

DSPP: Dataset-Specific Progressive Pruning for Efficient and Generalizable LLM Deployment

Lichuan Wan¹ Huiqiang Jiang² Wenxuan Liu¹ Hua Yang^{1*}

¹Shanghai Jiao Tong University, Shanghai, China

²Qwen Team, Alibaba Inc.

{wanlichuan, wenxuanliu, hyang}@sjtu.edu.cn iofu728@gmail.com

Abstract

Large language models (LLMs) have demonstrated exceptional performance across various domains. However, their increasing sizes pose significant deployment challenges, particularly in resource-constrained scenarios. While network pruning provides an effective compression strategy, existing approaches either depend on calibration datasets—leading to poor generalization—or require training auxiliary predictors, incurring significant computational overhead. In this paper, we propose DSPP (Dataset-Specific Progressive Pruning), a novel training-free structured pruning method that dynamically updates the importance scores of neurons during inference to identify and retain dataset-specific neurons. By employing a progressive pruning strategy, DSPP can further reduce computational overhead. Experimental results show that DSPP demonstrates remarkable generalization capabilities across diverse datasets and models, while achieving a 1.36× speedup at 50% sparsity level.

1 Introduction

LLMs have achieved remarkable performance across diverse domains (Grattafiori et al., 2024; Arefeen et al., 2024). However, as model sizes continue to grow, so do the computational and memory requirements, posing significant challenges for deployment, especially in resource-constrained scenarios. Among the compression techniques developed in response—quantization (Dettmers et al., 2022), speculative decoding (Kim et al., 2023), weight sparsification (Frantar and Alistarh, 2023), and pruning (Ma et al., 2023; Sun et al., 2024)—network pruning is particularly attractive: it reduces memory and compute simultaneously while often preserving most of the original model’s accuracy (Cheng et al., 2024).

Compared to unstructured pruning (Frantar and Alistarh, 2023; Sun et al., 2024; Zhang et al., 2024)

*Corresponding author.

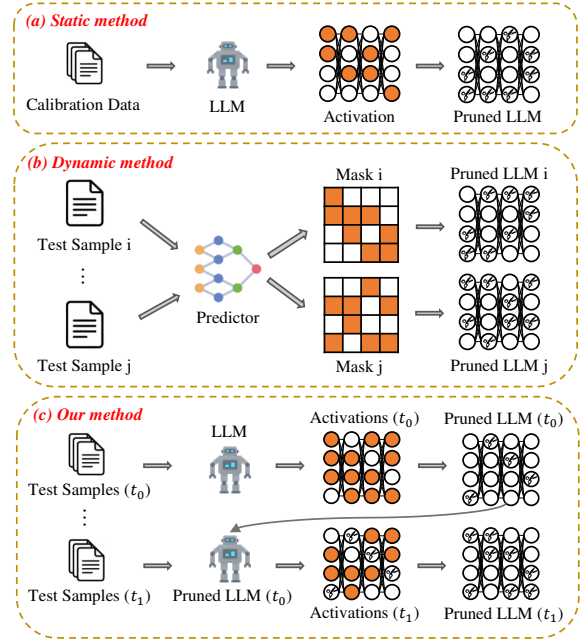


Figure 1: Comparison of pruning paradigms. **Static** methods produce a single mask from a generic calibration set and apply it to every input. **Dynamic predictor-based** methods generate per-sample masks via a trained predictor. **DSPP** progressively refines a dataset-specific mask online from the input distribution, without training a predictor.

that requires specialized hardware support for practical speedups, structured pruning (An et al., 2024; Wang et al., 2025a,b) achieves efficient inference acceleration across diverse hardware platforms by removing entire units (e.g., neurons). Existing structured pruning methods fall into static and dynamic categories, as illustrated in Figure 1. **Static** methods (Wang et al., 2025b; An et al., 2024; Ashkboos et al., 2024; Ma et al., 2023) rely on calibration data to produce pruning masks which are applied to all test tasks, leading to poor generalization when the calibration and test distributions differ. **Dynamic** methods (Liu et al., 2023; Song et al., 2025, 2024; Akhauri et al., 2024) generate pruning

masks from test data but typically require training additional predictors, which introduces substantial overhead. These limitations motivate a key question: *Can we obtain adaptive pruning masks without training predictors?*

Previous studies have identified language-specific neurons in LLMs (Zhao et al., 2024) and shown that task-specific calibration improves pruning effectiveness (Khanal and Capone, 2024). Extending these insights, we hypothesize the existence of *dataset-specific neurons* and empirically analyze this phenomenon. Experiments confirm that pruning masks converge rapidly within the same dataset: *using just 20% of samples yields over 80% Jaccard similarity with the full-dataset mask, while cross-dataset similarity stays low* (§3.1). This indicates that each dataset exhibits a stable, distinctive activation pattern that can be reliably captured using only a small subset of samples. However, computing masks directly on test data risks information leakage, rendering it unsuitable for deployment.

Therefore, we propose DSPP (Figure 3), an efficient structured pruning method that leverages dataset-specific activation patterns to adaptively prune the weights of LLMs. Specifically, we design a novel structured importance metric for each neuron based on statistical properties of both activations and model parameters. We then introduce a training-free adaptive pruning paradigm in which importance scores are computed and continuously updated during inference, enabling the identification of dataset-specific neurons. To accelerate this process, we adopt a progressive pruning strategy with exponentially increasing intervals, allowing higher sparsity levels to benefit from more samples. Since the dataset-specific pruning masks are dynamically generated and incrementally refined using current input samples, DSPP avoids the risk of test-data leakage.

In this work, we evaluate DSPP on the LLaMA family (LLaMA2-7B/13B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024), LLaMA3.1-8B (Grattafiori et al., 2024)) and additionally on Qwen2.5-7B (Qwen et al., 2025). DSPP demonstrates robust generalization in zero-shot, few-shot, and language modeling tasks while achieving a $1.36\times$ speedup at 50% sparsity level over the dense model. Our contributions are summarized as follows:

- We empirically validate the existence of

dataset-specific neurons in LLMs, demonstrating that using only calibration datasets fails to identify such neurons for diverse downstream datasets (§3.1).

- We propose a **training-free dataset-specific structured pruning** method that adaptively updates the proposed neuron importance scores on-the-fly and employs progressive pruning with exponentially increasing intervals to enhance efficiency (§3.3), accompanied by a theoretical analysis that motivates both design choices (§3.4).
- Extensive experiments demonstrate that DSPP substantially outperforms SOTA pruning methods with minimal overhead, exhibiting robust generalization across various datasets and models (§§4–5).

2 Related Works

2.1 Static Structured Pruning

Static structured pruning (An et al., 2024; Wang et al., 2025b; Ma et al., 2023; Ashkboos et al., 2024) obtains a fixed pruning mask on a small calibration dataset according to predefined importance metric and applies the same mask uniformly during inference. For instance, FLAP (An et al., 2024) introduces a fluctuation-based importance metric with bias compensation to minimize accuracy degradation, while CFSP (Wang et al., 2025b) optimizes model performance through layer-wise sparsity allocation. LLM-Pruner (Ma et al., 2023) employs structured second-order Taylor expansion to identify and remove less important neurons. Although static structured pruning significantly accelerates inference, it typically suffers from considerable performance degradation due to the distribution shift between calibration and test data.

2.2 Dynamic Structured Pruning

Dynamic structured pruning generates input-dependent pruning masks at test time, thereby achieving superior generalization across diverse test samples. However, most existing dynamic approaches (Song et al., 2024, 2025; Liu et al., 2023; Akhauri et al., 2024) rely on training auxiliary predictors to estimate importance scores of neurons for each sample, which introduces substantial computational overhead and memory footprint. Recently, training-free dynamic pruning strategies have emerged. Probe Pruning (Le et al., 2025) computes batch-wise pruning masks using a small set

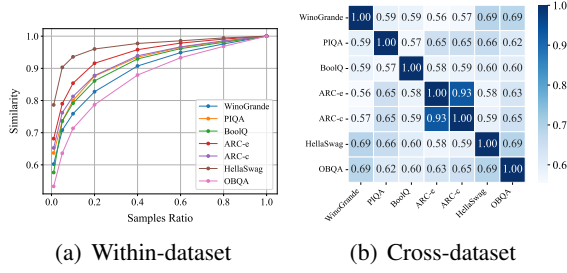


Figure 2: Jaccard similarity of pruning masks at 50% sparsity on LLaMA3.1-8B. (a) Within-dataset masks at sampling ratio α vs. the full-dataset reference. (b) Pairwise full-dataset masks across the seven datasets.

of probe samples and key tokens within the current batch, which are applied to the remaining samples. However, it degrades significantly when the batch size is 1 and still requires an initial calibration set. SEAP (Liang et al., 2026) builds task-specific knowledge corpus for each downstream task and derives masks from activation statistics on these corpora. In contrast, our method requires no calibration data or task-specific corpus and directly constructs dataset-specific pruning masks during inference.

3 Method

3.1 Motivation: Dataset-Specific Neurons

To test whether stable, dataset-level importance patterns exist, we compute pruning masks within and across seven datasets: WinoGrande (Sakaguchi et al., 2020), PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), ARC-easy (Clark et al., 2018), ARC-challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019) and OpenBookQA (Mihaylov et al., 2018). For each dataset we obtain a reference mask $\bar{\mathbf{M}}$ from the full dataset and partial masks $\mathbf{M}_\alpha$ from sampling ratios $\alpha \in \{1\%, 5\%, 10\%, 20\%, 40\%\}$. The layer-averaged Jaccard similarity is:

$$\bar{S}_\alpha = \frac{1}{L} \sum_{j=0}^{L-1} \text{Jaccard}(\mathbf{M}_\alpha^{(j)}, \bar{\mathbf{M}}^{(j)}), \quad (1)$$

where L is the number of layers.

Figure 2(a) shows that within-dataset similarity rises rapidly with α : with only 20% of the samples the Jaccard similarity already exceeds 80% across all seven datasets. In contrast, pairwise across-dataset similarity (Fig. 2(b)) stays below 70% (the lone exception being the closely related ARC-easy/ARC-challenge pair). The combination

of high intra- and low inter-dataset similarity provides strong empirical evidence for the existence of *dataset-specific neurons*—a stable, identifiable subset of MLP neurons that dominates the model’s behavior on a given dataset and differs systematically across datasets.

We formally quantify the strength of evidence for this pattern. Pairing each dataset’s within-Jaccard at $\alpha = 20\%$ (Fig. 2(a)) with its mean off-diagonal cross-Jaccard from Fig. 2(b), a paired t -test on the seven differences yields $t = 11.0$ on $\nu = n - 1 = 6$ degrees of freedom, two-sided $p < 10^{-4}$, and Cohen’s $d = 4.17$ (well above Cohen’s 0.8 threshold for “large” effects). The within-Jaccard mean of 0.87 ± 0.06 thus systematically and substantially exceeds the cross-Jaccard mean of 0.63 ± 0.03 , reinforcing the dataset-specific neuron hypothesis.

3.2 Pruning Metric

Given a large language model θ (e.g., LLaMA). In the j -th transformer layer, the MLP consists of a gate projection $W_g^j \in \mathbb{R}^{p \times q}$, an up projection $W_u^j \in \mathbb{R}^{p \times q}$ and a down projection $W_d^j \in \mathbb{R}^{q \times p}$, where p is the hidden dimension and q is the intermediate dimension. For an input $X_{ij} \in \mathbb{R}^{m_i \times p}$ to the j -th layer of sample i (with m_i tokens), the MLP output can be expressed as:

$$O_{ij} = (\sigma(X_{ij}W_g^j) \odot X_{ij}W_u^j)W_d^j = A_{ij}W_d^j, \quad (2)$$

where $\sigma(\cdot)$ denotes the activation function and A_{ij} is the intermediate activation.

The output of an MLP layer is jointly determined by activations and down-projection weights. Inspired by Wanda (Sun et al., 2024), we adopt a structured joint formulation in the combined parameter-activation space, scoring each dimension $k \in [0, q - 1]$ via the ℓ_2 -norm. To additionally capture magnitude dispersion across tokens, we introduce the variance of $|A_{ij}|$, which is large on neurons driven by a few high-magnitude “outlier” tokens that disproportionately determine post-pruning behaviour.

Formally, the activation-side score, weight-side score, and variance term are:

$$\mathbf{R}_{ij}(k) = \|A_{ij}[:, k]\|_2, \quad (3)$$

$$\mathbf{Q}_j(k) = \|W_d^j[k, :]\|_2, \quad (4)$$

$$\mathbf{V}_{ij}(k) = \text{Var}(|A_{ij}[:, k]|). \quad (5)$$

The fused structured importance score is the

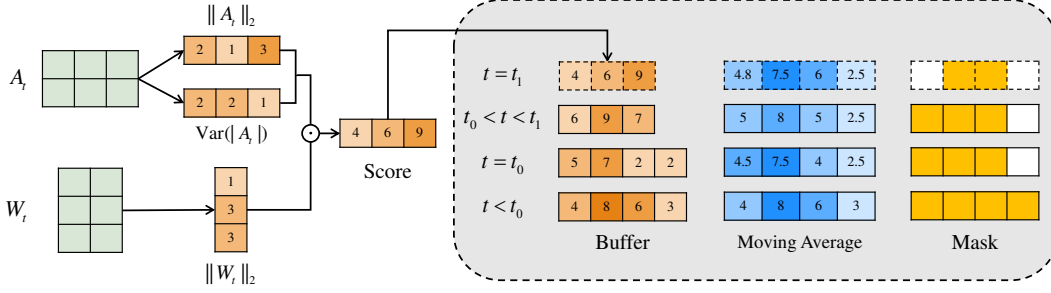


Figure 3: Overview of DSPP. As deployment-time inputs stream in, the model continuously accumulates $\|A_t\|_2$, $\|W_d\|_2$ and $\text{Var}(|A_t|)$ into a per-neuron importance buffer via CMA. At $t = t_0$ the accumulated buffer drives the first pruning step. At $t_1, t_2, \dots$ progressively higher sparsity levels are applied, with intervals growing geometrically. After $t = T$ the buffer freezes and inference proceeds with the final mask.

Algorithm 1 DSPP: Dataset-Specific Progressive Pruning

Require: model θ with L MLP layers (width q); target sparsity s^* ; warm-up budget T ; initial step t_0 ; growth factor γ

- 1: $\{t_i\}_{i=0}^{K-1} \leftarrow$ pruning steps from Eq. 8, $K = \max\{i : t_i \leq T\}$
- 2: $Q_j \leftarrow \mathbf{0} \in \mathbb{R}^q, \forall j \in [L]$
- 3: $i \leftarrow 0$
- 4: **for** $t = 1, \dots, T$ **do**
- 5: $\{A_{t,j}\}_{j=1}^L \leftarrow \text{FORWARD}(\theta; x_t)$
- 6: $Q_j \leftarrow \frac{t-1}{t} Q_j + \frac{1}{t} \text{Im}_{t,j}, \forall j$
- 7: **if** $t = t_i$ **and** $i < K$ **then**
- 8: $s^{(i)} \leftarrow (i+1) s^* / K$
- 9: $k \leftarrow \lceil (1 - s^{(i)}) q \rceil$
- 10: $\mathcal{M}_j \leftarrow \text{Top-}k(Q_j), \forall j$
- 11: $\theta \leftarrow \text{PRUNE}(\theta; \{\mathcal{M}_j\})$
- 12: $i \leftarrow i + 1$
- 13: **end if**
- 14: **end for**
- 15: **return** pruned θ used for all inputs $t > T$

element-wise product:

$$\text{Im}_{ij} = \mathbf{R}_{ij} \odot \mathbf{Q}_j^\top \odot \mathbf{V}_{ij}. \quad (6)$$

This score prioritizes dimensions that exhibit simultaneously large activations, large weight norms, and pronounced magnitude variance.

The three terms in Eq. 6 probe different aspects of a neuron’s contribution: the activation magnitude $\mathbf{R}$, the output coupling strength $\mathbf{Q}$, and the activation concentration $\mathbf{V}$. We verify in §5.1 that all three are jointly necessary, as removing any single term degrades the average accuracy.

3.3 DSPP: Dataset-Specific Progressive Pruning

Based on the dataset-specific neurons, we propose DSPP (illustrated in Figure 3), an efficient structured pruning method that dynamically refines the importance scores during inference.

Let $\mathcal{D}$ denote a test dataset of N samples. To prevent test-set leakage and ensure a realistic online evaluation setting, we process test samples sequentially as they arrive over time. At time step t , we first compute the current importance score $\text{Im}(t)$ using the proposed pruning metric in Eq. 6. This score is then stored in the buffer and subsequently used to update the cumulative score $Q(t)$ via **Cumulative Moving Average (CMA)**. For simplicity, we assume that one new test sample arrives at each time step, allowing $Q(t)$ to be expressed as:

$$Q(t) = \frac{1}{t} \sum_{\tau=1}^t \text{Im}(\tau), \quad t \leq T, \quad (7)$$

where T represents the total number of samples used to update $Q(t)$, corresponding to the first $p\%$ of the dataset.

Progressive pruning with exponentially growing intervals. Instead of pruning once at the end of the warm-up, we apply pruning iteratively, increasing sparsity in stages. To balance accuracy and efficiency, the interval between successive pruning steps grows geometrically:

$$t_i = \frac{t_0(1 - \gamma^{i+1})}{1 - \gamma}, \quad \gamma > 1, \quad (8)$$

where t_0 is the timing of the first pruning step. Higher sparsity levels are reached only after

more inputs have been processed and the importance scores have stabilized. We use $t_0 = \max(0.01N, 1)$ and $\gamma = 2$ throughout.

Algorithmic summary. Algorithm 1 summarizes the procedure. The only deployment-time bookkeeping is the per-layer CMA buffer (size q , identical for every layer), so the total memory overhead is $\mathcal{O}(Lq)$, dominated by the model’s own activations.

3.4 Theoretical Analysis

We motivate (i) why dataset-specific neurons exist and (ii) the sample budget DSPP needs to identify them.

Why dataset-specific neurons exist. Let D be a categorical random variable indexing the dataset from which an input is drawn, and let $A_k \in \mathbb{R}_{\geq 0}$ denote the per-token absolute activation at the k -th intermediate dimension, marginalised over tokens and prompts. The dataset-induced shift of A_k ’s distribution is quantified by the mutual information:

$$I(A_k; D) = \mathbb{E}_D[\text{KL}(p(A_k | D) \| p(A_k))], \quad (9)$$

where $\text{KL}(\cdot \| \cdot)$ is the Kullback–Leibler divergence and $\mathbb{E}_D$ averages over the dataset prior $\text{Pr}(D)$. Neurons with $I(A_k; D) \approx 0$ are activated identically across datasets, so pruning them costs the same on every D ; neurons with $I(A_k; D) \gg 0$ are activated differentially and must be preserved when adapting to a target dataset d^* . These high- I neurons are the formal counterpart of the dataset-specific neurons hypothesised in §3.1. In our fused metric, $\mathbf{R}$ and $\mathbf{V}$ are sample moments of $p(A_k | D = d^*)$ and adapt to the target distribution, while $\mathbf{Q}$ provides the data-independent output-coupling factor. Their product $\mathbf{R} \odot \mathbf{Q} \odot \mathbf{V}$ thus assigns each neuron a target-aware, output-weighted importance, whose top- k ranking is the dataset-specific neuron set that DSPP retains online.

How many samples suffice. DSPP estimates $\mathbf{Im}$ via the per-layer CMA $Q_j(t) = \frac{1}{t} \sum_{\tau=1}^t \mathbf{Im}(\tau)$ over the first t deployment prompts, where $j \in [L]$ indexes the MLP layer (suppressed in §3.3 for brevity; within-prompt tokens are correlated through attention, so the unit of independence is the prompt). Treating the prompt-level scores as approximately sub-Gaussian, standard concentration inequalities (Hoeffding, 1963; Boucheron et al.,

2013) yield, with probability at least $1 - \delta$ uniformly over $k \in [q]$:

$$\max_k |Q_j(t)(k) - \mu_k| \leq C \sigma_{\max} \sqrt{\frac{\log(2q/\delta)}{t}}, \quad (10)$$

where $\mu_k := \mathbb{E}[\mathbf{Im}(\tau)(k)]$, $\sigma_{\max}^2$ upper-bounds the per-prompt variance, q is the intermediate width, C is the universal constant, and the $\max_k$ follows from a union bound on the q coordinates.

DSPP keeps the top- k coordinates of $Q_j(t)$. Ordering $\mu_{(1)} \geq \dots \geq \mu_{(q)}$ and letting $\Delta_k := \mu_{(k)} - \mu_{(k+1)}$, the empirical and population top- k sets coincide whenever the deviation in Eq. 10 drops below $\Delta_k/2$, giving:

$$t \geq \frac{4C^2 \sigma_{\max}^2 \log(2q/\delta)}{\Delta_k^2}. \quad (11)$$

The sample count grows only logarithmically in q and δ , and is governed by the variance-to-gap ratio $\sigma_{\max}^2/\Delta_k^2$.

The $\mathcal{O}(1/\sqrt{t})$ rate also dictates the geometric growth of pruning intervals in §3.3: writing $\epsilon_i := C \sigma_{\max} \sqrt{\log(2q/\delta)/t_i}$, a geometric schedule ($t_{i+1}/t_i \rightarrow \gamma$) keeps $\epsilon_{i+1}/\epsilon_i = 1/\sqrt{\gamma}$ constant, so each step earns the same relative deviation reduction. A uniform schedule ($t_i = it_0$) instead gives $\sqrt{i/(i+1)} \rightarrow 1$, wasting late samples. §5.3 and §5.4 empirically verify both predictions.

4 Experiments

4.1 Experimental Setup

Models and Datasets We evaluate DSPP on four LLaMA-family models—LLaMA2-7B/13B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024), LLaMA3.1-8B (Grattafiori et al., 2024)—and additionally on Qwen2.5-7B (Qwen et al., 2025). Few-shot math is evaluated on GSM8K (5-shot) (Cobbe et al., 2021) and MATH (5-shot) (Hendrycks et al., 2021b). Zero-shot accuracy is reported on eight datasets: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), WinoGrande (Sakaguchi et al., 2020), OpenBookQA (Mihaylov et al., 2018), MMLU (Hendrycks et al., 2021a), HelLaSwag (Zellers et al., 2019), ARC-easy (Clark et al., 2018) and ARC-challenge (Clark et al., 2018). Perplexity is evaluated on WikiText-2 (Merity et al., 2017), the PTB (Marcus et al., 1993) validation set, and the PG-19 (Rae et al., 2020) validation set. More evaluation details can be found in Appendix A.

Baselines We compare **DSPP** with three representative static methods and one dynamic method. Among the static methods, **Wanda-SP** (An et al., 2024) is the extension of Wanda (Sun et al., 2024) for structured pruning. **FLAP** (An et al., 2024) introduces bias compensation to alleviate performance degradation. **CFSP** (Wang et al., 2025b) is the SOTA static structured pruning method with layer-wise sparsity allocation. The dynamic baseline, **SEAP** (Liang et al., 2026), generates task-specific pruning masks on-the-fly by leveraging corpus information extracted from the target task. Additionally, **DSPP+CFSP** combines DSPP with the sparsity allocation strategy of CFSP. For fairness, all methods only prune MLP layers and are applied without any post-pruning fine-tuning.

Implementation Details All experiments use NVIDIA A800-80GB or A100-40GB GPUs. We adopt the official implementations of CFSP (Wang et al., 2025b), Wanda-SP (An et al., 2024) and FLAP (An et al., 2024), and for SEAP (Liang et al., 2026) use Expert-based Pruning with the WIFV metric (the best variant in Liang et al. (2026)). Pruning is applied only during the pre-fill phase, as its latency is compute-bound. Letting N be the total number of test samples, we set $T = \max(0.1N, 10)$, $t_0 = \max(0.01N, 1)$, $\gamma = 2$, and report results at 20% and 50% sparsity. Unless explicitly marked as stream-average, these percentages denote the terminal sparsity reached after warm-up.

4.2 Zero-Shot and Few-Shot Performance

Table 1 reports zero-shot and few-shot performance on LLaMA3.1-8B; the remaining models are reported in Appendix B. DSPP consistently outperforms all baselines at both sparsity levels. Notably, at 50% sparsity DSPP achieves 49.67 average accuracy vs. CFSP’s 41.51 (a relative gain of $\sim 20\%$) and SEAP’s 47.37. DSPP+CFSP further pushes this to 51.86. Per-task gains over CFSP at 50% sparsity include +13.95 on HellaSwag, +12.92 on ARC-e, and +11.01 on MMLU.

4.3 Language Modeling Perplexity

Table 2 reports perplexity on the three language-modeling benchmarks. DSPP (and DSPP+CFSP) achieves the best average perplexity on every task except WikiText-2, where CFSP retains a small lead because CFSP’s calibration set is itself WikiText-2.

4.4 Inference Speed

We measure end-to-end forward time on 3,000 PIQA samples (sequence length 2,048, batch size 1) at 50% sparsity. Table 3 shows DSPP achieves $1.36\times$ speedup over the dense model, vs. $1.42\times$ for CFSP. The $\sim 4\%$ relative gap arises from computing importance scores during the warm-up phase; once $t > T$, DSPP’s inference cost is identical to a static pruned model. Given the $\sim 20\%$ relative accuracy advantage (Table 1), DSPP is markedly more favorable on the accuracy–speedup Pareto frontier.

5 Ablation and Analysis

5.1 Metric Ablation

To verify that the three components $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{V}$ of our fused metric are jointly indispensable, we evaluate the three two-term subsets: w/o VAR ($\mathbf{R} \odot \mathbf{Q}$), w/o W ($\mathbf{R} \odot \mathbf{V}$), and w/o A ($\mathbf{Q} \odot \mathbf{V}$). All variants run at 50% sparsity on 7 accuracy tasks (ARC-c/e, BoolQ, HellaSwag, OBQA, PIQA, WinoGrande) and 3 perplexity datasets (WikiText-2, PTB, PG-19).

Three observations follow from Table 4. (a) Dropping the variance term ($\mathbf{R} \odot \mathbf{Q}$) loses 1.95 pp of accuracy and 1.19 perplexity, with the accuracy gap concentrated on HellaSwag (-7.07 pp) where outlier-driven activations dominate. (b) Dropping the magnitude term ($\mathbf{Q} \odot \mathbf{V}$) is closer on accuracy (-0.18 pp) but loses 0.60 on PPL. (c) Dropping the weight-norm term ($\mathbf{R} \odot \mathbf{V}$) costs 0.21 pp on accuracy while being essentially tied on PPL. All three terms are therefore independently necessary, confirming the three-way fusion hypothesised in §3.2.

Disentangling the procedure and metric. Table 5 tests whether the progressive procedure generalizes across importance metrics and whether DSPP’s metric advantage persists under equal access to target-domain statistics. We keep the target stream, CMA update, warm-up, exponential schedule, and uniform MLP-only sparsity identical to DSPP, and replace only Eq. 6 with the official Wanda-SP, FLAP, or CFSP metric (with CFSP layer allocation disabled).

Table 5 supports both points. First, at 50% sparsity, every transplanted metric gains 5.10–10.68 accuracy points over its static counterpart in Table 1, showing that the progressive framework generalizes beyond our metric. Second, all four rows

Sparsity	Method	GSM8k	MATH	PIQA	WinoGrande	BoolQ	OBQA	MLLM	HellaSwag	ARC-e	ARC-c	Average
0%	LLaMA3.1-8B	49.43	14.02	81.34	73.72	82.08	45.00	63.30	79.01	81.06	53.50	62.25
20%	FLAP	25.63	7.84	75.03	64.96	70.61	40.60	40.54	62.72	61.03	37.71	48.67
	Wanda-SP	40.49	10.70	75.79	69.38	75.17	41.20	51.74	69.80	68.86	42.15	54.53
	CFSP	44.20	11.70	76.93	70.96	78.50	41.20	54.71	72.41	70.33	45.14	56.61
	SEAP	47.76	13.82	79.16	73.80	79.36	42.20	54.27	75.84	77.06	49.83	59.31
	DSPP	49.43	12.78	79.27	72.30	81.07	43.20	59.08	76.50	73.61	51.45	59.87
	DSPP+CFSP	49.28	13.36	80.30	72.22	80.89	43.00	59.78	76.91	78.32	52.22	60.63
50%	FLAP	6.82	3.72	62.89	56.67	62.45	31.00	23.77	41.87	41.79	25.77	35.68
	Wanda-SP	12.66	4.14	63.06	57.85	50.00	31.00	24.64	43.87	44.82	28.58	36.06
	CFSP	24.49	7.02	66.43	60.22	62.45	32.00	33.87	49.59	48.99	30.03	41.51
	SEAP	33.89	1.06	75.95	67.56	62.94	39.60	33.02	60.62	61.78	37.29	47.37
	DSPP	31.92	9.06	74.43	66.30	71.22	34.20	44.88	63.54	61.91	39.25	49.67
	DSPP+CFSP	37.91	10.32	75.35	66.14	69.48	36.40	48.62	65.83	67.38	41.13	51.86

Table 1: Zero-shot and few-shot accuracy ($\uparrow$) on LLaMA3.1-8B at 20% and 50% sparsity.

Sparsity	Method	WikiText-2 $\downarrow$	PTB $\downarrow$	PG-19 $\downarrow$	Avg $\downarrow$
0%	LLaMA3.1-8B	6.96	11.73	10.58	9.76
20%	FLAP	9.54	15.91	15.85	13.77
	Wanda-SP	9.21	14.45	14.51	12.72
	CFSP	8.99	13.91	13.91	12.27
	SEAP	10.24	14.44	15.06	13.25
	DSPP	9.37	13.65	12.57	11.86
	DSPP+CFSP	9.27	13.33	12.59	11.73
50%	FLAP	21.04	33.22	38.09	30.78
	Wanda-SP	19.72	37.12	38.78	31.87
	CFSP	17.43	24.71	28.23	23.46
	SEAP	21.42	27.00	29.24	25.89
	DSPP	21.15	24.97	24.08	23.40
	DSPP+CFSP	19.49	22.94	23.06	21.83

Table 2: Perplexity ($\downarrow$) on LLaMA3.1-8B at 20% and 50% sparsity.

Method	Time (s/sample)	Speedup
Dense Model	0.1914	1.00 $\times$
CFSP	0.1348	1.42 $\times$
DSPP (ours)	0.1403	1.36 $\times$

Table 3: Inference speedup on LLaMA3.1-8B at 50% sparsity.

Metric variant	Avg Acc $\uparrow$ (7 tasks)	Avg PPL $\downarrow$ (3 datasets)
DSPP w/o Var ($\mathbf{R} \odot \mathbf{Q}$)	56.77	24.62
DSPP w/o W ($\mathbf{R} \odot \mathbf{V}$)	58.52	23.40
DSPP w/o A ($\mathbf{Q} \odot \mathbf{V}$)	58.55	24.03
DSPP (full $\mathbf{R} \odot \mathbf{Q} \odot \mathbf{V}$)	58.73	23.43

Table 4: Metric ablation on LLaMA3.1-8B at 50% sparsity. Each two-term subset removes one of $\mathbf{R}$, $\mathbf{Q}$, $\mathbf{V}$.

receive the same target stream and use the same CMA and schedule, so DSPP’s margin here cannot be explained by privileged access to target-domain statistics. Thus the procedure is metric-agnostic, while the matched comparison supports Eq. 6 as the stronger pruning principle.

Metric	20% sparsity		50% sparsity	
	Acc. $\uparrow$	PPL $\downarrow$	Acc. $\uparrow$	PPL $\downarrow$
Wanda-SP	56.38	12.06	46.16	24.81
FLAP	56.60	12.41	46.36	25.38
CFSP	56.39	11.92	46.61	23.77
DSPP	59.87	11.86	49.67	23.40

Table 5: Controlled metric swap on LLaMA3.1-8B. Accuracy averages the 10 tasks in Table 1; PPL averages the three datasets in Table 2.

5.2 Cross-Dataset Mask Transfer

To verify that DSPP’s mask reflects genuine dataset structure rather than overfitting to the accumulation inputs, we measure transfer accuracy: for each pair (A, B) we accumulate the importance on the first T inputs of source A and evaluate the pruned model on the full test set of target B . Every cell uses a uniform 50% per-layer sparsity (CFSP’s layer-wise allocation disabled) so that only the metric-transfer effect is measured.

Figure 4 reports the resulting 6×6 matrix. The diagonal (within-dataset DSPP, red boxes) wins on four of the six targets (ARC-c, HellaSwag, PIQA, WinoGrande). Off-diagonal entries otherwise decay smoothly with source–target distance: closely related sources (e.g. HellaSwag $\rightarrow$ PIQA) recover most of the in-distribution accuracy, whereas structurally distant ones (e.g. WinoGrande $\rightarrow$ ARC-c) lose more. Together, the diagonal advantage and the smooth distance-dependent decay embody the dataset-specific high- I neuron sets posited in §3.4.

	ARC-c	BoolQ	HellaSwag	OBQA	PIQA	WinoGrande
ARC-c	36.7	63.9	53.7	37.2	70.5	58.6
BoolQ	28.9	61.7	50.5	34.8	68.1	58.3
HellaSwag	29.4	61.8	60.4	37.2	72.6	63.5
OBQA	35.0	58.8	54.2	37.0	72.2	59.8
PIQA	31.2	63.5	58.7	35.2	74.1	58.0
WinoGrande	26.4	60.0	57.0	34.4	71.3	65.3
	ARC-c	BoolQ	HellaSwag	OBQA	PIQA	WinoGrande

Figure 4: Cross-dataset transfer accuracy on LLaMA3.1-8B at 50% sparsity. Rows are source datasets, columns are targets; red boxes mark the within-dataset diagonal. Colour is normalised per column (min to max for each target), so the gradient inside each column reflects the relative ranking of sources for that target.

5.3 Effect of Test-Set Ratio

To further validate that dataset-specific neurons can be identified from a small input sample, we measure post-pruning accuracy as a function of the test-set ratio $T/N \in \{0.1\%, 0.5\%, 1\%, 5\%, 10\%, 20\%, 40\%\}$. For each T we compute the mask using a dense model on the first T samples and re-evaluate the pruned model on the full dataset to avoid any bias from the warm-up samples. Figure 5 shows that across ARC-e, HellaSwag, PIQA and WinoGrande, the accuracy plateaus by $T \sim 0.5\%$ of the data—decisively outperforming the static CFSP baseline at every $T \geq 0.1\%$. Together with the Jaccard analysis (§3.1), this is direct evidence that dataset-specific neurons are stable and can be efficiently identified from only a small fraction of the input distribution.

5.4 The Benefits of Progressive Pruning

To quantify the contribution of the progressive schedule, we compare DSPP against a “W/O PROGRESSIVE” variant that prunes only at $t = T$, at 50% sparsity on LLaMA3.1-8B. Fixing $t_0 = 1\%N$ and varying $T \in \{5\%, 10\%, 15\%, 20\%, 25\%\}N$ (Fig. 6), and then fixing $T = 10\%N$ and varying $t_0 \in \{0.1\%, 0.5\%, 1\%, 1.5\%, 2\%\}N$ (Fig. 7), we see that the progressive schedule consistently delivers higher speedup with no measurable loss in accuracy. The exponential growth factor allocates more samples between pruning steps as the estimate stabilises, so dense passes dominate the early noisy phase and pruning is only committed once the importance ranking is reliable. These observa-

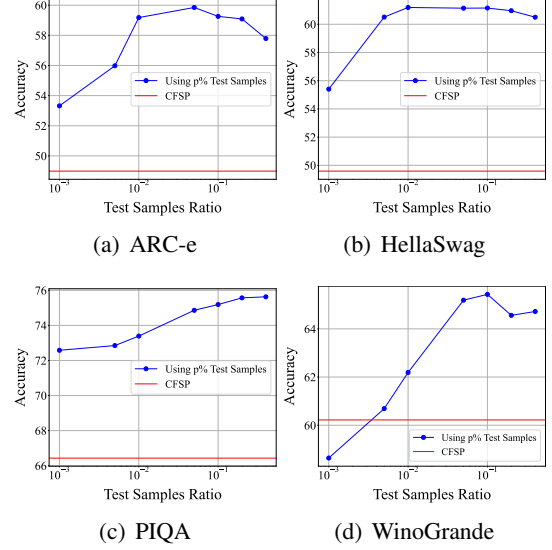


Figure 5: Accuracy under DSPP at 50% sparsity, as a function of the warm-up ratio T/N , on LLaMA3.1-8B. Accuracy converges by $T \sim 0.5\%$ on every dataset.

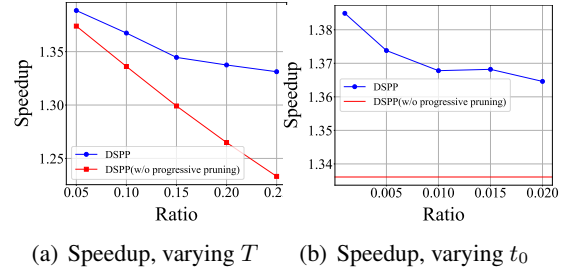


Figure 6: Speedup of DSPP vs. DSPP w/o progressive pruning across warm-up configurations.

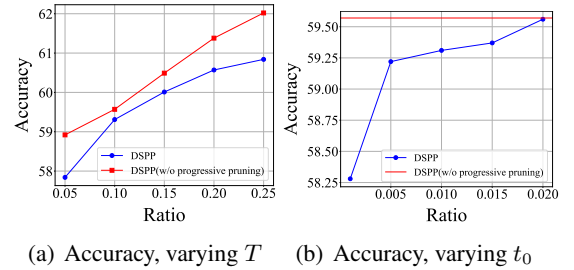


Figure 7: Accuracy of DSPP vs. DSPP w/o progressive pruning across warm-up configurations.

tions are consistent with the theoretical analysis in §3.4: the $\mathcal{O}(1/\sqrt{t})$ concentration rate of the CMA estimator (Eq. 10) renders early aggressive pruning unreliable, while the exponential schedule allocates increasing sample budgets to successive pruning steps in precise correspondence with this rate.

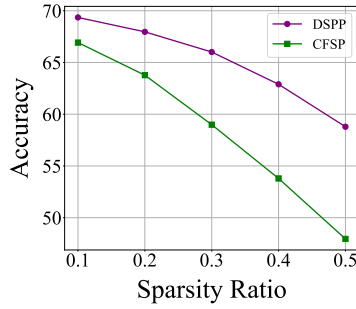


Figure 8: Average zero-shot accuracy of CFSP and DSPP on LLaMA3.1-8B under varying sparsity ratios.

5.5 Performance Across Sparsity Levels

To validate the effectiveness of DSPP across different sparsity levels, we evaluate the average zero-shot accuracy on LLaMA3.1-8B for DSPP and CFSP under sparsity ratios from 10% to 50%, as shown in Figure 8. DSPP consistently achieves higher accuracy than CFSP across all sparsity ratios, and the performance gap widens as sparsity increases. This further demonstrates DSPP’s robust generalization across the full sparsity range.

6 Conclusion

In this work, we propose DSPP (Dataset-Specific Progressive Pruning), a simple and efficient structured pruning method that adaptively tailors LLMs to test data through dynamically identifying the dataset-specific neurons. By introducing the progressive pruning strategy with exponentially increasing pruning interval, DSPP achieves less computational overhead. A complementary theoretical analysis further explains these design choices. Extensive experiments demonstrate that DSPP significantly improves model performance and generalizes robustly across diverse datasets and models. We hope that this method can inspire more related research in the future.

Limitations

While DSPP delivers consistent improvements over existing pruning methods, two limitations remain. First, at high compression ratios (e.g., $\geq 50\%$ sparsity) the post-pruning accuracy still degrades noticeably; although DSPP retains a clear margin over static baselines in this regime, closing the remaining gap to the dense model under aggressive sparsity is an open problem. Second, because DSPP starts from a dense model and progressively prunes during inference, the warm-up phase incurs additional time and memory overhead before the

target sparsity is reached, so memory usage is not reduced in the early stage. Reducing this initial overhead, e.g. via an informed prior mask or partial early pruning, is a valuable direction for future work.

Acknowledgments

We thank the anonymous reviewers and area chairs for their constructive feedback. We used OpenAI Codex to assist with code implementation and language editing of the manuscript. The authors reviewed and verified all code changes, experimental results, analyses, citations, and final text, and take full responsibility for the content of this work.

References

- Yash Akhauri, Ahmed F. AbouElhamayed, Jordan Dotzel, Zhiru Zhang, Alexander M. Rush, Safeen Huda, and Mohamed S. Abdelfattah. 2024. [ShadowLLM: Predictor-based contextual sparsity for large language models](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 19154–19167, Miami, Florida, USA. Association for Computational Linguistics.
- Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. [Fluctuation-based adaptive structured pruning for large language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(10):10865–10873.
- Md Adnan Arefeen, Biplob Debnath, and Srimat Chakradhar. 2024. [LeanContext: Cost-efficient domain-specific question answering using LLMs](#). *Natural Language Processing Journal*, 7:100065.
- Saleh Ashkboos, Maximilian L. Croci, Marcelo Genari do Nascimento, Torsten Hoeffler, and James Hensman. 2024. [SliceGPT: Compress large language models by deleting rows and columns](#). In *International Conference on Learning Representations*, volume 2024, pages 11682–11701.
- 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.
- Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. 2013. [Concentration Inequalities: A Nonasymptotic Theory of Independence](#). Oxford University Press.
- Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. 2024. [A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations](#). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):10558–10578.

- 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, Minneapolis, Minnesota. Association for Computational Linguistics.
- 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](#). *Preprint*, arXiv:1803.05457.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. [LLM.int8\(\): 8-bit matrix multiplication for transformers at scale](#). In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.
- Elias Frantar and Dan Alistarh. 2023. [SparseGPT: Massive language models can be accurately pruned in one-shot](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 540 others. 2024. [The Llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. [Measuring massive multitask language understanding](#). In *The Ninth International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.
- Wassily Hoeffding. 1963. [Probability inequalities for sums of bounded random variables](#). *Journal of the American Statistical Association*, 58(301):13–30.
- Bishwash Khanal and Jeffery M. Capone. 2024. [Evaluating the impact of compression techniques on task-specific performance of large language models](#). *Preprint*, arXiv:2409.11233.
- Sehoon Kim, Karttikeya Mangalam, Suhong Moon, Jitendra Malik, Michael W. Mahoney, Amir Gholami, and Kurt Keutzer. 2023. [Speculative decoding with big little decoder](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 39236–39256. Curran Associates, Inc.
- Qi Le, Enmao Diao, Ziyang Wang, Xinran Wang, Jie Ding, Li Yang, and Ali Anwar. 2025. [Probe pruning: Accelerating LLMs through dynamic pruning via model-probing](#). In *International Conference on Learning Representations*, volume 2025, pages 72494–72513.
- Xun Liang, Hanyu Wang, Huayi Lai, Simin Niu, Shichao Song, Jiawei Yang, Jihao Zhao, Feiyu Xiong, Bo Tang, and Zhiyu Li. 2026. [SEAP: Sparse expert activation pruning unlocks the brainpower of large language models](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(38):31934–31942.
- Zichang Liu, Jue Wang, Tri Dao, Tianyi Zhou, Binhang Yuan, Zhao Song, Anshumali Shrivastava, Ce Zhang, Yuandong Tian, Christopher Re, and Beidi Chen. 2023. [Deja vu: Contextual sparsity for efficient LLMs at inference time](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 22137–22176. PMLR.
- Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. [LLM-Pruner: On the structural pruning of large language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 21702–21720. Curran Associates, Inc.
- Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. [Building a large annotated corpus of English: The Penn Treebank](#). *Computational Linguistics*, 19(2):313–330.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. [Pointer sentinel mixture models](#). In *The Fifth International Conference on Learning Representations*.
- Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. [Can a suit of armor conduct electricity? a new dataset for open book question answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391, Brussels, Belgium. Association for Computational Linguistics.
- Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, and 24 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. [Compressive transformers for long-range sequence modelling](#). In *The Eighth International Conference on Learning Representations*.

- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. [WinoGrande: An adversarial winograd schema challenge at scale](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8732–8740.
- Chenyang Song, Xu Han, Zhengyan Zhang, Shengding Hu, Xiyu Shi, Kuai Li, Chen Chen, Zhiyuan Liu, Guangli Li, Tao Yang, and Maosong Sun. 2025. [ProSparse: Introducing and enhancing intrinsic activation sparsity within large language models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2626–2644, Abu Dhabi, UAE. Association for Computational Linguistics.
- Yixin Song, Haotong Xie, Zhengyan Zhang, Bo Wen, Li Ma, Zeyu Mi, and Haibo Chen. 2024. [Turbo sparse: Achieving LLM SOTA performance with minimal activated parameters](#). *Preprint*, arXiv:2406.05955.
- Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. [A simple and effective pruning approach for large language models](#). In *International Conference on Learning Representations*, volume 2024, pages 4942–4964.
- Lintang Sutawika, Hailey Schoelkopf, Leo Gao, Baber Abbasi, Stella Biderman, Jonathan Tow, ben fatori, Charles Lovering, farzanehnakhaee70, Jason Phang, Anish Thite, Fazz, Aflah, Niklas Muenighoff, Thomas Wang, sdtblck, nopperl, gakada, tttuntian, and 11 others. 2024. [EleutherAI/lm-evaluation-harness: v0.4.3](#).
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *Preprint*, arXiv:2307.09288.
- Pingjie Wang, Ziqing Fan, Shengchao Hu, Zhe Chen, Yanfeng Wang, and Yu Wang. 2025a. [Reconstruct the pruned model without retraining](#). *IEEE Journal of Selected Topics in Signal Processing*, 19(7):1279–1290.
- Yuxin Wang, MingHua Ma, Zekun Wang, Jingchang Chen, Shan Liping, Qing Yang, Dongliang Xu, Ming Liu, and Bing Qin. 2025b. [CFSP: An efficient structured pruning framework for LLMs with coarse-to-fine activation information](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9311–9328, Abu Dhabi, UAE. Association for Computational Linguistics.
- 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, Florence, Italy. Association for Computational Linguistics.
- Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. 2024. [Plug-and-play: An efficient post-training pruning method for large language models](#). In *International Conference on Learning Representations*, volume 2024, pages 50490–50508.
- Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. [How do large language models handle multilingualism?](#) In *Advances in Neural Information Processing Systems*, volume 37, pages 15296–15319. Curran Associates, Inc.

A Details of Evaluation

A.1 Zero-shot and Few-shot Tasks

We use the EleutherAI LM Evaluation Harness (Sutawika et al., 2024) to evaluate performance across eight zero-shot tasks and two few-shot tasks. The per-task metric is: GSM8K (Cobbe et al., 2021) (flexible-extract), MATH (Hendrycks et al., 2021b) (exact_match), ARC-challenge/easy (Clark et al., 2018) (normalized accuracy), BoolQ (Clark et al., 2019) (accuracy), HellaSwag (Zellers et al., 2019) (normalized accuracy), MMLU (Hendrycks et al., 2021a) (accuracy), OpenbookQA (Mihaylov et al., 2018) (normalized accuracy), PIQA (Bisk et al., 2020) (normalized accuracy), and WinoGrande (Sakaguchi et al., 2020) (accuracy).

A.2 Language Modeling Tasks

We evaluate three language modeling benchmarks: WikiText-2 (Merity et al., 2017), PTB (Marcus et al., 1993) and PG-19 (Rae et al., 2020). The input length is set to 1024 and we follow the evaluation protocol of Wanda (Sun et al., 2024).

A.3 Progressive Pruning Schedule

For the ablation in §5.4, we use six zero-shot tasks: ARC-c/e (Clark et al., 2018), BoolQ (Clark et al., 2019), OpenbookQA (Mihaylov et al., 2018), PIQA (Bisk et al., 2020) and WinoGrande (Sakaguchi et al., 2020). Since the growth factor γ has a minimal impact on accuracy and speedup in our preliminary sweep, we fix $\gamma = 2$ throughout the paper.

B Additional Results

B.1 Zero-Shot and Few-Shot Tasks

We further evaluate DSPP on three additional LLaMA models (LLaMA2-7B/13B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024)) and the cross-architecture Qwen2.5-7B (Qwen et al., 2025); the experimental details and baselines match

Sparsity	Method	WikiText-2↓	PTB↓	PG-19↓	Avg↓
0%	LLaMA2-7B	6.10	22.08	8.85	12.34
20%	FLAP	7.26	26.42	10.90	14.86
	Wanda-SP	7.04	25.36	10.61	14.34
	CFSP	6.97	25.14	10.32	14.14
	SEAP	7.71	24.72	11.04	14.49
	DSPP	7.33	24.63	9.87	13.94
	DSPP+CFSP	7.24	23.91	9.89	13.68
50%	FLAP	14.09	61.78	22.04	32.64
	Wanda-SP	12.75	67.79	21.47	34.00
	CFSP	11.60	43.68	18.10	24.46
	SEAP	13.51	42.33	18.57	24.80
	DSPP	14.92	47.44	16.08	26.15
	DSPP+CFSP	13.80	40.46	15.77	23.34

Table 6: Perplexity (↓) of pruning methods on LLaMA2-7B at 20% and 50% sparsity.

Sparsity	Method	WikiText-2↓	PTB↓	PG-19↓	Avg↓
0%	LLaMA3-8B	6.82	11.74	10.40	9.65
20%	FLAP	9.51	16.26	16.38	14.05
	Wanda-SP	9.12	14.50	14.54	12.72
	CFSP	8.97	13.89	13.86	12.24
	SEAP	9.79	14.21	14.52	12.84
	DSPP	9.29	13.63	12.52	11.81
	DSPP+CFSP	9.19	13.35	12.52	11.69
50%	FLAP	21.13	36.92	41.49	33.18
	Wanda-SP	20.58	40.36	40.49	33.81
	CFSP	17.43	24.94	28.06	23.48
	SEAP	22.19	26.24	29.52	25.98
	DSPP	21.59	24.94	24.70	23.74
	DSPP+CFSP	19.74	22.82	23.55	21.84

Table 7: Perplexity (↓) of pruning methods on LLaMA3-8B at 20% and 50% sparsity.

Sparsity	Method	WikiText-2↓	PTB↓	PG-19↓	Avg↓
0%	Qwen2.5-7B	7.64	13.37	12.01	11.01
20%	FLAP	9.10	16.69	14.65	13.48
	Wanda-SP	9.34	18.41	15.32	14.36
	CFSP	9.16	16.55	14.59	13.43
	SEAP	9.95	16.01	15.60	13.85
	DSPP	9.82	14.82	13.36	12.67
	DSPP+CFSP	9.71	14.85	13.40	12.65
50%	FLAP	19.23	50.43	36.26	35.31
	Wanda-SP	16.35	35.56	28.61	26.84
	CFSP	15.67	31.13	27.15	24.65
	SEAP	19.60	28.26	28.23	25.36
	DSPP	18.08	22.46	20.96	20.50
	DSPP+CFSP	17.79	23.05	20.74	20.53

Table 8: Perplexity (↓) of pruning methods on Qwen2.5-7B at 20% and 50% sparsity.

LLaMA3.1-8B. Results are reported in Table 9 (LLaMA2-7B), Table 10 (LLaMA2-13B), Table 11 (LLaMA3-8B), and Table 12 (Qwen2.5-7B).

DSPP and DSPP+CFSP consistently obtain the best average accuracy at both 20% and 50% sparsity across all four backbones, indicating that the dataset-specific neuron pattern generalises across model scales (7B–13B) and architecture families

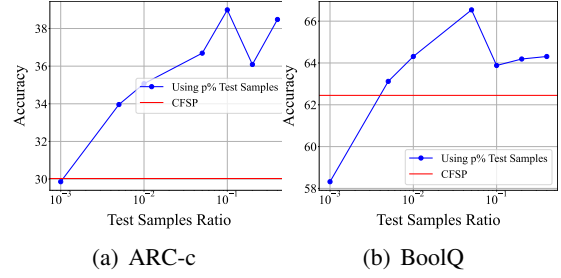


Figure 9: Accuracy under DSPP at 50% sparsity, as a function of the warm-up ratio T/N , on LLaMA3.1-8B for ARC-c and BoolQ.

(LLaMA / Qwen). At 50% sparsity, DSPP retains roughly 85–90% of the dense accuracy on every model, whereas the strongest static baseline drops considerably more. The cross-architecture transfer is particularly clean on Qwen2.5-7B; one notable exception is that DSPP+CFSP no longer consistently improves over DSPP alone on Qwen, suggesting that CFSP’s layer-wise sparsity allocation is partly tuned to LLaMA-family layer importance, whereas DSPP’s neuron-level mask transfers more cleanly across architectures.

B.2 Language Modeling Tasks

We evaluate perplexity on LLaMA2-7B (Touvron et al., 2023), LLaMA3-8B (Grattafiori et al., 2024) and Qwen2.5-7B (Qwen et al., 2025) under the same experimental setup and baselines. Results are reported in Table 6, Table 7 and Table 8. DSPP (and DSPP+CFSP) achieves the best average perplexity on every model at both sparsity targets, with the only exception being WikiText-2, where CFSP retains a small lead because CFSP’s calibration set is itself WikiText-2.

B.3 Effect of Test Set Ratio on Additional Datasets

We expand the §5.3 experiments by including two additional datasets, ARC-challenge (Clark et al., 2018) and BoolQ (Clark et al., 2019); the experimental conditions are unchanged. Figure 9 reports accuracy as a function of the warm-up ratio T/N on these two datasets. Consistent with §5.3, accuracy plateaus at a similarly small ratio ($T/N \sim 0.5\%$) on both ARC-challenge and BoolQ, reinforcing that DSPP’s dataset-specific neuron identification is sample-efficient across diverse task formats.

Sparsity	Method	GSM8k	MATH	PIQA	WinoGrande	BoolQ	OBQA	MMLU	HellaSwag	ARC-e	ARC-c	Average
0%	LLaMA2-7B	14.03	5.54	79.11	69.14	77.71	44.20	41.82	76.02	74.49	46.33	52.84
20%	FLAP	10.11	5.38	75.79	65.59	67.98	40.00	28.73	69.70	60.94	37.29	46.15
	Wanda-SP	11.75	4.98	76.33	68.59	70.86	40.80	34.72	71.95	69.07	40.87	48.99
	CFSP	12.21	5.40	76.71	67.72	67.74	40.60	35.24	72.08	68.86	41.55	48.81
	SEAP	14.40	5.20	77.53	69.77	71.01	41.00	40.56	73.65	72.01	43.17	50.83
	DSPP	13.42	5.20	78.18	68.03	72.78	40.40	39.92	74.85	71.34	44.37	50.85
	DSPP+CFSP	13.65	5.40	78.67	68.75	72.87	41.40	39.94	74.63	72.26	42.43	51.10
50%	FLAP	3.11	3.38	65.83	57.46	44.50	33.00	23.18	48.09	40.19	27.22	34.60
	Wanda-SP	5.00	3.14	66.38	59.83	43.52	34.40	26.61	50.26	46.17	27.13	36.24
	CFSP	7.58	3.96	67.30	62.12	39.76	35.20	23.66	53.95	48.86	30.20	37.26
	SEAP	9.25	4.36	74.16	66.06	67.34	38.20	29.06	64.25	59.60	38.05	45.03
	DSPP	8.19	4.28	74.59	63.69	67.61	36.00	32.18	65.43	61.83	37.88	45.17
	DSPP+CFSP	10.31	4.72	74.92	64.96	69.72	36.60	32.69	66.27	62.21	39.51	46.19

Table 9: Zero-shot and few-shot accuracy ($\uparrow$) on LLaMA2-7B at 20% and 50% sparsity.

Sparsity	Method	GSM8k	MATH	PIQA	WinoGrande	BoolQ	OBQA	MMLU	HellaSwag	ARC-e	ARC-c	Average
0%	LLaMA2-13B	23.50	6.80	80.52	72.14	80.55	45.20	52.09	79.37	77.48	49.15	56.68
20%	FLAP	19.03	5.92	77.64	68.67	72.81	44.40	41.16	73.53	69.78	42.92	51.59
	Wanda-SP	21.53	6.20	78.40	72.61	75.90	43.00	46.28	75.76	73.82	47.27	54.08
	CFSP	22.67	6.58	78.24	71.43	77.89	43.80	48.78	75.76	73.36	47.35	54.59
	SEAP	21.83	6.44	79.82	69.69	76.15	42.20	43.50	76.67	72.77	46.50	53.56
	DSPP	21.99	6.94	80.03	71.11	77.28	44.60	49.79	78.26	75.84	49.06	55.49
	DSPP+CFSP	24.11	6.78	79.92	70.64	76.94	44.20	50.41	78.58	76.56	50.00	55.81
50%	FLAP	9.33	3.30	69.31	60.77	63.03	37.60	23.19	55.16	54.84	34.04	41.06
	Wanda-SP	13.57	3.16	69.48	62.51	62.97	37.40	23.04	54.41	54.97	34.90	41.64
	CFSP	17.51	4.98	71.82	64.25	64.83	39.20	29.47	60.24	62.16	38.23	45.27
	SEAP	13.80	4.72	76.06	64.96	72.42	38.40	30.91	65.87	65.74	40.02	47.29
	DSPP	18.12	5.86	76.66	65.51	71.44	38.20	41.70	70.73	66.25	40.27	49.47
	DSPP+CFSP	18.88	6.12	77.04	66.61	75.47	40.40	43.77	72.69	68.98	42.49	51.25

Table 10: Zero-shot and few-shot accuracy ($\uparrow$) on LLaMA2-13B at 20% and 50% sparsity.

Sparsity	Method	GSM8k	MATH	PIQA	WinoGrande	BoolQ	OBQA	MMLU	HellaSwag	ARC-e	ARC-c	Average
0%	LLaMA3-8B	50.80	13.26	80.85	73.40	80.98	44.80	62.14	79.17	77.74	53.24	61.64
20%	FLAP	25.85	7.48	73.83	64.96	72.54	39.20	40.16	60.98	59.81	35.41	48.02
	Wanda-SP	38.21	10.88	76.71	70.01	74.80	41.00	50.48	70.01	67.76	43.43	54.33
	CFSP	44.88	11.60	77.48	71.35	78.99	41.80	55.89	72.83	68.18	43.26	56.63
	SEAP	48.52	13.02	79.38	73.95	77.58	42.60	56.84	75.88	72.90	47.27	58.79
	DSPP	47.76	12.56	79.49	73.56	80.09	42.60	57.31	76.49	74.71	50.77	59.53
	DSPP+CFSP	48.82	12.42	79.22	73.80	81.47	43.00	58.76	76.83	74.54	51.96	60.08
50%	FLAP	4.47	2.66	60.39	56.91	62.29	30.80	23.02	41.56	38.22	25.60	34.59
	Wanda-SP	10.16	3.76	62.79	58.72	63.06	32.20	26.66	43.63	43.98	27.47	37.24
	CFSP	22.52	7.12	66.38	61.56	61.22	32.20	32.05	49.94	48.44	30.63	41.21
	SEAP	31.24	9.36	74.16	66.93	62.66	37.40	29.56	60.37	60.27	37.46	46.94
	DSPP	36.01	9.34	74.21	67.72	70.31	35.40	45.36	63.89	60.65	39.59	50.25
	DSPP+CFSP	38.51	9.96	75.19	67.64	72.87	36.80	47.32	66.04	61.62	40.96	51.69

Table 11: Zero-shot and few-shot accuracy ($\uparrow$) on LLaMA3-8B at 20% and 50% sparsity.

Sparsity	Method	GSM8k	MATH	PIQA	WinoGrande	BoolQ	OBQA	MMLU	HellaSwag	ARC-e	ARC-c	Average
0%	Qwen2.5-7B	83.93	24.54	79.98	72.69	84.80	47.40	71.81	78.92	77.40	51.19	67.27
20%	FLAP	68.76	16.56	74.48	65.35	80.70	43.40	60.79	68.85	72.18	46.67	59.78
	Wanda-SP	73.69	16.62	75.03	65.43	81.74	41.40	61.33	69.57	73.82	48.46	60.71
	CFSP	73.84	16.58	74.59	67.64	56.33	41.40	63.26	68.88	71.46	48.12	58.21
	SEAP	80.82	22.96	79.33	69.06	83.33	43.60	65.54	75.92	73.82	48.72	64.31
	DSPP	81.35	23.04	80.20	68.43	83.61	43.00	70.09	78.19	76.30	52.05	65.63
	DSPP+CFSP	81.96	22.86	80.36	68.98	82.84	43.00	70.04	78.23	76.85	51.54	65.67
50%	FLAP	35.56	1.08	60.61	54.22	58.87	31.80	27.03	43.45	48.15	30.46	39.12
	Wanda-SP	43.52	6.16	63.55	55.88	62.78	35.20	24.75	45.32	51.30	34.22	42.27
	CFSP	44.28	6.08	62.46	57.22	59.17	32.40	23.08	44.93	52.23	32.68	41.45
	SEAP	72.25	16.02	74.05	59.83	65.54	39.80	41.06	59.10	59.85	36.35	52.38
	DSPP	75.59	16.94	75.68	61.88	75.99	39.60	61.43	68.66	70.75	44.62	59.11
	DSPP+CFSP	75.66	16.84	75.46	61.09	75.66	40.20	61.07	67.81	71.51	43.86	58.92

Table 12: Zero-shot and few-shot accuracy ($\uparrow$) on Qwen2.5-7B at 20% and 50% sparsity.