Toward Adaptive Large Language Models Structured Pruning via Hybrid-grained Weight Importance Assessment

Jun Liu[1,2], Zhenglun Kong[1], Pu Zhao[1], Changdi Yang[1], Xuan Shen[1], Hao Tang[3,2*], Geng Yuan[4], Wei Niu[4], Wenbin Zhang[5], Xue Lin[1], Dong Huang[2*], Yanzhi Wang[1*]

[1]Northeastern University
[2]Carnegie Mellon University
[3]Peking University
[4]University of Georgia
[5]Florida International University

{liu.jun2, kong.zhe, zhao.pu, yang.changd, shen.xu, xue.lin, yanz.wang}@northeastern.edu, donghuang@cmu.edu, haotang@pku.edu.cn, {geng.yuan, wniu}@uga.edu, wenbin.zhang@fiu.edu

Abstract

Structured pruning for large language models (LLMs) has garnered significant academic interest due to its ability to efficiently compress and accelerate LLMs by eliminating redundant weight groups at a coarse-grained granularity. Current structured pruning methods for LLMs typically depend on a singular granularity for assessing weight importance, resulting in notable performance degradation in downstream tasks. Intriguingly, our empirical investigations reveal that utilizing unstructured pruning, which achieves better performance retention by pruning weights at a finer granularity, *i.e.*, individual weights, yields significantly varied sparse LLM structures when juxtaposed to structured pruning. This suggests that evaluating both holistic and individual assessments for weight importance is essential for LLM pruning. Building on this insight, we introduce the Hybrid-grained Weight Importance Assessment (HyWIA), a novel method that merges fine-grained and coarse-grained evaluations of weight importance for the pruning of LLMs. Leveraging an attention mechanism, HyWIA adaptively determines the optimal blend of granularity in weight importance assessments in an end-to-end pruning manner. Extensive experiments on LLaMA-V1/V2, Vicuna, Baichuan, and Bloom across various benchmarks demonstrate the effectiveness of HyWIA in pruning LLMs. For example, HyWIA surpasses the cutting-edge LLM-Pruner by an average margin of 2.82% in accuracy across seven downstream tasks when pruning LLaMA-7B by 50%.

Introduction

Large Language Models (LLMs) have demonstrated unparalleled efficacy in various application domains (Li et al. 2023a; Touvron et al. 2023; Chowdhery et al. 2023). However, deploying LLMs at inference time incurs significant financial and energy costs, mainly due to their large model scale, which requires extensive computational resources and GPU memory (Zhao et al. 2023; Shen et al. 2024). In response, there has been marked increase in interest in compressing LLMs, which upholds the promise of LLMs while substantially reducing their memory requirements and computational costs. Prominent techniques include parameter quantization (Xiao et al. 2023; Shao et al. 2023), network pruning (Frantar and Alistarh 2023; Yuan et al. 2021, 2022; Zhao, Sun et al. 2024), token reduction (Zhan et al. 2024a,b) and low-rank decomposition (Bach and Jordan 2005), *etc*.

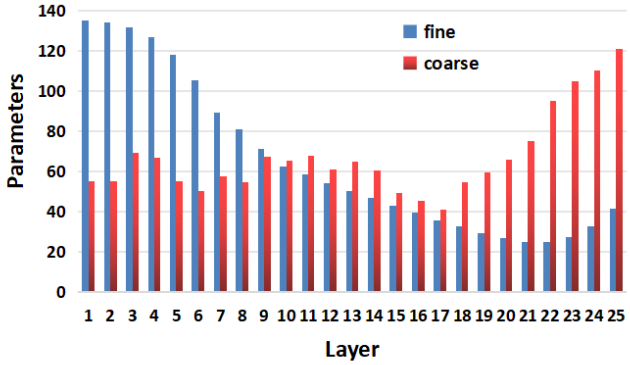

Figure 1: Sparsity allocation across different layers of LLaMA-7B pruned by fine-grained (Xia, Zhong, and Chen 2022) and coarse-grained (Lee et al. 2020) weight importance criteria (50% global pruning rate). Fine-grained pruning tends to preserve more weight in the shallow layers, which is in stark contrast to coarse-grained pruning. The vertical axis represents the parameter quantity of each layer in terms of millions. The horizontal axis represents the layer number of LLaMA-7B.

This paper focuses on pruning LLMs by removing redundant parameters to produce a sparse, lightweight model. Pruning methods vary in granularity, ranging from fine- to coarse-grained approaches. Fine-grained pruning evaluates the importance of individual weights, as seen in SparseGPT (Frantar and Alistarh 2023), which uses the Hessian matrix for layer-wise weight reconstruction, and Wanda (Sun et al. 2024), which combines weight magnitude with input activations to assess significance. While effective in reducing model size with minimal performance loss, fine-grained pruning creates irregular sparsity patterns, complicating deployment on conventional hardware.

In contrast, coarse-grained (structured) pruning elimi-

*Corresponding Authors.

nates entire columns, rows, or blocks of weights, leveraging metrics like gradient information (Ma, Fang et al. 2023) for importance assessment. This approach simplifies deployment and achieves acceleration but often incurs a greater performance drop compared to unstructured pruning, even with fine-tuning (Sun et al. 2024).

Broadly speaking, current LLM structured pruning methods typically rely solely on a single granularity of weight importance assessment. Interestingly, we empirically observed that estimating weight importance across different granularities can produce markedly diverse sparse structures in LLMs. As illustrated in Figure 1, fine-grained estimations prioritize the weights in the initial layers as most critical, thereby preserving a greater number of weights, while the coarse-grained counterparts exhibit the opposite tendency. Delving deeper, fine-grained estimation (Han et al. 2015; Frantar and Alistarh 2023) focuses on sustaining and calculating the contribution of each weight to the network output. In contrast, coarse-grained estimation (Ma, Fang et al. 2023; Zhang et al. 2023) predominantly considers the overall effect along weight groups, which may neglect the extreme values of individual weight that holds significance, *i.e.*, weight outliers (Xiao et al. 2023). Therefore, how to simultaneously perceive and evaluate the importance of individual weights and holistic weight groups remains an unresolved challenge in the field.

To address these bottlenecks, we propose the Hybrid-grained Weight Importance Assessment (HyWIA), which adaptively integrates fine-grained and coarse-grained weight importance estimations. By leveraging the attention mechanism (Vaswani et al. 2017), HyWIA automatically generates hybrid-granularity importance scores. This facilitates dynamic balancing and weighting of importance scores at various granularities, thus allowing for a more robust assessment of importance from both individual and collective weight group perspectives. Comprehensive experiments on pruning a variety of LLMs including LLaMA (Touvron et al. 2023), Vicuna (Chiang, Li et al. 2023), Baichuan (Yang 2023), and Bloom (Le Scao et al. 2022), demonstrate the superiority of HyWIA over many state-of-the-art methods. For example, HyWIA significantly enhances performance compared to LLM-pruner (Ma, Fang et al. 2023) and LoRAPruner (Zhang et al. 2023), further improving accuracy by 2.82% and 2.09% respectively with LLaMA-7B at the 50% pruning rate. The main contribution of this paper can be summarized as:

- We empirically observed that coarse-grained and fine-grained pruning generate markedly different sparsity distributions across LLM layers. This largely indicates that structured pruning methods overlook the importance assessment of individual weights, thereby explaining their performance deficit relative to unstructured pruning.
- We introduce HyWIA, a novel LLM pruning method that adaptively merges fine-grained and coarse-grained metrics to comprehensively assess the importance of weights. To the best of our knowledge, this is the first instance of proposing a hybrid-granularity assessment for weight importance in the community.
- Extensive experiments on pruning representative LLMs demonstrate the superiority of the proposed HyWIA over state-of-the-art methods.

Background and Motivation

Model pruning commonly comprises three steps (LeCun, Denker, and Solla 1989; Molchanov et al. 2016; Li et al. 2023b). Recently, some researchers introduced grouping (Ma, Fang et al. 2023; Sun et al. 2024) as the first step, aiming to group structures within large models. The second step is the importance estimation step, during which redundant weight groups selected for pruning are identified. The third step, LoRA fine-tuning (Kwon et al. 2022; Ma, Fang et al. 2023; Sun et al. 2024), concludes the pruning process, aiming to quickly restore any performance that may have been affected by the removal of parameters.

Problem Formulation

The pruning problem is framed as an optimization problem, where the goal is to find an optimal mask $\mathbf{m}$ under a constraint.

Given a loss function $\mathcal{L}(\mathbf{m})$ that depends on a mask $\mathbf{m}$ (where m determines which parameters are kept or pruned), the second-order Taylor series expansion around an initial mask $\mathbb{1}$ (which typically represents keeping all parameters) is given by (LeCun, Denker, and Solla 1989; Kwon et al. 2022):

$$\mathcal{L}(\mathrm{m}) \approx \mathcal{L}(\mathbb{1}) + \nabla_{\mathrm{m}}\mathcal{L}(\mathbb{1})^{\top}(\mathrm{m}-\mathbb{1}) + \frac{1}{2}(\mathrm{m}-\mathbb{1})^{\top}\mathbf{H}(\mathrm{m}-\mathbb{1}), \tag{1}$$

where:

- $\mathcal{L}(\mathbb{1})$ is the loss at the initial mask.
- $\nabla_{\mathrm{m}}\mathcal{L}(\mathbb{1})$ is the gradient of the loss with respect to the mask.
- $\mathbf{H}$ is the Hessian matrix (second-order derivative) of the loss with respect to the mask.

Assuming the model is near a local minimum where the gradient is small to 0 we ignore the linear term, and as $\mathcal{L}(\mathbb{1})$ is a constant, the optimization objective as follows:

$$\arg\min_{\mathrm{m}} \mathcal{L}(\mathrm{m}) \approx \arg\min_{\mathrm{m}} (\mathbb{1}-\mathrm{m})^{\intercal}\mathrm{H}(\mathbb{1}-\mathrm{m}). \tag{2}$$

Since directly computing the Hessian matrix $\mathbf{H}$ is impractical, it is approximated by the empirical Fisher information matrix (Kwon et al. 2022) $\mathbf{F}$.

Motivation. In LLMs, the decoders situated in the initial layers possess distinctive parameters that wield a vital role in capturing intricate characteristics of the input tokens. Consequently, fine-grained estimation manifests as highly suitable for these layers. Conversely, the decoders occupying the final layers of LLMs prioritize the comprehension of semantics and context. Here, a specific coupled structure assumes a pivotal role in grasping abstract semantics and establishing long-distance dependency relationships. As a result, coarse-grained estimation emerges as particularly fitting for these layers. The current LLM method (Frantar and Alistarh 2023; Ma, Fang et al. 2023; Sun et al. 2024) only

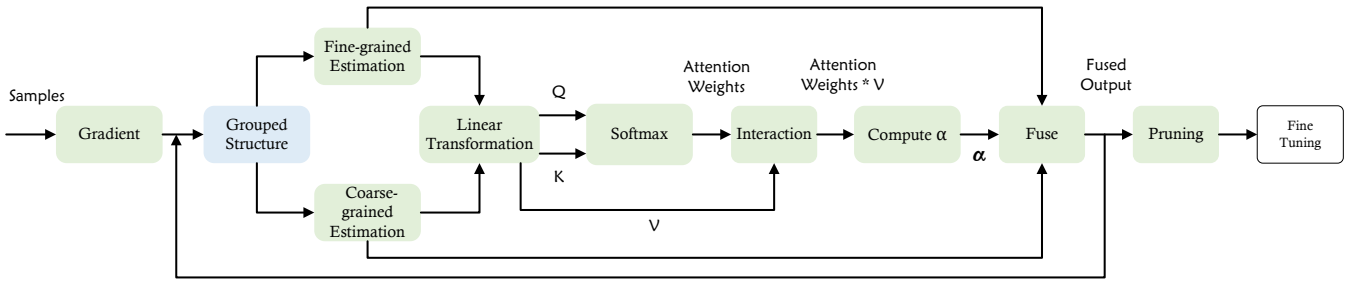

Figure 2: The framework of our proposed Hybrid-grained Weight Importance Assessment (HyWIA) consists of three stages: grouping (blue), adaptive estimation (green), and fine-tuning (white). In the grouping stage, we construct the dependency structure within the LLM. The adaptive estimation stage includes gradient calculation, fine-grained and coarse-grained importance estimation, adaptive fusion, element sorting, and pruning. Finally, the fine-tuning stage uses LoRA (Hu et al. 2022) to recover the pruned model's performance and functionality.

emphasizes general estimation methods such as fine-grained or coarse-grained, resulting in a limited holistic consideration that fails to integrate the strengths and advantages of both approaches. Consequently, challenges arise when estimating the importance of each layer.

We prune LLaMA-7B using fine-grained and coarse-grained estimation methods, each at a 50% pruning rate. Figure 1 shows that fine-grained pruning retains more parameters in the initial layers, aiding intricate information extraction, but fewer in later layers, which hampers global semantic understanding. In contrast, coarse-grained pruning preserves more parameters in later layers. To address this, we propose an adaptive algorithm that dynamically fuses coarse-grained and fine-grained importance estimations for each LLM sub-component, automatically adjusting their proportions during learning.

The Proposed Method

Figure 2 illustrates our proposed Hybrid-grained Weight Importance Assessment (HyWIA) method, which consists of three distinct steps: the weight grouping step (blue), hybrid-grained assessment step (green), and the fine-tuning step (white).

Highlights. Our solution is grounded in the use of Taylor expansion (LeCun, Denker, and Solla 1989; Molchanov et al. 2016) to calculate the fine-grained and coarse-grained gradients derived from the LLM for each input sample. Subsequently, HyWIA takes these fine-grained and coarse-grained gradients as inputs and performs adaptive fusion based on attention mechanism in an efficient training-free manner. In particular, HyWIA utilizes the attention mechanism to dynamically adjust the importance estimation of fine-grained and coarse-grained metric, such that the model can focus on the most relevant input features, thereby deciding the most suitable assessment for the importance of weights. This dynamic adjustment of weights is based on the input fine-grained and coarse-grained gradients. Consequently, our model can automatically adapt its output results under diverse input conditions, effectively accommodating changes in the input data.

Grouping Step

The first step in pruning involves building groups for LLMs. Assuming N_i and N_j are two neurons in the model. The connection between structures can be defined as:

$$\text{Connect}(N_i, N_j) = \begin{cases} w_{ij}, \\ \sum_{p \in \mathcal{P}(N_i, N_j)} \prod_{(u,v) \in p} w_{uv}, \\ 0, \end{cases} \quad (3)$$

- w_{ij} if there is a direct connection from N_i to N_j.
- $\sum_{p \in \mathcal{P}(N_i, N_j)} \prod_{(u,v) \in p} w_{uv}$ where $\mathcal{P}(N_i, N_j)$ is the set of all paths from N_i to N_j.
- 0 if there is no path from N_i to N_j.

This formula calculates the connection between neurons N_i and N_j within the sub-structure, which can be obtained and located through the defined connection relationships. This facilitates the estimation of the importance of each connection structure in LLM in terms of the entirety and the importance of individual elements within the connection structure. Consequently, it aids in the pruning of unimportant connection structures or specific elements within them. The grouping algorithm calculates connection importance based on direct connection, presence of path connection, or no connection.

Hybrid-grained Weight Importance Assessment

Gradient and importance estimation. The impact of each parameter on the loss function is estimated by gradients, utilizing the Taylor expansion approximation of the loss deviation function. Consequently, we utilize this information to estimate the coarse-grained importance and the fine-grained importance.

Coarse-grained formula. At a coarse level, the pruning mask m can be treated as a binary variable where each element indicates whether an entire block, layer, or a group of parameters in the model is kept (1) or pruned (0). The coarse-grained optimization can be represented as:

$$\arg\min_{\mathrm{m}_{\text{coarse}}} \mathcal{L}(\mathrm{m}) \approx \arg\min_{\mathrm{m}_{\text{coarse}}} (\mathbb{1} - \mathrm{m}_{\text{coarse}})^{\intercal} \mathrm{H}_{\text{coarse}} (\mathbb{1} - \mathrm{m}_{\text{coarse}}), \quad (4)$$

Algorithm 1: Attention Fusion Model

Input: fine_grained_grad, coarse_grained_grad
Parameter: d_f (dimension of fine-grained gradients), d_c (dimension of coarse-grained gradients), d_{model} (dimension of model)
Output: Weight importance score

Initialize the linear transformations: W_q, W_k, W_v, and $output\_layer$
Compute $Q = W_q(fine\_grained\_grad)$
Compute $K = W_k(coarse\_grained\_grad)$
Compute $V = W_v(coarse\_grained\_grad)$
Compute attention weights: $attention\_weights = softmax(\frac{Q \cdot K^T}{\sqrt{d_model}})$
Compute interaction output: $interaction\_output = attention\_weights \cdot V$
Compute $\alpha = \text{mean}(attention\_weights, \text{dim} = 1)$ # Compute mean across attention weights
Reshape α to shape $[n, 1]$
Compute fused output: $fused\_output = \alpha \cdot fine\_grained\_grad + (1 - \alpha) \cdot coarse\_grained\_grad$
return $fused\_output$

where $\mathrm{m}_{\text{coarse}}$ represents the mask at a coarse level, such as entire layers or blocks.

Fine-grained formula. At a fine-grained level, the mask m targets individual neurons, weights, or smaller subcomponents of the model. The fine-grained optimization can be represented as:

$$\arg\min_{\mathrm{m}_{\text{fine}}} \mathcal{L}(\mathrm{m}) \approx \arg\min_{\mathrm{m}_{\text{fine}}} (\mathbb{1} - \mathrm{m}_{\text{fine}})^{\intercal} \mathrm{H}_{\text{fine}} (\mathbb{1} - \mathrm{m}_{\text{fine}}), \quad (5)$$

where m_{fine} represents the mask at a finer level, such as individual weights or neurons.

Adaptive fusion. We propose a dynamic fusion method that combines coarse-grained and fine-grained importance estimations via an adaptive learning network. The complexity of LLMs with multi-layer decoders necessitates both holistic and element-wise assessments, making a single estimation approach insufficient.

Our method adaptively fuses the two criteria through a network that leverages sample-specific loss calculations. This fusion balances computational efficiency and model accuracy, expressed as a weighted combination of coarse- and fine-grained objectives:

$$\begin{aligned} &\operatorname{argmin}_{\mathrm{m}_{\text{adaptive}}} \mathcal{L}(\mathrm{m}) \\ \approx &\operatorname{argmin}_{\mathrm{m}} \alpha \cdot (\mathbb{1} - \mathrm{m}_{\text{coarse}})^{\intercal} \mathcal{F}_{\text{coarse}} (\mathbb{1} - \mathrm{m}_{\text{coarse}}) \\ &+ (1 - \alpha) \cdot (\mathbb{1} - \mathrm{m}_{\text{fine}})^{\intercal} \mathcal{F}_{\text{fine}} (\mathbb{1} - \mathrm{m}_{\text{fine}}), \end{aligned} \quad (6)$$

where:

- α is a weighting factor that controls the trade-off between coarse-grained and fine-grained pruning.
- $\mathcal{F}_{\text{coarse}}$ and $\mathcal{F}_{\text{fine}}$ represent the Hessian's approximations Fisher matrix corresponding to the coarse and fine-grained levels, respectively.
- $\mathrm{m}_{\text{coarse}}$ and m_{fine} are the coarse and fine-grained masks, respectively.

Algorithm design for adaptive fusion

To achieve the objective in Eq.(6), we propose the *Attention Fusion Model*, which enables adaptive fusion without traditional parameter training. Algorithm1 outlines its workflow, and the key design principles are detailed as follows.

Dynamic mapping of input features. The algorithm uses three linear transformations, W_q, W_k, and W_v, to map the input *fine_grained_grad* and *coarse_grained_grad* to a unified dimension (*d_model*). Although the parameters of these linear transformations are not updated or trained after model initialization, they still function to map different input features into the same space. Through these mappings, the model can flexibly handle inputs of varying dimensions, thereby adapting to different data characteristics.

Dynamic weight calculation via attention mechanism. The attention mechanism computes the dot product between Q and K (i.e., *attention_scores*) to measure the correlation between different features. Then, these correlations are converted into weights (i.e., *attention_weights*) using the softmax function. These weights are not fixed; they dynamically change according to different inputs. This means that even though the weight parameters in the model are not trained or updated, the weighted output still adapts dynamically based on the input variations. This dynamic weight allocation is the core manifestation of adaptiveness.

Flexible fusion of output features. The final interaction output is obtained by calculating the weighted average of V, followed by a linear layer to map the output to the desired shape. This attention-based mechanism adaptively fuses different input features, enabling the model to adjust effectively to varying inputs.

Adaptive fusion without training. Traditional models adjust parameters through training for specific tasks. In contrast, the Attention Fusion Model leverages input characteristics to achieve adaptive fusion via dynamic weight calculation, independent of training data. By utilizing the gradients (fine_grained_grad and coarse_grained_grad) from LLMs, which inherently carry rich contextual and dynamic features, the model performs adaptive processing through attention mechanisms.

Each sample is inputted into the LLM to generate gradients, which are connected to the second-order Taylor expansion of the loss function around current weights. Fine-grained estimation accumulates gradients over multiple samples, yielding detailed and accurate parameter importance. In contrast, coarse-grained estimation captures first-order Taylor series information by processing multiple samples simultaneously, providing a direct assessment of each parameter's impact on the loss.

Figure. 2 illustrates the framework for algorithm implementation, showcasing the interconnection between various modules, and showing the interconnections among different modules. Element-wise multiplication is an operation in which two matrices or tensors of the same dimensions are multiplied together, element by element.

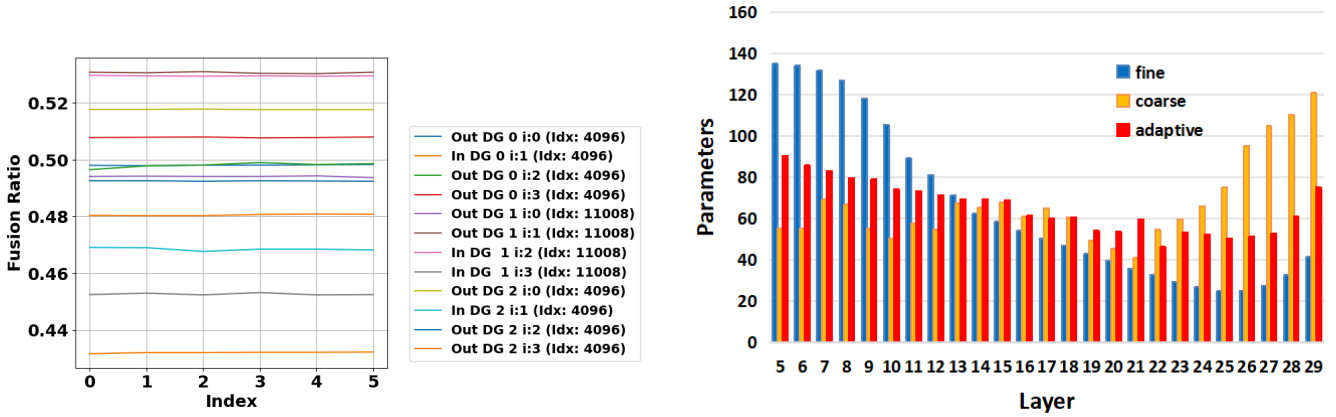

Figure 3: On the left, the adaptive fusion rate is shown, where Out DG 1 i:0 (idx:4096) indicates the output channel (Out), direct connection group 1 (DG 1), the 0th sub-group (i:0), and 4096 parameters (idx:4096). For clarity, only the fusion rates of the first six parameters in the first three groups are displayed. On the right, adaptive pruning is compared with fine-grained and coarse-grained methods.

Pruning. Based on the estimation results, the model parameters are sorted according to their respective importance. Subsequently, pruning is performed by removing the importance of these less significant parameters.

Fine-tuning Step

To accelerate the model recovery process and enhance efficiency with constrained data, the low-rank approximation (LoRA) (Hu et al. 2022) is used to post-train the pruned model. For a pre-trained weight matrix $m_0 \in \mathbb{R}^{r\times k}$. The update of m_0 is constrained by expressing it through a low-rank decomposition $m_0+\Delta m = m_0+\Gamma\beta$, where $\Gamma \in \mathbb{R}^{d\times r}$, $\beta \in \mathbb{R}^{r\times k}$. Throughout training, m_0 remains fixed and does not receive gradient updates, while Γ and β contain trainable parameters. The forward pass is given by:

$$\mathcal{R}(x) = m_0 x + \Delta m x = (m_0 + \Gamma\beta)x. \tag{7}$$

Experiments

Experimental Setup

Our experiment is implemented in PyTorch 2.1.2 (Paszke, Gross et al. 2019), CUDA 11.6 and HuggingFace 4.29.1 (Wolf et al. 2019), LLaMA-7B-V1/V2, 13B (Touvron et al. 2023), Vicuna-7B (Chiang, Li et al. 2023), BLOOM-7b1 (Le Scao et al. 2022), Baichuan-7B (Yang 2023), etc. All pruning experiments are performed on a single NVIDIA A6000 GPU with 48GB of memory.

The model was evaluated on a variety of datasets, including BoolQ (Clark, Lee et al. 2019), PIQA (Bisk, Zellers et al. 2020), HellaSwag (Zellers et al. 2019), WinoGrande (Sakaguchi et al. 2019), ARC-Easy and ARC-Challenge (Clark et al. 2018), as well as WikiText2 (Merity et al. 2016) and PTB (Marcus, Santorini, and Marcinkiewicz 1993), using the EleutherAI LM Harness (Gao et al. 2021)*. We utilize Parameter-Efficient Fine-Tuning (PEFT) (Mangrulkar et al. 2022) with fp16 precision for fine-tuning pruned LLMs generated via the adaptive fusion method. The dataset, sourced from yahma/alpaca-cleaned, is fine-tuned using the Adam optimizer (learning rate: 1×10^{-4}) with LoRA hyperparameters: rank=8, α=16, and a batch size of 64.

*https://github.com/EleutherAI/lm-evaluation-harness

Main Results

We selected LLaMA-7B as a representative case for analysis. In the scenario with a pruning rate of 20% and 50%, Table 1 presents the comparison results between our method and other methods. In terms of average accuracy, our results stand out as the highest among all methods. Our PPL metric for WikiText2 is the lowest among all methods in the 50% pruning rate. We also applied our method to Vicuna-7B, Baichuan-7B, Bloom-7B, and LLaMA-7B-V2 yielded identical conclusions.

Using the adaptive pruning algorithm, each LLaMA-7B parameter is assigned a fusion ratio for fine- and coarse-grained estimation. Figure 3 (left) visualizes the fusion ratios for the first three groups and their initial six sub-group parameters. In the figure, "out" and "in" denote Linear output and input channels, "DG" represents the connection group, "i" is the i-th sub-group, and "idx" the parameter count. The fusion ratios within the same channel show minimal differences, while across different dependency groups, they range from 0.4 to 0.6, indicating varying group importance during estimation.

After determining the ratio, our algorithm dynamically integrates coarse- and fine-grained estimations to generate a comprehensive pruning metric. The right side of the figure illustrates the parameter distribution in layers 5-29 of LLAMA-7B, comparing our adaptive method with fine- and coarse-grained approaches. Results indicate that adaptive pruning balances the significance of both early and later layers, achieving more uniform pruning and optimal outcomes.

Ablation Study

We conducted ablation experiments to analyze the impact of varying sample sizes and pruning rates, systematically assessing performance and the robustness of our approach.

Sample numbers. We experiment with sample numbers

Ratio	Method	WikiT2↓	PTB↓	BoolQ	PIQA	HellaS	WinoG	ARC-e	ARC-c	OBQA	Ave↑
0%	LLaMA-7B (Touvron et al. 2023)	-	-	76.5	79.8	76.1	70.1	72.8	47.6	57.2	68.59
	LLaMA-7B* (Ma, Fang et al. 2023)	12.62	22.14	73.18	78.35	72.99	67.01	67.45	41.38	42.40	63.25
20%	Magnitude (Zhang et al. 2023)	21.78	38.64	61.89	70.81	58.34	56.87	54.87	34.02	38.40	53.59
	SparseGPT * (Dettmers et al. 2023b)	-	-	71.13	75.24	51.58	67.56	68.98	36.09	30.80	57.34
	WANDA* (Sun et al. 2024)	18.43	33.16	65.75	74.70	64.52	59.35	60.65	36.26	39.40	57.23
	Element^2 (Ma, Fang et al. 2023)	45.70	69.33	61.47	68.82	47.56	55.09	46.46	28.24	35.20	48.98
	LoRAPrune (Zhang et al. 2023)	16.80	**28.75**	65.62	79.31	70.00	62.76	65.87	37.69	39.14	60.05
	Compresso (Guo et al. 2023)	-	-	79.08	75.46	53.44	67.8	68.64	37.97	34.20	59.51
	FLAP (An et al. 2024)	**14.62**	-	69.63	76.82	**71.20**	**68.35**	69.91	39.25	39.40	62.08
	SLEB (Song, Oh et al. 2024)	18.50	31.60	65.00	75.00	65.70	57.90	67.06	36.60	35.80	57.60
	Ours	16.42	31.16	**68.53**	**77.8**	70.58	67.49	**70.24**	**40.44**	**42.00**	**62.44**
50%	Magnitude (Zhang et al. 2023)	78.80	164.32	47.40	54.36	33.49	53.10	37.88	26.60	30.12	40.42
	SparseGPT * (Dettmers et al. 2023b)	-	-	64.52	69.9	43.29	64.95	61.86	30.37	23.80	51.24
	WANDA* (Sun et al. 2024)	43.89	85.87	50.90	57.38	38.12	55.98	42.68	34.20	38.78	45.43
	Element^2 (Ma, Fang et al. 2023)	45.70	69.33	61.47	68.82	47.56	55.09	46.46	28.24	35.20	48.98
	Vector (Ma, Fang et al. 2023)	43.47	68.51	62.11	64.96	40.52	51.54	46.38	28.33	32.40	46.61
	LoRAPrune-8bit (Zhang et al. 2023)	33.68	53.24	61.43	70.88	47.65	55.12	**45.78**	30.50	35.62	49.56
	LoRAPrune (Zhang et al. 2023)	30.12	50.30	61.88	71.53	47.86	55.01	45.13	31.62	34.98	49.71
	FLAP (An et al. 2024)	31.80	-	60.21	67.52	52.14	**57.54**	49.66	29.95	35.60	50.37
	Ours	**29.35**	**44.38**	**60.55**	**72.36**	**55.25**	55.09	**50.84**	**31.48**	**37.00**	**51.80**

Table 1: Zero-shot performance of the compressed LLaMA-7B models. The average is calculated among seven classification datasets. **Bold** denotes the best performance. * denotes the results obtained by reproduction.

Ratio	Method	WikiT2↓	PTB↓	BoolQ	PIQA	HellaS	WinoG	ARC-e	ARC-c	OBQA	Ave↑
0%	LLaMA-13B (Touvron et al. 2023)	-	-	78.1	80.1	79.2	73.0	74.8	52.7	56.4	70.61
	LLaMA-13B (Ma, Fang et al. 2023)	11.58	20.24	68.47	78.89	76.24	70.09	74.58	44.54	42.00	64.97
20%	L2 (Ma, Fang et al. 2023)	20.97	38.05	73.25	76.77	71.86	64.64	67.59	39.93	40.80	62.12
	Block (Ma, Fang et al. 2023)	15.18	28.08	70.31	77.91	75.16	67.88	71.09	42.41	43.40	64.02
	FLAP (An et al. 2024)	13.66	-	72.12	77.59	**76.01**	**69.24**	72.59	42.56	43.53	64.52
	Ours	**13.53**	**27.55**	**72.24**	**78.89**	75.63	67.56	**73.49**	**44.11**	42.40	**64.90**

Table 2: Zero-shot performance of the compressed LLaMA-13B at 20% pruning rate.

10, 20, 30, 40, and 50 to provide input to the model and compare the impact on accuracy. We investigate whether the sample numbers affect various aspects of training and model performance. From Table 3 , the first row of each section represents the experimental results for LLM-Pruner Element^2 (Ma, Fang et al. 2023), while the second row displays our experimental results. It is evident that the average accuracy exhibits an increasing trend with the number of example prompts. Concurrently, the perplexity (PPL) of WikiText2 and PTB decreases with increasing sample number. Our model consistently demonstrates higher accuracy compared to LLM-Pruner methods.

Pruning ratio. The choice of pruning rate directly affects the pruning effect and performance of the model. In our experiments, we tried pruning rates of 5%, 10%, 20%, and 50% to compare the accuracy of the models, studying whether different pruning rates impact the model performance. In Table 4 , results are categorized into four sections based on pruning rates of 5%, 10%, 15%, and 20%. The first row in each section shows the experimental outcomes for LLM-Pruner Element^2, while the second row displays our results. Overall, our experimental results outperform the LLM-Pruner method.

Adaptive Fusion Estimation. In Table 5, we maintained a fixed fusion rate of 0.5 throughout the experiment. Compared to the adaptive fusion method, which dynamically adjusts fusion rates, the fixed method showed an average accuracy about 1.4% lower. This demonstrates the superiority of adaptive fusion in enhancing performance.

Related Work

Pruning for LLMs. Various pruning techniques (Li, Zhao et al. 2022; Yang et al. 2023; Wu et al. 2022; Kong et al. 2022; Zhang et al. 2022; Shen et al. 2024) have been developed to reduce the model size and inference cost. *PtPF* (Kwon et al. 2022) proposes a fast post-training pruning framework for Transformers, eliminating the need for retraining *FGlP* (Lee et al. 2020) employs group-level pruning to accelerate deep neural networks. *CoFi* (Xia, Zhong, and Chen 2022) prunes both coarse-grained and fine-grained modules by using masks of varying granularity to control the pruning of each parameter. *LoRAPrune* (Zhang et al. 2023) designed a LoRA-guided pruning criterion, which uses the weights and gradients of LoRA. *FLAP* (An et al. 2024) developed structured importance indicators, and the adaptive search globally compresses the model. *COMPRESSO* (Guo et al. 2023) use a collaborative prompt to enhance the synergy between the LLM and the pruning algorithm. *PAP* (Zhang, Bai et al. 2024) use a pruning metric to combine weight and activation information in LLM.

Number	WikiText2↓	PTB↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average↑
10	17.30	30.74	65.14	76.01	67.89	61.4	51.43	38.23	40.6	57.24
	17.38	31.16	67.83	77.15	69.81	65.04	64.44	38.74	41.4	60.63
20	17.28	31.41	63.39	76.28	68.84	66.54	51.98	37.54	41.2	57.96
	17.89	33.83	69.14	77.64	69.70	63.46	64.44	40.10	40.80	60.75
30	17.25	31.41	63.49	76.12	69.04	66.14	52.36	37.80	41.20	58.02
	17.22	30.93	67.55	77.08	70.15	65.02	66.41	40.27	41.60	61.15
40	17.17	30.68	67.13	77.80	70.02	62.27	54.55	40.27	40.80	58.97
	17.15	30.66	68.53	77.53	70.30	64.96	68.86	40.10	41.80	61.73
50	17.16	30.11	64.62	77.20	68.80	63.14	64.31	36.77	39.80	59.23
	16.42	31.06	68.53	77.8	70.58	67.49	70.24	40.44	42.00	62.44

Table 3: Sample numbers for LLaMA-7B at 20% pruning rate. The first row in each section shows results for LLM-Pruner Element^2 (Ma, Fang et al. 2023).

Ratio	WikiText2↓	PTB↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average↑
5%	13.01	23.02	70.98	77.78	72.53	66.61	69.48	42.06	42.60	63.14
	12.91	22.96	70.60	77.35	72.54	67.01	70.54	42.15	42.20	63.20
10%	14.02	24.99	70.76	77.62	71.87	66.14	69.73	42.15	41.80	62.86
	14.02	24.99	70.54	78.02	72.12	66.43	70.45	42.52	42.20	63.18
20%	17.16	30.11	64.62	77.20	68.80	63.14	64.31	36.77	39.80	59.23
	16.42	31.16	68.53	77.80	70.58	67.49	70.24	40.44	42.00	62.44
50%	45.70	69.33	61.47	68.82	47.56	55.09	46.46	28.24	35.20	48.98
	29.35	44.38	60.55	72.36	55.25	55.09	50.84	31.48	37.00	51.80

Table 4: Prune ratio for LLaMA-7B with 50 samples. The first row in each section shows results for LLM-Pruner Element^2 (Ma, Fang et al. 2023).

Ratio	Method	WikiText2↓	PTB↓	BoolQ	PIQA	HellaSwag	WinoGrande	ARC-e	ARC-c	OBQA	Average↑
50%	No Adaptive	30.13	45.10	60.42	69.48	53.76	53.04	50.93	29.27	36.10	50.42
	Adaptive	**29.35**	**44.38**	**60.55**	**72.36**	**55.25**	**55.09**	**50.84**	**31.48**	**37.00**	**51.80**

Table 5: Adaptive estimation for LLaMA-7B with a pruning rate=50%.

SLEB (Song, Oh et al. 2024) leverages calibration data to estimate the redundancy of transformer blocks and uses a metric to identify and remove it. LLaMA shortened (Kim, Kim et al. 2024) by depth pruning that excludes several transformer blocks, producing speedups. *CompactPKD* (Muralidharan et al. 2024) integrates depth, width, attention, and MLP pruning with knowledge-distillation retraining. *Bonsai* (Dery et al. 2024) introduces a gradient-free perturbative pruning method, yielding compact and efficient models.

Efficient learning for LLMs. The goal of efficient learning (Zhan et al. 2021; Zhan, Wu et al. 2024; Liu et al. 2024a,b, 2021, 2022, 2023a,b; Li, Kong et al. 2020) is to achieve better results with fewer resources. *SpQR* (Dettmers et al. 2023b) employed a method involving the identification and isolation of outlier weights. *LLM-FP4* (yang Liu, Liu et al. 2023) suggests FP4 as a post-training method to quantify weights and activations in large language models (LLM) up to floating point values of 4 bits. *QLORA* (Dettmers et al. 2023a) introduces methods to save memory, which is information-theoretically optimal for normally distributed weights. *Less* (Liang et al. 2023) proposes Task-aware layer-wise distillation (TED) as a solution to reducing the knowledge gap between teacher and student models. *MiniLLM* (Gu et al. 2023) put forth a knowledge distillation approach aimed at condensing LLMs into more compact language models. *LoRD* (Kaushal, Vaidhya, and Rish 2023) utilizes Low Rank Decomposition (LoRD) to ensure that the compressed model remains compatible with the cutting-edge near-lossless quantization method. *LoRAShear* (Chen et al. 2023) initially constructs dependency graphs for LoRA modules to identify minimal removal structures and analyze knowledge distribution. *AdaPTwin* (Biju et al. 2024) compresses pairs of weight matrices that are dependent on products within the transformer attention layer simultaneously.

Conclusion

In this paper, we observe that coarse-grained and fine-grained pruning generate different sparsity distributions across LLM layers. We suggest that evaluating both holistic and individual assessments of weight importance is essential for LLM pruning. We introduce Hybrid-grained Weight Importance Assessment (HyWIA), a novel method that merges fine-grained and coarse-grained evaluations of weight importance for pruning LLMs. Leveraging an attention mechanism, HyWIA adaptively determines the optimal blend of granularity in weight importance assessments in an end-to-end pruning manner. Experiments on LLaMA-V1/V2, Vicuna, Baichuan, and Bloom across various benchmarks demonstrate HyWIA's effectiveness in pruning LLMs.

References

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.

Bach, F. R.; and Jordan, M. I. 2005. Predictive low-rank decomposition for kernel methods. In *International Conference on Machine Learning, ICML*, 33–40.

Biju, E.; et al. 2024. AdaPTwin: Low-Cost Adaptive Compression of Product Twins in Transformers. *arXiv preprint arXiv:2406.08904*.

Bisk, Y.; Zellers, R.; et al. 2020. PIQA: Reasoning about Physical Commonsense in Natural Language. In *AAAI Conference on Artificial Intelligence*, volume 34, 7432–7439.

Chen, T.; Ding, T.; Yadav, B.; Zharkov, I.; and Liang, L. 2023. Lorashear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356*.

Chiang, W.-L.; Li, Z.; et al. 2023. Vicuna: An Open-Source Chatbot Impressing GPT-4 with 90% ChatGPT Quality. https://lmsys.org/blog/2023-03-30-vicuna.

Chowdhery, A.; Narang, S.; Devlin, J.; et al. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240): 1–113.

Clark, C.; Lee, K.; et al. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. *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:1803.05457v1*.

Dery, L.; Kolawole, S.; Kagey, J.-F.; Smith, V.; Neubig, G.; and Talwalkar, A. 2024. Everybody prune now: Structured pruning of llms with only forward passes. *arXiv preprint arXiv:2402.05406*.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023a. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Dettmers, T.; Svirschevski, R.; Egiazarian, V.; et al. 2023b. SpQR: A Sparse-Quantized Representation for Near-Lossless LLM Weight Compression. In *International Conference on Learning Representations*.

Frantar, E.; and Alistarh, D. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML*, 10323–10337.

Gao, L.; Tow, J.; Biderman, S.; Black, S.; DiPofi, A.; Foster, C.; Golding, L.; Hsu, J.; McDonell, K.; Muennighoff, N.; et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.

Gu, Y.; Dong, L.; Wei, F.; and Huang, M. 2023. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*.

Guo, S.; Xu, J.; Zhang, L. L.; and Yang, M. 2023. Compresso: Structured pruning with collaborative prompting learns compact large language models. *arXiv preprint arXiv:2310.05015*.

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. *Advances in Neural Information Processing Systems*.

Hu, E. J.; yelong shen; Wallis, P.; et al. 2022. LoRA: Low-Rank Adaptation of Large Language Models. In *International Conference on Learning Representations*.

Kaushal, A.; Vaidhya, T.; and Rish, I. 2023. Lord: Low rank decomposition of monolingual code llms for one-shot compression. *arXiv preprint arXiv:2309.14021*.

Kim, B.-K.; Kim, G.; et al. 2024. Shortened LLaMA: A Simple Depth Pruning for Large Language Models. *International Conference on Learning Representations. Workshop*.

Kong, Z.; Dong, P.; Ma, X.; et al. 2022. Spvit: Enabling faster vision transformers via latency-aware soft token pruning. In *European Conference on Computer Vision*, 620–640.

Kwon, W.; Kim, S.; Mahoney, M. W.; Hassoun, J.; et al. 2022. A fast post-training pruning framework for transformers. *Advances in Neural Information Processing Systems*.

Le Scao, T.; Fan, A.; Akiki, C.; Pavlick, E.; et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

LeCun, Y.; Denker, J.; and Solla, S. 1989. Optimal Brain Damage. In *Advances in Neural Information Processing Systems*, volume 2.

Lee, K.; Kim, H.; Lee, H.; and Shin, D. 2020. Flexible group-level pruning of deep neural networks for on-device machine learning. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 79–84.

Li, B.; Kong, Z.; et al. 2020. Efficient transformer-based large scale language representations using hardware-friendly block structured pruning. *Findings of the Association for Computational Linguistics: EMNLP 2020*.

Li, Y.; Bubeck, S.; Eldan, R.; Del Giorno, A.; Gunasekar, S.; and Lee, Y. T. 2023a. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

Li, Y.; Yang, C.; Zhao, P.; Yuan, G.; et al. 2023b. Towards real-time segmentation on the edge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37.

Li, Y.; Zhao, P.; et al. 2022. Pruning-as-search: Efficient neural architecture search via channel pruning and structural reparameterization. *International Joint Conference on Artificial Intelligence (IJCAI-22)*.

Liang, C.; et al. 2023. Less is more: Task-aware layer-wise distillation for language model compression. In *International Conference on Machine Learning*, 20852–20867.

Liu, J.; Deng, F.; Yuan, G.; Yang, C.; et al. 2022. An Efficient CNN for Radiogenomic Classification of Low-Grade Gliomas on MRI in a Small Dataset. *Wireless Communications and Mobile Computing*, 2022(1).

Liu, J.; Deng, F.; Yuan, G.; et al. 2021. An Explainable Convolutional Neural Networks for Automatic Segmentation of the Left Ventricle in Cardiac MRI. In *CECNet*, 306–314.

Liu, J.; Kong, Z.; Zhao, P.; et al. 2024a. TSLA: A Task-Specific Learning Adaptation for Semantic Segmentation on Autonomous Vehicles Platform. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*.

Liu, J.; Wu, C.; Yuan, G.; Niu, W.; et al. 2023a. A Scalable Real-time Semantic Segmentation Network for Autonomous Driving. In *Advanced Multimedia Computing for Smart Manufacturing and Engineering (AMC-SME)*, 3–12.

Liu, J.; Yuan, G.; Yang, C.; Song, H.; and Luo, L. 2023b. An Interpretable CNN for the Segmentation of the Left Ventricle in Cardiac MRI by Real-Time Visualization. *CMES-Computer Modeling in Engineering & Sciences*, 135(2).

Liu, J.; Yuan, G.; Zeng, W.; Tang, H.; Zhang, W.; et al. 2024b. Brain Tumor Classification on MRI in Light of Molecular Markers. *arXiv preprint arXiv:2409.19583*.

Ma, X.; Fang, G.; et al. 2023. LLM-Pruner: On the Structural Pruning of Large Language Models. In *Advances in Neural Information Processing Systems*, 21702–21720.

Mangrulkar, S.; Gugger, S.; Debut, L.; et al. 2022. PEFT: State-of-the-art Parameter-Efficient Fine-Tuning methods. https://github.com/huggingface/peft.

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

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer Sentinel Mixture Models. arXiv:1609.07843.

Molchanov, P.; Tyree, S.; Karras, T.; Aila, T.; and Kautz, J. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*.

Muralidharan, S.; Sreenivas, S. T.; Joshi, R.; et al. 2024. Compact Language Models via Pruning and Knowledge Distillation. *arXiv preprint arXiv:2407.14679*.

Paszke, A.; Gross, S.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2019. WinoGrande: An Adversarial Winograd Schema Challenge at Scale. arXiv:1907.10641.

Shao, W.; Chen, M.; Zhang, Z.; et al. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*.

Shen, X.; Zhao, P.; Gong, Y.; Kong, Z.; et al. 2024. Search for Efficient Large Language Models. In *Advances in Neural Information Processing Systems*.

Song, J.; Oh, K.; et al. 2024. SLEB: Streamlining LLMs through Redundancy Verification and Elimination of Transformer Blocks. *arXiv preprint arXiv:2402.09025*.

Sun, M.; Liu, Z.; Bair, A.; et al. 2024. A Simple and Effective Pruning Approach for Large Language Models. In *International Conference on Learning Representations*.

Touvron, H.; Lavril, T.; Izacard, G.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; et al. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wolf, T.; Debut, L.; Sanh, V.; Chaumond, J.; et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Wu, Y.; Gong, Y.; Zhao, P.; Li, Y.; Zhan, Z.; Niu, W.; Tang, H.; et al. 2022. Compiler-aware neural architecture search for on-mobile real-time super-resolution. In *European Conference on Computer Vision*, 92–111.

Xia, M.; Zhong, Z.; and Chen, D. 2022. Structured Pruning Learns Compact and Accurate Models. In *Association for Computational Linguistics (ACL)*.

Xiao, G.; Lin, J.; Seznec, M.; et al. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML*, 38087–38099.

Yang, C.; Zhao, P.; Li, Y.; et al. 2023. Pruning parameterization with bi-level optimization for efficient semantic segmentation on the edge. In *CVPR*, 15402–15412.

Yang, e. a., Aiyuan. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

yang Liu, S.; Liu, Z.; et al. 2023. LLM-FP4: 4-Bit Floating-Point Quantized Transformers. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yuan, G.; Dong, P.; Sun, M.; et al. 2022. Mobile or FPGA? A Comprehensive Evaluation on Energy Efficiency and a Unified Optimization Framework. *ACM Transactions on Embedded Computing Systems*, 21(5): 1–22.

Yuan, G.; et al. 2021. Work in progress: Mobile or FPGA? A comprehensive evaluation on energy efficiency and a unified optimization framework. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 493–496.

Zellers, R.; Holtzman, A.; Bisk, Y.; et al. 2019. HellaSwag: Can a Machine Really Finish Your Sentence? In *Association for Computational Linguistics (ACL)*, 4791–4800.

Zhan, Z.; Gong, Y.; Zhao, P.; et al. 2021. Achieving on-mobile real-time super-resolution with neural architecture and pruning search. In *ICCV*, 4821–4831.

Zhan, Z.; Kong, Z.; Gong, Y.; et al. 2024a. Exploring Token Pruning in Vision State Space Models. In *The Conference on Neural Information Processing Systems*.

Zhan, Z.; Wu, Y.; Kong, Z.; et al. 2024b. Rethinking Token Reduction for State Space Models. In *the 2024 Conference on Empirical Methods in Natural Language Processin.*

Zhan, Z.; Wu, Y.; et al. 2024. Fast and Memory-Efficient Video Diffusion Using Streamlined Inference. In *Conference on Neural Information Processing Systems*.

Zhang, M.; Chen, H.; Shen, C.; et al. 2023. Loraprune: Pruning meets low-rank parameter-efficient fine-tuning. *arXiv preprint arXiv:2305.18403*.

Zhang, Y.; Bai, H.; et al. 2024. Plug-and-Play: An Efficient Post-training Pruning Method for Large Language Models. In *International Conference on Learning Representations*.

Zhang, Y.; Yao, Y.; Ram, P.; et al. 2022. Advancing model pruning via bi-level optimization. *NeurIPS*, 35: 18309–18326.

Zhao, P.; Sun, F.; et al. 2024. Pruning Foundation Models for High Accuracy without Retraining. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Zhao, W. X.; Zhou, K.; Li, J.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.