REVEALING LOCAL NONLINEARITY AMPLIFICATION FOR LARGE LANGUAGE MODEL COMPRESSION

Anonymous authors
Paper under double-blind review

ABSTRACT

The growing computational cost of large language models (LLMs) has driven increasing interest in structured compression to reduce inference overhead. Existing structured compression methods primarily assess layer importance or redundancy, but rarely characterize how errors introduced by compression propagate through subsequent computation. We observe a consistent layer-wise pattern in LLMs: across inference activation distributions, layers exhibit distinct levels of local nonlinear deviation, with many being well approximated by affine mappings. Meanwhile, the response of downstream computation to layer outputs exhibits stable distributional patterns across layer positions. We therefore introduce *Local Nonlinearity Amplification* (LNA), a quantitative measure that combines a layer's local nonlinear deviation with the downstream response to its output change, providing a direct estimation of the degradation risk associated with layer compression. Guided by LNA-based layer selection, we propose ReLoNA, which adaptively determines an affine-operator replacement strategy based on each layer's local nonlinearity. After layer replacement, ReLoNA applies Operator-Aligned Recovery to reduce the approximation errors introduced by replacement and maintain consistency with the original computation. Across 18 benchmarks and multiple models, ReLoNA outperforms existing structured compression methods at 25% compression ratios, retaining **94%** of dense-model performance on multiple-choice benchmarks and achieving better performance on generative benchmarks, with gains of up to **2**$\times$ over existing methods. Moreover, LNA-guided pruning consistently yields substantially lower degradation than existing criteria under the same compression ratio.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated remarkable capabilities in a wide range of natural language understanding and generation tasks (Singh et al., 2026; Grattafiori et al., 2024). However, the rapidly increasing model scale and depth introduce substantial computational overhead, memory consumption, and inference latency, limiting their deployment in practical scenarios (Pope et al., 2023; Bian et al., 2025). Therefore, reducing inference cost while preserving model capability has become a critical research direction for efficient LLM deployment (Chavan et al., 2024). Recent studies have explored various model compression techniques for reducing the computational cost of LLMs (Zhu et al., 2024), including weight pruning (Han et al., 2015; Frantar & Alistarh, 2023; Sun et al., 2024), quantization (Xiao et al., 2023; Frantar et al., 2023), and knowledge distillation (Hinton et al., 2015b; Liang et al., 2023; Wu et al., 2023). However, these approaches either rely on specialized sparse or low-precision computation support or incur substantial training costs, which can limit their practical deployment.

Structured compression (Wang et al., 2020; Xia et al., 2022) directly reduces the computational structure of Transformer models by removing or simplifying structured components, enabling inference acceleration while preserving regular dense architectures. Existing structured compression methods generally identify compressible structures based on representation similarity (Men et al., 2025; Yang et al., 2024; Shopkhoev et al., 2025), parameter statistics (Ashkboos et al., 2024), or heuristic importance scores (Ma et al., 2023). However, these approaches primarily characterize the intrinsic properties of individual structural components and do not explicitly account for how the errors introduced by structural modifications propagate through subsequent transformations and

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

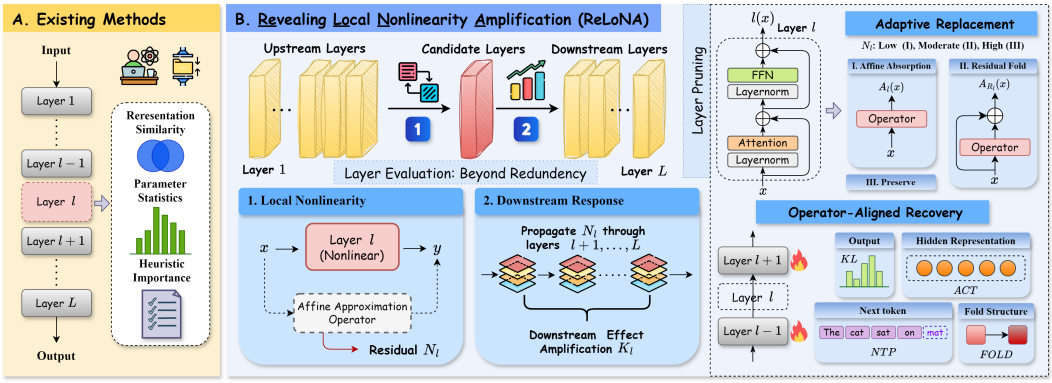


Figure 1: Existing structured compression methods and our ReLoNA (**Re**vealing **Lo**cal **N**onlinearity **A**mplification) for LLM layer compression.

affect final model outputs. In practice, removing or replacing a structural component can induce a representation shift that is further processed by downstream computations and may be amplified. This motivates considering both the local effects of structural modifications and their propagation through subsequent computations when assessing structural compressibility.

To investigate this problem, we analyze Transformer layers from the perspective of local functional behavior and compression-induced perturbations in Sec. 2. Our analysis reveals that the degradation induced by layer compression can be effectively estimated. First, many layers exhibit near-affine behavior around inference-time activations despite their nonlinear architectures, making local nonlinear deviation quantifiable. Second, downstream responses to the resulting errors vary across layer positions but remain relatively stable, allowing the error propagation amplification factor to be estimated. These findings lead to *Local Nonlinearity Amplification* (LNA), which jointly captures local nonlinear deviation and downstream response to quantify the compression risk of individual layers.

Based on this perspective, we propose **ReLoNA** for large language model compression (Figure 1). Unlike existing structured compression methods, ReLoNA formulates layer compression as a local operator replacement problem and uses LNA to identify suitable compression candidates. ReLoNA further adapts the replacement strategy to the local nonlinear behavior of each selected layer, using affine absorption, residual folding, or layer preservation as appropriate. To mitigate the representation shifts introduced by operator replacement, we also propose Operator-Aligned Recovery (OAR), which focuses recovery on the components directly involved in compression and preserves the resulting compressed architecture.

Extensive experiments show that ReLoNA consistently achieves leading performance across models. At 25% compression, it retains around 95% of the original performance on multiple-choice benchmarks and outperforms existing methods on generation tasks. Under fixed pruning budgets, its LNA-guided pruning and adaptive replacement yield substantially lower degradation than existing state-of-the-art methods. We further verify the positive correlation between LNA-estimated risk and actual degradation, while budget analysis, cross-dataset and cross-model evaluations, and ablations demonstrate the stability of ReLoNA.

The main contributions of this work are summarized as follows:

- **Local Nonlinearity Amplification as a new perspective.** We reveal the local affine characteristics of Transformer layers on real activation manifolds and the position-dependent response of compression perturbations in downstream networks, establishing a unified view of degradation in layer compression.

- **ReLoNA for LLM layer compression.** We introduce LNA to jointly characterize local approximation difficulty and downstream perturbation response, and develop an adaptive layer collapse strategy that selects appropriate replacement operators according to layer-wise local nonlinearity.

- **Operator-aligned recovery and comprehensive evaluation.** We propose OAR to recover compressed models within a localized parameter space aligned with operator replacement. Extensive experiments across benchmarks, compression ratios, model architectures, and ablation settings validate the effectiveness, predictive capability, and robustness of ReLoNA.

2 From Local Nonlinearity to Layer Compression

2.1 Problem Setup

Existing layer pruning methods typically select layers based on intrinsic properties, but these criteria do not directly address the question in layer compression: *how much degradation is expected when a layer is replaced by a simpler operator?* We therefore formulate layer compression as an operator-replacement problem and aim to select compressible layers by minimizing estimated degradation.

Consider a decoder-only Transformer (Vaswani et al., 2017) with L sequential layers. Let F_i denote the transformation implemented by layer i, and let $x_i \in \mathbb{R}^d$ be its input, such that $x_{i+1} = F_i(x_i)$. For a candidate layer B, let $\mathcal{D}_B$ denote the input activation distribution induced by a calibration set, and let G_B denote the subnetwork following B. The original and compressed downstream mappings from layer B to the model output can then be written as

$$F = G_B \circ B, \qquad \widetilde{F} = G_B \circ \widetilde{B}, \tag{1}$$

where $\widetilde{B}$ is a computationally simpler replacement operator. The local replacement error and the resulting output degradation are given by

$$e_B(x) = B(x) - \widetilde{B}(x), \qquad \Delta_B(x) = \|F(x) - \widetilde{F}(x)\|_2, \qquad x \sim \mathcal{D}_B. \tag{2}$$

Since Δ_B requires evaluating the compressed network and is unavailable before compression, we introduce the *estimated degradation* $\widehat{\Delta}_B$ as a proxy for the degradation induced by compressing layer B. Given a Transformer with L layers, let $\mathcal{S} \subseteq 1, \ldots, L$ denote the set of candidate layers for compression, and let K denote the target number of layers to compress. We aim to select K layers from $\mathcal{S}$ such that their cumulative estimated degradation is minimized:

$$\mathcal{S}^* = \arg \min_{\mathcal{S}' \subseteq \mathcal{S}\ |\mathcal{S}'|=K} \sum_{B \in \mathcal{S}'} \widehat{\Delta}_B, \tag{3}$$

where $\mathcal{S}^*$ denotes the selected set of K layers for compression. The key challenge is therefore to construct a measurable estimate $\widehat{\Delta}_B$ from the local replacement error e_B and the downstream response of G_B.

2.2 Local Nonlinearity of Transformer Layers

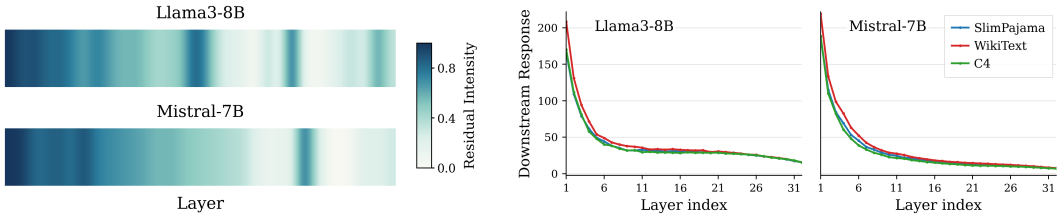
The local replacement error e_B in Eq. (2) depends on how closely a layer can be approximated on its inference-time activation distribution, which occupies only a restricted subset of its input space despite the layer's nonlinear architecture. We assess functional complexity locally, on $\mathcal{D}_B$, rather than from architecture alone. Let $\mathcal{A} = \{x \mapsto Wx + b\}$ be the class of affine operators and define the calibration-optimal affine surrogate

$$A_B \in \arg\min_{A \in \mathcal{A}} \mathbb{E}_{x \sim \mathcal{D}_B} \|B(x) - A(x)\|_2^2. \tag{4}$$

Given calibration pairs $\{(x_j, B(x_j))\}_{j=1}^N$ with $\sum_{j=1}^N \|B(x_j)\|_2 > 0$, we define the normalized local nonlinearity as

$$\widetilde{\mathcal{N}}(B) = \frac{\frac{1}{N}\sum_{j=1}^N \|B(x_j) - A_B(x_j)\|_2}{\frac{1}{N}\sum_{j=1}^N \|B(x_j)\|_2}. \tag{5}$$

The denominator makes the residual comparable across layers with different representation scales. Small $\widetilde{\mathcal{N}}(B)$ indicates that B is well approximated by an affine operator on calibration activations. As shown in Figure 2a, many intermediate and deeper layers exhibit small residuals, making affine substitution plausible.



(a) Local Nonlinearity. (b) Downstream Response.

Figure 2: **Observations motivating Local Nonlinearity Amplification.**

Proposition 1 (optimality of the local affine projection). Let A_B be any minimizer of Eq. (4), and define the residual $r_B(x) = B(x) - A_B(x)$. Then, for every affine perturbation $H \in \mathcal{A}$,

$$\mathbb{E}_{x\sim\mathcal{D}_B}\langle r_B(x), H(x)\rangle = 0, \qquad \mathbb{E}\|B - A_B - H\|_2^2 = \mathbb{E}\|r_B\|_2^2 + \mathbb{E}\|H\|_2^2. \tag{6}$$

Consequently, A_B is the minimum-error affine surrogate in $L_2(\mathcal{D}_B)$, and any affine modification with nonzero $L_2(\mathcal{D}_B)$ norm strictly increases the mean-squared local error. Proofs and derivations of the propositions are provided in Appendix A.1. The normalized residual in Eq. (5) measures the scale-normalized deviation of the best affine substitute. Yet, its impact depends on the layer position, as identical residuals can lead to different output discrepancies.

2.3 Downstream Response to Layer Perturbations

We next characterize how the local deviation propagates through the downstream network G_B in Eq. (1). A global Lipschitz constant, $L(G_B) = \sup_{h\neq h'} \|G_B(h) - G_B(h')\|_2/\|h - h'\|_2$, is both difficult to estimate and potentially dominated by regions never visited at inference (Huang et al., 2021). We instead estimate local response around calibration representations. Let $h = B(x)$, with δ denoting a small perturbation, and define

$$K_B(h, \delta) = \frac{\|G_B(h + \delta) - G_B(h)\|_2}{\|\delta\|_2}. \tag{7}$$

For sampled representations $\{h_j\}_{j=1}^N$ and perturbation directions, we set $\sigma = 0.01 \times \frac{1}{N}\sum_{j=1}^N \|h_j\|_2$ and use perturbations of radius σ. The empirical response estimate $\widehat{K}_B$ is the median of the resulting ratios, which reduces sensitivity to atypical directions and activations. Figure 2b shows that these response strengths vary substantially with layer position while remaining relatively stable across calibration corpora.

Proposition 2 (finite-difference interpretation). Let $J_{G_B}(h)$ denote the Jacobian of G_B at h. If G_B is differentiable at h, then for any unit direction u,

$$K_B(h, \sigma u) = \|J_{G_B}(h)u\|_2 + o(1), \qquad \sigma \to 0. \tag{8}$$

Thus, finite-difference response converges to the directional Jacobian response in the small-perturbation limit, providing a data-dependent estimate of local downstream sensitivity without explicitly computing the Jacobian or invoking a worst-case global Lipschitz constant.

3 Methods

3.1 Local Nonlinearity Amplification

The preceding analysis motivates *Local Nonlinearity Amplification* (LNA), the product of the two empirical quantities:

$$\mathrm{LNA}(B) = \widehat{K}_B\widetilde{\mathcal{N}}(B). \tag{9}$$

LNA provides an estimate of compression risk by characterizing the factors governing output degradation, rather than directly predicting model loss. The local nonlinearity $\widetilde{\mathcal{N}}(B)$ (in Sec. 2.2)defined above measures the relative deviation created by affine replacement, and $\widehat{K}_B$ (in Sec. 2.3) measures the typical downstream amplification of a local perturbation. We further discuss the relationship between inter-layer similarity and LNA in Appendix A.2.

Proposition 3 (local error propagation). For every $x \in \mathrm{supp}(\mathcal{D}_B)$, define the segment $\gamma_x(t) = \widetilde{B}(x) + t(B(x) - \widetilde{B}(x)), t \in [0,1]$. If G_B is differentiable on this segment and

$$\kappa_B(x) := \sup_{t\in[0,1]} \|J_{G_B}(\gamma_x(t))\|_{2\to 2}, \tag{10}$$

then, for $F = G_B \circ B$ and $\widetilde{F} = G_B \circ \widetilde{B}$,

$$\|F(x) - \widetilde{F}(x)\|_2 \le \kappa_B(x)\|e_B(x)\|_2. \tag{11}$$

Taking expectation and setting $\widetilde{B} = A_B$ and letting $s_B = \mathbb{E}_{x\sim\mathcal{D}_B}\|B(x)\|_2$ with $\kappa_B(x) \le \bar{\kappa}_B$, then

$$\frac{\mathbb{E}_{x\sim\mathcal{D}_B}\|F(x) - G_B(A_B(x))\|_2}{s_B} \le \bar{\kappa}_B \widetilde{\mathcal{N}}(B). \tag{12}$$

Thus, the normalized consequence of affine substitution is controlled by the product of downstream response and normalized local residual. In practice, $\bar{\kappa}_B$ is inaccessible, so LNA substitutes the robust finite-difference estimate $\widehat{K}_B$. Appendix D.2 supporting this substitution.

3.2 Layer Pruning with Adaptive Replacement

We first rank layers according to their LNA values defined in Eq. (9) and select candidate layers under a target pruning ratio or prescribed degradation budget. Let $\mathcal{C}$ denote the set of layers selected for compression, and its cumulative degradation risk estimate is defined as

$$S_{\mathrm{cum}}(\mathcal{C}) = \sum_{B\in\mathcal{C}} \mathrm{LNA}(B). \tag{13}$$

Under the first-order additive approximation, S_{cum} aggregates the local nonlinearity amplification contributions of selected layers and provides a unified criterion for constructing the compressed layer set. Given an allowable total degradation risk budget ϵ, the maximum achievable pruning rate is defined as

$$p^*(\epsilon) = \max_{\mathcal{C}\subseteq\mathcal{L}_{\mathrm{cand}}} \frac{|\mathcal{C}|}{L} \quad \text{s.t.} \quad S_{\mathrm{cum}}(\mathcal{C}) \le \epsilon, \tag{14}$$

where $\mathcal{L}_{\mathrm{cand}}$ denotes the set of eligible candidate layers and L is the total number of layers. Since Transformer layers generally share the same architectural scale, the computational benefit obtained from removing a single layer is approximately consistent. We prioritize layers with lower local nonlinearity amplification for compression. LNA determines *whether* a layer is a low-risk compression candidate; its local nonlinearity determines *how* to replace that layer. The fitted operator is then folded into the attention projections of the next retained layer without introducing extra modules or altering the residual topology. We use the following adaptive replacement policy.

Affine absorption. For layers with low local nonlinearity, we use $\widetilde{B}_{\mathrm{aff}} = A_B$, with A_B defined by Eq. (4). The fitted affine operator is absorbed into the input transformation of the next retained layer, and the original layer is removed.

Residual folding. For intermediate local nonlinearity, approximation of the complete mapping can create a larger representation shift. Using the residual form $B(x) = x + R_B(x)$, we instead fit

$$A_{R_B} \in \arg\min_{A\in\mathcal{A}} \mathbb{E}_{x\sim\mathcal{D}_B}\|R_B(x) - A(x)\|_2^2, \quad R_B(x) = B(x) - x, \tag{15}$$

and replace the layer by $\widetilde{B}_{\mathrm{res}}(x) = x + \alpha A_{R_B}(x)$, where $\alpha \in [0,1]$ controls the magnitude of the folded residual and is determined by least-squares scaling.

Keep Policy. For layers with high normalized local nonlinearity, we retain the original structure and do not perform operator folding.

3.3 Operator-Aligned Recovery

Layer compression replaces original transformations with simpler operators, introducing local representation shifts that propagate to the model output. Existing recovery strategies, such as full-model

fine-tuning (Han et al., 2015) and generic knowledge distillation (Hinton et al., 2015a), may globally compensate for local compression deviation, potentially altering the compressed structure.

We therefore propose **Operator-Aligned Recovery (OAR)**, which restricts optimization to two operator-aligned parameter tiers: **Tier-1**, the target layers receiving folded operators, and **Tier-2**, the boundary layers adjacent to compressed regions. By aligning recovery with the deviation introduction, OAR reduces residual perturbations while preserving the compressed structure. The overall objective is defined as:

$$\mathcal{L} = \lambda_{\mathrm{KL}}\mathcal{L}_{\mathrm{KL}} + \lambda_{\mathrm{ACT}}\mathcal{L}_{\mathrm{ACT}} + \lambda_{\mathrm{NTP}}\mathcal{L}_{\mathrm{NTP}} + \lambda_{\mathrm{FOLD}}\mathcal{L}_{\mathrm{FOLD}}. \tag{16}$$

Here $\mathcal{L}_{\mathrm{KL}} = \mathrm{KL}_{\tau}\big(p^T \,\|\, p^S\big)$ is the temperature-scaled teacher–student KL with $\tau = 2$, $\mathcal{L}_{\mathrm{NTP}}$ is the standard next-token cross-entropy, $\mathcal{L}_{\mathrm{ACT}}$ is the MSE between teacher and student hidden states at fold-landing and compression-boundary layers, and $\mathcal{L}_{\mathrm{FOLD}}$ penalizes deviation of trainable parameters from their post-fold initialization. Full definitions are provided in Appendix B. Following the formulation in Sec. 2.2, OAR reduces the effects of replacement deviation while restricting updates to remain aligned with the folded operators, preserving the structural benefits of layer replacement.

4 Experiments

4.1 Implementation Details

Models. We evaluate our method on six representative open-source large language models covering different model families, including **Llama2-7B** (Touvron et al., 2023), **Llama3-8B** (Grattafiori et al., 2024), **Baichuan2-7B** (Yang et al., 2025a), **Mistral-7B** (Jiang et al., 2023), **Qwen2.5-7B** (Yang et al., 2025c), and **Qwen3-8B** (Yang et al., 2025b). These models adopt diverse architectures and training recipes, providing a comprehensive evaluation of the generalizability of our approach.

Benchmarks. We evaluate the compressed models on a diverse set of language benchmarks, including 13 multiple-choice understanding tasks and 5 generative tasks. For the multiple-choice benchmarks, we categorize them into four groups: **Commonsense (CS)**: CoQA (Talmor et al., 2019), PIQA (Bisk et al., 2019), HellaSwag (Zellers et al., 2019), and SIQA (Sap et al., 2019); **Comprehension (CP)**: C3 (Sun et al., 2020), and RACE-Middle/High (Lai et al., 2017); **Semantics (SM)**: CMNLI (Xu et al., 2020), WSC (Levesque et al., 2012), and BoolQ (Clark et al., 2019); **Knowledge (KN)**: MMLU (Hendrycks et al., 2021), CMMLU (Li et al., 2024), and CHID (Zheng et al., 2019). In addition, we evaluate generative capabilities of compressed models on GSM8K (Cobbe et al., 2021), StrategyQA (Geva et al., 2021), XSum (Narayan et al., 2018), BBH (Suzgun et al., 2023), and NQ (Kwiatkowski et al., 2019).

Baselines. We compare our method with representative model compression methods focusing on structural redundancy reduction, including: (1) **LLM-Pruner** (Ma et al., 2023), which performs structured pruning by estimating the importance of coupled parameters and removing redundant structures in LLMs; (2) **SliceGPT** (Ashkboos et al., 2024), which compresses LLMs by removing low-importance hidden dimensions through activation-based slicing; (3) **LaCo** (Yang et al., 2024), which compresses LLMs by collapsing redundant Transformer layers according to layer-wise similarity; (4) **ShortGPT** (Men et al., 2025), which identifies redundant layers based on inter-layer representation similarity and removes layers with low contribution; and (5) **ReplaceMe** (Shopkhoev et al., 2025), which reduces depth redundancy by replacing the effects of removed layers with neighboring transformations.

Metrics. We use accuracy for multiple-choice benchmarks. Model degradation is measured by ΔNLL, ΔPPL, KL divergence, and Logit L_2.

Detailed experimental settings and metrics are provided in Appendix C. Additional results across LLM families, inference efficiency evaluations, among others, are provided in Appendices D.

4.2 Main Compression Results

To evaluate the effectiveness of ReLoNA, we compare it with several representative structured compression methods on Llama2-7B and Llama3-8B under a fixed compression ratio of **25%**.

Table 1: Main compression results on thirteen multiple-choice benchmarks. AVG. and RP. denote the average accuracy and **R**elative **P**erformance w.r.t. the dense model. **Bold** numbers indicate the best results among all compression methods. These definitions and the bolding convention apply in subsequent tables. The dense model serves as the reference for compression performance.

Methods	Benchmark													AVG.	RP.
	CoQA	C3	CMNLI	CMMLU	CHID	RACE-M	RACE-H	PIQA	MMLU	HellaSwag	SIQA	WSC	BoolQ		
Llama2-7B															
Dense	66.09	41.15	32.98	31.90	41.61	32.87	35.28	76.50	46.09	67.38	44.17	37.50	70.34	47.99	100.00%
LLM-Pruner	46.76	29.04	32.97	21.80	22.33	**71.11**	**58.88**	42.12	36.54	49.54	25.30	24.25	31.86	37.89	78.96%
SliceGPT	44.39	41.42	29.65	21.66	21.38	66.16	51.96	40.84	36.54	38.26	25.18	27.55	20.48	35.81	74.62%
LaCo	51.19	41.04	32.83	24.93	**34.48**	22.84	27.10	69.48	25.61	54.76	40.58	36.54	62.39	40.29	83.96%
ShortGPT	52.83	41.97	32.98	27.31	28.62	31.55	30.10	68.61	37.75	58.81	43.19	36.54	64.83	42.69	88.98%
ReplaceMe	**57.25**	40.71	32.97	30.11	28.38	35.52	36.56	69.31	46.29	58.57	43.24	**38.46**	**67.80**	45.01	93.80%
ReLoNA (w/o OAR)	51.43	**43.01**	**33.26**	30.20	31.48	31.89	32.73	68.12	42.87	54.33	41.66	36.54	63.30	43.14	89.90%
ReLoNA (w/ OAR)	56.67	41.21	32.99	**31.09**	30.72	34.68	34.82	**72.14**	**46.38**	**60.05**	**45.24**	36.54	**67.80**	**45.41**	**94.63%**
Llama3-8B															
Dense	72.81	61.37	33.00	53.64	63.74	81.82	76.47	79.82	67.95	67.64	51.89	55.77	78.81	64.98	100.00%
LLM-Pruner	47.83	48.88	32.20	50.63	42.17	69.75	49.66	48.11	36.54	**68.96**	31.90	40.09	49.22	47.38	72.92%
SliceGPT	48.98	39.56	32.06	28.20	25.84	65.29	50.63	42.94	39.20	39.20	26.56	38.48	23.18	38.27	58.90%
LaCo	40.13	36.88	32.58	25.39	34.06	21.38	21.47	67.08	24.29	43.50	41.50	39.42	61.83	37.65	57.95%
ShortGPT	40.05	41.70	**39.55**	26.65	22.59	36.21	31.76	67.25	28.94	44.69	42.63	49.04	42.63	39.51	60.81%
ReplaceMe	61.18	48.99	31.84	48.99	35.55	71.38	64.69	70.73	57.33	56.22	49.28	51.92	72.75	55.45	85.33%
ReLoNA (w/o OAR)	42.67	45.48	29.91	45.74	10.62	80.78	76.01	62.79	56.80	31.54	41.20	36.54	62.17	47.87	73.66%
ReLoNA (w/ OAR)	**64.62**	**53.10**	33.33	**53.27**	**45.64**	**81.41**	**76.39**	**74.76**	**66.30**	60.57	**49.54**	**58.65**	**75.99**	**61.04**	**93.94%**

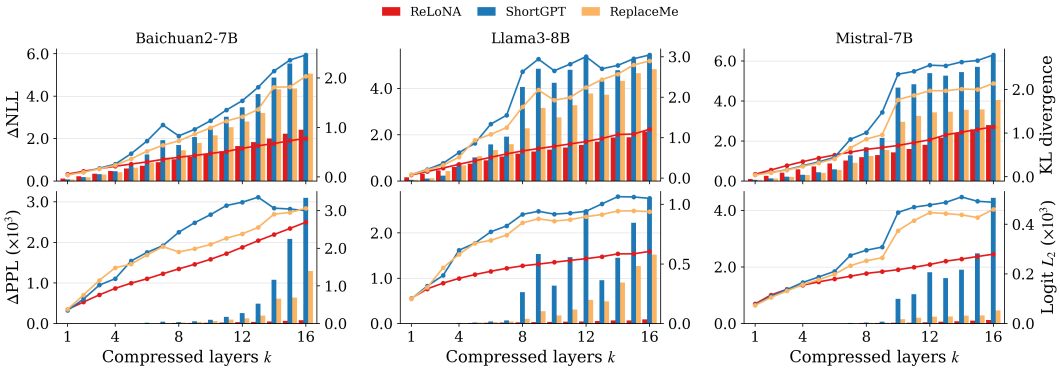


Figure 3: Fixed-k layer compression. **Left/right** y-axes correspond to **bars/lines**.

As shown in Table 1, ReLoNA consistently achieves the best average performance among all compared approaches on multiple-choice benchmarks, preserving approximately 94% of the relative performance of the original dense models. Table 2 further presents the results on generative benchmarks. Although several baselines maintain relatively stable accuracy on multiple-choice tasks, they suffer from substantial degradation on generation-oriented evaluations. In contrast, ReLoNA consistently achieves better performance preservation on generative tasks, indicating its ability to retain the layer-wise transformations that are essential for generation after depth reduction. The effectiveness of OAR further contributes to this performance by reducing residual perturbations introduced during operator replacement while preserving the compressed structure.

Table 2: Main compression results on generative language tasks.

Methods	Benchmark					AVG.	RP.
	GSM8K	StrategyQA	XSum	BBH	NQ		
Llama2-7B							
Dense	16.91	60.31	15.04	36.42	9.94	27.72	100.00%
LLM-Pruner	1.74	**53.01**	11.76	10.06	1.75	15.66	56.49%
SliceGPT	**2.96**	21.00	5.81	3.60	2.99	7.27	26.23%
LaCo	0.30	17.82	9.65	7.61	3.57	7.79	28.10%
ShortGPT	1.82	40.57	12.82	18.15	1.91	15.05	54.29%
ReplaceMe	2.27	44.98	12.95	26.66	1.47	17.67	63.74%
ReLoNA (w/o OAR)	1.90	28.56	4.46	12.72	**3.63**	10.25	36.98%
ReLoNA (w/ OAR)	1.52	48.43	**13.76**	**28.80**	2.22	**18.95**	**68.36%**
Llama3-8B							
Dense	79.00	72.49	14.67	60.76	24.96	50.38	100.00%
LLM-Pruner	2.35	53.45	14.05	5.87	3.19	15.78	31.32%
SliceGPT	16.68	52.10	**17.42**	3.94	2.91	18.61	36.94%
LaCo	0.23	4.67	3.56	7.45	0.69	3.32	6.59%
ShortGPT	1.67	10.35	3.72	5.14	1.52	4.48	8.89%
ReplaceMe	1.29	50.39	10.14	14.66	4.24	16.14	32.04%
ReLoNA (w/o OAR)	0.45	31.22	3.51	5.19	1.58	8.39	16.65%
ReLoNA (w/ OAR)	**47.23**	**66.77**	14.23	**49.84**	**5.51**	**36.72**	**72.89%**

4.3 Predictive Validity of LNA

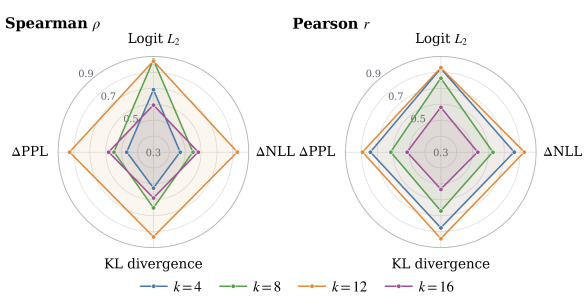


Figure 4: Correlation of Degradation Risk S_{cum} with actual degradation over fixed-k subsets.

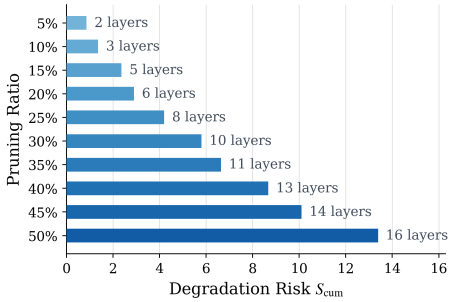


Figure 5: Relationship between the Degradation Risk S_{cum} and pruning ratio.

To evaluate the effectiveness of LNA-guided layer compression without any recovery, we compare the degradation against ShortGPT and ReplaceMe under the same k compressed layers (Figure 3). ReLoNA's advantage widens as k grows rather than shrinking. As k increases, these baselines become less reliable because their strategy do not account for the cumulative effects of successive compression. In contrast, LNA incorporates this propagation effect through its amplification-aware residual term, enabling more reliable layer ordering as compression becomes more aggressive.

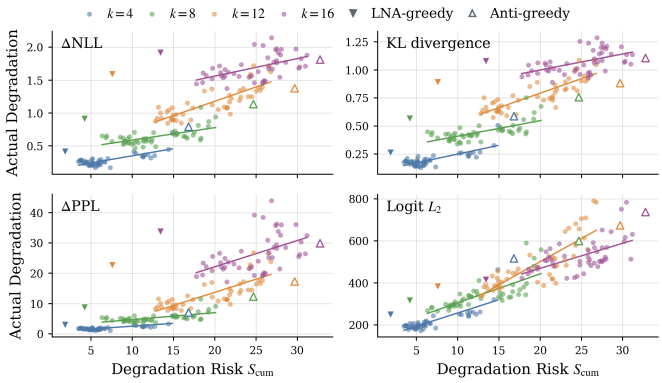


Figure 6: Fixed-k validation of Degradation Risk S_{cum} on Llama2-7B. Solid lines are fitted within each fixed-k group.

Figures 4 and 6 test a stronger property, namely whether the cumulative score predicts degradation for arbitrary, non-greedy subsets, not just greedy ones. The near-linear correlation across random subsets shows that S_{cum} is approximately additive rather than merely a relative ranking signal. This is consistent with the first-order additivity underlying LNA in Eq. (13), suggesting that the cumulative score can reflect the expected degradation of arbitrary layer subsets without exhaustive subset search, as observed in Appendix D.5.

4.4 Loss Budgets and Pruning Rates

To investigate whether ReLoNA can provide a controllable signal for compression budgeting, we evaluate under different pruning ratios on Llama2-7B. As shown in Table 3 and Figure 5, the performance degradation exhibits a consistent trend with the increase of the pruning ratio, while the cumulative risk score S_{cum} provides an effective indicator of the expected loss. ReLoNA maintains stable performance under moderate compression rates, preserving over 85% of the dense model performance even with 30% layer removal. PPL is evaluated on WikiText-2 using 1,000 sequences.

Table 3: Impact of pruning ratios on Llama2-7B. OAR is not applied unless specified.

Ratio	CS	CP	SM	KN	AVG. ↑	RP. ↑	PPL ↓
Dense	48.55	38.89	53.70	40.84	47.99	100.00%	12.63
10%	61.63	35.46	44.11	39.07	46.81	97.54%	21.23
20%	56.76	35.69	42.96	38.50	44.56	92.85%	48.02
30%	52.99	32.79	40.44	34.72	41.21	85.87%	63.57
40%	49.73	31.63	39.53	34.57	38.82	80.89%	131.13
50%	46.80	27.84	36.82	26.37	34.96	72.85%	186.35

Table 4: Evaluation results with different calibration datasets on Llama2-7B.

Calibration	CS	CP	SM	KN	AVG. ↑	RP. ↑	PPL ↓
Dense	48.22	45.19	54.20	40.84	47.99	100.00%	12.63
SlimOrca	53.26	39.73	40.72	38.20	42.88	89.35%	52.07
C4	53.95	39.34	40.39	37.45	42.87	89.33%	52.36
Wiki	53.25	39.21	40.25	38.22	42.69	88.96%	57.59
BookCorpus	56.95	39.20	40.42	32.51	42.66	88.89%	59.07
Alpaca	53.22	39.54	40.33	37.63	42.63	88.83%	54.27
SlimPajama	44.25	39.00	47.52	39.10	43.14	89.89%	49.97

4.5 Robustness and Ablation Studies

Calibration Data Robustness. We evaluate ReLoNA with different calibration datasets on Llama2-7B. As shown in Table 4, ReLoNA maintains consistent performance across diverse calibration distributions, indicating that LNA captures general layer properties rather than relying on specific data characteristics. Dataset details are provided in Appendix C.1.

Calibration Token Size. We further study the impact of the number of calibration tokens under a fixed parameter update ratio. Table 5 shows that ReLoNA achieves stable performance with moderate calibration sizes, with limited gains from further increasing the number of tokens, demonstrating its data efficiency in compression.

OAR Ablation. We conduct an ablation study of OAR on Llama3-8B. As shown in Table 6, removing any component degrades performance, confirming that the proposed recovery objectives jointly contribute to compensating compression-induced approximation errors.

Table 5: Evaluation results with different calibration sizes under a maximum parameter update ratio of 35% on Llama2-7B.

Token Size	CS	CP	SM	KN	AVG. ↑	RP. ↑
Dense	48.22	45.19	54.20	40.84	47.99	100.00%
8K	32.47	37.42	47.32	38.74	41.22	85.90%
16K	33.82	37.68	46.56	38.79	42.12	87.78%
32K	44.21	38.73	49.90	38.67	42.65	88.88%
64K	44.26	38.62	50.84	39.02	42.82	89.23%

Table 6: Ablation study of OAR on Llama3-8B.

Variant	CS	CP	SM	KN	AVG. ↑	PPL ↓
w/ OAR	77.52	54.91	59.05	54.10	61.04	35.61
w/o Tier-1	60.27	68.49	51.79	59.37	60.05	41.65
w/o Tier-2	60.79	67.89	51.56	59.56	59.78	40.40
w/o KL	62.11	63.13	46.17	51.86	55.94	32.24
w/o ACT	60.97	63.40	50.23	52.54	56.43	35.78
w/o NTP	60.32	62.32	45.60	52.27	55.40	42.02
w/o FOLD	61.05	62.73	46.64	52.57	56.15	35.58

5 Related Work

LLM Structural Pruning and Compression. Structural pruning reduces the computational cost of large language models by removing or transforming architectural components while preserving model capability. LLM-Pruner (Ma et al., 2023) identifies and prunes coupled structures based on gradient information, while SliceGPT (Ashkboos et al., 2024) reduces model width by slicing rows and columns based on PCA. For depth reduction, LaCo (Yang et al., 2024) performs layer collapse to remove redundant layers, ShortGPT (Men et al., 2025) uses Block Influence to identify redundant layers, and ReplaceMe (Shopkhoev et al., 2025) replaces pruned blocks with mergeable linear transformations. PruneMe (Gromov et al., 2025) identifies redundant layers based on block similarity, while LLM-Streamline (Chen et al., 2025) combines layer pruning with lightweight layer replacement. Despite these advances, how to quantify the degradation induced by an individual compression decision remains less explored.

Layer Redundancy and Sensitivity. Several studies characterize the functional importance and redundancy of individual layers. SLEB (Song et al., 2024) measures similarity between neighboring blocks to identify redundant blocks, while *Investigating Layer Importance in Large Language Models* (Zhang et al., 2024) evaluates layer importance using Shapley values and ablation. Maximum Redundancy Pruning (Gao et al., 2025) reveals non-uniform layerwise pruning sensitivity and allocates sparsity accordingly, while *Change Is the Only Constant* (Dumitru et al., 2024) uses input-output cosine similarity to estimate layer redundancy for dynamic slicing. These methods characterize layer importance or redundancy, but do not explicitly model how local approximation errors propagate through subsequent computation. ReLoNA instead couples local nonlinearity with downstream perturbation amplification to relate layer approximability to compression impact.

6 Conclusion

In this work, we present ReLoNA, an approach for structured compression of large language models based on local nonlinearity and downstream perturbation amplification. We propose Local Nonlinearity Amplification to assess compression risk. ReLoNA selects compression candidates based on LNA and adaptively applies appropriate replacement strategies according to their local nonlinear behavior, with operator-aligned recovery to mitigate residual perturbations. Experimental results demonstrate the effectiveness and generality of ReLoNA for structured LLM compression.

AI USE STATEMENT

In this work, we used generative AI tools to assist with language polishing and the refinement of mathematical formulations and equations. We have not used generative AI tools for research ideas, method design, experimental design, data analysis or scientific conclusions. All AI-assisted content was reviewed and revised by the authors for correctness, clarity, and consistency. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human subjects, personal or sensitive data collection, or other research procedures that require institutional ethical review. The experiments use publicly available language models and benchmark datasets for evaluating model compression. To the best of our knowledge, this work does not introduce significant privacy, security, discrimination, or other ethical risks. We have followed the applicable terms of use and research guidelines for the models and datasets used in this work.

REPRODUCIBILITY STATEMENT

We provide sufficient details to facilitate the reproduction of our results. The problem formulation and theoretical analysis are described in Appendix A, with complete proofs provided. The proposed compression and recovery procedures are detailed in Sec. 3. Experimental settings and reproducibility checklist are provided in Appendix C. We release the source code, configuration files, and scripts required to reproduce the reported results as supplementary materials.

REFERENCES

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress Large Language Models by Deleting Rows and Columns. In *International Conference on Learning Representations*, 2024. URL https://mlanthology.org/iclr/2024/ashkboos2024iclr-slicegpt/.

Song Bian, Minghao Yan, and Shivaram Venkataraman. Scaling inference-efficient language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 4303–4323. PMLR, 13–19 Jul 2025.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2019. URL https://api.semanticscholar.org/CorpusID:208290939.

Maosong Cao, Kai Chen, Haodong Duan, Yixiao Fang, Zhiwei Fei, Tong Gao, Ge Jiaye, Mo Li, Hongwei Liu, Junnan Liu, Yuan Liu, Chengqi Lyu, Han Lyu, Ningsheng Ma, Zerun Ma, Yu Sun, Zhiyong Wu, Linchen Xiao, Zhuozhi Xiong, Jun Xu, Haochen Ye, Zhaohui Yu, Yike Yuan, Songyang Zhang, Yufeng Zhao, Fengzhe Zhou, Peiheng Zhou, Dongsheng Zhu, Lin Zhu, and Jingming Zhuo. Opencompass: A universal evaluation platform for large language models, 2026. URL https://arxiv.org/abs/2605.19276.

Arnav Chavan, Raghav Magazine, Shubham Kushwaha, Merouane Debbah, and Deepak Gupta. Faster and lighter llms: A survey on current challenges and way forward. In Kate Larson (ed.), *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 7980–7988. International Joint Conferences on Artificial Intelligence Organization, 8 2024. doi: 10.24963/ijcai.2024/883. URL https://doi.org/10.24963/ijcai.2024/883. Survey Track.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and

R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 30362–30383, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/4b00a351b41358965613c118e87dc28c-Paper-Conference.pdf.

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

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. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Razvan-Gabriel Dumitru, Paul Ioan Clotan, Vikas Yadav, Darius Peteleaza, and Mihai Surdeanu. Change is the only constant: Dynamic LLM slicing based on layer redundancy. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9912–9920, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.579. URL https://aclanthology.org/2024.findings-emnlp.579/.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/frantar23a.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers, 2023. URL https://arxiv.org/abs/2210.17323.

Chang Gao, Kang Zhao, Runqi Wang, Jianfei Chen, and Liping Jing. Maximum redundancy pruning: A principle-driven layerwise sparsity allocation for llms. *Proceedings of the 33rd ACM International Conference on Multimedia*, 2025. URL https://api.semanticscholar.org/CorpusID:277272730.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361, 2021. doi: 10.1162/tacl_a_00370. URL https://aclanthology.org/2021.tacl-1.21/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers, 2025. URL https://arxiv.org/abs/2403.17887.

Song Han, Jeff Pool, John Tran, and William Dally. Learning both weights and connections for efficient neural network. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/ae0eb3eed39d2bcef4622b2499a05fe6-Paper.pdf.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding, 2021. URL https://arxiv.org/abs/2009.03300.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015a. URL https://arxiv.org/abs/1503.02531.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015b. URL http://arxiv.org/abs/1503.02531.

Yujia Huang, Huan Zhang, Yuanyuan Shi, J. Zico Kolter, and Anima Anandkumar. Training certifiably robust neural networks with efficient local lipschitz bounds. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 22745–22757. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/c055dcc749c2632fd4dd806301f05ba6-Paper.pdf.

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, , et al. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466, 2019. doi: 10.1162/tacl_a_00276. URL https://aclanthology.org/Q19-1026/.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding comprehension dataset from examinations. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel (eds.), *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL https://aclanthology.org/D17-1082/.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, KR'12, pp. 552–561. AAAI Press, 2012. ISBN 9781577355601.

Haonan Li, Yixuan Zhang, Fajri Koto, Yifei Yang, Hai Zhao, Yeyun Gong, Nan Duan, and Timothy Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11260–11285, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.671. URL https://aclanthology.org/2024.findings-acl.671/.

Chen Liang, Simiao Zuo, Qingru Zhang, Pengcheng He, Weizhu Chen, and Tuo Zhao. Less is more: Task-aware layer-wise distillation for language model compression. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 20852–20867. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/liang23j.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. Llm-pruner: On the structural pruning of large language models. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 21702–21720. Curran Associates, Inc., 2023. doi: 10.52202/075280-0950. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/44956951349095f74492a5471128a7e0-Paper-Conference.pdf.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1035. URL https://aclanthology.org/2025.findings-acl.1035/.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models, 2016. URL https://arxiv.org/abs/1609.07843.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. Orca: Progressive learning from complex explanation traces of gpt-4, 2023. URL https://arxiv.org/abs/2306.02707.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 1797–1807, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1206. URL https://aclanthology.org/D18-1206/.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference. In D. Song, M. Carbin, and T. Chen (eds.), *Proceedings of Machine Learning and Systems*, volume 5, pp. 606–624. Curan, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/file/c4be71ab8d24cdfb45e3d06dbfca2780-Paper-mlsys2023.pdf.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2023. URL https://arxiv.org/abs/1910.10683.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Social IQa: Commonsense reasoning about social interactions. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4463–4473, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1454. URL https://aclanthology.org/D19-1454/.

Dmitriy Shopkhoev, Ammar Ali, Magauiya Zhussip, Valentin Malykh, Stamatios Lefkimmiatis, Nikos Komodakis, and Sergey Zagoruyko. ReplaceMe: Network Simplification via Depth Pruning and Transformer Block Linearization. In *Advances in Neural Information Processing Systems*, 2025. URL https://mlanthology.org/neurips/2025/shopkhoev2025neurips-replaceme/.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, et al. Openai gpt-5 system card, 2026. URL https://arxiv.org/abs/2601.03267.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://cerebras.ai/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46136–46155. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/song24f.html.

Kai Sun, Dian Yu, Dong Yu, and Claire Cardie. Investigating prior knowledge for challenging Chinese machine reading comprehension. *Transactions of the Association for Computational Linguistics*, 8:141–155, 2020. doi: 10.1162/tacl_a_00305. URL https://aclanthology.org/2020.tacl-1.10/.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/14c856c7a41297804de4c4890e846b25-Paper-Conference.pdf.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13003–13051, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.824. URL https://aclanthology.org/2023.findings-acl.824/.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421/.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. Structured pruning of large language models. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 6151–6162, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.496. URL https://aclanthology.org/2020.emnlp-main.496/.

Siyue Wu, Hongzhan Chen, Xiaojun Quan, Qifan Wang, and Rui Wang. AD-KD: Attribution-driven knowledge distillation for language model compression. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8449–8465, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.471. URL https://aclanthology.org/2023.acl-long.471/.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. Structured pruning learns compact and accurate models. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1513–1528, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.107. URL https://aclanthology.org/2022.acl-long.107/.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/xiao23c.html.

Liang Xu, Hai Hu, Xuanwei Zhang, Lu Li, Chenjie Cao, Yudong Li, Yechen Xu, Kai Sun, Dian Yu, Cong Yu, Yin Tian, Qianqian Dong, Weitang Liu, Bo Shi, Yiming Cui, Junyi Li, Jun Zeng, Rongzhao Wang, Weijian Xie, Yanting Li, et al. CLUE: A Chinese language understanding evaluation benchmark. In Donia Scott, Nuria Bel, and Chengqing Zong (eds.), *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 4762–4772, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.419. URL https://aclanthology.org/2020.coling-main.419/.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, et al. Baichuan 2: Open large-scale language models, 2025a. URL https://arxiv.org/abs/2309.10305.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, et al. Qwen3 technical report, 2025b. URL https://arxiv.org/abs/2505.09388.

Qwen : An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, et al. Qwen2.5 technical report, 2025c. URL https://arxiv.org/abs/2412.15115.

Yifei Yang, Zouying Cao, and Hai Zhao. LaCo: Large language model pruning via layer collapse. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6401–6417, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.372. URL https://aclanthology.org/2024.findings-emnlp.372/.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 469–479, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.29. URL https://aclanthology.org/2024.blackboxnlp-1.29/.

Chujie Zheng, Minlie Huang, and Aixin Sun. ChID: A large-scale Chinese IDiom dataset for cloze test. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 778–787, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1075. URL https://aclanthology.org/P19-1075/.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12: 1556–1577, 2024. doi: 10.1162/tacl_a_00704. URL https://aclanthology.org/2024.tacl-1.85/.

Yukun Zhu, Ryan Kiros, Richard Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books, 2015. URL https://arxiv.org/abs/1506.06724.

A Theoretical Details

A.1 Proof and Derivation of Proposition

Proposition 1. For any $H \in \mathcal{A}$, expand the objective around A_B:

$$\begin{aligned}\mathbb{E}\|B - (A_B + H)\|_2^2 &= \mathbb{E}\|r_B - H\|_2^2 \\ &= \mathbb{E}\|r_B\|_2^2 - 2\mathbb{E}\langle r_B, H\rangle + \mathbb{E}\|H\|_2^2.\end{aligned}$$

Since A_B minimizes the objective over the affine class, the directional derivative at A_B in every affine direction H vanishes, giving $\mathbb{E}\langle r_B, H\rangle = 0$. Substitution yields the Pythagorean identity in Eq. (6), which proves optimality and the claimed irreducible-error interpretation within the affine function class.

Proposition 2. Differentiability of G_B at h gives

$$G_B(h + \sigma u) - G_B(h) = \sigma J_{G_B}(h)u + r(\sigma u),$$

where $\|r(\sigma u)\|_2/\sigma \to 0$ as $\sigma \to 0$. Since $\|\sigma u\|_2 = \sigma$ for $\|u\|_2 = 1$, we have

$$K_B(h, \sigma u) = \left\| J_{G_B}(h)u + \frac{r(\sigma u)}{\sigma} \right\|_2.$$

By the reverse triangle inequality,

$$|K_B(h, \sigma u) - \|J_{G_B}(h)u\|_2| \le \frac{\|r(\sigma u)\|_2}{\sigma} \to 0,$$

which gives $K_B(h, \sigma u) = \|J_{G_B}(h)u\|_2 + o(1)$.

Proposition 3. Since $F = G_B \circ B$ and $\widetilde{F} = G_B \circ \widetilde{B}$, we have

$$F(x) - \widetilde{F}(x) = G_B(B(x)) - G_B(\widetilde{B}(x)). \tag{17}$$

By construction, $\gamma_x(0) = \widetilde{B}(x)$, $\gamma_x(1) = B(x)$, and

$$\frac{\mathrm{d}\gamma_x(t)}{\mathrm{d}t} = B(x) - \widetilde{B}(x) = e_B(x). \tag{18}$$

Since G_B is differentiable on $\gamma_x(t)$, the fundamental theorem of calculus and the chain rule give

$$\begin{aligned}F(x) - \widetilde{F}(x) &= \int_0^1 \frac{\mathrm{d}}{\mathrm{d}t} G_B\big(\gamma_x(t)\big)\,\mathrm{d}t \\ &= \int_0^1 J_{G_B}\big(\gamma_x(t)\big) e_B(x)\,\mathrm{d}t.\end{aligned} \tag{19}$$

Taking the ℓ_2 norm and applying the triangle inequality yields

$$\begin{aligned}\|F(x) - \widetilde{F}(x)\|_2 &\le \int_0^1 \big\|J_{G_B}\big(\gamma_x(t)\big) e_B(x)\big\|_2\,\mathrm{d}t \\ &\le \int_0^1 \big\|J_{G_B}\big(\gamma_x(t)\big)\big\|_{2\to 2}\,\|e_B(x)\|_2\,\mathrm{d}t \\ &\le \kappa_B(x)\|e_B(x)\|_2.\end{aligned} \tag{20}$$

where the second inequality follows from the definition of the induced ℓ_2 operator norm, and the last inequality follows from the definition of $\kappa_B(x)$. This proves the pointwise bound in Eq. (11).

A.2 Inter-layer Similarity versus LNA

This section formalizes the relationship between inter-layer cosine similarity and the local affine mismatch, and further establishes why cosine similarity cannot generally capture the downstream response factor considered by LNA.

Proposition 4 (inter-layer similarity versus LNA). Let B and C denote two candidate layers, and let R denote a neighboring layer used as a local representation reference. Define their inter-layer cosine similarity as

$$\mathrm{Cos}(B,R)=\frac{\langle z_B,z_R\rangle}{\|z_B\|_2\|z_R\|_2}, \tag{21}$$

where z_B and z_R are the corresponding vectorized representations. Under the locally aligned neighboring-layer assumption described below, the normalized affine mismatch is monotonically related to inter-layer similarity:

$$\mathrm{Cos}(B,R)\geq\mathrm{Cos}(C,R)\Longrightarrow\widetilde{\mathcal{N}}(B)\leq\widetilde{\mathcal{N}}(C). \tag{22}$$

Thus, inter-layer cosine similarity can characterize the local approximability of a layer: higher similarity to a neighboring layer corresponds to a smaller local affine mismatch.

However, LNA additionally accounts for the downstream response to the induced perturbation. In particular, if $\widehat{K}_B=\widehat{K}_C=K>0$, then

$$\mathrm{Cos}(B,R)\geq\mathrm{Cos}(C,R)\Longrightarrow\mathrm{LNA}(B)\leq\mathrm{LNA}(C). \tag{23}$$

When the downstream responses are unequal but positive, the ordering is instead determined jointly by local mismatch and downstream response:

$$\mathrm{LNA}(B)\leq\mathrm{LNA}(C)\iff\frac{\widetilde{\mathcal{N}}(B)}{\widetilde{\mathcal{N}}(C)}\leq\frac{\widehat{K}_C}{\widehat{K}_B}. \tag{24}$$

Therefore, inter-layer cosine similarity captures the local approximability component of compression risk, whereas LNA further accounts for the position-dependent downstream response.

Proof of Proposition 4. We first establish the connection between inter-layer cosine similarity and local affine mismatch under the locally aligned neighboring-layer assumption. For a candidate layer B and its neighboring reference layer R, write their local functional decompositions as

$$f_B(x)=A_B(x)+r_B(x),\qquad f_R(x)=A_R(x)+r_R(x), \tag{25}$$

where A_B and A_R denote the corresponding local affine components, and r_B and r_R denote their local affine residuals.

Within the local activation region induced by the calibration data, assume that neighboring layers are locally aligned such that

$$A_B(x)=A_R(x),\qquad r_R(x)=\alpha r_B(x),\qquad 0\leq\alpha<1, \tag{26}$$

and that their representation scales are comparable, namely

$$\|f_B(x)\|_2=\|f_R(x)\|_2. \tag{27}$$

These conditions characterize a locally depth-smooth regime in which the dominant affine transformation changes slowly across neighboring layers, while the nonlinear residual evolves continuously along depth.

Under Eq. (25) and Eq. (26), the representation difference satisfies

$$\begin{aligned}f_B(x)-f_R(x)&=A_B(x)-A_R(x)+r_B(x)-r_R(x)\\&=(1-\alpha)r_B(x).\end{aligned} \tag{28}$$

Hence,

$$\frac{\|f_B(x)-f_R(x)\|_2^2}{\|f_B(x)\|_2^2}=(1-\alpha)^2\frac{\|r_B(x)\|_2^2}{\|f_B(x)\|_2^2}. \tag{29}$$

On the other hand, cosine similarity satisfies the exact identity

$$\left\|\frac{f_B(x)}{\|f_B(x)\|_2}-\frac{f_R(x)}{\|f_R(x)\|_2}\right\|_2^2=2\big(1-\mathrm{Cos}(B,R)\big). \tag{30}$$

Together with Eq. (27), this gives

$$2\big(1-\mathrm{Cos}(B,R)\big)=\frac{\|f_B(x)-f_R(x)\|_2^2}{\|f_B(x)\|_2^2}. \tag{31}$$

Combining Eq. (29) and Eq. (31), we obtain

$$\frac{\|r_B(x)\|_2^2}{\|f_B(x)\|_2^2}=\frac{2}{(1-\alpha)^2}\big(1-\mathrm{Cos}(B,R)\big). \tag{32}$$

Since $\alpha<1$ is fixed within the considered local regime, the right-hand side is strictly decreasing in $\mathrm{Cos}(B,R)$. Averaging over the calibration activation distribution and applying the normalization used in $\widetilde{\mathcal{N}}(B)$ therefore yields

$$\widetilde{\mathcal{N}}(B)=c_\alpha\big(1-\mathrm{Cos}(B,R)\big),\qquad c_\alpha>0, \tag{33}$$

up to the normalization convention in $\widetilde{\mathcal{N}}(B)$. Consequently,

$$\mathrm{Cos}(B,R)\geq\mathrm{Cos}(C,R)\Longrightarrow\widetilde{\mathcal{N}}(B)\leq\widetilde{\mathcal{N}}(C), \tag{34}$$

which proves the local similarity–approximability ordering in Eq. (22).

We next consider the LNA score. By definition,

$$\mathrm{LNA}(B)=\widetilde{\mathcal{N}}(B)\widehat{K}_B. \tag{35}$$

When $\widehat{K}_B=\widehat{K}_C=K>0$, multiplying the ordering in Eq. (34) by the same positive factor preserves the ordering:

$$\widetilde{\mathcal{N}}(B)\leq\widetilde{\mathcal{N}}(C)\Longrightarrow K\widetilde{\mathcal{N}}(B)\leq K\widetilde{\mathcal{N}}(C). \tag{36}$$

Therefore,

$$\mathrm{Cos}(B,R)\geq\mathrm{Cos}(C,R)\Longrightarrow\mathrm{LNA}(B)\leq\mathrm{LNA}(C), \tag{37}$$

which proves Eq. (23).

For unequal positive downstream responses,

$$\begin{aligned}\mathrm{LNA}(B)\leq\mathrm{LNA}(C)&\Longleftrightarrow\widetilde{\mathcal{N}}(B)\widehat{K}_B\leq\widetilde{\mathcal{N}}(C)\widehat{K}_C\\&\Longleftrightarrow\frac{\widetilde{\mathcal{N}}(B)}{\widetilde{\mathcal{N}}(C)}\leq\frac{\widehat{K}_C}{\widehat{K}_B},\end{aligned} \tag{38}$$

which proves Eq. (24). Hence, cosine similarity can recover the ordering induced by local affine mismatch under the locally aligned regime, but it cannot in general recover the ordering induced by LNA because LNA additionally depends on the position-dependent downstream response.

B Operator-Aligned Recovery Details

Let $(x_{1:T},m_{1:T})$ denote a calibration sequence and its attention mask, where $m_t\in\{0,1\}$ indicates a valid (non-padding) position. The dense teacher and the compressed student produce logits z_t^T,z_t^S and hidden states $h_{\ell,t}^T,h_{\ell,t}^S$, respectively. OAR updates only the parameters selected by the operator-aligned parameter mask: Tier-1 contains the retained target layers into which folded operators are absorbed, while Tier-2 contains the boundary layers immediately adjacent to each compressed block.

Output-distribution matching. For temperature τ, the per-token teacher-to-student divergence is

$$d_t^{\mathrm{KL}}=\mathrm{KL}\big(\mathrm{softmax}(z_t^T/\tau)\,\big\|\,\mathrm{softmax}(z_t^S/\tau)\big)\,. \tag{39}$$

The implementation uses $\tau=2$. With uniform token weighting, $\mathcal{L}_{\mathrm{KL}}$ is the masked mean

$$\mathcal{L}_{\mathrm{KL}}=\frac{\sum_{t=1}^T m_t d_t^{\mathrm{KL}}}{\max(\sum_{t=1}^T m_t,1)}. \tag{40}$$

In the default setting we additionally emphasize positions at which the current student is most inconsistent with the teacher. Define

$$w_t=\max\left(0,\frac{m_t d_t^{\mathrm{KL}}}{\frac{\sum_{u=1}^T m_u d_u^{\mathrm{KL}}}{\max(\sum_{u=1}^T m_u,1)}}\right),\qquad\mathcal{L}_{\mathrm{KL}}=\frac{\sum_{t=1}^T m_t w_t d_t^{\mathrm{KL}}}{\max(\sum_{t=1}^T m_t,1)}. \tag{41}$$

Thus, the weights are detached from the gradient and have mean one over valid tokens (up to masking), so they change where the KL signal is concentrated without introducing an additional learned weighting network.

Hidden-representation alignment. For a teacher–student hidden-state pair (ℓ_S, ℓ_T), we use the masked per-token mean-squared error

$$\mathcal{D}_{\text{hid}}(\ell_S, \ell_T) = \frac{\sum_{t=1}^{T} m_t \, \|h^S_{\ell_S,t} - h^T_{\ell_T,t}\|_2^2/d}{\max(\sum_{t=1}^{T} m_t, 1)}, \tag{42}$$

where d is the hidden width. For each compressed block, the boundary output alignment is always included. Alignment at the input of the fold-target layer is included only when its student hidden-state index is distinct from the boundary index; this avoids counting the same state twice when folding collapses the two locations. If $\mathcal{P}$ is the resulting set of hidden state pairs, then

$$\mathcal{L}_{\text{ACT}} = \frac{1}{|\mathcal{P}|} \sum_{(\ell_S,\ell_T)\in\mathcal{P}} \mathcal{D}_{\text{hid}}(\ell_S, \ell_T). \tag{43}$$

Next-token prediction. Using the ground-truth labels $y_{1:T}$, logits and labels are shifted by one position. Let $q_t = \text{softmax}(z_t^S)$ and let $\widetilde{m}_t$ be one when y_t is not an ignore label and $m_t = 1$. The student language-modeling loss is

$$\mathcal{L}_{\text{NTP}} = -\frac{\sum_{t=1}^{T-1} \widetilde{m}_{t+1} \log q_{t,y_{t+1}}}{\max(\sum_{t=1}^{T-1} \widetilde{m}_{t+1}, 1)}. \tag{44}$$

This term uses the data labels rather than teacher probabilities and therefore directly preserves next-token prediction under the recovered model.

Fold-preservation regularization. Let θ_j^0 be the parameter snapshot taken immediately after operator replacement, and let θ_j be its value during recovery. The regularizer is applied only to trainable parameters and averages their element-wise squared deviation:

$$\mathcal{L}_{\text{FOLD}} = \frac{1}{|\mathcal{J}|} \sum_{j\in\mathcal{J}} \frac{1}{|\theta_j|} \|\theta_j - \theta_j^0\|_2^2, \tag{45}$$

where $\mathcal{J}$ is the set of trainable parameter tensors. Consequently, the regularizer discourages unnecessary movement away from the folded operator while still allowing the localized recovery terms to correct residual errors.

Weights and rationale. The final objective is Eq. (16) with the default values

$$(\lambda_{\text{KL}}, \lambda_{\text{act}}, \lambda_{\text{NTP}}, \lambda_{\text{fold}}) = (1.0,\ 0.5,\ 0.15,\ 0.01), \qquad \tau = 2, \tag{46}$$

and importance weighting enabled. These coefficients assign primary weight to output-distribution matching, give hidden-state alignment a substantial but smaller role because it acts at a limited number of operator-aligned locations, retain next-token supervision as a stabilizing auxiliary signal, and keep the fold penalty weak so that it preserves the compressed initialization without preventing useful recovery.

C Experimental Settings and Details

C.1 Reproducibility checklist

Pruning Configurations. All experiments are conducted on a server with 2× NVIDIA A800 80GB GPUs. For each layer, we fit a ridge-regularized affine map with a regularization coefficient of 10^{-4} and estimate local nonlinearity and Lipschitz factors using 8 random perturbations per token with a perturbation radius $\sigma = 0.01 \times \mathbb{E}\|x\|$. The adaptive collapse thresholds are set to the 25th and 75th percentiles of the measured local nonlinearity. Model-specific settings are summarized in Table 7. The calibration token size and maximum parameter update ratio (Pruning Cap) are specified for each model.

Table 7: Pruning configurations used in our experiments.

Model	Calibration Tokens	Pruning Cap
Llama2-7B	64K	55%
Llama3-8B	64K	55%
Qwen2.5-7B	9.8K	55%
Qwen3-8B	8K	55%
Baichuan2-7B	8K	55%
Mistral-7B	64K	75%

Operator Folding Details. Affine absorption and residual folding produce a boundary map T equal to A_B or $I + \alpha A_{R_B}$, respectively; keep does not fold. Consecutive deleted layers compose into one T, which is folded into the next retained layer L^+ without adding modules or changing the residual topology. On calibration inputs $\{x_j\}$, set $x'_j = T(x_j)$. For each attention entry projection $P \in \{q, k, v\}$ after the input Norm of L^+, we solve

$$\min_{\widetilde{W}_P, \widetilde{b}_P} \sum_j \left\| \widetilde{W}_P \,\mathrm{Norm}(x_j) + \widetilde{b}_P - \left(W_P \,\mathrm{Norm}(x'_j) + b_P\right) \right\|_2^2 + \lambda \|\widetilde{W}_P\|_F^2, \tag{47}$$

and write the solution into P. Bias-free modules remain bias-free; W_o and the MLP weights are left unchanged. OAR then fine-tunes the landing and boundary layers.

OAR Configurations. OAR updates only the fold landing and pruning boundary layers, while keeping the remaining parameters frozen. It uses the same calibration data and runs for 1,500 optimization steps with learning rate 2×10^{-5}, warmup ratio 0.05, weight decay 0.01, and an effective batch size of 16. The loss weights are $(\lambda_{\mathrm{kl}}, \lambda_{\mathrm{act}}, \lambda_{\mathrm{ntp}}, \lambda_{\mathrm{fold}}) = (1.0, 0.5, 0.15, 0.01)$ with KL temperature $T = 2$.

Calibration datasets. Unless otherwise specified, we use SlimPajama (Soboleva et al., 2023) as the default calibration corpus. To evaluate the effect of calibration-domain choice, we consider SlimOrca (Mukherjee et al., 2023), C4 (Raffel et al., 2023), WikiText-2 (Merity et al., 2016), BookCorpus (Zhu et al., 2015), Alpaca (Taori et al., 2023), and SlimPajama. These datasets cover diverse distributions, including instruction data, web text, encyclopedic text, books, and pretraining mixtures. For each dataset, documents are tokenized with the corresponding model tokenizer and sampled until the prescribed calibration budget is reached. All other pruning are kept fixed across calibration datasets.

C.2 Evaluation Metrics

Evaluation framework. We evaluate all pruned models using OpenCompass (Cao et al., 2026), a unified benchmarking toolkit for large language models. Each benchmark is specified by a dataset configuration file and evaluated in one of two inference modes:

- **`ppl` mode** for multiple-choice / classification tasks, where the model scores each candidate answer by its conditional perplexity and selects the option with the lowest perplexity;
- **`gen` mode** for open-ended generation tasks, where the model produces free-form completions and task-specific post-processors extract the final answer before metric computation.

All results are reported as percentages unless otherwise noted. For benchmarks with multiple subtasks (e.g., MMLU, CMMLU, and BBH), we report the average over all subtasks.

Multiple-choice benchmarks. Table 8 lists the 13 multiple-choice benchmarks used in our main experiments. All tasks are evaluated in `ppl` mode with the `accuracy` metric. For each example, the model computes the perplexity of every candidate answer conditioned on the prompt, and the prediction is the candidate with the lowest perplexity:

$$\hat{y} = \arg\min_{c \in \mathcal{C}} \mathrm{PPL}(x, c), \tag{48}$$

Table 8: Evaluation metrics for multiple-choice benchmarks.

Benchmark	# Choices	Metric
CoQA	5	`accuracy`
C3	4	`accuracy`
CMNLI	3	`accuracy`
CMMLU	4	`accuracy`
CHID	10	`accuracy`
RACE-M	4	`accuracy`
RACE-H	4	`accuracy`
PIQA	2	`accuracy`
MMLU	4	`accuracy`
HellaSwag	4	`accuracy`
SIQA	3	`accuracy`
WSC	2	`accuracy`
BoolQ	2	`accuracy`

Table 9: Evaluation metrics for generative benchmarks.

Benchmark	Metric	Scoring protocol
GSM8K	`accuracy`	8-shot chain-of-thought, exact match on the final numeric answer.
StrategyQA	`accuracy`	6-shot prompting, binary yes/no accuracy after answer extraction.
XSum	`rougeL`	Abstractive summarization, reporting ROUGE-1/2/L/Lsum.
BBH	`score`	27 reasoning subtasks, macro-averaged exact-match accuracy.
NQ	`score`	Open-domain QA, based on normalized answer matching.

where x is the input prompt (optionally with in-context demonstrations) and $\mathcal{C}$ is the set of candidate answers. A prediction is counted as correct if $\hat{y}$ matches the gold label.

Generative benchmarks. Table 9 summarizes the five generative benchmarks. All tasks are evaluated in `gen` mode, where the model autoregressively generates a completion and a task-specific evaluator scores the extracted answer.

C.3 Pruning Degradation Metrics

To evaluate the predictive power of the LNA score, we measure the functional degradation induced by layer removal or affine replacement on a held-out evaluation set, relative to the original dense *teacher* model. Let z_t^{tea} and z_t^{stu} denote the teacher and student logits at token position t, and let $\mathcal{M}$ be the set of valid (non-padding) positions. Unless otherwise specified, all metrics are averaged over $\mathcal{M}$.

KL divergence. We compute the token-averaged Kullback–Leibler divergence from the teacher to the student distribution, using temperature scaling $T = 2$:

$$\mathrm{KL} = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \mathrm{KL}\big(p_t^{\mathrm{tea}} \,\|\, p_t^{\mathrm{stu}}\big), \quad p_t = \mathrm{softmax}(z_t/T). \tag{49}$$

Lower KL indicates closer output distributions between the compressed and dense models.

Logit L_2 distance. We measure the average ℓ_2 distance between teacher and student logits:

$$\mathrm{Logit}\, L_2 = \frac{1}{|\mathcal{M}|} \sum_{t \in \mathcal{M}} \big\| z_t^{\mathrm{tea}} - z_t^{\mathrm{stu}} \big\|_2 . \tag{50}$$

This metric directly measures the output-space perturbation induced by compression.

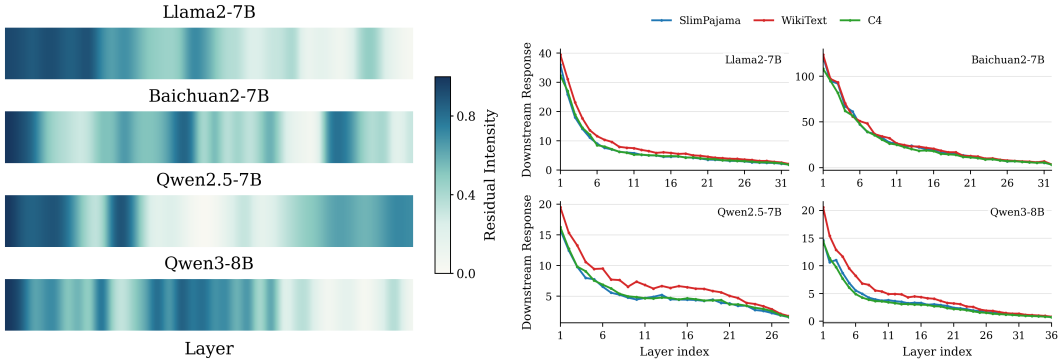
ΔNLL (change in negative log-likelihood). Let $\mathrm{NLL}^{\mathrm{tea}}$ and $\mathrm{NLL}^{\mathrm{stu}}$ denote the teacher and student next-token negative log-likelihoods, respectively. Using shifted logits and labels, we compute

$$\mathrm{NLL} = -\frac{1}{|\mathcal{M}'|} \sum_{t \in \mathcal{M}'} \log p(x_{t+1} \mid x_{\le t}), \tag{51}$$

where $\mathcal{M}'$ denotes the set of valid next-token positions. We define the degradation as

$$\Delta\mathrm{NLL} = \mathrm{NLL}^{\mathrm{stu}} - \mathrm{NLL}^{\mathrm{tea}}. \tag{52}$$

A positive ΔNLL indicates increased language-modeling loss relative to the dense teacher.



(a) Local Nonlinearity. (b) Downstream Response.

Figure 7: **Detailed Observations of other Models.**

ΔPPL (change in perplexity). Perplexity is computed as the exponential of NLL, and we report its absolute change:

$$\Delta\mathrm{PPL} = \exp\big(\mathrm{NLL}^{\mathrm{stu}}\big) - \exp\big(\mathrm{NLL}^{\mathrm{tea}}\big)\,. \tag{53}$$

D Additional Experimental Results

D.1 Detailed Observations

Figure 7 extends the observations in Sec. 2.2 and Sec. 2.3 to additional LLMs. Across these model families, the results show that local nonlinearity exhibits variation across layers, providing meaningful discrimination in the approximability of different layers. Meanwhile, the downstream amplification effect exhibits relatively stable distributional patterns across different layer positions, indicating that the response to compression-induced perturbations is not arbitrary but follows consistent characteristics over the data distribution. These observations support the generality of our underlying motivation and suggest that the combination of local approximation behavior and downstream perturbation response is not specific to a particular model family.

D.2 Validity of Random-Direction $\widehat{K}_B$ Estimates

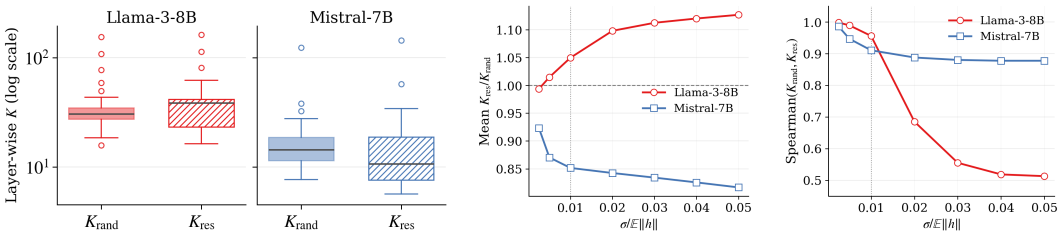
Proposition 3 bounds normalized output degradation by the product $\bar{\kappa}_B\widetilde{\mathcal{N}}(B)$, where $\bar{\kappa}_B$ is a local operator-norm response of G_B along the replacement path. This quantity is not available in practice: evaluating it would require Jacobians (or equivalent operator-norm estimates) at many points on γ_x. Following Proposition 2, LNA therefore replaces $\bar{\kappa}_B$ by a finite-difference probe $K_B(h,\delta)$ in Eq. (7) and summarizes probes by their median $\widehat{K}_B$ (Sec. 2.3).

A natural concern is that operator replacement does not perturb h in a random direction. When $\widetilde{B} = A_B$, the induced local error lies along the residual $e = B(x) - A_B(x)$, so the finite-difference most aligned with the compression mechanism would probe this residual direction. We therefore introduce a *diagnostic* residual-aligned probe K_{res} solely for validation—it is still a finite-difference statistic, not $\bar{\kappa}_B$, and ask whether the cheap random median agrees with it. On SlimPajama activations for Llama-3-8B and Mistral-7B we compare

$$K_{\mathrm{rand}}(u) = \frac{\|G_B(h+\sigma u) - G_B(h)\|_2}{\sigma}, \qquad K_{\mathrm{res}} = \frac{\|G_B\Big(h+\sigma\frac{e}{\|e\|_2}\Big) - G_B(h)\|_2}{\sigma}, \tag{54}$$

with $h = B(x)$ and $\sigma = \rho\cdot\mathbb{E}\|h\|_2$, scanning $\rho \in \{0.0025, 0.005, 0.01, 0.02, 0.03, 0.04, 0.05\}$. Here K_{res} requires fitting A_B first; K_{rand} does not, which is why the main method prefers random probes if the two agree.

Random probes match residual-aligned amplification. At the default radius $\rho = 0.01$, the two estimates are closely aligned: Spearman correlations between layer-wise K_{rand} and K_{res} are 0.95



(a) Layer-wise K distributions. (b) Mean ratio across radii. (c) Spearman across radii.

Figure 8: **Agreement between K_{rand} and K_{res}.** Dotted lines in (b)–(c) mark the default $\rho = 0.01$.

(Llama-3-8B) and 0.91 (Mistral-7B) (Figure 8c). Figure 8a shows matching layer-wise structure (shared high-response outliers; similar central mass), and Figure 8b shows mean ratios of order one (≈ 1.05 / ≈ 0.85). Thus the random median recovers essentially the same downstream sensitivity profile as the replacement-aligned diagnostic probe along e. This supports using cheap random-direction $\widehat{K}_B$ in LNA, rather than residual-conditioned probing or the inaccessible $\bar{\kappa}_B$.

Sensitivity of ρ. Proposition 2 asks for a small nonzero radius. For $\rho \leq 0.01$, agreement between K_{rand} and K_{res} remains strong on both models (Spearman ≥ 0.91). Larger radii weaken the correspondence, especially on Llama-3-8B (Figure 8c), while Mistral-7B stays flatter but gains nothing from enlarging ρ. Mean ratios vary smoothly with ρ (Figure 8b). Thus $\sigma = 0.01 \times \mathbb{E}\|h\|_2$ is large enough for stable finite differences, yet small enough that random and residual-aligned probes still describe the same local response.

D.3 Inference Efficiency

To evaluate the practical benefits of ReLoNA, we measure the inference efficiency of dense and compressed models on Llama2-7B and Llama3-8B. As shown in Table 10, compressing the models from 32 to 24 layers substantially reduces both parameter size and memory consumption, while consistently improving inference throughput. Specifically, ReLoNA achieves around $1.31\times$ speedup with only a fraction of the original memory footprint, demonstrating that the proposed depth compression strategy effectively translates model compactness into real-world inference acceleration. These results confirm that ReLoNA provides a favorable trade-off between compression efficiency and deployment cost.

Table 10: Inference efficiency comparison between dense and ReLoNA-compressed models. Llama2-7B and Llama3-8B are compressed from 32 layers to 24 layers.

Metric	Llama2-7B		Llama3-8B	
	Dense (32L)	ReLoNA (24L)	Dense (32L)	ReLoNA (24L)
Parameters (B)	6.738	5.119	8.030	6.285
Memory (GB)	12.551	9.543	14.958	11.715
Prefill L512 (tok/s)	10,259	13,388	9,346	12,125
Prefill L2048 (tok/s)	11,532	15,119	10,597	13,839
Decode (tok/s)	21.73	28.55	21.51	28.09
TTFT (ms)	49.85	37.85	50.44	39.37
Throughput Speedup (↑)	–	1.31×	–	1.31×
Memory Ratio (↓)	–	0.76×	–	0.78×

D.4 Main Results across LLM Families

To further examine the generalizability of ReLoNA beyond the Llama family, we extend the evaluation to four additional LLMs: Baichuan2-7B, Mistral-7B, Qwen2.5-7B, and Qwen3-8B. All models are compressed at the same pruning ratio of **25%** and evaluated on the same multiple-choice benchmarks as in the main experiments. The results are reported in Table 11.

ReLoNA achieves competitive performance across different LLM families, consistently outperforming most representative compression baselines in terms of average performance. In particular,

Table 11: Comparison of compression performance across different LLM families. The results are evaluated on multiple language benchmarks under the same pruning ratio of 25%.

Methods	Benchmark													AVG.	RP.
	CoQA	C3	CMNLI	CMMLU	CHID	RACE-M	RACE-H	PIQA	MMLU	HellaSwag	SIQA	WSC	BoolQ		
Baichuan2-7B															
Dense	63.39	62.96	33.14	56.90	83.10	50.97	52.72	74.81	54.74	64.97	42.99	43.27	61.53	57.35	100.00%
LLM-Pruner	30.79	37.10	31.72	21.94	21.41	61.59	35.34	38.95	41.35	49.60	25.26	23.93	69.03	37.54	65.46%
SliceGPT	48.40	37.26	34.92	26.25	26.73	66.05	54.67	41.10	37.50	48.50	25.69	31.14	28.70	38.99	67.99%
LaCo	45.78	51.40	33.01	27.78	80.50	24.23	24.39	68.17	26.84	49.29	39.56	49.04	59.08	44.54	77.66%
ShortGPT	45.86	47.12	33.00	44.38	65.34	22.14	24.59	63.93	40.31	48.44	40.89	36.54	62.87	44.26	77.18%
ReplaceMe	50.45	51.56	33.84	55.51	54.33	43.73	46.37	66.70	54.15	50.78	42.12	36.54	61.50	49.81	86.85%
ReLoNA (w/o OAR)	41.20	50.08	33.00	49.14	67.43	32.10	32.99	63.55	49.89	47.24	41.97	36.54	60.98	46.62	81.29%
ReLoNA (w/ OAR)	52.74	53.15	32.99	51.50	77.54	34.40	35.05	68.12	50.27	53.85	40.07	36.54	62.81	**49.92**	**87.04%**
Mistral-7B															
Dense	74.45	57.92	33.00	44.37	43.12	79.25	75.24	82.10	62.57	76.84	52.05	67.31	83.24	63.96	100.00%
LLM-Pruner	57.41	49.42	32.06	53.97	51.49	68.06	58.71	47.90	64.42	77.89	29.51	39.05	36.27	51.24	80.11%
SliceGPT	55.20	41.21	32.06	48.33	42.65	67.68	50.71	44.32	54.81	60.21	26.93	38.65	22.45	45.02	70.39%
LaCo	29.07	31.73	32.99	24.65	25.90	26.67	27.10	53.65	23.59	30.97	37.67	36.54	58.53	33.77	52.80%
ShortGPT	57.82	43.62	39.43	36.84	27.05	36.77	33.05	69.21	54.30	57.35	44.93	63.46	81.62	49.65	77.63%
ReplaceMe	55.28	48.00	36.28	40.28	22.62	66.09	58.43	68.50	57.90	56.11	46.32	53.85	79.57	53.02	82.90%
ReLoNA (w/o OAR)	49.55	50.19	32.06	27.84	29.99	35.93	33.22	71.82	44.73	52.49	46.01	37.50	76.57	45.22	70.70%
ReLoNA (w/ OAR)	60.69	47.29	32.93	41.14	31.04	66.92	58.72	76.17	58.24	65.30	46.11	40.38	81.41	**54.33**	**84.94%**
Qwen2.5-7B															
Dense	67.65	69.59	33.26	81.75	85.06	90.46	86.56	79.38	75.52	71.10	56.19	56.73	75.93	71.48	100.00%
LLM-Pruner	51.52	61.37	37.71	41.71	41.42	69.70	55.75	48.87	36.54	63.82	33.22	37.22	71.25	50.01	69.96%
SliceGPT	48.65	48.16	32.98	22.35	22.24	68.12	52.80	43.50	41.35	62.42	27.05	29.06	40.74	41.49	58.04%
LaCo	29.48	32.00	31.50	24.93	26.38	21.59	21.67	62.02	26.50	38.37	37.41	41.35	60.64	34.91	48.84%
ShortGPT	46.27	46.63	34.31	25.05	76.98	21.66	21.53	73.12	28.05	51.65	41.50	36.54	59.33	43.28	60.55%
ReplaceMe	48.57	51.51	34.48	29.30	77.45	28.83	27.19	72.47	30.95	51.04	43.14	36.54	55.78	45.17	63.20%
ReLoNA (w/o OAR)	36.28	50.08	33.53	31.09	47.45	42.97	44.65	63.66	41.98	42.91	44.73	59.62	63.03	46.31	64.78%
ReLoNA (w/ OAR)	47.34	55.34	32.99	41.13	66.57	54.25	54.57	70.40	46.47	51.79	45.75	36.54	62.35	**51.19**	**71.61%**
Qwen3-8B															
Dense	71.99	74.68	36.33	77.35	78.88	90.88	87.42	78.73	76.67	68.62	50.41	70.19	85.81	72.92	100.00%
LLM-Pruner	56.02	55.51	35.41	52.92	54.03	68.17	50.54	44.42	39.42	73.52	41.21	44.69	63.25	52.24	71.64%
SliceGPT	55.61	48.82	33.00	22.35	21.93	66.59	52.42	44.17	36.54	66.97	31.91	34.35	28.19	41.76	57.27%
LaCo	44.47	54.58	38.12	51.98	48.00	72.01	67.50	61.97	46.02	41.06	41.71	36.54	63.76	51.36	70.44%
ShortGPT	51.84	49.59	33.84	27.65	67.82	27.65	26.93	71.55	27.87	50.35	43.96	38.46	63.43	44.69	61.28%
ReplaceMe	52.66	45.70	33.13	28.96	31.45	43.94	36.96	70.08	28.77	44.11	43.14	36.54	59.11	42.66	58.50%
ReLoNA (w/o OAR)	39.89	48.22	40.56	60.74	45.91	41.16	37.94	64.36	52.63	52.46	42.94	38.46	62.17	48.26	66.19%
ReLoNA (w/ OAR)	45.54	55.45	45.98	67.75	42.33	79.39	68.12	63.76	57.59	45.93	42.68	58.65	62.51	**56.59**	**77.61%**

ReLoNA remains effective across both Llama-independent architectures and different model families, demonstrating the generality of its layer selection and operator replacement strategy beyond the models evaluated in the main experiments.

D.5 Progressive Compression Dynamics

To further investigate whether ReLoNA can track the accumulated degradation during progressive compression, we record the degradation risk and the corresponding performance degradation as the pruning process proceeds. The normalized degradation is computed with respect to the actual degradation at 16-layer pruning, which serves as the reference point. As shown in Figure 9, the degradation risk closely follows the evolution of the measured degradation across different compression stages. Moreover, the Pearson correlation coefficients between degradation risk and degradation metrics consistently exceed 0.9, demonstrating a strong linear relationship between the estimated compression risk and the actual performance loss. These results indicate that LNA provides not only effective layer ranking but also an accurate estimation of cumulative degradation throughout progressive compression.

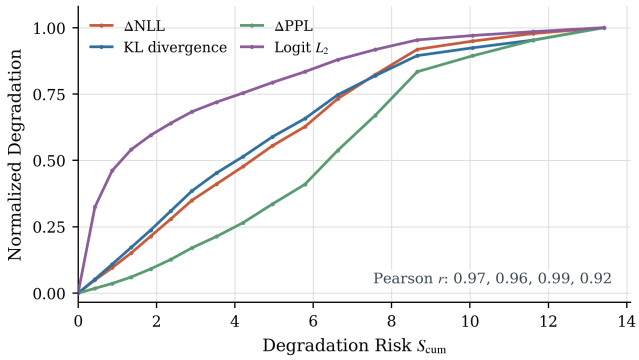


Figure 9: Co-evolution of degradation risk and actual degradation metrics during progressive pruning of Llama-2-7B.