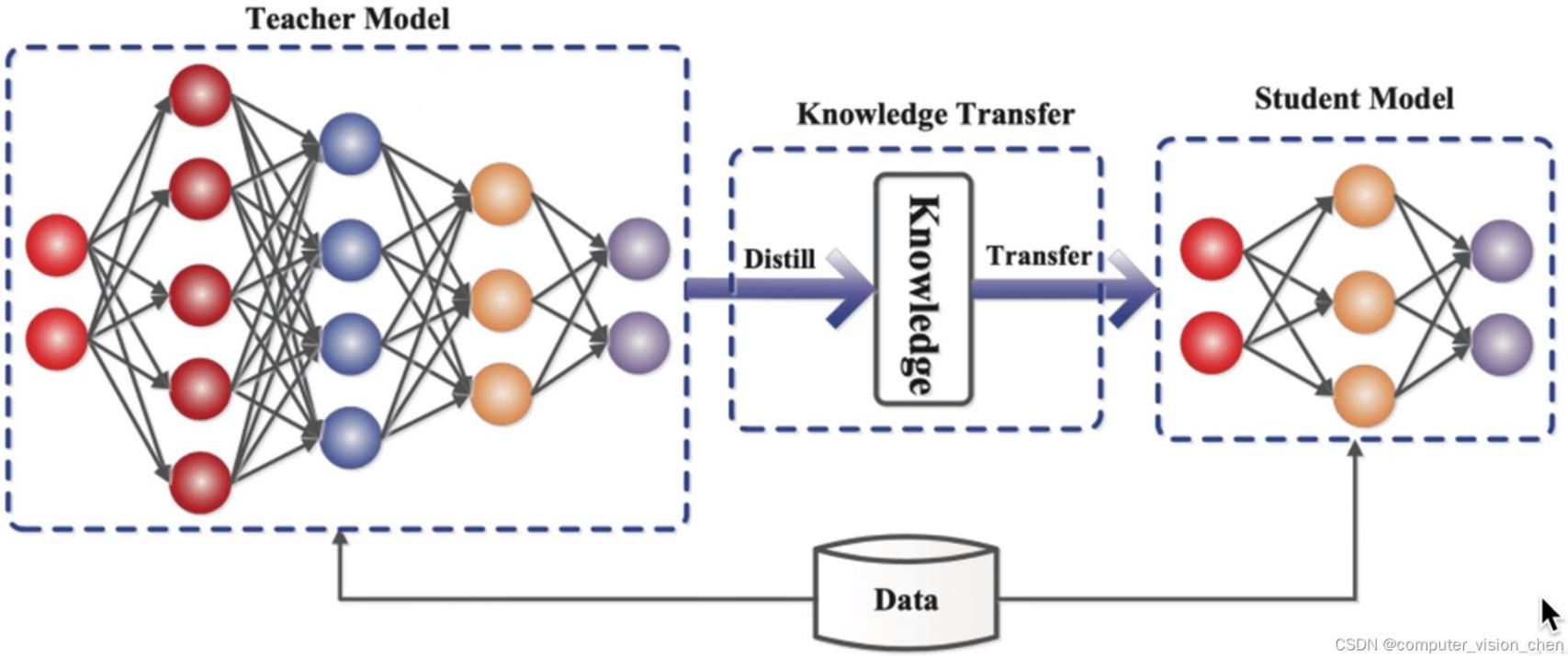
现有压缩技术分类

- 蒸馏
- 量化
- 剪枝

研究背景

蒸馏

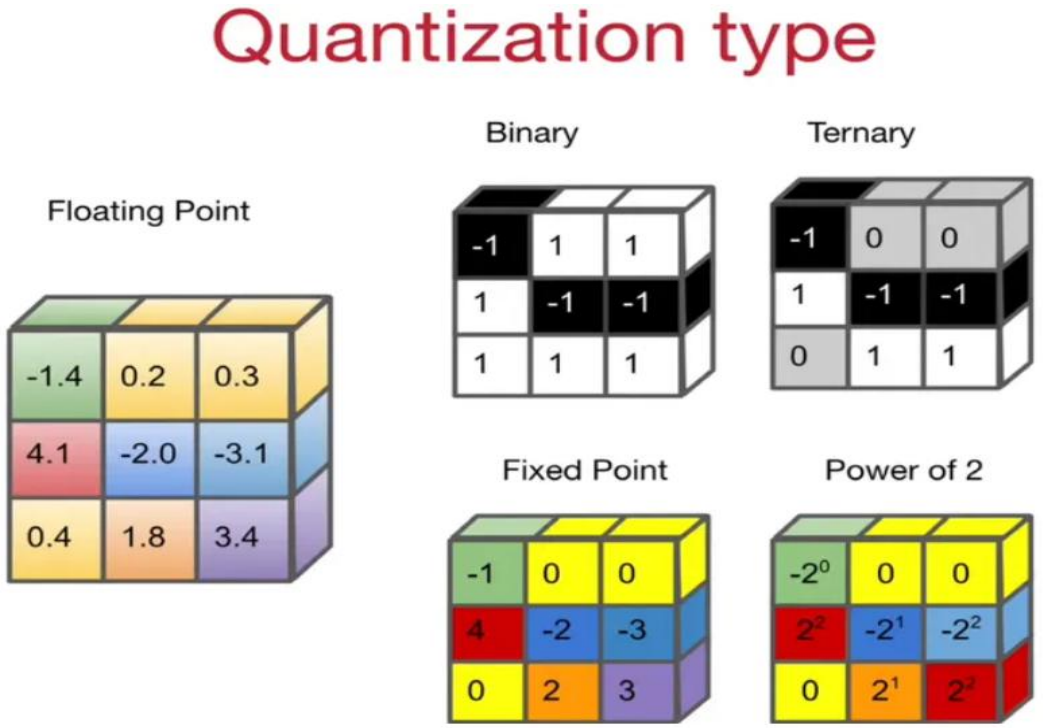
- 原理：用大模型（教师模型）指导小模型（学生模型）训练，把大模型知识迁移到小模型，使小模型规模小但性能高。
- 优点：几乎不损失性能，但模型体积和计算量大幅减小，推理速度变快，存储和部署成本降低。
- 适用场景：移动设备、嵌入式系统等资源受限场景。



研究背景

量化

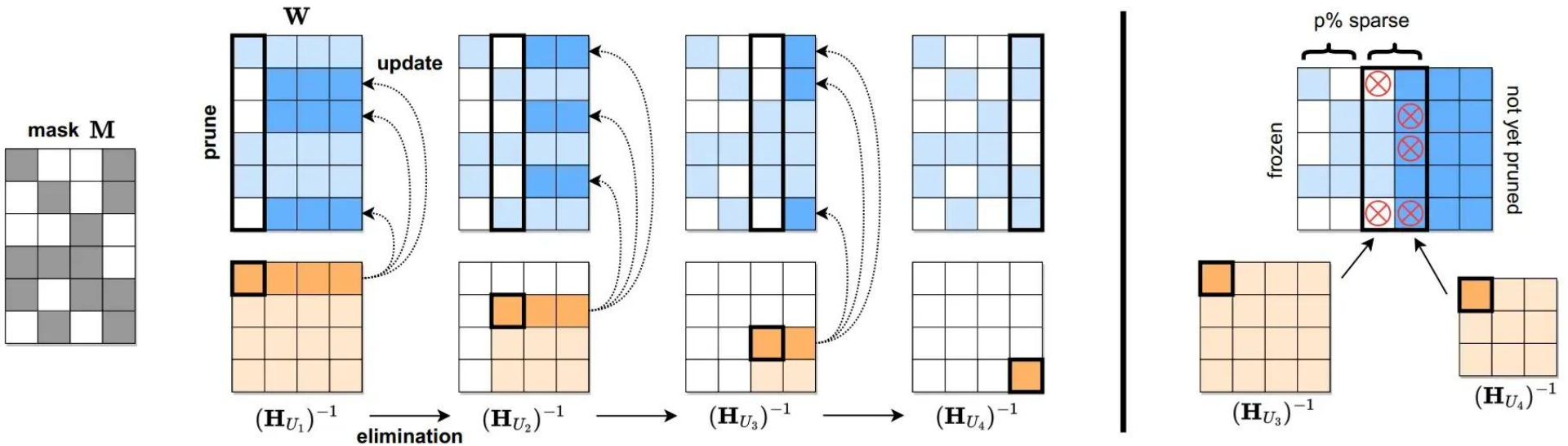
- 原理：将模型参数和计算从高精度数据类型转换为低精度数据类型，降低存储和计算需求，同时保持性能。
- 优点：模型体积减小，计算效率提升，适合在资源受限的环境中使用。
- 适用场景：适用于移动设备、物联网设备等场景。



研究背景

剪枝

- 原理：去除神经网络中冗余的参数或连接，使模型结构更紧凑，保留核心部分以维持性能。
- 优点：模型体积减小、计算量降低，推理速度加快，存储和计算成本减少。
- 适用场景：适用于移动设备、嵌入式系统等资源受限场景。



研究背景

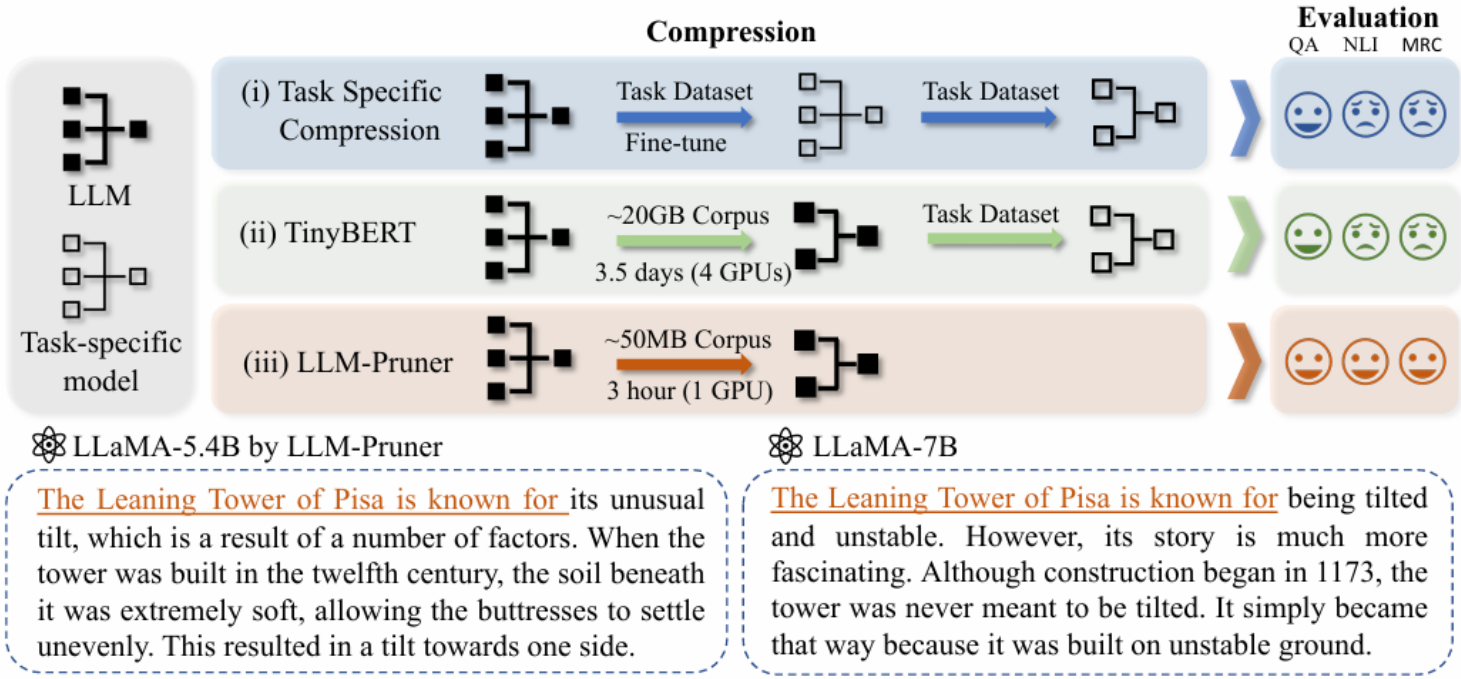
任务专用压缩

- 定义：针对单一任务（如文本分类）进行模型压缩，通过裁剪与任务无关的结构实现轻量化。
- 代表方法：

知识蒸馏（如TinyBERT、DistilBERT）：用大模型指导小模型学习特定任务知识

任务特定剪枝：仅保留与目标任务相关的神经元（如BERT仅保留最后几层分类层）。

- 通用性丧失
- 数据依赖重
- 效率问题



目录

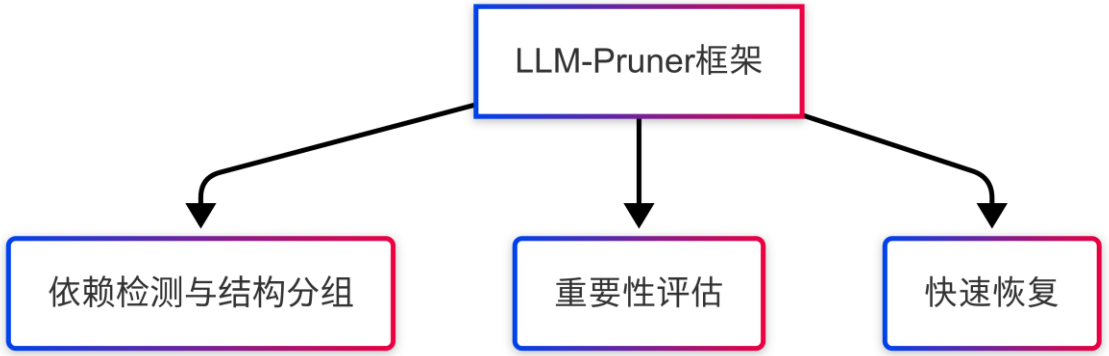
LLM-Pruner方法

核心思想

- 任务无关压缩：保留LLM的多任务能力（分类、生成、推理），无需针对特定任务微调。
- 结构化剪枝：通过依赖关系分析移除冗余结构（如注意力头、FFN层），而非随机剪枝。
- 快速恢复：仅需3小时LoRA微调（单GPU），显著低于传统蒸馏（如TinyBERT需14天）。

关键步骤

- 依赖检测与结构分组
- 重要性评估
- 快速恢复



LLM-Pruner方法

依赖检测

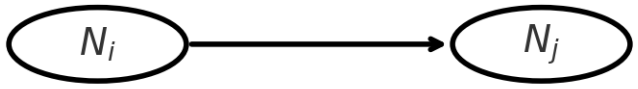
- **原理：通过神经元输入输出关系构建依赖图，识别必须同步剪枝的耦合结构。**
- **规则：**

前向依赖： $N_j \in Out(N_i) \wedge Deg^-(N_j) = 1 \Rightarrow N_j$依赖$N_i$

反向依赖： $N_i \in In(N_j) \wedge Deg^+(N_i) = 1 \Rightarrow N_i$依赖$N_j$

同步剪枝

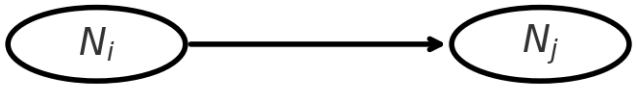
Forward Dependency



$N_j \in \text{Out}(N_i)$ and $\deg^-(N_j) = 1$

$\Rightarrow$ N_j depends on N_i

Backward Dependency



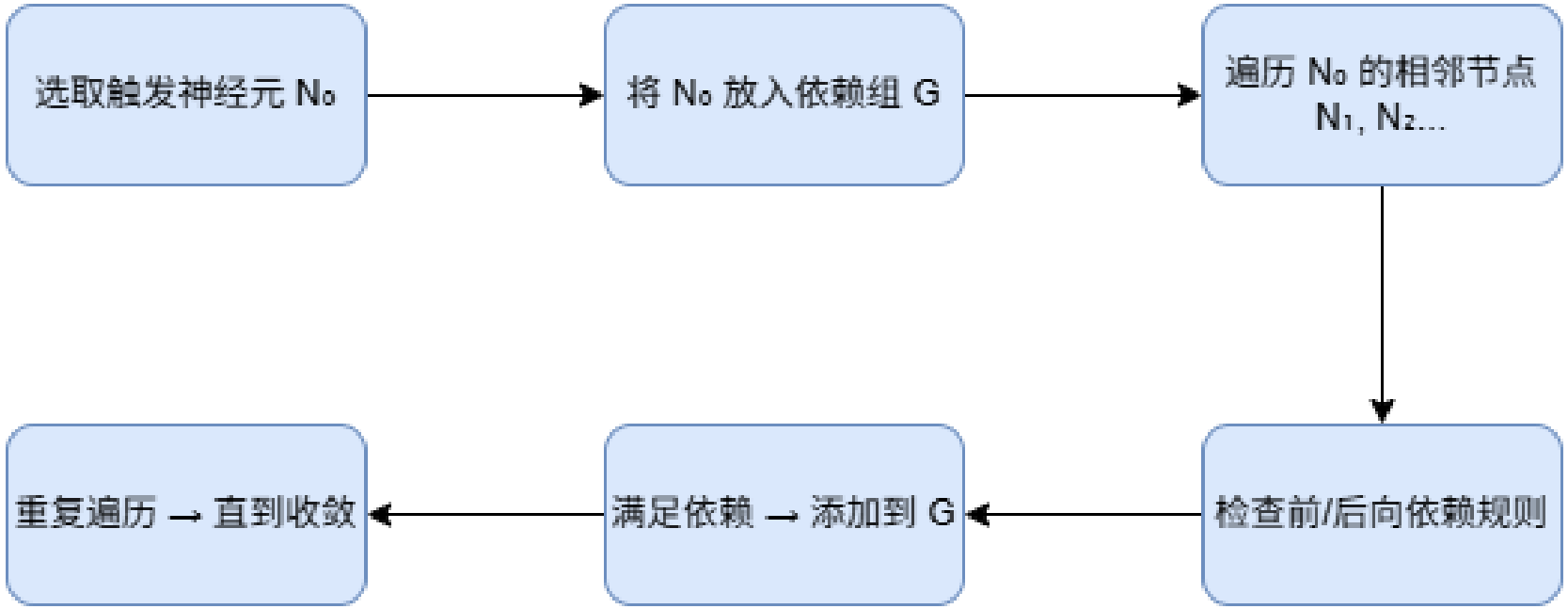
$N_i \in \text{In}(N_j)$ and $\deg^+(N_i) = 1$

$\Rightarrow$ N_i depends on N_j

LLM-Pruner方法

触发-遍历算法流程

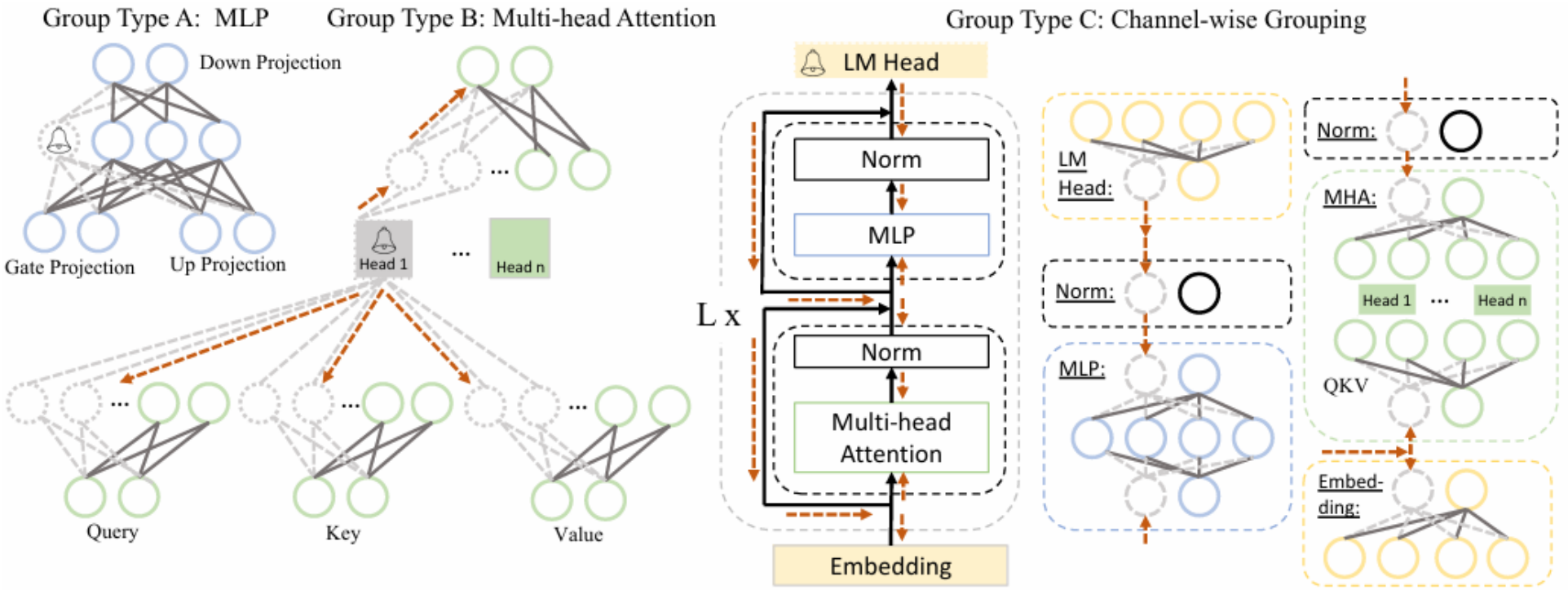
- 基于依赖检测规则，从单个神经元出发，递归遍历所有依赖的神经元，自动构建不可分割的依赖组，为结构化剪枝提供剪枝单元。
- 目标：无需人工干预，适配任意LLM架构（如LLaMA/Vicuna/ChatGLM）。



LLM-Pruner方法

结构分组

- 核心思想：在构建依赖图后，LLM-Pruner将结构自动划分为三个耦合组，作为剪枝基本单元。
- 结构耦合组是剪枝的最小单位，确保功能完整性和梯度一致性，是区别于随机剪枝的核心特征。



LLM-Pruner方法

校准集与梯度 / Hessian 概念

- **定义：**

  用于评估模型中 “依赖组” 重要性的小规模代表性数据集（如论文中采用的 Bookcorpus 子集）。

- **作用：替代原始大规模训练数据，高效计算梯度与损失变化，降低计算成本。**

概念	数学定义	物理意义	计算成本
梯度（一阶导数）	$\frac{\partial \mathcal{L}}{\partial W_{\mathrm{i}}}$	反映参数变化的“方向与快慢”	$O\ (N)$
Hessian（二阶导数）	$\frac{\partial^2 \mathcal{L}}{\partial W_i^2}$	反映参数变化的“敏感性变化”	$O\ (N^2)$

LLM-Pruner方法

重要性评估

- **目的：**

  量化每个“依赖组”(耦合结构)对模型整体性能的影响。

  识别并准备移除那些移除后对模型性能损失最小的组。

- **两种估计方式：**

  向量级重要性 (I_{W_i})： $I_{W_i} = |\Delta L(D)| = |L_{W_i}(D) - L_{W_i=0}(D)|$

  元素级重要性 ($I_{W_i^k}$)： $IW_i^k = |\Delta L(D)| = |\frac{\partial L(D)}{\partial W_i^k} W_i^k - \frac{1}{2} W_i^k H_{kk} W_i^k + O(\|W_i^k\|^3)|$

方法	描述	优缺点
向量级估计	整组结构作为一个整体评估对 Loss 的影响	高效，适合大模型
元素级估计	每个参数单独评估，再进行聚合	精细，计算开销较大

LLM-Pruner方法

重要性聚合策略对比

- 为什么需要聚合?

  我们已经识别了耦合结构组，同一个组内的结构需要一起剪枝。

  为了决定剪枝哪个“组”，我们需要一个单一的分数来代表整个组的重要性。

  聚合策略就是将组内所有结构的个体重要性分数合并（聚合）成一个组的总重要性分数的方法。

- 聚合策略

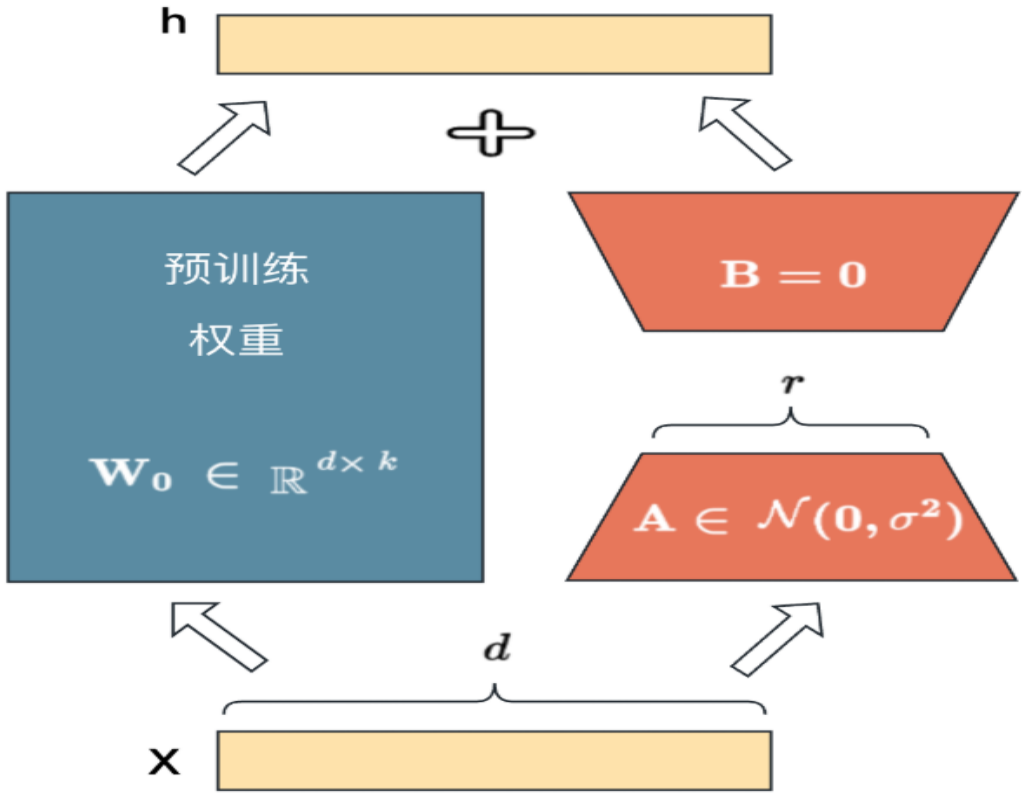
聚合策略	描述	特点	使用场景
求和（Summation）	所有元素重要度求和	稳定且平衡，保持结构一致性	大部分情况
最大（Maximum）	取组内重要度最大值	保守，保留最强的特征	需要保证最强特征的场景
乘积（Product）	所有元素重要度相乘	激进，去除对模型影响小的特征	目标是减少特征数的场景
仅最后节点（Last-only）	仅关注组内最后一层的节点	轻量，忽略中间节点的影响	需要关注最终结果输出的场景

LLM-Pruner方法

LoRA 低秩适配原理

- 为何需要快速恢复?
  - 剪枝操作虽然减小了模型尺寸，但通常会导致性能下降。
  - 需要一个后续的微调步骤来恢复模型的性能。
  - 传统的全模型微调计算量大、耗时且需要大量数据，与我们低资源压缩的目标不符。
- 解决方案：低秩适配 (LoRA)

  核心思想： 在预训练模型的权重矩阵上，不直接修改原始权重 W，而是引入低秩矩阵的更新。



LLM-Pruner方法

微调流程与超参数

- 微调步骤：
  - 剪枝后的模型：剪去多余的参数并使用 LoRA 低秩矩阵进行初始化。
  - 选择微调数据集：使用少量样本数据集（如 50K Alpaca 数据集）。
  - 选择优化器与算法：选择标准优化算法（如 Adam 或 LAMB）。
  - 微调训练：通常设置为 2 个 Epoch，使用 1 GPU 进行微调。
- 超参数设置：

超参数	值
学习率	1e-5 或 2e-5
Epoch 数	2-3
GPU 时间	3 小时

目录

实验与结果

实验设置

- 模型选择：

模型	特点	适配重点
LLaMA-7B	开源标杆模型，无预训练任务特定优化	验证通用剪枝能力
Vicuna-7B	对话优化，含 RLHF 训练	验证对话场景性能保留
ChatGLM-6B	中文预训练，支持多语言	验证跨语言任务适配性

这些模型代表了当下流行的开源 LLM，具有不同的结构特点（例如 LLaMA 和 Vicuna 基于 Transformer 解码器，ChatGLM 是双语模型）。

实验与结果

实验设置

- 评估数据集：
  - 零样本分类数据集 (Zero-shot Classification)：用于衡量模型在未见过特定任务训练数据情况下的推理和理解能力。

    数据集列表： BoolQ, PIQA, HellaSwag, WinoGrande, ARC-easy, ARC-challenge, OpenbookQA。
  - 零样本困惑度数据集 (Zero-shot Perplexity - PPL)：用于衡量模型在语言建模和生成流畅度方面的表现，PPL 值越低越好。

    数据集列表： WikiText2, PTB (Penn Treebank)。
- 评估指标：

  平均准确率 (Average Accuracy)、困惑度 (Perplexity - PPL)、剪枝前后与其他模型对比。

实验与结果

20% 剪枝性能对比

- 实验：LLaMaA-7B 20% 结构化剪枝效果评估
- 结果：经过 20% 结构化剪枝后，LLM-Pruner 通过低资源快速微调，能够将 LLaMA-7B 的零样本分类性能恢复到接近原始模型的水平。

Pruning Ratio	Method	WikiText2↓	PTB↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
Ratio = 0%	LLaMA-7B[51]	-	-	76.5	79.8	76.1	70.1	72.8	47.6	57.2	68.59
	LLaMA-7B*	12.62	22.14	73.18	78.35	72.99	67.01	67.45	41.38	42.40	63.25
Ratio = 20% w/o tune	L2	582.41	1022.17	59.66	58.00	37.04	52.41	33.12	28.58	29.80	42.65
	Random	27.51	43.19	61.83	71.33	56.26	54.46	57.07	32.85	35.00	52.69
	Channel	74.63	153.75	62.75	62.73	41.40	51.07	41.38	27.90	30.40	45.38
	Vector	22.28	41.78	61.44	71.71	57.27	54.22	55.77	33.96	38.40	53.25
	Element^2	19.77	36.66	59.39	75.57	65.34	61.33	59.18	37.12	39.80	56.82
	Element^1	19.09	34.21	57.06	75.68	66.80	59.83	60.94	36.52	**40.00**	56.69
Ratio = 20% w/ tune	Channel	22.02	38.67	59.08	73.39	64.02	60.54	57.95	35.58	38.40	55.57
	Vector	18.84	33.05	65.75	74.70	64.52	59.35	60.65	36.26	39.40	57.23
	Element^2	**17.37**	30.39	**69.54**	76.44	68.11	**65.11**	63.43	**37.88**	**40.00**	**60.07**
	Element^1	17.58	**30.11**	64.62	**77.20**	**68.80**	63.14	**64.31**	36.77	39.80	59.23

Table 1：压缩后的 LLaMA-7B 的零样本性能

实验与结果

20% 剪枝性能对比

- 实验：LLaMaA-13B 20% 结构化剪枝效果评估
- 结果：LLM-Pruner 同样能够成功对 LLaMA-13B 进行 20% 结构化剪枝，并通过低资源快速微调有效恢复模型性能。

Pruning Ratio	Method	WikiText2↓	PTB↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
Ratio = 0%	LLaMA-13B*	11.58	20.24	68.47	78.89	76.24	70.09	74.58	44.54	42.00	64.97
Ratio = 20% w/o tune	L2	61.15	91.43	61.50	67.57	52.90	57.54	50.13	31.14	36.80	51.08
	Random	19.24	31.84	63.33	73.18	63.54	60.85	64.44	36.26	38.00	57.09
	Channel	49.03	106.48	62.39	66.87	49.17	58.96	49.62	31.83	33.20	50.29
	Block	16.01	29.28	67.68	77.15	73.41	65.11	68.35	38.40	42.40	61.79
Ratio = 20% w/ tune	L2	20.97	38.05	**73.24**	76.77	71.86	64.64	67.59	39.93	40.80	62.12
	Random	16.84	31.98	64.19	76.06	68.89	63.30	66.88	38.31	40.80	59.78
	Channel	17.58	29.76	69.20	76.55	68.89	66.38	62.08	38.99	39.60	60.24
	Block	**15.18**	**28.08**	70.31	**77.91**	**75.16**	**67.88**	**71.09**	**42.41**	**43.40**	**64.02**

Table 2：压缩后的 LLaMA-13B 的零样本性能

实验与结果

20% 剪枝性能对比

- 实验：Vicuna-7B 20% 结构化剪枝效果评估
- 结果：LLM-Pruner 同样能够有效压缩 Vicuna-7B 模型 20%，并通过快速微调将性能保持在原始模型的较高水平（92.03%）。

Pruned Model	Method	WikiText2 ↓	PTB ↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
Ratio = 0%	Vicuna-7B	16.11	61.37	76.57	77.75	70.64	67.40	65.11	41.21	40.80	62.78
Ratio = 20% w/o tune	l2	3539.98	5882.21	55.90	56.15	32.37	51.85	30.01	28.41	28.20	40.41
	random	34.63	112.44	61.47	70.89	54.67	56.27	55.60	31.74	34.60	52.18
	Channel	71.75	198.88	51.77	63.93	42.58	55.17	43.94	29.27	33.40	45.72
	Vector	27.03	<u>92.51</u>	62.17	71.44	55.80	53.43	55.77	33.28	37.80	52.81
	Element^2	<u>24.70</u>	94.34	<u>62.87</u>	<u>75.41</u>	<u>64.00</u>	<u>58.41</u>	60.98	<u>37.12</u>	<u>39.00</u>	<u>56.83</u>
	Element^1	25.74	92.88	61.70	75.30	63.75	56.20	<u>63.22</u>	36.60	37.00	56.25
Ratio = 20% w/ tune	Vector	19.94	**74.66**	63.15	74.59	61.95	60.30	60.48	36.60	**39.40**	56.64
	Element^2	**18.97**	76.78	60.40	75.63	**65.45**	**63.22**	**63.05**	**37.71**	39.00	**57.78**
	Element^1	19.69	78.25	**63.33**	**76.17**	65.13	60.22	62.84	37.12	39.20	57.71

Table 4：压缩后的 Vicuna-7B 的零样本性能

实验与结果

20% 剪枝性能对比

- **小结：**

  LLM-Pruner 在多种大型语言模型（LLaMA-7B/13B, Vicuna-7B）上均能有效实现 20% 的结构化剪枝。

  剪枝后，模型性能通过低资源快速微调得到显著恢复，保持接近原始模型的零样本能力。

  实验结果验证了 LLM-Pruner 方法的有效性和在不同 LLMs 上的泛化能力。

通过对比剪枝前后的分类准确率，表明 LLM-Pruner 在进行 20% 剪枝后依然能够保持较高的性能，表明剪枝技术在减少计算量的同时不会显著影响模型的表现。

实验与结果

高剪枝率困惑度表现

- 实验：LLM-Pruner 在高压缩率下的表现
- 结果：尽管 LLM-Pruner 在高剪枝率下能比简单基线更好地保持性能，但模型困惑度仍显著增加，表明极高压缩率仍是挑战。

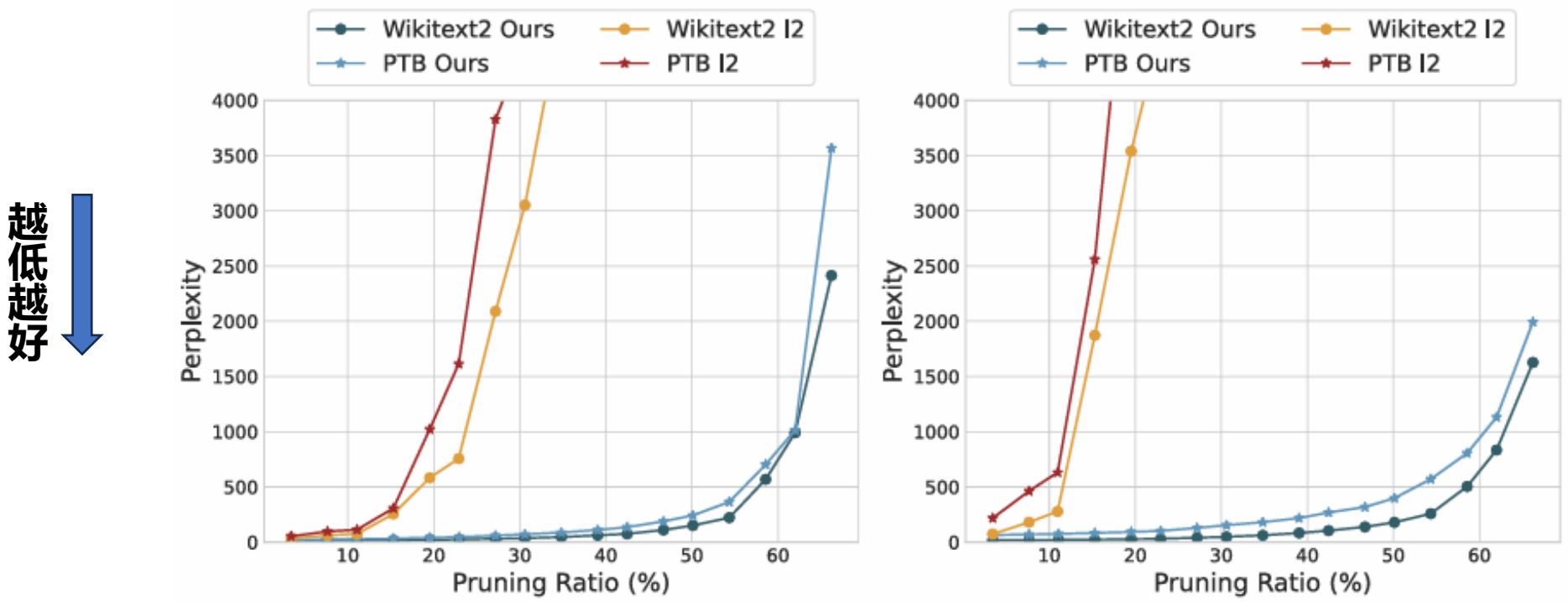


Figure 4：LLaMA-7B（左）和 Vicuna-7B（右）在不同剪枝率下的剪枝结果

实验与结果

高剪枝率困惑度表现

- 小结：
  - 提高剪枝率会导致性能下降。
  - 相比简单基线，LLM-Pruner 在高剪枝率下能更长时间保持性能稳定。
  - 极高压缩率下，性能显著下降，仍是当前挑战。

实验与结果

与现有方法对比

- 实验：与 DistilBERT 对比
- 结果：LLM-Pruner 剪枝的模型在参数量相近的情况下，平均准确率高于传统的蒸馏方法 DistilBERT 。

Pruning Ratio	#Param	Average
DistilBert	3.50B	44.64
LLM-Pruner	3.35B	48.88

Table 8： DistilBert 对比 LLM-Pruner

实验与结果

与现有方法对比

- 实验：与 StableLM 对比
- 结果：LLM-Pruner 能够在低资源下构建性能与同等规模从头训练模型（StableLM）相当甚至更优的轻量级大型语言模型。

Pruning Ratio	#Param	Latency	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
StableLM-3B	3.6B	31.69s	48.78	69.48	44.52	54.62	50.93	25.17	27.40	45.84
LLaMA-3B	3.6B	37.96s	61.41	70.08	51.01	55.01	46.80	30.38	37.40	50.30

Table 9：从头训练（StableLM-3B）对比剪枝（LLaMA-3B by LLM-Pruner）

实验与结果

与现有方法对比

- 小结：
  - LLM-Pruner 剪枝的模型在性能上优于传统的蒸馏方法 (DistilBERT)。
  - LLM-Pruner 能在低资源下构建性能与从头训练模型 (StableLM) 相当甚至更优的轻量级 LLM。
  - 实验表明 LLM-Pruner 是一个高效且有竞争力的 LLM 压缩与构建方法。

实验与结果

消融研究：依赖组 vs. 单独剪枝

- 实验：剪枝策略对比：考虑依赖 vs. 忽略依赖
- 结果：忽略结构依赖关系进行剪枝会导致大型语言模型性能急剧下降，甚至难以恢复。

	Method	WikiText2↓	PTB↓	Average↑
w/o Tuning	w/o dependency	68378.42	79942.47	38.32
	w/ dependency	19.09	34.21	56.69
w/ Tuning	w/o dependency	13307.46	13548.08	38.10
	w/ dependency	17.58	30.11	59.23

Table 6：基于依赖的结构化剪枝的效果

实验与结果

消融研究：依赖组 vs. 单独剪枝

- 小结：
  - 结构依赖关系对模型功能至关重要，剪枝必须考虑依赖。
  - 忽略依赖关系进行剪枝会导致模型性能崩溃，即使微调也难以恢复。
  - 这个结果验证了 LLM-Pruner “发现阶段”识别依赖组的必要性。

实验与结果

消融研究：聚合策略与层敏感性分析

- 实验：聚合策略的影响
- 结果： 聚合策略的选择会影响剪枝后模型在不同任务上的侧重表现。论文选择了“求和”策略作为默认，因为它综合性能较好。

Method	WikiText2↓	PTB↓	ARC-e↑	PIQA↑	OBQA↑
Summation	66.13	164.25	40.70	63.49	34.80
Max	62.59	144.38	39.60	63.71	34.60
Production	77.63	192.88	37.84	62.08	35.00
Last-only	130.00	170.88	41.92	64.75	35.20

Table 7：不同聚合策略对组重要性估计的影响

实验与结果

消融研究：聚合策略与层敏感性分析

- 实验：层敏感性分析
- 结果：模型不同层对剪枝的敏感性不同，其中初始层和最后层对性能影响尤其显著。

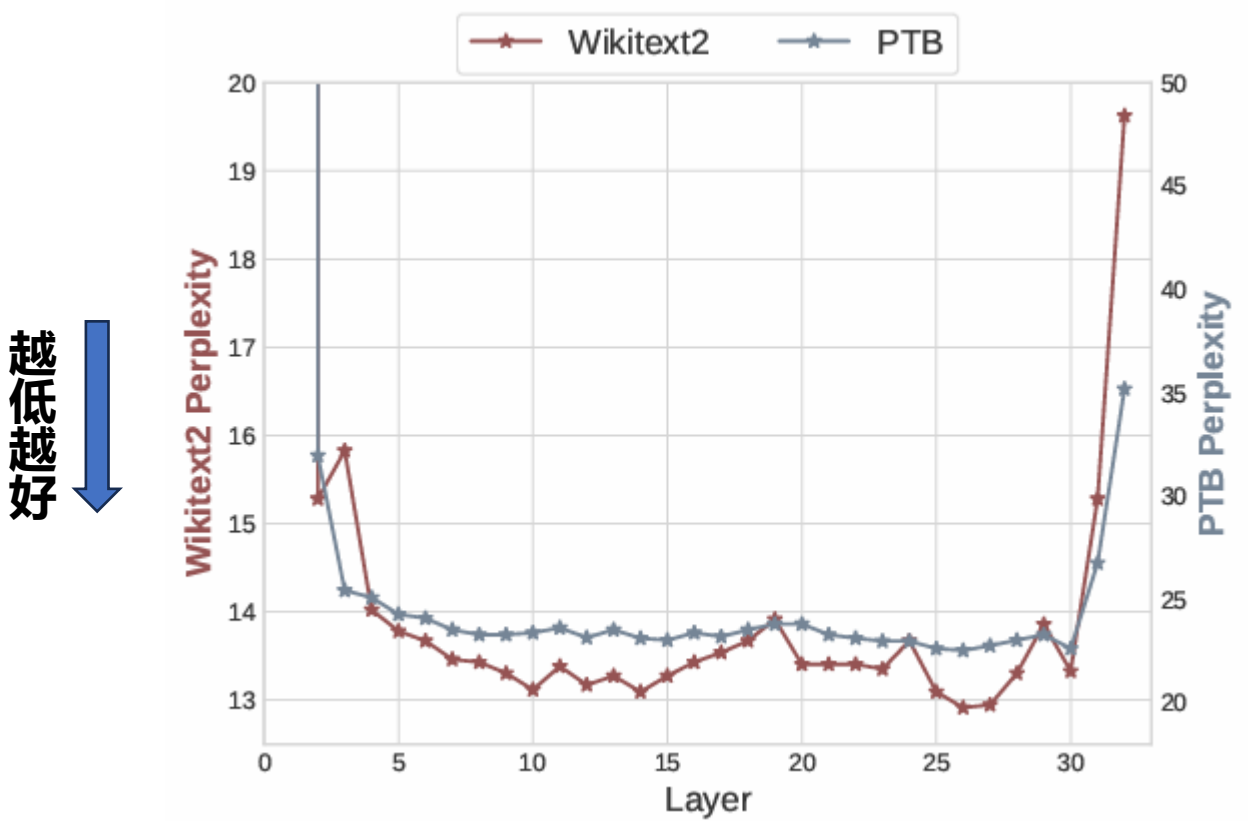


Figure 3：层敏感性分析：仅移除单层中的组

实验与结果

消融研究：聚合策略与层敏感性分析

- 小结：
  - 不同的重要性聚合策略会导致模型在不同下游任务上（如生成与分类）表现出权衡。
  - 模型不同层的重要性差异显著，特别是首尾层更为关键。
  - 消融研究表明，选择合适的聚合策略与层剪枝策略能显著优化剪枝效果，求和策略能在大多数情况下保持模型性能，而后层剪枝对性能影响较小。

实验与结果

微调数据量对恢复性能的影响

- 实验：数据量需求分析：50k vs. 2.59M
- 结果：仅使用 50k 样本进行微调即可实现与使用 2.59M 样本相近的模型性能恢复，突显了方法的数据高效性。

Model	#Samples	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average
LLaMA-7B	-	73.18	78.35	72.99	67.01	67.45	41.38	42.40	63.25
LLaMA-5.4B	50k [49]	64.62	77.20	68.80	63.14	64.31	36.77	39.80	59.23
LLaMA-5.4B	2.59M [59]	76.57	77.37	66.60	65.82	70.62	40.70	38.80	62.36

Table 10：模型恢复：50k 样本对比 2.59M 样本

实验与结果

微调数据量对恢复性能的影响

- 小结：
  - 少量数据已足够：使用 50K 样本 即可恢复 95% 的模型性能，是高效恢复的最佳平衡点。
  - 更多数据仍有效，但收益递减：进一步使用 2.59M 样本，性能提升有限，仅从 94.97% 增加到 98.02%。
  - LLM-Pruner 具备极强的数据效率：相比传统压缩方式（如蒸馏），无需大规模语料，适合快速部署。

实验与结果

消融研究：步长对性能的影响

实验：消融研究：后训练时长对性能的影响

- **结果：微调初期模型性能迅速恢复，但训练步长过长可能导致过拟合，损害泛化能力。**

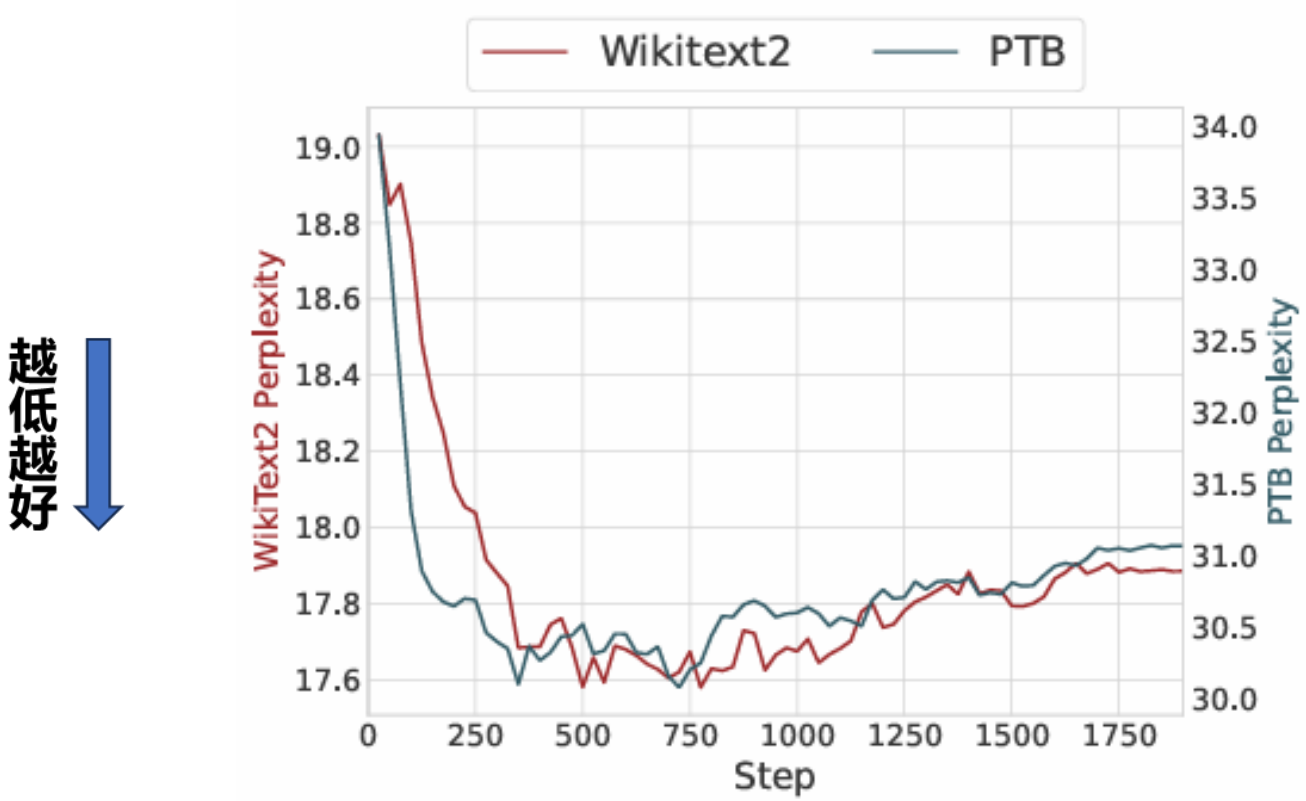


Figure 5：零样本数据集上的困惑度随训练步数的变化

实验与结果

消融研究：步长对性能的影响

- 小结：
  - 模型性能在后训练的早期阶段（少量步数/epoch）即可快速恢复。
  - 训练步长过长可能导致模型在微调数据集上过拟合，损害在其他通用任务上的泛化能力。
  - 实验支持采用短时高效微调策略（例如 2 个 epoch）。

目录

结论

LLM-Pruner: On the Structural Pruning Of Large Language Models

- LLM-Pruner 提出任务无关的结构化剪枝方法
- 依赖图驱动的结构分组机制
- 基于 LoRA 的快速恢复方案
- 在 20% 剪枝率下准确率几乎无损
- 消融研究验证依赖组剪枝优于单独剪枝
- 相比 DistilBERT、StableLM，LLM-Pruner 在准确率和效率上更具优势

目录

Thinking

- **跨模态模型的适配性**

  论文实验集中于文本类 LLM（LLaMA/ChatGLM），但对多模态模型（如 LLaVA、Flamingo）的剪枝效果尚未验证。多模态模型的跨模态交互层（如视觉编码器 - 文本解码器接口）依赖关系更复杂，现有依赖检测规则可能失效。

- **硬件兼容性与部署效率**

  论文仅测试 NVIDIA GPU（A100/4090），但边缘端设备（如 ARM 架构芯片，Arduino，树莓派）对模型稀疏性、计算密度有不同需求。依赖组剪枝的 “块状参数减少”（如整头剪除）是否比细粒度剪枝更适合低算力设备?

Thank you!