CRePE: Convolution-aware Relative Importance in Post-training Pruning with Efficient Search

Anonymous authors
Paper under double-blind review

Abstract

Post-Training Pruning (PTP) reduces the memory and computational costs of Large Language Models (LLMs) by removing weights without retraining. RIA, a state-of-the-art scoring method, normalizes each weight by its row and column sums, but uses only 1D cross-shaped information and weights the two directions equally. We propose **CRePE** (**C**onvolution-aware **Re**lative Importance in **P**ost-training Pruning with **E**fficient Search), which adds a 2D local neighborhood term and adaptive per-term coefficients to Relative Importance scoring. Since searching these coefficients by perplexity (PPL)-based hill climbing takes about 11 hours on LLaMA-2-7B, we further propose **PHO** (Proxy-based Hyperparameter Optimization), which uses the Gini coefficient of pruned importance scores as a surrogate objective and reduces the search to about 20 minutes, a 30$\times$ speedup with nearly the same PPL; the resulting coefficients transfer from LLaMA-2-7B to LLaMA-2-13B without re-tuning. Across LLaMA-1/2/3 under 2:4 and 4:8 sparsity and LLaMA-1/2 under unstructured sparsity, CRePE consistently improves over scoring-only baselines, with margins that widen under tighter structural constraints or higher sparsity. Under 2:4 sparsity, CRePE lowers PPL by 0.43 on average relative to RIA. CRePE also combines with channel permutation, non-uniform sparsity allocation, quantization, and re-pruning.

1 Introduction

Large Language Models (LLMs) have demonstrated remarkable performance across a wide range of natural language processing tasks (Brown et al., 2020; Touvron et al., 2023a), yet their billions of parameters impose substantial memory and computational costs during inference. To address this, various compression techniques have been explored, including quantization (Frantar et al., 2022; Xiao et al., 2023), knowledge distillation (Hinton et al., 2015), and weight pruning (Han et al., 2015; Sun et al., 2023). Among these, post-training pruning (PTP) removes unnecessary weights without any retraining, making it a practical and training-free solution well-suited for LLMs.

Among PTP methods, SparseGPT (Frantar & Alistarh, 2023) first demonstrated the viability of LLM pruning by leveraging Hessian approximations to jointly select and update weights. Wanda (Sun et al., 2023) introduced an importance criterion based on the product of weight magnitudes and input activations, achieving competitive performance without weight updates. RIA (Zhang et al., 2024) further advanced this line of work by introducing Relative Importance (RI), which normalizes each weight by the sum of its row and the sum of its column. Unlike weight update-based methods such as SparseGPT, which require approximately 90 minutes to prune LLaMA-2-70B, RIA achieves competitive accuracy through scoring alone, reducing pruning time to approximately 10 minutes.

Despite its efficiency, RIA has two fundamental limitations. **First**, RIA's importance scores rely solely on 1D cross-shaped directional information from the weight matrix. As weight matrices are inherently 2D structures, diagonal neighboring weights can also contribute to RI estimation. **Second**, RIA treats row and column contributions equally. Our preliminary experiments show that row and column directions contribute asymmetrically to pruning quality: using row-wise RI alone yields lower perplexity than column-wise RI alone (PPL 6.83 vs. 7.49 on LLaMA-2-7B), making uniform treatment suboptimal.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Motivated by these observations, we propose **CRePE** (**C**onvolution-aware **Re**lative Importance in **P**ost-training Pruning with **E**fficient Search), which extends RIA in two ways. First, it augments the cross-shaped row and column terms with a 2D neighborhood term to capture local weight context. Second, it assigns adaptive coefficients α, β, γ to the row, column, and neighborhood contributions, respectively, to explicitly model their asymmetric importance. CRePE achieves the lowest perplexity among the evaluated methods without weight updates across diverse models and sparsity settings.

However, CRePE introduces a new challenge: finding the optimal adaptive coefficients (α, β, γ) for each model and sparsity setting. A naive PPL-based hill climbing search requires approximately 11 hours even on LLaMA-2-7B, undermining the efficiency advantage of PTP. To address this, we propose **PHO** (**P**roxy-based **H**yperparameter **O**ptimization). We first conduct a systematic Spearman correlation analysis across diverse hyperparameter configurations and discover that the Gini coefficient (Ceriani & Verme, 2012) of the pruned weight importance scores at the first-layer `q_proj` exhibits a strong positive correlation with PPL ($\rho = 0.953$). This finding suggests that $\text{Gini}_{\text{pruned}}$ can serve as a reliable surrogate for PPL evaluation. PHO leverages this proxy within a CMA-ES-based search framework to screen candidates cheaply, so that full PPL evaluation is needed for only a small number of them, reducing total search time from 11 hours to approximately 20 minutes, a $30\times$ speedup.

The main contributions are as follows:

- We propose CRePE, which extends RIA's 1D cross-shaped Relative Importance with a 2D neighborhood term (Sec. 3.2) and adaptive coefficients (Sec. 3.3) to explicitly model the differential contributions of row, column, and neighborhood terms.
- We discover that the Gini coefficient of pruned weight importance scores strongly correlates with PPL ($\rho = 0.953$), and leverage this as a proxy objective in PHO, an efficient CMA-ES-based hyperparameter optimization framework that achieves a $30\times$ speedup over PPL-based hill climbing (Sec. 4).
- We demonstrate that CRePE attains the lowest perplexity among the evaluated methods without weight updates on LLaMA-1/2/3 under 2:4 and 4:8 sparsity and on LLaMA-1/2 under unstructured sparsity (Sec. 5.2).
- We empirically verify that CRePE can be orthogonally combined with existing pruning extensions, including Channel Permutation, non-uniform sparsity allocation (OWL, AlphaPruning, DLP), and re-pruning (Sec. 5.4).

2 Related Work

2.1 Pruning for LLMs

Post-training pruning. Post-training pruning (PTP) removes weights from pretrained models without retraining, making it particularly attractive for LLMs where fine-tuning at scale is prohibitively expensive. SparseGPT (Frantar & Alistarh, 2023) was the first practical PTP method for LLMs, framing pruning as a layerwise reconstruction problem and using approximate Hessian information to jointly select and update weights; subsequent reconstruction-based methods improve the weight update through ADMM (Boža, 2024), ALPS (Meng et al., 2024), and FISTA (Zhao et al., 2025). Wanda (Sun et al., 2023) instead introduced a scoring-only criterion based on weight magnitudes and the ℓ_2 norm of input activations, achieving performance close to SparseGPT without any weight updates, and RIA (Zhang et al., 2024) further normalizes each weight by its row and column sums to mitigate channel corruption. CRePE belongs to this scoring-only category, requiring neither retraining nor weight updates.

Automated pruning metrics. Other works automate the design of pruning metrics, through symbolic search (Dong et al., 2024) or learnable normalization factors tuned by block-wise reconstruction (Liu et al., 2025). CRePE differs from these metrics by adding a 2D neighborhood term and by tuning its coefficients through a single-module proxy. Beyond post-training methods, MaskLLM (Fang et al., 2024) learns semi-structured masks end-to-end at the cost of additional training.

Structured pruning. Structured pruning (Ma et al., 2023; Ashkboos et al., 2024; An et al., 2024) and layer-level pruning (Song et al., 2024; Yang et al., 2024; Men et al., 2024) remove entire com-

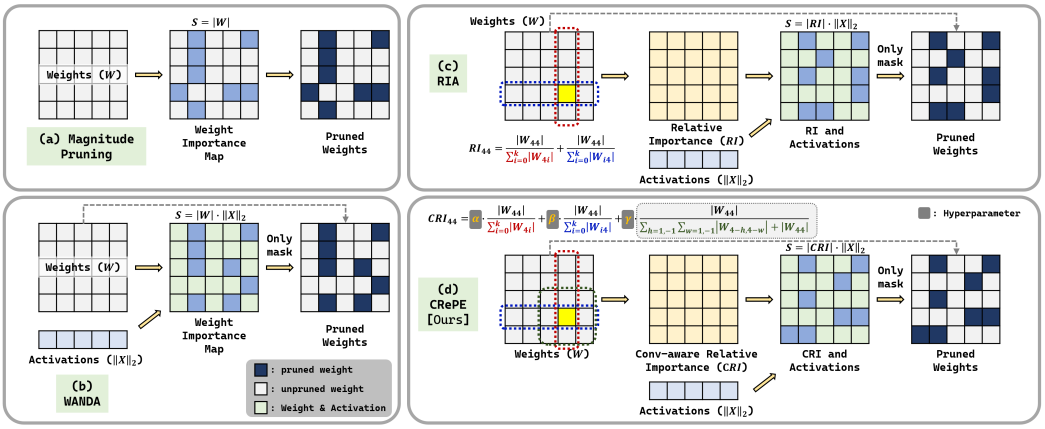


Figure 1: Importance score computation of (a) Magnitude Pruning, (b) Wanda, (c) RIA, and (d) CRePE. CRePE extends RIA's cross-pattern with a 2D neighborhood term (Sec. 3.2) and adaptive coefficients α, β, γ (Sec. 3.3).

ponents such as attention heads, layers, or neurons, yielding hardware-friendly models without specialized sparse kernels, at the cost of larger accuracy degradation at the same compression ratio.

2.2 Non-uniform LLM Pruning

Most LLM pruning methods apply a uniform sparsity ratio across all layers. OWL (Yin et al., 2024) showed that the layerwise outlier distribution in LLMs is highly non-uniform and assigns lower sparsity to layers with more outliers. AlphaPruning (Lu et al., 2024) estimates layerwise importance using Random Matrix Theory, and DLP (Chen et al., 2025) prioritizes layers dynamically. These methods determine *how much* to prune each layer, whereas CRePE determines *which weights* to prune within each layer; the two are complementary, and we show that CRePE combines with OWL, AlphaPruning, and DLP (Sec. 5.4).

3 CRePE

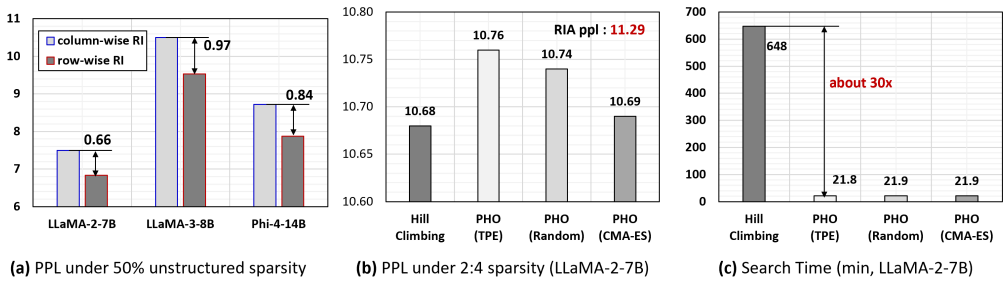
Preliminaries Given a pretrained weight matrix $\mathbf{W} \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}}}$ and the corresponding input activation $\mathbf{X} \in \mathbb{R}^{C_{\text{in}} \times T}$, the goal of post-training pruning is to find a binary mask $\mathbf{M} \in \{0,1\}^{C_{\text{out}} \times C_{\text{in}}}$ that zeros out a target fraction of weights while minimizing the degradation in model performance. Here, $X_j \in \mathbb{R}^T$ denotes the j-th row of $\mathbf{X}$, representing the input activation of the j-th input channel.

3.1 Motivation

The importance score of RIA is defined as:

$$S_{ij}^{\text{RIA}} = \left(\frac{|W_{ij}|}{\sum_k |W_{ik}|} + \frac{|W_{ij}|}{\sum_k |W_{kj}|} \right) \cdot \|X_j\|_2^{0.5} \tag{1}$$

As illustrated in Fig. 1-(c) and Equation 1, RIA evaluates the importance of each weight W_{ij} using only the 1D cross-shaped directional information from its row and column sums. However, weight matrices are inherently 2D structures, and diagonal neighboring weights can also contribute to importance estimation. Furthermore, RIA treats the row and column directions equally, despite the fact that their impacts on pruning quality differ in practice. As shown in Fig. 2-(a), using row-wise RI alone yields lower perplexity than column-wise RI alone on LLaMA-2-7B (6.83 vs. 7.49), indicating that the two directions contribute asymmetrically to pruning quality. Treating them equally, as RIA does, is therefore suboptimal. In the following, we propose **Convolution-aware Relative Importance (CRI)**, which addresses these two limitations by integrating a 2D neighborhood term (Sec. 3.2) and adaptive per-term coefficients (Sec. 3.3).



(a) PPL under 50% unstructured sparsity **(b)** PPL under 2:4 sparsity (LLaMA-2-7B) **(c)** Search Time (min, LLaMA-2-7B)

Figure 2: (a) PPL of RIA using only the row term (α only) or only the column term (β only). (b) PPL and (c) search time of search algorithms in PHO on LLaMA-2-7B under 2:4 sparsity.

3.2 2D Neighborhood Relative Importance

To extend the Relative Importance score with 2D structural information, we define a neighborhood term $N_{ij}^{(k)}$ that captures the local 2D context around each weight W_{ij} while excluding the cross-shaped neighbors already captured by the RIA terms:

$$N_{ij}^{(k)} = \sum_{(h,w)\in\mathcal{D}_k} |W_{i+h,\,j+w}| \tag{2}$$

where $\mathcal{D}_k = \{(h,w) : |h|,|w| \le \lfloor k/2 \rfloor,\ hw \neq 0\} \cup \{(0,0)\}$. The condition $hw \neq 0$ excludes the cross pattern (row and column neighbors), while $(0,0)$ retains the self-magnitude. Note that $(0,0) \in \mathcal{D}_k$ ensures that $|W_{ij}|$ itself contributes to the denominator of the neighborhood term, making it analogous in form to the row and column normalization in Equation 1. For example, $\mathcal{D}_3 = \{(0,0), (\pm1,\pm1)\}$ contains 5 elements, and $\mathcal{D}_5$ contains 17. This neighborhood sum is computed for the entire matrix in a single 2D convolution of $|\mathbf{W}|$ with a masked kernel, adding negligible cost to the scoring pass.

3.3 Adaptive Relative Importance

The row, column, and neighborhood terms may contribute differently to the importance of a weight. To explicitly model these differential contributions, we introduce per-term coefficients α, β, and γ that weight the three terms:

$$\text{CRI}_{ij} = \alpha \cdot \frac{|W_{ij}|}{\sum_k |W_{ik}|} + \beta \cdot \frac{|W_{ij}|}{\sum_k |W_{kj}|} + \gamma \cdot \frac{|W_{ij}|}{N_{ij}^{(k)}} \tag{3}$$

where α, β, and γ control the contributions of the row, column, and 2D neighborhood terms, respectively, and are determined via PHO (Sec. 4). The final CRePE importance score is obtained by combining CRI with the ℓ_2 norm of the input activation:

$$S_{ij}^{\text{CRePE}} = \text{CRI}_{ij} \cdot \|X_j\|_2^{0.5} \tag{4}$$

When $\alpha = \beta = 1$ and $\gamma = 0$, CRePE reduces to RIA, confirming that RIA is a special case of CRePE.

4 PHO

4.1 Motivation

With the kernel size fixed at $k = 5$ (Sec. 5.3.1), CRePE has three continuous hyperparameters, the adaptive coefficients (α, β, γ). Finding their optimal combination via PPL-based hill climbing requires full model pruning and PPL evaluation for each candidate, taking approximately 11 hours on LLaMA-2-7B (see Fig. 2-(c)), compared to only $\sim$110 seconds for RIA and Wanda. This substantially undermines the key advantage of post-training pruning: fast, training-free applicability. To address this, we propose **PHO** (Proxy-based Hyperparameter Optimization), a proxy-guided search framework that screens candidates with a cheap surrogate and reserves full PPL evaluation for only the most promising few.

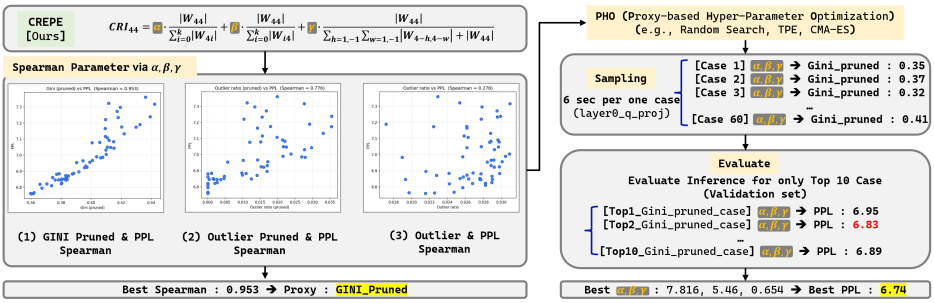


Figure 3: Overview of PHO. **Left**: $\text{Gini}_{\text{pruned}}$ is the most reliable surrogate among the candidate proxies ($\rho = 0.953$). **Right**: CMA-ES samples 60 candidates screened by proxy ($\sim$6 sec each); the top 10 receive full PPL evaluation ($\sim$90 sec each).

Table 1: Spearman correlation between candidate proxies and PPL at the first-layer `q_proj`.

	Outlier			Gini	
Model	All	Retained	Pruned	Retained	Pruned
Qwen3.5-0.8B	0.52	−0.27	0.17	0.90	**0.95**
LLaMA-2-7B	0.28	−0.05	0.78	0.72	**0.95**
LLaMA-2-13B	0.21	0.22	0.71	0.66	**0.92**

4.2 Proxy Definition

An effective proxy must (1) correlate strongly with PPL and (2) be cheap to compute. Rather than pruning the full model, we prune only the `q_proj` module of the first transformer layer, which is the first module processed in the pruning pipeline, and compute the proxy from the importance scores of the pruned weights. This reduces the number of pruned modules by 224× on LLaMA-2-7B (32 layers × 7 projections).

As the proxy, we use the Gini coefficient of the pruned scores, $\text{Gini}_{\text{pruned}} \in [0, 1]$, where 0 indicates a uniform distribution and 1 indicates maximal inequality. A high $\text{Gini}_{\text{pruned}}$ means that a few pruned weights carry disproportionately high scores, which may indicate that some important weights have been removed; a low value means that the removed weights are uniformly unimportant. Two pruned sets with the same mean score can therefore differ in how damaging they are, and we hypothesize that this tail is closely related to the pruning error. We measure the proxy on the pruned set because the error arises from what is removed rather than what is retained. Moreover, scaling (α, β, γ) by a common positive constant leaves the score ranking, and hence the mask, unchanged, so the absolute magnitude of the scores carries no information about pruning quality; a scale-invariant measure such as the Gini coefficient is therefore a natural choice.

We compare several proxy candidates at the first-layer `q_proj` by their Spearman correlation with PPL across three models (Qwen3.5-0.8B, LLaMA-2-7B/13B). As shown in Table 1, $\text{Gini}_{\text{pruned}}$ exhibits the strongest and most consistent positive correlation with PPL, outperforming $\text{Outlier}_{\text{pruned}}$ and $\text{Gini}_{\text{retained}}$, consistent with the intuition above.

To verify that the proxy is not specific to this module, we measure the Gini–PPL correlation at 63 positions spanning the full depth of the network: nine layers sampled at four-layer intervals (layers $0, 4, \ldots, 28, 31$), each with all seven projection types. All 63 positions exhibit positive correlation (mean $\rho = 0.964$, min 0.824), indicating that the surrogate is stable across network depth and projection type.

4.3 Search Algorithm

Analysis Various optimization algorithms can be applied to minimize $\text{Gini}_{\text{pruned}}$ as the surrogate objective. We compare three representative algorithms: CMA-ES (Covariance Matrix Adaptation Evolution Strategy), TPE (Tree-structured Parzen Estimator), and Random Search. As shown in

Table 2: WikiText-2 perplexity across unstructured sparsity ratios.

	LLaMA-1-7B (Dense: 5.68)			**LLaMA-2-7B** (Dense: 5.47)		
Method	**50%**	**60%**	**70%**	**50%**	**60%**	**70%**
Wanda (Sun et al., 2023)	7.26	10.71	86.09	6.92	18.59	237.05
RIA (Zhang et al., 2024)	7.12	10.41	86.07	6.81	10.39	68.92
CRePE	**7.07**	**10.06**	**72.95**	**6.74**	**9.64**	**58.07**

Table 3: WikiText-2 perplexity under 2:4 and 4:8 sparsity. WU: weight updates. **Bold**: best without weight updates; underline: best overall.

Method	**WU**	**2:4 Pruning**								**4:8 Pruning**							
		LLaMA				**LLaMA-2**			**LLaMA-3**	**LLaMA**				**LLaMA-2**			**LLaMA-3**
		7B	**13B**	**30B**	**65B**	**7B**	**13B**	**70B**	**8B**	**7B**	**13B**	**30B**	**65B**	**7B**	**13B**	**70B**	**8B**
Dense	×	5.68	5.09	4.77	3.56	5.47	4.88	3.32	6.14	5.68	5.09	4.77	3.56	5.47	4.88	3.32	6.14
Magnitude	×	42.55	18.32	9.11	7.11	37.76	8.74	6.76	2401.90	16.83	13.73	7.62	6.36	15.91	7.32	5.89	181.50
SparseGPT	✓	11.01	9.01	7.14	6.28	11.01	8.77	5.68	16.31	8.61	7.45	6.13	5.38	8.48	7.01	4.91	12.32
Wanda	×	11.54	9.59	6.90	6.25	12.07	9.00	5.48	24.08	8.57	7.41	5.98	5.30	8.60	7.00	4.77	14.58
RIA	×	11.10	8.95	6.78	6.03	11.29	8.41	5.38	23.07	8.32	7.15	5.78	5.10	8.38	6.74	4.68	13.74
CRePE	×	**10.69**	**8.47**	**6.59**	**5.90**	**10.69**	**7.79**	**5.32**	**22.15**	**8.16**	**6.98**	**5.76**	**5.05**	**8.16**	**6.59**	**4.54**	**13.47**

Fig. 2-(b), all three algorithms achieve comparable PPL within 0.07 of each other, while CMA-ES achieves the lowest PPL among the three samplers and is particularly well-suited to CRePE's continuous hyperparameter space ($\alpha, \beta, \gamma \in [0, 10]$) via its covariance adaptation mechanism. We therefore adopt CMA-ES as the default sampler.

Overall Procedure of PHO As illustrated in Fig. 3, the full PHO search procedure consists of two stages (reported for LLaMA-2-7B): **Stage 1) Proxy-based screening**: A CMA-ES sampler generates 60 hyperparameter candidates. For each candidate, $\text{Gini}_{\text{pruned}}$ is computed from the `q_proj` of layer 0. Each evaluation takes approximately 6 seconds. **Stage 2) PPL-based verification**: Full model pruning and PPL evaluation are performed only for the top 10 candidates with the lowest $\text{Gini}_{\text{pruned}}$. Each evaluation takes approximately 90 seconds. PPL in this stage is measured on the validation set.

The total search time is approximately 20 minutes, representing a speedup of approximately $30\times$ over PPL-based hill climbing ($\sim$11 hours). The number of trials and top-k candidates are set to 60 and 10, respectively, based on empirical validation; further analysis is provided in Sec. 5.3.2.

5 Experiments

5.1 Experimental Setup

Models and datasets We evaluate on LLaMA (7B–65B) (Touvron et al., 2023a), LLaMA-2 (7B–70B) (Touvron et al., 2023b), LLaMA-3 (8B) (Grattafiori et al., 2024), Qwen3.5 (Qwen Team, 2026), and Phi-4 (14B) (Abdin et al., 2024), using public checkpoints from HuggingFace. Language modeling performance is measured by perplexity (PPL) on WikiText-2, and zero-shot performance by average accuracy over seven tasks: RTE (Wang et al., 2018), HellaSwag (Zellers et al., 2019), ARC-Easy/Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), OpenBookQA (Mihaylov et al., 2018), and LAMBADA (Paperno et al., 2016). For calibration, we use 128 sequences of length 2048 randomly sampled from the C4 dataset.

Implementation The coefficients are searched over $\alpha, \beta, \gamma \in [0, 10]$ with the activation exponent fixed at 0.5, and the kernel size is set to $k = 5$ following the ablation in Sec. 5.3.1. We use the CMA-ES implementation in Optuna with a fixed random seed of 0 for both the sampler and calibration sampling. All experiments are conducted on two NVIDIA RTX PRO 6000 Blackwell GPUs with 96 GB of memory each.

Table 4: Effect of sequence length (calibration and evaluation) on LLaMA-2-7B under 2:4 sparsity (WikiText-2 PPL). 4096 follows Wanda (Sun et al., 2023) and BaWA (Liu et al., 2025).

Seq. length	**Wanda** (Sun et al., 2023)	**RIA** (Zhang et al., 2024)	**BaWA** (Liu et al., 2025)	**CRePE**
2048	12.07	11.29	10.95	**10.69**
4096	11.37	10.60	10.31	**10.17**

Table 5: Zero-shot accuracy (%) on LLaMA-2-13B under 2:4 sparsity. CRePE outperforms both baselines on every task. HellaS.: HellaSwag; ARC-e/c: ARC-Easy/Challenge; WinoG.: WinoGrande; OBQA: OpenBookQA; LAMB.: LAMBADA.

Method	**RTE**	**HellaS.**	**ARC-e**	**ARC-c**	**WinoG.**	**OBQA**	**LAMB.**	**Avg.**
Dense	67.15	60.20	78.96	47.44	72.45	34.60	76.50	62.47
Wanda (Sun et al., 2023)	57.76	46.95	68.86	34.81	66.54	27.20	65.55	52.52
RIA (Zhang et al., 2024)	58.12	48.29	69.53	34.90	67.80	26.60	67.90	53.31
CRePE	**60.29**	**50.63**	**70.20**	**35.92**	**68.82**	**30.40**	**68.78**	**55.01**

5.2 Main Results

5.2.1 Effect of the Sparsity Ratio

The benefit of CRePE grows as the pruning problem becomes harder (Table 2). At 50% unstructured sparsity, where post-training pruning is comparatively easy, the margin of CRePE over RIA is narrow (7.12 $\rightarrow$ 7.07 on LLaMA-1-7B and 6.81 $\rightarrow$ 6.74 on LLaMA-2-7B). In absolute PPL, the margin widens monotonically with the sparsity ratio, reaching 0.35 and 0.75 at 60% and 13.12 and 10.85 at 70%.

5.2.2 Semi-structured Pruning

CRePE attains the lowest perplexity among the evaluated methods without weight updates in all sixteen model and pattern combinations (Table 3), including LLaMA-2-7B (11.29 $\rightarrow$ 10.69) and LLaMA-2-13B (8.41 $\rightarrow$ 7.79) under 2:4. The average margin over RIA rises from 0.15 PPL at 4:8 to 0.43 PPL at 2:4, roughly threefold at the same 50% sparsity; since the sparsity ratio is fixed, this reflects the tightness of the structural constraint rather than how many weights are removed. These patterns are practically relevant, as they enable hardware acceleration via NVIDIA Sparse Tensor Cores (Mishra et al., 2021).

Sequence length. While we use a sequence length of 2048, Wanda (Sun et al., 2023) and BaWA (Liu et al., 2025) evaluate LLaMA-2 with 4096. The ranking of all methods is unchanged across the two settings (Table 4), and CRePE also outperforms BaWA, which tunes normalization exponents by block-wise zeroth-order search and which we reimplement following the paper as no official code is available. We further report results on Phi-4, Qwen3.5, and Mistral-Small 3 (Appendix Table 10), combination with 4-bit quantization (Appendix Table 11), and robustness to calibration and evaluation datasets (Appendix Table 12).

5.2.3 Zero-shot Performance

The perplexity gains carry over to downstream task accuracy (Table 5). On LLaMA-2-13B under 2:4 sparsity, CRePE attains an average accuracy of 55.01%, exceeding RIA by 1.70 points and Wanda by 2.49 points, and recovers roughly one fifth of the accuracy that RIA loses relative to the dense model. The improvement is not driven by a single high-variance task: CRePE outperforms both baselines on all seven tasks, with gains over RIA ranging from 0.67 points on ARC-Easy to 3.80 points on OpenBookQA, where RIA even falls below Wanda (26.60 vs. 27.20) while CRePE reaches 30.40.

Table 6: Cross-sparsity transferability of PHO hyperparameters on LLaMA-2-13B. Hyperparameters optimized on LLaMA-2-7B under each sparsity setting are directly applied to LLaMA-2-13B without re-tuning.

Model	**Sparsity**	**RIA**	**LLaMA-2-7B Best Hyperparameter**		
			un 50%	**2:4**	**4:8**
LLaMA-2-13B	un 50%	5.83	5.82	5.81	5.82
	2:4	8.41	8.09	8.03	8.08
	4:8	6.74	6.65	6.62	6.63

Table 7: Ablation on LLaMA-2-7B (WikiText-2 PPL). **(a)** Effect of kernel size k under 2:4 sparsity; the 2D neighborhood term accounts for 62% of the gain over RIA. **(b)** PHO budget analysis under 50% unstructured sparsity with CMA-ES.

(a) Kernel size

Method	PPL	ΔRIA	Δw/o conv
RIA ($\alpha = \beta = 1$)	11.29	—	—
CRePE w/o conv	11.06	−0.23	—
CRePE w/ 3×3	11.05	−0.24	−0.01
CRePE w/ 5×5	**10.69**	**−0.60**	**−0.37**
CRePE w/ 7×7	10.74	−0.55	−0.32
CRePE w/ 9×9	10.74	−0.55	−0.32

(b) PHO budget

	top_k		
n_trials	5	10	15
15	6.779 (8.6)	—	—
30	6.765 (10.1)	6.765 (17.1)	—
60	6.758 (13.3)	**6.740 (20.2)**	6.740 (27.3)
90	—	6.757 (23.4)	6.756 (30.4)
RIA	6.81 (1.8, no search)		

5.3 Ablation and Analysis

5.3.1 Effect of Kernel Size

Most of CRePE's gain comes from the 2D neighborhood term rather than from the adaptive coefficients alone (Table 7-(a)). On LLaMA-2-7B under 2:4 sparsity, tuning the coefficients without convolution (α, β only) lowers PPL from 11.29 to 11.06, and adding the neighborhood term with a 5×5 kernel lowers it further to 10.69. This additional reduction of 0.37 is $1.6\times$ the coefficient-only gain and accounts for 62% of the total improvement over RIA. The gain depends on the neighborhood size: a 3×3 kernel contributes almost nothing (0.01), while kernels larger than 5×5 perform slightly worse (10.74), possibly because larger neighborhoods approach a global average.

5.3.2 Analysis of PHO

The default budget (`n_trials`$= 60$, `top_k`$= 10$) attains the best PPL at the lowest cost among the configurations that reach it (Table 7-(b)). With `top_k` fixed at 5, increasing `n_trials` from 15 to 60 steadily improves perplexity ($6.779 \rightarrow 6.765 \rightarrow 6.758$), and enlarging `top_k` to 10 gives 6.740. Neither axis benefits from further enlargement: 90 trials does not improve on 60 (6.757), and `top_k`$= 15$ matches 6.740 at a 35% higher cost. If a tighter budget is required, (`n_trials`$= 30$, `top_k`$= 5$) halves the cost to about 10 minutes while retaining most of the gain (6.765). For reference, RIA requires no search and attains 6.81.

Pruning Efficiency Once the coefficients are determined, CRePE adds only a single masked 2D convolution to RIA's scoring pass, so its pruning cost is comparable to that of scoring-only methods. Zhang et al. (2024) report that RIA and Wanda prune LLaMA-2-70B in about 10 minutes, compared with about 90 minutes for SparseGPT, which requires Hessian computation.

Hyperparameter Transferability The one-time PHO search of about 20 minutes need not be repeated for every target model (Table 6). Hyperparameters optimized on LLaMA-2-7B under each sparsity setting and applied directly to LLaMA-2-13B outperform RIA in every setting (unstructured: $5.83 \rightarrow 5.81$–5.82; 2:4: $8.41 \rightarrow 8.03$–8.09; 4:8: $6.74 \rightarrow 6.62$–6.65), indicating that PHO hyperparameters transfer across model scales within the same family.

Table 8: Effect of combining CRePE with Channel Permutation (CP) and LSA CP.

Model	Method	2:4	2:4 + CP	2:4 + LSA	4:8	4:8 + CP	4:8 + LSA
LLaMA-2-7B (Dense: 5.47)	SparseGPT	11.01	10.43	9.97	8.48	8.14	8.11
	Wanda	12.07	11.65	11.12	8.60	8.39	8.20
	RIA	11.29	10.36	10.03	8.38	8.18	8.01
	CRePE	**10.69**	**10.20**	**9.80**	**8.16**	**8.00**	**7.85**
LLaMA-2-13B (Dense: 4.88)	SparseGPT	8.77	8.61	8.48	7.01	6.82	6.80
	Wanda	9.00	8.74	8.45	7.00	6.93	6.83
	RIA	8.41	8.12	7.67	6.74	6.62	6.53
	CRePE	**7.79**	**7.67**	**7.35**	**6.59**	**6.58**	**6.45**

Table 9: Combination of CRePE with existing techniques on LLaMA-1-7B (WikiText-2 PPL). **(a)** Non-uniform sparsity allocation at 70% unstructured sparsity. **(b)** Re-pruning with DSnoT under unstructured sparsity.

(a) Non-uniform sparsity allocation

Allocation	Wanda	RIA	CRePE
Uniform	86.09	86.07	**72.95**
OWL	24.29	24.76	**22.28**
AlphaPruning	24.00	24.07	**23.07**
DLP	21.04	20.99	**19.23**

(b) Re-pruning

	w/o DSnoT		w/ DSnoT	
Method	60%	70%	60%	70%
Wanda	10.71	86.09	10.44	70.44
RIA	10.41	86.07	10.19	71.39
CRePE	**10.06**	**72.95**	**9.98**	**67.82**

5.4 Compatibility with Existing Techniques

Channel permutation. The gains of CRePE are preserved when combined with Channel Permutation (CP) and LSA CP (Zhang et al., 2024) (Table 8). On LLaMA-2-7B, perplexity decreases from 10.69 (CRePE alone) to 10.20 with CP and 9.80 with LSA CP under 2:4, and from 8.16 to 8.00 and 7.85 under 4:8. In both cases, CRePE with LSA CP remains below RIA with the same permutation (10.03 and 8.01), and the same ordering holds on LLaMA-2-13B.

Non-uniform sparsity allocation. CRePE is complementary to layerwise sparsity allocation (Table 9-(a)). At 70% sparsity on LLaMA-1-7B, CRePE outperforms RIA with each of OWL (Yin et al., 2024), AlphaPruning (Lu et al., 2024), and DLP (Chen et al., 2025) (22.28, 23.07, and 19.23 vs. 24.76, 24.07, and 20.99), and CRePE + DLP attains the lowest perplexity among all configurations.

Re-pruning. CRePE also combines with the re-pruning method DSnoT (Zhang et al., 2023) (Table 9-(b)). On LLaMA-1-7B, CRePE + DSnoT attains the lowest perplexity among all combined methods at both 60% (9.98) and 70% sparsity (67.82). The gain from DSnoT is smaller for CRePE (0.08 and 5.13) than for Wanda and RIA (0.22–0.27 and 14.68–15.65), suggesting that CRePE's initial mask leaves less error for re-pruning to correct.

6 Conclusion

We presented CRePE, which extends Relative Importance scoring with a 2D neighborhood term and adaptive per-term coefficients, and PHO, which uses the Gini coefficient of pruned importance scores as a surrogate for perplexity to reduce the hyperparameter search from 11 hours to about 20 minutes with nearly the same perplexity. Across LLaMA-1/2/3, CRePE consistently improves over scoring-only baselines, and its margin grows under tighter structural constraints such as 2:4 sparsity. The gains hold across sequence lengths and additional model families, and CRePE combines with channel permutation, non-uniform sparsity allocation, re-pruning, and quantization, making it a drop-in replacement for existing importance scores.

AI USE STATEMENT

In this work, we used generative AI tools (large language model assistants) for editing and condensing the manuscript text, for aggregating measured results into tables, and for feedback on the framing of the paper's claims. All method design, implementation, experiments, measurements and numerical results are the authors' own; every AI-assisted passage and table was reviewed and verified against the underlying data by the authors, who take full responsibility for the content.

ETHICS STATEMENT

This work uses only publicly released models and datasets (WikiText-2, C4, PTB, and standard zero-shot benchmarks); no new data involving human subjects was collected. The method compresses model weights for efficient inference and does not alter model outputs beyond the accuracy changes reported here.

REPRODUCIBILITY STATEMENT

The CRePE scoring function is defined in Sec. 3, and the PHO procedure, search ranges, and random seeds are described in Sec. 4.3 and Sec. 5.1. All models and datasets are publicly available. Code will be released upon publication.

REFERENCES

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael Harrison, Russell J. Hewett, Mojan Javaheripi, Piero Kauffmann, James R. Lee, Yin Tat Lee, Yuanzhi Li, Weishung Liu, Caio C. T. Mendes, Anh Nguyen, Eric Price, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Xin Wang, Rachel Ward, Yue Wu, Dingli Yu, Cyril Zhang, and Yi Zhang. Phi-4 technical report, 2024. URL `https://arxiv.org/abs/2412.08905`.

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

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*, 2024.

Vladimír Boža. Fast and effective weight update for pruned large language models. *arXiv preprint arXiv:2401.02938*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Lidia Ceriani and Paolo Verme. The origins of the gini index: extracts from variabilità e mutabilità (1912) by corrado gini. *The Journal of Economic Inequality*, 10(3):421–443, 2012.

Yuli Chen, Bo Cheng, Jiale Han, Yingying Zhang, Yingting Li, and Shuhao Zhang. Dlp: Dynamic layerwise pruning in large language models. *arXiv preprint arXiv:2505.23807*, 2025.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

Peijie Dong, Lujun Li, Zhenheng Tang, Xiang Liu, Xinglin Pan, Qiang Wang, and Xiaowen Chu. Pruner-zero: Evolving symbolic pruning metric from scratch for large language models. *arXiv preprint arXiv:2406.02924*, 2024.

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

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In *Advances in neural information processing systems*, pp. 1135–1143, 2015.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*, 2023.

Lian Liu, Xiandong Zhao, Guanchen Li, Dong Li, Mengdi Wang, Yinhe Han, Xiaowei Li, et al. Bawa: Automatic optimizing pruning metric for large language models with balanced weight and activation. In *Forty-second International Conference on Machine Learning*, 2025.

Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael W Mahoney, and Yaoqing Yang. Alphapruning: Using heavy-tailed self regularization theory for improved layer-wise pruning of large language models. *Advances in neural information processing systems*, 37:9117–9152, 2024.

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

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

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect. *arXiv preprint arXiv:2403.03853*, 2024.

Xiang Meng, Kayhan Behdin, Haoyue Wang, and Rahul Mazumder. Alps: Improved optimization for highly sparse one-shot pruning for large language models. *Advances in Neural Information Processing Systems*, 37:37594–37625, 2024.

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

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 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*, pp. 2381–2391, 2018.

Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *arXiv preprint arXiv:2104.08378*, 2021.

Mistral AI. Mistral small 3. https://mistral.ai/news/mistral-small-3, 2025.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 1525–1534, 2016.

Qwen Team. Qwen3.5: Towards native multimodal agents, February 2026. URL https://qwen.ai/blog?id=qwen3.5.

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

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 8732–8740, 2020.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. Sleb: Streamlining llms through redundancy verification and elimination of transformer blocks. In *International Conference on Machine Learning*, pp. 46136–46155. PMLR, 2024.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models, 2023b. URL https://arxiv.org/abs/2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

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

Yifei Yang, Zouying Cao, and Hai Zhao. Laco: Large language model pruning via layer collapse. *arXiv preprint arXiv:2402.11187*, 2024.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, et al. Outlier weighed layerwise sparsity (owl) a missing secret sauce for pruning llms to high sparsity. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 57101–57115, 2024.

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

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. In *12th International Conference on Learning Representations (ICLR 2024)*, 2024.

Yuxin Zhang, Lirui Zhao, Mingbao Lin, Yunyun Sun, Yiwu Yao, Xingjia Han, Jared Tanner, Shiwei Liu, and Rongrong Ji. Dynamic sparse no training: Training-free fine-tuning for sparse llms. *arXiv preprint arXiv:2310.08915*, 2023.

Pengxiang Zhao, Hanyu Hu, Ping Li, Yi Zheng, Zhefeng Wang, and Xiaoming Yuan. Fistapruner: Layer-wise post-training pruning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 29457–29475, 2025.

Table 10: WikiText-2 perplexity under 2:4 and 4:8 sparsity on additional model families. **Bold** marks the best result in each column within each sparsity pattern.

	Method	Phi-4	Qwen3.5		Mistral-Small 3
		14B	0.8B	9B	24B
Dense		6.46	17.19	8.64	5.29
2:4	Magnitude	14.44	3281.00	33.20	16.50
	Wanda	10.84	126.34	16.00	19.79
	RIA	10.51	125.90	14.60	15.59
	CRePE	**10.45**	**120.83**	**14.34**	**11.84**
4:8	Magnitude	11.31	779.81	27.50	14.38
	Wanda	9.05	64.13	11.67	11.40
	RIA	8.97	61.73	11.29	11.07
	CRePE	**8.95**	**61.14**	**11.15**	**9.34**

Table 11: Combination with post-training quantization. Weights are quantized to 4 bits with a group size of 128 after pruning. Dense denotes the unpruned model with quantization only. **Bold** marks the best result in each column among the pruning methods.

	Method	No quant.	+GPTQ	+AWQ
	LLaMA-2-7B			
Dense		5.47	5.68	5.64
2:4	Wanda	12.07	14.48	12.62
	RIA	11.29	14.41	11.70
	CRePE	**10.69**	**13.04**	**11.45**
4:8	Wanda	8.60	9.63	8.95
	RIA	8.38	10.18	8.63
	CRePE	**8.16**	**9.26**	**8.50**
	LLaMA-2-13B			
Dense		4.88	4.99	5.02
2:4	Wanda	9.00	9.51	9.41
	RIA	8.41	8.65	8.68
	CRePE	**7.79**	**7.98**	**8.07**
4:8	Wanda	7.00	7.16	7.16
	RIA	6.74	6.84	6.91
	CRePE	**6.59**	**6.73**	**6.77**

A Appendix

Generalization Across Architectures. Table 10 evaluates CRePE on Phi-4 (14B) (Abdin et al., 2024), Qwen3.5 (0.8B, 9B) (Qwen Team, 2026), and Mistral-Small 3 (24B) (Mistral AI, 2025) under 2:4 and 4:8 sparsity. CRePE attains the lowest perplexity in all eight model and pattern combinations. The gain is most pronounced on Mistral-Small 3, where activation-based scoring alone is insufficient under 2:4: Wanda (Sun et al., 2023) (19.79) falls behind even magnitude pruning (16.50), whereas CRePE reduces perplexity to 11.84, an improvement of 3.75 over RIA (Zhang et al., 2024). These results confirm that CRePE generalizes beyond the LLaMA family without any architecture-specific modification.

Combination with Post-training Quantization. Table 11 reports perplexity when 4-bit weight quantization (GPTQ (Frantar et al., 2022) or AWQ (Lin et al., 2023), group size 128) is applied to the pruned models. CRePE attains the lowest perplexity in all twelve model, pattern, and quantization combinations, and its margin over RIA is preserved after quantization. On LLaMA-2-7B, RIA degrades more than the other two methods under GPTQ (11.29 $\rightarrow$ 14.41 at 2:4 and 8.38 $\rightarrow$ 10.18 at 4:8), and at 4:8 it falls behind Wanda (10.18 vs. 9.63) despite being lower without quantization.

On LLaMA-2-13B the degradation is far smaller for all three methods (at most 0.51 PPL), and the ranking is unchanged by quantization; CRePE quantized with GPTQ at 2:4 (7.98) still attains lower perplexity than RIA without any quantization (8.41), and the same holds at 4:8 (6.73 vs. 6.74). CRePE remains the best in every case, indicating that its advantage does not disappear when pruning is combined with quantization.

Table 12: Robustness to the choice of calibration and evaluation datasets under 2:4 and 4:8 sparsity. Each column corresponds to one (evaluation, calibration) pair; the WikiText-2 evaluation with C4 calibration matches the setting used in the main paper. **Bold** marks the best result in each column within each model and sparsity pattern.

Eval. dataset			**PTB**			**WikiText-2**			**C4**		
Calib. dataset			**WikiText-2**	**C4**	**PTB**	**WikiText-2**	**C4**	**PTB**	**WikiText-2**	**C4**	**PTB**
LLaMA-2 -7B	Dense		37.91			5.47			7.26		
	2:4	Wanda	223.86	219.34	219.59	11.39	12.07	12.42	16.29	15.52	16.27
		RIA	214.01	225.03	200.41	10.73	11.29	11.15	14.88	14.76	14.90
		CRePE	**201.95**	**207.25**	**185.60**	**10.38**	**10.69**	**10.80**	**13.93**	**13.82**	**14.03**
	4:8	Wanda	**105.46**	114.99	100.71	8.21	8.60	8.65	11.45	11.33	11.58
		RIA	114.70	121.71	106.10	8.11	8.38	8.35	11.10	11.08	11.15
		CRePE	109.29	**113.20**	**97.44**	**7.94**	**8.16**	**8.12**	**10.85**	**10.79**	**10.84**
LLaMA-2 -13B	Dense		50.94			4.88			6.73		
	2:4	Wanda	169.44	159.89	135.25	8.30	9.00	8.88	12.58	12.46	12.45
		RIA	144.62	142.34	131.10	8.02	8.41	8.31	11.60	11.47	11.63
		CRePE	**133.12**	**135.16**	**118.95**	**7.62**	**7.79**	**7.72**	**10.66**	**10.58**	**10.66**
	4:8	Wanda	102.13	105.28	**85.02**	6.70	7.00	6.90	9.87	9.71	9.73
		RIA	96.52	98.08	93.87	6.52	6.74	6.74	9.38	9.33	9.42
		CRePE	**92.10**	**93.88**	87.50	**6.46**	**6.59**	**6.57**	**9.21**	**9.17**	**9.18**

Robustness to Calibration and Evaluation Datasets. The gains of CRePE are not specific to C4 calibration and WikiText-2 evaluation (Table 12). Across all nine combinations of three calibration sets (WikiText-2 (Merity et al., 2016), C4 (Raffel et al., 2020), PTB (Marcus et al., 1993)) and three evaluation sets on LLaMA-2-7B and LLaMA-2-13B under 2:4 and 4:8 sparsity, CRePE attains the lowest perplexity in 34 of the 36 cells and in every cell under WikiText-2 and C4 evaluation. The two exceptions occur under 4:8 with PTB evaluation, where Wanda is lower; PTB evaluation is also the only setting in which Wanda outperforms RIA.