SoftIGM: A Soft Structured Pruning Method for LLMs via Intermediate Gradient Matching

Anonymous ACL submission

Abstract

Large Language Models (LLMs) impose substantial computational and memory costs, making structured pruning essential for deployment on resource-constrained devices. However, existing pruning methods often rely on single-step discrete importance criteria, which can misestimate structural importance and cause severe post-pruning performance collapse. To overcome this limitation, pruning should move beyond static forward heuristics and exploit backward gradients, which directly reflect how structural units respond to task loss during optimization. We therefore introduce **SoftIGM**, a soft structured pruning method driven by Intermediate Gradient Matching, which keeps prunable units in a continuous mask space and uses the previous-round model as a teacher to provide gradient-based backward feedback during multi-round pruning. By jointly optimizing masks and weights before final hard export, SoftIGM integrates structural search with pruning-time recovery without requiring a separate full post-pruning retraining stage. Experiments on LLaMA-2 show that SoftIGM consistently outperforms strong structured pruning baselines, achieving 52.49% average accuracy across seven downstream tasks at 60% sparsity, 4.11% higher than the best baseline.

1 Introduction

The advent of Large Language Models (LLMs) (OpenAI, 2023; Touvron et al., 2023) has catalyzed a paradigm shift in natural language processing, demonstrating unprecedented capabilities in reasoning, translation, and open-ended generation. However, the massive scale of model parameters introduces severe operational bottlenecks in real-world deployment. During autoregressive inference, immense memory footprints and high memory access costs result in significant latency and constrained throughput, hindering widespread deployment of LLMs in resource-constrained edge environments or high-concurrency cloud systems. To mitigate these substantial computational costs and address the critical need for parameter reduction in edge environments, various techniques have been proposed, including model pruning (Ma et al., 2023; Chen et al., 2023), quantization (Frantar et al., 2023; Xiao et al., 2023), and knowledge distillation (Saha et al., 2023). Among these paradigms, model pruning (Han et al., 2015; Wen et al., 2016; Anwar et al., 2017) has proven to be a highly effective strategy.

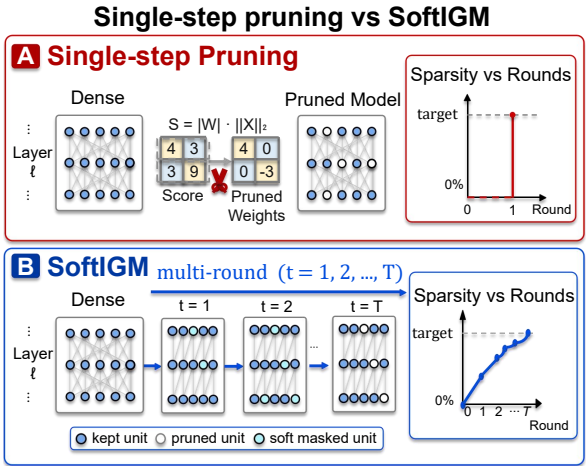

Figure 1: Conceptual comparison between conventional single-step hard pruning and the proposed **SoftIGM** paradigm. Unlike hard pruning, SoftIGM keeps prunable units in a soft search space and uses teacher-guided gradient feedback before final structured export.

Existing structural pruning techniques predominantly rely on discrete criteria to evaluate parameter importance. These approaches generally include magnitude- or activation-based heuristics (Sun et al., 2024; Zhang et al., 2024b), loss-based importance scoring (Ma et al., 2023; Zhang et al., 2024a), and reconstruction-based structural search (Frantar and Alistarh, 2023; Xu et al., 2024). As illustrated in Figure 1, such single-step or weakly corrected pruning makes sparsity jump directly to

the target level, so an early scoring error may permanently remove structural units that carry important signal and gradient pathways. Consequently, naive parameter removal can disrupt the learned optimization dynamics of the model and task-relevant structural pathways, triggering a severe collapse in post-pruning performance. To recover from this collapse, many existing methods depend on complex post-pruning fine-tuning or retraining procedures, effectively separating structural selection from model recovery. This raises a key challenge: how can we move beyond the conventional "discrete-pruning" paradigm and correct structural deviations through continuous backward feedback during pruning?

To address this challenge and realize this feedback-driven paradigm, we must fundamentally rethink how parameter importance is measured. Rather than relying only on discrete forward-pass heuristics, we focus on backward gradients as a more informative signal for structural importance. Unlike magnitude or activation, gradients encode how the model responds to task loss during optimization, providing a direct measure of structural sensitivity to error signals.

Building on this gradient-feedback perspective, we instantiate it through **Soft I**ntermediate **G**radient **M**atching (**SoftIGM**), a soft structured pruning method that corrects structural deviations during pruning optimization. Inspired by knowledge distillation (Hinton et al., 2014; Saha et al., 2023; Muralidharan et al., 2024), SoftIGM adopts a round-wise teacher-student paradigm, where the previous-round model guides the current soft-masked student toward a higher-sparsity structure. In each round, SoftIGM matches parameter-gradient signals between the teacher and student, using backward feedback to jointly update learnable masks and model weights while deferring hard structural removal until the final export. Figure 1 illustrates this distinction: conventional pruning commits to hard removal early, whereas SoftIGM keeps prunable units in a soft search space, repeatedly updating masks and weights before the final hard structured export. Comprehensive experiments on the LLaMA-2 architecture validate the effectiveness of our framework. Notably, under 60% sparsity, SoftIGM surpasses the leading Sheared LLaMA and LoRAPrune baselines, improving average accuracy by 4.11% over Sheared LLaMA and 6.85% over LoRAPrune.

The core contributions of this work are summarized as follows:

- We propose a gradient-based pruning criterion that uses parameter-gradient backward signals to guide structural mask optimization. By moving beyond discrete, single-step heuristics, this criterion provides a more direct measure of sensitivity to task loss.

- We design SoftIGM, a distillation-inspired, multi-round soft structured pruning method driven by intermediate gradient matching. By aligning the current soft-masked student with the previous-round teacher through continuous backward feedback, SoftIGM corrects structural deviations during pruning and preserves task-relevant structural pathways.

- We conduct extensive evaluations on language modeling, zero-shot reasoning, inference efficiency, and ablation analyses to comprehensively assess SoftIGM under aggressive structural sparsity. These experiments demonstrate its practical compression benefits while verifying the roles of gradient matching, pruning-time weight updates, and the multi-round soft optimization cycle.

2 Related Work

Structured pruning for LLMs can be grouped by the signals used to estimate the importance of weights or prunable structures. Based on these evaluation signals, existing structured pruning methods can be broadly categorized into the following paradigms.

2.1 Heuristic-based Pruning

Heuristic-based pruning estimates parameter or structural importance using inexpensive local signals, typically weight magnitudes and input activations. Magnitude-based pruning removes weights or structures with small absolute values, following the intuition that they contribute less to the output; early work by Han et al. (2015) applied this idea with retraining, while recent methods (Sengupta et al., 2025) use weight distributions for fast, calibration-free pruning. Since pure magnitude criteria ignore input-dependent effects, activation-aware methods incorporate calibration activations to refine importance estimation. For example, Wanda (Sun et al., 2024) combines weight magnitudes with input activation norms, while RIA (Zhang et al., 2024b) and DLP (Chen et al., 2025)

jointly consider weights and activations to adapt pruning rates under high sparsity.

2.2 Loss-based Pruning

To capture the actual role of prunable units more comprehensively, loss-based importance evaluation methods measure the sensitivity of the loss function to weight or structure variations. These methods operate on the premise that units whose removal induces smaller loss changes are deemed less critical. Molchanov et al. (2019) proposed estimating filter importance using Taylor expansions of the loss function to approximate the loss change induced by parameter removal. Similarly, LLM-Pruner (Ma et al., 2023) and LoRAPrune (Zhang et al., 2024a) rely on loss-based importance scoring combined with parameter-efficient fine-tuning techniques. While loss-based criteria reflect optimization patterns better than pure magnitudes, they typically rely on local sensitivity estimates and lack continuous feedback from a stable teacher model to revise early pruning decisions.

2.3 Optimization-based Pruning

Optimization-based pruning learns or identifies sparse structures by introducing additional optimization objectives, such as sparsity-inducing regularization or reconstruction loss. Regularization-based pruning (Wen et al., 2016) diverges from direct post-training truncation by introducing sparsity-inducing regularization during the training phase, driving a subset of weights or structures towards zero so they can be safely removed. Another line of work, reconstruction-based pruning, identifies redundancy by minimizing the model's output reconstruction loss. SparseGPT (Frantar and Alistarh, 2023), building upon the Optimal Brain Surgeon (OBS) (Hassibi and Stork, 1992) algorithm, utilizes an adaptive mask selection technique to overcome efficiency bottlenecks. Subsequent approaches like FastPruning (Kwon et al., 2022) and BESA (Xu et al., 2024) further refine this by learning sparsity masks through block-wise reconstruction error minimization. Despite advancing LLM compression, many of these paradigms still rely on local criteria or eventually commit to hard pruning decisions with limited corrective feedback. Such decisions may irreversibly discard important structures, frequently triggering a severe collapse in post-pruning performance. In contrast, our proposed soft pruning method provides continuous backward feedback, actively correcting structural deviations throughout the compression process rather than relying on discrete, single-step parameter removal.

3 Methodology

In this section, we formally detail the proposed **SoftIGM** method. Figure 2 provides the operational framework that instantiates the feedback-guided pruning mechanism introduced in Figure 1. We first establish the preliminaries of the structural pruning objective and the construction of the current soft-masked student. Then we formulate the intermediate gradient matching mechanism and describe the progressive sparsity-constrained optimization procedure.

3.1 Structural Pruning Objective

At pruning round r, let the soft-masked model state obtained at the end of round $r-1$ after weight updates serve as the teacher model, denoted as $\mathcal{M}_T^{(r-1)}$ and parameterized by $\mathbf{\Theta}^{(r-1)}$. This teacher represents a stable lower-sparsity structure without requiring physical hard pruning between rounds. The current-round student model, denoted as $\mathcal{M}_S^{(r)}$ and parameterized by $\boldsymbol{\theta}^{(r)}$, inherits the teacher parameters and applies learnable soft masks defined over structural units, forming a higher-sparsity candidate architecture. For the first pruning round, $\mathcal{M}_T^{(0)}$ is the original pre-trained dense model. Let $\boldsymbol{\mathcal{D}} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ represent the calibration dataset. The round-wise pruning objective can be written as

$$\min_{\mathbf{z}^{(r)}, \boldsymbol{\theta}^{(r)}} \quad \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\boldsymbol{\mathcal{D}}} \left[\mathcal{L}_{train}(\mathcal{M}_S^{(r)}(\mathbf{x};\boldsymbol{\theta}^{(r)}), \mathcal{M}_T^{(r-1)}, \mathbf{y})\right], \quad (1)$$

where $\mathcal{L}_{train}$ is the pruning-time objective optimized over the current student weights and the unit-level mask parameters in each round.

3.2 Soft-Masked Student Construction

Before defining the gradient matching objective, we specify how the current-round student is instantiated. Let $\mathcal{U}$ denote the set of prunable structural units, such as attention heads, FFN intermediate channels, hidden dimensions, and layers. For each unit $u \in \mathcal{U}$, SoftIGM maintains a learnable mask parameter z_u and generates a soft gate $m_u^{(r)} \in [0, 1]$ at each optimization step. The collection of gates $\mathbf{M}^{(r)} = \{m_u^{(r)}\}_{u\in\mathcal{U}}$ is therefore a unit-level mask, and defines the following masking operator:

$$\boldsymbol{\theta}^{(r)} = \Pi_{\mathbf{M}^{(r)}} \left(\mathbf{\Theta}^{(r-1)}\right). \quad (2)$$

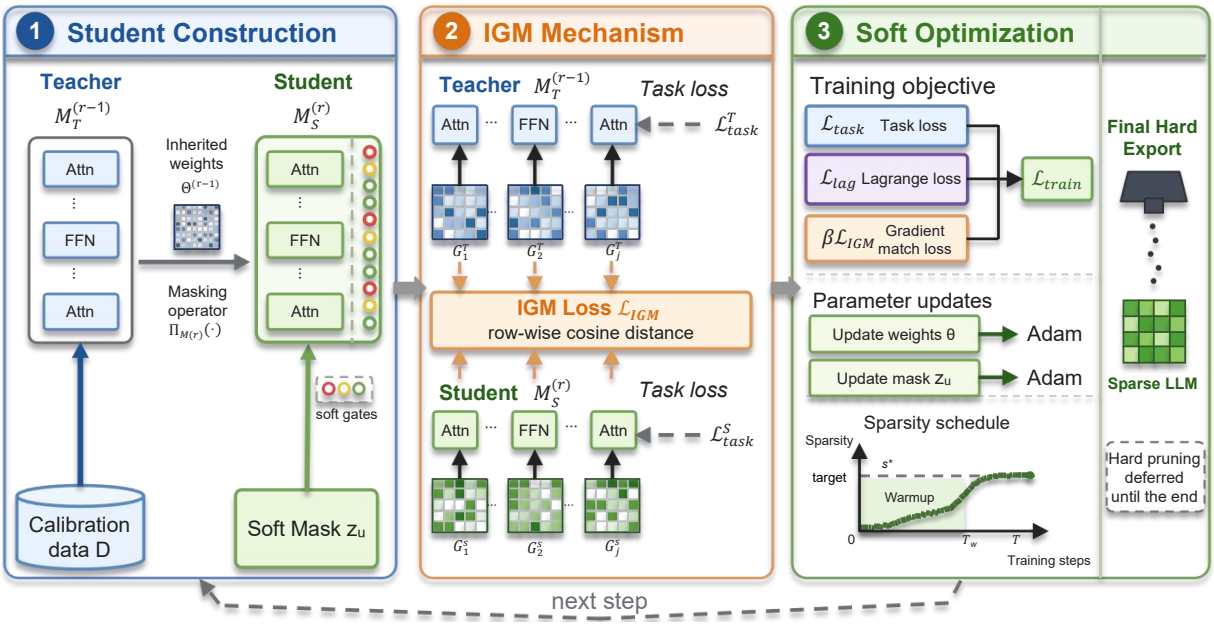

Figure 2: Overview of the proposed **SoftIGM** framework. In each pruning round, a soft-masked student inherits the previous-round teacher, jointly optimizes unit-level masks and model weights under task, sparsity, and gradient-matching objectives, and is finally exported as a hard structured sparse model.

Here $\Pi_{\mathbf{M}^{(r)}}(\cdot)$ broadcasts the soft gates to their corresponding structural groups. During pruning-time optimization, these gates are continuous and learnable rather than binary pruning decisions, so low-scored units remain in the search space and can still recover through later gradient updates. Thus, "structured pruning" refers to the final hard export stage: only after all soft-mask optimization rounds finish are the learned gate scores converted into discrete decisions that remove complete structural units.

3.3 Intermediate Gradient Matching Mechanism

Standard knowledge distillation seeks to minimize the discrepancy between the output logits of $\mathcal{M}_T^{(r-1)}$ and $\mathcal{M}_S^{(r)}$. However, for deep LLMs, output-level alignment alone may be insufficient to capture intricate internal information routing. To help ensure that the current-round student retains effective optimization pathways while moving toward a higher sparsity level, we propose aligning its backward learning signals with those of the previous-round teacher. In practice, we instantiate this alignment through parameter-gradient matching, which provides a tractable proxy for the gradient flow induced by intermediate transformer structures. Let $\mathcal{P}$ denote the set of corresponding gradient-bearing parameter tensors used for matching. For the j-th corresponding tensor, we compute the gradients of the task loss under the lower-sparsity teacher pathway and the current soft-masked student pathway, denoted as $\mathbf{G}_j^T = \nabla_{\mathbf{W}_j^T} \mathcal{L}_{task}^T$ and $\mathbf{G}_j^S = \nabla_{\mathbf{W}_j^S} \mathcal{L}_{task}^S$, respectively. The teacher gradients are used as reference signals for the current-round student.

To emphasize local structural similarity, we reshape each gradient tensor along its leading dimension and compare the corresponding gradient rows. For common projection and feed-forward tensors, this leading dimension often corresponds to output channels or structure-related directions, so row-wise comparison provides a practical approximation of structure-level gradient alignment. We define the Intermediate Gradient Matching Loss, $\mathcal{L}_{IGM}$, as the average row-wise cosine distance across matched parameter tensors:

$$\mathcal{L}_{IGM}(\boldsymbol{\theta}^{(r)}) = \frac{1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} \frac{1}{n_j} \sum_{i=1}^{n_j} \left(1 - \frac{\mathbf{g}_{j,i}^T \cdot \mathbf{g}_{j,i}^S}{\left(\|\mathbf{g}_{j,i}^T\|_2 \|\mathbf{g}_{j,i}^S\|_2 + \varepsilon \right)} \right), \quad (3)$$

where n_j is the size of the leading dimension of the j-th gradient tensor, and $\mathbf{g}_{j,i}^T$ and $\mathbf{g}_{j,i}^S$ denote the i-th reshaped gradient rows from the teacher and current soft-masked student pathways, respectively. The scalar term ε is a minimal constant (e.g., 10^{-20}) incorporated to ensure numerical stability. By minimizing this cosine distance-based $\mathcal{L}_{IGM}$, SoftIGM encourages the current-round student to preserve gradient directions that are consistent with the previous-round teacher, thereby maintaining

reliable optimization signals under structural compression.

3.4 Progressive Sparsity-Constrained Optimization

SoftIGM reaches the target architecture by jointly optimizing the current student weights and the learnable mask parameters under a progressively tightened sparsity constraint. Let T_w denote the sparsity warmup length and $s^\star$ the final target sparsity. At pruning step t, the target sparsity is scheduled as

$$s_t = s_0 + \min\left(1, \frac{t}{T_w}\right)(s^\star - s_0), \tag{4}$$

where s_0 is the initial sparsity. The Lagrangian penalty aligns the expected sparsity $\bar{s}$ induced by the current soft gates with the scheduled target:

$$\mathcal{L}_{lag} = \lambda_1(\bar{s} - s_t) + \lambda_2(\bar{s} - s_t)^2. \tag{5}$$

The effective pruning-time training objective is

$$\mathcal{L}_{train} = \mathcal{L}_{task} + \mathcal{L}_{lag} + \beta\mathcal{L}_{IGM}. \tag{6}$$

Here $\mathcal{L}_{task}$ preserves the task performance of the current soft-masked student, $\mathcal{L}_{lag}$ drives the soft gates toward the scheduled sparsity target, and $\mathcal{L}_{IGM}$ provides previous-round teacher-guided gradient feedback. In each optimization step, SoftIGM first instantiates the current soft gates from the learnable mask parameters, computes $\mathcal{L}_{train}$ on the resulting student, and then back-propagates the total objective to update both the student weights and the mask parameters:

$$\begin{aligned} \boldsymbol{\theta}_{t+1}^{(r)} &= \boldsymbol{\theta}_t^{(r)} - \eta_\theta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{train}\left(\boldsymbol{\theta}_t^{(r)}, \mathbf{z}_t^{(r)}\right), \\ \mathbf{z}_{t+1}^{(r)} &= \mathbf{z}_t^{(r)} - \eta_z \nabla_{\mathbf{z}} \mathcal{L}_{train}\left(\boldsymbol{\theta}_t^{(r)}, \mathbf{z}_t^{(r)}\right). \end{aligned} \tag{7}$$

Thus, mask learning and weight adaptation are not separated into two stages; they are coupled through the same pruning-time objective. After round r finishes, the optimized soft-masked student becomes the teacher for round $r+1$. Across all intermediate rounds, no units are physically removed. Only after the final soft optimization round does SoftIGM convert the learned gate scores into hard structured pruning decisions and remove the selected heads, layers, hidden dimensions, and FFN channels to obtain the deployable model.

4 Experiments

In this section, we empirically validate the proposed **SoftIGM** method. Our primary objective is to demonstrate its superiority in preserving language modeling and reasoning capabilities at extreme sparsity levels when compared against established compression baselines.

Before presenting the experimental protocol, we clarify that SoftIGM does not rely on a separate full post-pruning retraining stage. As defined in Equation 6, pruning-time optimization already updates the current student weights jointly with the unit-level mask parameters under the task, sparsity, and teacher-guided gradient-matching objectives. Thus, model recovery is integrated into the soft pruning process while the sparse structure is still being formed. Appendix A provides a first-order analysis showing that the resulting gradient-matched update remains close to a fine-tuning update on the current soft-gated sub-network, with a bounded correction from the previous-round lower-sparsity teacher.

4.1 Experimental Settings

Evaluation and Datasets: We evaluate SoftIGM using the widely adopted LLaMA-2-7B architecture to ensure comparability with prior work. To comprehensively assess the compressed model, we conduct evaluations across a diverse suite of established benchmarks. For foundational language modeling capabilities, we measure perplexity (PPL) on WikiText2 (Merity et al., 2017) and the Penn Treebank (PTB) (Marcus et al., 1993), evaluated using 512-token segments. For zero-shot reasoning and knowledge retrieval, we report accuracy metrics across seven diverse datasets: SciQ (Welbl et al., 2017), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), ARC-easy (ARC-e) (Clark et al., 2018), ARC-challenge (ARC-c) (Clark et al., 2018), and OpenBookQA (OBQA) (Mihaylov et al., 2018). Additionally, to probe the behavior of the pruned models in more open-ended settings, we reserve a subset of complex, long-context, and multi-round dialogue tasks for qualitative analysis and latency profiling. **Implementation Details:** During pruning optimization, we use RedPajama training batches (Together Computer, 2023) with a maximum sequence length of 4096. The unit-level soft structural masks and model weights are jointly optimized for 3200 batches with AdamW before the final hard structured export; these updates are part

Table 1: Zero-shot performance of the compressed LLaMA-2-7B. We evaluate WikiText2 and PTB on perplexity with 512-token segments. The average accuracy is calculated among seven classification datasets. **Bold** denotes the best performance at the same compression rate. * denotes the results obtained by our reproduction.

Ratio	Method	WikiText2 ↓	PTB ↓	SciQ	PIQA	HellaS	WinoG	ARC-e	ARC-c	OBQA	Average ↑	Avg/Dense ↑
0%	LLaMA-2-7B (Touvron et al., 2023)	7.18	27.13	94.00	78.07	57.19	68.90	76.35	43.34	31.40	64.18	100.00%
60%	Wanda* (Sun et al., 2024)	17367.05	17743.85	15.70	49.51	26.15	49.57	25.08	22.70	27.80	30.93	48.19%
	LoRAPrune (Zhang et al., 2024a)	38.64	140.51	75.60	62.08	38.76	50.04	41.33	21.84	29.80	45.64	71.11%
	Sheared LLaMA (Xia et al., 2024)	16.81	58.58	82.80	63.44	40.01	52.49	45.54	24.15	30.20	48.38	75.38%
	SlimLLM (Guo et al., 2025)	113.34	326.71	54.30	58.60	32.04	50.83	32.11	24.06	25.40	39.62	61.73%
	Ours (SoftIGM)	**14.16**	**44.74**	**85.60**	**68.28**	**47.63**	**54.78**	**53.24**	**25.68**	**32.20**	**52.49**	**81.79%**
80%	Wanda* (Sun et al., 2024)	20691.95	23092.35	13.60	49.24	25.49	49.96	24.79	24.66	**29.80**	31.08	48.43%
	LoRAPrune (Zhang et al., 2024a)	155.54	361.63	59.40	54.35	28.43	49.96	28.66	23.38	24.60	38.40	59.83%
	Sheared LLaMA (Xia et al., 2024)	26.92	90.37	74.80	58.87	30.40	**50.83**	35.06	21.42	25.80	42.45	66.14%
	SlimLLM (Guo et al., 2025)	2923.30	2672.11	22.20	52.56	25.89	47.67	26.64	**26.54**	21.20	31.81	49.56%
	Ours (SoftIGM)	**24.18**	**69.56**	**77.40**	**60.12**	**32.29**	48.62	**39.69**	22.78	26.60	**43.93**	**68.45%**

of pruning-time optimization rather than a separate post-pruning retraining stage. We use a global batch size of 32 and a microbatch size of 4. The main model learning rate is 1×10^{-4} with a cosine schedule and a 640-batch warmup, while the learning rate for the mask and Lagrange multiplier parameters is set to 1.0. The IGM loss weight in Equation 6 is set to $\beta = 2$. The sparsity target is linearly warmed up over the first 640 batches before remaining fixed at the final target architecture. All structural masking operations and gradient discrepancy computations are implemented in PyTorch, with experiments conducted on a server equipped with 2× NVIDIA A100 (80GB) GPUs.

Rationale for Extreme Sparsity Levels: We benchmark SoftIGM at 60% and 80% sparsity to evaluate its robustness under deployment-oriented compression regimes. LLaMA-2-7B requires about 14 GB of FP16 weight memory before KV cache and runtime overheads (Touvron et al., 2023; Meta, 2023), while consumer GPUs commonly provide 8 GB or less VRAM (Valve Corporation, 2026). Therefore, moderate sparsity levels such as 20%–40% are insufficient to demonstrate edge viability; 60% and 80% sparsity instead retain only 40% and 20% of parameters, making them stronger tests under practical memory constraints.

Baselines: We benchmark SoftIGM against four representative structural pruning baselines: **Wanda (Sun et al., 2024)**, a training-free discrete pruning baseline that evaluates weight importance via the product of weight magnitudes and input activation norms; **LoRAPrune (Zhang et al., 2024a)**, a pruning method that utilizes loss-based importance scoring integrated with Low-Rank Adaptation (LoRA) to facilitate performance recovery; **SlimLLM (Guo et al., 2025)**, a holistic structured pruning technique employing Pearson similarity for attention heads and PCA for feed-forward channels; and **Sheared LLaMA (Xia et al., 2024)**, a targeted structured pruning baseline. For Sheared LLaMA, we follow the pruning configuration from the original paper and run only its pruning stage for comparison.

4.2 Zero-shot performance

Table 1 reports the main perplexity and zero-shot results of SoftIGM and established baselines on the LLaMA-2-7B architecture under 60% and 80% sparsity.

As observed in Table 1, SoftIGM achieves the best average zero-shot performance and the lowest perplexity under both sparsity levels. Under the 60% sparsity constraint, SoftIGM obtains an average accuracy of 52.49%, surpassing the strongest baseline, Sheared LLaMA, by an absolute margin of 4.11%, corresponding to an 8.50% relative improvement over the 48.38% baseline average. It also achieves the lowest WikiText2 and PTB perplexities, indicating that language modeling quality is better preserved after structural compression. At the more aggressive 80% sparsity level, training-free or weakly corrected pruning baselines show severe degradation, especially in perplexity, whereas SoftIGM maintains an average accuracy of 43.93%, improving over the strongest baseline by 1.48% and yielding a 3.49% relative gain. By preserving task-relevant structural pathways through intermediate gradient matching, SoftIGM demonstrates strong robustness even under aggressive structural compression.

Internal Gradient Analysis. To complement the zero-shot results, we examine hidden-state gradient saliency on HellaSwag, computed from the hidden-dimension-normalized gradient norm of the loss with respect to intermediate hidden states. Figure 3

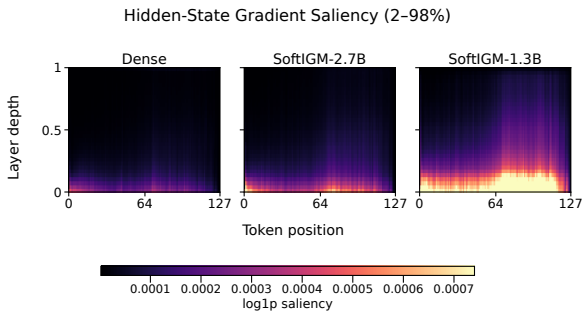

Figure 3: Hidden-state gradient saliency maps on HellaSwag for the dense model and SoftIGM-pruned models.

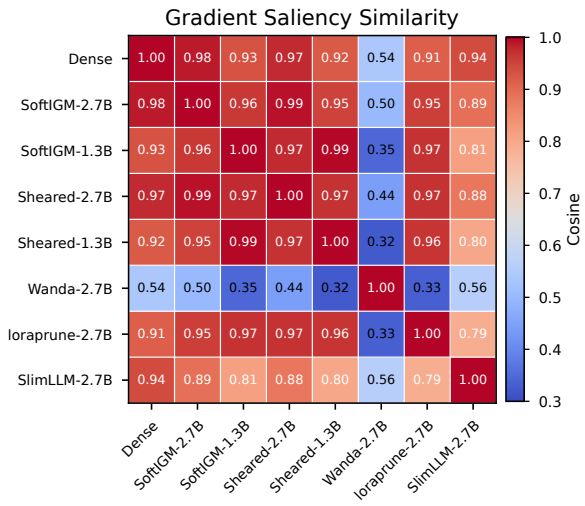

Figure 4: Pairwise cosine similarity of hidden-state gradient saliency maps across dense and compressed models on HellaSwag.

shows log-scaled saliency over normalized layer depth and token position, where brighter regions indicate stronger loss sensitivity rather than better task performance. The pairwise cosine similarities in Figure 4 further compare the resulting saliency maps across dense and compressed models; higher similarity indicates a closer internal loss-sensitivity pattern. Overall, these analyses suggest that SoftIGM better preserves the dense model's internal gradient-response pattern after structural pruning. We provide a layer-wise summary in Appendix B.2.

4.3 Inference Efficiency

Table 2 illustrates the parameter scale, model-weight memory footprint, and inference latency of the pruned LLaMA-2-7B models. The prefill latency is measured as time-to-first-token using an input sequence length of 256. To test decoding latency, we set the batch size to 8 and measured the latency for generating 256 tokens. The memory

Table 2: Inference efficiency of dense and SoftIGM-pruned LLaMA-2-7B models. Memory denotes FP16 model-weight footprint, and prefill/decoding latency is measured on a single NVIDIA RTX 4090 GPU.

Ratio	**Params**	**Memory** (GB) ↓	**Prefill** (s) ↓	**Decoding** (s) ↓
0%	6.74B	12.55	0.0340	5.72
60%	2.70B	5.03	0.0240	4.59
80%	1.35B	2.51	0.0164	3.48

footprint represents the VRAM required to load the model weights in half-precision (FP16/BF16). For each specified pruning ratio, the experiment was repeated 20 times to ensure statistical robustness, and the average latency was reported as the final result. All latency measurements were conducted on a single NVIDIA RTX 4090 GPU.

As detailed in Table 2, at a pruning ratio of 60%, the memory drops from 12.55 GB to approximately 5.03 GB. Simultaneously, the prefill latency decreases from 0.0340 seconds to 0.0240 seconds, and the decoding latency is reduced by about 19.8% compared to the unpruned model. At the more aggressive 80% pruning ratio, the memory is further reduced to 2.51 GB, with the decoding latency showing a substantial reduction of 39.16%. These results demonstrate that SoftIGM not only preserves reasoning capabilities but also alleviates practical deployment bottlenecks for compressed LLMs.

4.4 Ablation Studies

To systematically isolate and validate the individual contributions of the core components within our proposed method, we conduct ablation studies focusing on the gradient-matching feedback, pruning-time weight updates, and the formulation of the structure-aware matching objective. We move the detailed analysis of the multi-round soft optimization cycle to Appendix B.1.

Effect of Intermediate Gradient Matching (IGM): The core efficacy of our approach relies on IGM serving as a teacher-guided pruning signal during the soft-mask optimization phase. To validate this design, we replace the IGM feedback with (1) a pure magnitude-based metric, analogous to Wanda, and (2) a standard first-order Taylor expansion metric, both of which are used to guide unit-level mask learning while preserving the same multi-round soft-mask optimization frame-

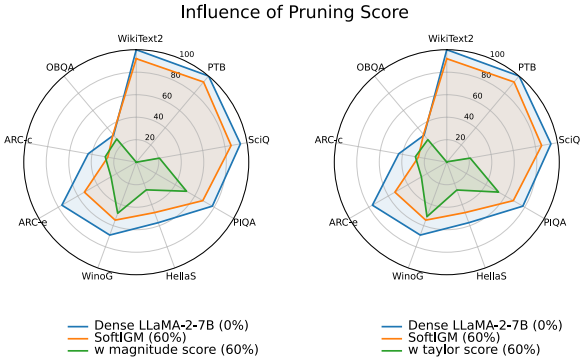

Figure 5: Influence of replacing IGM with magnitude or Taylor pruning scores at 60% sparsity.

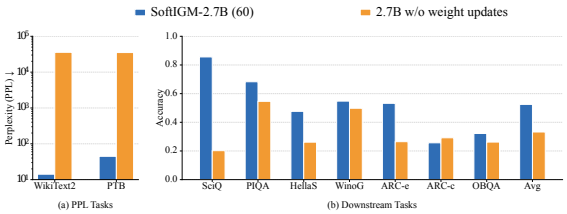

Figure 6: Ablation study on pruning-time weight updates at 60% sparsity.

work. Figure 5 reports direction-aware normalized profiles over language modeling and zero-shot tasks, where larger radial values indicate better performance. Relying on isolated magnitude or Taylor scores yields much weaker profiles, underscoring the importance of continuous teacher-guided gradient feedback for accurate mask learning.

Influence of Pruning-Time Weight Updates: To clarify the role of updating model weights while the soft masks are being optimized, we conduct an ablation in which the unit-level masks are optimized but the network weights are kept frozen throughout pruning-time mask optimization. As shown in Figure 6, freezing the network weights leads to severe perplexity collapse and weaker downstream accuracy, indicating that pruning-time weight updates are necessary for stabilizing the increasingly sparse soft structure.

Impact of Gradient Matching Mechanisms: We further investigate the behavior of different loss formulations within the Intermediate Gradient Matching Mechanism. Specifically, we compare our structure-aware row-wise gradient matching objective against a tensor-level cosine-distance variant, with the calibration batch size fixed at 4. Figure 7 compares WikiText2 and PTB perplexity, seven zero-shot tasks, and average accuracy. Our structure-aware mechanism yields a slightly better average accuracy and lower perplexity than the tensor-level cosine-distance variant.

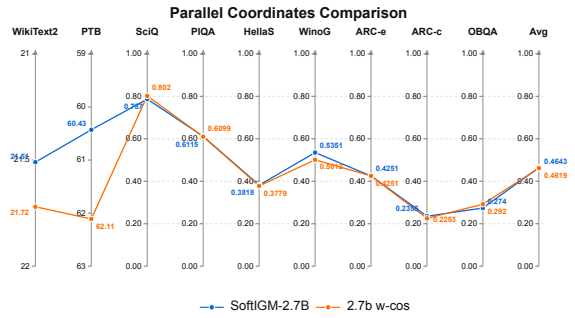

Figure 7: Ablation study on different Intermediate Gradient Matching mechanisms at 60% sparsity.

5 Conclusion

In this paper, we addressed the deployment bottlenecks of Large Language Models (LLMs) caused by their substantial computational and memory demands. We proposed **SoftIGM**, a soft structured pruning method that avoids committing early to hard binary pruning decisions. Instead, SoftIGM optimizes unit-level learnable masks and student weights in a multi-round soft-mask optimization process, while using intermediate gradient matching to align the current soft-masked student with the previous-round lower-sparsity teacher. Final hard structured pruning is performed only after the soft optimization rounds are completed.

Experiments on LLaMA-2-7B demonstrate that SoftIGM better preserves language modeling and zero-shot reasoning performance under aggressive structural compression. At 60% and 80% sparsity, SoftIGM achieves the best average zero-shot accuracy and the lowest perplexity among the compared methods, while also reducing model-weight memory footprint and inference latency. These results indicate that teacher-guided intermediate gradient matching provides an effective pruning-time signal for learning robust sparse structures without requiring a separate full post-pruning retraining stage. Future work will explore more memory-efficient gradient matching and extend SoftIGM to larger models, Mixture-of-Experts architectures, and multimodal compression settings.

Limitations

While SoftIGM shows strong empirical performance under high sparsity, it introduces additional pruning-time computational and memory overhead.

Computing parameter-gradient signals for both the previous-round teacher and the current soft-masked student requires extra forward/backward passes and gradient storage. Although memory-efficient training frameworks such as Fully Sharded Data Parallel (FSDP) can alleviate this cost, peak memory remains a practical constraint even on high-memory accelerators such as 2× NVIDIA A100 (80GB) GPUs. Scaling the gradient-matching process to substantially larger models, or applying it with much longer calibration contexts (e.g., $\geq$ 32k tokens), may therefore require further memory optimization.

In addition, SoftIGM replaces separate full post-pruning retraining with multi-round pruning-time optimization of unit-level masks and model weights. This design improves recovery during pruning, but it also leads to a longer pruning-time optimization wall-clock cost than single-step pruning heuristics. Future work should investigate lower-cost approximations to intermediate gradient matching, more selective gradient storage, and hardware-aware implementations for massive-scale LLM deployment.

References

Sajid Anwar, Kyuyeon Hwang, and Wonyong Sung. 2017. Structured pruning of deep convolutional neural networks. *ACM Journal on Emerging Technologies in Computing Systems*, 13(3):1–18.

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.

Tianyi Chen, Tianyu Ding, Badal Yadav, Ilya Zharkov, and Luming Liang. 2023. LoRAShear: Efficient large language model structured pruning and knowledge recovery. *arXiv preprint arXiv:2310.18356*.

Yuli Chen, Bo Cheng, Jiale Han, Yingying Zhang, Yingting Li, and Shuhao Zhang. 2025. DLP: Dynamic layerwise pruning in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 7934–7956.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

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*, pages 10323–10337.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. OPTQ: Accurate quantization for generative pre-trained Transformers. In *Proceedings of the Eleventh International Conference on Learning Representations*.

Jialong Guo, Xinghao Chen, Yehui Tang, and Yunhe Wang. 2025. SlimLLM: Accurate structured pruning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 20766–20776.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural network. In *Advances in Neural Information Processing Systems*, pages 1135–1143.

Babak Hassibi and David G. Stork. 1992. Second order derivatives for network pruning: Optimal Brain Surgeon. In *Advances in Neural Information Processing Systems*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2014. Distilling the knowledge in a neural network. In *NIPS Deep Learning Workshop*.

Woosuk Kwon, Sehoon Kim, Michael W. Mahoney, Joseph Hassoun, Kurt Keutzer, and Amir Gholami. 2022. A fast post-training pruning framework for Transformers. In *Advances in Neural Information Processing Systems*, pages 24101–24116.

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*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 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 *Proceedings of the Fifth International Conference on Learning Representations*.

Meta. 2023. Llama-2-7b. `https://huggingface.co/meta-llama/Llama-2-7b`. Hugging Face model card. Accessed: 2026-05-24.

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.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11264–11272.

Saurav Muralidharan, Sharath Turuvekere Sreenivas, Raviraj Joshi, Marcin Chochowski, Mostofa Patwary, Mohammad Shoeybi, Bryan Catanzaro, Jan Kautz, and Pavlo Molchanov. 2024. Compact language models via pruning and knowledge distillation. In *Advances in Neural Information Processing Systems*, pages 41076–41102.

OpenAI. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.

Swarnadeep Saha, Peter Hase, and Mohit Bansal. 2023. Can language models teach? teacher explanations improve student performance via personalization. In *Advances in Neural Information Processing Systems*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. WinoGrande: An adversarial Winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Ayan Sengupta, Siddhant Chaudhary, and Tanmoy Chakraborty. 2025. You only prune once: Designing calibration-free model compression with policy learning. In *Proceedings of the Thirteenth International Conference on Learning Representations*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Together Computer. 2023. RedPajama: An open dataset for training large language models. https://github.com/togethercomputer/RedPajama-Data. Accessed: 2026-05-26.

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. 2023. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Valve Corporation. 2026. Steam hardware & software survey: April 2026. https://store.steampowered.com/hwsurvey/Steam-Hardware-Software-Survey-Welcome-to-Steam. Accessed: 2026-05-24.

Johannes Welbl, Nelson F. Liu, and Matt Gardner. 2017. Crowdsourcing Multiple Choice Science Questions. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 94–106.

Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. 2016. Learning structured sparsity in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2074–2082.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2024. Sheared LLaMA: Accelerating language model pre-training via structured pruning. In *Proceedings of the Twelfth International Conference on Learning Representations*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 38087–38099.

Peng Xu, Wenqi Shao, Mengzhao Chen, Shitao Tang, Kaipeng Zhang, Peng Gao, Fengwei An, Yu Qiao, and Ping Luo. 2024. BESA: Pruning large language models with blockwise parameter-efficient sparsity allocation. In *Proceedings of the Twelfth International Conference on Learning Representations*.

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.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024a. LoRAPrune: Structured pruning meets low-rank parameter-efficient fine-tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3013–3026.

Yingtao Zhang, Haoli Bai, Haokun Lin, Jialin Zhao, Lu Hou, and Carlo Vittorio Cannistraci. 2024b. Plug-and-Play: An efficient post-training pruning method for large language models. In *Proceedings of the Twelfth International Conference on Learning Representations*.

A Implicit Recovery via Gradient-Matched Updates

We provide a first-order explanation for why Soft-IGM does not require a separate full post-pruning retraining stage. In each pruning round, the current student is optimized under unit-level soft structural masks while the masks themselves are still being learned. Hence, weight adaptation is already integrated into pruning-time optimization rather than postponed until after final hard structured pruning.

Consider a local view of pruning step t in round r, where the current unit-level soft mask $\mathbf{M}^{(t)} = \{m_u^{(t)}\}_{u \in \mathcal{U}}$ is treated as fixed for the backbone update. Let $\Pi_{\mathbf{M}^{(t)}}(\cdot)$ denote the broadcast masking operator that applies these unit-level gates to the corresponding structural groups. This operator is a notational shorthand for the current soft-gated student and should not be interpreted as a scalar-parameter mask or a permanent binary topology. If the current soft-gated sub-network were explicitly fine-tuned after pruning, a first-order masked update could be written as

$$\boldsymbol{\theta}_{t+1}^{ret} := \Pi_{\mathbf{M}^{(t)}} \left(\boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \mathcal{L}_{task}^{S}(\boldsymbol{\theta}_t) \right). \quad (8)$$

SoftIGM performs an analogous update during pruning, but augments the task objective with gradient matching against the previous-round lower-sparsity teacher:

$$\boldsymbol{\theta}_{t+1}^{IGM} := \Pi_{\mathbf{M}^{(t)}} \left(\boldsymbol{\theta}_t - \eta \nabla_{\boldsymbol{\theta}} \left(\mathcal{L}_{task}^{S}(\boldsymbol{\theta}_t) + \beta \mathcal{L}_{IGM}(\boldsymbol{\theta}_t) \right) \right). \quad (9)$$

Here $\mathcal{L}_{lag}$ is omitted from the displayed backbone update because it primarily constrains the unit-level mask variables and Lagrange multipliers; it continues to shape the soft structure through Equation 6.

Proposition. Assume that, within pruning step t, the current unit-level soft mask $\mathbf{M}^{(t)}$ is fixed for the local backbone update, the same mini-batch from $\mathcal{D}_{batch}$ is used, and the optimizer hyperparameters and learning-rate schedule are shared. If the masked gradient-matching direction is bounded by

$$\|\Pi_{\mathbf{M}^{(t)}} \nabla_{\boldsymbol{\theta}} \mathcal{L}_{IGM}\|_2 \leq C\delta, \quad (10)$$

where δ denotes the gradient-matching discrepancy, then the SoftIGM update differs from the corresponding masked fine-tuning update by

$$\left\| \boldsymbol{\theta}_{t+1}^{IGM} - \boldsymbol{\theta}_{t+1}^{ret} \right\|_2 \leq \eta \beta C \delta. \quad (11)$$

Proof. With fixed gates, the broadcast masking operation is linear with respect to the masked weights in this local update. Subtracting Equation 8 from Equation 9 gives

$$\boldsymbol{\theta}_{t+1}^{IGM} - \boldsymbol{\theta}_{t+1}^{ret} = -\eta \beta \Pi_{\mathbf{M}^{(t)}} \left(\nabla_{\boldsymbol{\theta}} \mathcal{L}_{IGM} \right). \quad (12)$$

Taking the ℓ_2 norm and applying the boundedness assumption on $\mathcal{L}_{IGM}$ yields Equation 11. Therefore, when the gradient-matching discrepancy is small, each SoftIGM backbone update remains close to a fine-tuning update on the current soft-gated sub-network, with an additional bounded correction from the previous-round lower-sparsity teacher. This supports the interpretation that SoftIGM performs pruning-time recovery continuously and thereby reduces the need for a separate full post-pruning fine-tuning stage.

B Additional Experimental Analyses

B.1 Impact of the Multi-round Soft Optimization Cycle

We investigate the fundamental necessity of the progressive soft-mask optimization curriculum. Specifically, we contrast our multi-round approach against a single-step unit-mask determination strategy that utilizes the exact same IGM metric to aggressively reach the target sparsity in a single, unadjusted step. The results in Table 3 demonstrate that executing compression in a single, unadjusted step severely compromises the model's capabilities, validating the essential role of our soft optimization cycle in preventing catastrophic structural collapse.

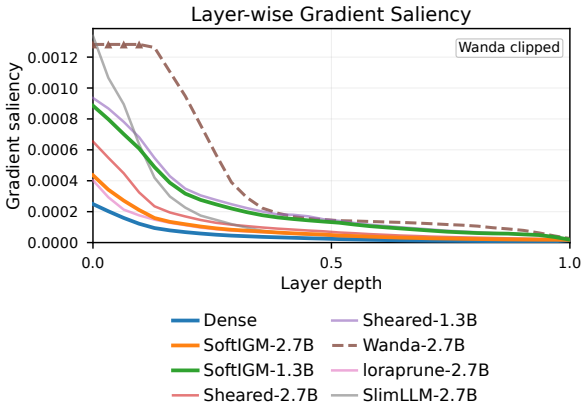

Figure 8: Layer-wise hidden-state gradient saliency on HellaSwag. Each curve averages the saliency map over token positions. Higher values indicate stronger loss sensitivity at a layer, not better task performance.

Table 3: Ablation study on the impact of the multi-round soft optimization cycle at a 60% pruning ratio.

Pruning Steps	WikiText2	PTB	Avg.
1 (Single-step)	59408.20	133878.44	32.99
3200 (SoftIGM)	**14.16**	**44.74**	**52.49**

B.2 Layer-wise Gradient Saliency Profiles

Figure 8 summarizes the hidden-state gradient saliency maps by averaging over token positions. Some baselines exhibit sharper early-layer sensitivity, whereas SoftIGM produces smoother profiles that remain closer to the dense model.

B.3 Qualitative Comparison of Model Outputs Before and After Pruning

To complement the quantitative zero-shot and perplexity results, we compare the model outputs before and after structural compression under an identical decoding configuration. We use the dense LLaMA-2-7B model as the reference and the final exported 60% SoftIGM-pruned model as the compressed model. The prompts are selected to cover commonsense reasoning, factual reasoning, and instruction following. Table 4 records the ex-

act output strings from both models and highlights whether the final exported model preserves the core answer, reasoning path, and instruction-following behavior.

Table 4: Qualitative comparison of model outputs before and after pruning under the same prompt and decoding settings. The examples cover factual recall, commonsense reasoning, and simple instruction following.

Model	Generated Sentences
LLaMA-2-7B (ratio=0%)	Instruction: Name the largest ocean on Earth. Response: *Pacific Ocean.*
	Instruction: Which is heavier, one kilogram of steel or one kilogram of feathers? Response: *One kilogram of steel is heavier than one kilogram of feathers.*
	Instruction: If all cats are animals, is a cat an animal? Answer yes or no. Response: *If all cats are animals, then a cat is an animal.*
SoftIGM-2.7B (ratio=60%)	Instruction: Name the largest ocean on Earth. Response: *The largest ocean on Earth is the Pacific Ocean.*
	Instruction: Which is heavier, one kilogram of steel or one kilogram of feathers? Response: *The heavier is the steel.*
	Instruction: If all cats are animals, is a cat an animal? Answer yes or no. Response: *Yes, a cat is an animal.*