LAYER PRUNING AS A NON-COOPERATIVE GAME: CONDITIONAL EXCHANGES FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Structured layer pruning reduces the computational and memory costs of large language models, but the effect of removing a layer depends on which other layers remain. Static rankings may overlook this dependence, while surrogate-based approaches introduce an additional prediction stage. We formulate fixed-budget layer selection as a finite non-cooperative game with a shared utility and develop a conditional exchange procedure. Starting from standalone layer scores, the method evaluates exchanges that restore one removed layer and remove one retained layer. At each iteration, it accepts the exchange that most reduces the Kullback–Leibler divergence between dense and pruned output distributions on calibration data. The search terminates when no permitted exchange improves the objective beyond a numerical tolerance, without weight updates, surrogate training, or Monte Carlo Shapley estimation. Experiments across multiple model families, pruning budgets, and language-modeling benchmarks show lower perplexity than the comparison method in most evaluated configurations, while zero-shot results reveal budget-dependent trade-offs. Matched inference measurements show comparable latency, throughput, and allocated GPU memory. In a separate wall-clock evaluation on a seven-billion-parameter model, our method is about $2.27\times$ faster overall than the comparison method across four pruning budgets. These results support conditional exchange as a practical approach to selecting layers while accounting for their interactions.

Keywords: structured pruning; common-interest game; exact potential game; exchange stability; large language models; KL divergence; layer pruning

1 INTRODUCTION

Large language models place substantial demands on inference memory and computation. Structured pruning reduces these demands by removing complete computational units, such as attention heads, neurons, or transformer blocks (Ma et al., 2023; Men et al., 2024). Layer pruning is particularly straightforward to deploy: it produces a shallower network while preserving the tensor shapes and dense operations of the retained blocks. Weight-pruning methods such as Wanda (Sun et al., 2024) induce sparsity within matrices, while SliceGPT (Ashkboos et al., 2024) reduces their dimensions. Studies of layer redundancy also demonstrate substantial depth reduction with subsequent fine-tuning (Gromov et al., 2025). Here, we focus on selecting complete layers for removal while keeping the retained weights frozen.

This choice depends on the layers that remain. Removing a transformer block changes the representations passed to subsequent blocks, so the effect of another deletion can change with the current network configuration. A layer that appears dispensable when removed alone may become important after other layers have been removed. Standalone importance scores provide a useful starting point, but a fixed ranking cannot revise its choices in response to these interactions. This motivates evaluating candidate removals conditionally on the current pruned model.

Cooperative-game approaches address interactions through coalition-based attribution. A recent method samples masked subnetworks and trains a surrogate to estimate their values for Shapley-style

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.

layer scoring (Ding et al., 2026). This raises a more direct question for fixed-budget pruning: can we improve the selected subset by evaluating feasible changes to it, without fitting an auxiliary predictor or estimating global layer attributions?

We formulate layer selection as a finite non-cooperative game with a shared utility. Each of K selectors chooses a distinct layer to remove, and all selectors seek to minimize the Kullback–Leibler (KL) divergence between dense and pruned next-token distributions on calibration data. A feasible change by one selector restores its selected layer and removes a currently retained layer, keeping the removal budget fixed. This construction connects unilateral improvement to conditional layer exchange: an exact equilibrium corresponds to a set with no improving one-for-one exchange. The guarantee is local to this neighborhood, rather than global subset optimality.

The resulting algorithm starts from the K lowest-damage standalone deletions. It evaluates every one-for-one exchange and accepts the candidate that most reduces calibration KL, stopping when no improvement exceeds a numerical tolerance. Strict improvement on a finite set of configurations gives finite termination under a fixed objective. The procedure uses frozen model weights and requires neither surrogate training nor Monte Carlo Shapley estimation. Its practical contribution is a direct exchange search; the game formulation specifies the feasible moves and characterizes their stopping condition.

Experiments evaluate language modeling and zero-shot tasks across several transformer models and removal budgets. Our method achieves lower perplexity than the surrogate-based baseline in most evaluated configurations, with budget-dependent trade-offs on downstream tasks. Figure 1 shows the local LLaMA-2-13B comparison on three corpora. Matched generation measurements show comparable latency, throughput, and allocated GPU memory. In a separate LLaMA-2-7B wall-clock experiment covering four removal budgets, our method is about $2.27\times$ faster overall under the measured protocol. Ablations examine how improvements in calibration KL transfer to held-out perplexity.

Our contributions are:

1. A fixed-budget non-cooperative formulation that relates feasible unilateral deviations to layer exchanges and characterizes exact equilibria as one-exchange-stable removal sets.
2. A KL-based best-improvement procedure with standalone initialization, finite termination under a fixed objective, and an explicit numerical stopping criterion.
3. An empirical evaluation of prediction quality, generation efficiency, and pruning time, together with ablations that distinguish calibration improvement from held-out performance.

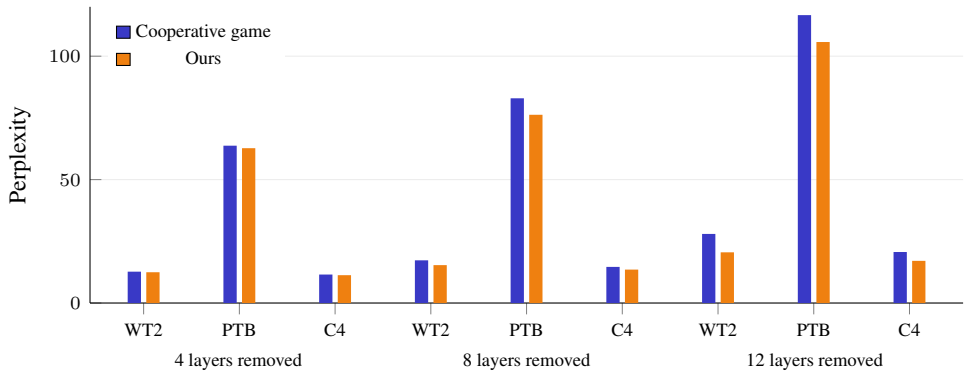

Figure 1: Perplexity on LLaMA-2-13B-hf after removing 4, 8, or 12 transformer layers. Bars compare cooperative game (Ding et al., 2026) with ours on WikiText-2 (WT2), PTB, and C4. Lower is better.

2 METHOD

We study layer removal at a fixed budget. The game is defined over feasible removal sets, and its unilateral moves are budget-preserving exchanges. This representation makes the strategic interaction

explicit while matching the candidate changes evaluated by the implementation. Appendix B provides a schematic overview (Figure 2).

2.1 Players, Actions, and Shared Utility

Let $\mathcal{L} = \{1, \dots, N\}$ index the layers and let K be the removal budget, with $1 \le K < N$. We model the K removal slots as selector players,

$$\mathcal{P} = \{1, \dots, K\}. \tag{1}$$

Each player r selects a layer $a_r \in \mathcal{L}$. A profile $\mathbf{a} = (a_1, \dots, a_K)$ is feasible if no two players select the same layer:

$$\mathcal{A}_K = \left\{ \mathbf{a} \in \mathcal{L}^K : a_r \neq a_s \text{ for all } r \neq s \right\}. \tag{2}$$

The induced removal set is

$$A(\mathbf{a}) = \{a_1, \dots, a_K\}, \qquad |A(\mathbf{a})| = K. \tag{3}$$

The selector labels are an analytical device: different assignments of the same K layers to selector slots represent the same pruned model.

Let $X_m = (x_{m,1}, \dots, x_{m,T})$ denote calibration sequence m, and let $p_0(\cdot \mid x_{m,1:t})$ and $p_A(\cdot \mid x_{m,1:t})$ be the dense and pruned next-token distributions at position t. For M calibration sequences, the measured divergence is

$$D(A) = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{T} \sum_{t=1}^{T} D_{\mathrm{KL}} \left(p_0(\cdot \mid x_{m,1:t}) \| p_A(\cdot \mid x_{m,1:t}) \right), \qquad v(A) = -D(A), \tag{4}$$

where all evaluated sequences have the same length T in our experiments. Thus, $D(A)$ is the sequence-average of the within-sequence mean over token positions; lower D (equivalently, larger v) is better. Every selector receives the same utility:

$$U_r(\mathbf{a}) = v(A(\mathbf{a})), \qquad r \in \mathcal{P}. \tag{5}$$

The model weights are frozen during selection, and the dense distributions are computed once and reused for candidate evaluations.

Implementation representation. The implementation represents the removal set by one binary gate per layer. For layer i, the gate $s_i = 1$ retains the layer and $s_i = 0$ bypasses its transformation. If $F_i(h_i)$ is the layer output, the gated residual update is

$$h_{i+1} = h_i + s_i \big(F_i(h_i) - h_i \big), \qquad s_i = \mathbb{I}[i \notin A]. \tag{6}$$

This is the binary specialization of the forward hook used by the code. The selector profile is not separately instantiated: its induced set A is converted directly to these N binary gates.

2.2 Feasible Deviations and Equilibrium

Given a feasible profile $\mathbf{a}$, selector r may replace its current choice $i = a_r$ with any layer not selected by the other $K - 1$ selectors. Changes to a currently unselected layer give the nontrivial feasible deviations:

$$j \in \mathcal{L} \setminus A(\mathbf{a}). \tag{7}$$

The resulting exchange is

$$A_{i \to j} = (A(\mathbf{a}) \setminus \{i\}) \cup \{j\}, \qquad i \in A(\mathbf{a}),\ j \notin A(\mathbf{a}). \tag{8}$$

We call a feasible profile a pure generalized Nash equilibrium (GNE) when no selector can improve its utility through a unilateral deviation that preserves joint feasibility. For a profile $\mathbf{a}^*$, write $A^* = A(\mathbf{a}^*)$. In terms of this induced removal set, the condition is

$$v(A^*) \ge v\big((A^* \setminus \{i\}) \cup \{j\} \big) \quad \forall i \in A^*, \quad \forall j \in \mathcal{L} \setminus A^*. \tag{9}$$

Algorithm 1 Fixed-Budget KL Exchange Search

Require: Dense model, layer set $\mathcal{L}$, budget K, calibration sequences, tolerance τ
Ensure: Pruned model and removal set A

Freeze model weights; compute dense distributions p_0
for all $i \in \mathcal{L}$ **do**
    Evaluate $\kappa_i = D(\{i\})$
end for
$A \leftarrow$ the K layers with smallest κ_i; evaluate $D(A)$
stopped $\leftarrow$ false
repeat
    $\Delta_{\max} \leftarrow 0$; $A_{\text{best}} \leftarrow A$
    for all $i \in A$ and $j \in \mathcal{L} \setminus A$ **do**
        $A' \leftarrow (A \setminus \{i\}) \cup \{j\}$; evaluate $\Delta = D(A) - D(A')$
        if $\Delta > \Delta_{\max}$ **then**
            $\Delta_{\max} \leftarrow \Delta$; $A_{\text{best}} \leftarrow A'$
        end if
    end for
    if $\Delta_{\max} > \tau$ **then**
        $A \leftarrow A_{\text{best}}$
    else
        stopped $\leftarrow$ true
    end if
until stopped
Physically remove the layers in A and return the resulting model

Proposition 1 (equilibrium characterization). A feasible selector profile is a pure GNE if and only if its induced removal set is one-exchange stable, as stated in Eq. 9. Moreover, $\Phi(\mathbf{a}) = v(A(\mathbf{a}))$ is an exact potential for the selector game.

Proof. Each removed layer i is selected by one player r. Replacing $a_r = i$ with an unselected layer j induces exactly the set in Eq. 8. Since every player's utility equals $v(A)$,

$$U_r(\mathbf{a}^{r\to j}) - U_r(\mathbf{a}) = v(A_{i\to j}) - v(A) = \Phi(\mathbf{a}^{r\to j}) - \Phi(\mathbf{a}). \tag{10}$$

Thus unilateral profitable deviations are precisely improving one-layer exchanges, and each utility change equals the potential change.

The exact-potential identity follows directly from assigning every selector the same utility. The game-theoretic content used by the algorithm is the fixed-budget feasible-deviation structure: unilateral selector moves correspond one-to-one with budget-preserving layer exchanges.

2.3 Initialization and Best-Improvement Search

We initialize the search by measuring each layer's standalone removal damage under the dense configuration:

$$\kappa_i = D(\{i\}), \qquad i \in \mathcal{L}. \tag{11}$$

The initial set A_0 contains the K layers with the smallest κ_i. This score is used only to initialize the search. At a current set A, the utility gain from exchanging removed layer i for retained layer j is

$$\Delta_{i\to j}(A) = v(A_{i\to j}) - v(A) = D(A) - D(A_{i\to j}). \tag{12}$$

Unlike the standalone score, this gain depends on the complete current set of removed layers.

Algorithm 1 evaluates the full one-exchange neighborhood, chooses the candidate with the largest KL reduction, and accepts it only if the reduction exceeds the numerical tolerance τ. The implementation uses $\tau = 10^{-9}$.

Proposition 2 (finite termination). For fixed, repeatable objective values and any $\tau \geq 0$, Algorithm 1 terminates after finitely many accepted updates. Its output satisfies

$$D(A^*) - D(A^*_{i\to j}) \leq \tau \quad \forall i \in A^*, \quad \forall j \in \mathcal{L} \setminus A^*. \tag{13}$$

For $\tau = 0$, this is exactly a pure GNE of the game.

Proof. There are $\binom{N}{K}$ feasible removal sets. Every accepted update decreases D by more than τ, so the search cannot revisit a previous set. There can therefore be at most $\binom{N}{K} - 1$ accepted updates. At termination, no neighbor reduces D by more than τ, which gives Eq. 13. When $\tau = 0$, Proposition 1 identifies this condition with a pure GNE.

Implementation Details. The code stores a binary removal mask and implements each candidate with residual gates from Eq. 6; it does not instantiate the selector labels. For one-exchange search, a round evaluates at most $K(N-K)$ candidate sets, requiring $O(K(N-K))$ candidate model evaluations. For exchanges of size at most q, where $q \leq \min(K, N-K)$, the candidate count per round is

$$\sum_{r=1}^{q} \binom{K}{r} \binom{N-K}{r}. \tag{14}$$

The reported primary method follows unilateral selector deviations. The $q > 1$ ablation additionally searches coordinated changes by multiple selectors; it is a larger neighborhood and its stopping guarantee is with respect to that neighborhood. Neither search guarantees a global minimum over all $\binom{N}{K}$ removal sets.

3 Experiments

3.1 Experimental Setup

Models and pruning budgets. We evaluate LLaMA-2-7B (Touvron et al., 2023), Vicuna-7B-v1.3, and Meta-LLaMA-3-8B at removal budgets of 3, 6, 9, and 12 layers. For LLaMA-2-13B, we compare removal of 4, 8, and 12 layers.

Benchmarks and baselines. We measure perplexity on WikiText-2 (Merity et al., 2016), PTB (Marcus et al., 1993), and C4 (Raffel et al., 2020). Downstream evaluation comprises nine zero-shot tasks for LLaMA-2-7B, eight-task comparisons for the larger models, and ANLI (Nie et al., 2020) for adversarial reasoning. MMLU (Hendrycks et al., 2021) is reported separately from the eight-task means. Appendix D gives the task inventory and dataset references. Comparisons include SliceGPT (Ashkboos et al., 2024), SLEB (Song et al., 2024), Shortened-LLaMA (Kim et al., 2024), ShortGPT (Men et al., 2024), and the cooperative-game method (Ding et al., 2026). We identify published baseline values and local checkpoint measurements separately throughout; Appendix C.1 documents their evaluation settings and provenance.

Pruning and evaluation protocol. Selection uses ten 128-token BookCorpus (Zhu et al., 2015) sequences, seed 42, frozen FP16 weights, standalone KL initialization, and one-for-one exchanges with improvement tolerance 10^{-9}. Local perplexity evaluation uses 128-token windows with a single initial BOS token. Held-out evaluation data are separate from calibration data and do not guide layer selection. Quality comparisons describe the evaluated configurations; timing repeats do not estimate variability across calibration seeds. Cross-entropy objective variants are reported separately in the appendix.

3.2 Language Modeling

Table 1 reports perplexity across three model families and four removal budgets. Our method achieves lower perplexity than the cooperative-game method in 30 of 36 dataset–budget comparisons. This comparison concerns the two game-based methods; the table also shows how both compare with the other pruning approaches. On LLaMA-2-13B, our method achieves lower perplexity in all nine locally evaluated comparisons across the three datasets and removal budgets (Table 2). Table 4 provides the separate local LLaMA-2-7B comparison, and Appendix E.1 reports additional compression results.

Table 1: Perplexity on WikiText-2, PTB, and C4 for LLaMA-2-7B-hf, Vicuna-7B-v1.3, and Meta-LLaMA-3-8B after removing 3, 6, 9, or 12 layers. Lower is better; bold indicates the lowest value within each model, dataset, and removal budget.

Model	Method	Remove 3			Remove 6			Remove 9			Remove 12		
		WT2	PTB	C4	WT2	PTB	C4	WT2	PTB	C4	WT2	PTB	C4
LLaMA-2-7B-hf	SliceGPT	108.0990	131.3884	103.9473	212.8867	219.2298	191.5134	291.8482	293.8186	257.7473	393.8880	365.7072	343.4214
	SLEB	14.2428	52.9183	12.9682	19.4676	63.8317	16.3933	27.4537	79.4398	21.3809	58.1194	135.1317	43.8708
	Shortened-LLaMA	16.6515	54.5982	13.8046	36.3702	105.2407	29.2243	81.9615	196.6155	61.8678	304.5240	486.6280	252.4593
	ShortGPT	16.6515	54.5982	13.5906	36.3702	105.2407	29.2243	81.9615	196.6155	61.8678	157.9850	295.1548	98.8645
	Cooperative game	14.6949	53.7517	12.9682	18.8686	**61.8678**	16.1392	24.6093	76.9957	20.7231	**38.1157**	**105.2407**	28.7712
	Ours	**13.6990**	**52.0190**	**12.2890**	**16.9720**	62.7290	**14.9530**	**23.3810**	**76.6220**	**18.6850**	38.4150	148.6100	**26.1280**
Vicuna-7B-v1.3	SliceGPT	151.4702	195.3507	133.0439	292.3765	339.8809	250.8779	401.3294	435.8758	343.5819	555.5617	566.6461	474.9028
	SLEB	**19.7741**	74.6268	**16.1392**	26.6090	87.2476	21.3809	38.1157	115.5843	30.6268	65.8579	196.6155	47.4357
	Shortened-LLaMA	20.4018	**70.1054**	17.4506	35.8063	98.8645	27.8860	67.9485	143.8470	48.1827	244.6919	356.0247	157.9850
	ShortGPT	23.1183	79.4398	18.0046	67.9485	143.8470	48.1827	67.9485	143.8470	48.1827	252.4593	518.0128	153.1243
	Cooperative game	20.7231	**70.1054**	17.7254	24.6093	81.9615	20.7231	37.5247	112.0281	29.2243	67.9485	209.2961	43.1907
	Ours	21.9220	72.8140	17.3070	**22.6480**	**80.6990**	**19.1680**	**31.5060**	**99.9300**	**25.4470**	**54.3050**	**127.8960**	**35.7800**
Meta-LLaMA-3-8B	SliceGPT	316.1936	307.9020	231.0099	447.8662	488.1207	316.6689	746.2088	802.4747	530.2491	1182.3573	1214.1042	830.1348
	SLEB	20.4018	38.1157	20.7231	33.6369	53.7517	30.1520	63.8317	115.5843	49.7122	126.9445	190.5663	90.0171
	Shortened-LLaMA	20.7231	37.5247	20.7231	79.4398	105.2407	50.4950	5928.3428	9774.1849	2048.7805	15138.5538	46630.0285	2113.8157
	ShortGPT	23.8522	41.2128	22.7599	84.5633	119.2533	61.8678	2549.7485	2714.1931	1364.7820	15138.5538	46630.0285	2113.8157
	Cooperative game	18.5761	**34.7047**	19.1658	25.3905	45.9763	24.9969	45.2635	70.1054	33.1155	304.5240	173.5126	65.8579
	Ours	**18.2680**	36.2090	**18.7150**	**25.2270**	**45.8540**	**24.8170**	**36.7250**	**64.0480**	**32.4900**	**69.3150**	**108.3660**	**57.9470**

Table 2: Locally evaluated LLaMA-2-13B perplexity. Bold marks the lower paired value.

Removed	Method	WikiText-2	PTB	C4
4	Cooperative game	12.7010	63.7403	11.5375
4	Ours	**12.4707**	**62.7255**	**11.2747**
8	Cooperative game	17.2960	82.9377	14.6329
8	Ours	**15.3497**	**76.2631**	**13.5408**
12	Cooperative game	27.9910	116.6358	20.6664
12	Ours	**20.5373**	**105.7390**	**17.0835**

Table 3: Zero-shot accuracy on nine tasks for LLaMA-2-7B-hf. The four pruned parameter groups correspond to 3, 6, 9, and 12 removed layers. Dense and baseline values are published results (Ding et al., 2026); Ours rows are local evaluations. HellaSwag (HeSw), ARC-challenge, and OpenBookQA use normalized accuracy; other tasks use raw accuracy. Higher is better. Bold identifies the higher score in each Cooperative game–Ours pair; other methods provide additional context.

Params	Method	PIQA	HeSw	ARC-e	ARC-c	OBQA	RACE	WSC273	LAMBADA	MMLU
6.7B	LLaMA-2-7B-hf	0.7807	0.7602	0.7630	0.4625	0.4420	0.3962	0.8059	0.7388	0.4177
6.1B	SliceGPT	0.6676	0.5299	0.4663	0.3020	0.3140	0.3397	0.8022	0.3152	0.2500
	SLEB	0.7644	0.7180	0.7138	0.3985	0.4080	0.3550	0.7839	0.6216	0.3084
	Shortened-LLaMA	0.7497	0.7298	0.7201	0.4360	0.4040	0.3799	0.8278	0.6301	0.3572
	ShortGPT	0.7573	0.7162	0.7104	0.4292	0.4040	0.3847	0.7692	0.6167	0.3510
	Cooperative game	0.7709	**0.7185**	**0.7155**	0.4096	**0.3940**	0.3694	**0.7912**	**0.6682**	**0.3663**
	Ours	**0.7715**	0.7181	0.6886	**0.4104**	0.3920	**0.3713**	0.7656	0.6338	0.3203
5.5B	SliceGPT	0.6077	0.4270	0.3590	0.2756	0.2700	0.3081	0.7473	0.2395	0.2441
	SLEB	0.7301	0.6656	0.6700	0.3951	0.3880	0.3502	0.7399	0.4477	0.2378
	Shortened-LLaMA	0.7095	0.6528	0.5934	0.3797	0.3740	0.3330	0.7692	0.4747	0.2589
	ShortGPT	0.7095	0.6528	0.5934	0.3797	0.3740	0.3330	0.7692	0.4747	0.2589
	Cooperative game	0.7372	**0.6719**	**0.6473**	**0.3729**	**0.3860**	**0.3646**	**0.7546**	0.4739	**0.2955**
	Ours	**0.7443**	0.6402	0.6044	0.3609	0.3740	0.3455	0.7070	**0.4964**	0.2609
4.9B	SliceGPT	0.5887	0.3856	0.3245	0.2619	0.2700	0.2947	0.7253	0.1846	0.2452
	SLEB	0.6910	0.5640	0.5947	0.3251	0.3520	0.3263	0.6850	0.3134	0.2372
	Shortened-LLaMA	0.6485	0.5617	0.4802	0.3276	0.3280	0.3225	0.7143	0.2915	0.3811
	ShortGPT	0.6485	0.5617	0.4802	0.3276	0.3280	0.3225	0.7143	0.2915	0.3811
	Cooperative game	0.7046	**0.5840**	**0.5821**	**0.3404**	0.3600	**0.3301**	**0.7033**	0.3427	**0.2725**
	Ours	**0.7095**	0.5623	0.5434	0.3336	**0.3620**	0.3062	0.6740	**0.3648**	0.2570
4.3B	SliceGPT	0.5718	0.3475	0.3077	0.2500	0.2560	0.2775	0.6923	0.1450	0.2493
	SLEB	0.6186	0.4665	0.4491	0.3020	0.3100	0.3024	0.5788	0.1527	0.2507
	Shortened-LLaMA	0.6023	0.4430	0.3611	0.3063	0.2760	0.2909	0.6081	0.0505	0.3373
	ShortGPT	0.5952	0.4371	0.4158	0.3003	0.3500	0.2900	0.6886	0.1100	0.3332
	Cooperative game	0.6659	0.4618	0.4937	**0.2876**	0.3260	0.2989	0.6117	0.1731	**0.2370**
	Ours	**0.6665**	**0.4667**	**0.4958**	0.2807	**0.3320**	**0.3033**	**0.6227**	**0.2150**	0.2342

3.3 Zero-Shot and Adversarial Evaluation

Table 3 reports all nine task accuracies for LLaMA-2-7B. Relative to the cooperative-game method, ours achieves higher PIQA accuracy at every budget and higher accuracy on seven of nine tasks

after twelve-layer removal. At this largest budget, ours also has the higher nine-task mean; the cooperative-game method has the higher mean at the three smaller budgets. For LLaMA-2-13B, ours achieves a higher eight-task mean at eight-layer removal in the local comparison and at twelve-layer removal against published results, while the cooperative-game method leads at four-layer removal. These eight-task means exclude MMLU, which is reported separately alongside the detailed results in the appendix.

The ANLI comparison shows a different pattern: ours has the higher round-average at one of four LLaMA-2-7B budgets, while the cooperative-game method leads at the other three and at all three paired LLaMA-2-13B budgets. Appendix D.1 reports the individual rounds. Layer selection therefore affects language modeling and downstream reasoning differently, with the preferred configuration depending on the task and removal budget.

3.4 Matched Perplexity and Generation Efficiency

To compare inference costs under the same workload, we benchmark both methods' LLaMA-2-7B checkpoints on one RTX 4500 Ada GPU using FP16, batch size one, a fixed prompt, and 128 generated tokens. We perform ten warm-ups followed by twenty timed generations. Table 4 reports latency, throughput, allocated GPU memory, and locally measured perplexity. The methods have similar inference costs: mean latency differs by less than 0.6% at every budget, and allocated memory is identical at the reported precision. Ours achieves lower WikiText-2 and C4 perplexity at all four budgets. PTB favors ours at three-layer removal and the cooperative-game method at the remaining budgets. Thus, the measured quality differences arise at comparable generation cost.

Table 4: Locally measured perplexity and generation efficiency for the Cooperative-game and Ours LLaMA-2-7B checkpoints. PPL columns are WikiText-2/PTB/C4. Latency and throughput are mean $\pm$ population standard deviation over 20 timed generations after 10 warm-ups. Both methods use the same benchmark settings and RTX 4500 Ada GPU. Allocated memory is the `torch.cuda.memory_allocated` reading in MiB. Lower PPL, latency, and memory are better; higher throughput is better. Bold marks lower paired PPL. Timing differences are treated as descriptive, without significance claims.

Removed	Params	Method	WT2 PPL	PTB PPL	C4 PPL	Latency (s)	Throughput (tokens/s)	Allocated (MiB)
3	6.13B	Cooperative game	14.6862	53.7773	12.8939	4.5604 ± 0.0556	28.07 ± 0.33	11772
		Ours	**13.6988**	**52.0189**	**12.2891**	4.5346 ± 0.0209	28.23 ± 0.13	11772
6	5.52B	Cooperative game	18.7759	**61.9097**	16.1296	4.0945 ± 0.0091	31.26 ± 0.07	10604
		Ours	**16.9716**	62.7292	**14.9525**	4.0834 ± 0.0058	31.35 ± 0.04	10604
9	4.92B	Cooperative game	24.6153	**76.1299**	20.6182	3.6464 ± 0.0011	35.10 ± 0.01	9442
		Ours	**23.3810**	76.6217	**18.6853**	3.6516 ± 0.0205	35.05 ± 0.19	9442
12	4.31B	Cooperative game	38.4436	**103.9357**	29.0377	3.2005 ± 0.0025	39.99 ± 0.03	8279
		Ours	**38.4150**	148.6104	**26.1276**	3.2002 ± 0.0025	40.00 ± 0.03	8279

3.5 Pruning Cost and Ablations

Across four LLaMA-2-7B removal budgets, our end-to-end pruning pipeline is about $2.27\times$ faster overall than the cooperative-game method's shared all-budget pipeline. Timing covers process startup, model loading, selection, and checkpoint saving. The comparison uses two successful paired repeats, with the methods running concurrently on separate GPUs. Appendix G gives the breakdown and protocol. The speedup applies to this complete workflow, including the shared preprocessing used by the cooperative-game method.

The standalone-ranking ablation tests the contribution of exchange refinement with the KL objective and calibration set held fixed. Refinement reduces calibration KL in every evaluated case and improves held-out C4 perplexity at all three tested budgets and WikiText-2 at two. Standalone ranking retains lower PTB perplexity. These results show that exchanges can improve both the calibration objective and held-out prediction quality, although the gains vary across corpora. Appendix F.1 reports the full comparison, together with calibration-source and initialization analyses; cross-entropy variants are presented separately.

DISCUSSION

The results highlight both the value and the limits of selecting layers conditionally on the current network configuration. Our method improves perplexity across many evaluated settings, including all three language-modeling datasets at the paired LLaMA-2-13B removal budgets. On LLaMA-2-7B, however, the locally matched results favor our method consistently on WikiText-2 and C4, while PTB and zero-shot accuracy show budget-dependent trade-offs. These differences are consistent with the distinction between the calibration objective and the evaluation metrics: reducing dense-to-pruned KL divergence on calibration data does not guarantee lower held-out perplexity or higher downstream accuracy. The standalone-ranking ablation likewise shows that exchange refinement need not improve every test metric, even when it improves the calibration objective.

The efficiency measurements identify where the practical benefit occurs. At matched removal budgets, the LLaMA-2-7B checkpoints have comparable generation latency, throughput, and allocated GPU memory (Table 4). This is consistent with their shared architecture and retained depth. In the separate pruning-time evaluation, our method is about $2.27\times$ faster overall than the surrogate-based baseline across four removal budgets. This comparison includes separate runs for our four budgets and the baseline's shared all-budget pipeline; the timing boundary and repeat statistics are given in Appendix G. Thus, the measured advantage is in constructing the pruned models while maintaining comparable inference costs at a given depth.

The game formulation gives a precise interpretation of the stopping condition. Because all selectors share the same utility, an improving exchange also improves the game's potential. With exact evaluation and zero tolerance, a removal set admitting no improving permitted exchange corresponds to a feasible pure equilibrium; the implementation uses a numerical tolerance. The solver implements local search over fixed-size removal sets, and this interpretation does not imply global subset optimality or explain the empirical gains by itself. Its practical value is a clearly defined objective, feasible neighborhood, and stopping rule for conditional layer selection.

LIMITATIONS

Our evaluation focuses on transformer-layer pruning for language modeling and zero-shot tasks. The calibration and initialization ablations examine sensitivity within this setting; broader evaluation across independently sampled calibration sets, architectures, and task domains would further characterize generalization. The search guarantees stability within the permitted exchange neighborhood rather than global subset optimality, and improvements in calibration KL need not translate uniformly to every downstream metric. Each search round evaluates up to $K(N-K)$ candidate exchanges for N layers and removal budget K, so search cost depends on model depth, budget, and the number of rounds. The measured pruning-time advantage and inference efficiency apply to the tested LLaMA-2-7B configuration; larger models and different hardware or decoding workloads require separate measurements. These questions motivate further study of calibration selection and search efficiency at greater scale.

4 CONCLUSION

We introduced a non-cooperative formulation of fixed-budget layer pruning and a conditional exchange procedure that refines standalone layer scores using dense-to-pruned KL divergence. The method selects layers without weight updates, surrogate training, or Monte Carlo Shapley estimation and terminates at a configuration stable to permitted exchanges within a numerical tolerance. Experiments show perplexity improvements across many evaluated configurations, alongside dataset- and budget-dependent downstream trade-offs. Matched checkpoints have comparable inference latency, throughput, and allocated GPU memory, while our method is about $2.27\times$ faster overall in the four-budget LLaMA-2-7B pruning-time comparison. These findings support evaluating layer exchanges in the current network configuration as a practical approach to structured LLM compression.

Ethics Statement. This work uses publicly available language-model checkpoints and benchmark datasets. We do not collect human data or release personally identifiable information. The intended

contribution is to reduce the computational and memory cost of model inference; users remain responsible for evaluating the behavior and risks of any compressed model in their application.

Reproducibility Statement. The paper and appendix specify the model checkpoints, calibration data, sequence lengths, pruning budgets, objective, exchange rule, evaluation datasets, and hyperparameters. The complete selection algorithm and all reported experimental protocols are documented in the paper and appendices.

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. https://openreview.net/forum?id=vXxardq6db.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about Physical Commonsense in Natural Language. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7432–7439, 2020. https://ojs.aaai.org/index.php/AAAI/article/view/6239.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge. *arXiv preprint arXiv:1803.05457*, 2018. https://arxiv.org/abs/1803.05457.

Xuan Ding, Pengyu Tong, Ranjie Duan, Yunjian Zhang, Rui Sun, and Yao Zhu. Pruning as a Cooperative Game: Surrogate-Assisted Layer Contribution Estimation for Large Language Models. In *International Conference on Learning Representations*, 2026. https://openreview.net/forum?id=AzgxJ7ZGCX.

Elias Frantar and Dan Alistarh. SparseGPT: Massive Language Models Can be Accurately Pruned in One-Shot. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 10323–10337. PMLR, 2023. 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. In *International Conference on Learning Representations*, 2023. https://arxiv.org/abs/2210.17323.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The Unreasonable Ineffectiveness of the Deeper Layers. In *International Conference on Learning Representations*, 2025. https://arxiv.org/abs/2403.17887.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring Massive Multitask Language Understanding. In *International Conference on Learning Representations*, 2021. https://openreview.net/forum?id=d7KBjmI3GmQ.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened LLaMA: Depth Pruning for Large Language Models with Comparison of Retraining Methods. *arXiv preprint arXiv:2402.02834*, 2024. https://arxiv.org/abs/2402.02834.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale ReAding Comprehension Dataset From Examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 785–794, 2017. 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*, 2012. https://cs.nyu.edu/~davise/papers/WinogradSchemas/WS.html.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-Pruner: On the Structural Pruning of Large Language Models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. https://papers.neurips.cc/paper_files/paper/2023/hash/44956951349095f74492a5471128a7e0-Abstract-Conference.html.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993. https://aclanthology.org/J93-2004/.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in Large Language Models are More Redundant Than You Expect. *arXiv preprint arXiv:2403.03853*, 2024. https://arxiv.org/abs/2403.03853.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer Sentinel Mixture Models. *arXiv preprint arXiv:1609.07843*, 2016. https://arxiv.org/abs/1609.07843.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018. https://aclanthology.org/D18-1260/.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. Adversarial NLI: A New Benchmark for Natural Language Understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4885–4901, 2020. https://aclanthology.org/2020.acl-main.441/.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, 2016. https://aclanthology.org/P16-1144/.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. https://jmlr.org/papers/v21/20-074.html.

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 *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 46136–46155. PMLR, 2024. https://proceedings.mlr.press/v235/song24f.html.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A Simple and Effective Pruning Approach for Large Language Models. In *International Conference on Learning Representations*, 2024. https://arxiv.org/abs/2306.11695.

Hugo Touvron et al. Llama 2: Open Foundation and Fine-Tuned Chat Models. *arXiv preprint arXiv:2307.09288*, 2023. https://arxiv.org/abs/2307.09288.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a Machine Really Finish Your Sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, 2019. https://aclanthology.org/P19-1472/.

Yukun Zhu, Ryan Kiros, Rich 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. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015. https://openaccess.thecvf.com/content_iccv_2015/html/Zhu_Aligning_Books_and_ICCV_2015_paper.html.

APPENDIX

A USAGE OF LARGE LANGUAGE MODELS

Large language models assisted with language editing, manuscript organization, code drafting and debugging, and suggestions for experimental analysis. The authors are responsible for the final manuscript, the implementation, the validity of the reported experiments, and the interpretation of their results.

B METHOD DETAILS

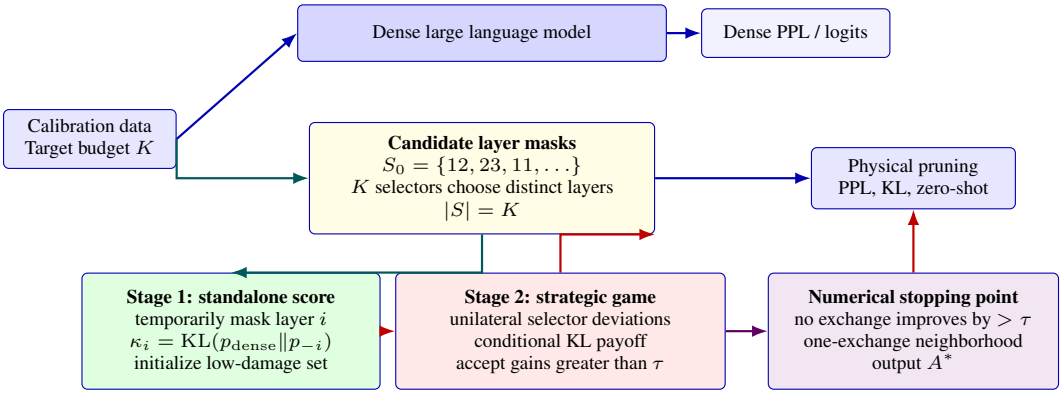

Figure 2: Framework of our fixed-budget common-interest pruning game. Standalone KL scores initialize K removal selectors. Each unilateral selector deviation replaces one removed layer with one retained layer. Strict best-improvement updates stop when no exchange improves utility by more than τ, which determines the layers physically pruned from the model.

B.1 STRATEGIC-EXCHANGE DIAGNOSTICS

Table 5 reports the selected removal sets and held-out perplexities for each calibration source and budget. Table 6 separately compares calibration KL before and after exchange refinement. The two measurements address different questions: accepted exchanges improve the selection objective, whereas held-out perplexity measures transfer to another corpus. All listed layer indices are zero-based; the method notation in Section 2 uses one-based indices.

Table 5: Strategic-exchange diagnostics for our non-cooperative KL method on LLaMA-2-7B-hf. Each row reports the final layer set and held-out WikiText-2/PTB/C4 perplexity for a completed calibration–budget pair.

Calibration	Removed	Final layer set	WT2	PTB	C4
BookCorpus-10	3	[11,12,24]	13.6988	52.0189	12.2891
BookCorpus-10	6	[9,10,11,12,21,24]	16.9716	62.7292	14.9525
BookCorpus-10	9	[8,9,10,11,12,21,24,25,27]	23.3810	76.6217	18.6853
BookCorpus-10	12	[8,9,10,11,12,13, 14,21,23,24,25,27]	38.4150	148.6104	26.1276
C4-10	3	[8,11,23]	14.5416	53.8311	12.5779
C4-10	6	[8,9,10,11,12,27]	17.5685	64.3925	14.9855
C4-10	9	[7,8,9,10,11,12,23,24,25]	24.6950	77.9085	18.9454
C4-10	12	[7,8,9,10,11,12,13,14,21,24,25,27]	37.7260	108.0630	25.8833
WikiText-2-10	3	[11,12,21]	14.1683	52.1239	12.5322
WikiText-2-10	6	[9,10,11,12,24,25]	17.0685	61.2535	14.6679
WikiText-2-10	9	[8,9,10,11,12,21,23,24,25]	23.4266	74.0569	18.7967
WikiText-2-10	12	[7,8,9,10,11,12,13,14,21,24,25,27]	37.7260	108.0630	25.8833

Table 6: Standalone ranking versus strategic exchange on LLaMA-2-7B-hf using the BookCorpus-10 calibration split. KL is dense-to-pruned calibration divergence; lower is better.

K	Static set	Final removal set	Static KL	Final KL	Improvement
3	[11,12,23]	[11,12,24]	0.12678	0.12629	0.00049
6	[10,11,12,14,23,24]	[9,10,11,12,21,24]	0.32041	0.27870	0.04171
9	[9,10,11,12,14, 21,23,24,25]	[8,9,10,11,12,21,24,25,27]	0.52854	0.49113	0.03741
12	[8,9,10,11,12,14, 21,22,23,24,25,26]	[8,9,10,11,12,13, 14,21,23,24,25,27]	1.13220	0.84648	0.28572

B.2 Hyperparameter Settings

Table 7 summarizes the primary KL-selection settings. All primary models use ten BookCorpus calibration sequences of length 128, seed 42, frozen FP16 weights, and one-for-one exchanges. The pruning budget varies with model depth. Quantization experiments use separate GPTQ calibration settings, stated below the table.

Table 7: Summary of hyperparameters used by our non-cooperative pruning experiments. The primary selection runs use direct KL evaluation, FP16 weights, seed 42, ten calibration examples of length 128, and one-for-one exchanges. The table covers the primary Transformer experiments.

Model	Layers	Samples	Seq. len.	Precision	Objective	q	Removal budgets
LLaMA-2-7B-hf	32	10	128	FP16	KL	1	$K \in \{3, 6, 9, 12\}$
Vicuna-7B-v1.3	32	10	128	FP16	KL	1	$K \in \{3, 6, 9, 12\}$
Meta-LLaMA-3-8B	32	10	128	FP16	KL	1	$K \in \{3, 6, 9, 12\}$
LLaMA-2-13B-hf	40	10	128	FP16	KL	1	$K \in \{4, 8, 12, 16\}$
GPTQ: 4-bit, 128 C4 calibration samples, sequence length 2048, group size 128, `desc_act = True`.							

B.3 Implementation and Stopping Rule

Algorithm 1 is the authoritative pseudocode. The implementation evaluates the mean dense-to-pruned KL in Eq. 4, using FP32 log-softmax calculations with frozen FP16 model weights. Each accepted update must reduce the objective by more than $\tau = 10^{-9}$. Termination therefore establishes tolerance stability over the searched neighborhood; it does not assert exact equilibrium or global optimality.

Standalone scores are ordered with NumPy's `argsort`. Candidate restore and remove indices are traversed in ascending order; equal-scoring exchange candidates retain the first encountered candidate. The code does not specify a separate stable tie-break for exactly equal standalone scores. Calibration data and dense distributions remain fixed during each search. Primary runs use one-for-one exchanges; the CE neighborhood ablation is explicitly identified as a separate objective variant.

Calibration sampling. BookCorpus uses the training split of `SamuelYang/bookcorpus`. The calibration-source ablations use the WikiText-2 raw training split or the first English C4 training shard. With random seed 42, the sampler chooses distinct document indices, rejects tokenized documents shorter than 128 tokens, and selects a uniformly sampled contiguous 128-token window from each accepted document. Tokenization uses the tokenizer's default special-token insertion before window sampling. The ten sampled windows are fixed throughout selection. These training splits are separate from the evaluation splits described in Appendix C.1.

C Experimental Details

C.1 Evaluation Protocol and Result Provenance

Perplexity. The local evaluator joins documents with two newlines, tokenizes without automatic special tokens, prepends one BOS token, and divides the stream into non-overlapping 128-token

windows, discarding the final incomplete window. WikiText-2 uses the `wikitext-2-raw-v1` test split. PTB uses `ptb.valid.txt` from the benchmark's standard text distribution. C4 uses the first 1,000 records of the first English validation shard. These settings describe local evaluation; published baseline values retain their source protocol. Equal removal counts should not be interpreted as a hardware-controlled comparison across structurally different compression methods.

Generation measurement. Both local LLaMA-2-7B checkpoint sets use the prompt "The Leaning Tower of Pisa is known for" (12 input tokens), batch size one, FP16, and exactly 128 generated tokens with KV caching. Sampling uses top-$k = 50$, top-$p = 0.95$, temperature 1, and seed 1234. Ten warm-ups precede twenty timed generations. CUDA events bracket generation and are synchronized before reading elapsed time. Throughput is output tokens divided by generation time. Timing dispersion is the population standard deviation across timed generations. Allocated memory is read after the generation loop and is not peak memory. Saved benchmark metadata identifies an RTX 4500 Ada GPU, PyTorch 2.7.1+cu126, CUDA 12.6, and driver 596.71.

Provenance. Table 1 contains published baseline values and local results for our method. Tables 2 and 4 evaluate both saved checkpoint sets locally. Published downstream rows are explicitly distinguished from the local 13B evaluations. All ANLI values in Table 12 are local outputs from `lm-evaluation-harness` 0.4.9.1 using zero demonstrations, full evaluation splits, batch size one, and FP16. The round sizes are 1,000, 1,000, and 1,200 examples; the reported average is the unweighted mean of round accuracies. The original experiment outputs retain their individual evaluator configurations.

C.2 Baseline Methods and Layer Selections

The main table includes published results for SLEB, ShortGPT, Shortened-LLaMA, and SliceGPT, together with the surrogate-based game comparator. SliceGPT compresses matrix dimensions, whereas the block-deletion methods remove layers; its reported values are retained as contextual results rather than a matched local removal protocol. The layer-index tables below retain published baseline selections and give our primary checkpoint manifests using zero-based indexing. Width-wise settings alone do not constitute a completed local comparison, so they are omitted.

Table 8: Depth-wise layer-index comparison on LLaMA-2-7B-hf. Baseline rows are reported settings; our final row is the selected layer set from the non-cooperative KL exchange method.

Method	Remove 3 layers	Remove 6 layers	Remove 9 layers	Remove 12 layers
SLEB	[15,14,24]	[15,14,24,13,25,22]	[15,14,24,13,25,22,8,23,12]	[15,14,24,13,25,22,8,23,12,29,21,7]
Shortened-LLaMA	[28,27,25]	[28,27,25,29,26,24]	[28,27,25,29,26,24,23,21,22]	[28,27,25,29,26,24,23,21,22,20,19,18]
ShortGPT	[25,27,26]	[25,27,26,24,28,29]	[25,27,26,24,28,29,23,22,21]	[25,27,26,24,28,29,23,22,21,19,30,20]
Ours	[11,12,24]	[9,10,11,12,21,24]	[8,9,10,11,12,21,24,25,27]	[8,9,10,11,12,13, 14,21,23,24,25,27]

Table 9: Depth-wise layer-index comparison on Vicuna-7B-v1.3. Baseline rows are reported settings; our final row is the selected layer set from the non-cooperative KL exchange method.

Method	Remove 3 layers	Remove 6 layers	Remove 9 layers	Remove 12 layers
SLEB	[7,27,24]	[7,27,24,17,22,10]	[7,27,24,17,22,10,26,13,14]	[7,27,24,17,22,10,26,13,14,8,9,25]
Shortened-LLaMA	[27,26,24]	[27,26,24,25,29,28]	[27,26,24,25,29,28,23,21,22]	[27,26,24,25,29,28,23,21,22,20,19,18]
ShortGPT	[27,28,26]	[27,28,26,29,25,24]	[27,28,26,29,25,24,23,22,21]	[27,28,26,29,25,24,23,22,21,30,20,19]
Ours	[23,24,27]	[9,10,11,22,24,26]	[8,9,10,11,14,22,24,26,29]	[8,9,10,11,12,13,14,22,23,24,26,29]

Table 10: Depth-wise layer-index comparison on Meta-LLaMA-3-8B. Baseline rows are reported settings; our final row is the selected layer set from the non-cooperative KL exchange method.

Method	Remove 3 layers	Remove 6 layers	Remove 9 layers	Remove 12 layers
SLEB	[7,25,18]	[7,25,18,23,28,26]	[7,25,18,23,28,26,14,13,22]	[7,25,18,23,28,26,14,13,22,10,8,21]
Shortened-LLaMA	[24,25,26]	[24,25,26,28,29,23]	[24,25,26,28,29,23,27,22,20]	[24,25,26,28,29,23,27,22,20,19,21,18]
ShortGPT	[25,27,26]	[25,27,26,24,28,23]	[25,27,26,24,28,23,22,29,21]	[25,27,26,24,28,23,22,29,21,20,19,18]
Ours	[10,11,26]	[8,10,11,19,25,26]	[7,8,9,10,11,12,22,25,26]	[7,8,9,10,11,12,13,14,22,23,26,28]

Table 11: Depth-wise layer-index comparison on LLaMA-2-13B-hf. Baseline rows are reported settings; our final row is the selected layer set from the non-cooperative KL exchange method.

Method	Remove 4 layers	Remove 8 layers	Remove 12 layers
SLEB	[15,28,31,29]	[15,28,31,29,22,13,18,8]	[15,28,31,29,22,13,18,8,30,27,19,33]
Shortened-LLaMA	[35,33,34,36]	[35,33,34,36,37,31,32,30]	[35,33,34,36,37,31,32,30,28,27,29,26]
ShortGPT	[33,31,32,30]	[33,31,32,30,34,35,29,28]	[33,31,32,30,34,35,29,28,27,36,37,26]
Ours	[10,27,29,31]	[10,11,12,24,27,29,31,33]	[9,10,11,12,22,25,26,27,29,30,31,35]

D Downstream Evaluation Details

The nine-task LLaMA-2-7B evaluation in Table 3 covers PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), ARC-easy and ARC-challenge (Clark et al., 2018), OpenBookQA (Mihaylov et al., 2018), RACE (Lai et al., 2017), WSC273 (Levesque et al., 2012), LAMBADA (Paperno et al., 2016), and MMLU (Hendrycks et al., 2021). The larger-model tables report eight-task results, with MMLU listed separately where available. Published comparisons and local checkpoint comparisons are identified separately; their means should not be interchanged.

D.1 Adversarial Reasoning

Table 12 reports local accuracy on ANLI R1–R3. The comparator has the higher unweighted round mean at three of four 7B budgets and all three 13B budgets. This evaluation therefore does not establish an adversarial-robustness advantage for our method.

Table 12: Local zero-shot ANLI accuracy. Average is the unweighted mean across rounds; higher is better. Both methods are evaluated at each listed budget.

Model	Removed	Method	R1	R2	R3	Average
7B	3	Cooperative game	0.3690	0.3520	0.3658	0.3623
7B	3	Ours	0.3360	0.3480	0.3650	0.3497
7B	6	Cooperative game	0.3460	0.3440	0.3683	0.3528
7B	6	Ours	0.3570	0.3410	0.3583	0.3521
7B	9	Cooperative game	0.3370	0.3320	0.3375	0.3355
7B	9	Ours	0.3330	0.3270	0.3300	0.3300
7B	12	Cooperative game	0.3410	0.3260	0.3308	0.3326
7B	12	Ours	0.3370	0.3310	0.3367	0.3349
13B	4	Cooperative game	0.3730	0.3940	0.3900	0.3857
13B	4	Ours	0.3500	0.3730	0.3808	0.3679
13B	8	Cooperative game	0.3410	0.3620	0.3542	0.3524
13B	8	Ours	0.3320	0.3350	0.3333	0.3334
13B	12	Cooperative game	0.3430	0.3740	0.3742	0.3637
13B	12	Ours	0.2970	0.3450	0.3333	0.3251

E Detailed Results of Zero-Shot Evaluation

We report per-task zero-shot results alongside Ding et al.'s published values (Ding et al., 2026). The published comparison method is labeled "Cooperative game" to distinguish it from the method introduced in this work. Our saved-checkpoint evaluations are labeled "Ours". Metric choices and evaluation provenance are specified in the table captions.

Table 15 contains paired local measurements at four and eight removed layers. A corresponding local eight-task result for the cooperative checkpoint at twelve removed layers is unavailable in this comparison; the twelve-layer comparison in Table 14 uses published baseline values. MMLU favors the cooperative checkpoint at both paired budgets, even though our eight-task mean is higher at eight removed layers. Thus, the task average does not establish a uniform downstream advantage.

Table 13: Detailed zero-shot downstream-task performance on Meta-LLaMA-3-8B. Remaining parameter sizes 7.4B, 6.7B, 6.1B and 5.4B correspond to removal of 3, 6, 9 and 12 layers. Cooperative game rows are published results; Ours rows are local evaluations on all eight tasks with no sample limit. HellaSwag, ARC-challenge and OpenBookQA use normalized accuracy; other entries use raw accuracy. All scores are shown without selective highlighting.

Params	Method	PIQA	HeSw	ARC-e	ARC-c	OBQA	Race	WSC273	LAMBADA
7.4B	SliceGPT	.6371	.5076	.4512	.2841	.2980	.3273	.7436	.3227
	SLEB	.7726	.6993	.7643	.4556	.4160	.3818	.7949	.6026
	Shortened-LLaMA	.7726	.7569	.7584	.4761	.4020	.3837	.8425	.6472
	ShortGPT	.7644	.7566	.7517	.4863	.4220	.3990	.8095	.4535
	Cooperative game	.7797	.7399	.7466	.4710	.4240	.3914	.8315	.6994
	Ours	0.7720	0.7212	0.7412	0.4556	0.4160	0.3732	0.7912	0.6957
6.7B	SliceGPT	.6034	.4440	.3830	.2585	.2740	.3110	.7106	.2827
	SLEB	.7394	.6352	.6856	.4215	.3660	.3770	.7143	.5447
	Shortened-LLaMA	.7155	.6419	.6557	.4411	.3760	.3627	.7399	.3776
	ShortGPT	.7247	.6815	.6263	.4352	.3700	.3569	.7399	.3433
	Cooperative game	.7459	.6473	.6662	.3840	.3560	.3292	.7582	.5581
	Ours	0.7454	0.6472	0.6658	0.3823	0.3560	0.3301	0.7546	0.5577
6.1B	SliceGPT	.5773	.3612	.3249	.2381	.2540	.2756	.6593	.2057
	SLEB	.6768	.5218	.5362	.3268	.3280	.2900	.6484	.2930
	Shortened-LLaMA	.5881	.2912	.3746	.3038	.2800	.2536	.5385	.0287
	ShortGPT	.6143	.3159	.3994	.3183	.3000	.2440	.6374	.0487
	Cooperative game	.7138	.5561	.5934	.3242	.3140	.2995	.6447	.2952
	Ours	0.7040	0.5258	0.5694	0.2918	0.3160	0.2900	0.6410	0.2851
5.4B	SliceGPT	.5571	.3106	.3114	.2312	.2560	.2584	.5897	.1225
	SLEB	.6246	.4063	.4125	.2884	.2780	.2766	.5934	.0768
	Shortened-LLaMA	.5876	.3774	.3737	.3038	.2800	.2699	.6337	.0349
	ShortGPT	.5876	.3774	.3737	.3038	.2800	.2699	.6337	.0349
	Cooperative game	.6474	.4530	.4583	.2807	.2940	.2842	.5971	.1149
	Ours	0.6360	0.4504	0.4604	0.2730	0.2900	0.2794	0.5568	0.2373

Table 14: Detailed zero-shot downstream-task performance on LLaMA-2-13B-hf. Remaining parameter sizes 11.7B, 10.5B, 9.2B and 7.9B correspond to removal of 4, 8, 12 and 16 layers, respectively. Cooperative game rows are published results; Ours rows are local evaluations. HellaSwag, ARC-challenge and OpenBookQA use normalized accuracy; other tasks use raw accuracy. All scores are shown without selective highlighting.

Params	Method	PIQA	HeSw	ARC-e	ARC-c	OBQA	Race	WSC273	LAMBADA
11.7B	SLEB	.7709	.7593	.7567	.4608	.4360	.4038	.8352	.6918
	Shortened-LLaMA	.7709	.7694	.7563	.4727	.4420	.4048	.8278	.6350
	ShortGPT	.7726	.7662	.7614	.4770	.4460	.4029	.8645	.6738
	SliceGPT	.6801	.5654	.5189	.3328	.3380	.3435	.8535	.3350
	Cooperative game	.7731	.7733	.7567	.4667	.4280	.3971	.8535	.7279
	Ours	0.7802	0.7588	0.7500	0.4471	0.4400	0.4077	0.8278	0.7316
10.5B	SLEB	.7470	.6851	.6902	.4070	.3840	.3636	.7729	.5595
	Shortened-LLaMA	.7291	.6536	.6246	.3857	.4320	.3455	.7949	.3062
	ShortGPT	.7399	.7240	.6852	.4377	.4120	.3828	.7985	.5224
	SliceGPT	.6219	.4525	.4015	.2910	.2960	.3177	.7875	.2608
	Cooperative game	.7443	.7257	.6890	.4155	.3840	.3694	.8095	.6389
	Ours	0.7666	0.7076	0.7146	0.4078	0.4020	0.3751	0.7949	0.6338
9.2B	SLEB	.7193	.6295	.6132	.3643	.3580	.3445	.7289	.4735
	Shortened-LLaMA	.6801	.5793	.5535	.3575	.3780	.3062	.7839	.2212
	ShortGPT	.6801	.5793	.5535	.3575	.3780	.3062	.7839	.2212
	SliceGPT	.5963	.4096	.3561	.2790	.2960	.2852	.7253	.2123
	Cooperative game	.7013	.6399	.5930	.3567	.3740	.3636	.7289	.5044
	Ours	0.7339	0.6362	0.6625	0.3660	0.3520	0.3311	0.7582	0.5319
7.9B	SLEB	.6823	.5485	.5311	.3328	.3300	.3081	.6557	.3569
	Shortened-LLaMA	.6230	.4718	.4524	.3174	.3560	.2775	.6630	.1304
	ShortGPT	.6230	.4718	.4524	.3174	.3560	.2775	.6630	.1304
	SliceGPT	.5642	.3309	.2976	.2585	.2500	.2612	.6520	.1285
	Cooperative game	.6567	.5265	.4566	.3046	.3220	.3263	.6520	.3419
	Ours	0.6942	0.5265	0.5244	0.2910	0.3320	0.2957	0.6520	0.2915

Table 15: Local paired zero-shot evaluations on LLaMA-2-13B-hf. All eight tasks use their full evaluation sets. HellaSwag, ARC-challenge and OpenBookQA use normalized accuracy; the other five tasks use raw accuracy. Mean is the unweighted eight-task mean. MMLU accuracy is reported separately and excluded from that mean. These local measurements are distinct from the published cooperative-game rows in Table 14.

Removed	Method	PIQA	HellaSwag	ARC-e	ARC-c	OBQA	RACE	WSC273	LAMBADA	Mean	MMLU
4	Cooperative game	0.7720	0.7732	0.7567	0.4650	0.4260	0.3962	0.8535	0.7281	0.6463	0.5195
4	Ours	0.7802	0.7588	0.7500	0.4471	0.4400	0.4077	0.8278	0.7316	0.6429	0.4631
8	Cooperative game	0.7437	0.7257	0.6890	0.4155	0.3840	0.3694	0.8095	0.6389	0.5970	0.4532
8	Ours	0.7666	0.7076	0.7146	0.4078	0.4020	0.3751	0.7949	0.6338	0.6003	0.2995
12	Ours	0.7339	0.6362	0.6625	0.3660	0.3520	0.3311	0.7582	0.5319	0.5465	0.2601

E.1 Additional LLaMA-2-13B Compression Results

Table 16 reports our local perplexity after removing sixteen layers. A corresponding cooperative-game perplexity measurement is unavailable, so this result is excluded from the paired comparisons in the main paper. Evaluation follows Appendix C.1.

Table 16: Additional LLaMA-2-13B perplexity at sixteen-layer removal. This is an unpaired local result; lower is better.

Removed	Method	WikiText-2	PTB	C4
16	Ours	29.5598	135.6001	23.3522

At sixteen-layer removal (7.94B retained parameters), our method attains an eight-task mean accuracy of 0.4509. All eight tasks were evaluated with zero demonstrations and no sample limit; this completes our four-budget LLaMA-2-13B downstream evaluation in Table 14.

F Ablation Study

F.1 Standalone Ranking and Exchange Refinement

Both settings use the same standalone KL scores and BookCorpus calibration set. Static ranking removes the lowest-scoring layers; exchange refinement subsequently optimizes calibration KL at the same budget. Table 17 reports all three budgets available in the saved static-ranking summary. The calibration objective improves (Table 6), but held-out gains vary by corpus.

Table 17: Held-out LLaMA-2-7B perplexity for static initialization and KL exchange. Lower is better; no test metric is used to accept exchanges.

Removed	Selection	WikiText-2	PTB	C4
6	Static ranking	17.0678	60.4869	15.2886
6	KL exchange	16.9716	62.7292	14.9525
9	Static ranking	23.2081	74.5081	19.3162
9	KL exchange	23.3810	76.6217	18.6853
12	Static ranking	48.0428	114.5477	33.2574
12	KL exchange	38.4150	148.6104	26.1276

F.1.1 Calibration Dataset

Table 5 reports KL-objective results using ten calibration sequences from BookCorpus, C4, or WikiText-2. The removal budget and exchange rule are fixed while the calibration source changes. The resulting held-out scores vary by source, illustrating that calibration KL and downstream perplexity are distinct criteria. The cross-entropy exchange-budget experiment below evaluates a separate objective variant.

F.1.2 EXCHANGE SEARCH BUDGET

This auxiliary experiment uses a cross-entropy (CE) calibration objective, separately from the primary KL method. It compares one-for-one exchanges with a neighborhood that also permits two-for-two exchanges, using the same initialization and BookCorpus settings. At both completed paired budgets, the expanded neighborhood returns the same layer set and held-out scores. This result is specific to the CE variant and these two configurations.

Table 18: Completed CE-objective exchange-budget comparisons on LLaMA-2-7B. Both neighborhoods give the same results at the listed budgets.

Removed	Maximum exchange size	WikiText-2	PTB	C4
3	1	13.9701	53.5106	12.3784
6	1	17.1246	61.9268	14.9007
3	2	13.9701	53.5106	12.3784
6	2	17.1246	61.9268	14.9007

F.1.3 INITIALIZATION SENSITIVITY

We vary the initial removal set using seeds 0, 1, and 2 while holding the BookCorpus calibration seed at 42. At budgets 3 and 6, all three random starts converge to the same removal masks as standalone initialization: $\{11, 12, 24\}$ and $\{9, 10, 11, 12, 21, 24\}$, respectively. Their separately recorded perplexities differ slightly from the primary-run evaluations despite identical masks; these differences cannot be attributed to layer selection. Table 19 preserves the recorded values, but cross-run score differences are descriptive rather than a controlled estimate of initialization effects. Random-start variability at budgets 9 and 12 characterizes those runs and does not measure sensitivity to independently sampled calibration data.

Table 19: Random-initialization evaluation on LLaMA-2-7B. Random rows report mean $\pm$ population standard deviation across three initial seeds; standalone rows reproduce the primary run. Zero dispersion at budgets 3 and 6 accompanies identical final masks across the random starts.

Removed	Initialization	WikiText-2	PTB	C4
3	Random	13.7078 ± 0	51.0900 ± 0	12.2890 ± 0
3	Standalone	13.6988	52.0189	12.2891
6	Random	16.9748 ± 0	60.8684 ± 0	14.9525 ± 0
6	Standalone	16.9716	62.7292	14.9525
9	Random	23.5353 ± 0.2328	74.5554 ± 1.1922	18.8695 ± 0.2605
9	Standalone	23.3810	76.6217	18.6853
12	Random	37.5653 ± 0.5237	112.7353 ± 19.6136	26.7378 ± 0.4319
12	Standalone	38.4150	148.6104	26.1276

G COMPUTATIONAL COST AND PRACTICAL OVERHEAD

We measure wall-clock pruning time on LLaMA-2-7B-hf for removal budgets of 3, 6, 9, and 12 layers. The model is cached before timing. The launcher runs our budget sequence on GPU 0 and the comparison pipeline concurrently on GPU 1 of the same host. Shared CPU and storage activity can affect the measured wall clock, so the ratio describes this concurrent execution protocol. The timing CSV records GPU indices but does not contain a per-device hardware snapshot. Timing begins when the pruning Python process starts and ends after the selected checkpoint is saved. Three repeats were attempted; the first two completed for both methods. We use these same two complete paired repeats for every row and for the aggregate; partial results from the third repeat are excluded. The third repeat failed during file serialization: our twelve-layer run encountered an I/O error while saving checkpoint weights, and the cooperative run failed while writing its sampled training data. These are incomplete end-to-end measurements, not evidence of search non-convergence. The Cooperative game implementation constructs all four budgets jointly, so its reported time is

one shared preprocessing cost and must not be added once per budget. Ours runs each budget independently. Table 20 reports the mean and sample standard deviation across the two successful runs. The uncertainty of the total time is computed from the per-repeat sums across budgets.

The total time for Ours across its four separate budgets is $1{,}705.60 \pm 1.86$ seconds (28.43 minutes), compared with 64.66 ± 0.33 minutes for the Cooperative game shared all-budget pipeline. Under this protocol, Ours is $2.27\times$ faster overall. This measurement concerns pruning-time wall clock; it is distinct from the matched generation-efficiency measurement in Table 4.

Table 20: Wall-clock pruning time on LLaMA-2-7B-hf. Ours is timed separately for each removal budget; Cooperative game constructs all four budgets jointly. Mean $\pm$ sample standard deviation over two successful repeats.

Method	Removal	Wall-clock time
Ours	3 layers	354.54 ± 14.14 s
Ours	6 layers	386.10 ± 6.39 s
Ours	9 layers	450.64 ± 5.02 s
Ours	12 layers	514.32 ± 27.41 s
Cooperative game	all 3/6/9/12 jointly	$3{,}879.40 \pm 19.72$ s

H Quantization Details

H.1 Selected Layers of Quantized Models

Our pruning method can be combined with quantization to further reduce memory usage. We apply 4-bit GPTQ (Frantar et al., 2023) using 128 randomly sampled C4 sequences of length 2048 for calibration. Table 21 summarizes the layer sets selected under different integration orders. For the prune–GPTQ–prune order, the table reports the union of the first and second pruning sets. The table records available selection manifests. Perplexity measurements are reported only for the two completed local integration orders in Table 22; a layer manifest alone is not a completed quality evaluation.

Table 21: Selected layers removed by our method under different pruning and 4-bit GPTQ integration orders on LLaMA-2-7B-hf. Layer indices are zero-based.

Integration order	Removed layers
Prune 3 layers $\rightarrow$ GPTQ	**[11,12,24]**
GPTQ $\rightarrow$ prune 3 layers	**[11,12,23]**
Prune 6 layers $\rightarrow$ GPTQ	**[9,10,11,12,21,24]**
Prune 3 layers $\rightarrow$ GPTQ $\rightarrow$ prune 3 more layers	**[9,10,11,12,21,24]**
GPTQ $\rightarrow$ prune 6 layers	**[9,10,11,12,21,24]**

Table 22: Perplexity on LLaMA-2-7B-hf under pruning–quantization integration orders. Columns report WikiText-2, PTB, and C4 perplexity; lower is better. "Cooperative game" gives their reported values; "Ours" gives local values. Only completed local configurations are shown.

Scheme	WT2	PTB	C4
Cooperative game: prune 3 layers, then apply 4-bit GPTQ	15.4001	56.3313	13.3799
Cooperative game: apply 4-bit GPTQ, then prune 3 layers	14.9263	56.3313	13.1724
Cooperative game: prune 6 layers, then apply 4-bit GPTQ	19.7741	65.8579	16.9137
Cooperative game: prune 3 layers, apply GPTQ, then prune 3 more	19.4676	67.9485	16.3933
Cooperative game: apply 4-bit GPTQ, then prune 6 layers	18.5761	65.8579	15.8890
Ours: prune 3 layers, then apply 4-bit GPTQ	**14.2778**	**55.0318**	**12.8906**
Ours: prune 6 layers, then apply 4-bit GPTQ	**17.9171**	**65.7993**	**15.6460**

I Structured and Unstructured Integration

We evaluate the feasibility of applying layer selection after 10% SparseGPT (Frantar and Alistarh, 2023) weight pruning on LLaMA-2-7B. Table 23 reports the three completed configurations. These are integration measurements, not a matched comparison with alternative pruning allocations.

Table 23: Local perplexity after 10% SparseGPT weight pruning followed by layer pruning. Lower is better.

Removed layers	WikiText-2	PTB	C4
6	17.5101	62.6060	15.2993
9	24.3804	78.8340	19.3128
12	42.4469	158.3409	27.9235