GLAS: GRADIENT-GUIDED LAYER-WISE SPARSITY ALLOCATION FOR LLM PRUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) have achieved remarkable success across diverse tasks, yet their colossal size renders deployment expensive in both memory and inference. Pruning offers a direct remedy, but prevailing techniques apply a uniform sparsity ratio across Transformer layers, disregarding their varying importance and incurring severe performance degradation at high sparsity. Recent efforts therefore shift toward non-uniform layer-wise pruning. However, they rank layers by empirically designed proxy scores and overlook the dependence of layer importance on the pruning metric in use. To address this, we propose GLAS (Gradient-Guided Layer-wise Sparsity Allocation), which casts sparsity allocation as a constrained optimization problem. GLAS-1 allocates sparsity guided by metric-dependent gradients in a single pass, while GLAS-K further refines the allocation iteratively. Experiments across mainstream LLM families show that GLAS surpasses state-of-the-art baselines in both perplexity and zero-shot accuracy, and transfers to semi-structured and structured pruning as well as to vision transformers.

1 INTRODUCTION

Over the past few years, large language models (LLMs) have achieved remarkable success in natural language understanding, reasoning, and code generation (Brown et al., 2020; Touvron et al., 2023a; OpenAI, 2023). However, their large parameter counts impose substantial memory requirements and inference costs, which hinders practical implementation (Patterson et al., 2021; Luccioni et al., 2023). To address this, LLM pruning has emerged as a practical compression strategy to remove redundant or less important weights while preserving model performance (LeCun et al., 1989; Han et al., 2015; Jaiswal et al., 2023; Frantar & Alistarh, 2023; Sun et al., 2024; Zhang et al., 2024). Among these, SparseGPT formulates pruning as a layer-wise reconstruction problem, solving it in closed form and updating the surviving weights to compensate for those removed (Frantar & Alistarh, 2023). Wanda scores weights by magnitude scaled by the corresponding input activation norm (Sun et al., 2024), while RIA further evaluates each weight relative to its own row and column, mitigating channel corruption (Zhang et al., 2024).

Although the aforementioned methods achieve strong performance, they focus exclusively on which weights to remove within a layer, and assign the same sparsity ratio to every Transformer layer, thereby ignoring the varying importance of different layers. This limitation becomes pronounced under high sparsity, where Transformer layers differ substantially in redundancy. Some tolerate aggressive pruning with little effect on the output, while others collapse at the same ratio (Wang & Tu, 2020; Bhojanapalli et al., 2021). Recent efforts therefore allocate sparsity non-uniformly across layers, redistributing a fixed overall sparsity budget so that less important layers are pruned more aggressively and critical layers are preserved (Yin et al., 2024; Lu et al., 2024; Li et al., 2024; Gao et al., 2025; Huang et al., 2025; Sun et al., 2025; Chen et al., 2025; Yang et al., 2026). These approaches operate in two stages. An allocator first assigns a sparsity ratio to each layer, and a pruning metric such as Wanda, RIA, or SparseGPT then selects the weights to remove within that layer at the assigned ratio. OWL allocates sparsity according to the layerwise outlier distribution, assigning lower sparsity to layers with more weight and activation outliers (Yin et al., 2024). DLP instead estimates layer redundancy by the median of weight scores (Chen et al., 2025), and LSA searches for the allocation that minimizes the layer-wise linear reconstruction error (Yang et al., 2026). Despite their effectiveness, these methods remain limited in scope. OWL and DLP both

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.

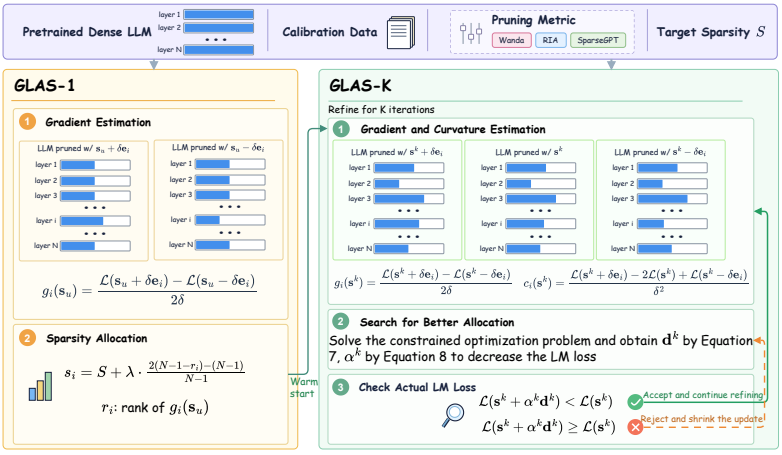

Figure 1: Overview of GLAS.

rely on proxy statistics to guide sparsity allocation, without detailed theoretical analysis (Yin et al., 2024; Chen et al., 2025). LSA optimizes the layer-wise linear reconstruction error in isolation, which diverges from the end-to-end language modeling loss (Yang et al., 2026). Moreover, prior work shows that layer sensitivity is metric-dependent, meaning that the relative importance of layers shifts with the applied pruning metric, yet most existing allocators determine the sparsity distribution without regard to the criterion subsequently applied (Gao et al., 2025).

Motivated by these observations, we propose **GLAS** (**G**radient-Guided **LA**yer-wise **S**parsity Allocation), a principled, metric-aware non-uniform pruning method that allocates sparsity across layers to preserve language modeling performance under a given pruning criterion. GLAS instantiates this idea in two algorithms. Specifically, GLAS-1 estimates, for each Transformer layer, how steeply the end-to-end language modeling loss rises with its sparsity ratio. A steep rise indicates that pruning the layer further is costly, and the budget is therefore redistributed toward layers whose loss rises more slowly. GLAS-K refines this allocation iteratively within a constrained optimization framework, re-estimating the gradients at each step and accepting only updates that reduce the language modeling loss. The pipeline of GLAS is illustrated in Figure 1.

We conduct extensive experiments across mainstream LLM families of varying scales and architectures. GLAS-1 consistently outperforms SOTA non-uniform pruning methods in both perplexity and zero-shot accuracy, and GLAS-K yields further gains. On the DeepSparse inference engine, GLAS achieves $2.5\times$-$3.3\times$ end-to-end acceleration on CPU with 70%-90% sparsity (Kurtic et al., 2023). Beyond unstructured sparsity, GLAS extends to semi-structured and structured pruning, generalizes to vision transformers, and remains effective after recovery fine-tuning.

Overall, our contributions are summarized as follows:

- We propose a novel non-uniform LLM pruning framework, GLAS, which allocates sparsity to directly preserve model performance. GLAS-1 determines the layer-wise sparsity according to estimated gradients of the language modeling loss with respect to each layer's sparsity ratio, and GLAS-K iteratively refines the allocation along a closed-form descent direction under the sparsity budget.
- Extensive experiments show that both GLAS-1 and GLAS-K consistently outperform SOTA baselines, while the pruned models attain up to 3.3x speedup on CPU.
- We further demonstrate the generality of GLAS, which extends to semi-structured and structured pruning, and transfers to vision transformers.

2 Related Work

2.1 LLM Pruning

As LLMs scale and advance rapidly, pruning has attracted considerable attention as a practical approach to reducing memory footprint and inference cost by removing redundant parameters from pretrained models. Magnitude pruning provides a simple baseline by removing the weights with the

smallest absolute values (Jaiswal et al., 2023). By contrast, SparseGPT frames pruning as a layer-wise sparse reconstruction problem and uses approximate second-order information to update the retained weights, compensating for the removed weights (Frantar & Alistarh, 2023). Wanda attains comparable performance without any weight update, scoring each weight by its magnitude scaled by the corresponding input activation norm (Sun et al., 2024). RIA further refines this criterion by evaluating each weight relative to its own row and column (Zhang et al., 2024). These methods determine which weights to remove within each weight matrix and apply a uniform sparsity ratio across layers. However, their performance degrades sharply at high sparsity ratios, where uniform allocation fails to account for the differing importance of layers (Yin et al., 2024).

2.2 Layer-wise Sparsity for Pruning

Non-uniform layer-wise allocation assigns each layer a sparsity according to its importance, and has proven effective at high sparsity (Yin et al., 2024; Lu et al., 2024; Li et al., 2024; Gao et al., 2025; Huang et al., 2025; Sun et al., 2025; Chen et al., 2025; Yang et al., 2026). OWL sets layer-wise sparsity in proportion to the outlier ratio of each layer, though its reliance on a predefined outlier threshold limits its adaptability across models and pruning settings (Yin et al., 2024). DLP, in turn, characterizes layer redundancy through the median of weight scores, but this criterion is empirically motivated and lacks theoretical justification (Chen et al., 2025). AlphaPruning adopts a more principled criterion, scoring layers by the heavy-tailed exponent of their weight spectra; however, this weight-only score does not incorporate input activation patterns (Lu et al., 2024). On the other hand, LSA departs from score-based schemes altogether and optimizes the allocation directly, but against the layer-wise linear reconstruction error, which diverges from the perplexity by which the pruned model is ultimately judged (Yang et al., 2026). Furthermore, layer-wise sensitivity has been shown to depend on the pruning metric used to select weights, a dependence none of these methods accounts for (Gao et al., 2025). To address these limitations, we propose a principled, gradient-guided algorithm that allocates sparsity by directly measuring how the language modeling loss responds to the sparsity of each layer under the pruning criterion in use.

3 Problem Formulation

The goal of the present work is to determine how to distribute a fixed sparsity budget across the Transformer layers of a pretrained LLM so that the pruned model retains as much language-modeling ability as possible.

Let $\mathcal{L}$ denote the language modeling loss, a primary criterion for assessing the quality of an LLM. We consider an LLM composed of N Transformer layers $\{l_1, l_2, \ldots, l_N\}$, each containing an attention module and a feed-forward network. Given a target sparsity S, each layer l_i is assigned a ratio $s_i \in [0, 1]$. A pruning metric $\mathcal{M}$ (e.g., SparseGPT, Wanda, or RIA) then scores the weights of l_i, and the lowest-scoring s_i fraction is removed from each of its weight matrices. The allocation is therefore fully specified by the vector $\mathbf{s} = (s_1, s_2, \ldots, s_N)$, and $\mathcal{L}(\mathbf{s}; \mathcal{M})$ denotes the loss of the model pruned according to $\mathbf{s}$ under $\mathcal{M}$. Layer-wise sparsity allocation can then be stated as the constrained optimization problem:

$$\min_{\mathbf{s}\in[0,1]^N} \mathcal{L}(\mathbf{s}; \mathcal{M}) \quad \text{s.t.} \quad \tfrac{1}{N}\textstyle\sum_{i=1}^{N} s_i = S, \tag{1}$$

where the constraint fixes the overall compression ratio, so that any reduction in $\mathcal{L}$ is obtained solely by redistributing sparsity across layers. Since all Transformer layers of the models we consider contain the same number of parameters, the average ratio $1/N \cdot \sum_{i=1}^{N} s_i$ equals the overall sparsity of the pruned layers. In general, the optimal allocation $\mathbf{s}^*(\mathcal{M})$ differs across pruning metrics. Uniform allocation, $\mathbf{s}_u = (S, \ldots, S)$, is a trivial feasible point, and most existing layer-wise allocation methods can be viewed as constructing feasible points through predefined mappings from layer-wise statistics to ratios, independently of $\mathcal{M}$ (Yin et al., 2024; Chen et al., 2025; Lu et al., 2024; Sun et al., 2025; Huang et al., 2025; Li et al., 2024; Yang et al., 2026).

4 Methodology

In this section, we introduce the proposed GLAS in three steps: the rationale for guiding allocation by the gradients of the loss with respect to $\mathbf{s}$, together with their estimation (Section 4.1); GLAS-1, a simple yet effective allocator (Section 4.2); and GLAS-K, which refines the allocation iteratively (Section 4.3). Throughout, the pruning metric $\mathcal{M}$ is fixed, and we write $\mathcal{L}(\mathbf{s})$ for $\mathcal{L}(\mathbf{s}; \mathcal{M})$ for brevity.

4.1 GRADIENT ESTIMATION

Gradient information has been consistently shown to be informative for pruning decisions in LLMs, where weights or layers with larger derivatives are retained as more important (Das et al., 2023; Huang et al., 2026). We bring this idea to layer-wise sparsity allocation, measuring how the language modeling loss responds to the sparsity ratio s_i rather than to the weights. Conceptually, the rate at which $\mathcal{L}(\mathbf{s})$ rises as layer l_i is pruned further indicates how costly its additional sparsity is. A layer with a steep rise should accordingly receive a lower ratio, with the freed budget redistributed to layers whose loss rises more slowly.

Nevertheless, $\mathcal{L}(\mathbf{s})$ admits no gradient through backpropagation, because each sparsity ratio s_i enters layer l_i through a discrete selection of the lowest-scoring weights under $\mathcal{M}$, rendering $\mathcal{L}(\mathbf{s})$ piecewise constant in s_i.[1] Its derivatives are zero within each constant piece and undefined at the jumps between pieces, and therefore carry no information about the allocation. We instead make the following assumption.

Assumption 1. *There exists a twice continuously differentiable function $\tilde{\mathcal{L}}$ such that $|\mathcal{L}-\tilde{\mathcal{L}}|$ is small relative to the change of $\tilde{\mathcal{L}}$ over sparsity increments spanning many steps.*

Intuitively, $\tilde{\mathcal{L}}$ captures the smooth trend of the loss, and $\mathcal{L}-\tilde{\mathcal{L}}$ is the small residual introduced by discrete weight selection. Evaluating $\mathcal{L}(\mathbf{s})$ is inexpensive, requiring only masking and a forward pass over a small calibration set $\mathcal{D}$, so we estimate the derivatives of $\tilde{\mathcal{L}}$ by central differences of $\mathcal{L}$ (Jordán, 1965):

$$g_i(\mathbf{s}) = \frac{\mathcal{L}(\mathbf{s}+\delta\mathbf{e}_i) - \mathcal{L}(\mathbf{s}-\delta\mathbf{e}_i)}{2\delta} \approx \frac{\partial\tilde{\mathcal{L}}(\mathbf{s})}{\partial s_i}, \tag{2}$$

$$c_i(\mathbf{s}) = \frac{\mathcal{L}(\mathbf{s}+\delta\mathbf{e}_i) - 2\mathcal{L}(\mathbf{s}) + \mathcal{L}(\mathbf{s}-\delta\mathbf{e}_i)}{\delta^2} \approx \frac{\partial^2\tilde{\mathcal{L}}(\mathbf{s})}{\partial s_i^2}, \tag{3}$$

where $\mathbf{e}_i$ is the i-th standard basis vector, δ denotes the perturbation size, and g_i and c_i are the estimated gradient and curvature of layer l_i. The perturbation must be large enough to rise above the residual, yet small enough for the local approximation to hold. Both g_i and c_i follow from the same three evaluations (i.e., $\mathcal{L}(\mathbf{s})$, $\mathcal{L}(\mathbf{s}+\delta\mathbf{e}_i)$, and $\mathcal{L}(\mathbf{s}-\delta\mathbf{e}_i)$), so obtaining $\{g_i(\mathbf{s}), c_i(\mathbf{s})\}_{i=1}^N$ costs $2N+1$ forward passes in total.

4.2 GLAS-1

GLAS-1 estimates the gradients at the uniform allocation $\mathbf{s}_u$ rather than at the dense model $\mathbf{s}=\mathbf{0}$, for two reasons. First, the loss is nearly flat at low sparsity, since only the least significant weights are removed, and the remaining dense layers retain sufficient capacity to compensate for the perturbed one; the resulting gradients are small and poorly separated across layers. Second, evaluating at $\mathbf{s}_u$ also measures the marginal cost of additional sparsity given that every other layer has already been pruned to S. The resulting $\{g_i(\mathbf{s}_u)\}_{i=1}^N$ indicate the cost of further sparsifying each layer, and thus induce a ranking over layers. As discussed in Section 4.1, layers with larger g_i should receive lower sparsity, while those with smaller g_i can be pruned more aggressively.

Existing non-uniform pruning techniques map layer scores to ratios through their magnitudes, typically by normalizing the scores to a fixed range and applying a linear transformation (Yin et al., 2024; Chen et al., 2025; Yang et al., 2026). This choice, however, makes the allocation sensitive to the dynamic range of the scores. For instance, a layer with an anomalously large score compresses all others toward one end of the range, and the allocation degenerates toward uniform (see Appendix B).

GLAS-1 therefore maps ranks rather than magnitudes. Let $r_i \in \{0,1,\ldots,N-1\}$ denote the rank of layer l_i when sorted by increasing $g_i(\mathbf{s}_u)$, and assign:

$$s_i = S + \lambda\Big(1 - \frac{2r_i}{N-1}\Big), \tag{4}$$

which satisfies $\frac{1}{N}\sum_{i=1}^N s_i = S$, and every ratio lies in $[S-\lambda, S+\lambda]$. λ controls the deviation from uniform allocation. A value that is too large drives the least important layers toward extreme sparsity

[1] $\mathcal{L}$ is piecewise constant in s_i, with steps of width $1/d_{\text{in}}$, where d_{in} is the number of input features of a weight matrix.

and damages them beyond recovery, whereas a value that is too small leaves the allocation close to uniform and forfeits the benefit of layer-wise allocation. Appendix B further shows that replacing OWL's magnitude mapping with ours also improves its performance. The complete pipeline is summarized in Algorithm 1 in Appendix C.

4.3 GLAS-K

GLAS-1 offers a simple and effective allocation, but its gradients $\{g_i(\mathbf{s}_u)\}_{i=1}^N$ are estimated at $\mathbf{s}_u$ and characterize the loss only locally. The mapping in Equation 4 may displace the allocation by as much as λ in every coordinate, at which point these gradients no longer describe the loss accurately. GLAS-K addresses this by refining the allocation iteratively. Starting from a warm start such as the output $\mathbf{s}$ of GLAS-1, each iteration re-estimates $\{g_i, c_i\}_{i=1}^N$ at the current allocation and adjusts the allocation to reduce $\mathcal{L}$ under the budget constraint.

Before describing the update, we characterize the allocations GLAS-K seeks. We do so through the optima of the surrogate $\tilde{\mathcal{L}}$. Let $\mathbf{s}^* \in (0,1)^N$ be a local minimizer of $\tilde{\mathcal{L}}$ under the budget constraint in Equation 1. Then all its partial derivatives coincide,

$$\partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_1 = \partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_2 = \cdots = \partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_N, \tag{5}$$

so that no layer is more costly to prune further than any other. Were this not the case, say $\partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_a > \partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_b$, transferring sparsity from l_a to l_b along $\mathbf{e}_b - \mathbf{e}_a$ would preserve the budget and change $\tilde{\mathcal{L}}$ by $-t\big(\partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_a - \partial\tilde{\mathcal{L}}(\mathbf{s}^*)/\partial s_b\big) + \mathcal{O}(t^2) < 0$ for small $t > 0$, contradicting the optimality of $\mathbf{s}^*$. Since the g_i estimate these derivatives, equal estimated gradients are the observable counterpart of this condition. GLAS-K follows this principle directly, shifting sparsity from layers with larger gradients toward those with smaller ones.

GLAS-K refines the initial allocation $\mathbf{s}^0$ over K iterations. At iteration $k \in \{0, 1, \ldots, K-1\}$, given the current allocation $\mathbf{s}^k$, we compute an update $\mathbf{d}^k = (d_1^k, d_2^k, \ldots, d_N^k)$ that preserves the pruning budget and is expected to lower the loss. To this end, we approximate the change of the surrogate by its second-order expansion around $\mathbf{s}^k$, keeping the diagonal of the Hessian (see Appendix D for justification) and replacing the derivatives with their estimates. This yields a separable quadratic model, $\tilde{\mathcal{L}}(\mathbf{s}^k + \mathbf{d}^k) - \tilde{\mathcal{L}}(\mathbf{s}^k) \approx \sum_{i=1}^N \big(g_i(\mathbf{s}^k)d_i^k + \frac{1}{2}c_i(\mathbf{s}^k)(d_i^k)^2\big)$. Minimizing it selects the update with the largest predicted loss reduction. Under the pruning budget, this yields the subproblem

$$\min_{\mathbf{d}^k} \; q(\mathbf{d}^k) = \textstyle\sum_{i=1}^N \big(g_i(\mathbf{s}^k)d_i^k + \frac{1}{2}H_i(\mathbf{s}^k)(d_i^k)^2\big) \quad \text{s.t.} \quad \sum_{i=1}^N d_i^k = 0, \tag{6}$$

where $H_i(\mathbf{s}^k) = \max(c_i(\mathbf{s}^k), \epsilon)$ for small $\epsilon > 0$, which makes the objective strictly convex (see Appendix D for more details). Introducing a Lagrange multiplier μ^k for the constraint, the stationarity conditions yield the closed-form solution

$$d_i^k = w_i^k\big(\mu^k - g_i(\mathbf{s}^k)\big) \quad \text{and} \quad \mu^k = \textstyle\sum_{j=1}^N w_j^k g_j(\mathbf{s}^k) \big/ \sum_{j=1}^N w_j^k, \tag{7}$$

where $w_i^k = 1/H_i(\mathbf{s}^k)$. Layers with $g_i(\mathbf{s}^k) > \mu^k$ receive less sparsity and those with $g_i(\mathbf{s}^k) < \mu^k$ receive more, with w_i^k scaling how far each moves. The update vanishes precisely when all gradients coincide. Since the quadratic model is accurate only locally, the update is scaled to $\alpha^k\mathbf{d}^k$ with

$$\alpha^k = \min\big(1,\, \rho/\|\mathbf{d}^k\|_\infty,\, \alpha_{\text{box}}^k\big), \tag{8}$$

where ρ is a trust-region radius and α_{box}^k is the largest scaling for which $\mathbf{s}^k + \alpha^k\mathbf{d}^k$ remains in $[0,1]^N$.

Proposition 1. *Consider any iteration of GLAS-K at current allocation $\mathbf{s}^k$, with the update $\mathbf{d}^k$ obtained from Equation 7 and the step size $\alpha^k > 0$ from Equation 8. Then the scaled update $\alpha^k\mathbf{d}^k$ preserves the pruning budget $\sum_{i=1}^N \alpha^k d_i^k = 0$, and is a descent direction with respect to the estimated gradient,*

$$\langle \mathbf{g}(\mathbf{s}^k), \alpha^k\mathbf{d}^k\rangle \le 0, \tag{9}$$

with equality if and only if $g_1(\mathbf{s}^k) = \cdots = g_N(\mathbf{s}^k)$.

The proof is given in Appendix E. Because $\mathbf{g}$ estimates $\nabla\tilde{\mathcal{L}}$, the update is also expected to decrease $\tilde{\mathcal{L}}$ when the estimation error is small relative to the spread of the gradients. Proposition 1 does not, however, guarantee a decrease of $\mathcal{L}$ itself, so each candidate allocation is verified directly. The update is accepted, $\mathbf{s}^{k+1} = \mathbf{s}^k + \alpha^k\mathbf{d}^k$, if $\mathcal{L}(\mathbf{s}^k + \alpha^k\mathbf{d}^k) < \mathcal{L}(\mathbf{s}^k)$. Otherwise it is rejected, $\mathbf{s}^{k+1} = \mathbf{s}^k$, the trust-region radius is halved, and the cached gradients are reused in the next iteration.

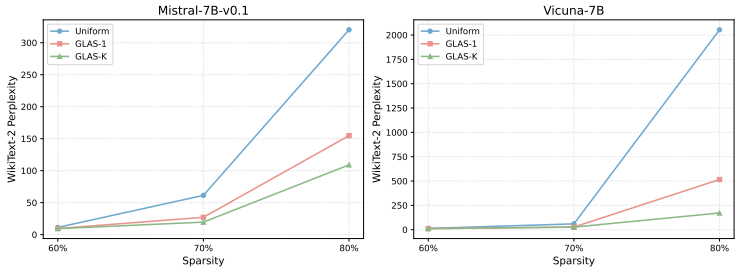

Figure 2: WikiText-2 perplexity (↓) of Mistral-7B-v0.1 and Vicuna-7B pruned with Wanda at increasing sparsity.

The loss of the iterates is therefore non-increasing by construction. Moreover, GLAS-K requires only a feasible starting allocation $\mathbf{s}^0$ and is not tied to GLAS-1, so the warm start may equally be obtained from uniform allocation or existing allocators such as OWL, DLP, and LSA. The full procedure is summarized in Algorithm 2 in Appendix F, and an illustration of the iterative process is given in Appendix P.

5 Experiments

In this section, we evaluate GLAS-1 and GLAS-K against SOTA sparsity allocation methods across multiple LLM families, model scales, and pruning metrics, measuring both language modeling proficiency and zero-shot capability.

5.1 Experimental Setup

Models and Datasets. We assess GLAS on a range of LLMs, including LLaMA2 (7B) (Touvron et al., 2023b), LLaMA3 (8B) (Team, 2024), LLaMA3.1 (8B) (Team, 2024), Qwen2.5 (7B/14B/32B) (Yang et al., 2024), Qwen3 (8B/32B) (Team, 2025), Mistral (7B) (Jiang et al., 2023), and Vicuna (7B) (Chiang et al., 2023). Following established practice in post-training LLM pruning, we evaluate both language modeling proficiency and zero-shot capabilities (Frantar & Alistarh, 2023; Sun et al., 2024). Specifically, we report perplexity on WikiText-2 (Merity et al., 2017), PTB (Marcus et al., 1994), and C4 (Raffel et al., 2020). For zero-shot evaluation, we measure accuracy on seven commonsense benchmarks from EleutherAI LM Harness (Gao et al., 2021), namely BoolQ (Clark et al., 2019), RTE (Wang et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2020), ARC Easy and Challenge (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018).

Baselines. We consider three LLM pruning metrics $\mathcal{M}$, SparseGPT (Frantar & Alistarh, 2023), Wanda (Sun et al., 2024), and RIA (Zhang et al., 2024), each of which prunes every Transformer layer to the same ratio by default. We then pair these metrics with layer-wise allocation methods: OWL (Yin et al., 2024), DLP (Chen et al., 2025), and LSA (Yang et al., 2026) alongside GLAS-1 and GLAS-K. Following prior work, we focus on the high-sparsity regime ($> 50\%$), where uniform allocation degrades sharply and the choice of allocation is consequential. Across all comparisons, the metric, calibration data, and target sparsity are held fixed, so that only the allocation differs.

Implementation Details. All experiments are conducted on a single NVIDIA A100 80GB GPU. Following OWL, we randomly sample 128 sequences of 2048 tokens from the C4 training split as the calibration data, which is disjoint from the C4 validation split used for perplexity evaluation (Yin et al., 2024; Raffel et al., 2020). Hyperparameter settings are reported in Appendix G.

5.2 Main Results

① GLAS-1 outperforms existing allocation methods across model families, scales, and pruning metrics, and GLAS-K improves further on every configuration.

GLAS-1. We report WikiText-2 perplexity at 70% unstructured sparsity (i.e., $S = 0.7$) in Table 1 and the corresponding zero-shot accuracy in Table 2. GLAS-1 outperforms existing layer-wise allocation techniques in nearly all configurations, reducing perplexity by up to 23.5% under Wanda,

Table 1: WikiText-2 perplexity at 70% unstructured sparsity (↓). Best result is indicated in **bold**.

Metric	Allocation	LLaMA3.1 8B	Qwen2.5 7B	Mistral 7B-v0.1	Qwen2.5 14B	Qwen2.5 32B	Qwen3 32B
Dense	—	6.24	6.85	5.25	5.29	5.02	7.90
Wanda	Uniform	109.63	75.96	61.37	145.87	22.16	15.21
	OWL	78.48	48.90	41.06	56.26	13.50	15.00
	DLP	84.70	51.80	29.65	39.55	21.12	42.92
	LSA	75.61	61.24	31.26	49.04	22.74	32.82
	GLAS-1 (Ours)	**67.87**	**40.10**	**27.07**	**35.60**	**11.44**	**11.47**
SparseGPT	Uniform	40.59	20.75	21.81	16.63	11.09	14.62
	OWL	33.20	**20.01**	18.65	16.82	10.82	14.19
	DLP	35.21	26.23	17.45	19.13	15.38	29.34
	LSA	35.08	28.66	18.32	20.18	17.02	29.13
	GLAS-1 (Ours)	**32.45**	20.37	**17.02**	**15.44**	**9.95**	**12.44**
RIA	Uniform	118.37	68.87	56.09	119.19	16.53	14.45
	OWL	73.58	46.00	35.78	41.83	13.45	13.74
	DLP	87.03	47.47	27.67	30.91	19.21	37.07
	LSA	77.96	56.88	29.79	33.08	20.76	29.95
	GLAS-1 (Ours)	**57.97**	**35.85**	**24.56**	**23.89**	**11.04**	**11.71**

Table 2: Zero-shot accuracy (%) of LLaMA3.1-8B at 70% unstructured sparsity (↑).

Metric	Allocation	ARC-c	BoolQ	OBQA	HellaSwag	RTE	WinoGrande	ARC-e	Avg.
Dense	—	53.41	82.54	45.00	78.88	69.31	73.48	81.14	69.11
Wanda	Uniform	20.31	58.07	27.20	28.99	52.71	48.70	31.36	38.19
	OWL	21.50	61.99	26.20	32.57	52.71	48.70	35.82	39.93
	DLP	23.21	62.02	27.80	34.02	52.71	52.96	37.92	41.52
	LSA	23.98	62.02	27.80	36.03	52.71	52.96	38.80	42.04
	GLAS-1 (Ours)	**24.15**	**62.08**	**28.20**	**36.24**	**52.71**	**53.99**	**39.35**	**42.39**
SparseGPT	Uniform	25.34	63.58	28.00	41.28	52.71	57.85	41.62	44.34
	OWL	28.24	70.37	29.20	46.47	**53.79**	61.80	46.55	48.06
	DLP	27.39	69.97	29.80	47.37	52.71	61.25	45.96	47.78
	LSA	27.82	68.10	29.60	48.29	53.07	61.64	47.05	47.94
	GLAS-1 (Ours)	**28.33**	**71.31**	**31.00**	**48.70**	53.07	**62.67**	**47.73**	**48.97**
RIA	Uniform	21.08	58.32	25.80	30.05	52.71	48.78	33.16	38.56
	OWL	24.06	61.96	26.80	35.27	52.71	51.70	39.86	41.77
	DLP	23.63	62.26	27.80	36.63	52.71	55.17	40.19	42.63
	LSA	25.17	62.05	28.40	38.55	52.71	56.59	41.50	43.57
	GLAS-1 (Ours)	**25.26**	**62.87**	**28.40**	**38.62**	**52.71**	**56.91**	**42.38**	**43.88**

22.7% under RIA, and 12.3% under SparseGPT relative to the strongest baseline in each configuration. The improvements hold across all three metrics (i.e., SparseGPT, Wanda, and RIA), indicating that the proposed gradient-guided non-uniform pruning method adapts to the pruning metric in use. On LLaMA3.1-8B, GLAS-1 further attains the highest zero-shot average under every metric. Beyond the absolute margins, GLAS-1 is notably more reliable. Several baselines fall behind uniform allocation on some models as shown in Table 1, whereas GLAS-1 improves upon it in every configuration we evaluate. Figure 2 shows that this advantage widens as the target sparsity increases. Results for magnitude pruning with layer-wise sparsity allocation are reported in Appendix H (Jaiswal et al., 2023), and PTB and C4 perplexities are reported in Appendix I. Extensions to $N:M$ semi-structured and structured pruning are presented in Appendix J. Comparisons with additional baselines are in Appendix K, and a projection-wise variant of the allocation is presented in Appendix L. Metric-dependence analysis is in Appendix M.

② GLAS-K directly improves any warm start, including uniform allocation and the outputs of OWL, DLP, and LSA, by 28% to 67%.

GLAS-K. Table 3 reports GLAS-K under Wanda and RIA, taking the GLAS-1 allocation as the warm start $\mathbf{s}^0$. Although GLAS-1 is already strong, GLAS-K improves upon it on every model

Table 3: Perplexity (↓) on WikiText-2 (Wiki), PTB and C4, and average zero-shot accuracy (ZS) (↑) across seven tasks (%), at 70% unstructured sparsity. Best result is indicated in **bold**.

		LLaMA2-7B				LLaMA3-8B				Qwen3-8B			
Metric	Allocation	Wiki	PTB	C4	ZS	Wiki	PTB	C4	ZS	Wiki	PTB	C4	ZS
Dense	—	5.47	24.09	7.53	58.32	6.14	9.91	9.86	68.46	9.72	15.44	15.82	69.51
Wanda	Uniform	74.50	323.19	81.62	38.43	118.23	148.75	163.01	37.86	61.42	86.53	90.45	43.38
	OWL	30.83	204.26	37.86	43.94	90.10	128.05	126.44	39.23	63.15	86.97	92.74	41.53
	DLP	22.75	137.00	27.24	46.03	104.57	164.36	125.75	40.52	63.16	92.25	74.52	44.36
	LSA	22.84	135.72	26.71	45.26	94.40	176.05	104.42	41.73	75.36	92.67	72.95	46.10
	GLAS-1 (Ours)	23.04	147.81	27.00	45.30	64.32	87.75	77.90	40.54	48.68	66.12	63.67	46.79
	GLAS-K (Ours)	**20.89**	**131.45**	**22.96**	**46.21**	**51.24**	**73.39**	**58.61**	**42.97**	**36.62**	**60.38**	**53.11**	**47.45**
RIA	Uniform	63.63	452.13	76.41	40.45	130.84	155.27	161.84	36.44	64.64	100.89	96.75	44.88
	OWL	29.06	241.61	36.76	45.34	87.88	118.61	105.92	40.26	72.93	102.54	105.20	42.41
	DLP	23.60	145.74	27.41	45.88	103.36	137.47	109.76	41.55	52.75	75.73	70.09	46.55
	LSA	23.76	146.65	26.72	45.76	93.75	137.35	99.08	42.55	57.60	96.68	65.57	48.36
	GLAS-1 (Ours)	21.60	146.97	25.20	45.61	61.93	76.77	72.46	42.24	42.87	66.86	63.49	48.26
	GLAS-K (Ours)	**20.26**	**143.19**	**21.67**	**46.73**	**50.47**	**68.55**	**58.01**	**44.18**	**41.38**	**59.83**	**54.24**	**49.27**

and metric, and obtains the best result across all evaluations, demonstrating its effectiveness. The baselines, by contrast, rank inconsistently across pruning metrics. On Qwen3-8B, OWL gives the lowest WikiText-2 perplexity among the baselines under Wanda, yet the highest under RIA. Both GLAS variants remain ahead in perplexity under either metric, which underscores the importance of metric-dependent allocation. As noted in Section 4.3, GLAS-K requires only a feasible allocation to begin from, and is therefore directly applicable to the output of any allocator. Table 4 applies it to uniform allocation and to the allocations produced by OWL, DLP and LSA on Mistral-7B-v0.1 under Wanda. Every warm start improves substantially, by between 28% and 67%. Initializing from GLAS-1 nevertheless gives the best result, confirming that the proposed GLAS-1 and GLAS-K are complementary. Notably, even starting from the naive uniform allocation, GLAS-K reduces perplexity from 61.37 to 20.53, surpassing every existing allocation method. Detailed zero-shot results are reported in Appendix N, results under SparseGPT are in Appendix O, and details of the iterative procedure are provided in Appendix P. The robustness of GLAS-K to randomness is examined in Appendix Q.

Table 4: GLAS-K applied to warm starts from existing layer-wise allocation methods. WikiText-2 perplexity of Mistral-7B-v0.1 with Wanda at 70% sparsity.

Warm start	Initial	+GLAS-K	Impr.
Uniform	61.37	20.53	66.5%
OWL	41.06	20.13	51.0%
DLP	29.65	20.75	30.0%
LSA	31.26	20.76	33.6%
GLAS-1	27.07	**19.54**	27.8%

Table 5: WikiText-2 perplexity of Qwen3-8B with Wanda at 70% unstructured sparsity under different numbers of calibration samples.

#Samples	32	64	128
OWL	65.54	63.21	63.15
DLP	73.65	66.77	63.16
LSA	77.56	73.66	75.36
GLAS-1	49.29	48.68	48.68
GLAS-K	**38.74**	**37.09**	**36.62**

Inference Speedup. To measure the inference acceleration of sparse LLMs, we deploy LLaMA2-7B-chat pruned with GLAS-1 under Wanda on the DeepSparse inference engine and measure end-to-end decode latency on 32 cores of an AMD EPYC 7742 CPU (Kurtic et al., 2023). As shown in Table 6, the speedup grows with sparsity, reaching 2.53× at 70% and 3.31× at 90%.

Table 6: End-to-end decode latency and throughput of LLaMA2-7B-chat on DeepSparse.

Sparsity	Dense	10%	20%	30%	40%	50%	60%	70%	80%	90%
Latency (ms)	688.63	694.08	671.51	657.12	625.46	421.08	355.83	272.49	236.05	207.94
Throughput (tokens/s)	1.45	1.44	1.49	1.52	1.60	2.37	2.81	3.67	4.24	4.81
Speedup	1.00×	0.99×	1.03×	1.05×	1.10×	1.64×	1.94×	2.53×	2.92×	3.31×

③ GLAS generalizes to vision transformers and composes with quantization and fine-tuning.

Vision Transformers. To evaluate the generality of our method beyond language models, we apply GLAS-1 to DeiT-Base (Touvron et al., 2021) and ViT-Base (Dosovitskiy et al., 2021) on

ImageNet-1K (Deng et al., 2009), replacing the language modeling loss with the classification cross-entropy. As shown in Table 7, GLAS-1 achieves the highest accuracy in all settings, with the gain widening as sparsity increases, showcasing that the gradient-guided allocation transfers effectively to vision transformers.

Table 7: Top-1 accuracy (%) of vision transformers pruned with Wanda on ImageNet-1K.

	DeiT-Base			ViT-Base		
Sparsity	50%	60%	70%	50%	60%	70%
Dense		81.09			84.57	
Uniform	75.57	67.55	44.15	79.56	67.22	16.71
OWL	74.75	65.17	38.47	79.61	65.82	13.42
DLP	72.79	63.14	35.06	77.60	64.43	20.17
GLAS-1	**77.06**	**70.10**	**50.28**	**79.65**	**68.76**	**25.53**

Sensitivity to the Number of Calibration Samples. We vary the number of calibration samples from 32 to 128. As shown in Table 5, GLAS-1 and GLAS-K achieve the two lowest perplexities at every calibration size. Notably, both variants with only 32 samples outperform every baseline using 128, indicating that the proposed techniques are data-efficient and remain effective under a reduced calibration budget. Sensitivity to hyperparameters is analyzed in Appendix R, and the pruning cost is reported in Appendix S.

5.3 Practical Applications

Quantization. Pruning and quantization are complementary forms of compression, and we examine whether the allocation remains effective when the pruned model is subsequently quantized. We apply GPTQ to LLaMA2-7B pruned with Wanda at 70% sparsity (Frantar et al., 2022). As shown in Table 8, GLAS-1 reduces perplexity relative to uniform allocation by 69% at 8 bits, and still by over 60% at 3 bits.

Fine-tuning. We further explore whether the performance of pruned models can be recovered through fine-tuning. We apply Low-Rank Adaptation (LoRA) to the pruned LLMs, training only a small set of low-rank matrices while the sparsity pattern remains frozen (Hu et al., 2022). As shown in Table 9, even under this aggressive pruning, the pruned models recover much of their capability, with perplexity decreasing from 23.04 to 15.85 for GLAS-1 and from 20.89 to 11.86 for GLAS-K.

Table 8: WikiText-2 perplexity (↓) of LLaMA2-7B pruned with Wanda at 70% unstructured sparsity and quantized by GPTQ.

	8-bit	4-bit	3-bit
Dense	5.48	5.65	6.56
Uniform	74.53	117.61	124.65
GLAS-1	**23.02**	**36.87**	**47.91**

Table 9: WikiText-2 perplexity of LLaMA2-7B pruned with Wanda at 70% sparsity.

Allocation	w/o FT	w/ FT
GLAS-1	23.04	15.85
GLAS-K	20.89	11.86

6 Conclusion

In this paper, we introduce GLAS, a principled, metric-dependent method for non-uniform LLM pruning that allocates sparsity according to the estimated gradients of the end-to-end language modeling loss with respect to each Transformer layer's sparsity ratio. GLAS-1 estimates these gradients and converts their ranking into an allocation in a single pass, while GLAS-K further refines the allocation by iteratively solving a constrained optimization problem. Extensive experiments show that both variants consistently outperform existing allocation methods under multiple pruning metrics, and that the approach extends to semi-structured and structured pruning as well as to vision transformers.

AI USE STATEMENT

(This section is **required** and does not count toward the page limit.)

In this work, we used generative AI tools for design or provide feedback on research methodology or experiments and assist with translation. We have not used generative AI tools forgenerate synthetic data sets, help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs, propose or refine hypotheses, implement methods, clean and reformat dataset, support qualitative and thematic data analysis, and interpret results are not applicable to this work. Additionally, we used generative AI tools for create or edit software code. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the author. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

See the ICLR 2027 AI Policy for Authors for more details. This statement should not be more than 1 page.

ETHICS STATEMENT

(This section is **recommended** and does not count toward the page limit.)

This work studies post-training pruning of publicly available pretrained models using standard public benchmarks, and does not involve human subjects, private data, or the release of new datasets.

REPRODUCIBILITY STATEMENT

(This section is **recommended** and does not count toward the page limit.)

We have made several efforts to ensure the reproducibility of this work. The problem formulation and both algorithms are described in Sections 3 and 4.2–4.3, with the complete procedures summarized in Algorithms 1 and 2. The assumptions underlying the diagonal approximation and the curvature floor, together with the derivation of the closed-form update, are given in Appendix D, and a full proof of Proposition 1 in Appendix E. The experimental setup, including models, evaluation datasets, baselines, and calibration data, is specified in Section 5, and all hyperparameter settings for both GLAS and the baselines are reported in Appendix G.

REFERENCES

Srinadh Bhojanapalli, Ayan Chakrabarti, Andreas Veit, Michal Lukasik, Himanshu Jain, Frederick Liu, Yin-Wen Chang, and Sanjiv Kumar. Leveraging redundancy in attention with reuse transformers. *CoRR*, abs/2110.06821, 2021. URL https://arxiv.org/abs/2110.06821.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html.

Yuli Chen, Bo Cheng, Jiale Han, Yingying Zhang, Yingting Li, and Shuhao Zhang. DLP: dynamic layerwise pruning in large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/chen25l.html.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 2924–2936. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1300. URL https://doi.org/10.18653/v1/n19-1300.

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. *CoRR*, abs/1803.05457, 2018. URL http://arxiv.org/abs/1803.05457.

Rocktim Jyoti Das, Liqun Ma, and Zhiqiang Shen. Beyond size: How gradients shape pruning decisions in large language models. *CoRR*, abs/2311.04902, 2023. doi: 10.48550/ARXIV.2311.04902. URL https://doi.org/10.48550/arXiv.2311.04902.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pp. 248–255. IEEE Computer Society, 2009. doi: 10.1109/CVPR.2009.5206848. URL https://doi.org/10.1109/CVPR.2009.5206848.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323, 2022. doi: 10.48550/ARXIV.2210.17323. URL https://doi.org/10.48550/arXiv.2210.17323.

Chang Gao, Kang Zhao, Runqi Wang, Jianfei Chen, and Liping Jing. Maximum redundancy pruning: A principle-driven layerwise sparsity allocation for llms. In Cathal Gurrin, Klaus Schoeffmann, Min Zhang, Luca Rossetto, Stevan Rudinac, Duc-Tien Dang-Nguyen, Wen-Huang Cheng, Phoebe Chen, and Jenny Benois-Pineau (eds.), *Proceedings of the 33rd ACM International Conference on Multimedia, MM 2025, Dublin, Ireland, October 27-31, 2025*, pp. 11062–11070. ACM, 2025. doi: 10.1145/3746027.3755683. URL https://doi.org/10.1145/3746027.3755683.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. *Zenodo*, 2021.

Song Han, Jeff Pool, John Tran, and William J. Dally. Learning both weights and connections for efficient neural network. In Corinna Cortes, Neil D. Lawrence, Daniel D. Lee, Masashi Sugiyama, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada*, pp. 1135–1143, 2015. URL https://proceedings.neurips.cc/paper/2015/hash/ae0eb3eed39d2bcef4622b2499a05fe6-Abstract.html.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wei Huang, Anda Cheng, and Yinggui Wang. Gradpruner: Gradient-guided layer pruning enabling efficient fine-tuning and inference for llms. *CoRR*, abs/2601.19503, 2026. doi: 10.48550/ARXIV.2601.19503. URL https://doi.org/10.48550/arXiv.2601.19503.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, and Rongrong Ji. Determining layer-wise sparsity for large language models through a theoretical perspective. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, volume 267 of *Proceedings of Machine Learning Research*. PMLR / OpenReview.net, 2025. URL https://proceedings.mlr.press/v267/huang25ax.html.

Ajay Jaiswal, Shiwei Liu, Tianlong Chen, and Zhangyang Wang. The emergence of essential sparsity in large pre-trained models: The weights that matter. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/7a69ab48efcbb0153e72d458fb091969-Abstract-Conference.html.

Yazdan Jamshidi and Alexey Shvets. Pals: Percentile-aware layerwise sparsity for llm pruning. *arXiv preprint arXiv:2607.07557*, 2026.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *CoRR*, abs/2310.06825, 2023. doi: 10.48550/ARXIV.2310.06825. URL https://doi.org/10.48550/arXiv.2310.06825.

Károly Jordán. *Calculus of finite differences*, volume 33. American Mathematical Soc., 1965.

Eldar Kurtic, Denis Kuznedelev, Elias Frantar, Michael Goin, and Dan Alistarh. Sparse fine-tuning for inference acceleration of large language models. *CoRR*, abs/2310.06927, 2023. doi: 10.48550/ARXIV.2310.06927. URL https://doi.org/10.48550/arXiv.2310.06927.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In David S. Touretzky (ed.), *Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]*, pp. 598–605. Morgan Kaufmann, 1989. URL http://papers.nips.cc/paper/250-optimal-brain-damage.

Lujun Li, Peijie Dong, Zhenheng Tang, Xiang Liu, Qiang Wang, Wenhan Luo, Wei Xue, Qifeng Liu, Xiaowen Chu, and Yike Guo. Discovering sparsity allocation for layer-wise pruning of large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/ff997469ac66cf893c4183efeb22212a-Abstract-Conference.html.

Haiquan Lu, Yefan Zhou, Shiwei Liu, Zhangyang Wang, Michael W. Mahoney, and Yaoqing Yang. Alphapruning: Using heavy-tailed self regularization theory for improved layer-wise pruning of large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/10fc83943b4540a9524af6fc67a23fef-Abstract-Conference.html.

Alexandra Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. Estimating the carbon footprint of bloom, a 176b parameter language model. *J. Mach. Learn. Res.*, 24:253:1–253:15, 2023. URL https://jmlr.org/papers/v24/23-0069.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/44956951349095f74492a5471128a7e0-Abstract-Conference.html.

Mitchell P. Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The penn treebank: Annotating predicate argument structure. In *Human Language Technology, Proceedings of a Workshop held at Plainsboro, New Jerey, USA, March 8-11, 1994*. Morgan Kaufmann, 1994. URL https://aclanthology.org/H94-1020/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=Byj72udxe.

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 Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pp. 2381–2391. Association for Computational Linguistics, 2018. doi: 10.18653/V1/D18-1260. URL https://doi.org/10.18653/v1/d18-1260.

Asit K. Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks. *CoRR*, abs/2104.08378, 2021. URL https://arxiv.org/abs/2104.08378.

Decebal Constantin Mocanu, Elena Mocanu, Peter Stone, Phuong H Nguyen, Madeleine Gibescu, and Antonio Liotta. Scalable training of artificial neural networks with adaptive sparse connectivity inspired by network science. *Nature communications*, 9(1):2383, 2018.

OpenAI. GPT-4 technical report. *CoRR*, abs/2303.08774, 2023. doi: 10.48550/ARXIV.2303.08774. URL https://doi.org/10.48550/arXiv.2303.08774.

David A. Patterson, Joseph Gonzalez, Quoc V. Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David R. So, Maud Texier, and Jeff Dean. Carbon emissions and large neural network training. *CoRR*, abs/2104.10350, 2021. URL https://arxiv.org/abs/2104.10350.

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 *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Chuan Sun, Han Yu, Lizhen Cui, and Xiaoxiao Li. Efficient shapley value-based non-uniform pruning of large language models. *CoRR*, abs/2505.01731, 2025. doi: 10.48550/ARXIV.2505.01731. URL https://doi.org/10.48550/arXiv.2505.01731.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=PxoFut3dWW.

Llama Team. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024. doi: 10.48550/ARXIV.2407.21783. URL https://doi.org/10.48550/arXiv.2407.21783.

Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 2021. URL http://proceedings.mlr.press/v139/touvron21a.html.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a. doi: 10.48550/ARXIV.2302.13971. URL https://doi.org/10.48550/arXiv.2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b. doi: 10.48550/ARXIV.2307.09288. URL https://doi.org/10.48550/arXiv.2307.09288.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In Tal Linzen, Grzegorz Chrupala, and Afra Alishahi (eds.), *Proceedings of the Workshop: Analyzing and Interpreting Neural Networks for NLP, BlackboxNLP@EMNLP 2018, Brussels, Belgium, November 1, 2018*, pp. 353–355. Association for Computational Linguistics, 2018. doi: 10.18653/V1/W18-5446. URL https://doi.org/10.18653/v1/w18-5446.

Wenxuan Wang and Zhaopeng Tu. Rethinking the value of transformer components. In Donia Scott, Núria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020*, pp. 6019–6029. International Committee on Computational Linguistics, 2020. doi: 10.18653/V1/2020.COLING-MAIN.529. URL https://doi.org/10.18653/v1/2020.coling-main.529.

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, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Zhiguo Yang, Changjian Deng, Qinke Chen, Zijing Zhou, and Jian Cheng. Lsa: Layer-wise sparsity allocation for large language model pruning based on minimal linear reconstruction error. In *The Fourteenth International Conference on Learning Representations*, 2026.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Kumar Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning llms to high sparsity. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57101–57115. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/yin24e.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In Anna Korhonen, David R. Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/V1/P19-1472. URL https://doi.org/10.18653/v1/p19-1472.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. Plug-and-play: An efficient post-training pruning method for large language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=Tr0lPx9woF.

A Comparison with PALS

Jamshidi & Shvets (2026) has recently reported that gradient-based allocation fails decisively, producing allocations worse than random on LLaMA2-7B. Since GLAS is also gradient-based, we clarify why this finding does not apply here. The two approaches differentiate different quantities. PALS aggregates $\partial\mathcal{L}/\partial w_{ij}$, the gradient with respect to individual weights. GLAS differentiates with respect to the sparsity ratio s_i, so g_i is not a summary of weight gradients but the rate at which the loss changes as the layer is pruned further.

B Score-to-Sparsity Mapping

Existing non-uniform pruning methods employ the magnitude mapping, which converts per-layer scores into sparsity ratios (Yin et al., 2024; Gao et al., 2025; Chen et al., 2025; Yang et al., 2026). This appendix describes the magnitude mapping, identifies a failure mode, and shows how the rank mapping adopted in GLAS-1 avoids it.

Let $\{v_i\}_{i=1}^N$ denote the layer scores produced by any sparsity allocator. As discussed in Section 4.2, magnitude mapping assigns

$$u_i = 2\lambda \cdot \frac{v_i - v_{\min}}{v_{\max} - v_{\min}}, \qquad s_i = S + \bar{u} - u_i, \tag{10}$$

where $\bar{u} = \frac{1}{N}\sum_{j=1}^{N} u_j$. When the scores contain a single outlier, every other layer's s_i is compressed into a narrow sub-interval. The allocation then degenerates toward uniform, forfeiting the benefit non-uniform allocation was intended to provide. The rank mapping avoids this because the assigned ratios depend only on the ordering of the scores, and are equispaced across $[S-\lambda, S+\lambda]$. Figure 3 compares both mappings on Qwen3-8B, where the largest per-layer score ($g_i(\mathbf{s}_u)$) obtained in GLAS-1 exceeds the second largest by a factor of 2.90. The magnitude mapping assigns nearly all layers a ratio close to S, while the rank mapping spans the full range $[S-\lambda, S+\lambda]$. We also evaluate how rank mapping works with OWL by retaining OWL's outlier score and replacing the magnitude mapping with ours. As shown in Table 10, the rank mapping improves OWL on both models. We also report results of GLAS-1 with magnitude and rank mappings in Table 11. Results of LSA and DLP are in Tables 12 and 13. The rank mapping improves every allocator on

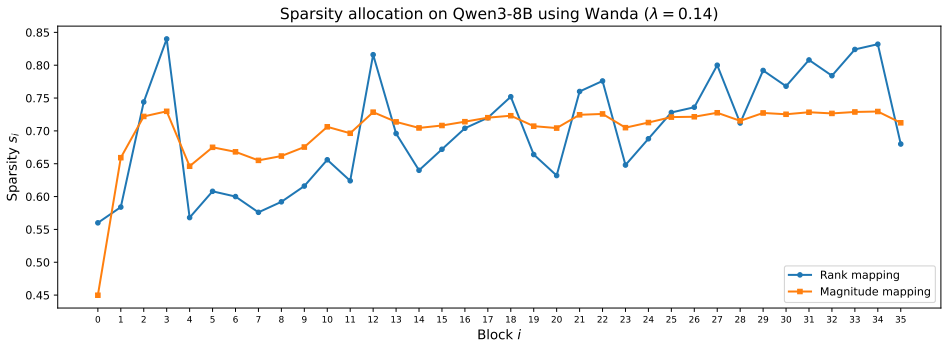

Figure 3: Sparsity allocations by rank mapping and magnitude mapping on Qwen3-8B under Wanda.

both models, suggesting that its benefit is not specific to GLAS. The comparison also separates the contributions of the score and the mapping. With the magnitude mapping, GLAS-1 already outperforms all baselines with their original mappings, and with the rank mapping applied to all methods, GLAS-1 remains the best. Both the gradient-based score and the rank mapping therefore contribute to its performance.

Table 10: OWL with Wanda at 70% sparsity (WikiText-2 perplexity ↓).

Mapping	LLaMA-3.1-8B	Mistral-7B-v0.1
Magnitude mapping	78.48	41.06
Rank mapping (ours)	**76.95**	**38.36**

Table 11: GLAS-1 with Wanda at 70% sparsity (WikiText-2 perplexity ↓).

Mapping	LLaMA-3.1-8B	Mistral-7B-v0.1
Magnitude mapping	69.37	29.51
Rank mapping (ours)	**67.87**	**27.07**

C Pseudocode of GLAS-1

Algorithm 1 summarizes the complete pipeline of GLAS-1 described in Section 4.2.

D Details of GLAS-K

This appendix supports the three choices made in Section 4.3: the diagonal approximation of the Hessian, the curvature floor $H_i = \max(c_i, \epsilon)$, and the closed-form solution of Equation 6.

First, when modeling the change of the surrogate $\tilde{\mathcal{L}}$ under an update $\mathbf{d}^k$, we retain only the diagonal of its Hessian, since the cross terms $\partial^2\tilde{\mathcal{L}}/\partial s_a \partial s_b$ for $a \neq b$ are small relative to the diagonal entries. To quantify this, we measure the normalized coupling

$$\kappa_{ab} = \frac{\left|\partial^2\tilde{\mathcal{L}}(\mathbf{s}_u)/\partial s_a \partial s_b\right|}{\sqrt{\left|\partial^2\tilde{\mathcal{L}}(\mathbf{s}_u)/\partial s_a^2\right| \left|\partial^2\tilde{\mathcal{L}}(\mathbf{s}_u)/\partial s_b^2\right|}}, \tag{11}$$

where the diagonal entries are estimated by Equation 3 and the mixed derivative by the four-point central difference

$$\frac{\partial^2\tilde{\mathcal{L}}(\mathbf{s}_u)}{\partial s_a \partial s_b} \approx \frac{\mathcal{L}(\mathbf{s}_u+\delta\mathbf{e}_a+\delta\mathbf{e}_b) - \mathcal{L}(\mathbf{s}_u+\delta\mathbf{e}_a-\delta\mathbf{e}_b) - \mathcal{L}(\mathbf{s}_u-\delta\mathbf{e}_a+\delta\mathbf{e}_b) + \mathcal{L}(\mathbf{s}_u-\delta\mathbf{e}_a-\delta\mathbf{e}_b)}{4\delta^2}. \tag{12}$$

Table 12: LSA with Wanda at 70% sparsity (WikiText-2 perplexity ↓).

Mapping	LLaMA-3.1-8B	Mistral-7B-v0.1
Magnitude mapping	75.61	31.26
Rank mapping (ours)	**70.73**	**30.74**

Table 13: DLP with Wanda at 70% sparsity (WikiText-2 perplexity ↓).

Mapping	LLaMA-3.1-8B	Mistral-7B-v0.1
Magnitude mapping	84.70	29.65
Rank mapping (ours)	**81.61**	**29.37**

Across sampled pairs, 95% satisfy $\kappa_{ab} < 0.2$ on LLaMA2-7B and 90% on Qwen2.5-7B. Each cross term is typically at least five times smaller than the diagonal entries it couples. The pairwise couplings between layers are therefore weak, and we retain only the diagonal. Any remaining error of the resulting model is caught by the acceptance test in Section 4.3, which verifies each update on $\mathcal{L}$ directly. Moreover, estimating the full Hessian would require $\mathcal{O}(N^2)$ loss evaluations per iteration instead of $\mathcal{O}(N)$.

Second, we substitute $H_i(\mathbf{s}^k) = \max(c_i(\mathbf{s}^k), \epsilon)$ for c_i, as negative curvatures occur on rare occasions. The clipping keeps $H_i > 0$, under which Equation 6 is a strictly convex problem and can be solved in closed form.

Third, the objective $q(\mathbf{d}^k)$ then has a positive definite Hessian and is strictly convex, and the constraint is affine. With a Lagrange multiplier μ^k, Equation 6 turns into

$$q(\mathbf{d}^k, \mu^k) = \sum_{i=1}^{N} \Big(g_i d_i^k + \tfrac{1}{2} H_i (d_i^k)^2 \Big) - \mu^k \sum_{i=1}^{N} d_i^k, \tag{13}$$

for which the KKT conditions are necessary and sufficient. Stationarity later gives Equation 7.

GLAS-K requires two safeguards when an allocation approaches the boundary of $[0, 1]^N$.

If $s_i < \delta$ or $s_i > 1 - \delta$, the central differences in Equations 2 and 3 would evaluate an infeasible ratio. In this case, we replace them with one-sided differences toward the interior. For $s_i < \delta$,

$$g_i(\mathbf{s}) = \frac{\mathcal{L}(\mathbf{s} + \delta \mathbf{e}_i) - \mathcal{L}(\mathbf{s})}{\delta}, \quad c_i(\mathbf{s}) = \frac{\mathcal{L}(\mathbf{s} + 2\delta \mathbf{e}_i) - 2\mathcal{L}(\mathbf{s} + \delta \mathbf{e}_i) + \mathcal{L}(\mathbf{s})}{\delta^2},$$

and symmetrically with $-\delta$ for $s_i > 1 - \delta$. If a layer lies at a bound and its update points outward, that is, $s_i^k = 0$ with $d_i^k < 0$ or $s_i^k = 1$ with $d_i^k > 0$, then $\alpha_{\text{box}}^k = 0$ and no progress is possible. We therefore fix $d_i^k = 0$ for such layers and recompute Equation 7 over the remaining layers.

E Proof of Proposition 1

Proof. By Equation 7, we have

$$\sum_{i=1}^{N} w_i^k g_i(\mathbf{s}^k) = \mu^k \sum_{i=1}^{N} w_i^k, \tag{14}$$

from which budget preservation follows directly,

$$\sum_{i=1}^{N} \alpha^k d_i^k = \alpha^k \sum_{i=1}^{N} w_i^k \big(\mu^k - g_i(\mathbf{s}^k)\big) = \alpha^k \Big(\mu^k \sum_{i=1}^{N} w_i^k - \sum_{i=1}^{N} w_i^k g_i(\mathbf{s}^k) \Big) = 0. \tag{15}$$

Algorithm 1 GLAS-1

Input: Pretrained LLM with N Transformer layers, pruning metric $\mathcal{M}$, target sparsity S, perturbation δ, spread λ, and calibration set $\mathcal{D}$
Output: Layer-wise allocation $\mathbf{s} = (s_1, \ldots, s_N)$

$\mathbf{s}_u \leftarrow (S, \ldots, S)$
for $i = 1$ to N **do**
  Evaluate $\mathcal{L}(\mathbf{s}_u + \delta \mathbf{e}_i)$ and $\mathcal{L}(\mathbf{s}_u - \delta \mathbf{e}_i)$ on $\mathcal{D}$ ▷ prune under $\mathcal{M}$
  Estimate $g_i(\mathbf{s}_u)$ according to Equation 2
end for
$\{r_i\}_{i=1}^N \leftarrow \text{Rank}(\{g_i\}_{i=1}^N)$
for $i = 1$ to N **do**
  Assign s_i by Equation 4
end for
return $\mathbf{s}$

For the descent property, substituting $d_i^k = w_i^k(\mu^k - g_i(\mathbf{s}^k))$ of Equation 7, and applying Equation 14

$$
\begin{aligned}
\big\langle \mathbf{g}(\mathbf{s}^k),\, \alpha^k \mathbf{d}^k \big\rangle &= \alpha^k \sum_{i=1}^{N} w_i^k\, g_i(\mathbf{s}^k)\big(\mu^k - g_i(\mathbf{s}^k)\big) \\
&= \alpha^k \Big((\mu^k)^2 \sum_{i=1}^{N} w_i^k \;-\; \sum_{i=1}^{N} w_i^k\, g_i(\mathbf{s}^k)^2 \Big) \\
&= -\alpha^k \Big(\sum_{i=1}^{N} w_i^k\, g_i(\mathbf{s}^k)^2 \;-\; 2(\mu^k)^2 \sum_{i=1}^{N} w_i^k \;+\; (\mu^k)^2 \sum_{i=1}^{N} w_i^k \Big) \\
&= -\alpha^k \Big(\sum_{i=1}^{N} w_i^k\, g_i(\mathbf{s}^k)^2 \;-\; 2\mu^k \sum_{i=1}^{N} w_i^k\, g_i(\mathbf{s}^k) \;+\; (\mu^k)^2 \sum_{i=1}^{N} w_i^k \Big) \\
&= -\alpha^k \sum_{i=1}^{N} w_i^k \big(g_i(\mathbf{s}^k) - \mu^k \big)^2 \le 0. \qquad (16)
\end{aligned}
$$

Since $w_i^k = 1/H_i(\mathbf{s}^k) > 0$ makes every term of the sum non-negative, equality holds if and only if $g_i(\mathbf{s}^k) = \mu^k$ for all i, that is $g_1(\mathbf{s}^k) = \cdots = g_N(\mathbf{s}^k)$. □

F Pseudocode of GLAS-K

Algorithm 2 summarizes the complete procedure of GLAS-K described in Section 4.3.

G Hyperparameter Setting

For unstructured pruning at 70% sparsity, each baseline uses the λ reported in its original work: 0.08 for OWL (Yin et al., 2024), 0.15 for DLP and LSA (Chen et al., 2025; Yang et al., 2026). GLAS-1 uses $\delta = 0.05$ and $\lambda = 0.14$. For GLAS-K, we set $K = 5$ and $\delta = 0.05$, initialize the trust-region radius to $\rho = 0.15$, and set the curvature floor ϵ to the 25-th percentile of the positive curvatures $\{c_i\}$ measured at the current allocation. In Figure 2, λ is set to 0.11, 0.14, and 0.10 at 60%, 70%, and 80% sparsity, respectively.

H Non-uniform Pruning Methods on Magnitude Pruning

Table 14 reports LLaMA2-7B pruned by magnitude at 70% unstructured sparsity (Jaiswal et al., 2023). Magnitude scores depend only on the weights and ignore activations, and at this sparsity the pruned model collapses under every allocation, with perplexity exceeding 8000 in all cases. GLAS-1 nonetheless obtains the lowest perplexity, improving on the strongest baseline by 5.7%.

Algorithm 2 GLAS-K

Input: Pretrained LLM with N layers, pruning metric $\mathcal{M}$, target sparsity S, warm start $\mathbf{s}^0$ (from GLAS-1 or any allocation method), perturbation δ, trust-region radius ρ, curvature floor ϵ, number of iterations K, calibration set $\mathcal{D}$
Output: Refined allocation $\mathbf{s}^K$

for $k = 0$ to $K - 1$ **do**
  if $k = 0$ **or** the previous update was accepted **then**
    Compute $\{g_i(\mathbf{s}^k), c_i(\mathbf{s}^k)\}_{i=1}^N$ using Equations 2 and 3 ▷ prune under $\mathcal{M}$, evaluate on $\mathcal{D}$
  end if
  Compute $\mathbf{d}^k$ by Equation 7
  Compute α^k by Equation 8
  Evaluate $\mathcal{L}(\mathbf{s}^k + \alpha^k \mathbf{d}^k)$ on $\mathcal{D}$
  if $\mathcal{L}(\mathbf{s}^k + \alpha^k \mathbf{d}^k) < \mathcal{L}(\mathbf{s}^k)$ **then**
    $\mathbf{s}^{k+1} \leftarrow \mathbf{s}^k + \alpha^k \mathbf{d}^k$ ▷ accept
  else
    $\mathbf{s}^{k+1} \leftarrow \mathbf{s}^k$, $\rho \leftarrow \rho/2$ ▷ reject, reuse cached gradients
  end if
end for
return $\mathbf{s}^K$

Table 14: WikiText-2 perplexity of LLaMA2-7B pruned by magnitude at 70% unstructured sparsity.

Allocation	WikiText-2 PPL
Dense	5.47
Uniform	49808.59
OWL	59561.54
DLP	8670.12
LSA	8795.53
GLAS-1	**8172.01**

I PTB AND C4 PERPLEXITY

Tables 15 and 16 report PTB and C4 perplexity under Wanda and RIA. GLAS-1 obtains the lowest perplexity in almost all configurations, confirming the effectiveness of the proposed method.

Table 15: PTB perplexity at 70% unstructured sparsity.

Metric	Allocation	LLaMA3.1-8B	Qwen2.5-7B	Mistral-7B-v0.1	Qwen2.5-14B	Qwen2.5-32B	Qwen3-32B
Dense	—	9.88	11.36	27.72	9.75	9.33	12.26
Wanda	Uniform	149.39	93.90	248.20	120.87	32.27	23.28
	OWL	132.28	70.29	196.52	65.50	23.94	22.94
	DLP	155.66	88.43	141.00	61.72	33.79	78.45
	LSA	156.35	100.16	141.69	69.56	40.59	57.93
	GLAS-1	**93.06**	**60.91**	**131.00**	**53.93**	**19.67**	**18.43**
RIA	Uniform	152.27	85.51	249.34	81.42	24.34	22.21
	OWL	121.23	68.11	191.45	64.31	21.26	21.57
	DLP	140.22	91.52	149.47	52.78	29.25	67.03
	LSA	123.60	102.85	147.39	54.02	31.56	49.54
	GLAS-1	**81.37**	**56.56**	**130.32**	**38.70**	**19.62**	**18.80**

J GLAS-1 UNDER SEMI-STRUCTURED AND STRUCTURED PRUNING

Semi-Structured Pruning. Under $N : M$ semi-structured pruning, at least N weights are set to zero within every group of M consecutive weights (Mishra et al., 2021). Standard practice applies a fixed N to every layer. We instead vary N across layers while keeping M fixed, rounding the GLAS-1 allocation to integer values of N so that the average sparsity matches the uniform target.

Table 16: C4 perplexity at 70% unstructured sparsity.

Metric	Allocation	LLaMA3.1-8B	Qwen2.5-7B	Mistral-7B-v0.1	Qwen2.5-14B	Qwen2.5-32B	Qwen3-32B
Dense	—	9.97	12.16	8.36	10.63	10.47	13.06
Wanda	Uniform	163.90	86.95	72.51	107.32	35.04	23.68
	OWL	111.06	69.41	50.76	67.58	22.50	23.31
	DLP	106.43	64.66	35.33	**48.44**	29.08	50.51
	LSA	93.99	72.84	35.06	48.95	29.32	38.44
	GLAS-1	**83.87**	**58.78**	**31.35**	52.29	**19.72**	**19.09**
RIA	Uniform	164.61	84.49	66.20	85.28	30.96	23.02
	OWL	98.20	68.57	44.02	65.10	23.35	21.95
	DLP	104.00	64.38	33.48	46.80	29.39	47.77
	LSA	86.21	74.20	34.33	45.66	29.33	36.86
	GLAS-1	**71.86**	**52.91**	**29.66**	**37.03**	**19.72**	**19.30**

As shown in Table 17, this improves on uniform $N:M$ at every level, and the gain increases as the pattern becomes more aggressive, from 4.0% at $4:8$ to 74.2% at $6:8$.

Table 17: WikiText-2 perplexity of LLaMA3.1-8B pruned with Wanda under $N:M$ sparsity.

Allocation	4:8	5:8	6:8
Uniform	13.74	84.57	3125.81
GLAS-1	**13.19**	**46.31**	**806.78**

Structured Pruning. Structured pruning removes entire components, such as attention heads and feed-forward channels, rather than individual weights, thereby reducing the physical dimensions of the model and supporting acceleration on standard hardware. We adopt LLM-Pruner (Ma et al., 2023) as the structured pruning method and replace its uniform per-layer ratio with the GLAS-1 allocation. As shown in Table 18, GLAS-1 achieves the lowest perplexity at every sparsity level on both evaluation datasets, indicating that the gradient-guided allocation extends naturally to structured pruning.

Table 18: Perplexity (↓) of LLaMA2-7B pruned by LLM-Pruner at different sparsity levels.

	20%		40%		60%	
Allocation	WikiText-2	PTB	WikiText-2	PTB	WikiText-2	PTB
Uniform	9.32	49.08	23.57	129.80	181.51	498.06
OWL	8.98	43.00	19.26	91.66	125.02	342.24
DLP	9.10	46.63	23.04	116.50	111.93	298.80
LSA	9.08	45.10	23.07	118.62	111.89	311.38
GLAS-1	**8.52**	**40.00**	**18.79**	**85.22**	**111.57**	**290.64**

K More Baselines

We further compare the proposed methods against ER (Mocanu et al., 2018), MRP (Gao et al., 2025), ATP (Huang et al., 2025), and AlphaPruning (Lu et al., 2024). As shown in Table 19, GLAS-K achieves the lowest perplexity on both models.

L Projection-wise Allocation

Following LSA, we extend GLAS-1 to allocate sparsity per projection matrix rather than per layer, with the budget weighted by parameter count since projections differ in size (Yang et al., 2026). As shown in Table 20, both granularities of GLAS-1 outperform both granularities of LSA. The finer granularity benefits LSA but not GLAS-1, which we attribute to the noisy estimation and

Table 19: WikiText-2 perplexity pruned with Wanda at 70% unstructured sparsity.

Allocation	Mistral-7B-v0.1	LLaMA2-7B
Uniform	61.37	74.50
ER	122.94	101.10
OWL	41.06	30.83
DLP	29.65	22.75
LSA	31.26	22.84
ATP	29.36	21.91
MRP	33.94	23.54
AlphaPruning	42.73	28.51
GLAS-1	27.07	23.04
GLAS-K	**19.54**	**20.89**

significant sparsity discrepancies between weight matrices, potentially disrupting information flow (Chen et al., 2025). Layer-wise allocation thus offers the better trade-off between resolution and estimation accuracy for our method, and we adopt it throughout.

Table 20: WikiText-2 perplexity of layer-wise and projection-wise allocation, pruned with Wanda at 70% unstructured sparsity.

Allocation	Qwen2.5-7B	Qwen3-8B
LSA (layer-wise)	61.24	75.36
LSA (projection-wise)	53.22	61.30
GLAS-1 (layer-wise)	**40.10**	**48.68**
GLAS-1 (projection-wise)	45.69	52.06

M Metric Dependence of the Allocation

GLAS-1 estimates its gradients under the same pruning metric $\mathcal{M}$ that prunes the final model, so the resulting allocation is specific to $\mathcal{M}$. To verify that this matters, we estimate the allocation under one metric and prune the model under another. Table 21 reports the results on LLaMA3.1-8B at 70% unstructured sparsity. For every pruning metric, the allocation estimated under the same metric yields the lowest perplexity, while a mismatched allocation increases perplexity by 1.3% to 24.8%.

Table 21: WikiText-2 perplexity (↓).

	Allocation estimated under		
Pruning metric	Wanda	RIA	SparseGPT
Wanda	**67.87**	69.63	70.45
RIA	58.70	**57.97**	72.33
SparseGPT	35.20	35.63	**32.45**

N Zero-shot Results of GLAS-K

Table 22 reports zero-shot accuracy for GLAS-K on three models under Wanda and RIA.

O SparseGPT with GLAS-K

Unlike Wanda and RIA, SparseGPT selects weights using $\mathrm{diag}(\mathbf{H}^{-1})$ and updates the retained weights after each removal, so re-running it at every perturbation would be prohibitively expensive (Frantar & Alistarh, 2023). We therefore compute $\mathrm{diag}(\mathbf{H}^{-1})$ once at $\mathbf{s}_u$ and reuse it to construct

Table 22: Zero-shot accuracy (%, ↑) at 70% unstructured sparsity.

Model	Metric	Allocation	ARC-c	BoolQ	OBQA	HellaSwag	RTE	WinoGrande	ARC-e	Avg.
LLaMA2-7B	Dense		40.61	70.12	40.80	72.93	63.18	67.01	53.58	58.32
	Wanda	Uniform	22.95	54.65	31.20	29.35	52.71	49.72	28.45	38.43
		OWL	26.71	62.20	32.60	36.42	58.48	54.70	36.49	43.94
		DLP	26.88	62.17	33.60	42.70	60.29	56.83	39.73	46.03
		LSA	27.39	62.20	32.80	43.89	53.79	57.62	39.14	45.26
		GLAS-K	27.65	62.39	33.80	44.61	57.40	58.17	39.44	46.21
	RIA	Uniform	23.72	61.77	30.80	29.54	53.43	52.01	31.86	40.45
		OWL	26.79	62.17	32.80	39.39	60.65	57.38	38.17	45.34
		DLP	27.90	62.17	33.80	44.13	55.23	58.25	39.65	45.88
		LSA	27.30	62.17	34.20	45.11	54.51	57.70	39.31	45.76
		GLAS-K	27.99	63.67	34.40	46.66	55.23	58.33	40.82	46.73
LLaMA3-8B	Dense		53.50	81.59	45.00	79.15	69.68	72.45	77.82	68.46
	Wanda	Uniform	21.08	56.88	25.80	29.02	52.71	48.15	31.40	37.86
		OWL	21.08	61.13	26.20	31.36	52.71	49.33	32.79	39.23
		DLP	23.04	61.04	27.00	32.95	52.71	51.30	35.61	40.52
		LSA	24.06	61.47	27.60	34.49	52.71	53.51	38.26	41.73
		GLAS-K	23.29	62.08	27.60	39.26	52.71	55.72	40.11	42.97
	RIA	Uniform	20.73	43.76	25.60	30.48	52.71	49.49	32.28	36.44
		OWL	22.35	58.75	25.80	33.84	52.71	51.07	37.33	40.26
		DLP	23.38	57.65	28.00	35.99	52.71	54.38	38.72	41.55
		LSA	24.32	60.18	28.40	38.09	52.71	54.46	39.69	42.55
		GLAS-K	25.17	62.20	29.20	41.73	52.71	56.83	41.41	44.18
Qwen3-8B	Dense		56.57	86.64	41.40	74.95	78.34	67.72	80.93	69.51
	Wanda	Uniform	24.49	62.29	28.00	35.96	55.96	53.20	43.77	43.38
		OWL	21.59	61.99	28.40	33.41	52.71	51.30	41.33	41.53
		DLP	25.51	62.26	28.00	37.76	57.04	56.20	43.73	44.36
		LSA	27.73	62.42	27.60	39.16	62.82	56.67	46.30	46.10
		GLAS-K	28.50	62.17	30.00	41.72	63.90	55.09	50.80	47.45
	RIA	Uniform	25.85	65.26	28.20	36.59	55.23	54.22	48.78	44.88
		OWL	23.12	61.62	28.00	33.53	52.71	52.64	45.24	42.41
		DLP	27.30	66.12	28.20	39.62	59.57	56.91	48.15	46.55
		LSA	28.41	64.62	29.20	41.26	66.79	57.30	50.97	48.36
		GLAS-K	30.38	65.66	30.60	42.53	64.62	57.93	53.20	49.27

the masks during estimation, omitting the weight update while probing. The final model is pruned by unmodified SparseGPT. Table 23 reports LLaMA3-8B pruned with SparseGPT at 70% sparsity. GLAS-K obtains the best result on all three corpora and the highest zero-shot average.

Table 23: LLaMA3-8B pruned with SparseGPT at 70% unstructured sparsity.

Allocation	WikiText-2	PTB	C4	Zero-shot avg.
Dense	6.14	9.91	9.86	68.46
Uniform	43.44	66.04	51.40	44.61
OWL	35.45	52.74	40.86	47.52
DLP	36.69	67.14	38.44	48.14
LSA	36.16	75.81	36.76	48.69
GLAS-1	31.44	50.75	38.60	48.66
GLAS-K	**31.13**	**50.30**	**36.42**	**49.05**

P Details of Iteration Procedure

Table 24 traces the iteration procedure of GLAS-K on Mistral-7B-v0.1 under Wanda. The evaluated $\mathcal{L}(\mathbf{s}^k + \alpha^k \mathbf{d}^k)$, which is the objective the algorithm evaluates, decreases monotonically by construction. WikiText-2 perplexity falls alongside it from 27.07 to 19.54, indicating that the improvement reflects a better allocation.

Table 24: GLAS-K iterates on Mistral-7B-v0.1 with Wanda at 70% sparsity.

Iteration k	0	1	2	3	4	5
Calibration loss $\mathcal{L}(\mathbf{s}^k + \alpha^k \mathbf{d}^k)$	3.161	3.093	2.996	2.976	2.922	**2.918**
WikiText-2 PPL under $\mathbf{s}^k$	27.07	25.34	23.84	21.54	20.17	**19.54**

Q RANDOMNESS OF GLAS-K

We repeat GLAS-K with three random seeds. As shown in Table 25, the variation across different random seeds is small.

Table 25: WikiText-2 perplexity of GLAS-K with different random seeds at 70% unstructured sparsity.

	Wanda		RIA	
Seed	Mistral-7B-v0.1	LLaMA3-8B	Mistral-7B-v0.1	LLaMA3-8B
0	19.54	51.24	20.12	50.47
1	20.53	50.40	19.90	49.78
42	20.32	49.35	20.10	49.65
Average	20.13	50.33	20.04	49.97

R HYPERPARAMETER SENSITIVITY

We vary the initial trust-region radius ρ and the percentile used to set the curvature floor ϵ, holding all other settings fixed. As shown in Table 26, perplexity changes slightly across the tested values, and every configuration remains well below the strongest baseline in Table 1, indicating that GLAS-K is robust to these choices.

Table 26: Sensitivity to the trust-region radius ρ and the percentile defining the curvature floor ϵ, on Mistral-7B-v0.1 pruned with Wanda at 70% unstructured sparsity (WikiText-2 PPL ↓).

ρ	0.15	0.10	0.05	Percentile	25	15	5
PPL	19.54	20.39	20.53	PPL	19.54	19.92	20.24

S COMPUTATIONAL COST

Table 27 reports the cost of sparsity allocation and of the subsequent pruning on LLaMA3-8B. Allocation is a one-time offline step. The cost of GLAS-1 scales linearly with the number of calibration samples. With 32 samples, GLAS-1 requires one third of the allocation time of LSA, while Table 5 shows that it still outperforms every baseline calibrated on 128 samples. The reported time for GLAS-K, measured with 32 calibration samples, covers only the refinement and excludes the GLAS-1 warm start. All methods subsequently prune in under one second, since they share the same pruning metric and differ only in the per-layer ratios.

Table 27: Sparsity allocation cost and subsequent pruning time on LLaMA3-8B with Wanda at 70% unstructured sparsity.

Method	Allocation time (s)	Peak memory (GB)	Pruning time (s)
Uniform	0.00	16.1	0.66
OWL	104.94	21.7	0.60
DLP	167.75	21.7	0.67
LSA	1199.75	23.2	0.64
GLAS-1 (32 samples)	392.38	32.7	0.61
GLAS-1 (64 samples)	765.83	32.7	0.61
GLAS-1 (128 samples)	1486.46	32.7	0.61
GLAS-K (32 samples)	1611.41	32.7	0.62