From Local to Global: Revisiting Structured Pruning Paradigms for Large Language Models

Anonymous ACL submission

Abstract

Structured pruning is a practical approach to deploying large language models (LLMs) efficiently, as it yields compact, hardware-friendly architectures. However, the dominant local paradigm is task-agnostic: by optimizing layer-wise reconstruction rather than task objectives, it tends to preserve perplexity or generic zero-shot behavior but fails to capitalize on modest task-specific calibration signals, often yielding limited downstream gains. We revisit global structured pruning and present GISP—Global Iterative Structured Pruning—a post-training method that removes attention heads and MLP channels using first-order, loss-based important weights aggregated at the structure level with block-wise normalization. An iterative schedule, rather than one-shot pruning, stabilizes accuracy at higher sparsity and mitigates perplexity collapse without requiring intermediate fine-tuning; the pruning trajectory also forms nested subnetworks that support a 'prune-once, deploy-many' workflow. Furthermore, because importance is defined by a model-level loss, GISP naturally supports task-specific objectives; we instantiate perplexity for language modeling and a margin-based objective for decision-style tasks. Extensive experiments show that across Llama2-7B/13B, Llama3-8B, and Mistral-0.3-7B, GISP consistently lowers WikiText-2 perplexity and improves downstream accuracy, with especially strong gains at 40–50% sparsity; on DeepSeek-R1-Distill-Llama-3-8B with GSM8K, task-aligned calibration substantially boosts exact-match accuracy.

1 Introduction

Pruning (Ma et al., 2023; Frantar and Alistarh, 2023; Sun et al., 2024; Kim et al., 2024; An et al., 2023) is a fundamental technique for compressing neural networks by removing redundant parameters while preserving accuracy. Broadly, existing approaches fall into two categories: unstructured pruning, which removes element-wise weights without shrinking the model architecture, and structured pruning, which eliminates entire groups of weights (e.g., channels, attention heads, layers). It is well established that unstructured pruning can achieve higher sparsity levels but typically requires specialized sparse computation kernels to realize runtime speedups, whereas structured pruning inherently produces compact, hardware-friendly architectures and is therefore preferred for practical deployment. With the rapid emergence of Large Language Model (LLMs) (Touvron et al., 2023a,b; OpenAI et al., 2024; Chiang et al., 2023; Workshop et al., 2023; Grattafiori et al., 2024) containing billions of parameters, pruning has become critical to improve inference efficiency and to enable deployment on resource-constrained devices.

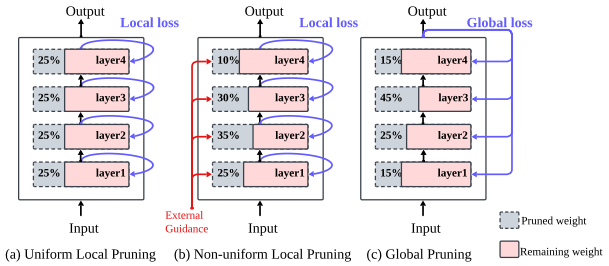

Figure 1: Comparison between (a) local pruning, which uses layer-wise reconstruction loss as the importance criterion, (b) non-uniform variants of local pruning, and (c) global pruning, which directly considers the impact of weight pruning on the final model loss.

The dominant paradigm for pruning LLMs is **local pruning**, exemplified by methods like SparseGPT (Frantar and Alistarh, 2023) and Wanda (Sun et al., 2024). These methods gained significant attention due to their simplicity and efficiency, breaking down the model-wide optimization into layer-wise sub-problems (Fig.1(a)). This decomposition allows them to prune each layer gradually, typically by minimizing a layer-wise reconstruction with calibration data, offering a post-training solution. Furthermore, to mitigate the

rigidity of uniform sparsity, recent work explores **non-uniform local pruning** (Fig. 1(b)), which adjusts layer-wise ratios based on estimated importance. Methods such as OWL (Yin et al., 2025), FLAP (An et al., 2023), and DarwinLM (Tang et al., 2025) leverage activation statistics or evolutionary search to assign non-uniform sparsity. While these approaches improve accuracy, they remain rooted in layer-wise reconstruction and introduce notable algorithmic complexity and overhead.

While local structured pruning is efficient and preserves broad behavior (often reflected in perplexity or generic zero-shot scores), **its objective is task-agnostic**. When modest task-informed calibration is available, local methods rarely capitalize on it, yielding limited downstream gains. *This gap calls for a loss-aligned alternative defined at the model level, instead of a local proxy.* We therefore revisit **global pruning** (Fig.1(c)) for LLMs and develop **GISP—Global Iterative Structured Pruning**. Unlike local approaches that optimize layer reconstructions, global pruning defines importance with respect to a model-level loss, naturally inducing non-uniform sparsity without extra heuristics. Operationally, GISP aggregates first-order, loss-based importance at the structure level (attention heads and MLP channels) with block-wise normalization, and we study it in a post-training setting (no fine-tuning between steps) to match practical constraints.

Building on this formulation, we first validate that making global pruning iterative fundamentally changes its behavior. A gradual, ratio-scheduled process turns the otherwise unstable one-shot global pruning into a robust procedure that preserves model quality even at high sparsity. Furthermore, the same iterative trajectory also reveals a nested structure across sparsity levels, showing that iterative global pruning can serve as a single, continuous optimization rather than a series of independent runs, thereby enabling a 'prune-once, deploy-many' workflow. Finally, because importance is defined by a model-level loss, GISP can directly integrate task objectives, bridging the gap between generic compression and task-aware optimization; this property consistently yields stronger downstream accuracy across models and pruning ratios.

We summarize our contributions as follows:

- We present GISP, a simple and effective global iterative structured pruning framework for LLMs that operates post-training and stabilizes performance at high sparsity.
- We demonstrate that iterative global pruning follows a smooth, nested trajectory of subnetworks, enabling a 'prune-once, deploy-many' workflow with a competitive amortized time cost per usable model compared to local pruning baselines.
- We examine the task-specific property of GISP and instantiate task-specific global importance via multiple objectives. Extensive experiments demonstrate consistent downstream gains across models and pruning ratio levels.

2 Preliminary: Local *vs* Global Pruning

2.1 Local Pruning and Non-uniform Variant

Rationale of local pruning. Given a pre-trained model with weights as θ and a set of calibration dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ with N samples, the structure pruning in LLMs with L transformer (Vaswani et al., 2023) layers can be interpreted as finding optimal $\widehat{\theta}$ under desired sparsity ratio constraints by removing sets of coupled structures W_l from $G = (\{W_{l,\text{Attn}}\}_{l=1}^{L}, \{W_{l,\text{MLP}}\}_{l=1}^{L})$ with minimal error on a pre-defined objective function.

As introduced by the pioneering work OBS (Hassibi and Stork, 1992) and layer-wise OBS (Dong et al., 2017), local pruning defines the objective function by breaking down the problem of full model compression into sub-problems for each layer. It constructs a local loss to measure the L_2 error between the outputs of the unpruned and pruned layers, which can be formulated as:

$$\min_{M_\ell, \widehat{W}_\ell} \left\| W_\ell X_\ell - \left(M_\ell \odot \widehat{W}_\ell\right) X_\ell \right\|_2^2, \tag{1}$$

where W_ℓ is the original weight of layer ℓ, X_ℓ is the input to layer ℓ, M_ℓ is the binary mask indicating which weights to keep, and $\widehat{W}_\ell$ denotes the post-pruning weights, which may be kept fixed as W_ℓ or re-optimized given M_ℓ.

Among local structured pruning methods, SparseGPT (Frantar and Alistarh, 2023) formulates pruning as a sparse regression problem solved via an approximate Hessian inversion. ZipLM (Kurtic et al., 2023) extends the OBS formulation to structured pruning and performs inference-aware search over structures. Wanda (Sun et al., 2024) simplifies SparseGPT's importance to weight–activation

products, achieving similar accuracy with higher efficiency. LLM-Pruner (Ma et al., 2023) further prunes entire attention heads and MLP channels using gradient information to capture inter-structure dependencies. GBLM (Das et al., 2024) augments Wanda's scores with gradient magnitude, while Wanda++ (Yang et al., 2025) adds block-level gradients from activation reconstruction to improve layer-wise pruning; Pruner-Zero (Dong et al., 2024) instead uses genetic programming to discover gradient-based layer-local importance metrics from weight/activation/gradient signals. Finally, several works explore layer-wise pruning (Kim et al., 2024; Men et al., 2024), such as ShortGPT (Men et al., 2024), which leverages layer-wise activation similarity.

Non-uniform variants of local pruning. To overcome the limitation of uniform sparsity in layer-wise pruning, several works introduce non-uniform local pruning (Fig. 1(b)) that adjusts pruning ratios across layers based on estimated importance. These methods extend the layer-wise reconstruction paradigm by incorporating inter-layer sensitivity through diverse heuristics: FLAP (An et al., 2023) exploits activation variability to assign flexible sparsity, OWL (Yin et al., 2025) reweights layers according to outlier statistics in activations, and DarwinLM (Tang et al., 2025) performs a training-aware evolutionary search to identify optimal sparsity configurations.

2.2 Global Pruning

Global pruning aims to find a global sparsity mask M and possibly updated weights $\widehat{W}$ to minimize the global loss between the final outputs of the uncompressed and compressed model. Hence, the learning objective can be formulated as:

$$\min_{M,\,\widehat{W}} \Delta\mathcal{L}\Big(f\big(X;\, M \odot \widehat{W}\big),\; f(X;W)\Big), \tag{2}$$

where f is the forward function, X denotes the inputs, W is the original (pre-trained) weight, M is the binary mask indicating which weights remain. Following the idea from OBD (LeCun et al., 1989) of conducting the Taylor series towards loss distance on parameter perturbation caused by pruning, we have element-wise importance given by

$$I_{W_i^j} = \left|\Delta\mathcal{L}(D)\right| = \left|\mathcal{L}(D;\theta_{W_i^j}) - \mathcal{L}(D;\theta_{W_i^j=0})\right| = \left|\frac{\partial\mathcal{L}(D)}{\partial W_i^j} W_i^j \;-\; \frac{1}{2} W_i^j H_{jj} W_i^j \;+\; \mathcal{O}(\|W_i^j\|^3)\right|, \tag{3}$$

where $I_{W_i^j}$ marks the $j-th$ estimated importance of element in θ, H_{jj} is diagonal of the hessian matrix. Global pruning has been extensively studied in smaller networks such as CNNs, Vision Transformers, and compact language models (Molchanov et al., 2016; Yang et al., 2023; Diao et al., 2023), consistently outperforming local approaches (Blalock et al., 2020; Diao et al., 2023). In LLMs, LLM-Pruner (Ma et al., 2023) applies Eq. 3 for element-wise importance and explores structure-level aggregation, while LoRAPrune (Zhang et al., 2024) adapts it to LoRA for memory-efficient fine-tuning. Although higher-order derivatives can be included, prior work in both CNNs and LLMs (Ma et al., 2023; Molchanov et al., 2019; Zhang et al., 2024) shows that first-order information alone is sufficient for competitive results.

Motivation. While local structured pruning is appealing for its efficiency, it remains fundamentally task-agnostic. These methods minimize layer-wise reconstruction errors to preserve input–output similarity with the dense model, which maintains perplexity and generic zero-shot behavior but does not optimize downstream accuracy. As shown in Table 1, we evaluate local methods on CMQA using Llama 2-13B under two calibration settings: a generic C4 corpus and task-specific CMQA samples. Even with task-informed calibration, the improvement is marginal, indicating that local pruning cannot effectively exploit task signals.

In contrast, global pruning defines importance with respect to the overall model loss, naturally producing non-uniform sparsity patterns. Because its importance scores are computed on calibration data, global pruning can directly align pruning decisions with downstream objectives. This motivates our exploration of task-specific global iterative structured pruning, which unifies the efficiency of post-training methods with the flexibility to incorporate task-aware objectives.

3 Method

3.1 A Naive Case Study: One-shot Global Pruning

To assess the effectiveness and limitations of global pruning, we first replicate prior pruning protocols designed for smaller models (Frankle and Carbin, 2018; Mallya and Lazebnik, 2018). Specifically, we conduct the following procedure:

1. Given a calibration dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, we calculate the importance score of each weight element by using the first order term

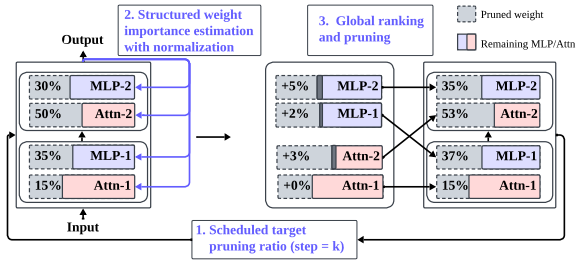

Figure 2: A detailed overview of the GISP. GISP performs iterative structured pruning guided by a ratio scheduler.

from eq. (3).

2. We construct the structure-wise importance scores by summing a group of importance scores of each weight element, where a group is defined as one head of the attention layer or a channel of the MLP layer. After the aggregation, a normalization process is conducted in consideration of group size among different structures, allowing all structural importance to be comparable across the model.
3. Given a current pruning ratio ρ, globally rank the normalized structure-wise scores and select the Top-K lowest-scoring structured weights for pruning.

Compared to prior pruning works on small models (Frankle and Carbin, 2018; Mallya and Lazebnik, 2018), we make two differences: (1) these methods typically adopt a prune–then–fine-tune paradigm. In contrast, we evaluate performance in the setting of post-training pruning without fine-tuning, consistent with common local pruning practices for LLMs, as the computational and memory costs of fine-tuning after each pruning iteration are prohibitive, and (2) we observe that attention blocks exhibit substantially higher importance scores than MLP blocks (see Fig 3(c)), often by an order of magnitude. To address this imbalance, we normalize importance scores within attention and MLP blocks separately, which empirically yields improved accuracy.

Empirical study. We perform experiments on one-shot global pruning and compare it with one representative structured local pruning method, Wanda. These experiments are conducted on Llama2-7B with a target pruning ratio of 20-50%. As shown in Table 2, one-shot global pruning surpasses Wanda at low ratios but degrades at high sparsity, indicating that naive one-shot pruning is viable yet unstable, especially in high-pruning-ratio regions,

Table 1: Overall average CMQA accuracy (%) under different calibration datasets and pruning ratios. Local pruning methods have limited performance improvement, even with a task-informed calibration dataset.

Method	Calibration Data	Pruning Ratio	AVG ACC
Wanda-sp	C4	20%	66.36
		40%	58.24
	CMQA	20%	66.30
		40%	59.11
LLM-Pruner	C4	20%	65.63
		40%	55.06
	CMQA	20%	57.30
		40%	41.99
FLAP	C4	20%	66.15
		40%	61.73
	CMQA	20%	65.00
		40%	59.89
OWL	C4	20%	66.62
		40%	59.87
	CMQA	20%	66.98
		40%	60.84

Table 2: Evaluation of one-shot global pruning (marked as one-shot GP) on perplexity (PPL) with C4 as the calibration dataset.

Method	**Pruning ratio**	**Wiki2**	**PTB**
Wanda-sp	20%	22.71	101.23
	30%	35.43	138.41
	40%	51.85	185.09
	50%	81.47	218.31
One-shot GP	20%	17.93	63.09
	30%	26.99	81.48
	40%	53.45	151.80
	50%	159.47	353.55

which motivates our iterative strategy introduced next.

3.2 GISP: Global Iterative Structured Pruning

3.2.1 Stabilizing High-Ratio Pruning

Motivation. We hypothesize that one potential issue of one-shot global pruning is that it removes a large portion of weights at once, increasing the risk of over-pruning important weights. A potential solution to this issue is iterative global pruning, which gradually prunes the model by applying a small pruning ratio in each step. This approach enables more precise identification of truly redundant weights, leveraging iterative feedback to refine pruning decisions.

Global Iterative Structured Pruning. Building upon the procedure detailed in Section 3.1, given a predefined number of iteration steps n and a target pruning ratio ρ, GISP performs pruning iteratively using a small pruning ratio ρ_k at iteration k, as

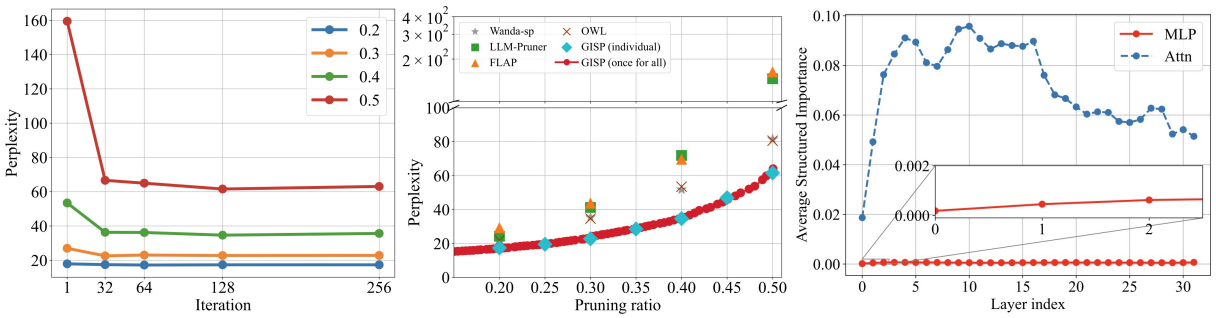

Figure 3: (a) Perplexity analysis for various iteration settings. Iteration alleviates high-pruning-ratio perplexity collapse. (b) Perplexity analysis between GISP and other baselines. (c) Magnitude comparison between different types of structured weight importance.

Algorithm 1 Global Iterative Structured Pruning

Require: Network parameters θ_0; global target pruning ratio ρ; iteration steps n; calibration dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$.
Ensure: Pruned model parameters θ_n.

$\{\rho_t\}_{t=1}^{n} \leftarrow \text{RATIOSCHEDULER}(\rho, n)$
for $k \leftarrow 1$ to n **do**
  $I(\theta_{k-1}) \leftarrow |\frac{\partial \mathcal{D}}{\partial \theta_{k-1}} \odot \theta_{k-1}|$ ▷ First-order importance
  $I(\theta_{k-1}) \leftarrow \text{SUMAGGREGATE}(I(\theta_{k-1}))$
  $I(\theta_{k-1}) \leftarrow \frac{I(\theta_{k-1})}{|\theta_{k-1}|}$ ▷ Normalized Aggregation over heads / MLP channels
  $\tau_k \leftarrow \text{TOPK}(I(\theta_{k-1}), \rho_k)$ ▷ Global pruning threshold across different structures at ratio ρ_k
  $m \leftarrow \mathbf{1}[I(\theta_{k-1}) > \tau_k]$ ▷ Binary mask
  $\theta_k \leftarrow \theta_{k-1} \odot m$ ▷ Apply pruning mask
end for
return θ_n

shown in Fig 2. The detailed algorithm block is provided in Algorithm 1. To control pruning at each step, we use a linear scheduler that gradually increases the pruning ratio across iterations, ensuring that each iteration prunes the same number of structures. More details can be found in Sec. A.2

Empirical study. For the iteration study, we vary the number of pruning steps (1, 32, 64, 128, and 256) across four target pruning ratios (20%, 30%, 40%, and 50%). For comparison with local pruning, we measure perplexity (PPL) and include four representative post-training structured pruning baselines: two uniform local pruning methods (Wanda and LLM-Pruner) and two non-uniform local pruning methods (FLAP and OWL). All experiments are performed on the Llama2-7B model using the C4 calibration dataset. The results are shown in Fig 3. We summarize our main findings below:

1) ***Iteration is the key for global pruning at a high pruning ratio region***. From Fig 3(a) and (b), we first observe that introducing iterative pruning alleviates the issue of global pruning at a high pruning ratio: even a coarse setting of 32 steps (equal to the layer count of Llama2-7B) is enough to cut the 50%-pruning-ratio PPL by 92.82. As a result, the GISP can consistently achieve lower PPL compared to local pruning methods.

Table 3: Pruning time comparison across methods. "Amortized time" divides the total time by the number of usable subnetworks produced.

Method	Usable sparsities	Total time (min)	Amortized time (min)
Wanda-sp	4	7.10	1.78
OWL	4	13.90	3.48
LLM-Pruner	4	6.80	1.70
GISP (ours)	112	125.84	1.12

2) ***Global iterative pruning outperforms local baselines at scale.*** With iteration, global pruning consistently achieves lower perplexity than local pruning methods across all sparsity regimes (Fig. 3(b)). This establishes that iteration not only stabilizes global pruning but also makes it competitive against strong local baselines in the post-training LLM setting. Crucially, these gains are obtained without intermediate recovery or fine-tuning, demonstrating that iteration alone is effective.

3.2.2 Achieving "Once-for-All" and Amortizing the Iteration Cost

Iterative global pruning is computationally more demanding than local or one-shot pruning. Running such a computationally intensive procedure to obtain only a single subnetwork at a fixed sparsity level would be impractical in deployment. Table 3 compares the wall-clock pruning time of several structured pruning methods under our experimental setup. While GISP requires a substantially longer total runtime due to its iterative steps, the *amortized* cost per deployable subnetwork is comparable to, or even lower than, that of local methods once the once-for-all property is considered.

Moreover, in iterative global pruning, each step removes new self-attention heads and MLP channels based on the already-pruned model from the previous step, naturally forming a *nested subnetwork* structure (Cai et al., 2019). This nested property and computational cost from iteration motivated us to wonder:

*Can GISP enable **once-for-all** pruning? In other words, if we iteratively prune toward a high target ratio (e.g., 50%), can the intermediate subnetworks with lower pruning ratios (e.g., 20%, 30%) already perform well, thereby eliminating the need to conduct separate pruning runs for each individual pruning ratio?*

To investigate this, we conduct a single iterative pruning procedure on Llama2-7B, targeting 50% sparsity over 112 iterations. We saved the intermediate pruned model at every step and evaluated its PPL. The results are presented in Fig 3(b). The relationship between perplexity and the pruning ratio is remarkably smooth, indicating a stable and well-behaved pruning trajectory. Crucially, the performance of the intermediate models at different pruning ratios is on par with the performance of models generated from individual, shorter pruning runs (marked as "individual" variant) tuned specifically for those respective targets. To this end, it demonstrates the *once-for-all* capacity of GISP.

It is important to note that this "once-for-all" capability is a unique advantage of the iterative global pruning. It enables practitioners to obtain an entire Pareto frontier of accuracy-vs-sparsity models from a single computational investment, offering immense practical flexibility. This property is not achievable with local pruning methods. As formulated in Eq. 1, local pruning is a layer-wise optimization that requires the target pruning ratio for each layer to be specified in advance. Consequently, creating models for different sparsity levels necessitates entirely separate and independent pruning runs.

3.3 GISP as a Task-Specific Pruner

As discussed in Sec. 2, local pruning remains task-agnostic because its layer-wise reconstruction objective cannot align with downstream goals, even when calibration data carries task-specific information. In contrast, global pruning defines importance with respect to a model-level loss, offering the potential for task alignment. We now instantiate and validate this property in GISP.

Objective-level formulation. Because GISP evaluates importance with respect to a *model-level* loss (Eq. 3), we can instantiate a *task-aligned* objective by replacing the loss in Eq. 2 with a task-specific target L_{task}. Our importance reduces to the same first-order form with a different objective:

$$I_W = \left|\langle \nabla_W L_{\text{task}}, W \rangle\right|. \tag{4}$$

This simple substitution turns GISP into a *task-specific* pruner while remaining post-training and structure-aware.

Two instantiations. We consider two common families of L_{task} that match our evaluation tasks:

(i) **Perplexity loss** for text generation (language modeling), where L_{task}=token-level cross-entropy on an open-domain (e.g., C4) or in-domain (e.g., GSM8K) corpus; To be specific, the importance metrics are obtained from objective:

$$L = -\frac{1}{N}\sum_{i=1}^{N} \log p(x_i|x_{<i}) \tag{5}$$

where L is the loss function, N is the number of tokens and $p(x_i|x_{<i})$ is the probability of token x_i given all previous tokens.

(ii) **Margin loss** for decision-oriented, multi-option QA. For example, the CMQA dataset differs from pure language modeling in that each question is paired with one correct (positive) and multiple incorrect (negative) answers. During inference, the model ranks each 'Question + Answer' pair by perplexity and selects the answer with the lowest score. Simply minimizing perplexity on positive answers is insufficient, as pruning may disproportionately reduce the loss of negative candidates relative to the correct one, causing the model to choose an incorrect answer even if the correct answer's loss remains largely unchanged. In other words, *the actual factor of classification performance is the model's ability to distinguish correct from incorrect answers (the decision boundary).*

To preserve the model's decision boundary, we define a margin-based importance using a task-formatted calibration set:

$$I_{W_i^j} = \left|\left(\frac{\partial L_+}{\partial W_i^j} - \frac{\partial L_-}{\partial W_i^j}\right) W_i^j\right| \tag{6}$$

Where L_+ denotes the average loss on positive labels and L_- denotes the average loss on negative labels. Intuitively, Eq. (6) preserves the loss gap between correct and incorrect candidates, aligning pruning with task decisions. We will examine the

Table 4: Comparison of different pruning methods on text generation perplexity and commonsense reasoning accuracy. All downstream accuracy is evaluated by using CMQA calibration.

Pruning Ratio	Method	Perplexity on Wikitext2 ↓		Downstream ACC (%) ↑	
		Llama2		Llama2	
		7B	13B	7B	13B
0%	Dense	12.19	10.98	66.68	69.19
20%	Wanda-sp	22.71	**14.64**	61.77	66.30
	LLM-Pruner	24.25	19.99	50.95	57.30
	FLAP	29.19	16.95	61.27	65.00
	ShortGPT	43.88	19.95	55.75	60.84
	OWL	21.80	14.76	62.64	66.98
	GISP (ours)	**17.01**	15.10	**63.46**	**67.61**
30%	Wanda-sp	35.43	19.73	57.14	62.69
	LLM-Pruner	41.24	28.47	41.37	46.26
	FLAP	43.75	21.32	56.90	63.28
	ShortGPT	126.42	84.84	50.01	56.86
	OWL	34.64	**19.02**	58.33	63.27
	GISP (ours)	**24.27**	19.53	**60.68**	**66.12**
40%	Wanda-sp	51.85	32.91	50.12	59.11
	LLM-Pruner	71.93	50.01	39.13	41.99
	FLAP	69.64	37.76	53.01	59.89
	ShortGPT	189.17	92.38	45.35	48.73
	OWL	53.47	31.13	51.50	60.84
	GISP (ours)	**34.54**	**26.56**	**55.28**	**63.34**
50%	Wanda-sp	81.47	64.17	43.52	51.60
	LLM-Pruner	144.99	86.34	38.62	40.92
	FLAP	161.84	66.38	47.84	56.29
	ShortGPT	387.94	276.08	41.75	41.75
	OWL	80.59	65.28	44.82	54.17
	GISP (ours)	**64.07**	**42.07**	**48.54**	**57.50**

Table 5: Comparison of different pruning methods for advanced models.

Pruning Ratio	Method	Perplexity on Wikitext2 ↓		Downstream ACC (%) ↑	
		Llama3 8B	Mistral-0.3 7B	Llama3 8B	Mistral-0.3 7B
0%	Dense	14.14	15.14	69.99	70.47
20%	Wanda-sp	29.92	20.42	57.45	64.39
	LLM-Pruner	**23.21**	\	56.51	\
	ShortGPT	118.62	52.74	57.68	57.75
	OWL	29.49	19.98	59.95	65.87
	GISP (ours)	24.18	**18.17**	**65.28**	**66.60**
30%	Wanda-sp	48.83	32.61	52.03	58.24
	LLM-Pruner	37.78	\	47.46	\
	ShortGPT	3972.28	599.82	43.53	41.10
	OWL	47.90	31.82	52.24	58.54
	GISP (ours)	**31.73**	**25.58**	**59.66**	**63.48**
40%	Wanda-sp	81.67	55.41	43.61	51.89
	LLM-Pruner	67.58	\	41.82	\
	ShortGPT	1576.47	909.21	43.37	39.68
	OWL	87.01	47.85	44.87	54.36
	GISP (ours)	**46.10**	**34.31**	**53.51**	**58.30**
50%	Wanda-sp	133.29	79.41	41.32	44.38
	LLM-Pruner	125.91	\	39.67	\
	ShortGPT	4135.73	1091.73	41.19	38.73
	OWL	130.77	76.20	41.86	46.09
	GISP (ours)	**79.42**	**58.16**	**45.68**	**49.79**

effectiveness of GISP as a task-specific pruner in Sec. 4.2. Importantly, such a transition from a perplexity-based loss to a task-specific loss is not feasible for local pruning methods, which rely on layer-wise MSE loss for importance estimation.

4 Experiments

Models and Evaluation. We evaluate GISP on the popular Llama2-7B/13B (Touvron et al., 2023b), Llama3-8B (Grattafiori et al., 2024), Mistral-0.3-7B (Jiang et al., 2023), and one reasoning model DeepSeek-R1-Distill-Llama-3-8B (DeepSeek-AI et al., 2025). Following previous work (Ma et al., 2023; An et al., 2023), we evaluate the pruned model on three categories of tasks: the perplexity metric on Wikitext2 (Merity et al., 2016) text generation, post-training accuracy on commonsense reasoning (CMQA), which including BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2019), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), and OpenbookQA (Mihaylov et al., 2018) and exact-match-accuracy on math task GSM8K (Cobbe et al., 2021) that require reasoning. We report average accuracy in this section, and detailed task-wise accuracy is presented in the Sec. A.4.

Baselines. We compare GISP with four local pruning approaches in two main categories: (1) local uniform baselines, including Wanda-sp (Sun et al., 2024; An et al., 2023), LLM-Pruner (Ma et al., 2023); and (2) local non-uniform baselines: FLAP (An et al., 2023), OWL (Yin et al., 2025) on Wanda-sp. Additionally, we compare against a layer-wise pruning approach, ShortGPT (Men et al., 2024). Following the general setting, we use the C4 dataset as the calibration dataset for text generation tasks. For the downstream accuracy on CMQA, we use its training set as the calibration dataset. For the exact-match-accuracy on GSM8K, we applied both the C4 dataset and the GSM8K training set as the calibration dataset. The iteration step in GISP is set to 112 in models with a 7-8B scale, and 280 in 13B to maintain the close iteration stride with these smaller variants. The detailed experimental setup is illustrated in the Sec. A.3.

4.1 Experimental Results

Perplexity of text generation tasks. Table 4 and Table 5 present the experimental results on the perplexity (PPL) of WikiText2 across four target pruning ratio levels. First of all, compared to the five baselines on dense Llama2-7B, 13B models, GISP achieves a clear lower PPL in most cases. Specifically, the improvement is particularly more significant at the higher sparsity level (e.g, 40%, and 50%). Moreover, for multi-query attention–based LLMs such as Llama3-8B and Mistral-0.3, we observe the same consistent trend [1].

[1]We exclude the results of FLAP on Llama3-8B and Mistral-0.3, and leave LLM-Pruner on Mistral-0.3 as blank

Table 6: CMQA Accuracy of GISP on all seven tasks under different calibration datasets and pruning ratios. The best results are marked in **bold**.

Calibration Dataset	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
C4 + Perplexity	20%	73.30	78.45	73.09	68.11	67.59	42.92	42.00	63.64
	30%	69.20	76.71	69.68	65.98	62.67	37.46	40.80	60.36
	40%	65.14	73.67	62.80	61.64	54.55	33.87	36.80	55.49
	50%	58.41	68.28	50.79	58.64	44.02	28.41	32.20	48.68
CMQA + Perplexity	20%	**80.80**	77.86	76.39	71.82	74.75	**46.33**	42.20	67.16
	30%	80.83	75.52	71.91	69.69	72.22	44.71	41.60	65.21
	40%	79.54	72.52	63.36	67.88	67.85	41.81	39.20	61.74
	50%	**76.18**	67.52	51.67	61.56	59.47	36.77	36.40	55.65
CMQA + Margin	20%	80.28	**79.00**	**76.83**	**72.22**	**75.17**	45.99	**43.80**	**67.61**
	30%	**81.16**	**76.77**	**72.87**	**71.59**	**72.94**	**45.73**	**41.80**	**66.12**
	40%	**80.00**	**73.83**	**65.79**	**70.09**	**70.16**	**43.52**	**40.00**	**63.34**
	50%	72.97	**69.91**	**55.15**	**65.59**	**63.09**	**38.23**	**37.60**	**57.50**

Downstream accuracy of commonsense reasoning tasks. Table 4 and Table 5 summarize the accuracy results of CMQA under downstream task evaluations. Note that Downstream Accuracy is evaluated by using CMQA calibration. We observe that on downstream tasks, GISP consistently achieves higher accuracy across all models and pruning ratios, with particularly strong gains at higher pruning levels, indicating its strength as a task-specific pruner.

Exact-match accuracy of answer generation tasks. While CMQA evaluates multiple-choice reasoning, we further validate GISP on the arithmetic reasoning benchmark GSM8K, which follows a text-generation format but evaluates against the presence of gold answers (marked as Gold ACC). Table 7 compares different pruning methods and calibration datasets under 8-shot evaluation. The same trend holds: task-informed calibration yields significant improvements for GISP, while local pruning (Wanda-sp) gains little, confirming that task-aligned calibration benefits generative reasoning tasks. Notify that conducting pruning on reasoning LLMs is a challenging task for current pruning methods (Zhang et al., 2025; Sui et al., 2025), where current baseline methods usually fail, as Wanda-sp even has zero accuracy at 20% pruning ratio at its default settings.

Table 7: Comparison of different pruning methods on GSM8K (8-shot) accuracy.

Model	Calibration	Method	Ratio	Gold ACC (%)
DeepSeek-R1-Distill-Llama-3-8B	\	Dense	0%	73.54%
	C4	Wanda-sp	20%	0.00%
		GISP	20%	25.25%
			30%	14.33%
			40%	5.46%
	GSM8K	Wanda-sp	20%	29.19%
		GISP	20%	67.93%
			30%	50.80%
			40%	31.84%

4.2 Task-specific Property of GISP

Ablation across seven CMQA tasks. We use CMQA as an example to validate the task-specific property of GISP. We reuse the settings from Sec. 3.2.2. We report detailed results on all tasks across pruning ratios $\{20\%, 30\%, 40\%, 50\%\}$. Table 6 summarizes two consistent trends: (1) **Task-informed calibration helps even with a perplexity target**: replacing C4 with CMQA data under the same perplexity objective yields gains at all ratios, indicating that GISP is an intrinsic task-specific pruner that can actively benefit from task signals from the calibration dataset. (2) **Task-specific loss target brings further improvements**: switching from perplexity to the proposed margin objective (Eq. 6) provides additional, consistent accuracy gains, especially at higher pruning ratios. These trends hold across tasks, supporting GISP as a practical task-specific pruner.

5 Conclusions

In this work, we propose GISP, a simple yet effective global iterative structured pruning method for LLMs. GISP prunes globally and iteratively, enabling more flexible, task-aware pruning. It supports once-for-all pruning across multiple sparsity levels and naturally incorporates loss functions tailored to downstream tasks to guide weight importance. Experiments conducted on Llama2-7B/13B, Llama3-8B, Mistral-0.3, and DeepSeek-R1-Distill-Llama-3-8B demonstrate clear performance gains compared to prior works, excelling as a task-specific pruner, particularly at high sparsity.

since it requires non-trivial, architecture-specific modifications, and these models are not officially supported in their open-sourced code.

Limitations

One limitation of our method is that, due to its reliance on gradient-based weight importance estimation, it can incur relatively high memory and computational costs. To address this, one could integrate parameter-efficient fine-tuning (PEFT) techniques to accelerate importance computations and reduce the memory footprint—a direction we leave for future work. Additionally, while GISP is designed to be architecture-agnostic and shows promising results on multi-query attention (MQA)-based architectures, we have not yet evaluated it on Mixture-of-Experts (MoE) models due to their significantly larger scale. Extending GISP to MoE architectures remains a valuable direction for future exploration.

References

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2023. Fluctuation-based adaptive structured pruning for large language models. *Preprint*, arXiv:2312.11983.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on Artificial Intelligence*.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146.

Han Cai, Chuang Gan, Tianzhe Wang, Zhekai Zhang, and Song Han. 2019. Once-for-all: Train one network and specialize it for efficient deployment. *arXiv preprint arXiv:1908.09791*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Rocktim Jyoti Das, Mingjie Sun, Liqun Ma, and Zhiqiang Shen. 2024. Beyond size: How gradients shape pruning decisions in large language models. *Preprint*, arXiv:2311.04902.

DeepSeek-AI, Daya Guo, Dejian Yang, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Enmao Diao, Ganghua Wang, Jiawei Zhan, Yuhong Yang, Jie Ding, and Vahid Tarokh. 2023. Pruning deep neural networks from a sparsity perspective. *Preprint*, arXiv:2302.05601.

Peijie Dong, Lujun Li, Zhenheng Tang, Xiang Liu, Xinglin Pan, Qiang Wang, and Xiaowen Chu. 2024. Pruner-zero: Evolving symbolic pruning metric from scratch for large language models. *Preprint*, arXiv:2406.02924.

Xin Dong, Shangyu Chen, and Sinno Jialin Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *Preprint*, arXiv:1705.07565.

Jonathan Frankle and Michael Carbin. 2018. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *arXiv preprint arXiv:1803.03635*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. *Preprint*, arXiv:2301.00774.

Aaron Grattafiori, Abhimanyu Dubey, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Preprint*, arXiv:1510.00149.

Babak Hassibi and David Stork. 1992. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 5. Morgan-Kaufmann.

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. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. 2024. Shortened llama: Depth pruning for large language models with comparison of retraining methods. *Preprint*, arXiv:2402.02834.

Eldar Kurtic, Elias Frantar, and Dan Alistarh. 2023. Ziplm: Inference-aware structured pruning of language models. *Preprint*, arXiv:2302.04089.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. *Preprint*, arXiv:1711.05769.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. Shortgpt: Layers in large language models are more redundant than you expect. *Preprint*, arXiv:2403.03853.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

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 *EMNLP*.

Pavlo Molchanov, Arun Mallya, Stephen Tyree, Iuri Frosio, and Jan Kautz. 2019. Importance estimation for neural network pruning. *Preprint*, arXiv:1906.10771.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2016. Pruning convolutional neural networks for resource efficient inference. *arXiv preprint arXiv:1611.06440*.

Pavlo Molchanov, Stephen Tyree, Tero Karras, Timo Aila, and Jan Kautz. 2017. Pruning convolutional neural networks for resource efficient inference. *Preprint*, arXiv:1611.06440.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, and 1 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Winogrande: An adversarial winograd schema challenge at scale. *Preprint*, arXiv:1907.10641.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Preprint*, arXiv:2503.16419.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. 2024. A simple and effective pruning approach for large language models. *Preprint*, arXiv:2306.11695.

Shengkun Tang, Oliver Sieberling, Eldar Kurtic, Zhiqiang Shen, and Dan Alistarh. 2025. Darwinlm: Evolutionary structured pruning of large language models. *Preprint*, arXiv:2502.07780.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

Zhongwei Wan, Xin Wang, Che Liu, Samiul Alam, Yu Zheng, Jiachen Liu, Zhongnan Qu, Shen Yan, Yi Zhu, Quanlu Zhang, Mosharaf Chowdhury, and Mi Zhang. 2024. Efficient large language models: A survey. *Preprint*, arXiv:2312.03863.

Wenxiao Wang, Wei Chen, Yicong Luo, Yongliu Long, Zhengkai Lin, Liye Zhang, Binbin Lin, Deng Cai, and Xiaofei He. 2024. Model compression and efficient inference for large language models: A survey. *Preprint*, arXiv:2402.09748.

Zi Wang, Chengcheng Li, and Xiangyang Wang. 2021. Convolutional neural network pruning with structural redundancy reduction. *Preprint*, arXiv:2104.03438.

BigScience Workshop, :, Teven Le Scao, Angela Fan, and 1 others. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Huanrui Yang, Hongxu Yin, Maying Shen, Pavlo Molchanov, Hai Li, and Jan Kautz. 2023. Global vision transformer pruning with hessian-aware saliency. *Preprint*, arXiv:2110.04869.

Yifan Yang, Kai Zhen, Bhavana Ganesh, Aram Galstyan, Goeric Huybrechts, Markus Müller, Jonas M. Kübler, Rupak Vignesh Swaminathan, Athanasios Mouchtaris, Sravan Babu Bodapati, Nathan Susanj, Zheng Zhang, Jack FitzGerald, and Abhishek Kumar. 2025. Wanda++: Pruning large language models via regional gradients. *Preprint*, arXiv:2503.04992.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. 2025. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *Preprint*, arXiv:2310.05175.

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*.

Mingyang Zhang, Hao Chen, Chunhua Shen, Zhen Yang, Linlin Ou, Xinyi Yu, and Bohan Zhuang. 2024. Loraprune: Structured pruning meets low-rank parameter-efficient fine-tuning. *Preprint*, arXiv:2305.18403.

Nan Zhang, Eugene Kwek, Yusen Zhang, Ngoc-Hieu Nguyen, Prasenjit Mitra, and Rui Zhang. 2025. When reasoning meets compression: Understanding the effects of llms compression on large reasoning models. *Preprint*, arXiv:2504.02010.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, Shengen Yan, Guohao Dai, Xiao-Ping Zhang, Yuhan Dong, and Yu Wang. 2024. A survey on efficient inference for large language models. *Preprint*, arXiv:2404.14294.

A Appendix

A.1 Related Works

Pruning is a fundamental model-compression technique that removes redundant parameters through sparsity. The pioneering OBD work (LeCun et al., 1989) established a Taylor-series framework for importance estimation, followed by extensive CNN successes (Han et al., 2016; Molchanov et al., 2017; Wang et al., 2021). With the rise of large language models, pruning has become crucial for efficient inference (Wan et al., 2024; Wang et al., 2024; Zhou et al., 2024). Because full retraining is prohibitive, recent work shifts to post-training pruning using lightweight calibration data. According to sparsity patterns, methods are either unstructured, removing individual weights but requiring specialized kernels, or structured, pruning entire heads, channels, or layers for hardware-friendly acceleration (Wan et al., 2024; Wang et al., 2024; Ma et al., 2023). For structural pruning, estimating structural importance remains central: early CNN studies proposed summation-based aggregation (Molchanov et al., 2019), and LLM-Pruner (Ma et al., 2023) extended this idea to element-, vector-, and channel-level metrics. Our work follows this line, focusing on post-training structured pruning for LLMs.

A.2 Detailed Explanation, Equations and Algorithm

Oneshot global pruning and GISP procedure. Specifically, given a predefined number of iteration steps n and a target pruning ratio ρ, GISP performs pruning iteratively using a small pruning ratio ρ_k at iteration k. For each iteration, we conduct the one-shot pruning procedure, as detailed in section 3.1. Additionally, to determine the pruning ratio ρ_k at each step, a linear scheduler is employed to linearly increase the pruning ratio until the target ratio ρ is reached in the final iteration. Notably, A key difference from one-shot pruning is that, in GISP, the importance scores in each iteration are computed based on the pruned model from the previous iteration. The detailed procedures are as follows:

① Pruning ratio scheduling. Before pruning starts, we adopt a linear scheduler to obtain a set of target pruning ratios for each iteration step, where the pruning ratios are linearly increased. By doing so, the actual number of pruned weights remains the same for each iteration. Specifically, the linear scheduler can be formulated as:

$$\rho_k = \begin{cases} \rho, & \text{if } n = 1, \\ \frac{k-1}{n-1}\rho, & \text{if } n > 1, \quad k = 1, 2, \ldots, n. \end{cases} \tag{7}$$

where ρ is the overall target ratio, n marks the predefined iteration steps, k is defined as the index of current step.

② Structured weight importance estimation. As illustrated in eq. (3), the importance score for global pruning is derived from the Taylor series expansion of the loss error with respect to weight perturbation caused by pruning. Since higher-order terms are significantly smaller in magnitude compared to the first-order term, we follow prior global pruning methods (Molchanov et al., 2019, 2017) and adopt the first-order approximation as our importance score:

$$I_{W_i^j} = \left| \frac{\partial \mathcal{L}(D)}{\partial W_i^j} W_i^j \right|, \tag{8}$$

Following that, to construct the structured importance score, we aggregate the element-wise importance scores by summing them within each structured group. These groups can correspond to attention heads in attention blocks or channels in the linear layers of feed-forward blocks.

③ Global ranking and pruning. Once the structured importance scores are obtained, we perform a TopK process to select the weights with minimum values according to the pruning ratio of the current iteration step.

Importantly, unlike prior global pruning methods for small models that directly rank importance scores across the entire network, we find that such

an approach is less effective for LLMs due to significant structural differences between attention and MLP blocks. Specifically, attention blocks tend to exhibit substantially higher importance scores than MLP blocks, as illustrated in fig. 3(c), where the scores for attention layers are an order of magnitude greater. To mitigate this imbalance, we normalize the importance scores within attention and MLP blocks separately, ensuring they are brought to a comparable scale across the model.

In the end, the formulation of the normalized importance metrics is given by:

$$I(\theta_{\text{head}}) = \frac{I(\theta_{\text{head}})}{|\theta_{\text{head}}|}, I(\theta_{\text{channel}}) = \frac{I(\theta_{\text{channel}})}{|\theta_{\text{channel}}|} \tag{9}$$

where $|\theta|$ marks the parameter count for the specified structure type. The formulation of the global ranking and the mask generation process can then be formulated as follows:

$$I(\theta_{k-1}) = [I(\theta^{\text{attn}}_{k-1}), I(\theta^{\text{mlp}}_{k-1})], \tau_k = \text{TopK}_{\rho_k}\Big(I(\theta_{k-1})\Big),$$
$$m_k = \begin{cases} 1, & I(\theta_{k-1}) > \tau_k, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

where τ is the threshold for eliminating ongoing pruned modules, m is the binary mask that applies to the model weight. Moreover, in each iteration, the importance score will be evaluated on the pruned model θ_{k-1} to reflect any cascading effect, distinguishing it from one-shot global pruning.

A.3 Detailed Experimental Setup

Experimental settings. The detailed settings are at table 8 and table 9. All baselines will receive the identical calibration dataset for pruning usage in each evaluation task. No re-training or recovery method is used, and only the pruning methods from baselines are evaluated for comparison. In addition, following the settings of Wanda-sp and LLM-pruner, we skip to prune the first 10% of layers and the last layer. All experiments are conducted on a cloud computing server with an AMD EPYC 9554 CPU, 318.6 GB of memory, 400GB SSD, and one Nvidia H100 80GB GPU.

Text generation and zero-shot commonsense reasoning tasks. Following the general setting, we use the C4 dataset as the calibration dataset for text generation tasks and zero-shot commonsense reasoning tasks, with 2000 samples, each having 256 token lengths. No template is used for this task. For the GSM8K task, we use both C4 and GSM8K as calibration and separately evaluate 8-shot accuracy.

Downstream commonsense reasoning tasks. For the downstream commonsense reasoning tasks (CMQA), we use the CMQA training set as the calibration dataset with a total token budget of 512000 (matching previous C4 settings), which is then evenly distributed across each sub-task's training split. To be specific, we include the gold answer (marked as positive labels) and all other options (marked as negative labels) for each sampled question from the training set, forming positive/negative pairs for margin evaluation. For individual tasks, we sample 2000 data points per task and set each task's token-length cap at the 99th percentile of these sampled data. The prompt templates follow the EleutherAI LM Harness pipeline conventions to ensure consistency between calibration and evaluation. We report plain accuracy (acc) for fixed-length tasks (e.g., true/false) and normalized accuracy (acc_norm) for tasks with variable-length answers, thus counteracting cumulative-loss biases on longer sequences.

A.4 Detailed Downstream CMQA Accuracy

We provided detailed downstream task accuracy evaluations at table 10, table 11, table 12, and table 13. We present our key observations of these detailed evaluations as follows.

(1) On downstream tasks, GISP consistently achieves higher accuracy across all models and pruning ratios, with particularly strong gains at higher pruning levels, *indicating its strength as a task-specific pruner*. For example, on the BoolQ task, GISP holds a 6–20% accuracy lead over the best baseline at every pruning ratio. Moreover, at 30–40% pruning ratio, GISP's accuracy remains close to the dense model—for instance, 80.00% (ours) vs. 80.55% (dense) on Llama2-13B at 40% pruning ratio—while other methods begin to lose accuracy even at lower pruning ratios.

(2) For local pruning methods, downstream performance remains similar to—or even lower than—their zero-shot performance, suggesting that while local pruning preserves general knowledge, it lacks task-specific optimization.

A.5 Extended Ablation of Calibration Dataset and Task-specific Loss Target

We provided an extended ablation of using different calibration datasets and the effectiveness of GISP as a task-specific pruner with task-specific loss target at table 14.

(1) GISP is inherently task-specific. When we

Table 8: Detailed settings for CMQA calibration dataset and evaluation.

Task	Token Length	Actual Tokens	Accuracy	Template
BoolQ	410	73 000	acc	`{passage}\nQuestion: {question}\nAnswer:`
PIQA	160	73 125	acc norm	`Question: {goal}\nAnswer:`
Hellaswag	144	73 027	acc norm	`{activity_label}: {ctx_a ctx_b}`
WinoGrande	38	73 117	acc	`{substituted_sentence_at_bottomline}`
ARC-E	92	73 081	acc norm	`Question: {question}\nAnswer:`
ARC-C	112	73 123	acc norm	`Question: {question}\nAnswer:`
OBQA	43	73 140	acc norm	`{question_stem}`

Table 9: Detailed experimental hyper-parameters.

Model	Random Seed	Precision	Pruning Ratio/Iter
Llama2-7B, Llama3-8B, Mistral 0.3-7B	0	bfloat16	0.625%
Llama2-13B	0	bfloat16	0.25%

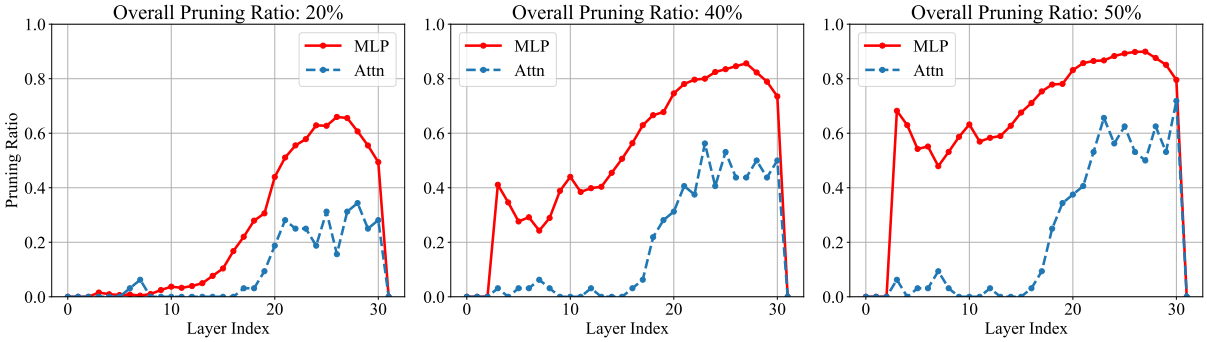

Figure 4: Visualization of the resulting model in various overall pruning ratios from GISP.

switch from C4 to CMQA for calibration, GISP gains significant accuracy improvements at every pruning ratio. In contrast, all baselines show no uplift (and sometimes even regress), reflecting their general-purpose properties with no sensitivity to task information and highlighting GISP's task-specific capability.

(2) Effectiveness of GISP as a task-specific pruner with task-specific loss target. Incorporating the margin-based loss provides consistent accuracy gains across nearly every task and pruning ratio, showing the necessity and effectiveness of GISP's design to accommodate various task-specific loss targets.

A.6 Visualization of the pruned model

Figure 4 provides the layer-wise pruning ratio distribution of various target pruning ratios of GISP on attention blocks and MLP blocks of the Llama2-7B model, respectively. We present our key observations of the generated pruned model as follows, aiming to provide insight for further works, such as LLMs architecture searching, design, and explanation:

(1) Layer-wise sparsity varies significantly for both attention and MLP components. Both the MLP and attention layers exhibit a similar trend of increasing pruning ratios from early to late layers, suggesting that earlier layers are more critical to model performance than later ones.

(2) MLP layers are more redundant than attention layers. First, the pruning ratio of MLP layers is consistently higher than that of attention layers across all layers. Additionally, we observe that early attention layers are particularly important—under a 50% overall pruning ratio, approximately the first half of the attention layers maintain very low sparsity. In contrast, between the 40% and 50% pruning ratio, more MLP channels are pruned in the early layers, while the attention layers in the same range remain largely intact.

A.7 Efficiency Analysis

Pruning procedure efficiency analysis. The runtime overhead of deploying GISP is discussed in Section 3.2.2 and summarized in Table 3.

We additionally report the peak GPU memory during pruning on Llama2-7B under the same en-

Table 10: Llama 2-7B Downstream CMQA Accuracy under Different Pruning Methods.

Method	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
Dense	0%	77.71	79.05	76.00	68.98	74.54	46.25	44.20	66.68
Wanda-sp	20%	65.84	78.73	71.07	62.75	69.32	43.09	41.60	61.77
	30%	62.35	75.73	63.44	57.85	63.30	38.48	38.80	57.14
	40%	61.38	73.39	44.69	50.91	53.87	30.38	36.20	50.12
	50%	58.17	65.29	34.60	50.75	40.82	24.57	30.40	43.52
LLM-Pruner	20%	64.37	71.44	48.49	57.85	55.85	28.24	30.40	50.95
	30%	60.31	60.94	31.31	50.67	38.80	20.73	26.80	41.37
	40%	60.55	55.17	28.71	49.41	33.12	20.14	26.80	39.13
	50%	60.92	53.70	28.07	50.12	31.57	20.73	25.20	38.62
FLAP	20%	67.16	77.48	70.64	62.35	66.54	42.49	42.20	61.27
	30%	62.87	75.24	63.47	57.85	61.32	38.74	38.80	56.90
	40%	61.65	72.03	53.86	54.22	55.35	36.95	37.00	53.01
	50%	59.45	68.28	42.58	53.12	48.53	30.72	32.20	47.84
ShortGPT	20%	62.17	70.18	62.73	65.82	55.93	36.18	37.20	55.75
	30%	62.20	63.38	50.80	62.98	45.08	34.22	31.40	50.01
	40%	62.17	57.83	41.16	58.09	37.08	30.12	31.00	45.35
	50%	62.17	52.61	33.35	56.91	31.73	26.71	28.80	41.75
OWL	20%	67.09	78.67	71.87	66.14	69.87	43.43	41.40	62.64
	30%	64.04	76.66	66.54	58.33	64.44	38.91	39.40	58.33
	40%	62.14	74.05	47.62	52.49	55.01	30.80	38.40	51.50
	50%	61.25	66.16	35.41	51.62	44.15	24.91	30.20	44.82
GISP (ours)	20%	77.77	75.30	70.71	69.14	69.65	41.64	40.00	**63.46**
	30%	77.19	72.85	64.19	65.35	65.24	40.53	39.40	**60.68**
	40%	70.55	68.88	53.68	62.51	57.74	35.41	38.20	**55.28**
	50%	65.29	64.09	41.27	56.27	49.71	29.18	34.00	**48.54**

Table 11: Llama 3-8B Downstream CMQA Accuracy under Different Pruning Methods.

Method	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
Dense	0%	81.28	80.79	79.13	72.61	77.69	53.41	45.00	69.99
Wanda-sp	20%	59.42	77.31	58.77	59.67	67.68	39.68	39.60	57.45
	30%	61.28	74.43	46.62	54.46	57.66	32.94	36.80	52.03
	40%	62.17	63.93	33.15	52.09	42.93	23.98	27.00	43.61
	50%	58.75	60.88	31.87	50.67	37.84	23.04	26.20	41.32
LLM-Pruner	20%	67.68	75.03	57.76	60.77	61.28	37.03	36.00	56.51
	30%	60.86	66.38	41.35	54.54	51.35	27.13	30.60	47.46
	40%	57.31	60.61	33.68	51.46	39.94	22.53	27.20	41.82
	50%	52.63	56.37	31.08	50.43	36.07	22.53	28.60	39.67
ShortGPT	20%	65.02	71.00	64.61	70.88	56.65	42.41	33.20	57.68
	30%	51.68	60.72	33.44	58.48	39.27	30.55	30.60	43.53
	40%	58.62	60.45	37.76	52.96	35.23	29.95	28.60	43.37
	50%	60.86	55.39	29.38	54.54	29.71	28.84	29.60	41.19
OWL	20%	64.74	77.97	61.82	62.83	70.58	41.13	40.60	59.95
	30%	62.84	73.29	47.51	55.88	56.90	32.25	37.00	52.24
	40%	62.11	65.18	34.24	52.09	45.58	24.66	30.20	44.87
	50%	60.09	60.12	31.37	52.72	38.76	22.95	27.00	41.86
GISP (ours)	20%	79.11	76.50	70.16	71.43	70.66	47.10	42.00	**65.28**
	30%	78.04	71.00	59.24	69.93	62.46	40.53	36.40	**59.66**
	40%	72.69	67.25	47.58	66.14	54.59	34.13	32.20	**53.51**
	50%	66.48	62.19	36.07	55.72	42.34	27.99	29.00	**45.68**

Table 12: Mistral 0.3-7B Downstream CMQA Accuracy under Different Pruning Methods.

Method	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
Dense	0%	82.08	82.21	80.44	73.88	78.24	52.22	44.20	70.47
Wanda-sp	20%	68.72	80.52	72.73	66.77	74.79	44.03	43.20	64.39
	30%	57.92	78.89	63.17	58.56	68.81	37.54	42.80	58.24
	40%	53.39	76.22	50.92	55.88	58.21	31.40	37.20	51.89
	50%	58.04	67.03	36.53	49.96	45.66	24.23	29.20	44.38
ShortGPT	20%	69.36	72.31	64.71	68.51	58.63	39.16	31.60	57.75
	30%	42.29	58.60	34.54	57.70	31.27	31.91	31.40	41.10
	40%	53.00	53.10	26.40	55.80	30.47	30.80	28.20	39.68
	50%	52.97	50.16	24.96	53.75	30.13	30.72	28.40	38.73
OWL	20%	68.65	80.69	74.52	70.01	75.76	46.25	45.20	65.87
	30%	52.60	79.05	66.04	61.25	69.40	39.25	42.20	58.54
	40%	57.06	78.07	53.76	55.56	62.79	33.87	39.40	54.36
	50%	61.19	69.70	38.66	50.99	46.38	26.11	29.60	46.09
GISP (ours)	20%	80.52	78.40	73.55	73.16	74.54	47.01	39.00	**66.60**
	30%	79.79	76.06	65.69	70.01	71.68	41.55	39.60	**63.48**
	40%	77.00	71.71	54.57	65.04	64.60	37.97	37.20	**58.30**
	50%	70.21	62.79	40.36	56.75	55.26	32.17	31.00	**49.79**

Table 13: Llama 2-13B Downstream CMQA Accuracy under Different Pruning Methods.

Method	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
Dense	0%	80.55	80.52	79.39	72.22	77.48	48.98	45.20	69.19
Wanda-sp	20%	73.09	79.98	76.04	67.09	75.46	47.87	44.60	66.30
	30%	70.83	78.94	68.68	62.83	71.30	44.03	42.20	62.69
	40%	64.16	78.02	62.27	59.27	67.26	41.21	41.60	59.11
	50%	62.14	71.98	48.91	53.35	55.13	32.68	37.00	51.60
LLM-Pruner	20%	66.51	74.92	58.47	62.67	66.41	35.92	36.20	57.30
	30%	61.77	67.19	36.64	51.54	51.01	25.26	30.40	46.26
	40%	58.93	61.43	31.23	49.25	42.51	22.78	27.80	41.99
	50%	61.96	58.11	28.81	51.70	36.78	22.27	26.80	40.92
FLAP	20%	73.00	79.92	72.26	66.85	73.74	45.82	43.40	65.00
	30%	69.94	78.35	67.79	65.11	71.72	44.88	45.20	63.28
	40%	66.21	76.82	64.74	63.14	65.53	42.58	40.20	59.89
	50%	64.16	74.48	57.94	57.77	60.82	38.65	40.20	56.29
ShortGPT	20%	61.80	74.16	70.62	70.17	65.87	42.49	40.80	60.84
	30%	61.53	69.80	64.57	69.53	55.47	39.33	37.80	56.86
	40%	44.98	65.02	33.19	51.66	66.46	46.17	33.60	48.73
	50%	62.17	52.61	33.35	56.91	31.73	26.71	28.80	41.75
OWL	20%	76.48	80.30	77.19	68.35	74.71	46.84	45.00	66.98
	30%	69.82	78.40	73.28	64.09	71.76	43.17	42.40	63.27
	40%	69.97	77.69	66.97	60.62	67.51	42.92	40.20	60.84
	50%	63.46	73.18	55.15	54.78	57.58	37.29	37.80	54.17
GISP (ours)	20%	80.28	79.00	76.83	72.22	75.17	45.99	43.80	**67.61**
	30%	81.16	76.77	72.87	71.59	72.94	45.73	41.80	**66.12**
	40%	80.00	73.83	65.79	70.09	70.16	43.52	40.00	**63.34**
	50%	72.97	69.91	55.15	65.59	63.09	38.23	37.60	**57.50**

Table 14: CMQA Accuracy on All Seven Tasks under Different Calibration Datasets and Pruning Ratios.

Method	Calibration Dataset	Pruning Ratio	BoolQ	PIQA	Hellaswag	WinoGrande	ARC-E	ARC-C	OBQA	AVG
Wanda-sp	C4	20%	74.07	79.33	77.94	70.48	72.77	45.14	44.80	66.36
		30%	67.86	78.02	74.65	68.11	69.57	43.52	44.20	63.70
		40%	64.40	77.31	68.36	61.40	57.41	37.97	40.80	58.24
		50%	62.63	72.31	58.69	55.96	48.32	31.23	36.60	52.25
	CMQA	20%	73.09	79.98	76.04	67.09	75.46	47.87	44.60	66.30
		30%	70.83	78.94	68.68	62.83	71.30	44.03	42.20	62.69
		40%	64.16	78.02	62.27	59.27	67.26	41.21	41.60	59.11
		50%	62.14	71.98	48.91	53.35	55.13	32.68	37.00	51.60
LLM-Pruner	C4	20%	71.68	79.54	74.95	67.48	74.33	46.84	44.60	65.63
		30%	66.97	79.16	70.58	65.04	67.47	42.66	41.00	61.84
		40%	62.78	75.46	60.77	58.33	56.23	33.62	38.20	55.06
		50%	62.02	70.51	49.34	53.75	43.52	27.99	33.80	48.70
	CMQA	20%	66.51	74.92	58.47	62.67	66.41	35.92	36.20	57.30
		30%	61.77	67.19	36.64	51.54	51.01	25.26	30.40	46.26
		40%	58.93	61.43	31.23	49.25	42.51	22.78	27.80	41.99
		50%	61.96	58.11	28.81	51.70	36.78	22.27	26.80	40.92
FLAP	C4	20%	70.89	80.20	77.62	70.80	72.18	46.59	44.80	66.15
		30%	70.37	79.43	75.05	68.43	68.27	44.11	43.80	64.21
		40%	67.00	77.20	70.60	67.09	65.74	43.09	41.40	61.73
		50%	62.75	73.56	63.09	62.27	57.53	39.42	37.60	56.60
	CMQA	20%	73.00	79.92	72.26	66.85	73.74	45.82	43.40	65.00
		30%	69.94	78.35	67.79	65.11	71.72	44.88	45.20	63.28
		40%	66.21	76.82	64.74	63.14	65.53	42.58	40.20	59.89
		50%	64.16	74.48	57.94	57.77	60.82	38.65	40.20	56.29
ShortGPT	C4	20%	61.80	74.16	70.62	70.17	65.87	42.49	40.80	60.84
		30%	37.77	69.75	57.88	69.30	52.90	35.84	38.40	51.69
		40%	62.20	61.81	47.23	62.51	44.87	31.83	35.60	49.43
		50%	62.20	58.43	40.87	61.40	37.21	31.57	30.80	46.07
	CMQA	20%	61.80	74.16	70.62	70.17	65.87	42.49	40.80	60.84
		30%	61.53	69.80	64.57	69.53	55.47	39.33	37.80	56.86
		40%	44.98	65.02	33.19	51.66	66.46	46.17	33.60	48.73
		50%	62.17	52.61	33.35	56.91	31.73	26.71	28.80	41.75
OWL	C4	20%	75.44	79.22	77.79	71.82	71.80	45.05	45.20	66.62
		30%	69.91	78.84	75.36	68.19	68.43	43.34	42.80	63.84
		40%	66.39	76.77	69.58	63.38	62.58	39.76	40.60	59.87
		50%	63.49	72.69	59.25	58.48	49.20	31.83	38.80	53.39
	CMQA	20%	76.48	80.30	77.19	68.35	74.71	46.84	45.00	66.98
		30%	69.82	78.40	73.28	64.09	71.76	43.17	42.40	63.27
		40%	69.97	77.69	66.97	60.62	67.51	42.92	40.20	60.84
		50%	63.46	73.18	55.15	54.78	57.58	37.29	37.80	54.17
GISP	C4 + Perplexity	20%	73.30	78.45	73.09	68.11	67.59	42.92	42.00	63.64
		30%	69.20	76.71	69.68	65.98	62.67	37.46	40.80	60.36
		40%	65.14	73.67	62.80	61.64	54.55	33.87	36.80	55.49
		50%	58.41	68.28	50.79	58.64	44.02	28.41	32.20	48.68
	CMQA + Perplexity	20%	80.80	77.86	76.39	71.82	74.75	46.33	42.20	67.16
		30%	80.83	75.52	71.91	69.69	72.22	44.71	41.60	65.21
		40%	79.54	72.52	63.36	67.88	67.85	41.81	39.20	61.74
		50%	76.18	67.52	51.67	61.56	59.47	36.77	36.40	55.65
	CMQA + Margin	20%	80.28	79.00	76.83	72.22	75.17	45.99	43.80	67.61
		30%	81.16	76.77	72.87	71.59	72.94	45.73	41.80	66.12
		40%	80.00	73.83	65.79	70.09	70.16	43.52	40.00	63.34
		50%	72.97	69.91	55.15	65.59	63.09	38.23	37.60	57.50

Table 15: Peak GPU memory during pruning on Llama2-7B.

Method	Peak memory (MiB)
GISP	43576
Wanda	22128
FLAP	21656
OWL	26528

Table 16: Inference efficiency of pruned subnetworks.

Pruning Ratio	TTFT (s)	Prefill (t/s)	Decode (t/s)	Prefill Inc.	Decode Inc.
Dense	0.105	5635.4	38.0	1.00x	1.00x
20%	0.090	6833.6	41.4	1.21x	1.09x
30%	0.082	7704.7	43.2	1.37x	1.14x
40%	0.076	8396.9	43.7	1.49x	1.15x
50%	0.069	9562.7	43.7	1.70x	1.15x

vironment (batch size and calibration setting) in Table 15. As expected and claimed in limitation section, GISP uses higher peak memory than local forward-only baselines, as it requires a backward pass for gradient computation.

Pruned model efficiency analysis. We evaluate the inference throughput and memory footprint of the pruned Llama2-7B model on an NVIDIA A6000 GPU with batch size $= 1$ and report time-to-first-token (TTFT), prefill/decode throughput, related speedup and memory cost in Table 16 and 17. GISP constantly reduces latency, improves both decoding and prefilling speedup and reduce memory footprint compared with the dense model.

A.8 Practical Impact of the GISP: Model saving and on-the-fly adaptation

Thanks to the property of the once-for-all pruning, GISP can produce a spectrum of models pruned to different pruning ratios to a target ratio ρ. To exploit this without extra storage overhead, we record only the indices of channels or heads removed at each iteration—orders of magnitude smaller than element-wise masks. Once pruning is complete, any intermediate pruned model can be reconstructed simply by reapplying the saved indices. For example, after apply GISP on Llama2-7B model, the user only need to store one set of weights with the corresponding masks (approximately 318KB per mask, which is 0.002% of the model weight for Llama2-7b), rather than maintaining numerous separate models.

This enables on-the-fly adaptation: by running GISP as a preprocessing step to capture the pruning schedule, users can dynamically deploy the most suitable pruned model according to each of the various computing resources and dynamic environments.

Table 17: Inference memory footprint.

Sparsity	Memory (GB)	Reduction (%)
Dense	14.05	–
20%	11.58	17.6
30%	10.27	26.9
40%	9.07	35.4
50%	7.73	45.0

Table 18: Performance of pruned model under 60% pruning ratio on Llama2-7B.

Method	Pruning ratio	0shot CMQA ACC (%)
Dense	0%	66.68
GISP	60%	42.17
Wanda	60%	37.77
FLAP	60%	37.98
OWL	60%	37.81

A.9 GISP Under Higher Pruning Ratio

To further demonstrate the performance advantage of GISP in higher sparsity, we add an additional pruning experiment on Llama2-7B, targeting 60% sparsity, for reference. The results in Table 18 show the expected trend: sharp performance drops for all baseline methods, while GISP remains competitive and degrades more gracefully.

A.10 LLM Usage

In accordance with the AAR AI Writing/Coding Assistance Policy, we disclose that LLM-based tools (e.g., ChatGPT) were used solely to aid in polishing the writing and improving the clarity of exposition. They were not used for research ideation, experimental design, data analysis, or other substantive contributions. All scientific content, results, and conclusions are the responsibility of the authors.