OCP: Outlier-Centric Probing for Dynamic Structured Pruning of LLMs

Yang Ji[1] **and Ying Sun**[2*]
[1]AI Thrust, Hong Kong University of Science and Technology (Guangzhou)
[2]The 63rd Research Institute, National University of Defense Technology
yji655@connect.hkust-gz.edu.cn, sunying@nudt.edu.cn

Abstract

Structured pruning offers a hardware-friendly approach for efficient LLM inference. Early static methods determine fixed subnetworks through offline calibration, suffering from performance degradation and calibration sensitivity. Recent methods explore input-adaptive pruning by selecting a subset of tokens as probes to estimate hidden activations for online pruning decisions. However, existing probe selection strategies fail to identify outlier-triggering tokens, and uniform layer-wise sparsity misaligns with heterogeneous outlier distributions, leading to critical channels being incorrectly pruned. Therefore, we propose OCP (Outlier-Centric Probing for structured pruning), a principled framework that prioritizes capturing outlier-triggering tokens rather than reconstructing full hidden distributions. Specifically, OCP includes three key components: (1) sensitivity-weighted probing for FFN layers that identifies outlier patterns via precomputed weight aggregations, (2) attention-accumulated probing that leverages preceding attention matrices to identify salient tokens, and (3) online adaptive sparsity allocation that dynamically adjusts layer-wise pruning based on history-guided outlier statistics. Extensive experiments on LLaMA2, LLaMA3, and OPT demonstrate that OCP consistently outperforms state-of-the-art methods across benchmarks, achieving up to 25% perplexity reduction at 1.6× speedup.

1 Introduction

Large Language Models (LLMs) have achieved remarkable success across diverse tasks (Yang et al., 2025a; Zhang et al., 2026b; Li et al., 2026; Xin et al., 2025). However, their substantial resource requirements challenge practical deployment, particularly in latency-sensitive applications (Wan et al., 2024). Structured pruning, which removes entire architectural units such as attention heads and FFN neurons, has emerged as a promising solution, offering hardware-friendly acceleration without specialized kernels (Zhu et al., 2024; Wan et al., 2024; He and Lin, 2025; Wei et al., 2024).

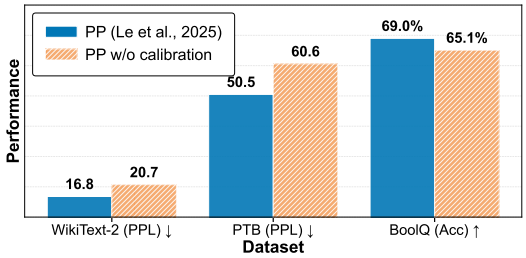


Figure 1: Effect of calibration on PP performance.

Existing structured pruning methods (Ma et al., 2023; An et al., 2024) typically determine fixed subnetworks via offline calibration, yet performance is sensitive to calibration data choice, causing inconsistent generalization (Ji et al., 2025b; Williams and Aletras, 2024; Bandari et al., 2024). To address these limitations, recent studies (Le et al., 2025; Hou et al., 2025) have shifted toward dynamic pruning with input-adaptive decisions, achieving notable improvements over static methods. As illustrated in Figure 2 (left), Probe Pruning (PP) (Le et al., 2025) ① selects a subset of input tokens as a probe, ② collects hidden activations through a forward pass, ③ determines masks based on hidden activation magnitude, and ④ prunes model for acceleration. Despite being designed for input-adaptive pruning, PP still relies on offline calibration for probing state fusion, and as shown in Figure 1, removing it causes notable degradation. This stems from two key limitations: (1) PP selects tokens by input energy (e.g., L^2 norm), yet layer normalization breaks the correlation between input norm and hidden activation magnitude. Thus, outlier-triggering tokens are overlooked while selected tokens may miss the current input's highly-activated channels (Figure 2 right). (2) PP applies uniform sparsity across

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6108–6124
July 2-7, 2026

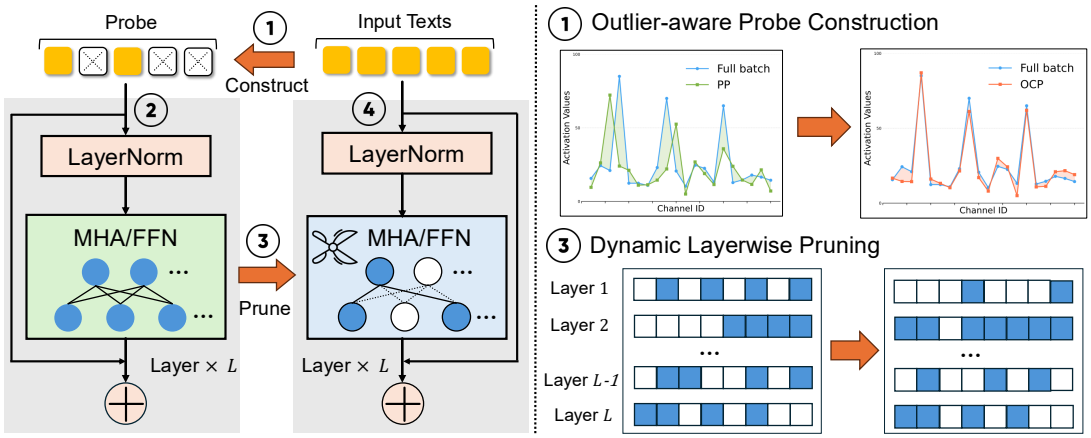


Figure 2: Overview of OCP. Left: Probe-based pruning workflow where ① a probe is constructed from input text, ② activations are computed via forward pass, ③ pruning masks are determined, and ④ pruning are applied for efficient inference. Right: OCP introduces (1) outlier-aware probe construction that better approximates full-batch importance compared to PP, and (2) dynamic layerwise pruning that adapts heterogeneous sparsity across layers.

layers, yet different layers exhibit varying importance (Yin et al., 2024; Chen et al., 2025). Uniform pruning risks over-removing critical channels, leading to performance sensitivity.

Recent studies (Xiao et al., 2023; Dettmers et al., 2022; An et al., 2025) reveal that activation outliers, features with magnitudes far exceeding typical values, are critical for model performance. Pruning these outlier channels causes substantial degradation, highlighting the need for probes that effectively capture outlier patterns. However, designing such probes poses three key challenges. First, outlier patterns are *input-dependent*, varying across samples. Approximating full-batch hidden activation distributions with a small probe is inherently ill-posed. Second, online pruning demands *low-latency* probe construction, yet identifying outlier-triggering tokens requires input-weight analysis, limiting room for complex strategies. Third, existing non-uniform sparsity methods rely on offline outlier analysis (Yin et al., 2024; Chen et al., 2025), yet in online settings, subsequent layers' statistics are unavailable when allocating sparsity for the current layer, complicating dynamic adjustment.

To address these challenges, we propose **OCP** (**O**utlier-**C**entric **P**robing), a principled framework for probe-based structured pruning of LLMs. Rather than reconstructing full distributions, OCP prioritizes capturing outlier-triggering tokens for accurate importance estimation without calibration. Given distinct outlier formation mechanisms in FFN and attention layers, we design layer-specific probing strategies. For FFN layers, we introduce a *sensitivity-weighted metric* that leverages precomputed weight aggregations to identify outlier-triggering tokens efficiently. For attention layers, we develop *attention-accumulated probing* that reuses attention matrices from preceding layers to identify salient tokens. For online sparsity allocation, we propose a history-guided algorithm with drift correction that dynamically adjusts layerwise pruning ratios based on probing outlier statistics.

Our contributions are summarized as follows:

- We introduce an outlier-centric perspective for probe-based pruning, showing that prioritizing outlier-triggering tokens enables accurate channel importance estimation.
- We propose three designs: sensitivity-weighted probing, attention-accumulated probing, and online adaptive sparsity allocation for efficient and accurate probe-aware pruning.
- Extensive experiments on LLaMA2, LLaMA3, and OPT demonstrate that OCP consistently outperforms state-of-the-art methods, achieving up to 25% perplexity reduction at 1.6× speedup.

2 Preliminaries

LLM Architecture. Consider an LLM with L transformer blocks. Let $\mathbf{X}^{(l)} \in \mathbb{R}^{B\times T\times D}$ denote the hidden states at layer l, where B is the batch size, T is the sequence length, and D is the hidden dimension. Each transformer block consists of a Multi-Head Self-Attention (MHA) module fol-

lowed by a Feed-Forward Network (FFN):

$$\begin{aligned} \mathrm{MHA}(\mathbf{X}) &= \mathrm{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}}\right)\mathbf{V}\mathbf{W}_O, \\ \mathrm{FFN}(\mathbf{X}) &= \left(\sigma(\mathbf{X}\mathbf{W}_g) \odot (\mathbf{X}\mathbf{W}_u)\right)\mathbf{W}_d, \end{aligned} \tag{1}$$

where $\mathbf{Q} = \mathbf{X}\mathbf{W}_Q$, $\mathbf{K} = \mathbf{X}\mathbf{W}_K$, and $\mathbf{V} = \mathbf{X}\mathbf{W}_V$ are the query, key, and value matrices, respectively. Here $\sigma(\cdot)$ denotes the activation function (e.g., SiLU), and $\odot$ represents element-wise multiplication. For simplicity, we omit attention head indices and bias terms. The residual transition is given by $\mathbf{X}^{(l+1)} = \mathbf{H}^{(l)} + \mathrm{FFN}(\mathrm{LN}(\mathbf{H}^{(l)}))$, where $\mathbf{H}^{(l)} = \mathbf{X}^{(l)} + \mathrm{MHA}(\mathrm{LN}(\mathbf{X}^{(l)}))$.

Structured Pruning. For hardware-friendly acceleration, structured pruning (Wang et al., 2025a; Ma et al., 2023) removes entire architectural units, such as attention heads and FFN neurons. For example, pruning an FFN neuron involves removing columns in the input projection matrices $\mathbf{W}_g$ and $\mathbf{W}_u$ and rows in the output projection matrix $\mathbf{W}_d$. Consequently, intermediate activations (i.e., inputs to $\mathbf{W}_O$ or $\mathbf{W}_d$) serve as the primary signal for importance estimation in existing methods (Ma et al., 2023; An et al., 2024; Zhang et al., 2024).

Probe-based Pruning. To enable input-adaptive pruning, PP (Le et al., 2025) proposes using a lightweight *probe* to estimate channel importance prior to each layer's full execution. Let $f_l : \mathbb{R}^{B\times T\times D} \to \mathbb{R}^{B\times T\times D'}$ denote the input projection at layer l that produces intermediate activations, and $g : \mathbb{R}^{B\times T\times D'} \to \mathbb{R}^{D'}$ be an importance scoring function that aggregates activations into D' scores corresponding to prunable units. A probing operator $\mathcal{S}_\theta$, parameterized by policy θ, selects a sparse subset of tokens from normalized input:

$$\mathbf{P}^{(l)} = \mathcal{S}_\theta(\mathrm{LN}(\mathbf{X}^{(l)})) \in \mathbb{R}^{B'\times T'\times D}, \tag{2}$$

where $B' \ll B$ and $T' \ll T$ denote the reduced batch size and sequence length, respectively. The full-batch importance is $\mathbf{I}^* = g(f_l(\mathrm{LN}(\mathbf{X}^{(l)})))$, and the probe-based estimate is $\hat{\mathbf{I}} = g(f_l(\mathbf{P}^{(l)}))$. Pruning masks are derived by removing the lowest-scored channels at sparsity ratio $\rho \in (0, 1)$. The goal is to design *an efficient online policy* θ such that probe-based masks align with full-batch decisions under probing budget.

Specifically, PP constructs the probe $\mathbf{P}^{(l)}$ by selecting the top-T' tokens and top-B' samples based on the L^2 norm of hidden states. The scoring function g then computes importance $\hat{\mathbf{I}}$ from the probe's intermediate activations $f_l(\mathbf{P}^{(l)})$, combined with pre-calibrated activation statistics. Hence, probe-based pruning is compatible with existing magnitude-based structured pruning methods (e.g., FLAP (An et al., 2024)). By default, PP applies a uniform sparsity ratio ρ across all layers.

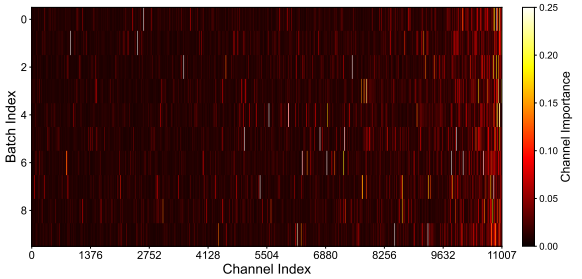
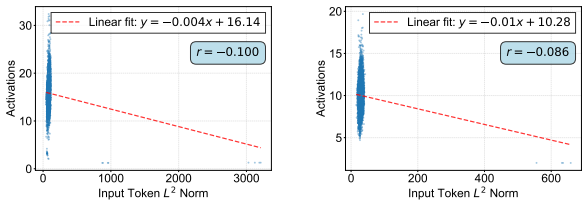


Figure 3: Channel importance heatmap of Layer 20's FFN in LLaMA-2-7B across different batches.



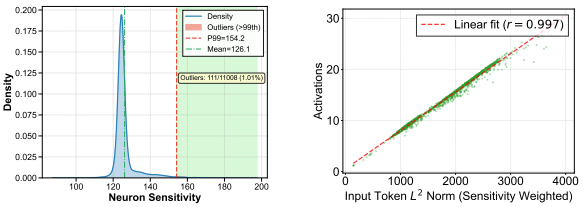
(a) LLaMA2-7B (b) LLaMA3-8B

Figure 4: Relationship between input token L^2 norm and FFN middle-layer activation magnitude.

Outlier Phenomenon in LLMs. Recent studies reveal that modern LLMs exhibit *activation outliers*: a small fraction of hidden dimensions display magnitudes orders larger than the mean (Dettmers et al., 2022; Sun et al., 2024a; Yin et al., 2024). These outliers emerge systematically across layers and significantly impact model behavior. Unlike previous studies that focus on *where* and *why* outliers occur, we investigate *which input tokens* are most likely to trigger outlier activations in hidden layers, and leverage these insights for more effective accurate input-adaptive pruning decisions.

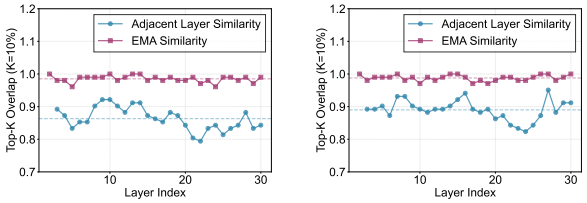
3 Proposed Method

We perform a preliminary study of outlier dynamics in LLMs. As visualized in Figure 3, a sparse set of outlier channels dominates activation importance, and these patterns vary across input batches. To select probe tokens that capture these large activations, existing methods (Le et al., 2025) rely on the *energy assumption*, selecting tokens with high L_2 norm. However, as shown in Figure 4, input



(a) Distribution of ω_k (b) Sensitivity norm

Figure 5: Analysis of sensitivity vector. (a) Distribution of per-dimension sensitivity ω_k, with 1% outlier dimensions. (b) Sensitivity-weighted norm strongly correlates with activation magnitude ($r = 0.997$).



(a) LLaMA2-7B (b) LLaMA3-8B

Figure 6: Attention similarity between adjacent layers in (a) LLaMA2-7B and (b) LLaMA3-8B.

norm exhibits near-zero correlation with hidden activation magnitude, causing the probe to miss critical outliers. This mismatch motivates us to enhance probe design by modeling outlier formation mechanisms (An et al., 2025; Sun et al., 2024a).

3.1 Sensitivity-Weighted FFN Probing

Observation. As shown in Figure 5a, the per-dimension sensitivity ω_k exhibits a skewed distribution, where approximately 1% of input dimensions contain disproportionately large values. This arises because $\mathbf{W}_g$ and $\mathbf{W}_u$ harbor outlier weights in sparse dimensions, corroborating prior findings that extreme activations emerge from interactions between inputs and outlier-heavy weights (An et al., 2025; Sun et al., 2024a). This motivates us to design a metric that prioritizes dimensions with high outlier-inducing potential.

Sensitivity-Weighted Metric. We define a per-layer *sensitivity vector* $\boldsymbol{\omega}^{(l)} \in \mathbb{R}^D$ that quantifies each input dimension's propensity to trigger outliers in layer l's FFN:

$$\omega_k^{(l)} = \sum_{j=1}^{D'} \left(|[\mathbf{W}_g^{(l)}]_{kj}| + |[\mathbf{W}_u^{(l)}]_{kj}| \right), \quad (3)$$

where $\omega_k^{(l)}$ aggregates the total weight magnitude connected to input dimension k. The *sensitivity-weighted score* for token position j is computed from $\mathrm{LN}(\mathbf{X}^{(l)}) \in \mathbb{R}^{B\times T\times D}$:

$$S_{\mathrm{ffn},i}^{(l)} = \left\| \mathrm{LN}(\mathbf{X}^{(l)})_{:,i,:} \odot \boldsymbol{\omega}^{(l)} \right\|_2, \quad (4)$$

which re-weights each input dimension by its outlier potential. The design rationale is straightforward: since FFN layers process tokens independently, activation magnitude depends on the element-wise alignment between input values and weight magnitudes. Unlike L2 norm that treats all dimensions equally, our metric prioritizes outlier-heavy dimensions that contribute disproportionately to activations, thereby accurately predicting which tokens will trigger extremes.

Empirically, this metric achieves strong correlation with true activation magnitude ($r = 0.997$, Figure 5b), compared to near-zero correlation ($r \approx 0$) of L2 norm (Figure 4). The FFN probe selects top-T' tokens: $\mathcal{T}_{T'}^{(l)} = \text{top-}T'(\{S_{\mathrm{ffn},i}^{(l)}\}_{i=1}^{T})$.

3.2 Attention-Accumulated Outlier Probing

Unlike FFN layers, which operate on tokens independently, attention layers dynamically aggregate contextual information, making static weight-based metrics less effective. To address this, we leverage the observation that tokens assigned high attention scores often carry outlier-inducing information (An et al., 2025; Chen et al., 2025), and use runtime attention patterns to identify outlier-triggering positions.

A key challenge is that attention weights at layer l are unavailable during probe construction (which precedes layer computation). We address this by using attention patterns of full-batch data from layer $l-1$. As shown in Figure 6 (blue curves), adjacent layers exhibit substantial overlap in attended tokens, validating this approximation. To improve stability, we further aggregate attention scores from multiple preceding layers via exponential moving average. This allows capturing persistent outlier tokens more robustly than single-layer snapshots.

Formally, we maintain a cumulative importance score $\mathbf{S}_{\mathrm{attn}}^{(l)} \in \mathbb{R}^T$ for layer l. We first compute the total attention received by each token. Let $\mathbf{A}_h^{(l-1)} \in \mathbb{R}^{T\times T}$ denote the attention matrix of head h at layer $l-1$. We aggregate across all heads and query positions:

$$A_{\mathrm{acc},i}^{(l-1)} = \sum_{h=1}^{H} \sum_{q=1}^{T} A_{h,q,i}^{(l-1)}, \quad (5)$$

where H is the number of attention heads. The cumulative score is then updated via:

$$S^{(l)}_{\text{attn},i} = \alpha \cdot S^{(l-1)}_{\text{attn},i} + (1-\alpha) \cdot A^{(l-1)}_{\text{acc},i}, \quad (6)$$

where $\alpha \in [0,1]$ controls the decay rate. As shown in Figure 6 (purple curves), history-informed predictions maintain top token similarity above 0.95 across layers, outperforming single-layer baselines (blue curves). Tokens with the highest $S^{(l)}_{\text{attn},i}$ scores are selected for probing at layer l.

Extending to Full-Batch Selection. Same as prior work (Le et al., 2025), we adopt a two-stage sequential selection process. After identifying top-T' tokens via our outlier-aware scores (specific for attention/FFN layers), we further select top-B' samples based on the accumulated scores of their hidden states over all tokens, constructing the final probe $\mathbf{P}^{(l)} \in \mathbb{R}^{B' \times T' \times D}$.

3.3 Heterogeneous Layerwise Sparsity

Outlier distributions vary significantly across layers: Layers with dense outlier activations often demand lower sparsity to preserve accuracy, whereas those with sparse outliers tolerate aggressive pruning (Wang et al., 2025a; Chen et al., 2025; Yin et al., 2024). Existing methods typically rely on offline calibration with fixed sparsity, overlooking input-dependent outlier patterns. To address this, we introduce an online algorithm that adapts layerwise sparsity using probe-based outlier estimation.

Outlier Density Estimation. For each layer l at batch t, we estimate outlier density using hidden activations from the probe $\mathbf{P}^{(l)}$ as:

$$D^{(l)}_t = \frac{1}{|f_l(\mathbf{P}^{(l)})|} \sum_{z \in f_l(\mathbf{P}^{(l)})} \mathbb{I}\left(|z| > \bar{a}^{(l)} + 2\sigma^{(l)}\right) \quad (7)$$

where $\mathbb{I}(\cdot)$ is the indicator function, $\bar{a}^{(l)}$ and $\sigma^{(l)}$ are the mean and deviation of probe activations.

History-Guided Adaptive Sparsity Allocation. Naïvely allocating sparsity directly based on $D^{(l)}_t$ faces two challenges: (1) in online settings, lack of subsequent outlier estimations makes sparsity allocation unstable; and (2) unconstrained adjustments could cause global sparsity to deviate from ρ_{target}. To tackle these, we propose a two-stage strategy: (1) deriving a stable baseline informed by historical outlier patterns, and (2) enforcing global constraints through drift correction.

We first track each layer's historical outlier density by maintaining a running mean $\mu^{(l)}_{\text{hist}}$:

$$\mu^{(l)}_{\text{hist},t} = \beta \cdot \mu^{(l)}_{\text{hist},t-1} + (1-\beta) \cdot D^{(l)}_t, \quad (8)$$

and deriving layer-specific baseline sparsity relative to the global mean $\bar{\mu}_{t-1} = \frac{1}{L}\sum_{k=1}^{L} \mu^{(k)}_{\text{hist},t-1}$:

$$\rho^{(l)}_{\text{base},t} = \rho_{\text{target}} - \gamma \cdot \left(\mu^{(l)}_{\text{hist},t} - \bar{\mu}_{t-1}\right), \quad (9)$$

where γ controls the adjustment strength. This baseline adjusts sparsity based on relative outlier density, assigning lower sparsity to historically outlier-dense layers.

To prevent systematic drift from ρ_{target}, we apply drift correction by tracking the cumulative deviation $\mathcal{R}^{(l-1)}_t$ from layers 1 to $l-1$ (initialized as $\mathcal{R}^{(0)}_t = 0$). The final sparsity is computed as:

$$\rho^{(l)}_t = \text{Clip}\left(\rho^{(l)}_{\text{base},t} + \frac{\mathcal{R}^{(l-1)}_t}{L-l+1}, \rho_{\min}, \rho_{\max}\right), \quad (10)$$

where the correction term $\frac{\mathcal{R}^{(l-1)}_t}{L-l+1}$ distributes the accumulated deviation evenly across the remaining layers. The cumulative deviation is then updated:

$$\mathcal{R}^{(l)}_t = \mathcal{R}^{(l-1)}_t + (\rho_{\text{target}} - \rho^{(l)}_t). \quad (11)$$

This mechanism balances historical outlier patterns with global sparsity constraints, enabling input-adaptive allocation.

3.4 Complexity Analysis

The computational overhead of probe-based pruning consists of probe construction and probe inference. Compared to the baseline (Le et al., 2025), our outlier-aware construction introduces minimal additional cost: (1) the FFN sensitivity vector $\boldsymbol{\omega}^{(l)} \in \mathbb{R}^D$ is precomputed offline in $O(DD')$ per layer with zero runtime overhead; (2) the attention probe reuses the existing full-batch attention matrices from layer $l-1$, requiring only $O(HT^2)$ for score aggregation; (3) adaptive sparsity maintains lightweight per-layer scalars ($\mu^{(l)}_{\text{hist},t}, \mathcal{R}^{(l)}_t \in \mathbb{R}$) updated in $O(L)$ per batch. Once the probe $\mathbf{P}^{(l)} \in \mathbb{R}^{B' \times T' \times D}$ is constructed, inference cost remains identical to the baseline: $O(r_B r_T \cdot BTD^2 + r_B r_T^2 \cdot BT^2D)$, where $r_B = B'/B$ and $r_T = T'/T$. In the default setting, the total overhead is approximately 1.5% of dense inference FLOPs. We present empirical runtime analysis in Section 4.4, and provide additional analysis in Appendix A.

Table 1: Main results across LLaMA-2, LLaMA-3, and OPT models. PPL denotes average perplexity on WikiText-2 and PTB (↓); ACC denotes averaged accuracy across seven benchmarks (↑). Best results are in **bold**.

Ratio	Method	Weight Update	Calib.-Data Free	Input Adaptive	LLaMA-2-7B		LLaMA-2-13B		LLaMA-3-8B		OPT-13B	
					PPL ↓	ACC ↑	PPL ↓	ACC ↑	PPL ↓	ACC ↑	PPL ↓	ACC ↑
0%	Raw Model	-	-	-	13.8	64.0	16.8	66.2	9.4	65.6	12.6	57.2
20%	Wanda-sp	✘	✘	✘	30.9	61.5	28.1	65.0	18.5	61.3	24.1	55.2
	FLAP	✘	✘	✘	24.4	61.4	22.1	64.6	21.8	62.5	21.1	54.9
	LoRAPrune	✔	✘	✘	19.4	59.2	22.6	61.0	-	-	-	-
	LLM-Pruner	✔	✘	✘	26.9	58.7	25.8	62.1	-	-	-	-
	CFSP	✘	✘	✘	22.5	61.3	19.8	65.3	18.1	62.1	24.2	55.1
	PP	✘	✘	✔	16.9	62.8	20.0	65.3	14.6	63.4	15.1	56.5
	OCP (ours)	✘	✔	✔	**16.1**	**63.9**	**18.1**	**66.3**	**12.6**	**64.5**	**13.9**	**57.0**
40%	Wanda-sp	✘	✘	✘	130.1	54.8	58.8	56.6	75.0	53.3	52.2	50.5
	FLAP	✘	✘	✘	84.4	54.9	45.5	60.6	109.7	53.6	58.2	50.8
	LoRAPrune	✔	✘	✘	29.7	52.1	37.6	55.5	-	-	-	-
	LLM-Pruner	✔	✘	✘	46.4	50.6	46.0	54.7	-	-	-	-
	CFSP	✘	✘	✘	87.8	54.8	48.7	57.9	72.0	53.7	57.6	50.8
	PP	✘	✘	✔	33.4	56.6	36.1	61.0	43.7	58.9	26.5	53.1
	OCP (ours)	✘	✔	✔	**25.2**	**58.4**	**30.5**	**63.2**	**25.9**	**60.2**	**21.4**	**54.8**

4 Experiments

4.1 Experimental Setup

Models. We conduct experiments on LLaMA-2 (7B and 13B) (Touvron et al., 2023), LLaMA-3 (8B) (Grattafiori et al., 2024), and OPT (13B) (Zhang et al., 2022), covering a range of model scales and architectures to validate the generalizability of our method.

Evaluation. We use perplexity metric to report language comprehension performance, evaluated on WikiText2 (Merity et al., 2016), and PTB (Marcus et al., 1993) datasets. We also evaluate zero-shot reasoning performance on various benchmarks, including ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), Winogrande (Sakaguchi et al., 2020), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), and OpenBookQA (Mihaylov et al., 2018).

Baselines. We evaluate OCP in comparison with a range of existing methods, including LLM-Pruner (Ma et al., 2023), FLAP (An et al., 2024), Wanda-sp (Sun et al., 2024b), LoRAPrune (Zhang et al., 2024), CFSP (Wang et al., 2025c), and PP (Le et al., 2025).

Implementation Details. We implement OCP in PyTorch on NVIDIA A6000 GPUs. Following (Le et al., 2025), we set the sequence length to 1024 for language modeling and the batch size to 20. The default probe configuration is 5% batch size and 50% sequence length. Sensitivity vectors for FFN probing are pre-collected offline. We evaluate at 20% and 40% overall sparsity. The target sparsity for attention and FFN modules are set equal by default. Results are averaged over three seeds. More experimental details are provided in Appendix B.

4.2 Main Results

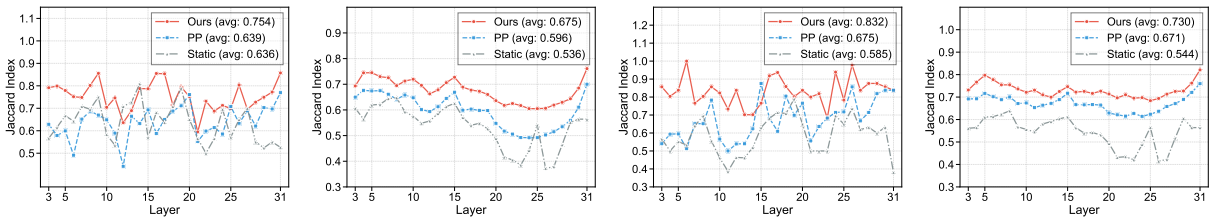
Table 1 presents the main evaluation results across three model families, and Table 2 provides detailed results on LLaMA-2-7B. Additional results are provided in Section F. We make the following observations: (1) For static pruning methods, fine-tuning becomes increasingly critical at higher sparsity ratios. At 40% sparsity on LLaMA-2-7B, LLM-Pruner (29.7 PPL) dramatically outperforms one-shot methods like FLAP (84.4 PPL) and Wanda-sp (130.1 PPL). (2) Dynamic pruning methods substantially outperform static approaches across diverse tasks. PP achieves 16.8 PPL versus the best static method's 43.8 PPL at 40% sparsity on LLaMA-2-7B. This advantage extends to zero-shot tasks, where PP and OCP consistently outperform static baselines across WikiText2, PTB, BoolQ, and PIQA, validating the importance of input-adaptive pruning decisions. (3) OCP further advances dynamic pruning by addressing PP's outlier handling limitations. At 40% sparsity, OCP achieves 12.5 PPL versus PP's 16.8 PPL on WikiText2 and demonstrates stronger generalization on PTB (38.0 PPL vs 50.0 PPL).

4.3 Ablation and Parameter Studies

Component Effect Analysis. Table 3 isolates each component's contribution. We can observe that outlier-aware probing alone (12.9 PPL) al-

Table 2: Performance comparison on LLaMA-2-7B across datasets and ratios. The best is highlighted in bold.

Ratio	Method	WikiText2 ↓	PTB ↓	BoolQ ↑	PIQA ↑	HellaSwag ↑	WinoGrande ↑	ARC-c ↑	ARC-e ↑	OBQA ↑
0%	Dense	6.0 0.0	21.5 0.0	74.6 0.0	77.9 0.0	75.0 0.0	67.7 0.0	42.7 0.0	67.3 0.0	42.6 0.0
20%	Wanda-sp	10.6 0.1	51.1 0.0	65.3 0.1	77.2 0.1	74.1 0.0	67.1 0.2	41.1 0.1	63.9 0.3	41.8 0.2
	FLAP	10.3 0.1	38.5 0.0	67.3 0.5	76.6 0.2	73.0 0.1	67.4 0.0	40.6 0.3	63.1 0.1	42.0 0.1
	LoRAPrune	8.7 0.2	30.2 0.6	67.0 0.9	76.5 0.2	69.9 0.1	63.2 0.3	36.7 0.2	58.9 0.9	42.3 0.2
	LLM-Pruner	10.2 0.3	43.5 1.2	66.6 1.3	76.1 0.6	68.4 0.5	62.8 1.1	36.3 0.4	59.8 0.3	40.7 0.7
	CFSP	10.3 0.0	34.7 0.2	65.2 0.2	76.7 0.2	73.4 0.2	67.2 0.2	41.2 0.2	63.8 0.2	41.9 0.1
	PP	8.1 0.1	25.7 0.0	69.0 0.1	78.1 0.0	73.5 0.0	66.7 0.3	42.8 0.1	68.5 0.0	40.9 0.2
	Ours	**7.5** 0.0	**24.7** 0.3	**70.6** 0.1	**78.1** 0.0	**74.8** 0.1	**67.8** 0.1	**43.2** 0.1	**68.9** 0.1	**43.6** 0.2
40%	Wanda-sp	43.8 1.5	216.4 0.0	62.5 0.1	72.5 0.1	63.3 0.0	56.9 0.1	33.4 0.2	54.4 0.1	40.8 0.4
	FLAP	38.9 1.3	129.9 0.0	63.5 0.1	71.7 0.3	63.3 0.1	**59.8** 0.1	33.8 0.6	52.5 0.2	40.0 0.6
	LoRAPrune	13.6 0.4	45.7 1.7	62.9 0.2	70.8 0.1	58.6 0.1	55.5 0.7	30.9 0.4	49.6 0.4	36.7 0.4
	LLM-Pruner	20.3 1.3	72.5 3.2	57.5 4.0	71.3 1.2	55.7 1.3	53.1 0.5	28.9 0.7	50.4 0.5	37.3 0.6
	CFSP	38.5 0.2	137.2 0.5	63.6 0.0	71.6 0.2	63.3 0.1	56.9 0.2	34.0 0.3	53.0 0.0	40.9 0.2
	PP	16.8 0.1	50.0 0.0	62.7 0.2	**74.9** 0.1	63.6 0.0	57.5 0.2	35.5 0.1	61.7 0.2	40.3 0.4
	Ours	**12.5** 0.0	**38.0** 0.2	**63.9** 0.2	74.6 0.1	**67.6** 0.0	58.7 0.2	**36.0** 0.1	**66.5** 0.1	**41.2** 0.0



(a) WikiText-2, Attention (b) WikiText-2, FFN (c) Arc-c, Attention (d) Arc-c, FFN

Figure 7: Jaccard similarity of pruning channels compared to full-batch probing across layers.

Table 3: Effect of each component at 40% sparsity.

Outlier-centric Probe	Non-uniform Sparsity	LLaMA-2-7B		OPT-13B	
		WikiText	PTB	WikiText	PTB
✓		12.9	39.3	21.1	22.9
	✓	15.6	46.2	24.7	25.0
✓	✓	12.5	38.0	20.5	22.2

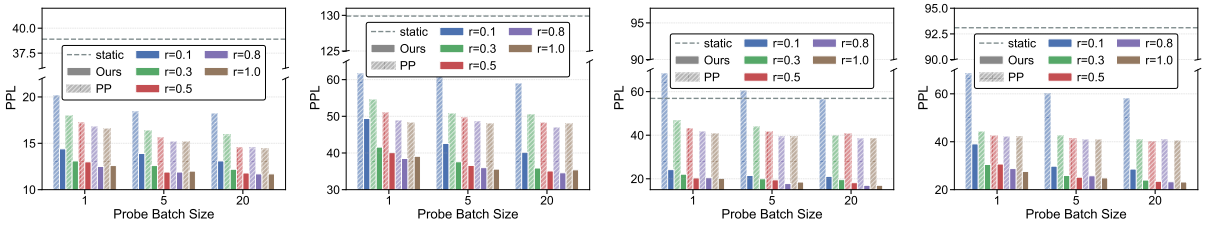
ready surpasses PP's full configuration with non-uniform sparsity (15.6 PPL), demonstrating that accurate probe construction is the primary bottleneck. Only allocating heterogeneous sparsity cannot compensate for probes that miss critical channels. Combining both components reduce perplexity to 12.5 by preserving capacity in outlier-dense layers. The consistent pattern on OPT-13B (20.5 combined vs. 21.1/24.7 individually) confirms both components contribute complementarily: probing identifies *which* channels matter, while adaptive allocation determines *how much* channels to preserve.

Effect of Probe Size. Figure 8 reveals that performance saturates beyond $r = 0.3$, indicating only 30% of tokens suffice to capture outlier distributions. This validates our design premise: outlier patterns are concentrated rather than uniformly distributed, making exhaustive probing unnecessary. Crucially, OCP's advantage over PP widens at smaller probe sizes (e.g., $r = 0.1$), showing that outlier-aware selection extracts more signal per token a critical property for latency-constrained deployment where probe budgets are tight.

Probe Quality Analysis. Figure 7 validates our core hypothesis by measuring Jaccard similarity between probe-based and full-batch oracle decisions. OCP achieves substantially higher alignment than PP on both WikiText2 (attention: 0.754 vs. 0.639; FFN: 0.675 vs. 0.596) and ARC-Challenge (attention: 0.832 vs. 0.675; FFN: 0.730 vs. 0.671). Notably, the improvement is more pronounced on reasoning tasks (+23% for attention), suggesting that outlier-triggering tokens carry task-critical semantic information that energy-based selection overlooks. This explains why OCP's gains amplify on downstream benchmarks beyond perplexity.

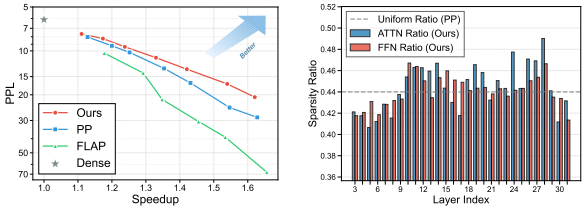
4.4 Efficiency and Sparsity Analysis

Accuracy-Speedup Trade-off. Figure 9a compares the perplexity-speedup Pareto frontier on LLaMA-2-7B. OCP consistently dominates across all speedup levels, with the performance gap widening at higher compression rates where accurate importance estimation becomes increasingly



(a) WikiText-2, LLaMA-2-7B (b) PTB, LLaMA-2-7B (c) WikiText-2, LLaMA-3-8B (d) PTB, LLaMA-3-8B

Figure 8: Ablation study on probe sequence ratio and batch size across different datasets and models. Different colors represent different probe sequence ratios (r=0.1, 0.3, 0.5). Darker bars indicate our method, while lighter bars represent the PP baseline. The dashed line indicates the performance of static scheme.



(a) Accuracy-speedup curve (b) Layer-wise sparsity

Figure 9: Analysis on LLaMA-2-7B. (a) Pareto frontier of perplexity versus speedup. (b) Layer-wise sparsity distribution at 40% sparsity. OCP achieves lower perplexity at equivalent speedups and learns non-uniform sparsity patterns that preserve outlier-dense layers.

critical. Notably, OCP at 1.6× speedup achieves comparable perplexity (∼20) to PP at 1.4×, effectively providing 14% additional acceleration at equivalent quality. This demonstrates that outlier-centric probing translates directly into practical efficiency gains: by preserving the channels that matter most, OCP maintains model quality under aggressive pruning where baseline methods degrade rapidly.

Learned Sparsity Patterns. Figure 9a(b) reveals the layer-wise sparsity distribution learned by our online adaptive allocation at 40% overall sparsity. In contrast to PP's uniform allocation, OCP automatically discovers a heterogeneous pattern: higher sparsity in middle layers, with more capacity preserved in early and late layers. This emergent behavior reflects the distinct functional roles across layers: early layers establish input representations while late layers are critical for output generation, making both ends more sensitive to pruning. Meanwhile, middle layers exhibit greater redundancy that can be exploited for compression. The pattern shows that our history-guided drift correction tracks input-dependent activation statistics, enabling principled capacity allocation adapted to each layer's sensitivity.

Table 4: Computational cost (GFLOPs) comparison on LLaMA-2-7B at 40% sparsity.

Method	WikiText2	PTB	ARC-c	BoolQ
Dense	4.42	1.11	4.38	27.02
PP	2.74 (↓38.0%)	0.69 (↓37.8%)	2.72 (↓37.9%)	16.75 (↓38.0%)
Ours	2.75 (↓37.8%)	0.69 (↓37.8%)	2.78 (↓36.5%)	17.08 (↓36.8%)

Table 5: Wall-clock latency (seconds per sample) on WikiText-2 at 60% sparsity using LLaMA-2-7B on a single NVIDIA A6000.

Batch	Dense (s)	PP			OCP		
		Latency (s)	Speedup	PPL	Latency (s)	Speedup	PPL
10	0.1776	0.0956	**1.86**×	61.1	0.0985	1.80×	**45.0**
20	0.1672	0.0821	**2.04**×	64.0	0.0853	1.96×	**41.2**
30	0.1664	0.0811	**2.05**×	59.8	0.0834	2.00×	**41.0**
40	0.1554	0.0747	**2.08**×	60.1	0.0761	2.04×	**43.9**
50	0.1406	0.0667	**2.11**×	59.1	0.0676	2.08×	**41.7**

High-Sparsity Regime. Table 5 extends our evaluation to the more aggressive 60% sparsity setting on LLaMA-2-7B, reporting end-to-end wall-clock latency per sample across batch sizes 10–50 on a single NVIDIA A6000. OCP achieves speedups of 1.80×–2.08× over the dense baseline, remaining within 3% of PP's latency across all batch sizes, confirming that our outlier-aware probe selection imposes negligible runtime overhead. Crucially, OCP consistently delivers approximately 30% lower perplexity than PP at every batch size (e.g., 41.2 vs. 64.0 at batch 20), demonstrating that the quality advantage of outlier-centric probing persists and even strengthens under more aggressive compression. This result further validates the scalability of OCP: as sparsity increases and accurate importance estimation becomes harder, the ability to identify outlier-triggering tokens becomes the dominant factor in maintaining model quality.

Computational Cost Analysis Table 4 compares the computational cost across different datasets. Both PP and OCP achieve substantial FLOPs reduction compared to the dense model,

with PP maintaining a consistent 38% reduction through uniform sparsity allocation. OCP exhibits slightly higher computational cost (36.5%–37.8% reduction) due to our adaptive sparsity mechanism that preserves more capacity in critical layers. However, this modest overhead (approximately 1–2% additional FLOPs) yields significant quality improvements: OCP achieves 12.5 perplexity on WikiText2 compared to PP's 16.8, representing a 25% quality gain for less than 2% additional computation. This favorable trade-off shows that strategically allocating compute to outlier-dense layers is more effective than uniform strategy.

5 Related Work

Model Pruning. Model compression techniques, including pruning (Zhong et al., 2025; Wei et al., 2025), quantization (Dettmers et al., 2022), and low-rank decomposition (Wang et al., 2025b), have been widely studied to reduce inference cost as neural models are applied across diverse tasks and domains (Ji et al., 2026; Zhang et al., 2026a). According to recent surveys (Zhu et al., 2024; Ji et al., 2025a), pruning methods for LLMs can be broadly categorized into three categories: unstructured (Zhao et al., 2025), semi-structured (Fang et al., 2024), and structured approaches (Wang et al., 2025c,a). Compared to the other two, structured pruning achieves hardware-friendly acceleration without requiring specialized hardware by removing entire architectural units (e.g., attention heads, FFN neurons). Most structured pruning methods (Ma et al., 2023; An et al., 2024; Zhang et al., 2024; Yang et al., 2025b; Wang et al., 2025a) adopt a static paradigm, where channel importance is estimated offline using calibration data to determine a fixed subnetwork applied to all inputs. Recently, dynamic pruning (Qiao et al., 2025; Wee et al., 2025) has gained attention for its ability to adaptively select subnetworks based on input semantics, leading to improved performance. For example, IF-Pruning (Hou et al., 2025) employs a well-trained sparsity predictor to generate dynamic masks based on input semantics, and PP (Le et al., 2025) utilizes a lightweight probe to estimate pruning decisions prior to each layer's execution. Building upon this probe pruning scheme, we advance it by designing outlier-aware probes coupled with heterogeneous layer-wise sparsity, leading to improved performance.

Activation Outliers in LLMs. Activation outliers, i.e., activations with exceptionally large magnitudes, have been recognized as a key factor influencing both the performance and efficiency of LLMs (Dettmers et al., 2022; Xiao et al., 2023; An et al., 2025). Prior studies have examined their effects in various settings, including quantization (Dettmers et al., 2022; Gong and Sun, 2025; Xiao et al., 2023), fine-tuning (Li et al., 2025), and sparsity allocation (Yin et al., 2024; Chen et al., 2025). More recent work has further investigated the mechanisms underlying the emergence of activation outliers (An et al., 2025). Building on these insights, our work improves probe design from the perspective of activation outliers, enabling more accurate identification of critical channels for dynamic pruning.

6 Conclusion

We present OCP, an outlier-centric framework for probe-based structured pruning of LLMs. Our key insight is that effective probes should prioritize capturing outlier-triggering tokens rather than reconstructing full activation distributions. Based on this insight, we propose sensitivity-weighted probing for FFN layers, attention-accumulated probing for attention layers, and online adaptive sparsity allocation with history-guided drift correction. Extensive experiments on LLaMA2, LLaMA3, and OPT demonstrate that OCP consistently outperforms state-of-the-art methods across various benchmarks, achieving up to 25% perplexity reduction at 1.6× speedup.

Limitations

We acknowledge the following limitations. First, OCP adopts a layer-by-layer probe-prune-execute pipeline that processes transformer blocks sequentially, which may limit potential speedup gains from multi-layer probing strategies that jointly optimize pruning decisions. Second, our OCP implementation is limited by standard PyTorch operations. Without hardware-specific optimizations such as custom sparse kernels, OCP cannot fully realize its potential for model acceleration.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. 62306255).

References

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

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2025. Systematic outliers in large language models. In *The Thirteenth International Conference on Learning Representations*.

Abhinav Bandari, Lu Yin, Cheng-Yu Hsieh, Ajay Kumar Jaiswal, Tianlong Chen, Li Shen, Ranjay Krishna, and Shiwei Liu. 2024. Is c4 dataset optimal for pruning? an investigation of calibration data for llm pruning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18089–18099.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*.

Yuli Chen, Bo Cheng, Jiale Han, Yingying Zhang, Yingting Li, and Shuhao Zhang. 2025. Dlp: Dynamic layerwise pruning in large language models. In *Forty-second International Conference on Machine Learning*.

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

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

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Llm. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in neural information processing systems*, 35:30318–30332.

Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. 2024. Maskllm: Learnable semi-structured sparsity for large language models. *Advances in Neural Information Processing Systems*, 37:7736–7758.

Zheng Gong and Ying Sun. 2025. Outlier-aware post-training quantization for discrete graph diffusion models. In *Forty-second International Conference on Machine Learning*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Jiujun He and Huazhen Lin. 2025. Olica: Efficient structured pruning of large language models without retraining. In *Forty-second International Conference on Machine Learning*.

Bairu Hou, Qibin Chen, Jianyu Wang, Guoli Yin, Chong Wang, Nan Du, Ruoming Pang, Shiyu Chang, and Tao Lei. 2025. Instruction-following pruning for large language models. In *Forty-second International Conference on Machine Learning*.

Yang Ji, Ying Sun, Yuting Zhang, Zhigaoyuan Wang, Yuanxin Zhuang, Zheng Gong, Dazhong Shen, Chuan Qin, Hengshu Zhu, and Hui Xiong. 2025a. A comprehensive survey on self-interpretable neural networks. *Proceedings of the IEEE*.

Yang Ji, Ying Sun, and Hengshu Zhu. 2026. Enhancing job salary prediction with disentangled composition effect modeling: a neural prototyping approach. *Frontiers of Computer Science*, 20(5):2005345.

Yixin Ji, Yang Xiang, Juntao Li, Qingrong Xia, Ping Li, Xinyu Duan, Zhefeng Wang, and Min Zhang. 2025b. Beware of calibration data for pruning large language models. In *The Thirteenth International Conference on Learning Representations*.

Qi Le, Enmao Diao, Ziyan Wang, Xinran Wang, Jie Ding, Li Yang, and Ali Anwar. 2025. Probe pruning: Accelerating llms through dynamic pruning via model-probing. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*.

Junyi Li, Chenweinan Jiang, Daixin Wang, Guo Ye, Libang Zhang, Huimei He, Binbin Hu, Zhiqiang Zhang, and Fuzhen Zhuang. 2026. Apl-llm: adaptive pseudo-labeling with large language models for few-shot node classification. *Frontiers of Computer Science*, 20(12):2012365.

Pengxiang Li, Lu Yin, Xiaowei Gao, and Shiwei Liu. 2025. Outlier-weighed layerwise sampling for llm fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19460–19473.

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

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 conference on empirical methods in natural language processing*.

Kangyu Qiao, Shaolei Zhang, and Yang Feng. 2025. Ig-pruning: Input-guided block pruning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

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

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *AAAI*.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024a. Massive activations in large language models. In *First Conference on Language Modeling*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024b. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey. *Transactions on Machine Learning Research*. Survey Certification.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. 2025a. Svd-llm v2: Optimizing singular value truncation for large language model compression. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4287–4296.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025b. Svd-llm: Truncation-aware singular value decomposition for large language model compression. In *The Thirteenth International Conference on Learning Representations*.

Yuxin Wang, Minghua Ma, Zekun Wang, Jingchang Chen, Shan Liping, Qing Yang, Dongliang Xu, Ming Liu, and Bing Qin. 2025c. Cfsp: an efficient structured pruning framework for llms with coarse-to-fine activation information. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9311–9328.

Juyun Wee, Minjae Park, and Jaeho Lee. 2025. Prompt-based depth pruning of large language models. In *Forty-second International Conference on Machine Learning*.

Jiateng Wei, Siqi Li, Jingyang Xiang, Jiandang Yang, Jun Chen, Xiaobin Wei, Yunliang Jiang, and Yong Liu. 2025. Oops: Outlier-aware and quadratic programming based structured pruning for large language models. *Neural Networks*, page 108332.

Jiateng Wei, Quan Lu, Ning Jiang, Siqi Li, Jingyang Xiang, Jun Chen, and Yong Liu. 2024. Structured optimal brain pruning for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13991–14007, Miami, Florida, USA. Association for Computational Linguistics.

Miles Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International conference on machine learning*, pages 38087–38099. PMLR.

Haoran Xin, Ying Sun, Chao Wang, and Hui Xiong. 2025. Llmcdsr: Enhancing cross-domain sequential recommendation with large language models. *ACM Transactions on Information Systems*, 43(5):1–33.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao,

Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. 2025b. Wanda++: Pruning large language models via regional gradients. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 4321–4333.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, AJAY KUMAR JAISWAL, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. 2024. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning LLMs to high sparsity. In *Forty-first International Conference on Machine Learning*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.

Mingxu Zhang, Huicheng Zhang, Jiaming Ji, Yaodong Yang, and Ying Sun. 2026a. Enhance the safety in reinforcement learning by adrc lagrangian methods. *arXiv preprint arXiv:2601.18142*.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024. Loraprune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3013–3026.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Yuting Zhang, Ziliang Pei, Chao Wang, Ying Sun, and Fuzhen Zhuang. 2026b. Enhancing llm-based recommendation with preference hint discovery from knowledge graph. *arXiv preprint arXiv:2601.18096*.

Pengxiang Zhao, Hanyu Hu, Ping Li, Yi Zheng, Zhefeng Wang, and Xiaoming Yuan. 2025. Fistapruner: Layer-wise post-training pruning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. 2025. BlockPruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 5065–5080, Vienna, Austria. Association for Computational Linguistics.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

Appendix

We provide the additional contents as follows:

- Appendix A: Detailed Analysis of Heterogeneous Layerwise Sparsity
- Appendix B: Experimental Details
- Appendix C: Sensitivity to Calibration Data
- Appendix D: Attention vs. FFN Pruning
- Appendix E: Generations From Pruned Models
- Appendix F: Additional Evaluation Results (LLaMA-2-13B, LLaMA-3-8B, OPT-13B, Qwen3-8B)

A Detailed Analysis of Heterogeneous Layerwise Sparsity

We provide theoretical analysis of the proposed heterogeneous layerwise sparsity mechanism, focusing on the convergence properties of history-guided estimation and the global constraint guarantee of drift correction.

A.1 Convergence and Variance Reduction Analysis of History-guided Estimation

The historical density estimate follows the recurrence $\mu^{(l)}_{\text{hist},t} = \beta\mu^{(l)}_{\text{hist},t-1} + (1-\beta)D^{(l)}_t$, which expands to:

$$\mu^{(l)}_{\text{hist},t} = (1-\beta)\sum_{k=0}^{t-1}\beta^k D^{(l)}_{t-k} + \beta^t \mu^{(l)}_{\text{hist},0}. \quad (12)$$

Convergence. When $D^{(l)}_t \to D^{(l)}$ (stationary), since the weights $(1-\beta)\beta^k$ form a valid probability distribution (summing to $1-\beta^t \to 1$), and $\beta^t \to 0$, we have $\mu^{(l)}_{\text{hist},t} \to D^{(l)}$.

Variance Reduction. For a fixed layer l, assume the sequence $\{D^{(l)}_t\}_{t=1}^{\infty}$ across different batches are i.i.d. with mean μ and variance σ^2. For sufficiently large t, the variance of the EMA estimator is:

$$\begin{aligned}\text{Var}(\mu^{(l)}_{\text{hist},t}) &= (1-\beta)^2\sum_{k=0}^{t-1}\beta^{2k}\sigma^2 \\ &\xrightarrow{t\to\infty} (1-\beta)^2\cdot\frac{\sigma^2}{1-\beta^2} \\ &= \frac{1-\beta}{1+\beta}\sigma^2.\end{aligned} \quad (13)$$

For typical values (e.g., $\beta = 0.9$), this yields a variance reduction factor of $\frac{1-\beta}{1+\beta} = \frac{1}{19} \approx 0.053$, reducing estimation variance by over 94% compared to single-sample estimation. This substantial noise reduction enables stable sparsity allocation across input batches.

A.2 Global Constraint Guarantee via Drift Correction

Proposition 1. *Without clipping, drift correction ensures* $\frac{1}{L}\sum_{l=1}^{L}\rho^{(l)}_t = \rho_{\text{target}}$.

Proof. Define the residual at layer l as $\delta^{(l)}_t = \rho_{\text{target}} - \rho^{(l)}_{\text{base},t}$, representing the deviation of baseline sparsity from the target. The cumulative deviation before layer l is $\mathcal{R}^{(l-1)}_t = \sum_{i=1}^{l-1}(\rho_{\text{target}} - \rho^{(i)}_t)$, initialized as $\mathcal{R}^{(0)}_t = 0$. The drift-corrected sparsity is:

$$\rho^{(l)}_t = \rho^{(l)}_{\text{base},t} + \frac{\mathcal{R}^{(l-1)}_t}{L-l+1}. \quad (14)$$

We prove by induction that $\mathcal{R}^{(l)}_t = \sum_{i=1}^{l}\delta^{(i)}_t \cdot \frac{L-l}{L-i+1}$.

Base case: For $l=1$, $\rho^{(1)}_t = \rho^{(1)}_{\text{base},t} + 0 = \rho^{(1)}_{\text{base},t}$, so $\mathcal{R}^{(1)}_t = \delta^{(1)}_t$, which equals $\delta^{(1)}_t \cdot \frac{L-1}{L}$ only when verified through the update rule.

Terminal condition: At layer L:

$$\begin{aligned}\rho^{(L)}_t &= \rho^{(L)}_{\text{base},t} + \frac{\mathcal{R}^{(L-1)}_t}{1} \\ &= \rho^{(L)}_{\text{base},t} + \mathcal{R}^{(L-1)}_t.\end{aligned} \quad (15)$$

Thus:

$$\begin{aligned}\mathcal{R}^{(L)}_t &= \mathcal{R}^{(L-1)}_t + (\rho_{\text{target}} - \rho^{(L)}_t) \\ &= \mathcal{R}^{(L-1)}_t + \delta^{(L)}_t - \mathcal{R}^{(L-1)}_t = \delta^{(L)}_t.\end{aligned} \quad (16)$$

Since $\sum_{l=1}^{L}\delta^{(l)}_t = L\cdot\rho_{\text{target}} - \sum_{l=1}^{L}\rho^{(l)}_{\text{base},t} = 0$ by design of $\rho^{(l)}_{\text{base},t}$, we have $\mathcal{R}^{(L)}_t = 0$, which implies:

$$\sum_{l=1}^{L}\rho^{(l)}_t = L\cdot\rho_{\text{target}}. \quad (17)$$

□

The correction term $\frac{\mathcal{R}^{(l-1)}_t}{L-l+1}$ acts as a sparsity controller: if preceding layers consume less sparsity budget (lower ρ), the accumulated surplus is evenly distributed to subsequent layers, ensuring the global budget is exactly met.

Effect of Clipping. In practice, we apply clipping to constrain $\rho_t^{(l)} \in [\rho_{\min}, \rho_{\max}]$, which may cause the global constraint to be violated. However, clipping is triggered only when the correction term causes excessive deviation from $\rho_{\text{base},t}^{(l)}$. Since the history-guided baseline estimation significantly reduces variance (as shown above), and the adjustment strength γ controls the magnitude of layer-wise deviations, appropriate parameter selection (moderate γ and β close to 1) ensures that $\rho_{\text{base},t}^{(l)}$ remains stable and well within $[\rho_{\min}, \rho_{\max}]$. Empirically, we observe that clipping is rarely activated under our default settings, and the actual global sparsity closely approximates ρ_{target}.

B Experimental Details

Details of Baseline Methods. We compare our method with the following baselines:

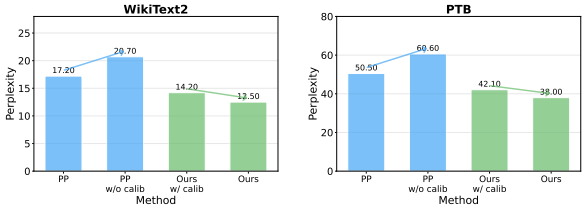
- **Wanda-sp** extends the unstructured pruning method Wanda (Sun et al., 2024b) to structured pruning by using the product of weight magnitudes and input feature norms as the importance metric.
- **FLAP** (An et al., 2024) introduces a fluctuation-based metric that measures the variance of output feature maps across samples. Channels with low fluctuation (i.e., high stability) are considered redundant and pruned.
- **LLM-Pruner** (Ma et al., 2023) uses second-order Taylor expansion to estimate the importance of structured channels, relying on pre-collected activation statistics of calibration data.
- **LoRAPrune** (Zhang et al., 2024) integrates pruning with Low-Rank Adaptation (LoRA) by using gradients and weights of LoRA adapters to estimate the importance of the model weights.
- **CFSP** (Wang et al., 2025c) employs a coarse-to-fine activation criterion to dynamically allocate sparsity budgets across blocks based on transformation saliency and prune internal dimensions using weight-activation products.
- **PP** (Le et al., 2025) employs lightweight probes to analyze hidden states and determine which channels contribute to the final prediction. It adopts a uniform sparsity ratio across all layers.

Implementation Details. Following the setting of prior work (Ma et al., 2023; Zhang et al., 2024), we keep the first three layers unchanged. Pruning ratios are averaged across attention and FFN blocks (layers 4-32): for 20%/40% target sparsity on LLaMA-2-7B, we prune 22%/44% respectively. All the baseline methods use the same sampled 2000 sequences (1024 tokens for text generation, 512 tokens for reasoning tasks) from the C4 subset (Raffel et al., 2020). We follow the official implementations and recommended hyperparameters of each baseline. For LLM-Pruner (Ma et al., 2023), we use 10 samples with 128 tokens to build importance metrics, followed by recovery retraining on 20,000 samples with 256 tokens each for 2 epochs using AdamW optimizer with learning rate 1×10^{-4}, batch size 64, and LoRA rank 8. For LoRAPrune (Zhang et al., 2024), we randomly sample 10,000 sequences of 512 tokens from C4 for training with AdamW optimizer (learning rate 1×10^{-4}, batch size 128, LoRA rank 8, 2 epochs).

For our OCP method, we implement the framework in PyTorch with the HuggingFace Transformers library (Wolf et al., 2020). For acceleration inference experiments, we use the DeepSpeed (Rasley et al., 2020) library to measure FLOPs and latency on a single NVIDIA A6000 GPU. By default, we set the following hyperparameters. The FFN sensitivity vectors $\boldsymbol{\omega}^{(l)}$ are precomputed offline once for each model and cached for runtime use. For attention probing, we set the decay rate $\alpha = 0.9$ to aggregate historical attention patterns. The adaptive sparsity allocation uses $\beta = 0.95$ for historical outlier density tracking, adjustment strength $\gamma = 0.1$. The clip boundaries are set to $[\rho_{\min}, \rho_{\max}] = [\rho_{\text{target}} - 0.1, \rho_{\text{target}} + 0.1]$ to allow 10% fluctuation around the target sparsity while preventing extreme allocations. For pruning metric, we use the PP scoring metric (PPsp) from (Le et al., 2025) as the default importance criterion for all the layers.

C Sensitivity to Calibration Data

Figure 10 reveals contrasting calibration dependencies between PP and OCP. PP relies on calibration statistics to compensate for inaccurate probe construction, and removing calibration causes over 20% perplexity degradation on both datasets. However, this dependency inevitably introduces distribution shift between calibration and test data. OCP's accurate outlier-aware probing eliminates



(a) WikiText2 (b) PTB

Figure 10: Effect of calibration data on LLaMA-2-7B at 40% sparsity. PP relies on calibration to compensate for inaccurate probing, suffering degradation when removed (blue). OCP's accurate probe construction eliminates calibration dependency, with performance degrading when calibration is added (green).

Table 6: Performance of pruning attention heads versus FFNs at different ratios on LLaMA-2-7B. We report WikiText2 perplexity (↓ shows improvement).

Attention	FFN	Overall	PP	**Ours**
0%	0%	0%	6.0	6.0
20%	0%	7%	6.8	**6.4** (↓5.9%)
0%	20%	13%	7.2	**6.6** (↓8.3%)
40%	0%	14%	10.0	**7.9** (↓21.0%)
0%	40%	25%	10.1	**8.8** (↓12.9%)
20%	40%	33%	11.9	**9.3** (↓21.8%)
40%	20%	27%	11.5	**8.9** (↓22.6%)
0%	60%	39%	21.1	**15.0** (↓28.9%)
60%	0%	21%	33.5	**15.6** (↓53.4%)
20%	60%	46%	23.8	**16.0** (↓32.8%)
60%	20%	34%	38.4	**16.8** (↓56.3%)
40%	60%	53%	33.5	**17.7** (↓47.2%)
60%	40%	47%	44.3	**18.2** (↓58.9%)

this dependency entirely: not only is calibration unnecessary, but adding it actually hurts performance due to distribution mismatch. This highlights that accurate probe construction removes a critical source of error in existing methods.

D Attention vs. FFN Pruning

Table 6 reveals that OCP's advantage over PP is more pronounced when pruning attention heads than FFNs. At 60% attention sparsity, OCP achieves 53.4%–58.9% relative improvement, compared to 12.9%–32.8% at 60% FFN sparsity. This asymmetry arises because attention heads exhibit concentrated outlier patterns that PP's energy-based selection misses, while our attention-accumulated probing effectively tracks salient tokens via aggregated attention scores. Furthermore, OCP demonstrates robustness across different attention/FFN pruning ratios, maintaining stable performance where PP degrades significantly under attention-heavy configurations.

E Generations From Pruned Models

We showcase the generation results of the dense and pruned models in Table 11, showing that OCP retains relatively high-quality text generation capabilities.

F Additional Evaluation Results

We provide detailed experimental results on LLaMA-2-13B (Table 8), LLaMA-3-8B (Table 9), OPT-13B (Table 10), and Qwen3-8B (Table 7). These results reinforce our main findings. OCP achieves the best perplexity across all model families, with the performance gap widening at higher sparsity (e.g., 54% reduction over PP on LLaMA-3-8B at 40% sparsity), confirming that accurate outlier identification becomes critical under aggressive compression.

Table 7: Performance of Qwen3-8B at 20% sparsity

Sparsity	**Method**	**ACC**	**PPL**
0%	Qwen3-8B (Base)	73.50	6.60
20%	Wanda-SP	67.85	7.65
20%	FLAP	66.90	7.90
20%	CFSP	68.20	7.85
20%	PP	70.60	7.15
20%	**OCP (ours)**	**71.85**	**6.88**

Table 8: Performance comparison on LLaMA-2-13B across datasets and ratios. The best is highlighted in bold.

Ratio	Method	WikiText2 ↓	PTB ↓	BoolQ ↑	PIQA ↑	HellaSwag ↑	WinoGrande ↑	ARC-c ↑	ARC-e ↑	OBQA ↑
0%	Dense	5.1 0.0	28.4 0.0	72.1 0.0	79.6 0.0	78.7 0.0	70.7 0.0	46.5 0.0	71.3 0.0	44.2 0.0
20%	Wanda-sp	9.0 0.0	47.3 0.0	70.4 1.0	79.4 0.0	78.4 0.0	70.2 0.1	44.3 0.6	69.9 0.3	42.5 0.2
	FLAP	7.5 0.1	36.7 0.2	71.1 0.5	78.7 0.1	77.3 0.1	**71.2** 0.2	44.6 0.1	66.7 0.1	42.5 0.2
	LoRAPrune	7.4 0.0	37.9 0.5	64.4 0.5	78.1 0.1	74.8 0.2	66.0 0.3	40.4 0.3	61.7 0.9	41.6 0.2
	LLM-Pruner	8.4 0.5	43.2 1.2	70.2 1.4	78.3 0.3	73.8 0.3	65.8 1.3	40.1 0.5	64.2 0.4	42.0 0.4
	CFSP	7.3 0.1	32.4 0.2	71.3 0.1	79.3 0.3	78.4 0.2	70.8 0.2	44.4 0.2	70.1 0.1	42.6 0.1
	PP	6.7 0.1	33.2 0.1	72.0 0.2	79.5 0.5	77.6 0.0	68.5 0.1	44.7 0.2	71.5 0.1	43.0 0.2
	Ours	**6.1** 0.0	**30.0** 0.1	**72.1** 0.0	**80.1** 0.1	**78.5** 0.0	70.1 0.2	**48.0** 0.1	**71.6** 0.0	**43.4** 0.0
40%	Wanda-sp	21.6 0.4	96.0 0.3	62.4 0.0	74.5 0.3	68.0 0.0	63.0 0.4	34.8 0.5	54.9 0.3	38.9 0.4
	FLAP	15.5 0.4	75.5 0.4	62.9 0.1	76.8 0.3	72.4 0.1	**66.9** 0.3	40.4 0.4	63.1 0.4	41.8 0.1
	LoRAPrune	11.1 0.3	64.1 2.8	62.5 0.1	74.1 0.4	65.5 0.1	60.4 0.3	33.0 0.4	53.9 0.7	39.3 0.6
	LLM-Pruner	15.3 0.7	76.8 3.4	63.9 0.4	73.5 0.6	62.4 1.4	57.5 1.1	33.2 1.2	55.2 0.7	37.5 0.8
	CFSP	15.6 0.2	81.8 0.3	62.3 0.2	75.2 0.3	70.6 0.2	68.3 0.0	36.3 0.2	52.9 0.2	40.0 0.3
	PP	11.3 0.1	61.0 0.1	65.8 0.1	77.1 0.2	71.6 0.0	61.3 0.4	40.9 0.3	67.9 0.1	42.5 0.3
	Ours	**9.9** 0.1	**51.1** 0.2	**67.1** 0.1	**77.9** 0.1	**75.1** 0.1	63.0 0.1	**45.6** 0.2	**70.7** 0.1	**43.2** 0.2

Table 9: Performance comparison on LLaMA-3-8B across datasets and ratios. The best is highlighted in bold.

Ratio	Method	WikiText2 ↓	PTB ↓	BoolQ ↑	PIQA ↑	HellaSwag ↑	WinoGrande ↑	ARC-c ↑	ARC-e ↑	OBQA ↑
0%	Dense	7.1 0.0	11.7 0.0	81.7 0.0	79.5 0.0	76.3 0.0	72.5 0.0	47.2 0.0	61.7 0.0	40.2 0.0
20%	Wanda-sp	16.2 0.0	20.8 0.0	75.1 0.3	78.5 0.0	69.6 0.2	71.4 0.4	38.7 0.4	56.9 0.4	39.0 0.2
	FLAP	18.6 0.0	24.9 0.0	**79.4** 0.2	78.7 0.1	70.3 0.0	71.4 0.5	40.8 0.1	57.8 0.0	39.4 0.3
	CFSP	16.1 0.1	20.2 0.2	79.0 0.1	78.8 0.2	70.2 0.1	71.3 0.3	39.8 0.0	57.0 0.1	38.9 0.1
	PP	12.1 0.0	17.0 0.0	77.4 0.0	78.5 0.0	73.1 0.0	72.5 0.3	43.2 0.3	59.1 0.2	40.2 0.5
	Ours	**10.2** 0.0	**15.0** 0.1	77.1 0.1	**78.9** 0.1	**74.7** 0.2	**72.8** 0.1	**45.1** 0.2	**62.0** 0.3	**41.2** 0.1
40%	Wanda-sp	56.9 0.0	93.1 0.0	66.6 0.1	73.4 0.2	56.7 0.1	63.2 0.2	31.8 0.2	47.0 0.5	34.5 0.2
	FLAP	86.2 0.0	133.1 0.0	67.3 1.0	73.5 0.0	57.2 0.2	66.7 0.5	31.7 0.3	44.6 0.3	34.4 0.3
	CFSP	54.8 0.1	89.2 0.6	66.6 0.2	73.4 0.3	56.7 0.3	66.3 0.1	31.8 0.1	46.3 0.2	34.5 0.1
	PP	44.6 0.0	42.8 0.0	70.3 0.1	76.3 0.2	65.3 0.1	67.2 0.2	39.0 0.3	57.4 0.1	36.9 0.3
	Ours	**20.4** 0.1	**31.4** 0.2	**71.0** 0.0	**76.6** 0.1	**66.2** 0.2	**68.6** 0.1	**40.4** 0.1	**58.9** 0.2	**39.6** 0.1

Table 10: Performance comparison on OPT-13B across datasets and ratios. The best is highlighted in bold.

Ratio	Method	WikiText2 ↓	PTB ↓	BoolQ ↑	PIQA ↑	HellaSwag ↑	WinoGrande ↑	ARC-c ↑	ARC-e ↑	OBQA ↑
0%	Dense	11.6 0.0	13.6 0.0	68.1 0.0	75.3 0.0	67.9 0.0	66.8 0.0	35.0 0.0	51.1 0.0	36.4 0.0
20%	Wanda-sp	17.4 0.1	30.9 0.0	66.0 0.2	75.4 0.1	63.0 0.1	64.8 0.3	33.7 0.0	48.2 0.2	35.0 0.1
	FLAP	18.8 0.2	23.5 0.0	68.1 0.4	75.1 0.1	62.5 0.2	62.6 0.3	31.8 0.3	49.5 0.1	34.5 0.1
	CFSP	19.0 0.0	29.4 0.3	66.2 0.1	75.2 0.1	63.2 0.2	63.9 0.2	33.3 0.1	49.2 0.0	34.5 0.2
	PP	14.7 0.1	15.6 0.0	67.4 0.1	75.5 0.1	65.7 0.0	64.9 0.3	33.8 0.1	51.6 0.0	36.5 0.2
	Ours	**13.5** 0.0	**14.3** 0.0	**68.2** 0.1	**75.7** 0.1	**66.8** 0.1	**65.3** 0.1	**34.2** 0.1	**51.8** 0.0	**37.0** 0.1
40%	Wanda-sp	42.7 0.7	61.7 0.0	63.7 0.1	71.8 0.3	53.2 0.1	57.6 0.2	29.6 0.4	43.3 0.1	34.3 0.2
	FLAP	51.0 0.7	65.4 0.0	62.7 0.0	72.4 0.0	53.3 0.2	58.3 0.5	29.4 0.3	45.2 0.4	34.1 0.1
	CFSP	53.4 0.2	61.9 0.8	63.1 0.0	72.5 0.2	53.3 0.1	58.4 0.2	29.5 0.1	44.8 0.0	34.2 0.1
	PP	26.7 0.3	26.3 0.0	61.1 0.2	74.3 0.1	58.7 0.0	59.3 0.1	33.6 0.1	49.7 0.1	35.3 0.4
	Ours	**20.5** 0.0	**22.2** 0.1	**64.0** 0.1	**74.5** 0.1	**62.5** 0.1	**62.0** 0.3	**33.9** 0.0	**50.8** 0.1	**35.6** 0.1

Table 11: Generation examples from pruned LLaMA models at 40% sparsity.

LLaMA-2-7B	
Prompt: The sun began to set behind the jagged mountains, casting long, purple shadows across the valley. Arthur stood at the edge of the cliff, holding the ancient, tattered map in his trembling hands. He looked down and saw	
Ours	the vast expanse of the desert before him. The wind tugged at his hair, and the sand blew through his robes. "Sir! It's time!" A young page appeared beside him, holding a torch. "Oh, right, my thanks." Arthur took the torch and held it high.
PP	the valley below, and the river that ran through it. He could see the village, and the castle, and the forest beyond. He could see the world. He could see the world, and he could see the world, and he could see the world. [repetitive]

LLaMA-3-8B	
Prompt: In a world where gravity works upwards, people have to tie themselves to the ground. I stepped out of my front door and forgot to clip my safety rope. Suddenly, I	
Ours	felt myself floating up and higher. I could see the Earth far below me. I tried to scream but my voice was gone. The clouds rushed past me as I ascended into the endless blue sky.
PP	was floating. I was floating up. I was going up. The sky was getting closer. I was scared. I was very scared. I tried to grab something. There was nothing to grab. [repetitive, simple]