

PRISM: Post-Training LLM Pruning via Residual Importance Signal Merging and Intra-Block Sparsity Allocation

Keyu Wen and Jianing Zhu and Kaiyang Hu and Lijun Yang and Rui Lin

JD.com, China

{wenkeyu1, zhujianing.loobee, hukaiyang, cdyanglijun, linrui}@jd.com

Abstract

Post-training pruning compresses Large Language Models (LLMs) without full-model re-training, but performance degrades sharply at high sparsity. We identify two limitations in existing methods: local importance estimates miss global spectral structure, and uniform within-block sparsity ignores heterogeneous sensitivity across transformer projections. We propose **PRISM**, a post-training pruning framework with training-free mask construction and optional fixed-mask reconstruction optimization. PRISM augments activation-based scoring with two adaptive-rank SVD-derived signals for principal-subspace and residual-aware importance, combines the factors through parameter-free CV-based fusion, and allocates sparsity non-uniformly across attention and MLP projections using a gradient-weighted matrix-level sensitivity proxy. Across six Qwen3 and LLaMA models, PRISM consistently outperforms existing pruning baselines, yielding up to 16.8% relative improvement in WikiText-2 perplexity under the evaluated sparsity settings.

1 Introduction

The proliferation of Large Language Models (LLMs) has revolutionized natural language processing, with models like GPT-4 (OpenAI, 2023), LLaMA-3.1 (Grattafiori et al., 2024), and Qwen3 (Team, 2025) demonstrating unprecedented capabilities across diverse tasks. However, this performance comes at the cost of massive parameter counts—ranging from billions to trillions of parameters—creating substantial barriers to deployment in resource-constrained environments (Frantar and Alistarh, 2023; Sun et al., 2024).

Model compression techniques, including quantization (Lin et al., 2024; Frantar et al., 2023), pruning (Frantar and Alistarh, 2023; Sun et al., 2024), and low-rank decomposition (Hsu et al., 2022; Yuan et al., 2023), have emerged as essential

strategies for reducing computational and memory requirements. Among these, **post-training pruning** is particularly attractive as it eliminates the need for expensive retraining while achieving significant compression ratios (Sun et al., 2024; Frantar and Alistarh, 2023). At high sparsity, however, current post-training pruning methods still suffer from two core bottlenecks:

(a) Local Scoring Without Global Structural Awareness. Methods like Wanda (Sun et al., 2024) score weights using per-channel activation norms ($|W_{ij}| \cdot \|X_j\|_2$) and do not explicitly model off-diagonal inter-feature correlations when constructing the mask. SparseGPT (Frantar and Alistarh, 2023) incorporates approximate Hessian information, but its row-by-row greedy update still solves a *local* regression for each row independently—it does not model the *global spectral structure* shared across all rows of the weight matrix. In transformer layers, correlated input features can make a global structural prior useful for identifying weights that are individually unremarkable yet jointly important. This bottleneck becomes harder once multiple complementary signals are available: recent training-free adaptive fusion methods such as HyWIA (Liu et al., 2025) combine fine- and coarse-grained importance assessments for structured pruning, while hand-tuned fixed mixtures increase adaptation cost and can generalize poorly across matrices. A direct, matrix-specific rule for fusing activation and spectral element-wise cues remains underexplored.

(b) Matrix-Level Allocation. OWL (Yin et al., 2024) uses outlier-weighted sparsity allocation and reports heterogeneous sparsity across projections, while BESA (Xu et al., 2024) uses blockwise reconstruction with differentiable masks and can learn sparsity for individual linear weights. PRISM uses SVD/CV signals and a gradient-weighted matrix-level sensitivity proxy to redistribute retention across the seven transformer projections ($\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o$ in atten-

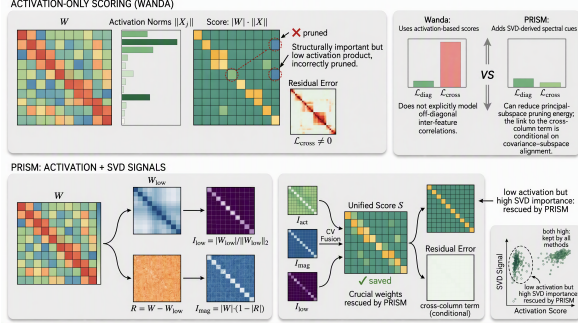


Figure 1: PRISM combines per-channel activation scores with SVD-derived spectral cues. The link from lower principal-subspace pruning energy to a lower cross-column term is conditional on covariance-subspace alignment (Section 3.6).

tion and $W_{\text{up}}, W_{\text{gate}}, W_{\text{down}}$ in MLP). Its gradient-weighted block-local allocation is complementary to AlphaPruning’s default block-level allocation; AlphaPruning also studies per-matrix and mixed variants in its appendix. Our ablations compare PRISM with a uniform within-block baseline rather than claiming that prior allocation methods lack within-block heterogeneity.

Figure 1 contrasts the first bottleneck with PRISM’s remedy: weights that look individually unremarkable can still be structurally important, and PRISM preserves them by adding SVD-derived importance signals. To address these two bottlenecks, we propose **PRISM** (Post-training pRuning via resIdual importance Signal Merging). The method is organized around two core ideas plus one enabling mechanism:

1. **Spectral-Aware Multi-Factor Scoring.** We introduce two complementary importance signals derived from adaptive-rank SVD that expose the global spectral structure of weight matrices—information invisible to both activation-based scoring (Wanda) and row-local Hessian regression (SparseGPT). By decomposing $W = W_{\text{low}} + R$, we derive: (a) *residual-gated magnitude*, a matrix-normalized measure that highlights weights well reconstructed by the low-rank component, and (b) *low-rank component importance*, which highlights weights lying in the principal singular subspace. These two signals capture complementary aspects of spectral structure. Unlike SVD-based compression methods that replace weight matrices with low-rank factorizations, PRISM uses SVD only for one-shot importance estimation and retains the origi-

nal weight structure.

2. **CV-Based Parameter-Free Fusion.** To combine activation and spectral cues without auxiliary training, we derive matrix-level fusion weights from the coefficient of variation (CV) of the candidate factors. This simple training-free rule adaptively reweights the three signals for each matrix and requires neither learnable parameters nor grid search (Section 3.3).
3. **Gradient-Weighted Intra-Block Allocation.** We allocate heterogeneous sparsity *within* each transformer block, across its 7 projection matrices ($W_q, W_k, W_v, W_o, W_{\text{up}}, W_{\text{gate}}, W_{\text{down}}$). This gradient-weighted block-local allocator is designed to reduce over-pruning of high-impact matrices and can operate under an external block-level budget.

Extensive evaluations across six Qwen3 and LLaMA models show consistent gains for PRISM under unstructured and N:M sparsity.

2 Related Work

Post-training pruning removes weights from pre-trained models without retraining, making it attractive for billion-scale LLMs. Wanda (Sun et al., 2024) scores weights by $|W_{ij}| \cdot \|X_j\|_2$, and Wanda++ (Yang et al., 2025) augments Wanda’s activation-based score with regional-gradient information and further introduces regional optimization. SparseGPT (Frantar and Alistarh, 2023), Thanos (Ilin and Richtárik, 2025), and FISTAPruner (Zhao et al., 2025) incorporate richer second-order or structured approximations, but they still operate locally rather than modeling the global spectral structure shared across rows of a weight matrix. SVD-based compression methods such as ASVD (Yuan et al., 2023), SVD-LLM (Wang et al., 2025b), and Dobi-SVD (Wang et al., 2025a), as well as sparse-low-rank hybrids such as HASSLE-free (Makni et al., 2025) and SALR (Zhang et al., 2026), exploit spectral structure by changing the deployed representation. PRISM instead uses SVD only as a one-shot diagnostic for importance estimation and keeps the original sparse architecture unchanged.

Combining multiple pruning signals is itself non-trivial. HyWIA (Liu et al., 2025) uses training-free adaptive attention to blend fine- and coarse-grained importance assessments for structured LLM pruning, UNComp (Xiong et al., 2025) uses entropy-

style uncertainty measures for KV-cache compression, and MaskLLM (Fang et al., 2024) learns adaptive semi-structured masks end-to-end. In contrast, our CV-based fusion directly sets matrix-level weights for activation and spectral element-wise cues from the relative dispersion of the candidate factors.

For non-uniform sparsity allocation, OWL (Yin et al., 2024) uses outlier-weighted statistics and reports heterogeneous sparsities across projections, while BESA (Xu et al., 2024) uses blockwise reconstruction and differentiable masks to learn sparsity for individual linear weights. PRISM uses matrix-level SVD/CV signals and a gradient-weighted sensitivity proxy to redistribute retention across $\mathbf{W}_q, \mathbf{W}_k, \mathbf{W}_v, \mathbf{W}_o, \mathbf{W}_{\text{up}}, \mathbf{W}_{\text{gate}}, \mathbf{W}_{\text{down}}$. It is complementary to AlphaPruning (Lu et al., 2024), whose default is block-level allocation and whose appendix also studies per-matrix and mixed allocation.

3 Methodology

3.1 Preliminaries and Problem Formulation

Figure 2 illustrates the complete PRISM pipeline. Consider an LLM with L transformer layers. For layer l , let $\mathbf{W}^{(l)} \in \mathbb{R}^{m_l \times n_l}$ denote a weight matrix (e.g., query projection in attention or up-projection in FFN). Given a calibration dataset $\mathcal{D}_{\text{cal}} = \{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^N$, our goal is to find a sparse mask $\mathbf{M}^{(l)} \in \{0, 1\}^{m_l \times n_l}$ for each layer such that:

$$\begin{aligned} \min_{\{\mathbf{M}^{(l)}\}} \quad & \mathcal{L}\left(f_{\text{pruned}}(\mathbf{X}; \{\mathbf{M}^{(l)} \odot \mathbf{W}^{(l)}\}), \mathbf{y}\right) \\ \text{s.t.} \quad & \frac{\|\mathbf{M}^{(l)}\|_0}{m_l \cdot n_l} = 1 - s^{(l)} \end{aligned} \quad (1)$$

where $s^{(l)}$ is the sparsity ratio for layer l and $\mathcal{L}$ is the language modeling loss.

3.2 Adaptive-Rank SVD for Multi-Signal Construction

Low-rank and sparse-low-rank decompositions have been used to compress LLM weight matrices (Hsu et al., 2022; Makni et al., 2025). Motivated by this structure, we write each matrix as a low-rank component plus a residual. This decomposition has a direct consequence for pruning: the layer reconstruction error $\mathcal{L} = \text{tr}(\Delta \Sigma_X \Delta^T)$ separates into an element-wise term and a *cross-column correlation term* that depends on how pruned weights interact through the input covariance; Section 3.6

summarizes this mechanism and Appendix F gives the full derivation.

Wanda’s per-channel activation statistics align with the element-wise term ($\mathcal{L}_{\text{diag}}$) in our reconstruction-error decomposition, but do not explicitly represent off-diagonal inter-feature correlations. SparseGPT incorporates the full input covariance Σ_X via Hessian-based regression, but it processes each *row* of $\mathbf{W}$ independently in a greedy column-by-column update—the resulting mask for row i does not account for the spectral structure *shared across all rows* of $\mathbf{W}$. The SVD of $\mathbf{W}$ captures this shared structure: the right singular vectors $\mathbf{V}_k$ span the principal directions along which all rows concentrate their energy. By extracting per-element importance from $\mathbf{V}_k$, PRISM provides a *global structural prior* that complements both Wanda’s per-channel score and SparseGPT’s row-local Hessian through a one-shot offline scoring step. The end-to-end offline pruning cost, including the optional RO stage in Section 3.5, is reported separately in Appendix G.

ASVD (Yuan et al., 2023) also uses sensitivity-based truncation-rank search for layer-specific low-rank compression. PRISM uses a fixed-probe, sensitivity-guided rank mapping solely to form SVD-derived pruning-importance signals while preserving the original sparse architecture; we evaluate this choice as a component rather than claiming a new rank-selection principle.

3.2.1 Layer Sensitivity Analysis

We first compute the sensitivity of each layer to rank reduction at a fixed probe rank $\bar{k} = k_{\min}$:

$$\mathcal{S}^{(l)} = \frac{\|\mathbf{W}^{(l)}\|_F^2 - \|\mathbf{W}_{\text{low}}^{(l)}(\bar{k})\|_F^2}{\|\mathbf{W}^{(l)}\|_F^2} \cdot \left\| \frac{\partial \mathcal{L}}{\partial \mathbf{W}^{(l)}} \right\|_F \quad (2)$$

where the first term measures the relative reconstruction error and the second term (gradient norm) captures the layer’s importance for the task. Higher $\mathcal{S}^{(l)}$ indicates the layer is more sensitive to compression and should retain more parameters. Using a fixed probe rank makes sensitivities comparable across matrices, keeping the procedure one-shot.

3.2.2 Sensitivity-Guided Rank Mapping for Importance Estimation

For importance estimation, we map sensitivity to an adaptive rank $k^{(l)}$ for layer l :

$$k^{(l)} = k_{\min} + (k_{\max} - k_{\min}) \cdot \frac{\mathcal{S}^{(l)} - \mathcal{S}_{\min}}{\mathcal{S}_{\max} - \mathcal{S}_{\min}} \quad (3)$$

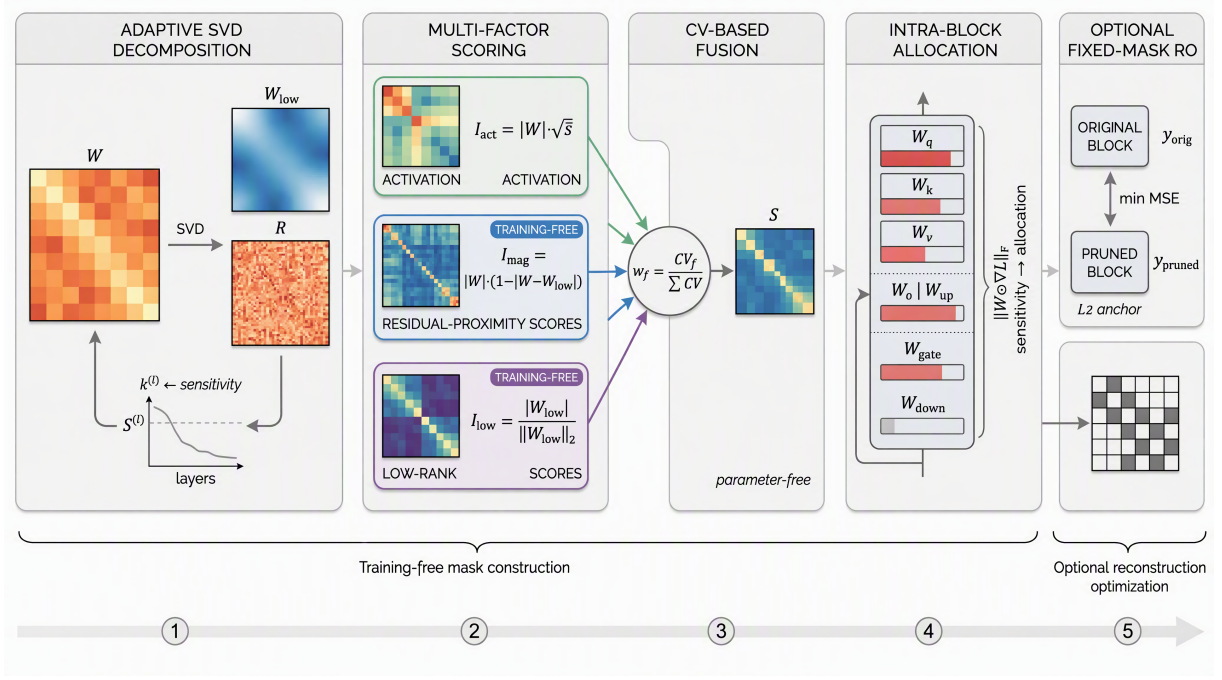


Figure 2: PRISM pipeline. Stages 1–4 construct the mask without training; optional fixed-mask RO then updates surviving weights through reconstruction optimization.

where $k_{\max} = 0.5 \times \min(m, n)$ and $k_{\min} = 0.1 \times \min(m, n)$. In implementation, the mapped value is converted to an integer by truncation and clipped to the feasible interval $[k_{\min}, k_{\max}]$.

Rank-selection intuition. Higher sensitivities should receive larger ranks: under a smoothness bound, the loss increase from SVD truncation grows with both the residual norm and the layer gradient, and both are reflected in $S^{(l)}$. Therefore, we use the monotone mapping in Eq. (3); Appendix E gives the full directional argument.

Definition 1 (SVD-Derived Importance Signals). *Given the selected rank $k^{(l)}$ and the corresponding rank- $k^{(l)}$ truncation $\mathbf{W}_{\text{low}}^{(l)}$, the adaptive SVD yields two complementary per-element importance signals (used as $\mathbf{I}_{\text{mag}}$ and $\mathbf{I}_{\text{low}}$ in Section 3.3):*

$$\mathbf{I}_{\text{mag},ij}^{(l)} = |\mathbf{W}_{ij}^{(l)}| \cdot \max\left(1 - \frac{|\mathbf{W}_{ij}^{(l)} - \mathbf{W}_{\text{low},ij}^{(l)}|}{\sigma_W^{(l)} + \epsilon_0}, 0\right) \quad (4)$$

$$\mathbf{I}_{\text{low},ij}^{(l)} = \frac{|\mathbf{W}_{\text{low},ij}^{(l)}|}{\|\mathbf{W}_{\text{low}}^{(l)}\|_F + \epsilon_0} \quad (5)$$

where $\sigma_W^{(l)} = \text{std}(\mathbf{W}^{(l)})$ is the weight standard deviation, and $\|\cdot\|_F$ denotes the Frobenius norm. $\mathbf{I}_{\text{mag}}$ is a residual-gated magnitude score built from a matrix-normalized reconstruction gap. Dividing $|\mathbf{W}_{ij} - \mathbf{W}_{\text{low},ij}|$ by $\sigma_W^{(l)} + \epsilon_0$ makes the

gating term dimensionless and less sensitive to matrix-wise weight rescaling, while the outer factor $|\mathbf{W}_{ij}|$ preserves absolute salience. $\mathbf{I}_{\text{low}}$ measures principal subspace membership: weights with large Frobenius-normalized low-rank magnitude lie within the dominant singular directions and should be preserved during pruning.

Complementarity of the two signals. A weight can have large $\mathbf{I}_{\text{low}}$ but moderate $\mathbf{I}_{\text{mag}}$: this occurs when the low-rank component is large at that position (the weight lies in the principal subspace) yet the high-rank residual is also non-negligible, so the normalized reconstruction gap remains sizeable. Conversely, a weight can have large $\mathbf{I}_{\text{mag}}$ but small $\mathbf{I}_{\text{low}}$: this occurs when the weight is small in absolute value yet almost entirely captured by the low-rank part, making the normalized reconstruction gap small and the gate close to 1 while $|\mathbf{W}_{\text{low},ij}|$ itself remains small. Thus $\mathbf{I}_{\text{mag}}$ and $\mathbf{I}_{\text{low}}$ identify overlapping but distinct subsets of structurally important weights. Both signals are designed to reduce the principal-subspace pruning energy defined in Eq. (23), which, under the alignment condition summarized in Section 3.6 and detailed in Appendix F, is associated with lower cross-column reconstruction error.

3.3 CV-Based Parameter-Free Fusion

We fuse three per-element importance factors: activation importance following Wanda,

$$\mathbf{I}_{\text{act},ij}^{(l)} = |\mathbf{W}_{ij}^{(l)}| \cdot \sqrt{\bar{s}_j^{(l)}} \quad (6)$$

where $\bar{s}_j^{(l)} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{X}_{i,j}^{(l)}\|_2^2$ is the average squared activation norm of input channel j over the N calibration samples; and the two SVD-derived factors $\mathbf{I}_{\text{mag}}$ and $\mathbf{I}_{\text{low}}$ defined in Eqs. (4) and (5). The matrix-normalized construction in Eq. (4) reduces within-matrix scale sensitivity. We do not further standardize the three raw factors before summation; instead, we use the coefficient of variation (CV) to derive matrix-level adaptive weights from each factor’s relative dispersion:

$$\text{CV}_f^{(l)} = \frac{\text{std}(\mathbf{I}_f^{(l)})}{\text{mean}(\mathbf{I}_f^{(l)}) + \epsilon_0}, \quad (7)$$

$$f \in \mathcal{F} = \{\text{act}, \text{mag}, \text{low}\}$$

CV is scale-invariant, so it remains comparable as a measure of relative spread across factors. We therefore assign matrix-level fusion weights directly proportional to CV:

$$w_f^{(l)} = \frac{\text{CV}_f^{(l)}}{\sum_{f' \in \mathcal{F}} \text{CV}_{f'}^{(l)}}, \quad \epsilon_0 = 10^{-6} \quad (8)$$

Higher-CV factors receive larger weight, while nearly uniform factors are down-weighted automatically. Since the raw factor values are still summed directly, CV fusion should be viewed as a training-free adaptive reweighting heuristic. The final unified score is

$$\mathcal{S}_{ij}^{(l)} = \sum_{f \in \mathcal{F}} w_f^{(l)} \cdot \mathbf{I}_{f,ij}^{(l)} \quad (9)$$

which is computed independently for each weight matrix and used directly to generate the sparse mask $\mathbf{M}^{(l)}$. Section 4.4 shows this CV-based reweighting outperforms uniform, standard-deviation-proportional, and global hand-tuned fusion.

3.4 Gradient-Weighted Intra-Block Allocation

Each transformer block b contains $T=7$ projection matrices $\{\mathbf{W}_t^{(b)}\}_{t \in \mathcal{T}}$ with $\mathcal{T} = \{q, k, v, o, \text{up}, \text{gate}, \text{down}\}$. Instead of

assigning them a uniform sparsity ratio, we allocate retention within each block, a granularity that shares a block budget while remaining compatible with external inter-layer allocation.

For each matrix t in block b , we define local importance as the block-normalized RMS unified score from Eq. (9):

$$\mathcal{I}_{\text{local}}^{(b,t)} = \frac{\|\mathcal{S}^{(b,t)}\|_F / \sqrt{|\mathbf{W}^{(b,t)}|}}{\sum_{t' \in \mathcal{T}} \|\mathcal{S}^{(b,t')}\|_F / \sqrt{|\mathbf{W}^{(b,t')}|}} \quad (10)$$

We define a matrix-level gradient-weight sensitivity proxy as

$$\mathcal{I}_{\text{sens}}^{(b,t)} = \left\| \mathbf{W}^{(b,t)} \odot \nabla_{\mathbf{W}^{(b,t)}} \mathcal{L} \right\|_F, \quad (11)$$

which is inspired by Fisher-style saliency measures but does not estimate the Fisher matrix or its diagonal. We then combine the two signals by a block-wise normalized geometric mean:

$$\mathcal{I}^{(b,t)} = \left(\tilde{\mathcal{I}}_{\text{local}}^{(b,t)} \right)^\lambda \cdot \left(\tilde{\mathcal{I}}_{\text{sens}}^{(b,t)} \right)^{1-\lambda} \quad (12)$$

where $\tilde{\cdot}$ denotes normalization within block b and $\lambda = 0.8$. We use $\lambda > 0.5$ because local importance already aggregates all three fused element-wise signals, while the gradient-weight sensitivity proxy provides a complementary matrix-level correction.

We convert $\mathcal{I}^{(b,t)}$ into matrix budgets by ranking the seven matrices within each block. Higher-ranked matrices receive larger initial retention through a bounded linear schedule over $[r_{\min}, r_{\max}]$; a block-wise scaling factor then enforces the target retention, and each matrix is pruned using the unified score $\mathcal{S}^{(b,t)}$. This rank-based schedule avoids relying on uncalibrated absolute scales across projection types while preserving the sensitivity ordering. In the default setting, the target is the global uniform budget; with an external inter-layer allocator such as AlphaPruning, PRISM instead redistributes its layer-wise target within each block, as evaluated in Section 4.4. Algorithm 1 in Appendix A.1 gives the full procedure.

3.5 Regional Optimization

We adopt Wanda++’s Regional Optimization (RO) (Yang et al., 2025) as an optional post-pruning reconstruction step, but use a fixed-mask variant tailored to PRISM. Unlike Wanda++, we do not alternate re-pruning and weight updates: once the mask is produced by the SVD-derived scoring, CV fusion, and intra-block allocation stages, RO updates only the surviving weights. To stabilize this

fixed-mask setting, we add an L2 anchor that keeps the updated weights close to their pre-pruning values. The resulting per-block objective is

$$\min_{\{\mathbf{W}_{\text{surv}}^{(b)}\}} \frac{1}{B} \sum_{j=1}^B \left\| f_b(\mathbf{x}_j; \mathbf{M}^{(b)} \odot \mathbf{W}^{(b)}) - f_b(\mathbf{x}_j; \mathbf{W}_{\text{orig}}^{(b)}) \right\|^2 + \mu \left\| \mathbf{W}_{\text{surv}}^{(b)} - \mathbf{W}_{\text{orig, surv}}^{(b)} \right\|^2, \quad (13)$$

where gradients and updates are applied to non-zero weights. This step is optional and orthogonal to the scoring stage; implementation and cost are deferred to Appendix A and Appendix G.

3.6 Analysis of SVD-Guided Pruning

We summarize the mechanism and defer proofs to Appendix F. The analysis is *conditional*: it does not claim global optimality, but identifies when SVD-derived signals can reduce reconstruction error beyond activation-only scoring. Appendix B tests the central condition and two observable consequences.

Pruning error decomposes into a diagonal term and a cross-column term from input-feature correlations. Activation-only scoring mainly targets the diagonal term, whereas the SVD-derived signals discourage pruning weights in the dominant right-singular subspace. This motivates the *principal-subspace pruning energy* $E_{\text{prin}}(M) = \|\Delta V_k\|_F^2$.

When off-diagonal input covariance aligns with this subspace, lower $E_{\text{prin}}(M)$ can imply lower cross-column error. Thus I_{act} targets diagonal reconstruction cost, while I_{mag} and I_{low} provide complementary spectral terms. Section 4.4 measures principal-subspace pruning energy, while Appendix B additionally reports covariance–subspace alignment, post-mask cross-term decomposition, and hidden-state preservation. These diagnostics support the proposed mechanism in aggregate but do not establish a universal per-matrix guarantee.

4 Results and Analysis

4.1 Experimental Setup

We evaluate PRISM on Qwen3 (0.6B, 4B, 8B) (Team, 2025), LLaMA-3.1 8B (Grattafiori et al., 2024), and LLaMA-3.2 (1B, 3B) (Meta AI, 2024). The main tables use calibration seed 42 with 128 C4 sequences of length 1024 (Raffel et al., 2020; Sun et al., 2024; Frantar and Alistarh, 2023); Appendix C reports calibration-subset variance over

seeds 42, 123, and 256. We report WikiText-2 perplexity (Merity et al., 2016) and zero-shot accuracy on HellaSwag (Zellers et al., 2019), ARC-easy/challenge (Clark et al., 2018), Winogrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020), and BoolQ (Clark et al., 2019). We compare against SparseGPT (Frantar and Alistarh, 2023), Wanda (Sun et al., 2024), Thanos (Ilin and Richtárik, 2025), AlphaPruning (Lu et al., 2024), and Wanda++ (Yang et al., 2025); all baseline results are rerun under identical calibration samples, sparsity targets, and evaluation code. Because inter-layer allocation requires adjustable layer-wise budgets, AlphaPruning is reported only at 50% unstructured sparsity and omitted for fixed N:M sparsity. In Table 1, Wanda++ and PRISM use RO by default, while “w/o RO” disables reconstruction to isolate pruning quality. Table 3(e) contrasts PRISM and PRISM + AlphaPruning on LLaMA-3.1 8B. Detailed settings are in Appendix A.

4.2 Main Results

Table 1 presents WikiText-2 perplexity across three sparsity patterns (50% unstructured, 2:4, and 4:8) on Qwen3, LLaMA-3.1, and LLaMA-3.2 models. PRISM achieves the lowest perplexity in every setting. At 50% unstructured sparsity, PRISM w/o RO is the strongest no-RO method on all six models; on LLaMA-3.1 8B, it reaches 10.12, ahead of Wanda++ w/o RO (10.24), Thanos (10.40), SparseGPT (10.54), AlphaPruning (10.62), and Wanda (10.72). With RO, PRISM improves to 9.65 and surpasses Wanda++ (9.98), with the largest gains on smaller models. PRISM targets settings where pruning cost is acceptable; Appendix G shows PRISM + RO is costlier than Wanda and SparseGPT, but $2.28\times$ faster than Wanda++ with RO on LLaMA-3.1 8B. Table 3(e) further shows that AlphaPruning improves PRISM from 9.65 to 9.62, demonstrating compatibility with inter-layer allocation. Under N:M sparsity, cross-matrix allocation is inapplicable, so gains primarily come from within-group selection using the SVD-derived signals and CV fusion, plus optional RO. PRISM outperforms the applicable baselines, for example 48.13 vs. 55.24 on Qwen3-0.6B at 2:4 and 15.10 vs. 16.89 on LLaMA-3.1 8B; PRISM w/o RO also remains better than Wanda++ w/o RO across all six models under both 2:4 and 4:8. The three-seed comparisons in Appendix C confirm that the large gains exceed calibration-subset variance, while the smallest margins are interpreted cautiously.

Table 1: WikiText-2 (Merity et al., 2016) perplexity on Qwen3 (Team, 2025), LLaMA-3.1 (Grattafiori et al., 2024), and LLaMA-3.2 (Meta AI, 2024) under unstructured and N:M sparsity. Lower is better. Wanda++ and PRISM use RO by default; the corresponding “w/o RO” rows disable the reconstruction stage to isolate the underlying pruning quality. Percentages denote relative improvement over the strongest external baseline.

Sparsity	Method	Qwen3			LLaMA-3.1	LLaMA-3.2	
		0.6B	4B	8B	8B	1B	3B
50%	Dense	23.89	11.18	10.93	6.96	11.00	8.74
	SparseGPT	38.77	14.17	13.05	10.54	21.86	13.76
	Wanda	39.90	14.17	12.92	10.72	26.36	14.74
	AlphaPruning (Wanda)	41.73	14.51	13.12	10.62	26.16	14.64
	Thanos	40.58	14.21	13.29	10.40	22.41	13.84
	Wanda++ w/o RO	38.88	14.09	13.01	10.24	21.68	13.74
	Wanda++	34.43	13.93	12.71	9.98	18.92	12.93
	PRISM w/o RO	37.64	13.99	12.81	10.12	21.04	13.56
	PRISM	32.36 (-6.0%)	13.65 (-2.0%)	12.52 (-1.5%)	9.65 (-3.3%)	17.70 (-6.4%)	12.61 (-2.5%)
2:4	SparseGPT	81.63	24.95	18.31	18.45	39.52	25.38
	Wanda	215.29	25.16	18.28	26.30	101.76	39.88
	Thanos	91.11	22.78	16.58	18.78	38.74	25.42
	Wanda++ w/o RO	82.32	24.45	17.56	17.68	39.13	24.95
	Wanda++	55.24	22.53	16.22	16.89	30.13	21.83
	PRISM w/o RO	77.64	22.12	16.53	17.42	35.23	23.67
	PRISM	48.13 (-12.9%)	18.75 (-16.8%)	15.95 (-1.7%)	15.10 (-10.6%)	27.96 (-7.2%)	21.77 (-0.3%)
4:8	SparseGPT	58.34	17.87	14.93	13.76	28.45	18.60
	Wanda	90.15	18.01	14.89	15.50	50.91	24.00
	Thanos	59.35	17.41	14.40	13.85	28.00	18.50
	Wanda++ w/o RO	57.64	17.68	14.82	13.69	27.23	17.88
	Wanda++	45.12	16.93	14.39	12.95	25.33	16.99
	PRISM w/o RO	54.22	17.33	14.54	13.12	25.12	17.01
	PRISM	39.00 (-13.6%)	16.37 (-3.3%)	14.17 (-1.5%)	11.93 (-7.9%)	22.21 (-12.3%)	16.18 (-4.8%)

4.3 Zero-Shot Results

Table 2 reports zero-shot accuracy for LLaMA-3.1 8B at 50% unstructured sparsity on the six benchmarks introduced in Section 4.1. PRISM achieves the best results on ARC-e, Winogrande, PIQA, and BoolQ, while remaining competitive on HellaSwag and ARC-c, where Wanda++ is slightly stronger. These results provide a downstream transfer check beyond perplexity rather than an exhaustive evaluation. Appendix D further reports a text-branch pruning transfer check on Qwen3-VL.

Table 2: Zero-shot accuracy (%) for LLaMA-3.1 8B at 50% unstructured sparsity. Higher is better.

Method	HellaSwag	ARC-e	ARC-c	Winogrande	PIQA	BoolQ
Dense	79.31	82.53	54.95	74.59	81.18	83.06
SparseGPT	<u>71.88</u>	69.74	43.52	<u>71.03</u>	<u>77.74</u>	<u>80.95</u>
Wanda	69.70	69.61	45.05	69.53	77.26	79.17
Wanda++	71.95	<u>70.75</u>	45.90	69.14	77.69	79.72
PRISM	<u>71.81</u>	73.06	<u>45.56</u>	71.32	77.75	81.18

4.4 Ablation Studies

4.4.1 Component Analysis

Table 3 summarizes the incremental contribution of each PRISM component for the main calibration

subset. Panels (a)–(d) use **Wanda + RO** as the baseline to isolate PRISM’s scoring, fusion, and intra-block allocation stages; panel (e) separately tests compatibility with AlphaPruning. Starting from 10.23 PPL, the full PRISM pipeline reaches 9.65, a 5.7% relative reduction; the largest single gain comes from adding the SVD-derived signals. The three-seed results in Appendix C show that the full pipeline gain is stable, while several fine-grained design differences are comparable to calibration-subset variance.

4.4.2 SVD-Derived Signal Analysis

Table 3(b) isolates the SVD-derived scoring terms before CV fusion and intra-block allocation. Adding $\mathcal{I}_{\text{mag}}$ to activation-only scoring reduces PPL from 10.23 to 10.13, while $\mathcal{I}_{\text{low}}$ alone gives only a small change to 10.22. We therefore treat $\mathcal{I}_{\text{low}}$ as a complementary subspace-preservation term rather than an independently dominant signal. Combining both signals is more effective, reaching 9.89 with a fixed rank and 9.83 with adaptive rank selection.

Figure 3 further shows that PRISM reduces layer-wise normalized principal-subspace pruning energy

(a) Components				(b) SVD Signals	
Configuration	PPL	Δ	Role	Scoring Variant	PPL
Wanda (no RO)	10.72	—	—	Act.-only scoring (Wanda + RO)	10.23
Wanda + RO (base.)	10.23	—	—	$\mathcal{I}_{\text{act}} + \mathcal{I}_{\text{mag}}$	10.13
+ SVD-derived signals	9.89	−3.3%	Core 1	$\mathcal{I}_{\text{act}} + \mathcal{I}_{\text{low}}$	10.22
+ Adaptive rank	9.83	−3.9%	Core 1	$\mathcal{I}_{\text{act}} + \mathcal{I}_{\text{mag}} + \mathcal{I}_{\text{low}}$ (fix. rank)	9.89
+ CV fusion	9.74	−4.8%	Enabler	$\mathcal{I}_{\text{act}} + \mathcal{I}_{\text{mag}} + \mathcal{I}_{\text{low}}$ (adapt. rank)	9.83
+ Intra-block alloc. (PRISM)	9.65	−5.7%	Core 2		

(c) Fusion		(d) Intra-Block Allocation		(e) Inter-Layer Compatibility	
Fusion Strategy	PPL	Allocation Strategy	PPL	Setting	PPL
Uniform ($\frac{1}{3}, \frac{1}{3}, \frac{1}{3}$)	9.83	Uniform within block	9.74	Wanda	10.72
Global tuned ratio	9.79	Local-score-only allocation	9.69	Wanda + AlphaPruning	10.62
Std.-prop. weighting	9.81	Gradient-weight-only allocation	9.72	PRISM	9.65
CV-prop. (ours)	9.74	Score-proportional allocation	9.68	PRISM + AlphaPruning	9.62
		Arithmetic-mean fusion	9.67		
		Geom.-mean allocation (ours)	9.65		

Table 3: Ablations on LLaMA-3.1 8B at 50% unstructured sparsity. Panels (a)–(d) use PRISM without inter-layer allocation; panel (e) compares PRISM and PRISM + AlphaPruning. Lower is better.

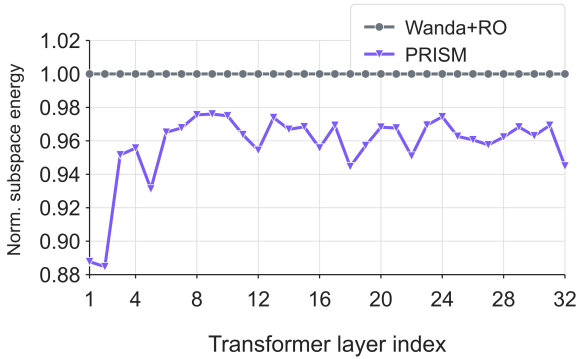


Figure 3: Layer-wise normalized principal-subspace pruning energy for Wanda + RO and PRISM, normalized by the Wanda + RO baseline (1.00). Lower is better.

relative to the activation-only baseline, supporting the mechanism in Section 3.6 and Appendix F. This diagnostic suggests that the fused spectral signals preferentially rescue weights aligned with dominant singular directions, though it does not establish causality.

4.4.3 Fusion Strategy Ablation

Table 3(c) compares fusion rules with the scoring pipeline fixed. Uniform weighting gives 9.83 PPL, standard-deviation-proportional weighting gives 9.81, and a global hand-tuned ratio gives 9.79. CV-proportional fusion gives 9.74, consistent with matrix-specific relative dispersion being a useful training-free weighting signal; the small gaps among fusion variants are interpreted with the variance in Appendix C.

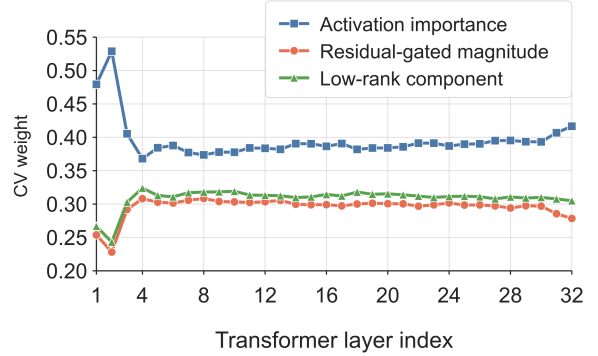


Figure 4: Layer-wise mean CV-based fusion weights for $\mathcal{I}_{\text{act}}$, $\mathcal{I}_{\text{mag}}$, and $\mathcal{I}_{\text{low}}$.

Figure 4 shows that activation importance usually receives the largest share, while both SVD-derived factors remain substantial and vary with depth. We therefore interpret CV fusion as a training-free adaptive reweighting heuristic rather than a complete normalization scheme.

4.4.4 Intra-Block Allocation Analysis

Table 3(d) evaluates intra-block allocation with the element-wise scorer fixed. Uniform within-block retention gives 9.74 PPL; local-score-only, gradient-weight-only, and score-proportional variants give 9.69, 9.72, and 9.68, respectively. The proposed geometric-mean allocator gives 9.65. Non-uniform allocation is consistently favorable in the three-seed study, but the differences among non-uniform variants are small and do not establish a statistically decisive advantage for the geometric

mean.

Table 3(e) shows that PRISM remains complementary to inter-layer allocation. Adding AlphaPruning improves PRISM from 9.65 to 9.62, giving the best overall PPL under the same external layer-wise budget.

Appendix Figure 5 visualizes the learned retention shift relative to uniform 50% within-block allocation. The allocator consistently protects $\mathbf{W}_k$, $\mathbf{W}_q$, and $\mathbf{W}_{\text{gate}}$, while pruning $\mathbf{W}_o$ and $\mathbf{W}_{\text{down}}$ more aggressively; this structured pattern explains why allocation based on matrix sensitivity improves over uniform and single-signal alternatives.

5 Conclusion

We present PRISM, a post-training pruning framework organized around two core ideas and one enabling mechanism: spectral-aware multi-factor scoring from SVD-derived signals, parameter-free CV-based fusion, and gradient-weighted intra-block allocation. Across six Qwen3 and LLaMA models under unstructured and N:M sparsity, PRISM shows consistent improvements over representative baselines, with up to 16.8% relative perplexity improvement. The intra-block allocator is further shown to be complementary to existing inter-layer methods such as AlphaPruning.

Limitations

PRISM targets accuracy-oriented offline pruning rather than minimum pruning cost. Its SVD-derived statistics raise peak memory to 43–46 GB on LLaMA-3.1 8B and make it slower than Wanda, SparseGPT, and Wanda++ without RO (Appendix G). The current formulation is also designed primarily for the language decoder, so in our multimodal experiment we prune only the text branch while keeping the visual encoder and fusion modules unchanged. Extending PRISM to jointly prune all multimodal components remains future work.

Potential risks are mainly related to deployment. By reducing LLM deployment cost, PRISM may also lower the barrier for harmful uses of existing models. Pruning can change robustness, factuality, bias, toxicity, refusal behavior, or other safety-relevant capabilities in ways not captured by our benchmarks; we do not evaluate these dimensions experimentally. Bias and safety evaluation should therefore accompany deployment of pruned models.

References

- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.
- Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. PIQA: Reasoning about physical commonsense in natural language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):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 *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.
- 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*.
- Gongfan Fang, Hongxu Yin, Saurav Muralidharan, Greg Heinrich, Jeff Pool, Jan Kautz, Pavlo Molchanov, and Xinchao Wang. 2024. [MaskLLM: Learnable semi-structured sparsity for large language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 7736–7758.
- Elias Frantar and Dan Alistarh. 2023. [Sparsegpt: Massive language models can be accurately pruned in one-shot](#). In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.
- Elias Frantar, Saleh Ashkboos, Torsten Hoeffer, and Dan Alistarh. 2023. [OPTQ: Accurate quantization for generative pre-trained transformers](#). In *The Eleventh International Conference on Learning Representations*.
- Aaron Grattafiori, Abhimanyu Dubey, et al. 2024. [The llama 3 herd of models](#). *arXiv preprint arXiv:2407.21783*.
- Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and et al. 2024. Hal-lusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14375–14385.
- Yen-Chang Hsu, Ting Hua, Sung-En Chang, Qian Lou, Yilin Shen, and Hongxia Jin. 2022. [Language model compression with weighted low-rank factorization](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

- Ivan Ilin and Peter Richtárik. 2025. Thanos: A block-wise pruning algorithm for efficient large language model compression. *arXiv preprint arXiv:2504.05346*.
- 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 *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.
- Jun Liu, Zhenglun Kong, Pu Zhao, Changdi Yang, Xuan Shen, Hao Tang, Geng Yuan, Wei Niu, Wenbin Zhang, Xue Lin, Dong Huang, and Yanzhi Wang. 2025. [Toward adaptive large language models structured pruning via hybrid-grained weight importance assessment](#). In *Thirty-Ninth AAAI Conference on Artificial Intelligence, Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence, Fifteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2025, Philadelphia, PA, USA, February 25 - March 4, 2025*, pages 18879–18887. AAAI Press.
- Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024. [MMBench: Is your multi-modal model an all-around player?](#) In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, volume 15064 of *Lecture Notes in Computer Science*, pages 216–233. Springer.
- Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael W 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, pages 9117–9152.
- Mehdi Makni, Kayhan Behdin, Zheng Xu, Natalia Ponomareva, and Rahul Mazumder. 2025. A unified framework for sparse plus low-rank matrix decomposition for llms. In *The Second Conference on Parsimony and Learning (Proceedings Track)*.
- Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 2200–2209.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.
- Meta AI. 2024. [Llama 3.2: Revolutionizing edge AI and vision with open, customizable models](#).
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. [A simple and effective pruning approach for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Qwen Team. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. 2025a. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *International Conference on Learning Representations (ICLR)*.
- 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 (ICLR)*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.
- Jing Xiong, Jianghan Shen, Fanghua Ye, Chaofan Tao, Zhongwei Wan, Jianqiao Lu, Xun Wu, Chuanyang Zheng, Zhijiang Guo, Min Yang, Lingpeng Kong, and Ngai Wong. 2025. UNComp: Can matrix entropy uncover sparsity? A compressor design from an uncertainty-aware perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. 2024. BESA: Pruning large language models with blockwise parameter-efficient sparsity allocation. In *International Conference on Learning Representations (ICLR)*.

- Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. 2025. [Wanda++: Pruning large language models via regional gradients](#). In *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, volume ACL 2025 of *Findings of ACL*, pages 4321–4333. Association for Computational Linguistics.
- Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. 2024. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning llms to high sparsity. In *Proceedings of the 41st International Conference on Machine Learning*, pages 57101–57115.
- Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. 2023. [ASVD: activation-aware singular value decomposition for compressing large language models](#). *CoRR*, abs/2312.05821.
- Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pages 4791–4800.
- Longteng Zhang, Sen Wu, Shuai Hou, Zhengyu Qing, Zhuo Zheng, Danning Ke, Qihong Lin, Qiang Wang, Shaohuai Shi, and Xiaowen Chu. 2026. [Salr: Sparsity-aware low-rank representation for efficient fine-tuning of large language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(33):28337–28345.
- Pengxiang Zhao, Hanyu Hu, Ping Li, Yi Zheng, Zhefeng Wang, and Xiaoming Yuan. 2025. FISTApruner: Layer-wise post-training pruning for large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29469–29487.

A Implementation Details

We implement PRISM in PyTorch (Paszke et al., 2019) with HuggingFace Transformers (Wolf et al., 2020). All experiments are conducted on a single NVIDIA H800 GPU. We use the checkpoints Qwen/Qwen3-0.6B, Qwen/Qwen3-4B, Qwen/Qwen3-8B, meta-llama/Llama-3.1-8B, meta-llama/Llama-3.2-1B, and meta-llama/Llama-3.2-3B. For fairness, all baselines are run with their released implementations under the same data protocol as PRISM: identical C4 calibration samples (128 sequences of length 1024), the same sparsity targets, and the same WikiText-2 / zero-shot evaluation sets. The main tables use calibration seed 42; the variance studies use seeds 42, 123, and 256. For clarity, our Wanda++ w/o RO implementation is exactly the original Wanda++ pruning pipeline, corresponding to the “Wanda++ RGS” variant reported in the Wanda++ paper.

The main WikiText-2 results use a 1024-token evaluation context for every dense and pruned model. Calibration length does not affect dense-model evaluation; the LLaMA-3.1 8B dense PPL difference from reports using a 2048-token evaluation context is protocol-dependent. Re-evaluating the dense checkpoint at length 2048 gives 6.39 instead of the 6.96 reported under our length-1024 protocol. We compare methods within the unified protocol used in each table. Adaptive SVD uses $k_{\min} = 0.1 \min(m, n)$ and $k_{\max} = 0.5 \min(m, n)$; sensitivity is evaluated once at the probe rank $\bar{k} = k_{\min}$ and then mapped to an integer rank via Eq. (3). CV-based fusion uses $w_f^{(l)} = \text{CV}_f^{(l)} / \sum_{f'} \text{CV}_{f'}^{(l)}$ with $\epsilon_0 = 10^{-6}$, and the activation-factor CV is averaged over $B = 5$ mini-batches. Intra-block allocation uses the gradient-weight sensitivity proxy $\|\mathbf{W} \odot \nabla_{\mathbf{W}} \mathcal{L}\|_F$ and geometric-mean fusion with $\lambda = 0.8$; this proxy is inspired by Fisher-style saliency measures but does not estimate the Fisher matrix or its diagonal. We use an initial retention spread $[r_{\min}, r_{\max}] = [0.46, 0.54]$. Regional Optimization uses RM-Sprop with learning rate 1×10^{-5} , gradient clipping $\|\nabla\|_{\max} = 1.0$, L2 anchor $\mu = 1.0$, 10 iterations, and 64 random calibration samples per iteration. For the cost results in Section G, all methods are timed under the same calibration setup, and the reported wall-clock includes the full offline pruning pipeline: activation collection, gradient collection, scoring/mask generation, and optional RO.

A.1 Intra-Block Allocation Procedure

Algorithm 1 Intra-block cross-matrix allocation

Require: Target sparsity S_{target} , initial retention bounds $[r_{\min}, r_{\max}]$, balancing λ

Require: Unified scores $\{\mathcal{S}^{(b,t)}\}$, pre-computed gradients $\{\nabla_{\mathbf{W}^{(b,t)}} \mathcal{L}\}$

- 1: **for** each block $b = 1, \dots, B$ **do**
- 2: Compute local importance $\mathcal{I}_{\text{local}}^{(b,t)}$ and gradient-weight sensitivity $\mathcal{I}_{\text{sens}}^{(b,t)}$ for each $t \in \mathcal{T}$
- 3: Normalise both within the block; compute $\mathcal{I}^{(b,t)}$ via geometric mean
- 4: Rank the T matrices by $\mathcal{I}^{(b,t)}$; assign initial retention ratios linearly from $[r_{\min}, r_{\max}]$:

$$r^{(b,t)} = r_{\min} + (r_{\max} - r_{\min}) \cdot \frac{\text{rank}(t)}{T - 1} \quad (14)$$

- 5: Apply a global scaling factor κ so the block satisfies the target:

$$\kappa = \frac{(1 - S_{\text{target}}) \sum_t |\mathbf{W}^{(b,t)}|}{\sum_t r^{(b,t)} \cdot |\mathbf{W}^{(b,t)}|}, \quad (15)$$

$$r^{(b,t)} \leftarrow \text{clip}(\kappa \cdot r^{(b,t)}, 0, 1)$$

- 6: Prune each matrix t to sparsity $1 - r^{(b,t)}$ using unified scores $\mathcal{S}^{(b,t)}$

7: **end for**

We use $\mathcal{I}^{(b,t)}$ to rank the seven matrices rather than allocating retention proportionally, as its absolute scale is not calibrated across projection types. The linearly spaced initialization in $[r_{\min}, r_{\max}]$ yields a bounded redistribution less sensitive to score outliers while preserving the within-block ordering. After global scaling, actual retention may shift outside this window (clipped to $[0, 1]$); $[r_{\min}, r_{\max}]$ controls the spread of the initial allocation, not a hard constraint.

B Mechanism Diagnostics

We first test the covariance–subspace condition used in Proposition 3. For each selected layer–matrix pair, we estimate the off-diagonal input covariance Σ_{off} from dense-model calibration activations and measure $\|V_k^T \Sigma_{\text{off}} V_k\|_F / \|\Sigma_{\text{off}}\|_F$, normalized by the same quantity averaged over random k -dimensional subspaces. Each model uses 20 pairs: five layers and four projections (q , o , up, and down). Table B.1 shows that most probed pairs

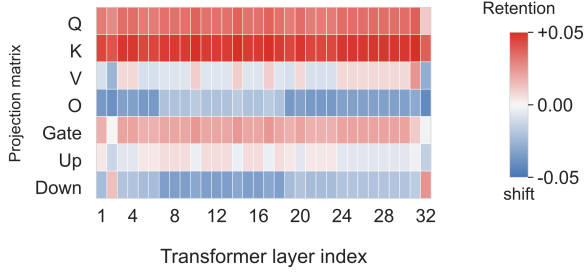


Figure 5: Layer-wise retention shift from uniform 50% within-block allocation under the final geometric-mean allocator. Positive means protection; negative means more pruning.

Table B.1: Covariance-subspace alignment. Ratio > 1 indicates more off-diagonal covariance energy in the weight matrix’s top- k right-singular subspace than in a random subspace.

Model	Pairs	> 1	Mean	Median
LLaMA-3.1 8B	20	65%	$14.64\times$	$1.29\times$
Qwen3-8B	20	80%	$31.07\times$	$3.24\times$
Qwen3-4B	20	80%	$16.58\times$	$2.35\times$

have a ratio above one, while the fractions below 100% confirm that the condition is not universal.

We next evaluate representation preservation on saved LLaMA-3.1 8B checkpoints using 32 C4 sequences of length 1024. Table B.2 compares token-averaged hidden states with the dense model. PRISM gives the lowest MSE and highest cosine similarity among the evaluated methods, both across layers and at the final hidden state.

Table B.2: Hidden-state preservation on LLaMA-3.1 8B at 50% unstructured sparsity. Lower MSE and higher cosine are better.

Method	Mean MSE $\downarrow$	Mean cos. $\uparrow$	Final MSE $\downarrow$	Final cos. $\uparrow$
Wanda	0.0810	0.9521	1.6450	0.8348
SparseGPT	0.0546	0.9709	1.1656	0.8832
Wanda++	0.0579	0.9685	1.2245	0.8767
PRISM	0.0492	0.9737	1.0471	0.8951

Finally, we extract each method’s final mask, apply it to the dense weights, and decompose the mask-induced error $\|X(W - W \odot M)^T\|_F^2$ using dense-model C4 activations. Table B.3 reports aggregate values across LLaMA-3.1 8B decoder projections. PRISM has the lowest aggregate total error, absolute cross-term, and cross fraction, although Wanda has a slightly lower median cross-term. This supports an aggregate interpretation, not a universal per-matrix guarantee.

Table B.3: Post-mask reconstruction diagnostic on LLaMA-3.1 8B. Total and cross-term sums are in millions; the median is in thousands. Lower is better.

Method	Total $\downarrow$	Cross abs. $\downarrow$	Cross frac. $\downarrow$	Median $\downarrow$
Wanda	68.16	6.24	0.0916	4.49
SparseGPT	57.21	11.87	0.2075	15.74
Wanda++	61.64	5.94	0.0964	5.90
PRISM	55.45	5.82	0.0901	4.80

C Calibration-Subset Robustness and Sensitivity

We repeat the reviewer-raised comparisons using calibration seeds 42, 123, and 256. Table C.1 shows that the LLaMA-3.1 8B and Qwen3-4B gains are much larger than calibration-subset variation. The LLaMA-3.2 3B 2:4 gain is directionally consistent but small, so we do not treat it as individually decisive.

Table C.1: Three-seed WikiText-2 PPL (mean $\pm$ std). All rows compare PRISM with Wanda++ using RO.

Setting	Wanda++	PRISM
LLaMA-3.1 8B, 50%	9.98 ± 0.01	9.65 ± 0.02
Qwen3-4B, 2:4	22.48 ± 0.09	18.81 ± 0.06
LLaMA-3.2 3B, 2:4	21.82 ± 0.05	21.71 ± 0.06

D Additional Multimodal Evaluation

We further examine whether PRISM transfers to multimodal large language models using Qwen3-VL-4B-Instruct (Bai et al., 2025). Since our current importance estimation and sparsity allocation are designed for the language decoder, we prune only the text branch and keep the visual encoder and multimodal fusion modules unchanged.

Table D.1 reports results on MMBench-en and MMBench-cn (Liu et al., 2024), HallusionBench (Guan et al., 2024), and DocVQA test (Mathew et al., 2021). PRISM consistently achieves the best performance among all pruned variants, reaching 78.8, 78.4, 54.1, and 91.3, respectively. Compared with the strongest pruning baseline Wanda++, PRISM improves by 0.5 points on MMBench-en, 0.2 on MMBench-cn, 0.2 on HallusionBench, and 0.4 on DocVQA test. These results suggest that the proposed text-side importance signals remain effective in multimodal decoders. Pruning the visual and fusion modules is left for future work.

Table C.2: Three-seed ablation PPL (mean $\pm$ std) on LLaMA-3.1 8B at 50% unstructured sparsity. The full pipeline improvement is stable; several fine-grained differences are small relative to variance.

(a) Components and SVD signals		(b) Fusion, allocation, and compatibility	
Configuration	PPL	Configuration	PPL
Wanda (no RO)	10.69 $\pm$ 0.02	Uniform fusion	9.83 $\pm$ 0.01
Wanda + RO	10.22 $\pm$ 0.02	Global tuned ratio	9.78 $\pm$ 0.01
+ SVD-derived signals	9.88 $\pm$ 0.02	Std.-proportional fusion	9.81 $\pm$ 0.02
+ Adaptive rank	9.83 $\pm$ 0.01	CV-proportional fusion	9.73$\pm$0.02
+ CV fusion	9.73 $\pm$ 0.02		
+ Intra-block allocation	9.65$\pm$0.02	Uniform within block	9.74 $\pm$ 0.02
$I_{\text{act}} + I_{\text{mag}}$	10.12 $\pm$ 0.01	Local-score-only allocation	9.68 $\pm$ 0.02
$I_{\text{act}} + I_{\text{low}}$	10.20 $\pm$ 0.02	Gradient-weight-only allocation	9.73 $\pm$ 0.01
All signals (fixed rank)	9.90 $\pm$ 0.02	Score-proportional allocation	9.69 $\pm$ 0.02
All signals (adaptive rank)	9.83$\pm$0.02	Arithmetic-mean fusion	9.67 $\pm$ 0.02
		Geometric-mean allocation	9.65$\pm$0.02
		Wanda + AlphaPruning	10.61 $\pm$ 0.02
		PRISM + AlphaPruning	9.62$\pm$0.01

Table C.3: Hyperparameter sensitivity on LLaMA-3.1 8B at 50% unstructured sparsity over three calibration subsets. Defaults are fixed across models; nearby settings remain close.

(a) Allocation balance		(b) Rank bounds		(c) Retention range	
λ	PPL	Bounds	PPL	Range	PPL
0.5	9.79 $\pm$ 0.02	[0.05, 0.3]	9.88 $\pm$ 0.02	[0.49, 0.51]	9.71 $\pm$ 0.02
0.6	9.69 $\pm$ 0.02	[0.1, 0.5]	9.83$\pm$0.02	[0.46, 0.54]	9.65$\pm$0.02
0.7	9.68 $\pm$ 0.02	[0.2, 0.5]	9.85 $\pm$ 0.02	[0.43, 0.57]	9.67 $\pm$ 0.02
0.8	9.65$\pm$0.02				
0.9	9.69 $\pm$ 0.02				

Table D.1: Multimodal results on Qwen3-VL-4B-Instruct with only the text branch pruned. Higher is better.

Method	MMBench-en	MMBench-cn	HallusionBench	DocVQA test
Dense	83.1	82.5	56.9	94.3
SparseGPT	77.4	77.2	52.5	90.1
Wanda	76.5	76.1	50.1	89.6
Wanda++	<u>78.3</u>	<u>78.2</u>	<u>53.9</u>	<u>90.9</u>
PRISM	78.8	78.4	54.1	91.3

E Justification of Sensitivity-Guided Rank Selection

We give a short directional argument linking sensitivity and rank allocation. Let $\phi_l(U)$ denote the loss function with layer l replaced by U , assume ϕ_l is β_l -smooth, let $R_l(k) = W^{(l)} - W_{\text{low}}^{(l)}(k)$ be the residual from rank- k SVD truncation, and let $g_l = \|\nabla_{W^{(l)}} \mathcal{L}\|_F$. A standard smoothness bound gives

$$\begin{aligned} \Delta_l(k) &= \phi_l(W_{\text{low}}^{(l)}(k)) - \phi_l(W^{(l)}) \\ &\leq g_l \|R_l(k)\|_F + \frac{\beta_l}{2} \|R_l(k)\|_F^2. \end{aligned} \quad (16)$$

Since $\|R_l(k)\|_F^2 = \sum_{i>k} (\sigma_i^{(l)})^2$ decreases monotonically with k , the upper bound also decreases

with k . Since $\mathcal{S}^{(l)} = r_l(\bar{k}) \cdot g_l$ for a fixed probe rank $\bar{k}$, higher sensitivity implies either larger gradient norm or larger residual ratio at the probe rank; both favor assigning a larger rank to keep the loss increment under control. Eq. (3) instantiates this intuition through an increasing affine mapping,

$$\frac{\partial k^{(l)}}{\partial \mathcal{S}^{(l)}} = \frac{k_{\max} - k_{\min}}{\mathcal{S}_{\max} - \mathcal{S}_{\min}} > 0. \quad (17)$$

Thus higher sensitivities map monotonically to larger ranks, while robust layers use lower ranks. Here β_l appears only in the smoothness upper bound above; it is not estimated or used as an algorithmic hyperparameter. We therefore keep the final rule as the simple monotone mapping in Eq. (3) rather than introducing a separate per-layer optimization objective over candidate ranks.

F Detailed Analysis of SVD-Guided Pruning

This appendix provides the full conditional analysis summarized in Section 3.6. Our goal is not to claim global optimality of the final mask, but to identify a mechanism through which the SVD-derived signals

can reduce pruning-induced reconstruction error beyond activation-only scoring.

F.1 Reconstruction Error Decomposition

Consider a linear layer with input $X \in \mathbb{R}^n$ and output $y = WX$, where $W \in \mathbb{R}^{m \times n}$. Let $M \in \{0, 1\}^{m \times n}$ denote the binary pruning mask, and define the pruned error matrix

$$\Delta = (1 - M) \odot W. \quad (18)$$

Then the output reconstruction error of pruning is

$$L(M) = \mathbb{E} \|\Delta X\|_2^2 = \text{tr}(\Delta \Sigma_X \Delta^\top), \quad (19)$$

$$\Sigma_X = \mathbb{E}[XX^\top].$$

Proposition 1 (Reconstruction error decomposition). *Let $s_j = [\Sigma_X]_{jj}$ and let $\Sigma_X^{\text{off}} = \Sigma_X - \text{diag}(s_1, \dots, s_n)$. For row i , define $\delta_i^\top$ as the i -th row of Δ and $P_i = \{j : M_{ij} = 0\}$ as the pruned set in that row. Then*

$$L(M) = \underbrace{\sum_{i=1}^m \sum_{j=1}^n (1 - M_{ij})^2 W_{ij}^2 s_j}_{L_{\text{diag}}(M)} + \underbrace{\sum_{i=1}^m \sum_{\substack{j \neq j' \\ j, j' \in P_i}} W_{ij} W_{ij'} [\Sigma_X]_{jj'}}_{L_{\text{cross}}(M)}. \quad (20)$$

Equivalently,

$$L_{\text{cross}}(M) = \sum_{i=1}^m \delta_i^\top \Sigma_X^{\text{off}} \delta_i. \quad (21)$$

Proof. From (19), $L(M) = \sum_{i=1}^m \delta_i^\top \Sigma_X \delta_i$. Expanding each quadratic form into diagonal and off-diagonal parts yields (20). Equation (21) is the same expression written using Σ_X^{off} . $\square$

Implication. Activation-only scoring mainly aligns with $L_{\text{diag}}(M)$ because it ranks entries by a magnitude-activation proxy. However, when transformer features are correlated across channels, $L_{\text{cross}}(M)$ can be substantial, and controlling only the diagonal term is insufficient.

F.2 Principal-Subspace Pruning Energy

Let the adaptive-rank SVD of the weight matrix be

$$W = W_{\text{low}} + R = U_k \Sigma_k V_k^\top + R, \quad (22)$$

where $V_k \in \mathbb{R}^{n \times k}$ contains the top- k right singular vectors and R is the residual. We define the *principal-subspace pruning energy* as

$$E_{\text{prin}}(M) := \|\Delta V_k\|_F^2 = \sum_{i=1}^m \|V_k^\top \delta_i\|_2^2. \quad (23)$$

This quantity measures how much of the *pruned* weight mass falls inside the dominant right-singular subspace of W .

Proposition 2 (Upper bound induced by the SVD decomposition). *For any pruning mask M ,*

$$E_{\text{prin}}(M) \leq 2\|((1 - M) \odot W_{\text{low}})\|_F^2 + 2\|((1 - M) \odot R)\|_F^2. \quad (24)$$

Proof. Using (18) and (22),

$$\begin{aligned} \Delta V_k &= ((1 - M) \odot W) V_k \\ &= ((1 - M) \odot W_{\text{low}}) V_k \\ &\quad + ((1 - M) \odot R) V_k. \end{aligned}$$

Applying $\|A + B\|_F^2 \leq 2\|A\|_F^2 + 2\|B\|_F^2$ gives

$$E_{\text{prin}}(M) \leq 2\|((1 - M) \odot W_{\text{low}}) V_k\|_F^2 + 2\|((1 - M) \odot R) V_k\|_F^2.$$

Since V_k has orthonormal columns, right multiplication by V_k cannot increase the Frobenius norm, i.e., $\|AV_k\|_F \leq \|A\|_F$ for any conformable matrix A . Substituting this inequality yields (24). $\square$

Proposition 2 gives a direct interpretation of the two SVD-derived importance factors. The low-rank score I_{low} explicitly protects entries with large $|W_{\text{low},ij}|$, thereby reducing the first term in (24). The residual-gated magnitude score I_{mag} complements this by favoring entries whose observed weight magnitude is large while the low-rank reconstruction gap is small. In particular, for each entry,

$$|W_{\text{low},ij}| = |W_{ij} - R_{ij}| \geq \max(|W_{ij}| - |R_{ij}|, 0), \quad (25)$$

so entries with large $|W_{ij}|$ and small residual $|R_{ij}|$ necessarily carry substantial low-rank mass. This makes I_{mag} a residual-aware proxy for low-rank salience, complementary to the explicit low-rank score I_{low} .

F.3 Conditional Relation to Cross-Column Error

To connect (23) to the cross-column term $L_{\text{cross}}(M)$, decompose each row residual in the singular basis:

$$\begin{aligned}\delta_i &= V_k \pi_i + V_{\perp} \rho_i, \\ \pi_i &= V_k^{\top} \delta_i, \\ \rho_i &= V_{\perp}^{\top} \delta_i,\end{aligned}\quad (26)$$

where $[V_k, V_{\perp}]$ is an orthonormal basis of $\mathbb{R}^n$. Define

$$\begin{aligned}A_k &= V_k^{\top} \Sigma_X^{\text{off}} V_k, \\ B_k &= V_k^{\top} \Sigma_X^{\text{off}} V_{\perp}, \\ C_k &= V_{\perp}^{\top} \Sigma_X^{\text{off}} V_{\perp},\end{aligned}\quad (27)$$

and the tail energy

$$E_{\text{tail}}(M) := \|\Delta V_{\perp}\|_F^2 = \sum_{i=1}^m \|\rho_i\|_2^2. \quad (28)$$

Proposition 3 (Cross-column error in the singular basis). *With the definitions above,*

$$L_{\text{cross}}(M) = \sum_{i=1}^m \left(\pi_i^{\top} A_k \pi_i + 2 \pi_i^{\top} B_k \rho_i + \rho_i^{\top} C_k \rho_i \right). \quad (29)$$

Consequently,

$$\begin{aligned}|L_{\text{cross}}(M)| &\leq \|A_k\|_2 E_{\text{prin}}(M) \\ &\quad + 2 \|B_k\|_2 \sqrt{E_{\text{prin}}(M) E_{\text{tail}}(M)} \\ &\quad + \|C_k\|_2 E_{\text{tail}}(M).\end{aligned}\quad (30)$$

Moreover, if $A_k \succeq \mu I$ for some $\mu > 0$, then

$$\begin{aligned}L_{\text{cross}}(M) &\geq \mu E_{\text{prin}}(M) \\ &\quad - 2 \|B_k\|_2 \sqrt{E_{\text{prin}}(M) E_{\text{tail}}(M)} \\ &\quad - \|C_k\|_2 E_{\text{tail}}(M).\end{aligned}\quad (31)$$

Proof. Substituting (26) into $L_{\text{cross}}(M) = \sum_i \delta_i^{\top} \Sigma_X^{\text{off}} \delta_i$ gives (29). Then

$$\begin{aligned}\sum_i \pi_i^{\top} A_k \pi_i &\leq \|A_k\|_2 \sum_i \|\pi_i\|_2^2 \\ &= \|A_k\|_2 E_{\text{prin}}(M),\end{aligned}$$

$$\begin{aligned}2 \sum_i |\pi_i^{\top} B_k \rho_i| &\leq 2 \|B_k\|_2 \sum_i \|\pi_i\|_2 \|\rho_i\|_2 \\ &\leq 2 \|B_k\|_2 \sqrt{E_{\text{prin}}(M) E_{\text{tail}}(M)},\end{aligned}$$

and

$$\sum_i \rho_i^{\top} C_k \rho_i \leq \|C_k\|_2 E_{\text{tail}}(M),$$

which yields (30). If $A_k \succeq \mu I$, then $\pi_i^{\top} A_k \pi_i \geq \mu \|\pi_i\|_2^2$, and (31) follows by lower-bounding the leading term and upper-bounding the remaining two terms in absolute value. $\square$

Proposition 3 clarifies the condition under which the SVD-derived scores are most beneficial. When the off-diagonal input covariance is well aligned with the dominant right-singular subspace—that is, when $\|A_k\|_2$ is non-negligible while the coupling and tail terms $\|B_k\|_2$, $\|C_k\|_2$, and $E_{\text{tail}}(M)$ remain moderate—the cross-column error is largely governed by $E_{\text{prin}}(M)$. In this regime, preserving entries that carry substantial low-rank mass can reduce the dominant component of $L_{\text{cross}}(M)$ beyond what activation-only scoring achieves.

F.4 Interpretation of the Unified Score

Our final element-wise score for layer l is

$$\begin{aligned}S_{ij}^{(l)} &= w_{\text{act}}^{(l)} I_{\text{act},ij}^{(l)} + w_{\text{mag}}^{(l)} I_{\text{mag},ij}^{(l)} \\ &\quad + w_{\text{low}}^{(l)} I_{\text{low},ij}^{(l)}, \\ w_f^{(l)} &\geq 0, \quad \sum_{f \in \{\text{act}, \text{mag}, \text{low}\}} w_f^{(l)} = 1,\end{aligned}\quad (32)$$

where the matrix-level weights $w_f^{(l)}$ are obtained by the CV-based rule in Section 3.3. The three terms play complementary roles:

- I_{act} is aligned with the diagonal term $L_{\text{diag}}(M)$ in Proposition 1, as it favors entries with large weight magnitude on channels with strong activation energy.
- I_{low} explicitly protects entries with large low-rank magnitude and therefore tightens the upper bound in Proposition 2.
- I_{mag} acts as a residual-aware proxy for low-rank salience via (25), preferentially preserving entries that are both large in magnitude and well explained by the low-rank component.

Table G.1: Offline pruning wall-clock and peak GPU memory on LLaMA-3.1 8B at 50% unstructured sparsity under the same calibration setup. For Wanda++ and PRISM, the corresponding “w/o RO” rows disable the reconstruction stage and isolate the base pruning pipeline. Lower is better.

Method	Time (s)	Peak GPU Mem. (GB)
Wanda	34.65	19.23
SparseGPT	260.02	19.51
Wanda++ w/o RO	144.35	32.57
Wanda++	1929.28	34.75
PRISM w/o RO	836.54	43.22
PRISM + RO	844.77	45.91

Therefore, the activation term mainly targets the diagonal reconstruction cost, while the two SVD-derived terms bias the mask toward lower principal-subspace pruning energy and, under the alignment condition described above, lower cross-column error. The role of the CV-based fusion is to heuristically balance these complementary objectives on a per-matrix basis.

Empirical validation. The analysis above motivates several empirical diagnostics. In Section 4.4, we partially validate the mechanism by focusing on the most direct observable prediction: the reduction of layer-wise principal-subspace pruning energy $E_{\text{prin}}(M)$ by PRISM relative to activation-only pruning. This observation does not constitute a global proof of optimality, but it supports the central mechanism identified by the analysis.

G Offline Pruning Cost

Table G.1 reports end-to-end offline pruning cost under the same calibration setup for all methods; the timing includes activation collection, gradient collection, scoring/mask generation, and optional RO. Post-training pruning is an offline, one-time procedure for a given model and sparsity target, so our objective is not to minimize pruning time at all costs, but to improve the final sparse model within a practical offline budget when such a budget is acceptable. Under this view, PRISM occupies a middle operating regime: it is materially more expensive than lightweight one-shot baselines such as Wanda and SparseGPT, and its SVD-based scoring stage is also heavier than Wanda++ w/o RO (836.54 vs. 144.35 seconds), but it remains much cheaper than the stronger iterative Wanda++ pipeline with RO. Under this protocol, RO adds only 8.23 seconds over PRISM without

RO (844.77 vs. 836.54 seconds), i.e., 0.98% extra wall-clock in the full offline pipeline. By contrast, the RO stage increases Wanda++ from 144.35 to 1929.28 seconds, an additional 1784.93 seconds. As a result, the final PRISM + RO pipeline remains $2.28\times$ faster than Wanda++ with RO in wall-clock (844.77 vs. 1929.28 seconds). The main reason is algorithmic: PRISM keeps the sparse mask fixed during RO, so each iteration performs only reconstruction updates on the surviving weights. Wanda++, in contrast, repeatedly re-prunes during RO and re-collects sublayer-specific inputs before each iteration, which substantially increases offline cost. PRISM’s peak GPU memory is higher because the SVD-based scoring stage and the fused matrix-level statistics are computed in addition to activation-driven pruning.