Compress Large Language Models via Collaboration Between Learning and Matrix Approximation

Yuesen Liao[1*], **Zhiwei Li**[1*], **Binrui Wu**[1*], **Zihao Cheng**[1]
Su Zhao[2], **Shuai Chen**[2], **Weizhong Zhang**[1,3†]
[1]Fudan University, [2]Meituan Inc.
[3]Shanghai Key Laboratory of Intelligent Information Processing
{ysliao24, zwli23, brwu23, zhcheng25}@m.fudan.edu.cn
{zhaosu04, chenshuai31}@meituan.com
weizhongzhang@fudan.edu.cn

Abstract

Sparse and low-rank matrix composite approximation has emerged as a promising paradigm for compressing large language models (LLMs), offering a more flexible pruning structure than conventional methods based solely on sparse matrices. The significant variation in weight redundancy across layers, along with the differing rank and sparsity structures of weight matrices, makes identifying the globally optimal pruning structure extremely challenging. Existing methods often depend on uniform or manually designed heuristic rules to allocate weight sparsity across layers, subsequently compressing each matrix using matrix approximation techniques. Given the above theoretical difficulty in global compression of LLMs and the limited computational and data resources available compared to the training phase, we argue that a collaboration between learning and matrix approximation is essential for effective compression. In this paper, we propose a novel LLM compression framework based on generalized bilevel optimization that naturally formulates an effective collaborative mechanism. Specifically, the outer loop frames the weight allocation task as a probabilistic optimization problem, enabling the automatic learning of both layer-wise sparsities and matrix-wise retained ranks, while the inner loop solves the corresponding sparsity and rank-constrained model compression problem via matrix approximation. Our main technical contributions include two key innovations for efficiently solving this bilevel optimization problem. First, we introduce a truncated Gaussian prior-based probabilistic parameterization integrated with a policy gradient estimator, which avoids expensive backpropagation and stabilizes the optimization process. Second, we design an adapted QR-based matrix approximation algorithm that significantly accelerates inner loop computations. Extensive experiments on Phi-3 and the LLama-2/3 family demonstrate the effectiveness of our method. Notably, it maintains over 95% zero-shot accuracy under 50% sparsity and achieves up to 2× inference speedup.

1 Introduction

Model compression [8, 22, 29, 31] is a widely adopted paradigm for improving the inference efficiency of large language models (LLMs). Its core principle is to reduce model size by removing redundant parameters or approximating the model weights with low-rank matrices [37] while preserving the performance. Although promising results have been repeatedly reported in the literature,

[*]Equal Contribution.
[†]Corresponding Author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

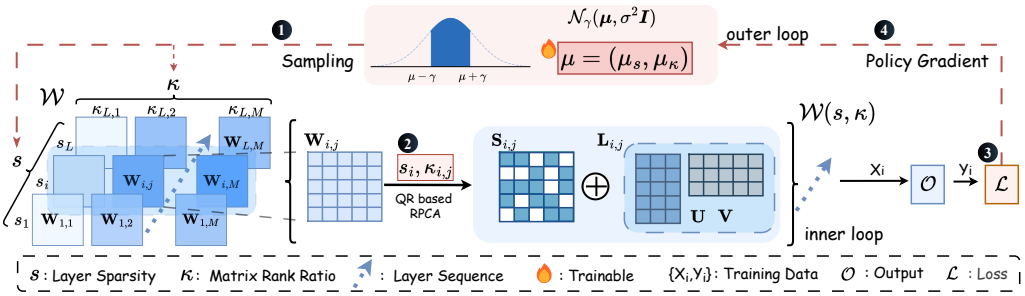


Figure 1: Process diagram of our bilevel framework. ❶: sample $\boldsymbol{s}$ and $\boldsymbol{\kappa}$ from $\mathcal{N}_\gamma(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$ to assign sparsity allocation for each layer; ❷: compress matrix $\mathbf{W}_{i,j} \in \mathcal{W}$ via adapted QR-based RPCA under the sparsity allocation s_i and $\kappa_{i,j}$; ❸: forwardpass the compressed model $\mathcal{W}(\boldsymbol{s}, \boldsymbol{\kappa})$ and get the loss; ❹: update the distribution $\mathcal{N}_\gamma(\boldsymbol{\mu}, \sigma^2 \boldsymbol{I})$ based on policy gradient estimator.

challenges remain due to the complex model architecture, vast optimization space, and limited data and computational resources compared to those available during the training stage.

In this paper, we focus on the emerging compression paradigm based on sparse and low-rank matrix composite approximation [17, 20, 44], referred to as robust principal component analysis (RPCA) [4] in the field of classical matrix analysis, which adopts a more flexible pruning structure than conventional methods based solely on sparse matrices. Existing methods [17, 44] typically adopt a uniform sparsity allocation over layers, i.e., setting an equal pruning proportion for each layer and subsequently compressing each matrix using matrix approximation techniques. Recognizing the heterogeneous redundancy across layers, recent works [18, 19, 40, 41] have introduced manually designed heuristic rules to allocate varying sparsity levels to different layers.

However, the performance of these methods is often less effective than expected. The main reason is the significant variation in weight redundancy across layers, along with differing rank and sparsity structures of weight matrices. These factors make finding the globally optimal pruning structure extremely challenging. This highlights the need for layer-wise sparsity and matrix-wise rank allocation in RPCA-based compression methods. Given the theoretical difficulty of global compression for LLMs and the limited computational resources and data compared to the training phase, we argue that collaboration between learning and matrix approximation is essential for effective compression.

In this paper, we propose a novel bilevel optimization framework [15, 30] that naturally formulates an effective collaborative mechanism. In line with recent perspectives [19, 40, 41], we adopt the view that once a global sparsity allocation is provided, the compression task can be reduced to a matrix approximation problem. Instead of metric-based heuristics [19, 41], we model the weight allocation task of **outer loop** as a probabilistic optimization problem, enabling the automatic learning of both layer-wise sparsities and matrix-wise retained ranks, while the **inner loop** solves the corresponding RPCA subproblem to obtain the sparse and low-rank decomposition under the current allocation scheme. The bilevel framework poses difficulties due to the implicit differentiation through the inner loop solutions and the substantial computational overhead of the inner RPCA problem. To address these challenges, we introduce the following two key technical innovations. First, for the outer loop, we use a truncated Gaussian prior to enable continuous probabilistic modeling within bounded support. The truncation helps stabilize training by preventing gradient explosion in low-density regions. Through this reparameterization, we apply policy gradient [32] to update the prior parameters without backpropagating through the compressed model, reducing memory overhead. Second, instead of costly SVD-based solvers [21, 49], we use an adapted QR-based matrix fitting scheme [42], which significantly accelerates inner loop computations. Our method is intuitively visualized in Figure 1. Empirical results on the Phi-3 and Llama family model show that our method consistently learns better compression configurations and achieves superior performance under various sparsities. For the LLama2-13B model, our method preserves **over 95%** MMLU accuracy under a **50% sparsity** setting, with a practical speedup of $2\times$.

Our main contributions are summarized as follows:

- We propose a bilevel optimization framework that enables effective collaboration between learning and matrix approximation for LLM compression.

- We introduce two main technical contributions: a policy gradient method based on truncated Gaussian modeling, and a QR-based RPCA algorithm for efficient matrix approximation.
- Extensive experiments demonstrate that our method consistently outperforms existing model compression methods, even under high prune rates.

2 Related Works

LLM Pruning. LLM pruning methods can be broadly categorized into structured [12, 13, 22, 39] and unstructured [8, 9, 34, 38, 45, 46] approaches. Structured pruning removes entire components (e.g., layers, neurons, channels), with methods like LLM-Pruner [22] using gradient-based importance scores. Unstructured pruning, such as Wanda [31] and SparseGPT [8], prunes individual weights and can remove up to 30% with little accuracy drop. Wanda uses activation-aware scoring, while SparseGPT estimates Hessians for efficient weight reconstruction. However, both approaches face trade-offs between speedup and performance. Hybrid methods that combine sparsity and low-rank decomposition can better balance these aspects by integrating the strengths of both.

Sparsity and Low Rank. Compression methods combining sparsity and low-rank decomposition are increasingly used for LLM compression. LoRAP [17] applies low-rank approximation to Attention matrices and enforces sparsity on MLP blocks, reflecting their distinct structures. LoSparse [20] decomposes weight matrices into low-rank factors $\mathbf{U}, \mathbf{V}$ and a sparse component $\mathbf{S}$, updating all parts while pruning $\mathbf{S}$ to meet a sparsity budget. OATS [44] and HASSLE-free [23] alternate between low-rank approximation and sparsification using fixed sparsity allocations. We focus on this class of RPCA-based compression methods as our base framework to achieve stronger performance.

Sparsity Allocation. Many LLM pruning methods minimize layer-wise reconstruction loss $\|W_l X_l - \tilde{W}_l X_l\|_F^2$ and assume uniform sparsity across layers, often yielding suboptimal results.Several recent methods explore sparsity allocation strategies[26, 40]. FLAP [2] allocates sparsity based on fluctuation scores, OWL [41] leverages activation outliers, DSA [18] searches for optimal allocation functions, and ALS [19] formulates the problem as linear programming. However, these methods depend on fixed validation sets and lack joint optimization with training. In contrast, we formulate sparsity allocation as a learnable optimization problem driven by training data, while treating pruning as a matrix approximation task solvable by existing frameworks.

3 Preliminary

RPCA Framework for Matrix Approximation. As we adopt RPCA [44] as a base solver in our proposed compression method, we present its basics as follows. Given a weight matrix $\mathbf{W} \in \mathbb{R}^{m \times n}$, the target sparsity K and rank r, RPCA approximates $\mathbf{W}$ as the sum of a low-rank matrix $\mathbf{L}$ and a sparse matrix $\mathbf{S}$ by solving the following optimization problem:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{W} - \mathbf{L} - \mathbf{S}\|_F^2 \quad s.t. \quad \text{rank}(\mathbf{L}) \leq r,\ \|\mathbf{S}\|_0 \leq K. \tag{1}$$

Problem 1 is usually solved via alternating optimization, with the following update rules:

$$\begin{cases} \mathbf{L}_{t+1} = \text{TruncatedSVD}(\mathbf{W} - \mathbf{S}_t,\ r), \\ \mathbf{S}_{t+1} = \mathcal{P}_\omega(\mathbf{W} - \mathbf{L}_{t+1}), \end{cases} \tag{2}$$

where $\text{TruncatedSVD}(\mathbf{W} - \mathbf{S}, r)$ denotes the rank-r approximation of matrix $\mathbf{W} - \mathbf{S}$ obtained by retaining only the top-r singular values and their corresponding singular vectors. The operator $\mathcal{P}_\omega(\cdot)$ denotes the projection into the feasible set ω of the sparse matrix $\mathbf{S}$, i.e., $\omega \triangleq \{\mathbf{S} : \|\mathbf{S}\|_0 \leq K, \mathbf{S} \in \mathbb{R}^{m \times n}\}$. Typically, this projection enforces a sparsity constraint by retaining only the top-K largest-magnitude entries and setting the rest to zero. Following Wanda and OATS, we apply a diagonal scaling matrix $\mathbf{D} = \sqrt{\text{diag}(\boldsymbol{X}^\top \boldsymbol{X})} \in \mathbb{R}^{n \times n}$ to the weight matrix and perform approximation on $\mathbf{WD}$, where $\boldsymbol{X}$ denotes the input activation. This is used by default unless otherwise specified.

Discussion. It is important to note that the above solver is not efficient enough due to the expensive SVD process and cannot be directly adapted to develop our compression method. Therefore, in this paper, we introduce an efficient RPCA algorithm based on QR decomposition, detailed in Section 4.3.

4 Method

In this section, we present the details of our method. Section 4.1 introduces the overall design of the bilevel optimization framework. Section 4.2 describes our first technical contribution: we propose a truncated Gaussian prior and integrate it with policy gradient estimator to stabilize the training process and avoid expensive implict differentiation. Section 4.3 presents our second technical contribution: an efficient RPCA algorithm adapted from QR decomposition.

4.1 Bilevel Framework

We introduce our bilevel optimization framework for LLM compression, which formulates an effective collaborative mechanism between learning and matrix approximation. The inner loop performs model compression by solving an RPCA problem under a specific sparsity allocation scheme given by the outer loop. The outer loop formulates the learning problem of sparsity allocation into a probabilistic optimization task, enabling the automatic learning of both layer-wise sparsities and matrix-wise retained ranks based on the model compressed by the inner loop. The workflow is shown in Figure 1.

Inner Loop. We first describe how a given allocation scheme determines the sparsity structure of each matrix in the model. To capture differences in parameter redundancy across layers, we allocate a sparsity ratio s_i to each layer, indicating the proportion of parameters to be pruned. For all matrices $\{\mathbf{W}_{i,j}\}_{j=1}^{M}$ in layer i that lie in $\mathbb{R}^{m\times n}$, the total number of retained parameters after compression is $mn(1-s_i)$. We let $\kappa_{i,j}$ be the proportion of parameters allocated to $\mathbf{W}_{i,j}$ for the low rank matrix $\mathbf{L}_{i,j}$, i.e., we assign $mn(1-s_i)\kappa_{i,j}$ parameters to $\mathbf{L}_{i,j}$. This yields the target rank and sparsity:

$$r_{i,j} = \frac{mn(1-s_i)\kappa_{i,j}}{m+n}, \quad K_{i,j} = mn(1-s_i)(1-\kappa_{i,j}). \tag{3}$$

We group all s_i and $\kappa_{i,j}$ into two vectors $(\boldsymbol{s}, \boldsymbol{\kappa})$ and compress each matrix accordingly by solving a series of RPCA problems described in Section 3. That is, we can obtain a set of RPCA problems presented in the definition below.

Definition 1. *The* RPCA *decomposition of the full parameter set* $\mathcal{W} \triangleq \{\mathbf{W}_{i,j} | i \in [1, L],\ j \in [1, M]\}$ *under sparsity allocation scheme* $(\boldsymbol{s}, \boldsymbol{\kappa})$ *is denoted as*

$$\text{RPCA}(\mathcal{W}, \boldsymbol{s}, \boldsymbol{\kappa}) = \left\{ \tilde{\mathbf{W}}_{i,j} = \underset{\substack{\|\mathbf{S}_{i,j}\|_0 \le K_{i,j} \\ \text{rank}(\mathbf{L}_{i,j}) \le r_{i,j}}}{\arg\min} \|\mathbf{W}_{i,j} - \mathbf{L}_{i,j} - \mathbf{S}_{i,j}\|_F^2 \;\middle|\; \mathbf{W}_{i,j} \in \mathcal{W} \right\},$$

where each matrix $\mathbf{W}_{i,j}$ *is decomposed into a low-rank component* $\mathbf{L}_{i,j}$ *and a sparse component* $\mathbf{S}_{i,j}$ *with target rank* $r_{i,j}$ *and sparsity budget* $K_{i,j}$ *computed from Eqn. (3).*

Outer Loop. In the outer loop, we begin by modeling the allocation scheme $(\boldsymbol{s}, \boldsymbol{\kappa})$ using a suitable probabilistic distribution $p(\cdot|\boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$. A sparsity allocation is sampled from this distribution and passed into the inner loop to generate a compressed model $\mathcal{W}(\boldsymbol{s}, \boldsymbol{\kappa})$. The performance of the resulting model is then evaluated using a loss function. The overall objective is to minimize the expected loss over sampled allocation schemes. To this end, we optimize the probability parameters $\boldsymbol{\theta}$ via gradient-based methods, enabling the framework to adaptively explore and refine sparsity patterns that lead to improved model performance.

Therefore, the overall bilevel optimization framework can be formulated as follows:

$$\begin{aligned} \min_{\boldsymbol{\theta}\in\mathcal{C}} \quad & \mathbb{E}_{(\boldsymbol{s},\boldsymbol{\kappa})\sim p(\cdot|\boldsymbol{\theta})}\, \mathcal{L}(\mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa})) = \frac{1}{N}\sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i, \mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa})), \boldsymbol{y}_i), \\ & s.t. \quad \mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa}) = \text{RPCA}(\mathcal{W}, \boldsymbol{s}, \boldsymbol{\kappa}), \end{aligned} \tag{4}$$

where $\mathcal{C}$ is the feasible region for $\boldsymbol{\theta}$ to control the sparsity, which will be specified in Section 4.2. $\{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{N}$ represents the training dataset, $f(\cdot, \mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa}))$ is the compressed model under allocation scheme $(\boldsymbol{s},\boldsymbol{\kappa})$, and $\ell(\cdot,\cdot)$ denotes the loss function.

Challenges. This bilevel optimization framework presents two main challenges. First, the outer objective is hard to optimize due to the implicit differentiation through the inner loop, requiring appropriate gradient estimators and a well-designed probabilistic model $p(\cdot \mid \boldsymbol{\theta})$. Second, repeatedly solving RPCA problems in the inner loop is computationally expensive. In the following sections, we introduce our proposed techniques to address these challenges.

4.2 Outer Optimization

Policy Gradient. To address the difficulty of computing gradients with respect to the parameter $\boldsymbol{\theta}$, we adopt a policy gradient estimator. This approach avoids implicit differentiation by directly computing gradients based on the loss function. The derivation is as follows:

$$\nabla_{\boldsymbol{\theta}} \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})} \left[\mathcal{L}(\mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa}))\right] = \mathbb{E}_{p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})} \left[\mathcal{L}(\mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa})) \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})\right]. \tag{5}$$

In practice, we sample a mini-batch $\mathcal{B}$, evaluate the loss of the compressed model under each sampled allocation $(\boldsymbol{s},\boldsymbol{\kappa})$, and compute the policy gradient of the parameters $\boldsymbol{\theta}$ as: $\boldsymbol{g}_{\boldsymbol{\theta}} = \mathcal{L}_{\mathcal{B}}(\mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa})) \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})$. This yields an unbiased estimator; the proof is provided in Appendix D.1.

Remark 1. *Policy gradient methods are known to suffer from high variance due to the stochastic nature of the sampling process, which can lead to instability during training. To mitigate this issue, we subtract a control variate $\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\boldsymbol{s}',\boldsymbol{\kappa}')) \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})$, which has zero mean but is highly correlated with the original gradient. Here, $(\boldsymbol{s}',\boldsymbol{\kappa}')$ is an independent sample drawn from the same distribution as $(\boldsymbol{s},\boldsymbol{\kappa})$. This variance-reduction technique leads to the final gradient estimator:*

$$\boldsymbol{g}_{\boldsymbol{\theta}}^{vr} = [\mathcal{L}_{\mathcal{B}}(\mathcal{W}(\boldsymbol{s},\boldsymbol{\kappa})) - \mathcal{L}_{\mathcal{B}}(\mathcal{W}(\boldsymbol{s}',\boldsymbol{\kappa}'))] \cdot \nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta}). \tag{6}$$

Truncated Gaussian Distribution. Computing policy gradients requires $\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\theta})$. Gaussian distributions are often used for convenience, but their support is unbounded, conflicting with the constraints (e.g., sparsity ratio in $[0,1]$). Moreover, shrinking variance for convergence can cause gradient explosion. To address these issues, we employ a truncated Gaussian distribution $\mathcal{N}_{\gamma}(\mu,\sigma^2)$ for probabilistic modeling, which restricts the Gaussian distribution $\mathcal{N}(\mu,\sigma^2)$ to the interval $[\mu-\gamma,\mu+\gamma]$. This truncation limits the sampling range and provides bounded support, thereby stabilizing training and facilitating policy gradient computation. For a random variable $x \sim \mathcal{N}_{\gamma}(\mu,\sigma^2)$, its probability density function (PDF) is given by:

$$p(x;\mu,\sigma^2,\gamma) = \begin{cases} \dfrac{1}{\sigma} \cdot \dfrac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)}, & \text{for } x \in [\mu-\gamma,\ \mu+\gamma], \\ 0, & \text{otherwise,} \end{cases}$$

where ϕ and Φ denote the PDF and CDF of the standard Gaussian distribution $\mathcal{N}(0,1)$, respectively. The detailed sampling method for the truncated Gaussian is described in Appendix C.2.

Remark 2. *For the $p(x;\mu,\sigma^2,\gamma)$, its parameter vector is $[\mu,\sigma^2,\gamma]$. To avoid the gradient explosion during training, we fix the variance σ^2. In addtion, to control the range of x and ensure convergence, we manually reduce γ according to the annealing schedule [50]. More details are provided in Appendix A.2. Therefore, the only trainable parameter of $p(\cdot)$ is μ, i.e., $\boldsymbol{\theta} = \mu$. For simplicity, we do not distinguish between μ and $\boldsymbol{\theta}$ in the remainder of this section.*

Each element in gradient $\nabla_{\boldsymbol{\mu}} \log p(\boldsymbol{s},\boldsymbol{\kappa}|\boldsymbol{\mu},\sigma^2,\gamma)$ can be computed using the lemma below.

Lemma 1. *Let a random variable x follow the truncated Gaussian distribution $x \sim \mathcal{N}_{\gamma}(\mu,\sigma^2)$. The gradient of the log-density with respect to the mean parameter μ is given by:*

$$\nabla_{\mu} \log p(x|\mu,\sigma^2,\gamma) = \nabla_{\mu} \log \left(\frac{1}{\sigma} \cdot \frac{\phi\left(\frac{x-\mu}{\sigma}\right)}{\Phi\left(\frac{\gamma}{\sigma}\right) - \Phi\left(\frac{-\gamma}{\sigma}\right)} \right) = \nabla_{\mu} \log \left(\phi\left(\frac{x-\mu}{\sigma}\right) \right) = \frac{x-\mu}{\sigma^2}. \tag{7}$$

Combining Eqn. (6) and Eqn. (7), we compute the gradient $\boldsymbol{g}_{\boldsymbol{\mu}}^{vr}$ with $\boldsymbol{\mu} = (\boldsymbol{\mu}_{\boldsymbol{s}}, \boldsymbol{\mu}_{\boldsymbol{\kappa}})$. Recall that $\boldsymbol{\mu}$ is the mean of $(\boldsymbol{s},\boldsymbol{\kappa})$, its feasible region can be defined as $\mathcal{C} \triangleq \left\{\boldsymbol{\mu} : \|\boldsymbol{\mu}_{\boldsymbol{s}}\|_1 \geq \rho L,\ \boldsymbol{\mu} \in [0,1]^{L+LM}\right\}$. Since $\mathcal{C}$ can be rewriten as $\mathcal{C} = \left\{\boldsymbol{\mu} : \mathbf{1}^{\top}\boldsymbol{\mu}_{\boldsymbol{s}} \geq \rho L,\ \boldsymbol{\mu} \in [0,1]^{L+LM}\right\}$, which is convex, we can project $\boldsymbol{\mu}$ onto $\mathcal{C}$ using projection $\text{proj}_{\mathcal{C}}(\cdot)$ after gradient descent. See Appendix C.1 for details.

Algorithm 1 Bilevel Optimization Framework

Input: Model weights $\mathcal{W}$, over all prune rate ρ, rank ratio κ_0, parameter σ^2 and γ, learning rate η

1: Initialize parameters $\boldsymbol{\mu} = (\boldsymbol{\mu_s}, \boldsymbol{\mu_\kappa})^3$
2: **for** each iteration **do**
3: Sample a mini batch $\mathcal{B}$
4: Reduce γ according to annealing schedule
5: Sample $(\boldsymbol{s}^{(i)}, \boldsymbol{\kappa}^{(i)})$ from $p(\boldsymbol{s}, \boldsymbol{\kappa}|\boldsymbol{\mu}, \sigma^2, \gamma), i = 1, 2$
6: Apply RPCA, i.e., Algorithm 2, to obtain the compressed weights $\mathcal{W}(\boldsymbol{s}^{(i)}, \boldsymbol{\kappa}^{(i)}), i = 1, 2$
7: Compute $\mathcal{L}_\mathcal{B}(W(\boldsymbol{s}^{(i)}, \boldsymbol{\kappa}^{(i)})), i = 1, 2$, and the gradient:

$$\boldsymbol{g}_{\boldsymbol{\mu}}^{vr} = \Big[\mathcal{L}_\mathcal{B}(\mathcal{W}(\boldsymbol{s}^{(1)}, \boldsymbol{\kappa}^{(1)})) - \mathcal{L}_\mathcal{B}(\mathcal{W}(\boldsymbol{s}^{(2)}, \boldsymbol{\kappa}^{(2)}))\Big] \cdot \nabla_{\boldsymbol{\mu}} \log p(\boldsymbol{s}^{(1)}, \boldsymbol{\kappa}^{(1)}|\boldsymbol{\mu}, \sigma^2, \gamma)$$

8: Update: $\boldsymbol{\mu} \leftarrow \text{proj}_\mathcal{C}(\boldsymbol{\mu} - \eta \boldsymbol{g}_{\boldsymbol{\mu}}^{vr})$
9: **end for**

Output: Compressed weights $\mathcal{W}(\boldsymbol{\mu_s}, \boldsymbol{\mu_\kappa})$

4.3 Inner Optimization

QR-based RPCA algorithm. Conventional RPCA algorithms repeatedly perform costly SVD computations, resulting in high computational overhead. In the inner loop, we follow and adapt the method proposed in [42], replacing SVD with a more efficient QR-based algorithm. Specifically, noting that the low-rank matrix $\mathbf{L}$ in Problem 1 can be factorized as the product of two matrices $\mathbf{U} \in \mathbb{R}^{m\times r}$ and $\mathbf{V} \in \mathbb{R}^{r\times n}$, where r is the target rank, we obtain the following reformulation:

$$\min_{\mathbf{U},\mathbf{V},\mathbf{S}} \|\mathbf{W} - \mathbf{U}\mathbf{V} - \mathbf{S}\|_F^2 \quad s.t. \quad \text{rank}(\mathbf{U}\mathbf{V}) \le r,\ \|\mathbf{S}\|_0 \le K \tag{8}$$

This problem can be solved by alternating minimization over $\mathbf{U}$, $\mathbf{V}$, and $\mathbf{S}$, yielding the update rules:

$$\begin{cases} \mathbf{U}_{t+1} = (\mathbf{W} - \mathbf{S}_t)\mathbf{V}_t^\top (\mathbf{V}_t \mathbf{V}_t^T)^\dagger, \\ \mathbf{V}_{t+1} = (\mathbf{U}_{t+1}^\top \mathbf{U}_{t+1})^\dagger \mathbf{U}_{t+1}^\top (\mathbf{W} - \mathbf{S}_t), \\ \mathbf{S}_{t+1} = \mathcal{P}_\omega(\mathbf{W} - \mathbf{U}_{t+1}\mathbf{V}_{t+1}). \end{cases} \tag{9}$$

The optimization objective only depends on the product $\mathbf{U}\mathbf{V}$, rather than the specific factorization. Therefore, we aim to find any pair $(\mathbf{U}', \mathbf{V}')$ such that $\mathbf{U}'\mathbf{V}' = \mathbf{U}\mathbf{V}$. This insight allows us to reinterpret the optimization as a projection problem.

$$\mathbf{U}_{t+1}\mathbf{V}_{t+1} = \mathbf{U}_{t+1}\left(\mathbf{U}_{t+1}^\top \mathbf{U}_{t+1}\right)^\dagger \mathbf{U}_{t+1}^\top (\mathbf{W} - \mathbf{S}_t) = \Pi_{\mathcal{C}(\mathbf{U}_{t+1})}(\mathbf{W} - \mathbf{S}_t), \tag{10}$$

where $\Pi_{\mathcal{C}(\mathbf{U}_{t+1})}$ denotes the orthogonal projection onto the column space of $\mathbf{U}_{t+1}$.

Since $\left(\mathbf{V}_t \mathbf{V}_t^\top\right)^\dagger$ is full-rank, the column space of $\mathbf{U}_{t+1}$ is equivalent to that of $(\mathbf{W} - \mathbf{S}_t)\mathbf{V}_t^\top$. Thus, we perform a QR decomposition on this matrix:

$$(\mathbf{W} - \mathbf{S}_t)\mathbf{V}_t^\top = \mathbf{Q}_t \mathbf{R}_t, \tag{11}$$

where $\mathbf{Q}_t \in \mathbb{R}^{m\times r}$ is orthonormal and spans the column space of $\mathbf{U}_{t+1}$, leading to the expression:

$$\mathbf{U}_{t+1}\mathbf{V}_{t+1} = \mathbf{Q}_t \mathbf{Q}_t^\top (\mathbf{W} - \mathbf{S}_t), \tag{12}$$

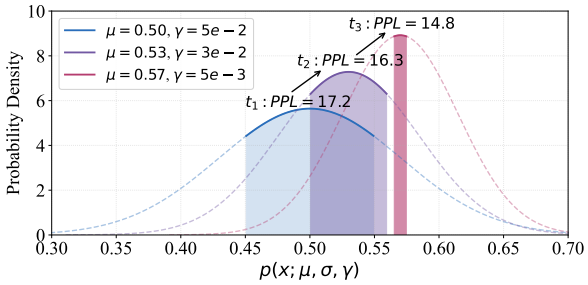
and we accordingly set the update rules as:

$$\mathbf{U}_{t+1} := \mathbf{Q}_t, \quad \mathbf{V}_{t+1} := \mathbf{Q}_t^\top (\mathbf{W} - \mathbf{S}_t). \tag{13}$$

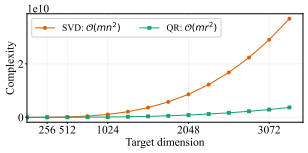
We present the RPCA algorithm based on QR decomposition in Appendix C.3 Algorithm 2.

Remark 3. *The QR-based method reduces the per-iteration complexity to $\mathcal{O}(mr^2)$, much lower than the $\mathcal{O}(mn\min(m,n))$ cost of SVD when $r \ll \min(m,n)$, while maintaining approximation quality.*

Remark 4. *Rather than reinitializing the inner RPCA subroutine from scratch at each iteration, we warm-start the optimization using the previous solution and adjust the low-rank factors $\mathbf{U}, \mathbf{V}$ incrementally according to the rank update Δr, thereby improving efficiency and stability.*



(a) Trajectory of the 25-th layer's sparsity during optimization.



(b) Complexity of QR and SVD.

Decomposition Type	Time Cost (80 iters)
SVD	5 h
QR	4 min

(c) Time cost of QR and SVD.

Figure 2: (a) the variation of the truncated Gaussian distribution as training progresses reflects how the structure gradually learns to approach the optimum. (b) due to $r \ll \min(m, n)$, QR has a lower complexity. (c) QR decomposition provides significant acceleration in practical implementation.

By integrating the truncated Gaussion prior with policy gradient estimator and the QR-based RPCA solver, we obtain the complete bilevel optimization framework presented in Algorithm 1. For the last step, since the range of $(\boldsymbol{s}, \boldsymbol{\kappa})$ is vanished, we compress the model under $(\boldsymbol{\mu_s}, \boldsymbol{\mu_\kappa})$ without sampling. We present the convergence analysis of the bilevel optimization framework in Appendix D.4.

5 Experiments

In Section 5.1, we introduce the overall experimental setup and the baselines used for comparison. Section 5.2 presents the pruning performance across multiple LLMs. In Section 5.3, we explores the effectiveness of our method under high sparsity settings. Finally, Section 5.4 conducts ablation studies to validate the effectiveness of each component in our framework.

5.1 Experimental Setups

Models. For our main experiments, we select representative models from two prominent open-source architectures: the Phi family (specifically Phi-3-mini [1]) and the Llama family (including Llama2-7B, Llama2-13B [33], and Llama3-8B [7]).

Baseline. We select SOTA compression methods as baselines, including unstructured pruning methods such as SparseGPT [8] and Wanda [31], as well as RPCA based approaches like OATS and QR, where QR refers to using only the inner QR-based algorithm in Section 4.3, without the bilevel optimization framework.[44]. Our experiments compare these compression methods at low prune rates ($\leq 50\%$) and validate the effectiveness of sparsity allocation at higher prune rates ($\geq 60\%$).

Configurations. We use C4 [27] as the training dataset, with batch size set to 32, and length set to 256. In addition, the inner-level optimization employs the fixed 32 samples as the calibration dataset. Gamma is set from 0.05 to 0.005. In training, we use the Adam optimizer [16] and set the learning rate to 1e-2. The experiments are all completed with one single 80GB NVIDIA A100.

Evaluation. We use LM-evaluation-harness [10] to evaluate the performance after pruning. The main benchmarks include: 1) WikiText2 [24] perplexity, 2) zero-shot tasks (including PIQA [3], HellaSwag [43], Winogrande [28], OpenBookQA [25], RTE [35], BoolQ [5], ARC-e and ARC-c [6]), and 3) few shot tasks, like MMLU [14]. In addition, we test the CPU inference speedup of the pruned model on Intel(R) Xeon(R) Platinum 8369B CPU @ 2. 90GHz with 32 cpu cores.

5.2 Comparison of Compression Methods

Table 1 presents the main results, comparing the performance of different models using various compression methods across multiple prune rates. Our approach achieves top performance across all three task types. Notably, on the WikiText2 benchmark, it reduces Phi-3-mini's perplexity by about 5% over the SOTA OATS at 50% sparsity. Results on MMLU and zero-shot accuracy further show that RPCA-based methods (OATS, QR, and ours) outperform methods relying solely on sparse matrix

[3]The initialization details of $\boldsymbol{\mu_s}$ and $\boldsymbol{\mu_\kappa}$ are provided in Appendix A.2.

Table 1: Performance comparison across various methods and models with different prune rates. The best performance for each prune rate is in **bold**.

Prune rate	Method	Phi-3-mini			Llama2-7B			Llama2-13B			Llama3-8B		
		↓ WikiText2	↑ MMLU	↑ zero-shot	WikiText2	MMLU	zero-shot	WikiText2	MMLU	zero-shot	WikiText2	MMLU	zero-shot
0%	Dense	9.50	70.34	71.99	8.79	50.12	66.27	7.91	56.41	68.72	10.18	64.97	69.71
30%	SparseGPT	11.19	68.31	70.36	9.29	49.10	64.99	8.29	54.48	68.35	9.71	64.25	69.08
	Wanda	10.71	67.63	70.66	9.23	47.56	65.31	8.29	55.1	68.23	9.71	63.67	68.63
	OATS	10.27	68.84	71.48	9.06	49.98	65.11	8.11	55.97	**68.76**	9.59	65.22	69.34
	QR	10.34	68.35	71.06	9.10	50.02	65.89	8.17	53.99	68.36	**8.00**	63.28	**70.00**
	Ours	**9.98**	**69.60**	**71.51**	**8.83**	**50.10**	**66.19**	**8.02**	**56.13**	68.74	8.06	**65.37**	69.82
40%	SparseGPT	13.03	63.47	69.18	9.94	45.52	64.13	8.85	54.48	68.35	10.01	60.91	67.58
	Wanda	12.59	64.15	68.80	9.86	44.8	64.70	8.77	53.65	68.06	9.74	60.33	67.04
	OATS	11.53	65.75	70.04	9.53	47.21	65.63	8.45	55.25	68.16	9.24	62.46	**68.68**
	QR	11.67	64.28	69.98	9.56	46.64	64.53	8.56	54.96	68.6	8.70	61.94	68.29
	Ours	**11.03**	**65.92**	**70.56**	**9.14**	**47.62**	**65.86**	**8.31**	**55.70**	**68.64**	**8.59**	**62.53**	68.62
50%	SparseGPT	16.80	53.22	66.36	11.66	41.94	62.69	10.21	48.91	66.68	11.95	53.60	64.66
	Wanda	17.23	54.57	65.03	11.43	37.16	62.53	10.05	49.59	66.28	12.36	49.83	63.27
	OATS	15.18	59.99	68.41	10.87	44.7	63.49	9.49	52.44	67.77	10.87	56.46	65.71
	QR	15.30	58.28	67.48	10.86	42.53	63.09	9.58	53.31	67.65	10.70	55.30	65.54
	Ours	**14.87**	**60.57**	**69.37**	**10.49**	**46.10**	**64.08**	**9.23**	**53.79**	**67.92**	**10.18**	**56.97**	**66.28**

compression. Our method further excels by adaptively allocating sparsity. Interestingly, Llama3-8B even surpasses its unpruned version at low prune rates, suggesting that pruning redundant weights may be beneficial. Moreover, the QR-based method with uniform allocation achieves performance close to OATS while requiring only 1/20 of its runtime, highlighting both efficiency and effectiveness, thereby offering a promising direction for scalable LLM compression.

5.3 Comparison with OWL Sparsity Allocation

We further investigate the effectiveness of different sparsity allocation strategies under high prune rates. To ensure a fair comparison, we adopt the QR decomposition introduced in Section 4.3 as the sole compression algorithm. Table 2 compares the results of various sparsity allocation strategies, where uniform denotes applying the same prune rates to all layers, and OWL leverages outlier information to allocate sparsity. It can be observed that our method leads to better performance, providing valuable insights for future research on high sparsity compression.

Table 2: Comparison of sparsity allocation methods under high sparsity.

Prune rate	Method	Phi-3-mini			Llama2-7B			Llama3-8B		
		↓ PPL	↑MMLU	↑zero-shot	PPL	MMLU	zero-shot	PPL	MMLU	zero-shot
60%	Uniform	48.8	38.22	56.01	16.92	32.99	58.61	20.03	34.06	56.62
	OWL	35.37	51.43	58.18	15.38	39.33	59.79	17.39	44.07	59.27
	Ours	**32.46**	**52.27**	**59.24**	**15.07**	**39.78**	**60.52**	**17.03**	**43.91**	**59.34**
70%	Uniform	1375.75	25.5	40.88	122.9	24.53	42.49	111.37	26.8	41.14
	OWL	462.67	28.5	46.33	56.38	26.66	48.05	67.72	26.9	45.81
	Ours	**208.7**	**30.27**	**49.53**	**47.21**	**29.38**	**51.62**	**58.34**	**27.6**	**47.04**

5.4 Ablation Study

We conduct ablation studies on the components of the proposed bilevel optimization framework using Phi-3-mini with 50% sparsity. We first compare the effects of layer-wise sparsity and matrix rank ratio allocation on the performance of the compressed model, where w/o Rank denotes only sparsity allocation without rank ratio allocation, w/o Sparsity denotes only rank ratio allocation without sparsity allocation, and w/o Rank & Sparsity denotes neither allocation being applied. The results, shown in Table 3, demonstrate that rank ratio allocation has a more significant impact on performance, a factor that has been overlooked in other sparsity allocation methods. We also compare the runtime of different RPCA solvers in Table 2c, where QR-based method is significantly faster than SVD. Additionally, we further compare the performance and runtime of the two decomposition methods under varying iteration counts, as shown in Figure 3. At 80 iterations, both methods achieve similar zero-shot accuracy, while the QR-based method completes in just a few minutes.

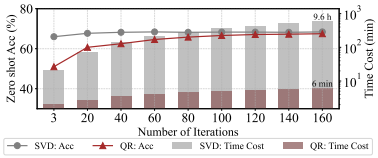
6 Further Analysis

In this section, we discuss the allocation results and actual acceleration effects of various types of model compression, as well as the performance of our bilevel optimization framework when combined with other metric-based compression methods.

Table 3: Ablation study on layer-wise sparsity and matrix rank ratio allocation.

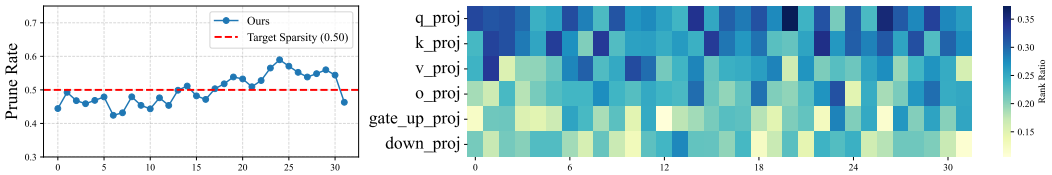
Alloc type	↓ Perplexity	↑ MMLU	↑ zero-shot
Sparsity & Rank	14.87	60.57	69.37
w/o Rank	14.94	59.94	68.48
w/o Sparsity	14.89	60.04	68.93
w/o Rank & Sparsity	15.30	58.28	67.48

Figure 3: Impact of the number of iterations on accuracy and time cost.



6.1 Allocation Visualization

Figure 4 illustrates the allocation results of Phi-3-mini obtained by our bilevel framework. The prune rate gradually increases across layers. This reflects the variation in parameter redundancy across layers. Regarding rank ratios across matrices, our method accurately captures the heterogeneous low-rankness. For matrices in MLP, fewer budgets are allocated to the low-rank component, allowing more flexibility for the sparse component to preserve model fitting capacity. Conversely, for attention blocks, our method allocates more rank budget to fully exploit the underlying structure.



(a) Layer-wise Sparsity Allocation (b) Matrix-wise Rank Ratio Allocation

Figure 4: Left: Layer-wise sparsity allocation $\boldsymbol{s}$; Right: Matrix-wise rank ratio allocation $\boldsymbol{\kappa}$.

6.2 CPU SpeedUp

Following the settings in OATS and OWL, we use the DeepSparse engine to evaluate the actual CPU acceleration achieved by various compression methods on the Llama2-13B model. As shown in Table 4, RPCA-based methods yield greater speedups (up to $1.99\times$ at 50% sparsity) thanks to structured low-rank components, and also outperform purely sparse methods in accuracy.

Table 4: Throughput and speedup comparison among different prune types.

Prune rate	Prune type	Method	↑ Throughput ($\mathcal{B}$/s)	↑ Speedup ($\times$)	↓ Perplexity
0%	Dense	–	2.38	1.00	7.91
40%	Unstructured	Wanda	2.93	1.23	8.77
	Low rank & Sparsity	Ours	**3.92**	**1.64**	**8.31**
50%	Unstructured	Wanda	3.99	1.68	10.05
	Semi-unstructured (2:4)	Wanda	4.38	1.84	16.53
	Low rank & Sparsity	Ours	**4.75**	**1.99**	**9.23**

6.3 Integration with Other Compression Methods

To extend our framework beyond low-rank and sparse decomposition, we apply it to metric-based pruning methods like Wanda and SparseGPT. Table 5 shows results on Phi-3-mini at 50% sparsity. Compared to uniform and OWL-based allocations, our method consistently outperforms both. This demonstrates our approach's potential as a general sparsity allocation mechanism across compression methods, providing new insights for model compression. The performance improvement brought by our method is less pronounced compared to the RPCA-based approach, suggesting that the latter presents a more challenging problem that requires accurate sparsity allocation optimization.

Table 5: Integration with other compression methods.

Base method	Alloc method	↓ Perplexity	↑ MMLU	↑ zero-shot
Wanda	Uniform	17.23	54.57	65.03
	OWL	16.22	55.27	65.97
	Ours	**16.01**	**55.78**	**66.14**
SparseGPT	Uniform	16.80	53.22	66.36
	OWL	17.39	56.35	65.95
	Ours	**16.18**	**56.86**	**66.63**

7 Conclusion

In this work, we propose a bilevel optimization framework that unifies learning and matrix approximation for LLM compression. By formulating sparsity and rank allocation as a probabilistic optimization problem and solving the matrix approximation subtask via RPCA, our method effectively captures weight redundancy structures. We introduce a truncated Gaussian prior for probabilistic parameterization, combined with a policy gradient estimator, which avoids impilicit differentiation through the inner loop and stabilizes training. Additionally, we design a QR-based RPCA solver that significantly accelerates the inner loop computation. Our collaborative mechanism offers a new perspective and practical methodology for advancing efficient and effective model compression.

8 Acknowledgements

This work was supported by the National Nature Science Foundation of China (62472097), Shanghai Municipal Science and Technology Commission (Grant No.24511106102), Fudan Kunpeng&Ascend Center of Cultivation and AI for Science Foundation of Fudan University (FudanX24AI028). The computations in this research were performed on the CFFF platform of Fudan University.

References

[1] Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, Weizhu Chen, Yen-Chun Chen, Yi-Ling Chen, Hao Cheng, Parul Chopra, Xiyang Dai, Matthew Dixon, Ronen Eldan, Victor Fragoso, Jianfeng Gao, Mei Gao, Min Gao, Amit Garg, Allie Del Giorno, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Wenxiang Hu, Jamie Huynh, Dan Iter, Sam Ade Jacobs, Mojan Javaheripi, Xin Jin, Nikos Karampatziakis, Piero Kauffmann, Mahoud Khademi, Dongwoo Kim, Young Jin Kim, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Yunsheng Li, Chen Liang, Lars Liden, Xihui Lin, Zeqi Lin, Ce Liu, Liyuan Liu, Mengchen Liu, Weishung Liu, Xiaodong Liu, Chong Luo, Piyush Madan, Ali Mahmoudzadeh, David Majercak, Matt Mazzola, Caio César Teodoro Mendes, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Liliang Ren, Gustavo de Rosa, Corby Rosset, Sambudha Roy, Olatunji Ruwase, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Yelong Shen, Swadheen Shukla, Xia Song, Masahiro Tanaka, Andrea Tupini, Praneetha Vaddamanu, Chunyu Wang, Guanhua Wang, Lijuan Wang, Shuohang Wang, Xin Wang, Yu Wang, Rachel Ward, Wen Wen, Philipp Witte, Haiping Wu, Xiaoxia Wu, Michael Wyatt, Bin Xiao, Can Xu, Jiahang Xu, Weijian Xu, Jilong Xue, Sonali Yadav, Fan Yang, Jianwei Yang, Yifan Yang, Ziyi Yang, Donghan Yu, Lu Yuan, Chenruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yue Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone, 2024.

[2] Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10865–10873, 2024.

[3] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language, 2019.

[4] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):1–37, 2011.

[5] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019.

[6] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge, 2018.

[7] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[8] Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR, 2023.

[9] Elias Frantar, Carlos Riquelme Ruiz, Neil Houlsby, Dan Alistarh, and Utku Evci. Scaling laws for sparsely-connected foundation models. In *The Twelfth International Conference on Learning Representations*.

[10] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024.

[11] Saeed Ghadimi, Guanghui Lan, and Hongchao Zhang. Mini-batch stochastic approximation methods for nonconvex stochastic composite optimization. *Mathematical Programming*, 155(1):267–305, 2016.

[12] Song Guo, Jiahang Xu, Li Lyna Zhang, and Mao Yang. Compresso: Structured pruning with collaborative prompting learns compact large language models. *arXiv preprint arXiv:2310.05015*, 2023.

[13] Shwai He, Guoheng Sun, Zheyu Shen, and Ang Li. What matters in transformers? not all attention is needed. *arXiv preprint arXiv:2406.15786*, 2024.

[14] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021.

[15] Kaiyi Ji, Junjie Yang, and Yingbin Liang. Bilevel optimization: Convergence analysis and enhanced design. In *International conference on machine learning*, pages 4882–4892. PMLR, 2021.

[16] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR 2015)*, 2015.

[17] Guangyan Li, Yongqiang Tang, and Wensheng Zhang. Lorap: transformer sub-layers deserve differentiated structured compression for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 28657–28672, 2024.

[18] Lujun Li, Peijie Dong, Zhenheng Tang, Xiang Liu, Qiang Wang, Wenhan Luo, Wei Xue, Qifeng Liu, Xiaowen Chu, and Yike Guo. Discovering sparsity allocation for layer-wise pruning of large language models. *Advances in Neural Information Processing Systems*, 37:141292–141317, 2024.

[19] Wei Li, Lujun Li, Mark Lee, and Shengjie Sun. Adaptive layer sparsity for large language models via activation correlation assessment. *Advances in Neural Information Processing Systems*, 37:109350–109380, 2024.

[20] Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Losparse: Structured compression of large language models based on low-rank and sparse approximation. In *International Conference on Machine Learning*, pages 20336–20350. PMLR, 2023.

[21] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[22] Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720, 2023.

[23] Mehdi Makni, Kayhan Behdin, Zheng Xu, Natalia Ponomareva, and Rahul Mazumder. A unified framework for sparse plus low-rank matrix decomposition for llms. In *The Second Conference on Parsimony and Learning (Proceedings Track)*.

[24] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2022.

[25] Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering, 2018.

[26] J. Pablo Muñoz, Jinjie Yuan, and Nilesh Jain. Multipruner: Balanced structure removal in foundation models, 2025.

[27] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

[28] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale, 2019.

[29] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[30] Ankur Sinha, Pekka Malo, and Kalyanmoy Deb. A review on bilevel optimization: From classical to evolutionary approaches and applications. *IEEE transactions on evolutionary computation*, 22(2):276–295, 2017.

[31] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

[32] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12, 1999.

[33] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[34] Tycho FA van der Ouderaa, Markus Nagel, Mart Van Baalen, and Tijmen Blankevoort. The llm surgeon. In *The Twelfth International Conference on Learning Representations*.

[35] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding, 2019.

[36] Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.

[37] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. *arXiv preprint arXiv:2403.07378*, 2024.

[38] Haojun Xia, Zhen Zheng, Yuchao Li, Donglin Zhuang, Zhongzhu Zhou, Xiafei Qiu, Yong Li, Wei Lin, and Shuaiwen Leon Song. Flash-llm: Enabling cost-effective and highly-efficient large generative model inference with unstructured sparsity. *Proceedings of the VLDB Endowment*, 17(2):211–224, 2023.

[39] Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *The Twelfth International Conference on Learning Representations*.

[40] Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. Besa: Pruning large language models with blockwise parameter-efficient sparsity allocation. *arXiv preprint arXiv:2402.16880*, 2024.

[41] Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, et al. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. In *The Forty-First International Conference on Machine Learning*, 2024.

[42] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7370–7379, 2017.

[43] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence?, 2019.

[44] Stephen Zhang and Vardan Papyan. OATS: Outlier-aware pruning through sparse and low rank decomposition. In *The Thirteenth International Conference on Learning Representations*, 2025.

[45] Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

[46] Yuxin Zhang, Lirui Zhao, Mingbao Lin, Sun Yunyun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. Dynamic sparse no training: Training-free fine-tuning for sparse llms. In *The Twelfth International Conference on Learning Representations*.

[47] Xiao Zhou, Renjie Pi, Weizhong Zhang, Yong Lin, Zonghao Chen, and Tong Zhang. Probabilistic bilevel coreset selection. In *International conference on machine learning*, pages 27287–27302. PMLR, 2022.

[48] Xiao Zhou, Weizhong Zhang, Hang Xu, and Tong Zhang. Effective sparsification of neural networks with global sparsity constraint. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3599–3608, 2021.

[49] Zihan Zhou, Xiaodong Li, John Wright, Emmanuel Candes, and Yi Ma. Stable principal component pursuit. In *2010 IEEE international symposium on information theory*, pages 1518–1522. IEEE, 2010.

[50] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. *arXiv preprint arXiv:1710.01878*, 2017.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main contributions of this paper lie in the proposed bilevel optimization framework that jointly integrates learning and matrix approximation, along with a series of technical innovations, which are elaborated in detail in the method section. Furthermore, comprehensive experiments are conducted to validate the effectiveness of our approach and demonstrate its advantages.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitations of our work in Appendix Section E

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: We provide a full derivation of our theoretical results, including assumptions and proofs, in the appendix Section D.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We provide detailed descriptions of all experimental configurations, and we believe the results can be reproduced regardless of whether the code and data are provided.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: Yes, we will make our experimental code available in the supplementary materials. The data used in our experiments is open source.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental configuration has been described throughout the manuscript and supplementary materials.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Standard metrics and statistical summaries are reported consistently.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Details of the computing platform and training resources are provided in Appendix A.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work was carried out in strict accordance with the ethical guidelines set forth by NeurIPS.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discussed the broader impacts in Appendix Section F.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: Our study does not involve any such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We used publicly available datasets and models in accordance with their licenses and terms, and provided proper citations.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: We did not release any new assets in this work.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No part of our experiments involved crowdsourcing or the use of human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No part of our experiments involved crowdsourcing or the use of human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: Large language models (LLMs) serve as the primary subjects of our study. As detailed in Section 5, we provide comprehensive documentation on: (1) the pruning methodology applied to LLMs; (2) evaluation protocols used to assess the pruned models.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.