# SLIDE&PRUNE: A UNIFIED PRUNING FRAMEWORK FOR JOINT LOW RANK COMPRESSION AND CHANNEL SPARSIFICATION VIA SLIDING WINDOW

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Low-rank compression reduces the deployment cost of large language models (LLMs), but preserving model quality under aggressive compression remains challenging. Conventional top-$k$ singular-value truncation is optimal for the corresponding layer-wise reconstruction objective, yet this guarantee does not extend to the behavior of multiple composed Transformer blocks. We introduce Slide&Prune (S&P), a post-training framework that extends sliding-window reconstruction to low-rank compression, enabling out-of-order singular-component selection according to its effects across neighboring blocks. This formulation generalizes rank truncation into a pruning problem, allowing learnable masks to jointly select singular components, attention heads, and feed-forward network channels under a shared parameter budget. S&P uses a two-stage pipeline of global allocation and sliding-window refinement while keeping pretrained weights and SVD factors fixed. Experiments across OPT, Vicuna, and LLaMA models from 6.7B to 30B parameters establish state-of-the-art compression–quality trade-offs among the evaluated structured pruning and low-rank compression methods, with particularly strong gains under aggressive compression. On LLaMA-1-7B at a compression ratio of $0.4$, S&P achieves WikiText-2 perplexity of $13.75$, compared with $45.17$ for the strongest evaluated baseline.

## 1 INTRODUCTION

Large language models (LLMs) have demonstrated strong capabilities across a wide range of tasks, but their memory and computational requirements remain a barrier to (Touvron et al., 2023b; Kaplan et al., 2020). Post-training compression offers a practical way to reduce these requirements while preserving pretrained capabilities using limited calibration data (Zhu et al., 2024). Among existing approaches, low-rank approximation and structured pruning produce smaller operators compatible with standard dense kernels (Wang et al., 2025; Ma et al., 2023). However, preserving model quality under aggressive compression remains challenging.

Singular value decomposition (SVD) provides a principled basis for low-rank compression by approximating a weight matrix with two smaller factors. SVD-based compression is commonly formulated as minimizing layer-wise reconstruction error, retaining the leading singular components of the weight matrix or an activation-aware transformation (Yuan et al., 2023; Wang et al., 2025). Under the corresponding matrix reconstruction objective, top-$k$ truncation is optimal by the Eckart–Young–Mirsky theorem. This guarantee, however, applies to an isolated transformation. In a Transformer, approximation errors propagate through attention operations, nonlinearities, and residual connections. Consequently, the singular components that best reconstruct an individual layer need not be those that best preserve the behavior of several compressed blocks.

Sliding-window reconstruction provides a way to evaluate compression decisions beyond individual layers. It optimizes a group of consecutive blocks by matching their outputs to those of the uncompressed model, then shifts the window through the network. Each reconstruction objective captures the combined effects of compression within the window, while overlap allows decisions to be revisited alongside different neighboring blocks. Recent post-training quantization methods, such as SliderQuant, use this approach to account for interactions between quantized layers (Wang



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.

et al., 2026a). However, the application of sliding-window reconstruction to low-rank compression or structured pruning problems remains unexplored.

To close this gap, we extend sliding-window reconstruction to low-rank compression. By evaluating singular components through multiple Transformer blocks, this objective moves beyond the traditional layer-wise reconstruction setting in which top-$k$ truncation is optimal. Singular components can therefore be selected in an out-of-order fashion according to their effects on window outputs, rather than being restricted to a leading prefix. This formulation generalizes conventional top-$k$ truncation into a pruning problem, providing a common basis for jointly selecting which singular components, attention heads, and FFN channels to retain.

Building on this perspective, we introduce **Slide&Prune (S&P)**, a post-training framework that jointly optimizes internal ranks and structural dimensions under a shared parameter budget. S&P uses learnable masks to select singular components, attention heads, and FFN channels while keeping pretrained weights and SVD factors fixed. The budget accounts for their coupled costs: removing channels changes the parameter cost of the corresponding low-rank factors. S&P first establishes a global allocation of compression capacity, then refines singular-component and structural selections through overlapping windows. Finally, cost-aware materialization chooses between dense and factorized representations according to their post-pruning parameter counts, translating the learned selections into compact operators. Experiments across OPT, Vicuna, and LLaMA (Zhang et al., 2022; Zheng et al., 2023; Touvron et al., 2023a;b) models establish state-of-the-art compression–quality trade-offs among the evaluated structured pruning and low-rank compression methods, with particularly strong gains under aggressive compression.

Our contributions are summarized as follows:

- We extend sliding-window reconstruction to low-rank compression, generalizing conventional top-$k$ truncation into out-of-order singular-component pruning.
- We unify low-rank compression and structured pruning through learnable masks that jointly select singular components, attention heads, and FFN channels under a shared parameter budget that accounts for their coupled costs.
- We develop a two-stage pipeline comprising global rank and structural allocation followed by sliding-window refinement, with cost-aware materialization into compact dense or factorized operators.
- Experiments across OPT, Vicuna, and LLaMA (Zhang et al., 2022; Zheng et al., 2023; Touvron et al., 2023a;b) models from 6.7B to 30B parameters establish state-of-the-art compression–quality trade-offs among the evaluated methods. At a compression ratio of $0.4$ on LLaMA-1-7B, S&P achieves WikiText-2 perplexity of $13.75$, compared with $45.17$ for the strongest evaluated baseline.

## 2 Related Works

### 2.1 Structured Pruning

Structured pruning removes entire heads, channels, or layers, producing smaller operators compatible with standard dense kernels. LLM-Pruner (Ma et al., 2023) uses gradient information to identify removable groups of structurally coupled parameters, while FLAP (An et al., 2023) combines fluctuation-based channel importance, adaptive compression allocation, and bias compensation.

These methods establish structural removal as a practical route to compact LLMs. However, restricting compression to whole heads or channels does not explicitly exploit low-rank redundancy within the retained transformations. This places the full compression burden on structural removal, even when some retained transformations could be represented more compactly. This motivates our joint search over structural units and singular components, allowing parameter savings to come from both channel removal and reduced rank.

### 2.2 Low Rank Compression

Low-rank compression approximates dense weight matrices $\mathbf{W}$ in LLM layers by performing Singular Value Decomposition (SVD) and truncating low-magnitude singular values, representing the original matrix as a product of low-rank factors $\mathbf{W}_u$ and $\mathbf{W}_v$. To preserve critical activation distributions and manage channel-wise outliers, methods like Activation-aware SVD (ASVD) (Yuan et al., 2023) scale the weight matrix using an invertible transformation matrix derived from input activation patterns prior to decomposition. Expanding on this activation-aware paradigm, SVD-LLM (**?**) incorporates activation covariance metrics by computing a Cholesky factor $\mathbf{S}$ of the input covariance matrix $\mathbf{X}\mathbf{X}^\top$, performing SVD on the whitened matrix $\mathbf{W}\mathbf{S}$, and minimizing truncation loss under explicit compression budgets.

A complementary line of work improves the allocation of low-rank capacity. SVDLLM v2 (Wang et al., 2025) adjusts compression ratios according to truncation losses across layers. Dobi-SVD (Qinsi et al., 2025) differentiably learns activation truncation positions and reconstructs the corresponding weights. Zero Sum SVD (Abbasi et al., 2026) selects singular components globally using first-order estimates of calibration-loss changes. These advances show that compression quality depends on where capacity is retained, beyond the choice of single layer decomposition alone. There low-rank formulations, however, do not jointly select the original attention heads and FFN channels to retain. Our method extends the selection space to include these structures, since we view channel pruning and low-rank rank selection as a single optimization problem to achieve extremem model compression without performance degradation.

### 2.3 Hybrid Compression

Hybrid methods exploit the complementary effects of pruning and factorization in several ways. LoSparse (Li et al., 2023) represents each weight matrix as the sum of low-rank and sparse components. LoRAP (Li et al., 2024) assigns different compression mechanisms to different Transformer sublayers, applying low-rank approximation to attention and structured pruning to FFNs. SliM (Mozaffari et al., 2025) combines quantization and semi-structured sparsity with low-rank compensation for the resulting approximation error. In the DNN compression literature, LeSS (Eo et al.) learns filter-pruning masks and low-rank thresholds within a differentiable, budget-constrained framework.

### 2.4 Sliding-Window Optimization

A compression choice that appears optimal for one layer may become suboptimal as its effects propagate through later layers and intervening nonlinearities. Multi-layer reconstruction provides a practical way to expose these interactions. In post-training quantization, SliderQuant (Wang et al., 2026a) optimizes overlapping layer windows and adopts their extent across network depth to accommodate differences in quantization sensitivity. CAT-Q (Wang et al., 2026b) optimizes sequential layer blocks jointly by minimizing the L2 reconstruction error of high-precision block outputs. Within structured pruning, algorithms like SWM (Ding et al., 2025) apply sliding window formulations to merge redundant adjacent layers greedily.

We have noticed that no prior low-rank compression literature considers such inter-layer dynamics. To address this issue, our method applies sliding window reconstruction objective to the joint selection of ranks, heads, and FFN channels. The resulting objective evaluates the combined effect of several compressed layers, while overlap allows a layer's decisions to be revisited with different neighboring layers.

## 3 Methodology

### 3.1 Overview of the Two-Stage Compression Pipeline

Our framework employs a two-stage compression pipeline comprising macro- and micro-level budget allocation. In Stage 1 (Macro Allocation), continuous masks for rank, attention heads, and FFN intermediate channels are jointly trained end-to-end across the entire model; the objective balances output reconstruction error against the target budget via a compression ratio regularization term.

In Stage 2 (Micro Allocation), the layer-wise rank parameters are mapped to singular-value-wise masks to enable out of order singular value selection. Training of the attention head, FFN, and singular value masks then continues in a sliding-window fashion, capturing localized yet multi-layer dynamics to further refine student fidelity to the uncompressed teacher.

### 3.2 Preparation: Data-Whitened SVD and Factorization

Following SVD-LLM, we align low-rank truncation with the feature-level reconstruction error $\|\mathbf{W}\mathbf{X} - \hat{\mathbf{W}}\mathbf{X}\|_F^2$ rather than the weight error $\|\mathbf{W} - \hat{\mathbf{W}}\|_F^2$, since activation channels are highly non-uniform. Given calibration activations $\mathbf{X} \in \mathbb{R}^{d_{\text{in}} \times N}$, we compute the Cholesky factorization $\mathbf{X}\mathbf{X}^\top = \mathbf{S}\mathbf{S}^\top$ and apply SVD to the whitened weight $\tilde{\mathbf{W}} = \mathbf{W}\mathbf{S} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_{r_{\max}})$ holds the singular values in descending order and $r_{\max} = \min(d_{\text{in}}, d_{\text{out}})$. We then initialize the trainable low-rank factors as $\mathbf{W}_u = \mathbf{U}\mathbf{\Sigma}^{1/2}$ and $\mathbf{W}_v = \mathbf{\Sigma}^{1/2}\mathbf{V}^\top\mathbf{S}^{-1}$, so that $\mathbf{W}_u\mathbf{W}_v = \mathbf{W}$ exactly before truncation.

### 3.3 Stage 1: Macro Rank and Mask Learning

In Stage 1, our framework jointly trains continuous mask parameters for low-rank singular values, attention heads, and FFN intermediate channels to determine a global parameter budget allocation end-to-end. We design continuous surrogate masks that guarantee smooth gradient propagation while steadily driving the architecture toward the global compression budget.

#### 3.3.1 Low Rank Mask Parameters

Directly optimizing discrete integer ranks $r \in \{1, \ldots, r_{\max}\}$ is non-differentiable. We adopt a differentiable rank allocation surrogate where the mask value applied to the $i$-th singular value ($i \in \{0, \ldots, r_{\max} - 1\}$) is parameterized by a continuous scalar rank $k$:

$$m_{\text{svd}}(i; k) = 0.5 + 0.5 \tanh\left(\beta \cdot (k - i)\right), \tag{1}$$

where $\beta$ controls the transition steepness around the continuous cutoff threshold $k$.

#### 3.3.2 Attention Head and FFN Channel Masks

For every transformer layer $l \in \{1, \ldots, L\}$, we create a learnable parameter $z$ to learn the mask of the attention head, FFN intermediate channel, and singular values of the decomposed weight matrix. The learned $z$ parameters are turned into a mask through the following:

$$m = \sigma\left(\mu \cdot (z - 0.5)\right). \tag{2}$$

To prune attention heads efficiently without backpropagating gradients through the attention softmax, the head-wise mask values are broadcast and applied directly to the concatenated multi-head representations at the input of the output projection $\mathbf{W}_o$. For FFN intermediate channels, channel-wise masks are applied to the intermediate activations produced by the gate and up projections.

To ensure broad architectural exploration early in training followed by sharp binary convergence, the temperature scaling parameter $\mu$ is annealed linearly across Stage 1 optimization steps $t \in \{1, \ldots, T\}$:

$$\mu(t) = \mu_{\text{init}} + (\mu_{\text{stage1}} - \mu_{\text{init}})\frac{t}{T}. \tag{3}$$

Crucially, $\mu$ is annealed only up to $\mu_{\text{stage1}} = 15$ during Stage 1. This prevents premature gradient saturation and keeps the masks sufficiently malleable for the subsequent Stage 2 sliding-window optimization.

#### 3.3.3 Stage 1: Macro Budget Optimization

Let $\mathbf{H} \in \mathbb{R}^{B \times L \times d}$ and $\hat{\mathbf{H}} \in \mathbb{R}^{B \times L \times d}$ denote the final hidden representations produced by the uncompressed teacher and the compressed student models, respectively, for an input calibration batch $\mathbf{X}$ with sequence length $L$. To preserve representational fidelity, we optimize the continuous rank, attention head, and FFN channel masks using the relative Mean Squared Error (MSE):

$$\mathcal{L}_{\text{MSE}} = \frac{\|\hat{\mathbf{H}} - \mathbf{H}\|_F^2}{\|\mathbf{H}\|_F^2 + \epsilon}, \tag{4}$$

where $\epsilon = 10^{-8}$ prevents numerical instability.

To steer the compressed student toward the global target parameter ratio $r_{\text{target}}$, we formulate the constrained optimization objective:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{MSE}} + \gamma \cdot |r_{\text{current}} - r_{\text{target}}|, \tag{5}$$

where $r_{\text{current}}$ is the differentiable parameter ratio evaluated using the current soft masks.

At each optimization step $t$, the Lagrangian multiplier $\gamma_t$ is updated dynamically via dual ascent:

$$\gamma_{t+1} = \max\left(0.1,\ \gamma_t + \alpha_\gamma \cdot (r_{\text{current}} - r_{\text{target}})\right), \tag{6}$$

where $\alpha_\gamma$ denotes the dual learning rate. When the student is over-budget ($r_{\text{current}} > r_{\text{target}}$), $\gamma$ increases to intensify the budget penalty; conversely, when under-budget, $\gamma$ decreases, allowing the optimizer to prioritize reconstruction fidelity. All mask parameters are jointly trained end-to-end via gradient descent, yielding a globally balanced parameter budget allocation across layers and components.

#### 3.3.4 Dense Breakeven Rank and Capping

For a linear operator with effective dimensions $d_{\text{in}}^{\text{eff}}$ and $d_{\text{out}}^{\text{eff}}$, which change as channels are pruned, a rank-$r$ factorization costs $r(d_{\text{in}}^{\text{eff}} + d_{\text{out}}^{\text{eff}})$ parameters, versus $d_{\text{in}}^{\text{eff}} d_{\text{out}}^{\text{eff}}$ when kept dense. Factorization therefore saves parameters only below the breakeven rank $k_{\text{break}} = \lfloor d_{\text{in}}^{\text{eff}} d_{\text{out}}^{\text{eff}} / (d_{\text{in}}^{\text{eff}} + d_{\text{out}}^{\text{eff}}) \rfloor$; above it, the operator costs more than a dense weight while adding approximation error. We therefore cap each layer's cost and define the global size ratio relative to the uncompressed model:

$$\mathcal{P}_{\text{layer}} = \min\left(r(d_{\text{in}}^{\text{eff}} + d_{\text{out}}^{\text{eff}}),\ d_{\text{in}}^{\text{eff}} d_{\text{out}}^{\text{eff}}\right), \qquad r_{\text{current}} = \frac{\sum_{l=1}^{L} \sum_{m \in \text{linears}(l)} \mathcal{P}_{\text{layer}}^{(l,m)}}{\mathcal{P}_{\text{dense}}}. \tag{7}$$

### 3.4 Stage 2: Sliding-Window Mask Learning

Following the global macro-allocation from Stage 1, we refine the mask parameters within localized layer windows to capture complex inter-layer interactions. In this micro stage, we convert the scalar rank parameter $k$ into fine-grained, singular-value-wise parameters $\mathbf{z}_{\text{sv}} \in \mathbb{R}^{r_{\max}}$, initialized as:

$$z_{\text{sv},i} = 1.0 - 0.5 \cdot \frac{i}{\max(1, K)}, \quad \text{for } i \in \{0, \ldots, r_{\max} - 1\}, \tag{8}$$

where $K = \text{round}(k)$ is the integer rank determined by Stage 1. This linear ramp naturally anchors the continuous gates around $0.5$, activating prefix singular values ($i < K$) while suppressing trailing ones ($i > K$).

The Eckart–Young–Mirsky theorem guarantees that retaining the leading top-$k$ singular values minimizes Frobenius error for an isolated linear layer. However, in deep non-linear networks, out of order subsets of singular values often interact more constructively with downstream representations. Our singular-value-wise formulation liberates the model from strict prefix truncation: essential singular directions outside the top-$k$ prefix can be recruited, while redundant directions within the top-$k$ prefix can be pruned.

Let the model layers be partitioned into sliding windows of size $s$ and number of windows $N_{\text{win}}$:

$$\mathbf{W}_i = \{\mathbf{W}_i, \mathbf{W}_{i+1}, \ldots, \mathbf{W}_{i+s-1}\}, \quad \text{for } i \in \{0, \ldots, N_{\text{win}} - 1\}. \tag{9}$$

We train the mask parameters based on a local, window-wise manner by optimizing a given window $i$'s objective

$$\mathcal{L}_i = \mathcal{L}_{\text{MSE}}^{(i)} + \gamma_i \, |r_{i,\text{current}} - r_{\text{i,target}}| \tag{10}$$

where

$$\mathcal{L}_{\text{MSE}}^{(i)} = \frac{\left\| \mathcal{F}(\mathbf{W}_i, \mathbf{X}) - \mathcal{F}(\hat{\mathbf{W}}_i, \mathbf{X}) \right\|_F^2}{\|\mathcal{F}(\mathbf{W}_i, \mathbf{X})\|_F^2 + \epsilon}. \tag{11}$$

$\mathcal{F}$ denotes the output feature of the window and $\mathbf{W}_i = \{\mathbf{W}_1, \mathbf{W}_2, ..., \mathbf{W}_s\}$ the set of weights in the $i$th uncompressed model window, and $\hat{\mathbf{W}}_i$ the $i$th compressed model window. $\gamma_i$ is a learnable

penalty unique to each window, and $r_{\text{i, target}}$ is the window-wise target compression ratio obtained from the macro-level Stage 1 search. After optimizing window $i$, we move on to the next window, which is a set of blocks offest by stride $t$. We complete a single optimization cycle after we reach the last window, then start from the first window for the next epoch.

### 3.5 Compression: Model Pruning by $z$ Parameters

Once Stage 2 concludes, the mask parameters are hard-discretized to produce an efficient, physically sliced architecture compatible with standard inference engines. Active head indices $\mathcal{I}_{\text{head}}$, FFN channel indices $\mathcal{I}_{\text{FFN}}$, and singular value indices $\mathcal{I}_{\text{sv}}$ are selected by thresholding each $z$ parameter by

$$\mathcal{I}_{\text{type}} = \{i \in \{0, \dots, N-1\} \mid z_{\text{type},i} > 0.5\}, \tag{12}$$

where $N$ refers to the number of each component type.

With the obtained attention head, FFN channel, and singular value indices, we truncate each linear weight accordingly. To ensure that no parameters are waisted, we leave layers as dense form if the layer's decomposed inner rank $r$ is greater than breakeven point $k_{\text{break}}$:

$$\mathbf{W}_u[\mathcal{I}_{\text{out}}, \mathcal{I}_{\text{sv}}]\, \mathbf{W}_v[\mathcal{I}_{\text{sv}}, \mathcal{I}_{\text{in}}] \tag{13}$$

If the decomposed form is more parameter efficient, we leave the weights in its decomposed form:

$$\mathbf{W}_v = \mathbf{W}_v[\mathcal{I}_{\text{sv}}, \mathcal{I}_{\text{in}}], \mathbf{W}_u = \mathbf{W}_u[\mathcal{I}_{\text{out}}, \mathcal{I}_{\text{sv}}]. \tag{14}$$

## 4 Experiments & Results

We evaluate S&P to answer two main questions: (1) how it compares with existing SVD-based and structured pruning compression methods across different LLM architectures and model scales; (2) whether its advantage becomes more pronounced as the compression budget decreases.

### 4.1 Experimental Setup

We conduct experiments on NVIDIA H200 Tensor Core GPU. Perplexity is reported on WikiText-2 (Merity et al., 2016), Penn Treebank (Marcus et al., 1993), and C4 (Raffel et al., 2020) datasets, while zero-shot accuracy experiments are conducted on OpenBookQA (Mihaylov et al., 2018), ARC-Easy/Challenge (Clark et al., 2018), WinoGrande (Sakaguchi et al., 2020), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and MathQA (Amini et al., 2019), computed with (Gao et al., 2023). We also report the mean accuracy over the seven tasks and its relative drop from the uncompressed model. Following prior work, we prune only the main transformer linear matrices. We use a WikiText2 calibration set of 256 sequences of length 2048, matching SVD-LLM (**?**), Dobi-SVD (Qinsi et al., 2025), and Zero Sum SVD (Abbasi et al., 2026). A lower $CR = \frac{\mathcal{P}_{\text{compressed}}}{\mathcal{P}_{\text{dense}}}$ (Qinsi et al., 2025; Abbasi et al., 2026) corresponds to a more restrictive parameter budget. To check the robustness of our pipeline, we test our framework on diverse model architecture and sizes.

### 4.2 Main Results

S&P consistently outperforms prior SVD-based methods, and the margin widens with compression. Table 1 reports results on LLaMA-1-7B (Touvron et al., 2023a). At a moderate ratio of 0.8, S&P achieves the best WikiText-2 (Merity et al., 2016) (6.27) and C4 (Raffel et al., 2020) (9.18) perplexity and the highest average accuracy (0.55), matching the uncompressed model. The advantage grows sharply as compression becomes more aggressive. At CR 0.6, S&P reduces WikiText-2 perplexity from 11.44 (ZS-SVD (Abbasi et al., 2026)) to 8.28 and cuts the accuracy drop from 25.5% to 15.7%. At CR 0.4, every baseline degrades severely: WikiText-2 perplexity exceeds 45 and more than 41% of average accuracy is lost. S&P remains usable at a perplexity of 13.75, 3.3× lower than the best baseline, with an accuracy drop of 27.3%.

S&P generalizes across model families and scales. Table 2 evaluates S&P on OPT-6.7B (Zhang et al., 2022), LLaMA-1-30B (Touvron et al., 2023a), and Vicuna-7B (Zheng et al., 2023) at CR 0.8, where it achieves the best perplexity and accuracy on all three. On OPT-6.7B, vanilla SVD

Table 1: Performance of LLaMA-7B under different compression ratios (CR). We report perplexity (↓) on WikiText-2, PTB, and C4, and zero-shot accuracy (↑) on seven benchmarks. Drop denotes the relative decrease (%) in average accuracy compared to the uncompressed baseline. Ours denotes S&P. Best results within each compression ratio are in **bold**.

| CR | Method | PPL (↓) | | | Accuracy (↑) | | | | | | | Avg. | Drop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | Wiki2 | PTB | C4 | OBQA | ARC-e | ARC-c | WinoG | HellaS | PIQA | MathQA | ↑ | ↓ |
| 1.0 | Baseline | 5.68 | 8.35 | 7.34 | 0.34 | 0.75 | 0.42 | 0.70 | 0.57 | 0.79 | 0.28 | 0.55 | 0.0 |
| 0.8 | ASVD | 11.14 | 16.55 | 15.93 | 0.25 | 0.53 | 0.27 | 0.64 | 0.41 | 0.68 | **0.24** | 0.43 | 21.8 |
| | SVD-LLM | 7.94 | 16.22 | 15.84 | 0.22 | 0.58 | 0.29 | 0.63 | 0.43 | 0.69 | **0.24** | 0.44 | 20.0 |
| | Dobi-SVD | 8.54 | 14.83 | 10.01 | 0.26 | 0.59 | 0.31 | 0.66 | 0.44 | 0.70 | 0.23 | 0.46 | 16.4 |
| | ZS-SVD | 6.74 | **11.87** | 10.74 | 0.31 | **0.70** | 0.37 | **0.68** | 0.49 | 0.74 | **0.24** | 0.50 | 9.1 |
| | **Ours** | **6.27** | 12.71 | **9.18** | **0.41** | 0.67 | **0.38** | **0.68** | **0.69** | **0.77** | **0.24** | **0.55** | **0.0** |
| 0.6 | ASVD | 1407 | 3292 | 1109 | 0.13 | 0.28 | 0.22 | 0.48 | 0.26 | 0.55 | 0.19 | 0.30 | 45.5 |
| | SVD-LLM | 13.11 | 63.75 | 49.83 | 0.19 | 0.42 | 0.25 | 0.58 | 0.33 | 0.60 | 0.21 | 0.37 | 32.7 |
| | Dobi-SVD | 13.54 | 46.38 | 23.54 | 0.22 | 0.41 | 0.27 | 0.58 | 0.34 | 0.61 | **0.23** | 0.38 | 30.9 |
| | ZS-SVD | 11.44 | 43.19 | 34.13 | 0.23 | 0.52 | 0.26 | **0.62** | 0.35 | 0.64 | 0.22 | 0.41 | 25.5 |
| | **Ours** | **8.28** | **27.10** | **19.02** | **0.38** | **0.55** | **0.33** | 0.61 | **0.54** | **0.69** | **0.23** | **0.48** | **15.7** |
| 0.4 | ASVD | 57057 | 45218 | 43036 | 0.12 | 0.26 | 0.21 | 0.49 | 0.26 | 0.53 | 0.18 | 0.29 | 47.3 |
| | SVD-LLM | 53.74 | 438.58 | 345.49 | 0.14 | 0.28 | 0.22 | 0.50 | 0.27 | 0.55 | 0.21 | 0.31 | 43.6 |
| | Dobi-SVD | 46.18 | 238.91 | 190.62 | 0.15 | 0.31 | 0.20 | 0.52 | 0.28 | 0.54 | 0.22 | 0.32 | 41.8 |
| | ZS-SVD | 45.17 | 334.85 | 212.57 | 0.15 | 0.31 | 0.21 | 0.54 | 0.28 | 0.55 | 0.21 | 0.32 | 41.8 |
| | **Ours** | **13.75** | **103.53** | **48.73** | **0.35** | **0.41** | **0.26** | **0.55** | **0.40** | **0.60** | **0.23** | **0.40** | **27.3** |

Table 2: Results on different LLM families at a compression ratio of 0.8. We report perplexity on WikiText-2 (↓) and average zero-shot accuracy (↑). Ours denotes S&P. Best results among compressed models are in **bold**.

| Method | OPT-6.7B | | LLaMA-30B | | Vicuna-7B | |
|---|---|---|---|---|---|---|
| | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ | PPL ↓ | Acc. ↑ |
| Original | 10.86 | 0.52 | 4.10 | 0.61 | 6.78 | 0.56 |
| SVD | 66275 | 0.03 | 946.31 | 0.33 | 18644 | 0.05 |
| FWSVD | 14559 | 0.06 | 15.98 | 0.42 | 2758 | 0.09 |
| ASVD | 82.00 | 0.32 | 6.74 | 0.44 | 16.23 | 0.33 |
| SVD-LLM | 16.04 | 0.41 | 6.61 | 0.54 | 8.41 | 0.51 |
| ZS-SVD | 11.40 | 0.51 | 4.83 | 0.59 | 8.08 | 0.53 |
| **Ours** | **11.25** | **0.54** | **4.63** | **0.64** | **7.57** | **0.59** |

and FWSVD (Hsu et al., 2022) collapse to perplexities above $10^4$, while S&P stays within 0.4 of the uncompressed model (11.25 vs. 10.86). On the instruction-tuned Vicuna-7B, S&P reduces perplexity from 8.08 (ZS-SVD (Abbasi et al., 2026)) to 7.57. The gains persist at the 30B scale (4.63 vs. 4.83).

S&P is competitive with structured pruning. S&P jointly performs rank truncation and channel pruning, so we also compare it against dedicated structured pruning methods on LLaMA-1-13B (Touvron et al., 2023a) (Table 3). S&P outperforms LLM-Pruner (Ma et al., 2023) (0.63) and FLAP (An et al., 2023) (0.66) in average accuracy, and it matches the strongest SVD baseline, ZS-SVD (Abbasi et al., 2026) (0.68). It achieves the best results on PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), and ARC-Challenge (Clark et al., 2018).

### 4.3 Ablation Studies

Table 4 shows that allowing out-of-order singular value selection consistently lowers perplexity on all three datasets. The benefit grows with compression: PTB (Marcus et al., 1993) perplexity decreases by 1.4% at CR 0.8 and by 7.0% at CR 0.6. This matches the intuition that the tighter the rank budget, the more the greedy top-k choice costs. Table 5 compares S&P with a variant that

Table 3: Zero-shot accuracy (↑) of LLaMA-1-13B at a compression ratio (CR) of 0.8, compared with pruning and SVD-based methods. Ours denotes S&P. Best results are in **bold** and second-best are underlined.

| CR | Method | BoolQ | PIQA | WinoG | ARC-e | ARC-c | Avg. ↑ |
|---|---|---|---|---|---|---|---|
| 1.0 | Baseline | 0.78 | 0.79 | 0.73 | 0.77 | 0.47 | 0.70 |
| 0.8 | LLM-Pruner | 0.67 | 0.77 | 0.65 | 0.68 | 0.38 | 0.63 |
| | FLAP | 0.70 | **0.78** | 0.69 | **0.73** | **0.43** | 0.66 |
| | SVD-LLM | 0.63 | 0.73 | 0.69 | 0.69 | 0.35 | 0.62 |
| | ZS-SVD | **0.77** | 0.77 | 0.70 | **0.73** | **0.43** | **0.68** |
| | **Ours** | 0.76 | **0.78** | **0.71** | 0.70 | **0.43** | **0.68** |

Table 4: Ablation on Out-of-Order (OoO) rank selection in Stage 2 on LLaMA-1-7B. We report perplexity (↓). Best results within each compression ratio (CR) are in **bold**.

| CR | OoO | Wiki2 ↓ | PTB ↓ | C4 ↓ |
|---|---|---|---|---|
| 0.8 | ✗ | 6.32 | 12.89 | 9.33 |
| | ✓ | **6.27** | **12.71** | **9.18** |
| 0.6 | ✗ | 8.42 | 29.14 | 20.10 |
| | ✓ | **8.28** | **27.10** | **19.02** |

optimizes each layer independently, without the sliding window, at CR 0.6. Without the window, WikiText-2 (Merity et al., 2016) perplexity explodes from 8.28 to 147.67, and the accuracy drop nearly triples, from 15.7% to 42.0%.

### 4.4 Analysis: What does S&P Prune?

Out-of-order (OoO) selection concentrates in attention. Figure 1 reports, for each layer type, the fraction of retained ranks that fall outside the top-k. Attention projections depart from the top-$k$ ordering far more than FFN layers do. The key and value projections of LLaMA-1-7B (Touvron et al., 2023a) reach 11.7% and 12.0%, respectively, while FFN layers stay at or below 4.0%. The pattern holds across all three models, though it is weaker on OPT-6.7B (Zhang et al., 2022), whose FFN layers are almost entirely in order. This suggests that the importance of a singular direction in attention depends strongly on downstream interactions, such as the query–key product and softmax, that a per-matrix criterion cannot capture.

Compression is spread across the three compensation components. Figure 2 decomposes the parameters S&P removes into components. Rank truncation accounts for 76–81% of the removed parameters and removes about two thirds of all singular values. Channel and head pruning contribute the remaining 19–24%. They do this while pruning only 12–15% of FFN channels and 5–7% of attention heads, so a small number of pruned structural units yields a disproportionate share of the savings. The breakdown is remarkably stable across different models, which indicates that S&P finds a consistent allocation between the two compression axes rather than a model-specific one.

### 4.5 Efficiency

Table 7 measures inference efficiency on LLaMA-1-7B (Touvron et al., 2023a) at CR 0.6, with batch size 32 and sequence length 64. Because channel and head pruning shrink the hidden dimensions themselves, S&P reduces activation memory by 3.6× relative to SVD-LLM (Wang et al., 2025) z(2.79 vs. 9.94 GiB). It also achieves the highest throughput, 1390 tokens/s, which is 15% faster than SVD-LLM. Pure low-rank factorization cannot provide this benefit, because it leaves the width of intermediate activations unchanged.

Compression cost. As shown in Table 6, S&P takes 99.4 minutes to compress LLaMA-1-7B (Touvron et al., 2023a), compared with 8.7 minutes for ZS-SVD (Abbasi et al., 2026). We consider this cost acceptable for three reasons. It is paid once, before deployment. It fits on a single GPU. And it

Table 5: Ablation on the sliding window (SW) on LLaMA-1-7B at a compression ratio of 0.6. We report perplexity (↓) on WikiText-2, PTB, and C4, and zero-shot accuracy (↑) on seven benchmarks. Drop denotes the relative decrease (%) in average accuracy compared to the uncompressed baseline. Best results are in **bold**.

| SW | PPL (↓) | | | Accuracy (↑) | | | | | | | Avg. | Drop |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Wiki2 | PTB | C4 | OBQA | ARC-e | ARC-c | WinoG | HellaS | PIQA | MathQA | ↑ | ↓ |
| ✗ | 147.67 | 341.15 | 147.67 | 0.28 | 0.30 | 0.22 | 0.50 | 0.27 | 0.52 | 0.20 | 0.33 | 42.0 |
| ✓ | **8.28** | **27.10** | **19.02** | **0.38** | **0.55** | **0.33** | **0.61** | **0.54** | **0.69** | **0.23** | **0.48** | **15.7** |

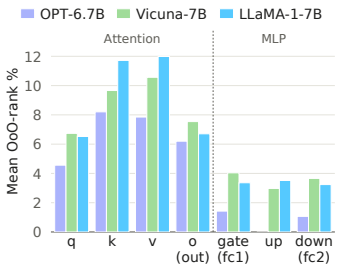


Figure 1: Mean Out-of-Order (OoO) rank percentage across layer types for three LLMs.

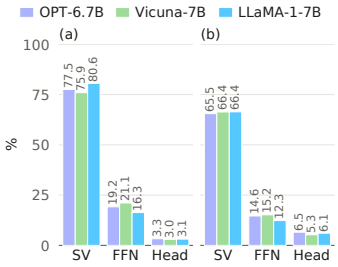


Figure 2: (a) Share of total parameter reduction per component. (b) Fraction of units pruned within each component.

buys large accuracy gains at aggressive ratios, for example reducing the accuracy drop from 41.8% to 27.3% at CR 0.4.

Table 6: Compression time of LLaMA-1-7B at CR 0.6 on H200 (16 CPU threads). Ours denotes S&P. Best in **bold**.

| Method | Time (s) ↓ | Time (min) ↓ |
|---|---|---|
| SVD-LLM | 700 | 11.7 |
| ZS-SVD | **521** | **8.7** |
| **Ours** | 5965 | 99.4 |

Table 7: Memory and throughput of LLaMA-1-7B at CR 0.6, measured on H200 with batch size 32 and seqeunce length 64. Ours denotes S&P. Best in **bold**.

| Method | Peak Mem. (GiB) ↓ | Act. Mem. (GiB) ↓ | Throughput (tok/s) ↑ |
|---|---|---|---|
| SVD-LLM | **10.32** | 9.94 | 1207 |
| ZS-SVD | 27.70 | 15.11 | 1178 |
| **Ours** | 22.38 | **2.79** | **1390** |

## 5 Conclusion

We introduced S&P, a post-training compression framework that jointly allocates a shared parameter budget across structured pruning and low-rank approximation while accounting for inter-layer error propagation through overlapping Transformer-block windows. S&P further supports non-prefix singular-value selection and retains operators in dense form when factorization would not reduce their parameter count. Across OPT, Vicuna, and LLaMA models from 6.7B to 30B parameters, S&P generally improves the trade-off between model size and predictive quality, with its largest gains under aggressive compression: on LLaMA-1-7B at CR $= 0.4$, it substantially reduces perplexity on all three evaluated corpora and improves average zero-shot accuracy by seven percentage points over the strongest baseline. Our ablations demonstrate consistent benefits from sliding-window optimization, while also showing that the best balance between structural pruning and singular-value selection can vary across evaluation corpora. Overall, these results support treating structural units, singular directions, and cross-layer interactions as coupled compression choices rather than independent stages.

## AI USE STATEMENT

We used generative AI tools to assist with methodology and experimental design, code implementation and debugging, result analysis, figure preparation, and manuscript drafting. All AI-assisted code, analyses, figures, and text were reviewed and verified by the authors. The authors take full responsibility for the methodology, experiments, results, claims, and final content of this work.

## REPRODUCIBILITY STATEMENT

Sections 3–5 describe the method and evaluation protocol.

## REFERENCES

Ali Abbasi, Chayne Thrash, Haoran Qin, Shansita Sharma, Sepehr Seifi, and Soheil Kolouri. Zero sum SVD: Balancing loss sensitivity for low rank LLM compression. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 2357–2367, 2019. doi: 10.18653/v1/N19-1245.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models, 2023.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239.

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.

Xuan Ding, Rui Sun, Yunjian Zhang, Xiu Yan, Yueqi Zhou, Kaihao Huang, Suzhong Fu, Angelica I Aviles-Rivero, Chuanlong Xie, and Yao Zhu. A sliding layer merging method for efficient depth-wise pruning in llms, 2025. URL https://arxiv.org/abs/2502.19159.

Moonjung Eo, Suhyun Kang, and Wonjong Rhee. Less: Learning to select a structured archi-tecture over filter pruning and low-rank de.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, December 2023. URL https://zenodo.org/records/10256836.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *International Conference on Learning Representations (ICLR)*, 2022.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Guangyan Li, Yongqiang Tang, and Wensheng Zhang. Lorap: Transformer sub-layers deserve differentiated structured compression for large language models. *arXiv preprint arXiv:2404.09695*, 2024.

Yixiao Li, Yifan Yu, Qingru Zhang, Chen Liang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Losparse: Structured compression of large language models based on low-rank and sparse approximation. In *International Conference on Machine Learning*, pp. 20336–20350. PMLR, 2023.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems*, 2023.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

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 (EMNLP)*, pp. 2381–2391, 2018. doi: 10.18653/v1/D18-1260.

Mohammad Mozaffari, Amir Yazdanbakhsh, and Maryam Mehri Dehnavi. SLiM: One-shot Quantized Sparse Plus Low-rank Approximation of LLMs. 2025. URL https://openreview.net/forum?id=4UfRP8MopP.

Wang Qinsi, Jinghan Ke, Masayoshi Tomizuka, Kurt Keutzer, and Chenfeng Xu. Dobi-SVD: Differentiable SVD for LLM compression and some new perspectives. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=kws76i5XB8.

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. doi: 10.1609/aaai.v34i05.6399.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. Sliderquant: Accurate post-training quantization for llms. In *International Conference on Learning Representations*, 2026a.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, and Anbang Yao. Cat-q: Cost-efficient and accurate ternary quantization for llms. *arXiv preprint arXiv:2606.26650*, 2026b.

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

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models, 2023.

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 (ACL)*, pp. 4791–4800, 2019. doi: 10.18653/v1/P19-1472.

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

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 46595–46623, 2023.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577, 2024. doi: 10.1162/tacl_a_00704.