

Kurtosis-Aware Coupled Sparsity: Training-Free Activation Sparsity for Large Language Models

Guangyan Li^{1,2*}, Luoyang Sun^{1,2*}, Siyuan Li³
Haifeng Zhang^{1,2†}, Yongqiang Tang^{1,2†}, Wensheng Zhang⁴

¹Institute of Automation, Chinese Academy of Sciences,

²University of Chinese Academy of Sciences,

³Shanghai Jiao Tong University, ⁴Guangzhou University

Correspondence: liguangyan2022@ia.ac.cn, sunluoyang2022@ia.ac.cn

Abstract

The massive scale of Large Language Models (LLMs) incurs prohibitive computational costs, limiting their ubiquitous deployment. While activation sparsity is a promising training-free solution, existing magnitude-based methods overlook weight structures and dependencies. We reveal that standard weight norms fail as importance proxies, identifying instead that weight distribution heavy-tailedness is critical. To this end, we propose **Kurtosis-Aware Coupled Sparsity** (KACS), a novel training-free framework that introduces a Kurtosis-based metric to explicitly capture and prioritize outlier-rich weight columns. Furthermore, we develop an Interaction-Aware evaluation mechanism that assesses the joint importance of structurally coupled projections (e.g., Gate-Up and Q-K-V), ensuring information retention across interacting pathways. In addition, an adaptive sparsity allocation strategy is employed to distribute the sparsity budget across layers and components. Extensive experiments on Llama and Mistral models demonstrate that KACS consistently outperforms state-of-the-art baselines, retaining over 97% of the original performance at 50% sparsity.

1 Introduction

Large Language Models (LLMs) have demonstrated exceptional capabilities across a wide spectrum of tasks (Wei et al., 2022). However, the massive scale of these models, often exceeding billions of parameters, results in significant computational costs and memory overheads that limit their practical deployment. To address these computational challenges, various model compression techniques have been extensively explored, such as quantization (Frantar et al., 2023; Lin et al., 2024) for reducing numerical precision, weight pruning (Han

et al., 2015; Ma et al., 2023) for removing redundant parameters, and low-rank approximation (Li et al., 2024; Wang et al., 2025b) for decomposing weight matrices. Distinct from these static compression methods, activation sparsity has emerged as a promising paradigm. This approach leverages the inherent sparsity of activations, where a significant portion of neurons contribute negligibly to the output. By skipping these redundant computations, activation sparsity substantially reduces floating-point operations (FLOPs) and improves inference speed without modifying the model parameters.

The transition to SwiGLU-based architectures in modern LLMs has effectively compromised the natural sparsity inherent in earlier ReLU models, rendering traditional zero-skipping techniques ineffective. While attempts have been made to regain sparsity via costly retraining, the research focus has largely shifted toward training-free approaches. Dominant methods in this category typically prioritize neurons solely based on their activation magnitude. However, this metric presents a critical blind spot: it evaluates input signals in isolation, neglecting both the weight structures that process them and the functional dependencies between coupled projections (e.g., Gate-Up and Q-K-V). Consequently, these methods risk pruning signals that are structurally critical due to downstream amplification or multiplicative interactions, leading to a significant saliency mismatch. Furthermore, existing pipelines either enforce a uniform ratio that ignores redundancy or resort to expensive search strategies, prompting a key challenge: how to adaptively distribute the budget without prohibitive overhead.

To surmount these obstacles, we introduce KACS, a novel training-free activation sparsity framework. Our investigation begins with a counter-intuitive finding, illustrated in Figure 1: contrary to prevailing assumptions, simple weight norms (e.g., ℓ_1 , ℓ_2) fail to serve as effective proxies for activation importance and can

* Equal Contribution.

† Corresponding author.

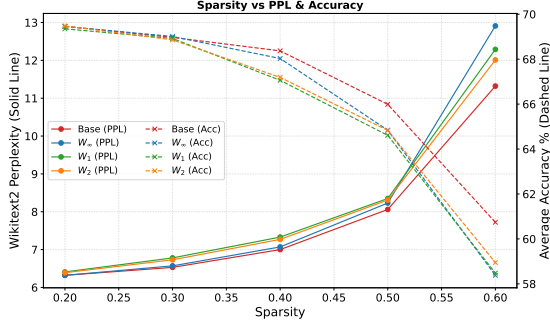


Figure 1: **Performance comparison of weight-norm scaling variants on LLaMA-3-8B.** “Base” refers to the activation magnitude baseline ($|X|$). W_1 , W_2 , and W_∞ denote scaling by the respective weight column norms. **Solid lines** indicate Perplexity ($\downarrow$), while **dashed lines** denote Accuracy ($\uparrow$). The results show that simple weight-norm scaling fails to yield improvement.

even degrade performance. Crucially, we identify that the determinative factor for importance is the presence of outliers. Consequently, we propose a Kurtosis-based metric that explicitly captures the heavy-tailedness of weight distributions. This metric effectively enables us to differentiate critical, outlier-rich weight columns from less informative ones. Simultaneously, we recognize that LLM components often function as coupled units. We introduce an Interaction-Aware Evaluation mechanism that moves beyond isolated weight analysis. This module groups functionally related projections (e.g., FFN Gate-Up pairs and Attention Q-K-V) to assess importance jointly, ensuring that retained activations are significant across these interacting paths. Finally, to address varying structural redundancy, we implement a practical, multi-level sparsity allocation strategy. By combining a lightweight layer-wise distribution based on input-output similarity with a reconstruction-aware optimization for intra-layer components, this approach achieves adaptive budget allocation with minimal computational overhead.

Contributions. Our contributions are as follows:

1. Kurtosis-based Weight Metric. We investigate the efficacy of standard weight norms (e.g., ℓ_1 , ℓ_2) in evaluating activation importance. We find that they fail to distinguish outlier-rich columns from others. To address this, we propose a Kurtosis-based metric that captures the heavy-tailedness distribution of weight columns, enabling the precise identification of outlier-rich weight columns that are critical for model performance.

2. Interaction-Aware Evaluation. Moving beyond the assessment of isolated weight matrices,

we introduce an Interaction-Aware framework that models the functional dependencies between coupled components. By jointly evaluating functionally related projections (e.g., Gate-Up in FFNs and Q-K-V in Attention), we ensure that sparsity decisions maximize information retention across interacting pathways.

3. Adaptive Sparsity Allocation. We employ a two-level allocation strategy that adapts to varying layer redundancy. By combining a similarity-based layer-wise mapping with a greedy intra-layer optimization, we achieve a flexible budget distribution that maintains high feature fidelity.

4. SOTA Performance. Across Llama-2, Llama-3, and Mistral, KACS maintains $> 97\%$ of dense performance at 50% sparsity, consistently outperforming SOTA baselines under a unified setting.

2 Related Work

Training-based Activation Sparsity. To mitigate the lack of natural sparsity in SwiGLU-based architectures, early approaches sought to restore sparsity through extensive retraining. **DejaVu** (Liu et al., 2023) introduces auxiliary predictors to dynamically forecast activation sparsity, while **ReLUfication** (Mirzadeh et al., 2023) explicitly replaces Swish activation functions with ReLU variants followed by fine-tuning. Despite their effectiveness, these methods impose significant training overhead and permanently alter the pre-trained model weights, limiting their practical utility.

Training-free Activation Sparsity. To enable plug-and-play efficiency, recent research prioritizes training-free approaches. The prevailing approach relies on activation magnitude as the sole metric for importance. Initial approaches like **CATS** (Lee et al., 2024) and **GRIFFIN** (Dong et al., 2024) restrict their scope to the FFN blocks, applying activation sparsity only to specific projections (e.g., Gate or Up layers). Subsequently, **TEAL** (Liu et al., 2025) generalizes this paradigm, extending the sparsity mechanism to target all weight matrices across the model. However, these methods typically evaluate input signals in isolation, neglecting the weight matrices that process them. To address this, subsequent methods explore two distinct directions. **R-Sparse** (Zhang et al., 2025) creates a separate computational branch, where high-magnitude outliers are computed exactly, while low-magnitude components are approximated via a low-rank SVD path. Alternatively, other ap-

proaches integrate weight information directly into the importance evaluation. **WAS** (Wang et al., 2025a) explicitly incorporates weight sensitivity by scaling activation scores with the ℓ_1 norm of the corresponding weight columns. **Amber** (An et al., 2025) adopts a similar philosophy by employing a variant of ℓ_2 column norms. Consequently, existing weight-aware methods face a dilemma: they either incur significant computational overhead (e.g., dual-branch execution costs) or yield suboptimal performance preservation due to the limited expressiveness of standard norm-based metrics.

3 Background

3.1 Transformer Model

The Transformer architecture stacks multiple layers consisting of a Multi-Head Attention (MHA) and a Feed-Forward Network (FFN). Given the input $\mathbf{X}$, the MHA sub-layer projects the input $\mathbf{X}$ into queries, keys, and values:

$$\begin{aligned} \text{head}_i &= \text{Att}(\mathbf{X}\mathbf{W}_{q_i}, \mathbf{X}\mathbf{W}_{k_i}, \mathbf{X}\mathbf{W}_{v_i}). \\ \text{MHA}(\mathbf{X}) &= \text{Concat}(\text{head}_1, \dots, \text{head}_h)\mathbf{W}_o, \end{aligned} \quad (1)$$

Here, $\mathbf{W}_{q_i}$, $\mathbf{W}_{k_i}$, $\mathbf{W}_{v_i}$ denote the projection matrices for the i -th head, and $\mathbf{W}_o$ is the output projection. The FFN sub-layer usually adopts the SwiGLU with three linear transformations:

$$\text{FFN}(\mathbf{X}) = (\sigma(\mathbf{X}\mathbf{W}_{\text{gate}}) \odot (\mathbf{X}\mathbf{W}_{\text{up}}))\mathbf{W}_{\text{down}}, \quad (2)$$

where $\sigma(\cdot)$ is the SiLU activation function, and $\odot$ denotes the Hadamard product. Here, $\mathbf{W}_{\text{gate}}$, $\mathbf{W}_{\text{up}}$, and $\mathbf{W}_{\text{down}}$ denote the weight matrices for the gate, up, and down projections. For simplicity of subsequent analysis, we formulate the generic linear transformation in the model as $\mathbf{Y} = \mathbf{W}\mathbf{X}^T$. Here, $\mathbf{X} \in \mathbb{R}^{L \times d_{in}}$ represents the input activations with sequence length L and input dimension d_{in} , while $\mathbf{W} \in \mathbb{R}^{d_{out} \times d_{in}}$ denotes the weight matrix.

3.2 Activation Sparsity

In Transformer models, the computational cost is dominated by linear transformations. To achieve computational acceleration during inference, activation sparsity is typically implemented by selectively dropping unimportant input activation values. The most common approach defines the importance score S_{ij} for an activation element X_{ij} (corresponding to the j -th channel of the i -th token) simply as its absolute magnitude: $S_{ij} = |X_{ij}|$. Given a target sparsity level p , a threshold t_p is first established.

In practice, t_p is calculated empirically by analyzing the activation distribution obtained offline from a calibration dataset. This threshold serves as a filter: if $S_{ij} \leq t_p$, the corresponding activation X_{ij} is zeroed out. Consequently, the forward pass is executed by retaining only the input elements where $S_{ij} > t_p$. This mechanism significantly reduces the computational overhead, particularly the Multiply-Accumulate (MAC) operations.

3.3 Weight Pruning and Activation Sparsity

Weight pruning methods directly sparsify model parameters, producing static and input-independent sparse weight matrices. Wanda (Sun et al., 2024) scores individual weights using their magnitudes and the norms of the corresponding input activations, while SparseGPT (Frantar and Alistarh, 2023) uses approximate second-order information to perform one-shot weight pruning. Building on this paradigm, OWL (Yin et al., 2024) assigns non-uniform layer-wise sparsity ratios according to activation outlier distributions, whereas AlphaPruning (Lu et al., 2024) uses heavy-tailed self-regularization theory to guide layer-wise pruning. In contrast, activation sparsity retains the original dense weights and dynamically skips input activations during inference, yielding token-dependent sparsity patterns.

3.4 Activation-Aware Weight Importance

Previous works, such as Wanda (Sun et al., 2024), have demonstrated the significant coupling between weights and activations. Wanda integrates activations into the weight importance evaluation to guide the pruning process. Specifically, it defines the importance score $\mathcal{I}_{ij}$ for each weight element W_{ij} as the product of its magnitude and the ℓ_2 -norm of the corresponding input feature channel $\mathbf{X}_{:,j}$:

$$\mathcal{I}_{ij} = |W_{ij}| \cdot \|\mathbf{X}_{:,j}\|_2. \quad (3)$$

4 Method

We present a training-free framework (Figure 2), which comprises three key components: Kurtosis-Based metric, Interaction-Aware Importance metric, and Layer-Wise Sparsity Allocation strategy.

4.1 Motivation

Inspired by Wanda (Sun et al., 2024), we investigate from the perspective of activation sparsity whether static weight norms can serve as effective proxies for assessing activation importance.

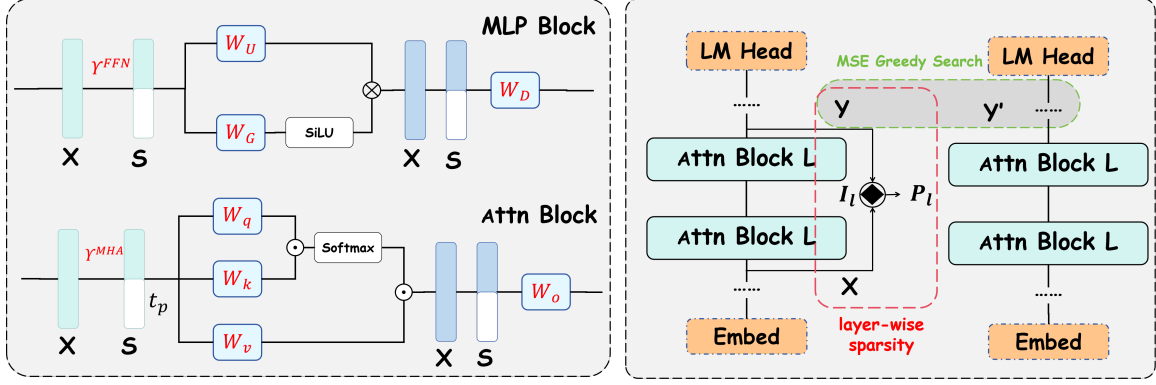


Figure 2: **Overview of the proposed method.** **Left:** The importance metric for precise activation sparsity in FFN and MHA layers. **Right:** The sparsity allocation strategy that dynamically assigns sparsity budgets.

Intuitively, weights with larger magnitudes are expected to contribute more to the output, suggesting that their corresponding input channels should be prioritized. Based on this duality, we propose to incorporate the column norm of the weight matrix into the activation importance estimation. Formally, for the activation element X_{ij} , the importance score S_{ij} is defined as:

$$S_{ij} = \|\mathbf{W}_{:,j}\|_p \cdot |X_{ij}|, \quad p \in \{1, 2, \infty\}. \quad (4)$$

To assess the validity of this hypothesis, we performed a preliminary empirical study benchmarking these variants against the standard activation magnitude baseline ($|\mathbf{X}|$) on LLaMA-3-8B. As shown in Figure 1, although integrating weight information appears intuitively promising, we observe that the direct application of column norms fails to yield any improvement and consistently leads to performance degradation. This suggests that standard norms are insufficient for capturing the complex distribution of weights, highlighting the necessity for a more sensitive metric.

4.2 Kurtosis Metric

We hypothesize that the ineffectiveness of standard norms stems from their inability to distinguish outlier-rich channels from trivial channels with flat distributions. To validate this, we visualize the ℓ_2 norm scores in Figure 3, where the standard metric (blue line) exhibits a flat, low-variance baseline across channel indices. This uniformity implies that the ℓ_2 norm assigns similar scores to all channels, thereby failing to distinguish critical signals from the trivial background.

To address this limitation and explicitly capture the structural importance of weights, we introduce a **Kurtosis-based Metric**. Our approach is motivated by the statistical insight that Kurtosis effec-

tively captures the outlier-driven nature of weight distributions—a phenomenon recently characterized as a decisive factor in determining quantization robustness (Egiazarian et al., 2025). While that work focuses on mitigating high Kurtosis to improve quantization grid alignment, we take a different perspective by proactively leveraging this higher-order statistic as a discriminative tool to isolate salient channels for sparsity. Specifically, for each input channel j , we compute the empirical Excess Kurtosis κ_j of the weight column $\mathbf{W}_{:,j}$:

$$\kappa_j = \mathbb{E} \left[\left(\frac{\mathbf{W}_{:,j} - \mu_j}{\sigma_j} \right)^4 \right] - 3, \quad (5)$$

where μ_j and σ_j denote the mean and standard deviation of the elements in $\mathbf{W}_{:,j}$. Based on this, we derive the channel-wise scaling factor γ_j :

$$\gamma_j = 1 + \alpha \cdot \log(\text{ReLU}(\kappa_j) + 1). \quad (6)$$

Here, α is a hyperparameter controlling the sensitivity. The ReLU function acts as a filter to exclusively target **outlier-rich channels** (where $\kappa_j > 0$), while the logarithmic term compresses the dynamic range of extreme outliers. The constant 1 ensures $\gamma_j \geq 1$, thereby selectively amplifying outlier-rich channels without diminishing the baseline importance. Consequently, the importance score S_{ij} is obtained by modulating the corresponding activation magnitude with this kurtosis scaling factor:

$$S_{ij} = |X_{ij}| \cdot \gamma_j. \quad (7)$$

The advantage of this proposed metric is clearly demonstrated in Figure 3. In stark contrast to the uniform distribution of the ℓ_2 norm, the Kurtosis metric (orange line) exhibits superior discriminative power. It reveals distinct, sharp peaks that precisely align with outlier-rich channels (marked with

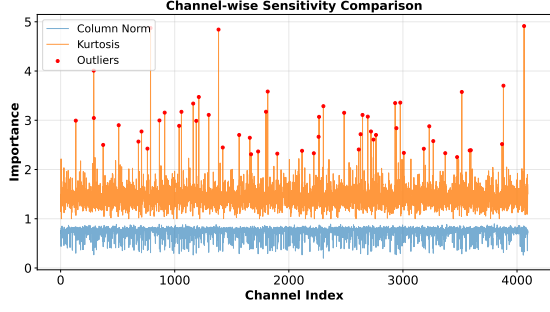


Figure 3: **Metric Comparison.** The column norm exhibits a flat distribution, failing to distinguish critical channels. In contrast, the Kurtosis demonstrates superior discriminative power, revealing sharp peaks that precisely capture outlier-rich channels (marked by **red dots**, identified via the 3σ rule, i.e., values $> \mu + 3\sigma$).

red dots). This confirms that our formulation successfully distinguishes critical signals and assigns high importance scores specifically to the channels that contain essential information. A comparative study against alternative statistical metrics and an extensive ablation analysis on outlier importance are detailed in Appendix D.

4.3 Interaction-Aware Importance Metric

While recent advances have begun to incorporate weight sensitivity, they often treat weight matrices as isolated entities, neglecting the functional dependencies between them. However, given the intrinsic coupling of model operations, the effective importance of an activation should be governed by the collaborative impact of its interconnected components. To address this, we shift from independent metrics to an interaction-aware evaluation. In the following, we first formulate this metric for FFN sublayers, and subsequently extend it to MHA sublayers.

As formulated in Eq. (2), the generation of the intermediate feature is governed by the element-wise multiplication ($\odot$) between the Up and Gate projections. This multiplicative structure creates an intrinsic coupling: a large activation value in the up branch is effectively nullified if the corresponding gate value is near zero, and vice versa. Consequently, evaluating $\mathbf{W}_{\text{up}}$ or $\mathbf{W}_{\text{gate}}$ in isolation fails to capture this mutual suppression effect.

We posit that the saliency of an input activation is defined by the synergistic coupling of its corresponding gate and up branches. Accordingly, we propose an **Interaction-Aware Scaling Factor**

(γ^{FFN}) specifically for the FFN input activations:

$$\begin{aligned}\gamma^{\text{FFN}} &= \gamma^{\text{gate}} \odot \gamma^{\text{up}}, \\ S_{ij} &= |X_{ij}| \cdot \gamma_j^{\text{FFN}}.\end{aligned}\quad (8)$$

where $\gamma^{\text{gate}}, \gamma^{\text{up}} \in \mathbb{R}^{d_{\text{in}}}$ denote the importance vectors (constructed as $\gamma = [\gamma_1, \dots, \gamma_{d_{\text{in}}}]$) derived from the weight matrices of Gate and Up projections. Mechanistically, this operates as a dual-verification filter, which enforces a consensus constraint: it preserves channels only if they are salient in both projections, thereby effectively suppressing inconsistent signals.

We extend this interaction-aware paradigm to MHA layers. Although MHA lacks the explicit gating structure of SwiGLU, the dot-product attention mechanism operates as an implicit routing gate. Specifically, the interaction between the Query and Key projections generates an attention map that functions as a dynamic filter, determining the information flow from the Value projection to the output. This establishes a critical dependency, where the Q-K pair governs where to focus, while the V projection determines what to transmit. Consequently, evaluating any single matrix in isolation fails to capture this systemic dependency. Based on this synergy, we define the Interaction-Aware Scaling Factor γ^{MHA} for MHA input activations as:

$$\begin{aligned}\gamma^{\text{MHA}} &= \gamma^{\text{Q}} \odot \gamma^{\text{K}} \odot \gamma^{\text{V}}, \\ S_{ij} &= |X_{ij}| \cdot \gamma_j^{\text{MHA}}.\end{aligned}\quad (9)$$

where $\gamma^{\text{Q}}, \gamma^{\text{K}}, \gamma^{\text{V}}$ denote the importance vectors derived from the Query, Key, and Value weights via Eq. (6), respectively. By enforcing this strict consensus, we ensure that retained activations contribute significantly to both the addressing (Q, K) and the content (V) of the attention mechanism.

Reconstruction Analysis. To empirically validate the efficacy of these formulations, we conduct a layer-wise reconstruction analysis on LLaMA-3-8B. We evaluate the fidelity of the sparse representations using the Relative Reconstruction Error ($\mathcal{E}$), defined via the Frobenius norm $\|\cdot\|_F$. This metric measures the normalized deviation between the original dense output $\mathbf{Y}$ and the sparse approximation $\hat{\mathbf{Y}}$ (at 50% sparsity):

$$\mathcal{E} = \|\mathbf{Y} - \hat{\mathbf{Y}}\|_F^2 / \|\mathbf{Y}\|_F^2. \quad (10)$$

As illustrated in Figure 4, the Weight ℓ_2 Norm yields the highest error, performing even worse

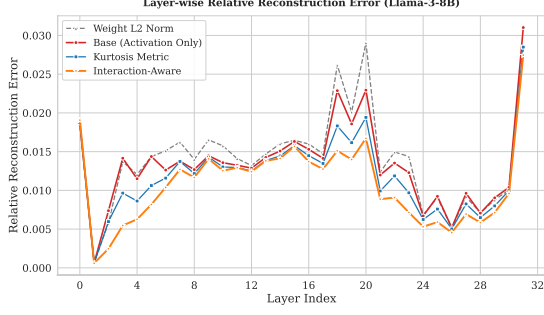


Figure 4: Layer-wise Relative Reconstruction Error (LLaMA-3-8B, 50% Sparsity). The Weight ℓ_2 Norm yields the highest error. The proposed **Kurtosis** metric significantly reduces the error, while the **Interaction-Aware** strategy consistently achieves the lowest error, validating the effectiveness of our method.

than the naive activation magnitude baseline, which aligns with the observations in Figure 1. In contrast, substituting the standard norm with our proposed Kurtosis metric leads to a significant error reduction. Building upon this, our complete Interaction-Aware strategy consistently achieves the lowest reconstruction error across layers. This confirms that both the sensitivity to weight outliers (via Kurtosis) and the synergistic evaluation of coupled weights (via Interaction-Awareness) are essential for minimizing information loss. More detailed results and further analyses are provided in Appendix B.

4.4 Adaptive Sparsity Allocation

To practically distribute the sparsity budget, we employ a hierarchical allocation strategy that considers both inter-layer redundancy and intra-layer reconstruction error.

Layer-wise Distribution. Following observations in layer pruning, we use the cosine similarity between layer inputs $\mathbf{X}$ and outputs $\mathbf{Y}$ as a redundancy metric R_n . A higher R_n suggests a near-identity mapping, indicating that the layer is more redundant and can tolerate higher sparsity. To maintain a global target sparsity p_t , we distribute the sparsity budget p_n for the n -th layer using a standardized linear mapping:

$$p_n = p_t + \beta \cdot \frac{R_n - \mu}{\sigma}, \quad (11)$$

where μ and σ are the mean and standard deviation of similarity scores across all N layers, and β is a scaling factor controlling the allocation variance. This formulation ensures that the expected sparsity across the model remains $\mathbb{E}[p] = p_t$, while effectively preserving critical layers with low similarity

scores. This approach provides an efficient alternative to expensive search-based allocation methods.

Intra-layer Optimization. Given the determined layer-wise sparsity p_n , the sparsity configuration is further optimized across the linear projections, specifically the attention matrices ($\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$) and MLP weights ($\mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{up}}, \mathbf{W}_{\text{down}}$). Following the methodology in TEAL (Liu et al., 2025), we formulate this intra-layer allocation as a constrained optimization problem, solved via a greedy search algorithm. This optimization minimizes the reconstruction error between the dense and sparsified layer outputs, ensuring maximal feature fidelity while strictly adhering to the assigned budget p_n . Further implementation details are deferred to Appendix C.1.

5 Experiments

5.1 Experimental Settings

Models. We evaluate our method on the Llama-3-8B (Grattafiori et al., 2024), Llama-2-7B (Touvron et al., 2023), and Mistral-7B (Jiang et al., 2023) models. We additionally evaluate Llama-3-70B to verify scalability; the results are summarized in Table 4 and detailed in Appendix A.

Calibration Data and Evaluation. Following TEAL, we use a calibration set sampled from the Alpaca dataset (Taori et al., 2023), consisting of 10 samples with a sequence length of 2048 tokens. We assess model performance across two categories: language modeling and zero-shot downstream tasks. For language modeling, we report the perplexity (PPL) on WikiText-2 (2048 tokens). For downstream capabilities, we report the average accuracy across eight tasks: Winogrande (WG) (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), SciQ (Welbl et al., 2017), OpenBookQA (OBQA) (Mihaylov et al., 2018), HellaSwag (HS) (Zellers et al., 2019), BoolQ (Clark et al., 2019), ARC-E, and ARC-C (Clark et al., 2018). All evaluations are conducted using the lm-eval-harness framework (Gao et al., 2024).

Baselines. We benchmark against: **TEAL** (Liu et al., 2025) (activation magnitude-based), **R-Sparse** (Zhang et al., 2025) (hybrid with SVD), **WAS** (ℓ_1 weight column norm), **Amber Pruner** (An et al., 2025) (a variant of the ℓ_2 weight column norm), and **WANDA-S** (a symmetric adaptation of WANDA). We exclude approaches with

Sparsity	Method	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
Llama-3-8B											
0%	Baseline	6.14	72.77	79.71	96.4	34.8	60.19	81.31	80.13	50.51	69.48
50%	TEAL	8.06	68.90	76.17	96.00	31.20	55.20	77.80	72.20	43.26	65.09
	R-Sparse	8.71	68.77	75.84	96.00	31.60	53.97	78.47	75.76	42.24	65.33
	WAS	8.35	68.59	75.79	95.80	31.00	53.80	74.04	75.42	42.41	64.61
	Amber Pruner	8.35	69.14	76.39	95.50	31.00	54.15	74.50	75.55	42.41	64.83
	WANDA-S	8.31	68.19	75.90	95.60	31.00	53.87	75.05	76.09	42.92	64.83
	KACS	7.45	69.06	77.53	96.40	31.20	56.18	76.76	76.47	45.48	66.14
	TEAL*	7.37	68.90	77.04	95.30	32.20	57.32	78.46	78.58	46.59	66.80
	KACS*	7.27	71.82	78.62	95.50	33.00	57.33	78.65	77.65	46.67	67.41
70%	TEAL	58.43	51.54	62.68	84.30	19.40	30.65	59.88	53.79	22.44	48.09
	R-Sparse	85.89	52.17	59.36	83.00	15.60	29.57	61.25	48.06	20.81	46.23
	WAS	74.35	52.33	59.03	76.30	17.60	28.33	60.95	42.38	18.43	44.42
	Amber Pruner	65.11	51.93	61.81	78.30	17.20	29.53	60.34	50.25	22.18	46.44
	WANDA-S	67.52	53.35	61.26	79.20	17.20	29.25	61.19	47.90	20.65	46.25
	KACS	28.80	55.09	67.08	85.40	19.40	35.44	61.80	61.66	26.54	51.55
	TEAL*	14.53	60.30	71.65	92.50	24.80	45.70	67.34	66.88	33.96	57.89
	KACS*	13.69	63.14	72.25	91.20	25.60	46.71	68.99	68.39	35.07	58.92
Mistral-7B											
0%	Baseline	5.32	73.88	80.25	95.80	33.00	60.88	82.14	79.67	48.98	69.33
50%	TEAL	7.10	68.74	79.33	96.40	29.00	58.52	81.33	77.82	46.50	67.21
	R-Sparse	6.75	68.98	78.99	95.90	29.20	57.17	80.06	76.73	44.03	66.38
	WAS	6.91	67.64	79.49	95.50	26.00	58.48	80.89	77.31	45.90	66.40
	Amber Pruner	6.91	67.17	79.16	95.60	26.80	58.47	80.86	78.28	45.56	66.49
	WANDA-S	6.69	66.54	79.54	96.20	27.00	58.66	81.25	77.69	45.73	66.58
	KACS	6.80	68.03	79.43	95.70	30.40	58.62	81.71	78.28	46.84	67.38
	TEAL*	7.72	68.98	79.11	95.10	29.20	59.36	81.99	78.75	46.59	67.39
	KACS*	6.79	70.17	79.43	95.30	28.40	59.94	81.93	79.21	47.87	67.78
70%	TEAL	13.36	60.85	74.10	89.50	20.80	45.78	68.65	68.64	33.36	57.71
	R-Sparse	18.47	59.16	67.63	90.10	23.00	43.76	64.48	62.44	32.75	55.42
	WAS	18.36	61.72	69.75	89.10	21.00	41.98	67.09	64.44	30.55	55.70
	Amber Pruner	17.00	60.93	67.68	91.00	22.00	42.26	67.61	63.38	33.70	56.07
	WANDA-S	18.23	60.38	69.48	89.60	22.60	42.17	67.77	62.04	32.85	55.86
	KACS	13.20	62.67	73.34	91.10	23.80	45.63	68.35	67.68	33.36	58.24
	TEAL*	13.44	65.11	70.13	85.40	28.40	49.56	74.50	68.35	39.16	60.08
	KACS*	13.56	62.19	76.88	85.50	22.60	52.54	75.93	73.06	40.87	61.20

Table 1: Evaluation across Llama-3-8B and Mistral-7B at 50% and 70% sparsity levels, showing that KACS consistently outperforms baselines. Note that ‘*’ implies the use of sparsity allocation.

suboptimal performance or limited scope (e.g., Re-lutification, CATS, GRIFFIN) from the main comparison. Methods marked with * adopt our sparsity allocation strategy for fair comparison.

5.2 Main Results

Table 1 presents comprehensive results across both Llama-3-8B and Mistral-7B. To ensure a fair comparison protocol, methods marked with ‘*’ utilize the same layer-wise sparsity allocation as KACS (redundancy-based allocation plus intra-layer MSE Greedy). At the standard 50% sparsity level, KACS* demonstrates exceptional capability, achieving average scores of **67.41** and **67.78**, respectively. These results closely approach the

dense baselines (**69.48** and **69.33**), retaining approximately 97% of the original performance. Furthermore, in the uniform sparsity regime, a clear hierarchy emerges: while TEAL sets a strong baseline, our approach consistently delivers superior average performance. This establishes KACS as a robust foundation even without relying on sparsity allocation strategies.

The advantage of KACS* becomes even more pronounced when compared against the state-of-the-art TEAL*. At 50% sparsity, our method not only surpasses TEAL* on Llama-3-8B accuracy (**67.41** vs. **66.80**) but also maintains superior linguistic coherence on Mistral-7B (PPL **6.79** vs. **7.72**). Crucially, in the aggressive 70% sparsity

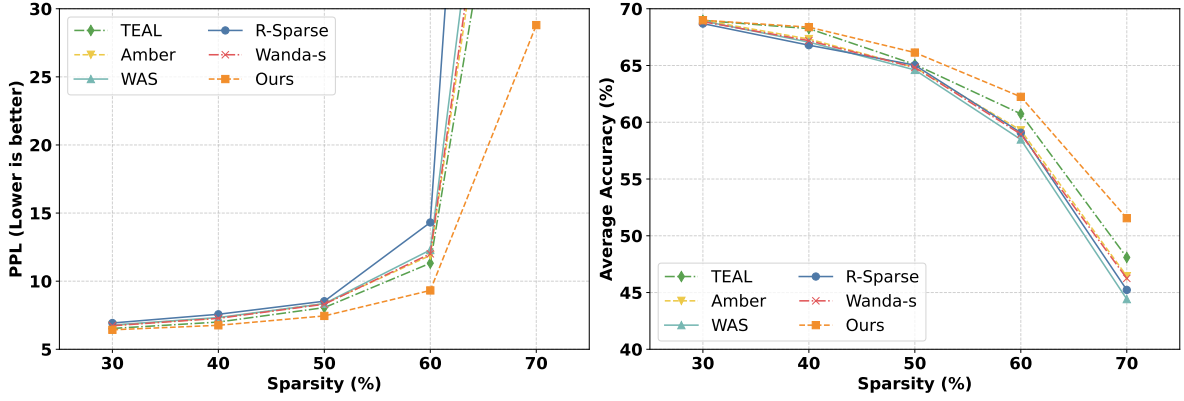


Figure 5: Comparative performance of different methods across multiple sparsity ratios.

regime, our method exhibits remarkable resilience. While baselines suffer severe degradation, KACS* remains robust on Mistral-7B with an average score of **61.20**, outperforming TEAL* by **1.12 points**. These results indicate that KACS* effectively identifies and protects critical activations, establishing a new state-of-the-art for efficient sparse models.

5.3 Performance Across Sparsity Levels

As illustrated in Figure 5, we compare the performance of our method against TEAL, R-Sparse, Amber, WAS, and WANDA-S on the Llama-3-8B across various sparsity ratios. Up to 50% sparsity, all methods demonstrate competitive performance, maintaining low Perplexity (PPL) and high average accuracy with comparable results. However, a distinct performance divergence emerges once the sparsity exceeds this 50% threshold.

Notably, as sparsity increases to 60% and 70%, existing baseline methods suffer from severe degradation. Their PPL scores rise sharply, and average accuracy precipitously drops to the 45%–48% range at 70% sparsity. In stark contrast, KACS exhibits superior robustness. Even at the extreme sparsity of 70%, the increase in PPL for our method remains relatively graceful, and average accuracy is maintained above 51%, significantly outperforming comparative methods.

5.4 Ablation Study

To investigate the individual contributions of our proposed components under extreme compression, we conduct an ablation study at 70% sparsity on Llama-3-8B, with results shown in Table 2. The baseline model struggles significantly, with a high PPL of 58.43. Incorporating the Kurtosis Metric alone yields a substantial improvement (PPL 48.61), while adding Interaction-Aware achieves a marked gain, reducing PPL to 28.80.

Method	PPL ↓	Avg
Baseline (TEAL)	58.43	48.09
+ Kurtosis Metric	48.61	50.71
+ Interaction-Aware	28.80	51.55
+ Layer-wise Sparsity	17.96	55.21
+ MSE Greedy	13.69	58.92

Table 2: Ablation study on Llama-3-8B at 70% sparsity. The results demonstrate the cumulative benefits of sequentially integrating our proposed components.

Most notably, the application of our Layer-wise Sparsity strategy plays a pivotal role, lifting the average accuracy to 55.21% and further reducing PPL to 17.96. Finally, applying the standard MSE Greedy adjustment as a complementary refinement yields the optimal performance with an accuracy of 58.92%. These results confirm that our proposed modules provide the fundamental robustness required at high sparsity, which can be further refined by established greedy methods. We additionally compare Kurtosis against a range of alternative statistical metrics in Appendix D, where it consistently outperforms all alternatives, further justifying its selection as the importance proxy.

5.5 Hardware Acceleration

For practical deployment, our method is fully compatible with the inference paradigm established by TEAL. While we employ a more sophisticated metric to evaluate activation importance, the derivation of the scaling factors and thresholds is performed entirely offline. These pre-computed thresholds are stored and loaded during inference. Furthermore, we can employ weight re-parameterization by absorbing the scaling factors directly into the model weights. This allows our method to function as mathematically equivalent to the TEAL without requiring any modification to the inference kernels or thresholding logic. Thus, our method can seamlessly leverage TEAL’s custom GPU kernels. By

utilizing these highly optimized CUDA kernels on an A800 GPU, our method achieves significant end-to-end acceleration comparable to the baseline.

For Llama3-8B, we achieve $1.39\times$, $1.49\times$, and $1.69\times$ speedups at 40%, 50%, and 70% sparsity levels, respectively. This demonstrates that our method achieves practical acceleration in decoding-dominant workloads. More detailed results and comparisons are provided in Appendix G.

6 Conclusion

We introduce KACS, a training-free framework that transcends simple magnitude-based activation sparsity by incorporating Kurtosis-based metric and Interaction-Aware mechanisms. These components accurately evaluate activation sensitivity and address the structural coupling within the model to preserve critical signals, further enhanced by a dynamic, similarity-guided sparsity allocation strategy. Crucially, our method aligns with the TEAL execution paradigm, achieving comparable speedups with equivalent hardware optimizations. Extensive experiments across the Llama and Mistral models demonstrate that our approach outperforms SOTA methods in performance, offering a promising solution for efficient LLM inference.

7 Acknowledgment

The authors are thankful for the financial support by the National Natural Science Foundation of China (62476274, U22B2048).

References

- Tai An, Ruwu Cai, Yanzhe Zhang, Yang Liu, Hao Chen, Pengcheng Xie, Sheng Chang, Yiwu Yao, and Gongyi Wang. 2025. [Amber pruner: Leveraging n:m activation sparsity for efficient prefill in large language models](#). *Preprint*, arXiv:2508.02128.
- Yonatan Bisk, Rowan Zellers, Ronan Le bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proc. AAAI*, pages 7432–7439.
- 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 *Proc. 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. *arXiv preprint arXiv:1803.05457*.
- Harry Dong, Beidi Chen, and Yuejie Chi. 2024. Prompt-prompted adaptive structured pruning for efficient LLM generation. *arXiv preprint arXiv:2404.01365*.
- Vage Egiazarian, Roberto L. Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2025. Bridging the gap between promise and performance for microscaling FP4 quantization. *arXiv preprint arXiv:2509.23202*.
- Elias Frantar and Dan Alistarh. 2023. SparseGPT: Massive language models can be accurately pruned in one-shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.
- Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *Proc. ICLR*.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, and 5 others. 2024. [The language model evaluation harness](#).
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.
- Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for

- efficient neural network. *Advances in neural information processing systems*, 28.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.
- Donghyun Lee, Je-Yong Lee, Genghan Zhang, Mo Tiwari, and Azalia Mirhoseini. 2024. Cats: Contextually-aware thresholding for sparsity in large language models. *arXiv preprint arXiv:2404.08763*.
- Guangyan Li, Yongqiang Tang, and Wensheng Zhang. 2024. Lorap: Transformer sub-layers deserve differentiated structured compression for large language models. In *Proc. ICML*.
- Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proc. MLSys*.
- James Liu, Pragaash Ponnusamy, Tianle Cai, Han Guo, Yoon Kim, and Ben Athiwaratkun. 2025. Training-free activation sparsity in large language models. In *International Conference on Learning Representations*.
- Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023. Deja vu: Contextual sparsity for efficient llms at inference time. In *Proceedings of the International Conference on Machine Learning*. PMLR.
- Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael Mahoney, and Yaoqing Yang. 2024. AlphaPruning: Using heavy-tailed self regularization theory for improved layer-wise pruning of large language models. In *Advances in Neural Information Processing Systems*, volume 37.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. In *Proc. NeurIPS*.
- 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 *Proc. EMNLP*.
- Iman Mirzadeh, Keivan Alizadeh, Sachin Mehta, Carlo C Del Mundo, Oncel Tuzel, Golnoosh Samei, Mohammad Rastegari, and Mehrdad Farajtabar. 2023. Relu strikes back: Exploiting activation sparsity in large language models. *arXiv preprint arXiv:2310.04564*.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. Winogrande: An adversarial winograd schema challenge at scale. In *Proc. AAAI*, pages 8732–8740.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*.
- Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B Hashimoto. 2023. [Stanford alpaca: An instruction-following LLaMA model](#).
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.
- Ming Wang, Miao Zhang, Xuebo Liu, and Liqiang Nie. 2025a. [Weight-aware activation sparsity with constrained bayesian optimization scheduling for large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1086–1098. Association for Computational Linguistics.
- Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. 2025b. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *International Conference on Learning Representations*.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, and 1 others. 2022. Emergent abilities of large language models. *Transactions on Machine Learning Research*.
- Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. [Crowdsourcing multiple choice science questions](#). In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106. Association for Computational Linguistics.
- Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay 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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 57101–57115. PMLR.
- Chong You, Kan Wu, Zhipeng Jia, Lin Chen, Srinadh Bhojanapalli, Jiaxian Guo, Utku Evci, Jan Wassenberg, Praneeth Netrapalli, Jeremiah J. Willcock, Suvinay Subramanian, Felix Chern, Shreya Pathak, Felix Yu, Prateek Jain, David E. Culler, Henry M. Levy, and Sanjiv Kumar. 2025. [Spark transformer: Reactivating sparsity in transformer FFN and](#)

[attention](#). In *Advances in Neural Information Processing Systems*, volume 38.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proc. ACL*.

Zhenyu Zhang, Zechun Liu, Yuandong Tian, Harshit Khaitan, Zhangyang Wang, and Steven Li. 2025. R-sparse: Rank-aware activation sparsity for efficient llm inference. In *The Thirteenth International Conference on Learning Representations*.

A More Detailed Experimental Results

We present comprehensive, per-benchmark experimental results on Llama-2-7B and Llama-2-13B in Table 3 to further substantiate the efficacy of our proposed method. While most approaches perform comparably at 50% sparsity, a significant divergence in robustness emerges at the aggressive 70% level. Baselines such as R-Sparse and WAS suffer catastrophic degradation (e.g., PPL > 120 on Llama-2-7B), whereas KACS degrades gracefully, outperforming TEAL by over 1% in average accuracy on Llama-2-13B at this sparsity level. Furthermore, incorporating adaptive sparsity allocation (denoted by ‘*’) consistently achieves the

best performance. Notably, KACS* attains 62.50% average accuracy on Llama-2-13B at 70% sparsity, surpassing TEAL* (61.33%) and significantly closing the gap with the dense baseline.

To further substantiate the scalability of KACS, we conduct additional experiments on Llama-3-70B, a model substantially larger than those evaluated in the main paper. As shown in Table 4, KACS consistently outperforms all baselines at both sparsity levels. At 50% sparsity, our method achieves a PPL of 5.46 and an average accuracy of 74.42%, surpassing the strongest baseline TEAL (PPL 5.60, Avg 74.19%). Notably, WAS (ℓ_1) suffers from numerical instability (NaN PPL) at

Sparsity	Method	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
Llama-2-7B											
0%	Baseline	5.47	69.14	78.07	93.80	31.40	57.14	77.74	76.35	43.43	65.88
50%	TEAL	6.89	66.54	76.28	94.10	28.40	53.51	74.62	73.65	40.10	63.40
	R-Sparse	7.33	65.03	74.26	93.40	28.00	48.82	70.24	71.93	38.40	61.26
	WAS	6.91	67.09	74.76	93.40	29.60	51.03	71.19	71.93	39.93	62.37
	Amber Pruner	6.77	65.90	76.50	94.60	26.40	51.94	73.15	73.15	41.55	62.90
	WANDA-S	6.77	67.72	75.19	94.30	28.20	52.08	73.24	72.56	39.93	62.90
	KACS	6.82	65.82	77.48	94.10	30.00	53.75	74.43	73.86	40.36	63.73
	TEAL*	6.36	65.75	76.64	93.50	30.60	54.22	74.77	74.20	40.15	63.73
	KACS*	6.34	66.77	76.66	93.30	31.20	54.98	74.62	73.11	40.36	63.88
	TEAL	30.62	56.27	66.97	85.60	20.40	34.54	62.84	57.20	24.91	51.09
	R-Sparse	153.68	50.59	54.62	63.70	16.20	26.83	56.94	48.78	28.69	43.29
70%	WAS	124.32	49.57	55.93	64.20	13.20	27.07	61.19	31.69	17.83	40.09
	Amber Pruner	123.39	49.64	57.56	64.10	12.80	27.23	61.62	31.52	17.75	40.28
	WANDA-S	125.49	48.70	56.75	66.90	13.00	26.94	61.28	31.31	17.32	40.28
	KACS	25.49	56.83	67.68	85.90	21.40	35.79	62.78	60.65	25.51	52.07
	TEAL*	12.91	61.12	70.44	86.60	25.40	43.54	64.68	64.27	32.08	56.02
	KACS*	12.51	61.88	70.57	87.80	26.60	44.04	65.50	65.28	32.85	56.82
Llama-2-13B											
0%	Baseline	4.88	72.22	79.00	94.50	35.20	60.07	80.61	79.38	48.46	68.68
50%	TEAL	5.47	68.51	77.09	94.90	32.20	57.71	79.63	77.78	44.88	66.59
	R-Sparse	5.98	70.00	77.36	95.10	32.00	56.39	76.69	77.78	46.16	66.44
	WAS	5.89	67.40	76.77	94.90	32.60	56.85	78.90	77.53	45.39	66.29
	Amber Pruner	6.09	69.46	75.90	95.30	32.00	56.14	76.85	77.19	42.58	65.68
	WANDA-S	5.83	67.80	77.26	94.50	32.40	57.19	78.75	76.56	43.94	66.05
	KACS	5.44	70.56	77.53	95.40	32.80	57.74	79.42	77.65	46.42	67.19
	TEAL*	5.44	70.32	78.11	94.40	33.40	58.35	78.78	77.95	46.70	67.25
	KACS*	5.43	70.82	78.24	94.70	33.60	58.99	78.93	77.65	46.93	67.48
	TEAL	12.56	57.22	70.84	90.70	22.00	40.15	66.15	65.91	31.48	55.56
	R-Sparse	39.58	49.41	61.40	75.50	18.60	27.36	61.80	48.11	22.35	45.57
70%	WAS	33.42	49.17	60.23	76.90	14.80	29.16	61.44	44.11	20.39	44.53
	Amber Pruner	25.28	58.17	66.92	89.20	22.40	40.99	63.21	57.83	30.12	53.61
	WANDA-S	30.59	51.07	63.33	81.80	17.40	30.37	61.65	50.21	22.35	47.27
	KACS	10.28	59.27	71.87	90.90	23.80	41.10	66.94	67.34	31.83	56.63
	TEAL*	8.32	62.35	74.54	91.90	29.40	51.37	71.65	70.83	38.57	61.33
	KACS*	8.04	66.61	74.76	92.20	29.20	52.59	73.15	71.80	39.68	62.50

Table 3: Evaluation across Llama-2-7B and Llama-2-13B at 50% and 70% sparsity levels, showing that KACS consistently outperforms baselines. Note that ‘*’ implies the use of sparsity allocation.

Sparsity	Method	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
Llama-3-70B											
0%	Baseline	2.85	80.58	82.37	96.70	48.60	84.89	85.17	86.95	64.33	78.70
50%	TEAL	5.60	73.88	79.82	96.30	43.40	79.93	81.50	82.20	56.48	74.19
	Amber Pruner	9.05	64.33	75.79	92.60	36.40	69.91	70.64	77.36	46.84	66.73
	WAS (ℓ_2)	9.12	63.06	75.95	93.80	35.80	68.96	70.49	76.30	43.77	66.02
	WAS (ℓ_1)	NaN	49.57	49.51	0.00	27.60	25.04	37.83	25.08	22.70	29.67
	KACS	5.46	74.19	79.60	96.00	44.40	79.94	81.65	82.95	56.66	74.42
70%	TEAL	29.17	58.25	74.70	83.70	32.40	58.14	59.27	72.31	44.54	60.41
	Amber Pruner	82.28	50.91	66.10	72.30	30.00	38.68	50.21	52.57	26.37	48.39
	WAS (ℓ_2)	64.84	49.57	65.29	74.40	29.80	40.92	52.84	52.02	25.09	48.74
	WAS (ℓ_1)	NaN	49.57	49.51	0.00	27.60	25.04	37.83	25.08	22.70	29.67
	KACS	22.67	58.01	75.19	84.80	35.40	59.26	60.64	73.15	45.14	61.45

Table 4: Scalability evaluation on Llama-3-70B at 50% and 70% sparsity. KACS consistently outperforms all baselines, demonstrating effective generalisation to large-scale models.

Sparsity	Method	WikiText-2 ↓	PTB ↓	C4 ↓
0%	Baseline	6.14	9.90	9.07
50%	TEAL	8.05	12.35	12.69
	WAS (ℓ_1)	8.67	12.36	13.28
	WAS (ℓ_2)	8.32	12.62	13.22
	Amber Pruner	8.37	12.67	13.33
	KACS	7.48	11.40	11.67
70%	TEAL	58.80	78.82	92.15
	WAS (ℓ_1)	73.76	82.28	99.64
	WAS (ℓ_2)	67.42	81.32	95.45
	Amber Pruner	65.60	80.38	92.87
	KACS	28.80	51.69	52.20

Table 5: Perplexity comparison across WikiText-2, PTB, and C4 on Llama-3-8B. KACS consistently achieves the lowest perplexity across all datasets, confirming that the improvements are not specific to WikiText-2.

this scale, leading to near-random performance on downstream tasks. At 70% sparsity, KACS continues to exhibit superior robustness, confirming that the performance advantages of KACS generalise consistently to larger-scale models.

Perplexity Across Multiple Datasets. To verify that the perplexity improvements are not specific to WikiText-2, we report PPL on two additional language modelling benchmarks—Penn Treebank (PTB) and C4—for Llama-3-8B at both 50% and 70% sparsity. As shown in Table 5, KACS consistently achieves the lowest PPL across all three datasets and both sparsity levels, with particularly pronounced advantages at the aggressive 70% sparsity regime.

To further validate that the advantages of KACS extend beyond perplexity and classification-style tasks, we evaluate on generative benchmarks and long-context understanding tasks using Llama-3.1-8B-Instruct at 50% sparsity.

Generative Benchmarks. We evaluate on GSM8K (mathematical reasoning), HumanEval (code generation), and IFEval (instruction following). As shown in Table 6, KACS achieves the best overall performance, with particularly strong gains on GSM8K (63.00 vs. 59.06 for TEAL) and HumanEval (56.10 vs. 53.05 for TEAL). These results confirm that the benefits of KACS extend to autoregressive generation scenarios.

Long-Context Evaluation. We further evaluate on five datasets from LongBench, covering diverse long-context understanding tasks. As shown in Table 7, KACS achieves the highest average score (34.76) across all baselines, most notably recovering performance close to the dense baseline on QMSum (25.21 vs. 25.22) and Multi-News (27.13 vs. 27.14), demonstrating strong robustness under long-context scenarios.

Method	PPL	GSM8K	IFEval (prompt)	IFEval (inst)	HumanEval
Baseline	7.21	78.09	41.96	56.35	60.37
TEAL	8.74	59.06	36.97	52.88	53.05
WAS (ℓ_1)	9.07	58.45	35.12	50.00	45.73
WAS (ℓ_2)	9.05	58.83	35.86	51.32	47.56
Amber Pruner	9.02	57.24	34.94	49.64	47.56
KACS	8.46	63.00	38.45	52.52	56.10

Table 6: Evaluation on generative benchmarks (Llama-3.1-8B-Instruct, 50% sparsity). KACS achieves the best overall performance across mathematical reasoning, instruction following, and code generation tasks.

Method	NarrativeQA	Qasper	HotpotQA	QMSum	Multi-News	Avg
Baseline	29.87	45.49	54.97	25.22	27.14	36.54
TEAL	25.47	43.61	49.83	24.23	26.90	34.01
WAS (ℓ_1)	24.23	42.90	49.60	24.85	26.65	33.65
WAS (ℓ_2)	25.04	43.92	49.93	24.70	26.63	34.04
Amber Pruner	25.16	43.93	49.78	24.42	26.46	33.95
KACS	26.97	44.05	50.44	25.21	27.13	34.76

Table 7: Long-context evaluation on LongBench (Llama-3.1-8B-Instruct, 50% sparsity). KACS achieves the highest average score across all five datasets.

B More Layer-wise Relative Reconstruction Error Analysis

To provide a more fine-grained verification of our proposed selection criteria, we conduct a reconstruction error analysis. Figure 6 visualizes the layer-wise relative reconstruction error defined in Eq. 10 (Δ MSE) of different sparsity metrics relative to the TEAL baseline. We observe three key phenomena across all four models (Llama-3-8B, Mistral-7B, Llama-2-7B/13B):

- **Validation of Kurtosis-based Metric.** First, we observe that our proposed Kurtosis-based metric consistently outperforms other weight norm metrics (e.g., ℓ_1/ℓ_2 norms). Whereas the weight norm metrics frequently fluctuate above the zero line (indicating errors higher than TEAL), the Kurtosis metric generally maintains reconstruction errors lower than the TEAL baseline. This demonstrates that Kurtosis serves as a superior metric for identifying activation importance.
- **Superiority of Interaction-Aware Strategy.** Building upon the effective foundation of the Kurtosis metric, our Interaction-Aware metric demonstrates continuous improvement, achieving the lowest reconstruction error across almost all layers. By further incorporating the interaction information within coupled components, this method refines the activa-

tion selection process and consistently yields lower MSE compared to using the Kurtosis metric alone. This confirms that integrating interaction information leads to more precise activation sparsity decisions.

- **High Sensitivity in Initial Layers.** We notice an exception in the very first few layers (e.g., layers 0-1 in Llama-2-7B or 0-3 in Llama-2-13B), where our metrics do not strictly achieve the minimum relative MSE compared to baselines. This indicates that the initial layers are highly sensitive to compression, consistent with previous research (Ma et al., 2023) which highlights that these layers are both hypersensitive to pruning and critical for maintaining model performance. Consequently, practical deployment strategies often recommend keeping these initial layers dense to avoid performance degradation.

C Additional Implementation Details

C.1 MSE-based Greedy Allocation

In Table 2, the row “+ MSE Greedy” corresponds to the standard MSE-based greedy allocation strategy adopted in TEAL (Liu et al., 2025), which we also employ as a complementary refinement.

Given a target effective sparsity for each layer (obtained from our layer-wise allocation), we need to distribute this budget across the seven projections (q, k, v, o , gate, up, down). Starting from the

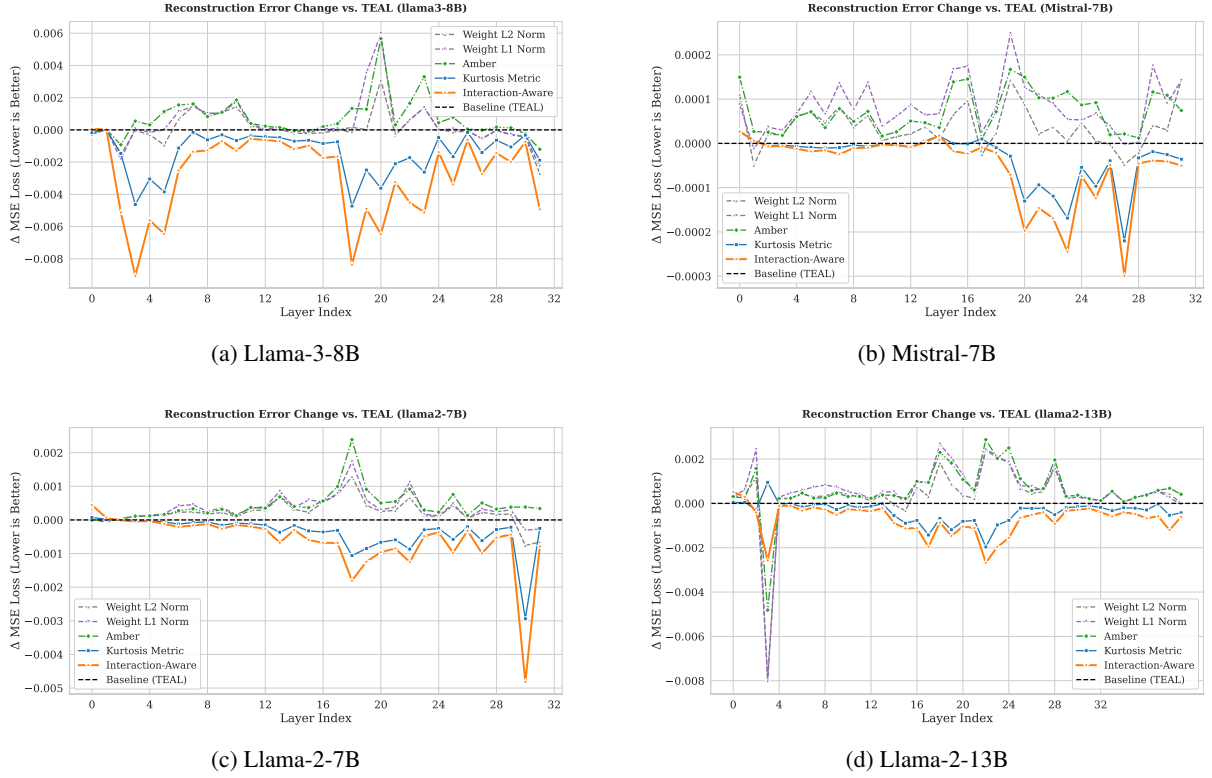


Figure 6: Layer-wise reconstruction error comparison against the TEAL baseline. We report the Δ MSE loss relative to the TEAL baseline (i.e., subtracting the TEAL MSE from each metric’s MSE) across four models. Consequently, negative values indicate lower error compared to TEAL. The dashed line at 0 represents the performance of TEAL.

dense configuration (all projection sparsities set to zero), we gradually *increase* sparsity in small steps: at each iteration, for every projection we tentatively raise its sparsity by a projection-specific step size (defined as a fixed number of parameters), run a forward pass on the calibration activations, and measure the reconstruction error between the dense output Y and the sparse output $\hat{Y}$ by computing the mean squared error between their layer activations on the last 25% of tokens in the sequence, strictly following the implementation details in TEAL. We then select the projection whose sparsity increase leads to the *smallest* increase in this error, commit that update, and repeat this process until the desired effective sparsity of the layer is reached. In this way, the global sparsity budget is fixed, but the per-projection sparsities are chosen to minimize the MSE with respect to the dense model.

This greedy allocation is performed *offline* on the same small calibration set as TEAL using pre-recorded layer activations. It only requires a limited number of additional forward passes and no back-propagation. At inference time, we simply load the resulting per-projection thresholds.

C.2 Threshold Mechanism

Thresholds are computed *per projection*. For each linear layer ($q, k, v, o, \text{gate}, \text{up}, \text{down}$), we first form the importance scores as the element-wise product $S_{ij} = |X_{ij}| \cdot \gamma_j$, where γ_j is a fixed, precomputed channel-wise scale derived from the weight matrix and is *not* thresholded or normalized itself. We then collect the distribution of these scores over the calibration set (all tokens and channels for that projection) and set a single scalar threshold t for that layer as the empirical quantile corresponding to the target sparsity (e.g., 50%). At inference time, the same scalar t is applied to all tokens in that layer; only $|X_{ij}|$ changes across tokens and contexts, while γ_j remains fixed. Thus, the product $|X_{ij}| \cdot \gamma_j$ is computed *before* thresholding, and sparsity is controlled by one threshold per projection, without any additional cross-projection normalization of the γ vectors.

D Analysis of Kurtosis-based Metric

D.1 Validation of Outlier Importance

To empirically validate the rationale behind using Kurtosis as a proxy for feature importance, we

conducted a destructive sensitivity analysis. We compared our method against targeted and random interventions, specifically focusing on the Q, K, and V projections. As shown in Table 8, neutralizing the identified high-kurtosis features by clipping them to their minimum values (Outliers $\rightarrow$ Min) resulted in a catastrophic performance collapse, yielding the lowest average accuracy (47.55%) and a significant spike in perplexity (62.81). Crucially, this degradation is markedly more pronounced than that of the random control (Random $\rightarrow$ Min), which maintained a relatively stable average accuracy of 50.61%. This 3.06% margin provides definitive evidence that features identified by Kurtosis possess a much higher information density and are structurally "load-bearing" for the model. Consequently, preserving these specific outliers is imperative for minimizing information loss during sparsification.

D.2 Comparative Study of Statistical Metrics

Beyond validating the importance of outliers, we further investigate why Kurtosis is the most suitable statistic compared to other heavy-tailed or outlier-related measures. We benchmarked Kurtosis against several metrics suggested by recent literature, including Skewness, Quantile (top 1% magnitude), Max-to-Mean Ratio, Gini Coefficient, and Median Absolute Deviation (MAD).

As shown in Table 9, while simpler metrics such as Quantile and Max-to-Mean Ratio already outperform the TEAL baseline (achieving 46.52 and 45.35 PPL, respectively), they still lag significantly behind Kurtosis (31.36 PPL). This suggests that while magnitude-based sensitivity is partially effective for identifying important features, it fails to fully capture the complex heavy-tailed characteristics of LLM activations. The substantial performance margin demonstrates that "importance" in LLMs is not solely determined by absolute values or simple ratios, but is more fundamentally tied to the extreme tail-heaviness that the fourth-order moment Kurtosis is uniquely suited to characterize. This empirical evidence justifies our choice of Kurtosis as the primary proxy for feature importance.

D.3 Proportion of Channels with Positive Kurtosis

A natural question is how many channels are actually affected by the ReLU filter in Eq. (6), i.e., how many weight columns exhibit negative (sub-Gaussian, platykurtic) excess kurtosis and thus receive no boosting. We compute the excess kurtosis

for every weight column across all projection layers of Llama-3-8B and find that 88.8% of channels exhibit positive excess kurtosis on average. The most pruning-sensitive projections (q_proj, k_proj, gate_proj) approach 100%, while the remaining projections (v_proj, o_proj, up_proj, down_proj) range from 80.9% to 88.9%. The ReLU filter therefore only neutralizes the remaining $\sim 11\%$ of channels, which by definition lack the outlier structure that benefits from protective scaling. This confirms that the Kurtosis metric is not sparse in its coverage — it provides meaningful differentiation for the vast majority of channels, while correctly assigning a neutral scaling factor to the small fraction of platykurtic channels.

D.4 Layer-wise Discriminability of Kurtosis

To verify that the superior discriminability of the Kurtosis metric observed in Figure 3 is not specific to a single layer, we compute a discriminability ratio (top-1% channel scores divided by the mean score) for both Kurtosis and column norm across representative layers of Llama-3-8B (gate projection). Across layers 0, 8, 16, 24, and 31, the Kurtosis metric achieves discriminability ratios of $51.5\times$, $17.4\times$, $17.9\times$, $33.3\times$, and $11.0\times$, respectively. In contrast, the column norm remains near $1\times$ throughout (ranging from 1.06 to 1.37), indicating that it assigns nearly uniform scores to all channels regardless of depth. This consistent gap of $10\sim 51\times$ across all sampled layers confirms that the phenomenon observed in Figure 3 is a universal property of the model rather than a layer-specific artefact, and provides strong evidence that Kurtosis is a reliable proxy for identifying salient channels throughout the network.

E Ablation Study on Interaction Rules

To investigate the effectiveness of the interaction-aware scaling factor, we explore different fusion strategies for combining the importance vectors across multiple weight projections (i.e., $\{\gamma^Q, \gamma^K, \gamma^V\}$ in MHA). Specifically, we compare the proposed element-wise product (Product) with two alternative rules: element-wise addition (Sum) and element-wise maximum (Max). While Sum and Max represent looser constraints by preserving channels that are salient in any single branch, the Product rule enforces a stricter synergistic coupling across all related projections.

As presented in Table 10, the Product rule signifi-

Method	PPL ↓	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg ↑
TEAL	58.43	51.54	62.68	84.30	19.40	30.65	59.88	53.79	22.44	48.09
Kurtosis	31.36	53.99	66.54	85.90	18.60	35.28	61.71	60.73	27.56	51.29
Outliers → Min	62.81	52.72	64.25	80.30	19.20	29.78	60.12	51.35	22.70	47.55
Random → Min	37.00	53.99	66.70	84.70	18.20	33.73	61.59	59.09	26.88	50.61

Table 8: Ablation study validating the importance of kurtosis. The targeted suppression of outliers (Outliers → Min) leads to the most severe performance degradation, significantly worse than random suppression. This performance gap confirms that Kurtosis effectively identifies the most critical, load-bearing parameters in the network.

Importance Metric	PPL ↓	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg ↑
Skewness	50.42	53.67	64.96	84.30	19.20	31.62	60.73	56.14	24.91	49.44
Quantile	46.52	55.25	64.74	85.10	19.40	32.48	61.31	57.79	25.51	50.20
Max-to-Mean Ratio	45.35	55.49	65.56	85.30	18.20	32.70	61.44	57.83	25.77	50.29
Gini Coefficient	58.89	52.57	63.11	84.00	19.60	30.80	60.12	54.08	23.04	48.42
MAD	50.40	52.25	64.04	85.10	19.20	32.13	60.18	59.09	26.71	49.84
Kurtosis (Ours)	31.36	53.99	66.54	85.90	18.60	35.28	61.71	60.73	27.56	51.29

Table 9: Comparative analysis of different statistical metrics for identifying salient features. The proposed Kurtosis-based metric consistently achieves the lowest perplexity and highest average accuracy, validating its effectiveness in capturing critical outliers.

Method	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
Sum	43.34	55.33	64.25	86.30	20.20	33.13	61.77	58.16	26.28	50.68
Max	41.32	52.49	66.27	85.70	18.00	33.48	60.58	57.66	25.77	49.99
Product	31.36	53.99	66.54	85.90	18.60	35.28	61.71	60.73	27.56	51.29

Table 10: Performance comparison of different methods on various benchmarks.

cantly outperforms both the Sum and Max variants. Notably, Product achieves a substantially lower perplexity (31.36) compared to the alternatives, which both exceed 40. These results suggest that the complex interactions necessitate a consensus-based saliency metric. Looser fusion rules tend to introduce "pseudo-salient" noise by preserving channels that lack consistent importance across multiple branches, ultimately leading to suboptimal compression performance.

Scale Control in the Interaction-Aware Metric. No explicit normalisation is applied to the joint scaling factor before thresholding, because potential scale imbalances are already controlled by the bounded design of γ_j in Eq. (6). Specifically, three properties of the formulation jointly prevent unbounded or degenerate scales: (1) the ReLU operator enforces a hard lower bound of $\gamma_j \geq 1$ for all channels; (2) the logarithmic term $\log(1 + \cdot)$ compresses extreme kurtosis values, preventing any single channel from receiving a disproportionately large multiplier; (3) the additive constant of 1 ensures a consistent baseline scale across all channels regardless of kurtosis magnitude. Consequently,

the residual scale variation in the element-wise product across coupled projections is not an undesirable artefact, but rather a direct reflection of genuine importance differences between channels. It is precisely this discriminative spread that allows our method to outperform baselines that assign more uniform importance scores.

F Robustness and Sensitivity Analysis

F.1 Impact of Calibration Batch Size

To evaluate the sensitivity of KACS to the scale of calibration data, we conducted experiments across various batch sizes ranging from 10 to 50. As summarized in Table 11, the performance remains remarkably stable across all settings. Specifically, the average accuracy across eight benchmarks exhibits minimal fluctuation ($66.42\% \pm 0.17\%$), and the WikiText2 PPL shows negligible variance (7.455 ± 0.010). These results demonstrate that KACS is highly robust to calibration data volume and can achieve competitive performance even with a minimal batch size of 10. This high sample efficiency significantly reduces the computational overhead and time required for the offline calibra-

Batch	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
10	7.45	67.56	76.82	96.60	31.80	56.47	78.01	77.53	45.99	66.35
20	7.45	68.51	76.93	96.20	32.20	56.29	76.88	77.23	45.56	66.23
40	7.47	70.17	77.04	96.40	32.60	56.33	76.85	77.06	46.67	66.64
50	7.45	68.98	77.69	96.30	33.40	56.60	76.79	77.61	44.37	66.47
Mean	7.45	68.81	77.12	96.38	32.50	56.42	77.13	77.36	45.65	66.42
Std	0.010	1.08	0.39	0.17	0.68	0.14	0.59	0.26	0.97	0.17

Table 11: Performance of KACS method with different batch sizes.

Seed	PPL	WG	PIQA	SciQ	OBQA	HS	BoolQ	Arc-E	Arc-C	Avg
42	7.45	69.69	77.37	96.00	33.40	56.29	77.77	76.73	45.31	66.57
100	7.46	68.82	77.64	96.30	32.80	56.30	77.61	77.99	45.73	66.65
123	7.46	68.51	77.42	96.50	33.00	56.46	76.54	77.36	45.73	66.44
2026	7.46	67.80	77.58	96.40	34.80	56.46	77.13	78.28	45.56	66.75
3407	7.46	70.17	77.75	95.60	34.40	56.25	77.31	77.27	44.54	66.66
9999	7.46	68.88	77.53	96.30	31.80	56.24	77.77	77.40	44.71	66.32
Mean	7.46	68.98	77.55	96.18	33.37	56.33	77.36	77.50	45.26	66.56
Std	0.00	0.84	0.14	0.33	1.10	0.10	0.47	0.55	0.52	0.16

Table 12: Performance stability analysis of KACS across different random seeds.

tion phase.

F.2 Sensitivity to Random Seeds

We further investigate the robustness of the KACS method across different random seeds. Specifically, we select several widely used seeds in the research community (e.g., 42 and 3407) to ensure the breadth of our evaluation. As presented in Table 12, our method exhibits high consistency across all six independent runs. The average accuracy remains stable at $66.56\% \pm 0.16\%$, while the WikiText2 PPL shows effectively zero fluctuation at two decimal places (7.46 ± 0.00). These results indicate that KACS is insensitive to random initialization, ensuring reliable and reproducible performance in practical applications.

G Decoding Speedup

To validate the practical efficiency of our framework, we evaluated the end-to-end single-batch decoding latency on an NVIDIA A800 GPU. We utilize the official implementation provided by TEAL. We enable both **CUDA Graphs** and **torch.compile** to ensure a competitive dense baseline. Similar to standard benchmarks, we set the input prompt length to 6 tokens and generated 200 output tokens. With only 6 input tokens, the prefill phase is negligible, so the reported speedups reflect primarily the **decoding phase** (memory-bound), which is the dominant cost in interactive

inference and the scenario that activation sparsity is designed to accelerate. Table 13 presents the comprehensive benchmarking results across Llama-3-8B, Llama-2-7B/13B, and Mistral-7B. Our method achieves significant wall-clock acceleration that scales near-linearly with sparsity, closely matching the state-of-the-art TEAL baseline. For instance, on Llama-2-13B, we observe speedups of $1.58\times$ and $1.92\times$ at 50% and 70% sparsity, respectively, demonstrating effective hardware acceleration. The latency overhead compared to TEAL is consistently negligible ($< 2\%$), confirming that our interaction-aware metric improves selection accuracy with minimal impact on inference throughput.

H Discussion

Comparison against training-based methods.

Training-based activation sparsity methods (e.g., ReLUfication (Mirzadeh et al., 2023), Spark Transformers (You et al., 2025)) have demonstrated that, given sufficient compute and data, it is possible to recover or even improve performance under aggressive sparsity by retraining the model. In that sense, they can achieve a stronger absolute accuracy-sparsity trade-off than purely training-free methods, especially when sparsity is tightly integrated into the pre-training process.

However, these gains come at the cost of substantial additional training: either fine-tuning a large dense checkpoint with modified activations (as in

Model	Sparsity	Dense Speed (tok/s)	TEAL		Ours	
			Speed (tok/s)	Speedup	Speed (tok/s)	Speedup
Llama-3-8B	25%	76.50	91.82	1.20×	91.54	1.20×
	40%		106.26	1.39×	105.06	1.37×
	50%		113.67	1.49×	114.14	1.49×
	70%		129.54	1.69×	128.79	1.68×
Llama-2-7B	25%	83.44	102.82	1.23×	101.57	1.22×
	40%		117.98	1.41×	117.11	1.40×
	50%		130.14	1.56×	128.12	1.54×
	70%		148.01	1.77×	145.54	1.74×
Llama-2-13B	25%	47.05	58.14	1.24×	57.46	1.22×
	40%		68.04	1.45×	67.34	1.43×
	50%		75.71	1.61×	74.52	1.58×
	70%		91.72	1.95×	90.56	1.92×
Mistral-7B	25%	79.02	98.18	1.24×	96.69	1.22×
	40%		112.73	1.43×	112.66	1.43×
	50%		124.35	1.57×	123.63	1.56×
	70%		152.40	1.93×	149.42	1.89×

Table 13: Decoding-dominant throughput on NVIDIA A800 GPU with a 6-token prompt and 200 generated tokens. Speed is measured in generated tokens per second (tokens/s) **averaged over 5 independent runs**. Our method achieves acceleration comparable to TEAL across all sparsity levels.

ReLUfication), or training a new sparse architecture from scratch (as in Google’s Spark Transformers). Such pipelines are only feasible when developers control the full training stack and can afford the extra compute. Our goal in this work is different: KACS is a *training-free*, post-hoc method that operates on off-the-shelf dense checkpoints and plugs directly into existing inference kernels. It is intended for deployment scenarios where retraining is impractical, but practitioners still desire a strong sparsity–performance trade-off without modifying weights. In settings where retraining is possible and the additional compute is justified, training-based methods may indeed achieve higher absolute performance; KACS aims to provide a competitive option in the complementary, zero-training regime.

Discussion on Allocation Strategies. In this study, our primary focus is the development of reliable importance metrics. The sparsity allocation component is intentionally kept simple to serve a clear purpose: demonstrating that our Kurtosis-based importance metric is orthogonal to, and composable with, adaptive allocation strategies. As shown in the ablation study (Table 2), combining KACS with layer-wise allocation progressively improves performance, confirming that the two com-

ponents are complementary rather than entangled. For the layer-wise redundancy estimate, we adopt cosine similarity as a lightweight proxy that requires no additional computation beyond a single forward pass and has been widely adopted in the layer pruning literature (Ma et al., 2023). We do not claim it is the uniquely optimal choice—alternative redundancy indicators could be substituted and may yield further gains, as could more sophisticated allocation algorithms such as evolutionary search or reinforcement learning-based budget distribution. Exploring such strategies is an interesting direction, but falls outside the scope of this work, whose primary focus is the development of reliable activation importance metrics.

I Hyperparameter Selection

To ensure the effectiveness of our method, we determined the values of hyperparameters α and β by performing a grid search on the calibration dataset. We empirically observed that our method achieves robust performance when the hyperparameters are selected within the ranges of $\alpha \in [0.3, 0.8]$ and $\beta \in [0.05, 0.08]$. These ranges consistently yield competitive results across various sparsity levels.

Sensitivity Analysis for α . To further characterise the robustness of KACS to the choice of α , we conduct a sensitivity analysis on Llama-3-8B, sweeping α across $[0.3, 0.8]$ in steps of 0.1. At 50% sparsity, PPL varies by only 0.07 and average accuracy by only 0.28 across the entire range. At 70% sparsity, performance also remains competitive throughout, with mean PPL of 29.72 and mean accuracy of 51.63%. These results confirm that KACS requires no careful per-model tuning of α ; we adopt $\alpha = 0.6$ as the default across all experiments. We further note that the grid search itself is fundamentally different from training-based hyperparameter optimisation: it requires only a single forward pass per candidate value with no gradient computation or weight updates, making it negligibly cheap and fully consistent with the plug-and-play nature of KACS.

J Limitations and Future Work

While effective at moderate sparsity, our method faces performance drops in aggressive sparsity regimes. Mitigating this accuracy loss to fully exploit the potential of high-sparsity inference requires further investigation. Furthermore, combining our approach with complementary techniques like quantization or MoE architectures represents a promising avenue for broader application. Since activation sparsity and weight quantization target orthogonal sources of redundancy—the former eliminates redundant activation channels at runtime while the latter reduces the numerical precision of stored weights—KACS can in principle be straightforwardly combined with quantization methods without conflict. Exploring such combinations is left for future work.

K Ethical Considerations

Our work primarily contributes to “Green AI” by significantly reducing the energy consumption and carbon footprint of LLM inference. By lowering hardware barriers, our method also promotes the democratization of advanced AI, enabling broader deployment on resource-constrained devices. As is standard practice for any LLM deployment, we recommend that users verify the safety alignment of the compressed models prior to sensitive applications.

L AI Assistance Disclosure

In the preparation of this manuscript, the authors utilized AI language models (Gemini, Google) to assist with language refinement and document formatting. Specifically, AI tools were employed to enhance the clarity and grammatical correctness of the writing, generate LaTeX code for tables and layout, and suggest structural organizations for the appendix. All core scientific contributions, including the conceptualization of the proposed method, algorithm design, experimental implementation, and the analysis of results, were conceived and executed entirely by the authors. The authors have carefully reviewed all AI-assisted content for accuracy and assume full responsibility for the final manuscript.