/TemplateVersion (2026.1)

Prune & Comp：通过带幅度补偿的迭代修剪对层修剪的 LLM 提供免费午餐

Xinrui Chen[1], Hongxing Zhang[2], Fanyi Zeng[1], Yongxian Wei[1], Yizhi Wang[1], Xitong Ling[1], Guanghao Li[1], Chun Yuan[1*]

[1] Shenzhen International Graduate School, Tsinghua University

[2] School of Information Science and Technology, Guangdong University of Foreign Studies cxr22@tsinghua.org.cn, yuanc@sz.tsinghua.edu.cn

Abstract

层剪枝已成为压缩大型语言模型（LLMs）的一个有前途的技术，同时实现与剪枝比例成正比的加速。在这项工作中，我们发现移除任何层都会在隐藏状态中引入显著的幅度差距，导致性能大幅下降。为了解决这一问题，我们提出了 Prune & Comp，一种新颖的即插即用层剪枝方案，通过幅度补偿以无训练方式来减轻这种差距。具体来说，我们首先估计由层移除引起的幅度差距，然后通过离线重新缩放剩余权重消除这一差距，不会产生运行时开销。我们进一步通过迭代剪枝策略展示了 Prune & Comp 的优势。当与一个迭代剪枝和补偿循环结合使用时，Prune & Comp 一直在提升现有的层剪枝度量。例如，当使用流行的块影响度量剪掉 LLaMA-3-8B 的 5 层时，Prune & Comp 几乎将困惑度减半，并保留原始模型问答性能的 93.19 %，优于基线 4.01 %。

介绍

近年来，大型语言模型（LLMs）在广泛的自然语言处理任务中取得了显著成功（Achiam et al. 2023; Jiang et al. 2023; Team 2025; Dubey et al. 2024; Team et al. 2025; Liu et al. 2024; Guo et al. 2025）。随着模型规模的增大，LLMs 表现出显著的性能提升；然而，庞大的参数数量带来了难以承受的计算成本和较长的推理延迟。因此，模型压缩领域引入了有效的 LLM 压缩方案，主要包括量化、知识蒸馏和剪枝（Sreenivas et al. 2024; Muralidharan et al. 2024; Ashkboos et al. 2024b; Sun et al. 2024; Ashkboos et al. 2024a; van der Ouderaa et al. 2023; Xia et al. 2023; Sarah et al. 2024; Hu et al. 2024）。其中，剪枝是一种有前景的方法，通过去除不重要的参数或组件来减小模型规模。

结构化剪枝是主流的剪枝范式。不同于半结构化稀疏或非结构化剪枝，这些方法不规则地消除权重或模块，因此导致不规则的内存访问，结构化剪枝避免依赖于专门的硬件或软件优化，并提供真正的加速。在结构化剪枝的文献中，主要技术分为深度剪枝和宽度剪枝。宽度剪枝去除不重要的权重通道和注意力头，从而缩小大语言模型（LLMs）的宽度。深度剪枝，也称为层剪枝，则丢弃整个 Transformer 层以减少模型深度。现有研究（Gromov et al. 2024; Kim et al. 2024）表明，在中等剪枝比例下，层剪枝可以在不重新训练的情况下保留大部分原始性能。此外，通过减少模型深度，层剪枝缩短了 LLM 推理延迟，并在相同的剪枝比例下实现更高的加速，而不需要硬件或软件的特定支持。

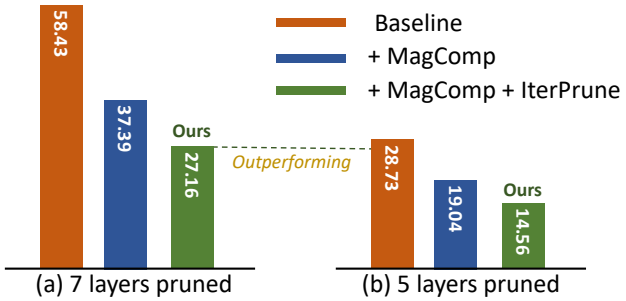

Figure 1: 在 WikiText-2、C4 和 PTB 数据集上修剪后的 LLaMA-3-8B 的平均困惑度（↓）。我们的 7 层修剪模型优于 5 层修剪基准。基准：使用 BI 指标（Men et al. 2024）的简单修剪；+MagComp：幅度补偿；+IterPrune：迭代修剪。

大多数层剪枝方法（Gromov et al. 2024; Kim et al. 2024; Song et al. 2024; Chen, Hu, and Zhang 2024; Men et al. 2024）都设计了复杂的层重要性度量来定位冗余层。基于层输出相似性（Men et al. 2024; Chen, Hu, and Zhang 2024; Gromov et al. 2024）、性能影响度量（Kim et al. 2024; Song et al. 2024）、以及基于梯度的度量（Kim et al. 2024; Ma, Fang, and Wang 2023）的代表性工作由于其强大的实证结果而引起了广泛关注。这些方法通常采用一次性或迭代策略来识别和移除冗余层。迭代方法考虑了层间依赖关系，通常能获得更具竞争力的性能（Kim et al. 2024; Song et al. 2024）。

尽管现有的层剪枝策略能够在不进行再训练的情况下保留大部分性能，但它们仍然存在明显的性能下降。我们调查了这一现象，并给出了两个关键观察：

(1) 我们证明了从一个大型语言模型中移除任何层都会在隐藏状态的量级上产生显著差距，如图 2 所示。这是大型语言模型的一个内在特性，并且与剪枝指标无关。我们的直觉是弥补这种量级差距，并展示简单弥补策略可以轻松恢复模型性能，如图 1 所示。

（2）尽管迭代层剪枝方法考虑了层间依赖关系，但已被破坏的剪枝模型会将错误注入随后的层重要性评估中。我们展示了一种在迭代过程中进行的简单补偿操作，能够让剪枝模型更好地识别冗余层，从而实现更优异的结果，如图 1 所示。

*Corresponding author.

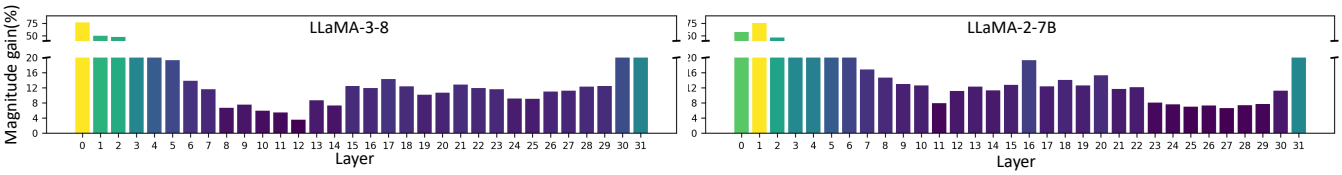

Figure 2: 通道平均幅度增益比的可视化，显示每一层的情况。所有层都产生幅度增益。

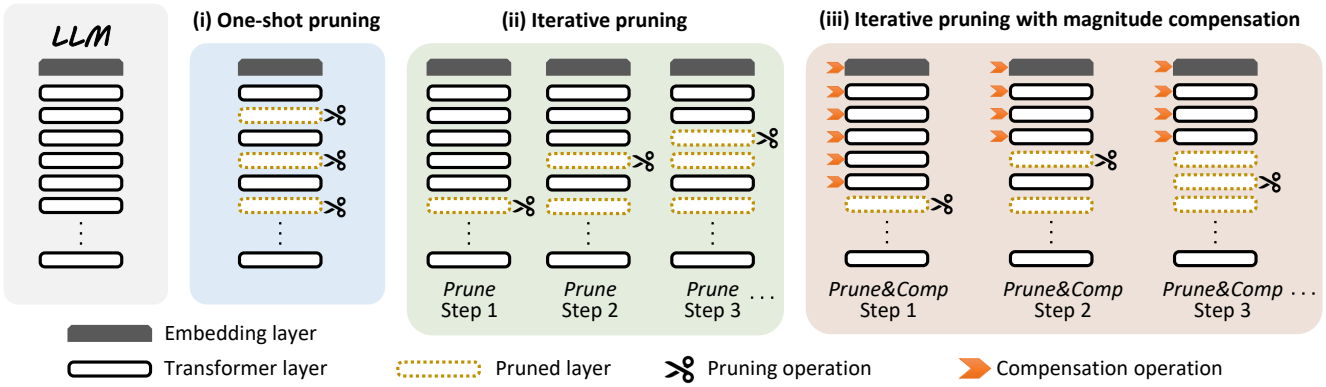

Figure 3: 常规剪枝策略与提出的 Prune & Comp 的比较。

基于这些观察，我们引入了一种简单而有效的方法 Prune & Comp ，通过权重修改来弥补由于层剪枝导致的幅度差距，并且不会产生在线推理开销。当与迭代层剪枝结合时，Prune & Comp 提高了剪枝模型的鲁棒性，从而增强了迭代层重要性估计的有效性。我们的主要贡献如下：

- 我们揭示了层裁剪不可避免地引入了大的幅度差距，导致大型语言模型（LLM）的性能严重下降。这种退化根植于模型的内在属性，而不论裁剪指标是什么。
- 我们引入了 Prune & Comp ，这是一种无需训练的方法，通过迭代层剪枝与幅度补偿相结合，以无需训练的方式弥合差距。
- 大量实验表明，Prune & Comp 始终可以在不增加在线推理开销的情况下，大幅提升常见的剪枝指标。

相关工作

宽度剪枝

宽度剪枝针对大型语言模型（LLMs），旨在通过选择性地去除冗余或信息量较少的通道、注意力头及其耦合结构来减少计算开销。LLM-Pruner （Ma, Fang, and Wang 2023）是第一个专为 LLMs 设计的结构化剪枝框架，它利用基于梯度的重要性估计来识别并去除非必要的耦合结构，在保留核心模型能力的同时实现大幅度压缩。Sheared LLaMA （Xia et al. 2023）通过结合有针对性的结构化剪枝和动态批量加载加速 LLM 的预训练，显示出更高的训练效率，并在相似规模的基准模型上表现出色。Wanda （Sun et al. 2023）介绍了一种无需再训练的技术，通过消除幅值与输入隐藏状态乘积最小的权重来引入稀疏性，提供了一种简单而有效的参数减少方法。类似地，FLAP （An et al. 2024）利用激活波动提出了一种新颖的无需再训练的结构化剪枝框架，从而提升存储效率和推理速度。

深度剪枝

与宽度剪枝相比，深度剪枝（也称为层剪枝）通过移除整个模型层来压缩大型语言模型（LLM），同时保留剩余组件的参数维度。在相同的剪枝率下，层剪枝可以在不依赖额外架构的情况下获得更高的加速。ShortGPT （Men et al. 2024）提出了块影响（BI），这是一种层级的重要性度量，捕捉层的输入和输出表示之间的语义变化。LLM-Streamline （Chen et al. 2024）使用余弦相似度来评估层的重要性，并引入了一个新的压缩指标——稳定性，以量化模型的结构敏感性。SLEB （Song et al. 2024）利用基于块级困惑度（PPL）评估的相邻变换器块的冗余。缩短的 LLaMA （Kim et al. 2024）采用了简单的深度剪枝方法，以基于梯度和幅度的指标来确定可移除的层。

动机

关于 LLM 层修剪的预备知识

大型语言模型主要基于 Transformer 架构构建，该架构由带有残差连接的 Transformer 解码器层堆栈组成。我们将第 ℓ 个 Transformer 层表示为 $f(X^{(\ell)}, \theta^{(\ell)})$ ，其中 $X^{(\ell)}$ 表示其输入隐状态，$\theta^{(\ell)}$ 表示其对应参数。鉴于在大型语言模型中广泛使用的预归一化架构，ℓ 层的输出可以表示为：

$$X^{(\ell+1)} = X^{(\ell)} + f(X^{(\ell)}, \theta^{(\ell)}). \quad (1)$$

为了移除从第 ℓ^* 层到第 $(\ell^* + n)$ 层的 LLM 层，我们跳过这些层，并将第 ℓ^* 层的输入传递给第 $(\ell^* + n)$ 层，即

$$X^{(\ell^*+n)} = X^{(\ell^*)} + f(X^{(\ell^*)}, \theta^{(\ell^*+n)}). \quad (2)$$

特别地，$n = 1$ 表示去除第 ℓ^* 层。

层剪枝产生幅度差距

我们定量分析每个单独的 LLM 层引入的增幅大小。对于每一层 ℓ，我们计算其输入与输出隐藏状态在一个校准集上的通道平均增幅比。层 ℓ 的增幅比定义为：

$$\boldsymbol{\delta}^{(\ell)} = \left(\mathbb{E}_{(X^{(\ell)}, X^{(\ell+1)}) \in \mathcal{D}} \frac{1}{C} \sum_{k}^{C} \frac{\|X_{:,k}^{(\ell+1)}\|_1}{\|X_{:,k}^{(\ell)}\|_1} - 1 \right) \times 100\%, \tag{3}$$

其中 $\mathcal{D}$ 表示校准集，$X^{(\ell)} \in \mathbb{R}^{B \times T \times C}$ 代表通过校准样本获得的第 ℓ 层的输入隐藏状态，批大小为 B，分词长度为 T，隐藏维度为 C。图 2 可视化了 LLaMA-3-8B 和 LLaMA-2-7B 的各个层的幅度增益比。可以观察到，每一层都显著增加了幅度，最大的增长超过了 70 %，并且发生在早期层。因此，不论剪枝指标如何，移除任何一层都不可避免地会产生相应的幅度差距。我们的直觉是补偿这个差距并保持隐藏状态的规模。如图 1 所示，简单补偿这个幅度差距 (+MagComp) 提高了性能。

重新思考迭代层剪枝

图 3 提供了单次剪枝、迭代剪枝和所提 Prune & Comp 方案的详细比较图。单次剪枝在一次遍历中完成其全部的剪枝决策，提供了较高的搜索效率。然而，同时移除多层忽略了层间依赖关系，累积影响可能远比单个效果的总和更具破坏性。相比之下，迭代剪枝在每次移除后重新评估层的重要性。这个过程捕捉到二阶交互作用，并通常能取得更好的精度-压缩折衷 (Kim et al. 2024; Song et al. 2024)。尽管如此，每一步迭代剪枝都会损害模型，进而妨碍对剩余层的稳健评估。受此启发，我们建议在迭代过程中对剪枝后的模型进行补偿，使其能够更准确地定位冗余层。

方法

层剪枝指标

在这项工作中，采用了以下五种流行的层剪枝指标来识别 LLMs 中的冗余层。

基于余弦相似度的度量

- CosSim(BI) (Men et al. 2024)：区块影响力 (BI) 通过评估一层的输入与输出之间的相似性来衡量每一层的重要性。ℓ 层的 BI 得分可以通过以下方式计算：

$$BI_\ell = \mathbb{E}_{X,t} \frac{(X_t^{(\ell)})^T X_t^{(\ell+1)}}{||X_t^{(\ell)}||_2 ||X_t^{(\ell+1)}||_2}, \tag{4}$$

  ，其中 $X_t^{(\ell)}$ 表示来自 t 标记位置的 ℓ 层输入隐藏状态的隐藏状态。更高的 BI 得分表示 $X^{(\ell)}$ 和 $X^{(\ell+1)}$ 之间具有较高的余弦相似度，表明 ℓ 层的冗余性。

- CosSim(CL) (Chen, Hu, and Zhang 2024; Gromov et al. 2024)：同样，输入和输出之间余弦相似度高的一系列连续层 (CL) 表明存在冗余。剪枝 n 层时，层 ℓ 到层 $\ell+n$ 之间 n 连续层的重要性可以计算为：

$$CL_{\ell,n} = \mathbb{E}_{X,t} \frac{(X_t^{(\ell)})^T X_t^{(\ell+n)}}{||X_t^{(\ell)}||_2 ||X_t^{(\ell+n)}||_2}, \tag{5}$$

困惑度 (PPL) (Song et al. 2024; Kim et al. 2024)：较低的 PPL 表明 LLM 文本生成的流畅性。冗余块对模型性能的贡献较小，其移除导致 PPL 增加较小。第 n 个块的 PPL 重要性得分 I_{PPL}^n 定义为：

$$I_{PPL}^\ell = \exp\left\{ -\frac{1}{ST} \sum_{s=1}^{S} \sum_{t=1}^{T} \log\ p_{\theta^\ell}(x_t^{(s)} | x_{<t}^{(s)}) \right\}, \tag{6}$$

其中 θ^ℓ 表示没有第 ℓ 个块的模型，$s = 1, \ldots, S$ 是序列的索引，$t = 1, \ldots, T$ 是校准集 D 中标记的索引。

Taylor 度量通过评估去除某个权重参数所导致的错误来估计其重要性。对于给定的校准数据集 D，这可以表示为训练损失 $\mathcal{L}$ 的变化：

$$|\mathcal{L}(W_{i,j}^{k,\ell}; D) - \mathcal{L}(W_{i,j}^{k,\ell} = 0; D)| \approx \left| \frac{\partial \mathcal{L}(D)}{\partial W_{i,j}^{k,\ell}} W_{i,j}^{k,\ell} \right|,$$

，其中省略了二阶导数。Taylor+ 块的重要性分数 I_{Taylor}^ℓ 可以定义为：

$$I_{Taylor}^\ell = \sum_k \sum_i \sum_j \left| \frac{\partial \mathcal{L}(D)}{\partial W_{i,j}^{k,\ell}} W_{i,j}^{k,\ell} \right|, \tag{7}$$

，其中 $W^{k,\ell}$ 是操作类型 k 在第 ℓ 个 Transformer 块中的线性权重矩阵，而 $W_{i,j}^{k,\ell}$ 是其元素。Taylor+ 方法通过保留前四个和最后两个 Transformer 块来建立在 Taylor 度量的基础上，因为去除这些块会导致严重的性能下降。较低的 I_{Taylor}^ℓ 分数表示一个重要性较低的块，因此更适合修剪。

幅度 + (Mag+) (Kim et al. 2024)：Mag+ 结合了一种基于 Magnitude 度量的启发式规则（参见 Li et al., 2017b，该规则假设范数较小的权重信息量较少），并保留了模型的前四个和最后两个模块。潜在的 Mag+ 重要性分数可以计算为：

$$I_{Magnitude}^\ell = \sum_k \sum_i \sum_j |W_{i,j}^{k,\ell}|, \tag{8}$$

，其中 $W_{i,j}^{k,\ell}$ 是 ℓ -th Transformer 块中操作类型 k 的线性权重矩阵的一个元素。较低的 $I_{Magnitude}^\ell$ 分数表示该块的重要性较低，因此更适合进行剪枝。

幅度补偿

幅度间隙估计 一旦通过某种剪枝指标确定层的重要性，最不重要的层便按照公式 2 被移除。如前所述，移除任何一层都会引入一个与剪枝指标无关的幅度差。为补偿这一差异，我们在剪枝前在一个小的校准集上估计幅度差。目标是获得一个最优的幅度补偿标量因子 α。假设要移除的层是 ℓ，则 α 被定义为：s

$$\alpha = \mathbb{E}_{(X^{(\ell)}, X^{(\ell+1)}) \in \mathcal{D}} \frac{1}{C} \sum_{k=1}^{C} \frac{\|X_{:,k}^{(\ell+1)}\|_1}{\|X_{:,k}^{(\ell)}\|_1}, \tag{9}$$

其中，$X^{(\ell)}$ 和 $X^{(\ell+1)}$ 分别是从校准样本中收集到的层 ℓ 输入和输出的隐藏状态。在估计 α 后，我们进行层剪枝，并用 α 作为补偿来缩放层 $\ell+1$ 的输入。形式上，包含补偿的前向传播变为：

$$X^{(\ell+1)} = \alpha X^{(\ell)} + f(\alpha X^{(\ell)}, \theta^{(\ell+1)}). \tag{10}$$

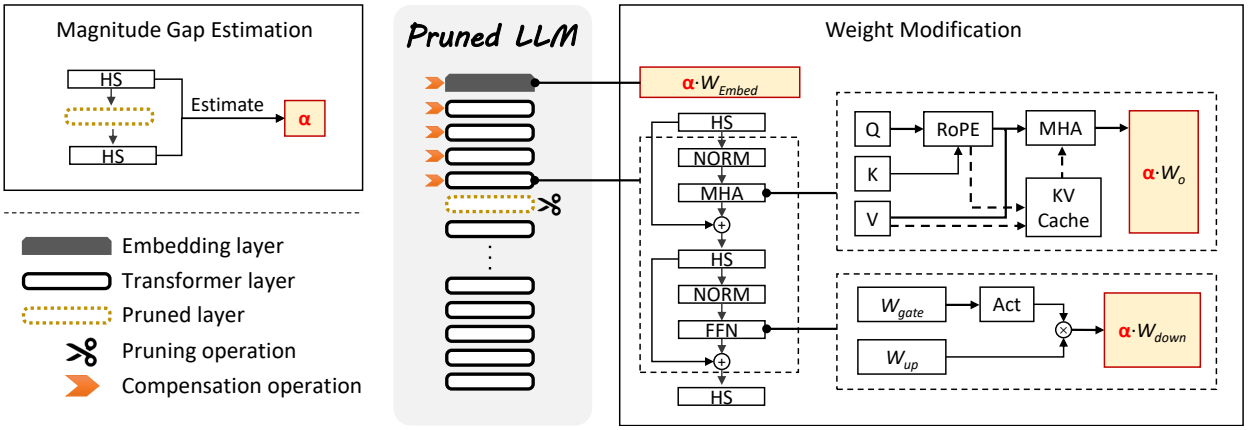

Figure 4: 应用于 LLM 的幅度补偿。高斯海赛堡 (HS) ： 隐藏状态； MHA ： 多头注意力；前馈神经网络 ： 前馈网络；W_{Embed} ： 嵌入权重；NORM：归一化层； RoPE ： 旋转位置嵌入；$W_{gate}, W_{up}, W_{down}$ ： FFN 门，上下投影权重；W_O ： MHA 输出投影权重；行为 ： 激活函数。

Table 1: 在困惑度（PPL）基准上的性能比较。稀疏度由已剪枝层数/总层数表示。

LLaMA-2-7B						LLaMA-3-8B					
Sparsity	Metric	WikiText-2	C4	PTB	Average	Sparsity	Metric	WikiText-2	C4	PTB	Average
-	Dense	5.47	6.97	22.51	11.65	-	Dense	6.14	8.88	10.59	8.54
7/32	PPL	9.81	12.36	48.53	23.57	5/32	PPL	12.37	15.28	18.91	15.52
	+Prune & Comp	8.57	10.55	40.51	19.88		+Prune & Comp	9.65	13.20	16.02	12.96
	CosSim(CL)	18.45	20.99	62.18	33.87		CosSim(CL)	21.14	24.13	37.41	27.56
	+Prune & Comp	13.78	15.31	49.40	26.16		+Prune & Comp	16.90	18.57	21.64	19.04
	Mag+	49.39	34.65	184.78	89.61		Mag+	37.57	34.99	60.80	44.45
	+Prune & Comp	11.73	13.28	59.60	28.20		+Prune & Comp	13.60	18.24	22.52	18.12
	Taylor+	18.45	20.99	63.02	34.15		Taylor+	602.96	388.41	546.98	512.78
	+Prune & Comp	10.61	12.10	51.02	24.58		+Prune & Comp	12.78	16.20	20.03	16.34
	CosSim(BI)	18.45	20.99	62.18	33.87		CosSim(BI)	27.33	27.06	31.81	28.73
	+Prune & Comp	11.45	12.96	42.29	22.23		+Prune & Comp	11.87	14.87	16.93	14.56
9/32	PPL	14.91	17.03	67.73	33.22	7/32	PPL	15.08	17.57	22.09	18.2
	+Prune & Comp	10.23	12.08	50.96	24.42		+Prune & Comp	12.40	15.98	20.27	16.22
	CosSim(CL)	35.68	36.10	96.52	56.10		CosSim(CL)	2287.73	1491.37	4738.81	2839.30
	+Prune & Comp	19.37	20.13	58.20	32.57		+Prune & Comp	204.06	231.6	256.53	230.73
	Mag+	362.15	48.79	273.07	228.00		Mag+	40.70	36.95	44.85	40.83
	+Prune & Comp	19.09	18.88	95.98	44.65		+Prune & Comp	33.33	35.02	42.93	37.09
	Taylor+	35.68	36.10	96.52	56.10		Taylor+	2287.86	1491.38	4741.9	2840.38
	+Prune & Comp	13.80	14.52	69.04	32.45		+Prune & Comp	21.10	22.05	32.47	25.21
	CosSim(BI)	35.68	36.10	96.52	56.10		CosSim(BI)	57.76	50.13	67.39	58.43
	+Prune & Comp	18.53	17.99	60.38	32.30		+Prune & Comp	28.43	24.48	28.57	27.16

权重修改 方程 10 在推理期间通过元素逐乘将隐藏状态按标量 α 进行缩放，从而产生在线开销。我们通过直接将 α 融合到修剪层之前的模型层的权重中来消除这种成本。如图 4 所示，一旦估计出 α ，它会通过三个步骤融合到模型权重中。

步骤 1：修改嵌入层。词元嵌入层 W_{embed} 的权重被更新：

$$W_{embed} \leftarrow \alpha W_{embed}. \tag{11}$$

步骤 2：修改 MHA 的输出投影。对于每个索引为 k 的层，其中 $\in [1, \ell\text{ -1}]$，MHA 的输出投影矩阵 W_o 被更新：

$$W_o^{(k)} \leftarrow \alpha W_o^{(k)}. \tag{12}$$

步骤 3：修改 MLP 向下投影。同样，每个索引为 $\in [1, \ell\text{ -1}]$ 的 MLP 层的向下投影权重 W_{Down} 被缩放：

$$W_{down}^{(k)} \leftarrow \alpha W_{down}^{(k)}. \tag{13}$$

我们逐步解释权重修改的原理。首先，在步骤 1 中，嵌入层的输出隐藏状态按比例缩放为 α 。后续的 Transformer 层包括多头注意力机制（MHA）、多层感知机（MLP）和归一化层（NORM）。第一个 Transformer 层接收来自嵌入层的缩放隐藏状态，然后隐藏状态依次被 (i) 复制为残差分支，(ii) 归一化，(iii) 送入 MHA，(iv) 添加到残差中。对于一个尺度不变的归一化层，即 NORM(X)=NORM(α X)，我们重新引入因子 α ，并将 MHA 的输出投影权重乘以 α ，这实际上在残差相加之前将 MHA 的输出按比例缩放为 α 。同样的逻辑也适用于 MLP 块。缩放后的 MHA 输出被复制为残差，归一化后传递给 MLP。再次由于归一化的尺度不变性，我们将 MLP 的下投影权重乘以 α ，以保持量级。然后我们将此过程应用于修剪层 ℓ 之前的每一层。通过这些

Table 2: 在问答（QA）基准上的性能比较。稀疏度通过被剪枝层/总层（压缩率）来表示。RP 表示相对性能（% ）。

Model	Sparsity	Metric	ARC-c	ARC-e	BoolQ	CoPa	HeSw	PIQA	Race-h	WG	WSC	Average	RP
LLaMA-3-8B	0/32	Dense	53.41	77.78	81.28	89.00	79.16	80.85	40.19	72.85	86.45	73.44	100.00
	5/32 (13.58 %)	PPL	32.76	61.36	56.42	75.00	61.77	75.52	32.25	54.22	65.20	57.17	77.84
		+Prune & Comp	40.87	66.58	56.27	85.00	67.44	76.22	33.97	62.59	73.26	62.47	85.06
		CosSim(CL)	47.35	66.20	73.52	84.00	71.10	74.27	36.65	71.03	76.56	66.74	90.88
		+Prune & Comp	48.63	69.99	74.07	85.00	72.63	75.95	37.51	72.61	79.12	68.39	93.12
		Mag+	29.95	56.36	53.21	73.00	40.35	69.64	27.18	52.80	60.44	51.44	70.04
		+Prune & Comp	32.76	57.87	58.96	82.00	59.85	73.23	30.72	55.25	66.30	57.44	78.21
		Taylor+	33.53	45.58	55.00	55.00	35.86	59.52	24.31	60.85	65.57	48.36	65.85
		+Prune & Comp	46.16	70.24	69.36	81.00	69.67	73.83	37.99	70.56	79.85	66.52	90.57
		CosSim(BI)	45.56	63.51	73.12	79.00	70.13	74.92	36.94	71.19	75.09	65.50	89.18
		+Prune & Comp	46.50	70.54	71.31	84.00	72.43	76.01	37.99	73.32	83.88	68.44	93.19
	7/32 (19.01 %)	PPL	32.76	58.84	45.38	75.00	59.22	73.56	30.72	53.83	67.77	55.23	75.20
		+Prune & Comp	33.53	60.48	47.52	76.00	59.03	73.56	30.72	54.54	67.40	55.86	76.07
		CosSim(CL)	28.92	39.56	38.07	60.00	33.26	59.47	24.02	55.56	59.71	44.29	60.30
		+Prune & Comp	32.76	45.62	52.20	66.00	43.41	63.55	27.66	57.85	62.64	50.19	68.34
		Mag+	25.60	46.04	56.18	70.00	43.36	64.91	27.46	53.43	55.31	49.14	66.92
		+Prune & Comp	30.63	49.37	58.10	76.00	51.31	68.82	28.71	55.01	60.81	53.20	72.43
		Taylor+	29.01	39.56	38.00	60.00	33.24	59.30	24.02	55.49	59.71	44.26	60.27
		+Prune & Comp	42.66	67.51	67.61	78.00	65.84	72.36	37.22	70.01	77.66	64.32	87.58
		CosSim(BI)	42.41	56.65	65.26	75.00	64.70	70.89	34.16	71.19	73.63	61.54	83.80
		+Prune & Comp	41.55	62.37	72.78	80.00	65.44	71.38	36.65	71.11	78.02	64.37	87.64
Qwen3-8B	0/32	Dense	56.57	80.93	86.57	85.00	74.93	77.80	40.96	67.80	83.15	72.63	100.00
	5/36 (11.78 %)	PPL	46.93	74.41	69.63	82.00	76.06	77.20	38.37	61.25	75.46	66.81	91.98
		+Prune & Comp	46.84	74.33	75.17	78.00	61.39	76.44	37.80	58.88	74.36	64.80	89.22
		CosSim(CL)	42.41	61.15	77.98	73.00	58.80	67.30	33.78	65.27	72.16	61.32	84.42
		+Prune & Comp	43.17	65.57	86.30	77.00	60.82	69.15	37.03	67.40	77.29	64.86	89.29
		Mag+	42.75	70.03	77.00	77.00	63.86	75.90	36.46	58.56	73.26	63.87	87.93
		+Prune & Comp	45.31	75.51	72.75	82.00	61.82	74.27	39.52	60.54	74.73	65.16	89.71
		Taylor+	40.02	63.38	55.90	73.00	62.05	68.99	37.89	59.98	71.79	59.22	81.53
		+Prune & Comp	43.09	66.08	82.91	72.00	62.57	69.75	37.51	62.83	78.39	63.90	87.98
		CosSim(BI)	46.42	73.74	77.40	80.00	64.54	76.82	37.89	62.75	76.19	66.19	91.13
		+Prune & Comp	47.27	74.07	81.56	82.00	63.54	77.09	37.80	63.06	79.12	67.28	92.63
	7/36 (16.49 %)	PPL	41.13	68.64	67.77	74.00	60.44	74.59	35.02	55.64	69.23	60.72	83.59
		+Prune & Comp	43.60	72.26	68.23	77.00	58.64	75.03	36.84	56.99	72.16	62.31	85.78
		CosSim(CL)	33.87	48.57	73.55	69.00	50.97	61.86	31.96	59.98	67.03	55.20	76.00
		+Prune & Comp	35.67	51.52	84.59	72.00	54.56	64.53	34.45	62.59	74.36	59.36	81.73
		Mag+	40.10	68.27	55.60	71.00	58.9	73.12	33.59	56.75	69.23	58.51	80.56
		+Prune & Comp	37.54	63.55	76.33	75.00	55.80	71.22	35.89	59.04	71.06	60.60	83.44
		Taylor+	35.75	49.62	45.17	67.00	53.29	64.53	34.64	57.54	64.47	52.45	72.20
		+Prune & Comp	36.60	54.50	85.50	67.00	56.31	65.78	35.50	62.51	72.53	59.58	82.03
		CosSim(BI)	42.49	69.07	68.35	80.00	61.48	75.24	35.50	54.38	67.77	61.59	84.79
		+Prune & Comp	40.70	69.44	74.07	76.00	58.86	74.70	34.93	57.38	68.86	61.66	84.89

修改，模型架构除去被去除的层之外保持不变，在推理时不产生额外的在线成本。

带幅度补偿的迭代剪枝（ Prune & Comp ）

此外，我们将迭代修剪与幅度补偿结合在一起。Prune & Comp 从原始模型 M 中迭代删除层，直到消除目标数量的 n 层，如算法 ?? 所总结。在每次迭代中，算法执行以下三个步骤：

- $Metric(M, C)$：给定模型 M ，在校准数据集 C 上使用选择的剪枝度量选择最冗余的层进行移除，并返回其索引 idx 。
- $Prune(M, idx)$：从 M 模型中移除索引为 idx 的块，并返回修剪后的模型。
- $Comp(M, idx)$：对因层移除而导致的隐藏状态中的量级差异进行补偿，并返回补偿后的模型 M 。

通过无训练的幅度差距估计和权重修改来积极补偿这个差距，Prune & Comp 确保裁剪后的模型保持强大的内部表示，这反过来增强了后续层重要性估计的准确性，并最终保持性能。这种修剪和补偿的迭代循环使算法能够有效减少模型深度，同时最小化性能下降，从而生成一个更精简和高效的 LLM，避免产生在线推理开销。

实验

实验设置和细节

我们在开源的大型语言模型上评估了 Prune & Comp ，包括 LLaMA-2-7B/13B (Touvron et al. 2023) 、LLaMA-3-8B (Dubey et al. 2024) 、Qwen3-8B (Team 2025) 。

我们以一次性方式比较流行的逐层剪枝指标，包括基于余弦相似度的指标 (Men et al. 2024; Chen, Hu, and Zhang 2024; Gromov et al. 2024) 、基于困惑度（PPL）的指标 (Song et al. 2024; Kim et al. 2024) 、Taylor+ (Kim et al. 2024) 和 Magnitude+ (Kim et al. 2024) ，如方法中所述。为了确定被剪枝的层并初始化幅度补偿因子，从 WikiText-2 数据集中随机选择长度为 2048 个标记的 128 个序列。所有实验均在 24GB 的 NVIDIA V100 GPU 上进行。

使用了三个基准进行评估: 困惑度 (PPL)，包括 WikiText2 (Merity et al. 2016) 、C4 (Raffel et al. 2020) 和 PTB (Marcus, Santorini, and Marcinkiewicz 1993) ；大

Table 3: 在 MMLU 基准测试上的性能比较。5 层的 LLaMA-3-8B 通过 CosSim(BI) 指标剪枝。

Metric	STEM	Humanities	Social Sciences	Others	Weighted Accuracy
PPL	28.76	24.36	27.10	28.22	26.80
+Prune & Comp	30.72	25.59	32.17	30.44	29.25
CosSim(CL)	53.47	56.08	74.58	68.32	62.40
+Prune & Comp	54.31	56.98	75.04	70.76	63.55
Mag+	27.24	23.53	24.76	27.64	25.54
+Prune & Comp	28.10	24.14	31.04	25.91	26.91
Taylor+	31.41	37.4	44.65	41.39	38.64
+Prune & Comp	39.36	43.97	57.26	54.90	48.42
CosSim(BI)	46.92	53.92	65.65	65.42	57.64
+Prune & Comp	49.93	52.09	69.45	67.46	58.98

Table 4: 所提出的 Prune & Comp 的有效性。

Method	WikiText-2	C4	PTB	Average
Naive one-shot pruning	57.76	50.13	67.39	58.43
+IterPrune	36.91	32.2	46.00	38.37
+MagComp	35.38	33.71	43.08	37.39
+MagComp +IterPrune	28.43	24.48	28.57	27.16

型多任务语言理解 (MMLU) (Hendrycks et al. 2020) ；常识性问答 (QA)，包括 ARC-Challenge (ARC-c)、ARC-Easy (ARC-e) (Clark et al. 2018) 、BoolQ (Clark et al. 2019) 、HellaSwag (HeSw) (Zellers et al. 2019) 、PIQA (Bisk et al. 2020) 、WinoGrande (WG) (Sakaguchi et al. 2020) 、WSC273 (WSC) (Levesque, Davis, and Morgenstern 2012) 、Race-high (Race-h) (Lai et al. 2017) 和 CoPA (Sarlin et al. 2020) 。

PPL 基准测试结果

表格 1 对于 LLaMA-2-7B 和 LLaMA-3-8B 在不同剪枝配置下的困惑度基准进行了分析。我们报告了在 WikiText-2、C4 和 PTB 数据集上的平均性能。

对于 LLaMA-2-7B，插入式 Prune & Comp 在所有剪枝指标和数据集上均一致地改善了基准。例如，当使用 CosSim(BI) 指标剪掉 32 层中的 7 层时，结合 Prune & Comp 将平均 PPL 从 23.57 降至 19.88，而使用 Mag+ 指标时，Prune & Comp 将 PPL 显著从 89.61 提升至 28.20。通过 Prune & Comp 达到显著 PPL 降低的这一趋势在更高稀疏度下依然存在，证明了其在减轻层剪枝导致的性能下降方面的有效性。同样地，LLaMA-3-8B 从 Prune & Comp 中持续获得大幅提升，与每种基准结合时都在各指标上表现出更低的困惑度。最突出的例子是 Taylor+ 剪掉 32 层中的 5 层：当与 Prune & Comp 结合时，其平均 PPL 从极高的 512.78 大幅降至 16.34。Prune & Comp 的优势在剪掉 32 层中的 7 层时更加明显。Taylor+ 指标结合 Prune & Comp 时从 2840.38 骤降至 25.21，而 CosSim(CL) 指标结合 Prune & Comp 同样大幅下跌，从 2839.30 降至 230.73。

表 2 展示了对于 LLaMA-3-8B 和 Qwen3-8B 在多种剪枝配置下的问答（QA）基准的综合分析。

对于 LLaMA-3-8B 模型，与其基线相比，结合 Prune & Comp 在所有剪枝指标上持续提升了剪枝模型的性能，证明了其有效性。例如，当使用 Taylor+ 指标从 32 层中剪掉 5 层时，相关性能显著从 65.85 % 提高到 90.57 %，领先基线 24.72 %。同样地，当剪掉 7 层时，使用 Taylor+ 指标，Prune & Comp 将相关性能从 60.27 提高到 87.58，显著高于基线。对于 Qwen3-8B，也观察到了类似的正向趋势。通过整合 Prune & Comp ，模型在所有剪枝指标上的相对性能都有所提升。例如，当使用 CosSim(CL) 指标从 32 层中剪掉 5 层时，Prune & Comp 将相对性能从 84.42 % 提升到 89.29 %，超过基线 4.87 %。

对大量多任务语言理解 (MMLU) 基准的进一步分析总结在表 3 中。对于大多数子任务，传统的剪枝方法结合 Prune & Comp 可实现卓越的性能，展示了它们在提高模型跨多种知识领域性能方面的强大能力。

例如，当对 LLaMA-3-8B 的 32 层中剪枝 5 层时，使用 Taylor+ 度量，Prune & Comp 显著提升了加权准确率，从 38.64 % 提升到 48.42 %，比基线高出 9.78 %。同样，对于 CosSim(CL) 度量，性能从 62.40 % 提高到 63.55 %。这些结果表明，Prune & Comp 在多种剪枝度量中保持了卓越的性能。

消融研究

表中 4 调查了 Prune & Comp 的个体贡献，包括迭代剪枝 (+IterPrune) 和幅度压缩 (+MagComp)，对于困惑度基准。迭代剪枝和幅度压缩单独比较于简单的一次性剪枝，都在 WikiText-2、C4 和 PTB 数据集上产生了显著的改进。具体来说，使用简单的一次性剪枝，困惑度平均为 58.43，迭代剪枝将其降低到 38.37，而幅度压缩则实现了 37.39 的平均值。关键是，两者结合达到最佳的平均困惑度 27.16，显示出强大的协同效应，其组合应用显著优于单独的组件。

我们提出了 Prune & Comp ，这是一种无需训练即插即用的层裁剪方案，通过幅度补偿来缓解在大型语言模型（LLM）中因移除层而产生的幅度差异。通过将这种补偿与迭代裁剪相结合，该方法涉及估计由层裁剪引起的差距，并通过离线的权重重缩放来弥补这一差距，我们的方法避免了推理的开销，同时解决了由隐藏状态差异驱动的性能下降问题。Prune & Comp consistently 提升了现有的层裁剪指标，验证了其在实际 LLM 压缩中的有效性。该工作促进了在资源受限环境中的 LLM 压缩，并标志着向开发无需训练的层裁剪方案迈出的重要一步。

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. arXiv preprint arXiv:2303.08774.

An, Y.; Zhao, X.; Yu, T.; Tang, M.; and Wang, J. 2024. Fluctuation-based adaptive structured pruning for large language models. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 38, 10865–10873.

Ashkboos, S.; Croci, M. L.; Nascimento, M. G. d.; Hoefler, T.; and Hensman, J. 2024a. Slicegpt: Compress large language models by deleting rows and columns. arXiv preprint arXiv:2401.15024.

Ashkboos, S.; Mohtashami, A.; Croci, M. L.; Li, B.; Jaggi, M.; Alistarh, D.; Hoefler, T.; and Hensman, J. 2024b. Quarot: Outlier-free 4-bit inference in rotated llms. arXiv preprint arXiv:2404.00456.

Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In Proceedings of the AAAI conference on artificial intelligence, 7432–7439.

Chen, X.; Hu, Y.; and Zhang, J. 2024. Compressing large language models by streamlining the unimportant layer. arXiv preprint arXiv:2403.19135.

Chen, X.; Hu, Y.; Zhang, J.; Wang, Y.; Li, C.; and Chen, H. 2024. Streamlining redundant layers to compress large language models. arXiv preprint arXiv:2403.19135.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. arXiv preprint arXiv:1905.10044.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. arXiv preprint arXiv:1803.05457.

Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Yang, A.; Fan, A.; et al. 2024. The llama 3 herd of models. arXiv preprint arXiv:2407.21783.

Gromov, A.; Tirumala, K.; Shapourian, H.; Glorioso, P.; and Roberts, D. 2024. The Unreasonable Ineffectiveness of the Deeper Layers. In NeurIPS 2024 Workshop on Scientific Methods for Understanding Deep Learning.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. arXiv preprint arXiv:2501.12948.

Hendrycks, D.; Burns, C.; Basart, S.; Zou, A.; Mazeika, M.; Song, D.; and Steinhardt, J. 2020. Measuring massive multitask language understanding. arXiv preprint arXiv:2009.03300.

Hu, Y.; Zhang, J.; Zhao, Z.; Zhao, C.; Chen, X.; Li, C.; and Chen, H. 2024. SP3: Enhancing Structured Pruning via PCA Projection. In Findings of the Association for Computational Linguistics ACL 2024, 3150–3170.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; Casas, D. d. l.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; et al. 2023. Mistral 7B. arXiv preprint arXiv:2310.06825.

Kim, B.-K.; Kim, G.; Kim, T.-H.; Castells, T.; Choi, S.; Shin, J.; and Song, H.-K. 2024. Shortened llama: A simple depth pruning for large language models. arXiv preprint arXiv:2402.02834, 11.

Lai, G.; Xie, Q.; Liu, H.; Yang, Y.; and Hovy, E. 2017. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683.

Levesque, H.; Davis, E.; and Morgenstern, L. 2012. The winograd schema challenge. In Thirteenth international conference on the principles of knowledge representation and reasoning.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024. Deepseek-v3 technical report. arXiv preprint arXiv:2412.19437.

Ma, X.; Fang, G.; and Wang, X. 2023. Llm-pruner: On the structural pruning of large language models. Advances in neural information processing systems, 36: 21702–21720.

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. Computational Linguistics, 19(2): 313–330.

Men, X.; Xu, M.; Zhang, Q.; Wang, B.; Lin, H.; Lu, Y.; Han, X.; and Chen, W. 2024. Shortgpt: Layers in large language models are more redundant than you expect. arXiv preprint arXiv:2403.03853.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer sentinel mixture models. arXiv preprint arXiv:1609.07843.

Muralidharan, S.; Turuvekere Sreenivas, S.; Joshi, R.; Chochowski, M.; Patwary, M.; Shoeybi, M.; Catanzaro, B.; Kautz, J.; and Molchanov, P. 2024. Compact language models via pruning and knowledge distillation. Advances in Neural Information Processing Systems, 37: 41076–41102.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. Journal of machine learning research, 21(140): 1–67.

Sakaguchi, K.; Le Bras, R.; Bhagavatula, C.; and Choi, Y. 2020. Winogrande: An adversarial winograd schema challenge at scale. In Proceedings of the AAAI Conference on Artificial Intelligence, 8732–8740.

Sarah, A.; Sridhar, S. N.; Szankin, M.; and Sundaresan, S. 2024. LLaMA-NAS: Efficient Neural Architecture Search for Large Language Models. arXiv preprint arXiv:2405.18377.

Sarlin, P.-E.; DeTone, D.; Malisiewicz, T.; and Rabinovich, A. 2020. Superglue: Learning feature matching with graph neural networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 4938–4947.

Song, J.; Oh, K.; Kim, T.; Kim, H.; Kim, Y.; and Kim, J.-J. 2024. SLEB: Streamlining LLMs through Redundancy Verification and Elimination of Transformer Blocks. arXiv preprint arXiv:2402.09025.

Sreenivas, S. T.; Muralidharan, S.; Joshi, R.; Chochowski, M.; Mahabaleshwarkar, A. S.; Shen, G.; Zeng, J.; Chen, Z.; Suhara, Y.; Diao, S.; et al. 2024. Llm pruning and distillation in practice: The minitron approach. arXiv preprint arXiv:2408.11796.

Sun, M.; Liu, Z.; Bair, A.; and Kolter, J. Z. 2023. A simple and effective pruning approach for large language models. arXiv preprint arXiv:2306.11695.

Sun, Y.; Liu, R.; Bai, H.; Bao, H.; Zhao, K.; Li, Y.; Hu, J.; Yu, X.; Hou, L.; Yuan, C.; et al. 2024. Flatquant: Flatness matters for llm quantization. arXiv preprint arXiv:2410.09426.

Team, K.; Du, A.; Gao, B.; Xing, B.; Jiang, C.; Chen, C.; Li, C.; Xiao, C.; Du, C.; Liao, C.; et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. arXiv preprint arXiv:2501.12599.

Team, Q. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.

van der Ouderaa, T. F.; Nagel, M.; Van Baalen, M.; Asano, Y. M.; and Blankevoort, T. 2023. The llm surgeon. arXiv preprint arXiv:2312.17244.

Xia, M.; Gao, T.; Zeng, Z.; and Chen, D. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. arXiv preprint arXiv:2310.06694.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? arXiv preprint arXiv:1905.07830.