DFR: DIRECT FEATURE REFIT FOR RESTORING DEPTH-PRUNED LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Depth pruning reduces the inference cost of large language models, but removing Transformer blocks can substantially degrade model quality. Existing training-free methods often address this loss by fitting corrections on the low-dimensional FFN output. This makes fitting substantially cheaper, but can limit recovery quality when useful feature directions fall outside the subspace exposed by the pretrained down projection. We introduce **Direct Feature Refit (DFR)**, which instead fits the pruning-induced correction directly from the surviving intermediate FFN features and merges it into the existing down projection, requiring no fine-tuning or additional inference module. We further derive the globally optimal rank-r solution to characterize how the useful correction is distributed across recovery modes. At rank 512, DFR retains 86.39–86.79% of the regularized recovery gain across the three main models, while recovering 94.06–96.47% of the full DFR NLL improvement, showing that most of the language-model improvement is captured by a moderate number of recovery modes. Across two pruning strategies and pruning ratios from 25% to 50%, DFR consistently improves post-pruning model quality on these models, reducing perplexity by up to 53.6% under aggressive pruning relative to the strongest competing baseline.

1 INTRODUCTION

Depth pruning is an appealing approach to reducing the inference cost of large language models because removing complete Transformer blocks reduces sequential computation without relying on specialized sparse kernels (Song et al., 2024; Men et al., 2025; Chen et al., 2025a; Wee et al., 2025). However, deleting blocks also removes substantial nonlinear computation and can severely degrade perplexity and downstream performance (Kim et al., 2024; Chen et al., 2025a; Gromov et al., 2025). To recover model quality without costly fine-tuning, recent training-free methods use lightweight calibration-based correction or compensation after pruning (Chen et al., 2025b; Shopkhoev et al., 2025; Chen et al., 2026). A common strategy is to fit the correction on the FFN output after the down projection.

This choice is natural because the FFN intermediate dimension is much larger than the hidden dimension, so fitting on the FFN output produces a substantially smaller problem. Classical randomized linear algebra further shows that reduced-space formulations can closely approximate the corresponding full-space problem when the projection preserves the directions relevant to the target (Sarlos, 2006; Woodruff, 2014). Thus, working on the FFN output offers an attractive trade-off between fitting cost and approximation quality.

For this reduced-space fit to remain accurate, the down projection must preserve the feature directions needed by the correction. Let $Z \in \mathbb{R}^{N\times m}$ denote the surviving intermediate FFN features and $W_d \in \mathbb{R}^{m\times d}$ the pretrained down projection. For a linear correction on the FFN output, ZW_dT, the corresponding feature-side update is $\Delta_T = W_d(T-I)$, so $\mathrm{col}(\Delta_T) \subseteq \mathrm{col}(W_d)$. We refer to $\mathrm{col}(W_d)$ as the pretrained readout subspace, and to recovery restricted to this subspace as Readout. This means that output-space fitting can use only feature directions in $\mathrm{col}(W_d)$; any required feature-side correction outside this subspace cannot be represented.

Figure 1 examines this restriction on post-pruning recovery targets. Figure 1(a) compares recovery restricted to the pretrained readout subspace with recovery using the full surviving FFN feature space. Across pruning spans from 1 to 16 removed blocks, full feature access consistently achieves lower

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.

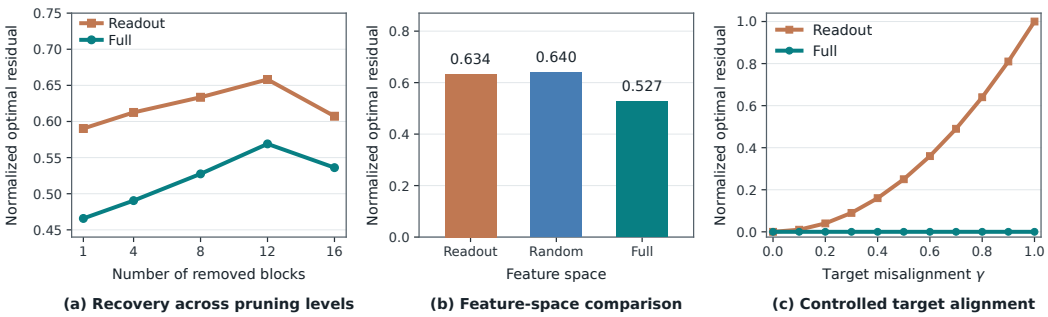


Figure 1: **Feature accessibility under post-pruning recovery. (a)** On real multi-layer pruning targets, recovery using the full surviving FFN feature space consistently achieves lower held-out residual than recovery restricted to the pretrained readout subspace across 1–16 removed blocks. **(b)** At the representative 8-block pruning setting, the pretrained readout performs slightly better than a dimension-matched random subspace, but remains substantially worse than full feature access, indicating that the learned readout is structured yet incomplete for the new recovery target. **(c)** Under controlled target misalignment, readout-constrained recovery incurs increasing residual as the target moves outside the pretrained readout subspace, whereas full-feature recovery remains unaffected.

held-out residual than the pretrained readout. This indicates that real post-pruning corrections require feature directions that are not fully exposed by the pretrained readout. Figure 1(b) compares the pretrained readout with a dimension-matched random subspace at the representative 8-block pruning setting. The pretrained readout achieves slightly lower residual than the random subspace, while full feature access further reduces the residual. Thus, the pretrained readout captures useful directions, but does not fully cover those needed by the new recovery target. Figure 1(c) makes this relation explicit by controlling the alignment between the recovery target and the readout subspace. When the target lies within the readout subspace, both Readout and Full achieve nearly zero residual. As the target increasingly moves outside this subspace, the Readout residual grows, while Full remains near zero. Together, these results show that limited access to recovery-relevant feature directions can impose a measurable ceiling on post-pruning recovery.

We therefore introduce **Direct Feature Refit (DFR)**, which fits the pruning-induced correction directly from the surviving intermediate FFN features and merges the fitted update into the existing down projection. DFR requires only calibration data and introduces no additional inference module or FLOPs. We further derive the globally optimal rank-r solution to study the structure of recovery. Across the three main models, DFR consistently improves post-pruning recovery across two pruning strategies and pruning ratios from 25% to 50%, reducing perplexity by up to 53.6% relative to the strongest competing baseline.

The contributions of this work can be summarized as follows: **(i) Feature accessibility.** We show that post-pruning recovery can be limited by the FFN feature directions preserved by the pretrained down projection, creating a measurable recovery gap when useful correction directions fall outside this subspace; **(ii) Direct Feature Refit.** We introduce DFR, which performs recovery directly in the surviving pretrained FFN feature space, and derive the exact globally optimal rank-r solution under the same recovery objective; **(iii) Recovery structure.** We characterize the rank structure of post-pruning recovery and show that most of the model-quality improvement can be captured by a moderate number of recovery modes, even though the full recovery spectrum remains long-tailed.

2 Method

We begin by formulating post-pruning recovery at a single pruning boundary and defining the feature space available to recovery. Building on this formulation, we introduce Direct Feature Refit (DFR), derive its globally optimal rank-constrained solution, and analyze the accessibility of the resulting recovery modes.

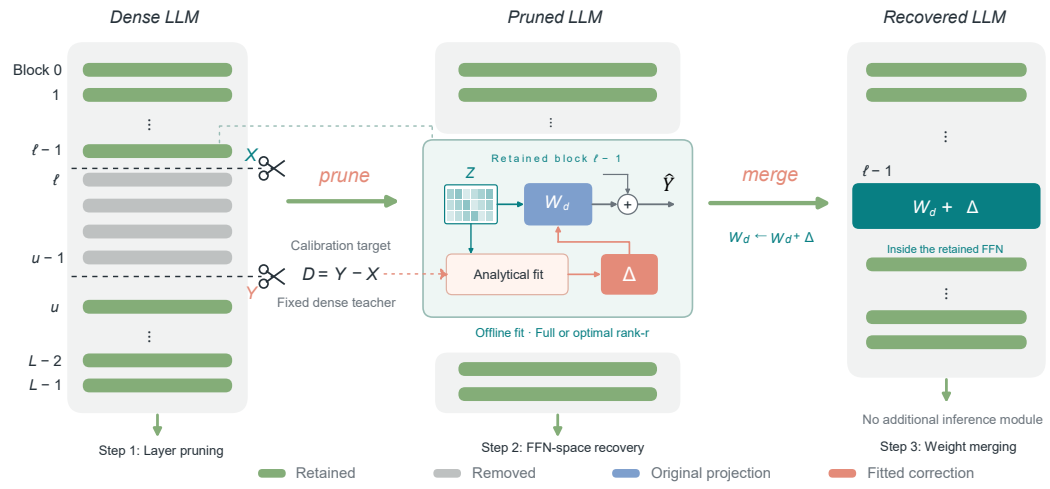


Figure 2: **Overview of Direct Feature Refit (DFR). Step 1: Layer pruning.** Removing a contiguous block interval creates a new pruning boundary and a representation gap $D = Y - X$ relative to the dense model. **Step 2: FFN-space recovery.** DFR uses the surviving intermediate FFN features Z to fit a correction Δ analytically, either in the full feature space or under a rank constraint. **Step 3: Weight merging.** The fitted correction is merged into the retained FFN down projection as $W_d \leftarrow W_d + \Delta$, introducing no additional inference module at deployment.

2.1 Recovery Setup and Feature Accessibility

Consider a decoder-only Transformer in which a contiguous interval of blocks is removed by depth pruning. We focus on the new boundary created by this removal. Let $Z \in \mathbb{R}^{N\times m}$ denote the nonlinear features produced by the surviving FFN immediately preceding the removed interval, $W_d \in \mathbb{R}^{m\times d}$ its pretrained down projection, and $R \in \mathbb{R}^{N\times d}$ the incoming residual stream, where N is the number of calibration tokens, m is the FFN intermediate width, and d is the hidden dimension. For a gated FFN, for example, $Z = \mathrm{SiLU}(HW_g) \odot (HW_u)$. After pruning, the output of this retained FFN, $X = R + ZW_d$, forms the representation at the new pruning boundary.

Let $Y \in \mathbb{R}^{N\times d}$ denote the corresponding representation produced by the dense model after the removed computation, and define the pruning-induced gap as $D = Y - X$. We express recovery in a common feature-space coordinate system: a correction is written as $Z\Delta$, with $\Delta \in \mathbb{R}^{m\times d}$, giving $X(\Delta) = X + Z\Delta$. We fit such corrections using the ridge objective

$$J(\Delta) = \|Z\Delta - D\|_F^2 + \lambda\|\Delta\|_F^2, \tag{1}$$

where $\lambda > 0$ regularizes the correction (Hoerl & Kennard, 1970).

Different recovery mechanisms correspond to different feasible sets for Δ. Let $\mathcal{S} \subseteq \mathbb{R}^m$ denote the pretrained feature subspace exposed to recovery, and define

$$\mathcal{F}(\mathcal{S}) = \left\{\Delta \in \mathbb{R}^{m\times d} : \mathrm{col}(\Delta) \subseteq \mathcal{S}\right\}. \tag{2}$$

This separates the capacity of a recovery mechanism from the feature directions available to it.

Post-output linear recovery provides a transparent restricted instance. Applying $T \in \mathbb{R}^{d\times d}$ to the FFN branch output after the pretrained down projection gives $R + (ZW_d)T = X + ZW_d(T - I)$, so its effective feature-space correction is $\Delta_T = W_d(T - I)$. Consequently, $\mathrm{col}(\Delta_T) \subseteq \mathcal{S}_{\mathrm{read}}$ with $\mathcal{S}_{\mathrm{read}} := \mathrm{col}(W_d)$. We refer to recovery restricted to this pretrained readout subspace as *readout-constrained recovery*, or simply *Readout*. Under the same objective, its optimum is $\Delta^\star_{\mathrm{read}} := \arg\min_{\Delta\in\mathcal{F}(\mathcal{S}_{\mathrm{read}})} J(\Delta)$.

Thus, Readout can freely recombine directions within $\mathcal{S}_{\mathrm{read}}$, but feature directions outside this subspace remain inaccessible.

2.2 Direct Feature Refit

An overview of DFR is shown in Fig. 2. Direct Feature Refit (DFR) removes the readout-subspace restriction by directly refitting the existing FFN down projection as

$$W_d^\star = W_d + \Delta. \tag{3}$$

This exposes the full surviving FFN feature space to recovery, with $\mathcal{S}_{\mathrm{DFR}} = \mathbb{R}^m$. The Full DFR correction is therefore defined as the unconstrained optimum $\Delta_{\mathrm{full}} := \arg\min_{\Delta \in \mathbb{R}^{m\times d}} J(\Delta)$.

Define $C = Z^\top Z + \lambda I$ and $B = Z^\top D$. Setting the gradient of Eq. (1) to zero gives

$$C\Delta_{\mathrm{full}} = B, \qquad \Delta_{\mathrm{full}} = C^{-1}B. \tag{4}$$

Since $\lambda > 0$, $C \succ 0$ and the solution is well defined. In practice, we accumulate $Z^\top Z$ and $Z^\top D$ over calibration batches and solve the linear system in Eq. (4) without explicitly forming C^{-1}. The same objective also induces the geometry relevant to structured recovery. Completing the square around Δ_{full} yields

$$J(\Delta) = J(\Delta_{\mathrm{full}}) + \left\| C^{1/2}(\Delta - \Delta_{\mathrm{full}}) \right\|_F^2. \tag{5}$$

Thus, deviations from the Full DFR solution are measured in a data-dependent geometry determined by the calibration features through $C = Z^\top Z + \lambda I$, rather than by ordinary Euclidean distance in parameter space. This geometry provides the basis for the rank-constrained recovery developed next. The derivation of Eqs. (4) and (5) is provided in Appendix A.1.

After fitting, the correction is merged into the existing projection as $W_d \leftarrow W_d + \Delta$. DFR therefore introduces no additional inference module, model parameters, or FLOPs after deployment.

2.3 Optimal Rank-Structured Recovery

Full feature access enlarges the feasible correction space, while useful recovery may be concentrated in a smaller number of modes. We therefore consider the rank-constrained problem (Izenman, 1975)

$$\Delta_r^\star = \arg \min_{\mathrm{rank}(\Delta)\le r} J(\Delta). \tag{6}$$

Using Eq. (5), this problem reduces to a standard low-rank approximation in the recovery geometry. Let $A = C^{1/2}\Delta$ and $A^\star = C^{1/2}\Delta_{\mathrm{full}} = C^{-1/2}B$. Since $C^{1/2}$ is invertible, $\mathrm{rank}(A) = \mathrm{rank}(\Delta)$, so Eq. (6) is equivalent to $\min_{\mathrm{rank}(A)\le r} \|A - A^\star\|_F^2$.

Let $A^\star = U\Sigma V^\top$ be the singular value decomposition of the whitened optimum. By the Eckart–Young theorem (Eckart & Young, 1936), the optimal rank-r approximation is $U_r\Sigma_r V_r^\top$. Transforming back gives

$$\Delta_r^\star = C^{-1/2}U_r\Sigma_r V_r^\top = \Delta_{\mathrm{full}} V_r V_r^\top. \tag{7}$$

The rank-r solution therefore retains the leading recovery modes under the objective-induced geometry. A complete derivation is provided in Appendix A.2.

These solutions induce an ordered recovery spectrum. Define $K = (A^\star)^\top A^\star = B^\top C^{-1}B$, with $Kv_i = \mu_i v_i$ and $\mu_1 \ge \mu_2 \ge \cdots \ge 0$. Here v_i gives the output-side direction of recovery mode i, while μ_i is its regularized recovery gain. Since μ_i is the squared singular value associated with mode i, the rank-r solution captures

$$J(0) - J(\Delta_r^\star) = \sum_{i=1}^{r} \mu_i, \qquad E_{\mathrm{spec}}(r) := \frac{J(0) - J(\Delta_r^\star)}{J(0) - J(\Delta_{\mathrm{full}})} = \frac{\sum_{i=1}^r \mu_i}{\sum_i \mu_i}. \tag{8}$$

Thus, $E_{\mathrm{spec}}(r)$ is the fraction of the Full regularized recovery gain captured at rank r. The corresponding gain decomposition is derived in Appendix A.3.

2.4 Accessibility of Recovery Modes

The recovery spectrum quantifies how much gain each mode carries, but not how much of that mode is expressible within the pretrained readout subspace. For each output-side direction v_i, define its

Algorithm 1 Iterative Layer Pruning with DFR

Require: Dense model M_{dense}, calibration set $\mathcal{C}$, target number of removed blocks K

$M_0 \leftarrow M_{\text{dense}}$
for $t = 0, \ldots, K-1$ **do**
  Compute the pruning score for each candidate block in M_t
  $\ell^\star \leftarrow$ block with the smallest pruning score
  Remove block $\ell^\star$ from M_t
  Fit DFR at the newly created pruning boundary using $\mathcal{C}$
  Merge the fitted correction into the surviving FFN projection
  $M_{t+1} \leftarrow$ updated recovered model
end for
return M_K

feature-side direction as $a_i = \Delta_{\text{full}} v_i$, so that the corresponding Direct recovery mode is $a_i v_i^\top$. Its recovery gain satisfies $\|a_i\|_C^2 := a_i^\top C a_i = \mu_i$.

Because recovery is measured in the C-geometry, we measure accessibility in the same geometry. Let U_o be an orthonormal basis of $\mathcal{S}_{\text{read}}$, and define the C-orthogonal projector onto this subspace as

$$P_C = U_o (U_o^\top C U_o)^{-1} U_o^\top C. \tag{9}$$

We define the recovery-aware accessibility of mode i and its corresponding inaccessible gain as

$$\eta_i^C = \frac{\|P_C a_i\|_C^2}{\|a_i\|_C^2}, \qquad g_i^{\text{inac}} = \mu_i (1 - \eta_i^C). \tag{10}$$

Here η_i^C is the fraction of mode-i recovery gain expressible within the pretrained readout subspace under the recovery metric, while g_i^{inac} is the remaining inaccessible gain.

These mode-wise quantities exactly decompose the gap between Full DFR and optimal readout-constrained recovery. Let $G_{\text{DFR}} = J(0) - J(\Delta_{\text{full}})$ and $G_{\text{Readout}} = J(0) - J(\Delta^\star_{\text{read}})$. Then

$$G_{\text{DFR}} - G_{\text{Readout}} = \sum_i \mu_i (1 - \eta_i^C). \tag{11}$$

The proof is given in Appendix A.4. Equation (11) connects feature accessibility directly to the attainable recovery ceiling: the DFR–Readout gap is exactly the recovery gain that lies outside the pretrained readout subspace.

2.5 Iterative Layer Pruning with DFR

DFR can also be integrated into an iterative pruning procedure. Rather than selecting all layers from the original dense model, we remove one block at a time and recompute the pruning criterion on the current recovered model. At pruning step t, let M_t denote the current model. We evaluate each remaining block using the Block Influence (BI) criterion of ShortGPT (Men et al., 2025), select the block with the smallest score, and remove it. DFR is then fitted at the newly created pruning boundary and merged into the corresponding surviving FFN projection. The updated recovered model becomes M_{t+1}, and the process is repeated until the target depth is reached.

This procedure keeps the original layer-selection criterion unchanged; DFR only updates the model after each pruning step before the next layer is selected. Algorithm 1 summarizes the procedure.

3 Experiments

3.1 Experimental Setup

We evaluate DFR on Llama-3-8B, Llama-3.1-8B, and DeepSeek-R1-Distill-Llama-8B, using LLM-Streamline and ShortGPT with approximately 25%, 37.5%, and 50% of Transformer blocks removed. Within each setting, all recovery methods use the same pruning layout. We compare against LinearPatch, Prune&Comp, and ReplaceMe, and report both DFR-Full and DFR-512. Recovery is fitted

Table 1: Main recovery results on three models under LLM-Streamline and ShortGPT. Each configuration reports WikiText-2 perplexity (PPL, ↓) and nine-task zero-shot average accuracy (QA, ↑). All recovery methods use the same pruning layout within each configuration; DFR-512 uses rank 512 at each recovery boundary. Bold and underline denote the best and second-best recovery methods, respectively, excluding Dense and Pruned. Some selector settings yield identical pruning masks and therefore identical recovery results.

	LLM-Streamline						ShortGPT					
	25% (8L)		37.5% (12L)		50% (16L)		25% (8L)		37.5% (12L)		50% (16L)	
Method	PPL ↓	QA ↑	PPL ↓	QA ↑	PPL ↓	QA ↑	PPL ↓	QA ↑	PPL ↓	QA ↑	PPL ↓	QA ↑
Llama-3-8B	Dense: PPL 6.14, QA 67.59											
Pruned	3944.22	40.96	37397.24	44.06	2758.91	41.70	3944.22	40.96	37397.24	44.06	1852.13	41.67
LinearPatch-D	164.29	46.45	316.49	49.99	59039.40	42.01	164.29	46.45	316.49	49.99	20955.21	41.84
LinearPatch-R	100.05	47.36	439.71	49.10	25568.81	42.14	100.05	47.36	439.71	49.10	9928.15	42.05
Prune&Comp	305.31	45.53	909.92	47.36	32600.45	42.61	305.31	45.53	909.92	47.36	13062.69	42.35
ReplaceMe-LS	16.79	55.18	48.93	49.46	111.29	42.11	16.79	55.18	48.93	49.46	79.15	41.44
ReplaceMe-Cos	19.63	54.84	90.01	49.28	4281.57	42.73	19.63	54.84	90.01	49.28	1924.21	42.22
DFR-512	17.55	55.67	41.14	49.89	98.65	42.84	17.55	55.67	41.14	49.89	79.37	42.14
DFR-Full	**13.51**	**56.23**	**22.64**	**53.81**	**64.53**	**44.28**	**13.51**	**56.23**	**22.64**	**53.81**	**46.12**	**44.57**
Llama-3.1-8B	Dense: PPL 6.24, QA 67.77											
Pruned	2812.73	42.12	80347.86	44.83	50416.95	37.70	2812.73	42.12	80347.86	44.83	2163.20	42.01
LinearPatch-D	129.39	47.79	314.96	50.32	7147.90	40.82	129.39	47.79	314.96	50.32	21021.61	42.11
LinearPatch-R	70.11	49.72	388.39	49.28	5232.53	40.89	70.11	49.72	388.39	49.28	8405.80	42.36
Prune&Comp	215.74	47.02	922.12	48.77	8810.33	40.86	215.74	47.02	922.12	48.77	11974.34	42.61
ReplaceMe-LS	16.66	55.63	39.39	49.44	162.97	38.48	16.66	55.63	39.39	49.44	86.96	40.68
ReplaceMe-Cos	19.52	55.51	111.56	49.11	490.36	39.97	19.52	55.51	111.56	49.11	1038.95	**45.27**
DFR-512	17.47	56.37	42.14	49.27	125.56	39.08	17.47	56.37	42.14	49.27	84.37	42.81
DFR-Full	**13.50**	**57.19**	**22.70**	**53.27**	**75.64**	**41.46**	**13.50**	**57.19**	**22.70**	**53.27**	**47.87**	43.58
DeepSeek-R1-Distill-Llama-8B	Dense: PPL 13.13, QA 63.86											
Pruned	3136.70	46.70	30616.01	43.26	8853.84	42.47	2088.56	52.03	30616.01	43.26	8853.84	42.47
LinearPatch-D	502.42	49.84	3940.05	47.84	113961.81	42.41	365.54	54.49	3940.05	47.84	113961.81	42.41
LinearPatch-R	321.63	50.38	3982.29	47.72	42614.53	42.69	195.61	55.99	3982.29	47.72	42614.53	42.69
Prune&Comp	821.71	49.23	10617.61	46.18	56240.67	42.74	529.48	54.84	10617.61	46.18	56240.67	42.74
ReplaceMe-LS	34.08	53.11	80.12	49.07	384.52	42.07	26.37	**57.12**	80.12	49.07	384.52	42.07
ReplaceMe-Cos	51.47	53.14	1151.88	48.14	1635.20	42.67	29.85	56.51	1151.88	48.14	1635.20	42.67
DFR-512	32.81	52.47	73.73	48.93	279.52	43.45	26.24	56.61	73.73	48.93	279.52	43.45
DFR-Full	**27.26**	**53.82**	**47.18**	**50.82**	**179.40**	**44.77**	**22.84**	55.35	**47.18**	**50.82**	**179.40**	**44.77**

on 1,024 WikiText-2 sequences of length 2,048 using the same calibration examples across methods. In the main text, we report WikiText-2 perplexity and the average accuracy over nine zero-shot tasks. Complete WikiText-2, PTB, and C4 perplexities and task-wise downstream results are reported in Appendix C; additional Qwen and Mistral experiments are in Appendix E, and implementation details are in Appendix B.

3.2 Overall Recovery Performance

Table 1 compares recovery methods under matched pruning layouts across three models, two layer-selection strategies, and pruning ratios from 25% to 50%. DFR-Full achieves the lowest perplexity for every unique pruning layout and also delivers consistently strong downstream accuracy. Its advantage becomes more pronounced under aggressive pruning: at 50% LLM-Streamline pruning, DFR-Full reduces perplexity by 42.0–53.6% relative to the strongest competing recovery baseline across the three models. These results show that directly refitting the surviving FFN feature space substantially improves the attainable recovery quality.

The rank-constrained variant retains much of this benefit with a fixed recovery-mode budget. At 50% LLM-Streamline pruning, DFR-512 still outperforms the strongest competing baseline in perplexity

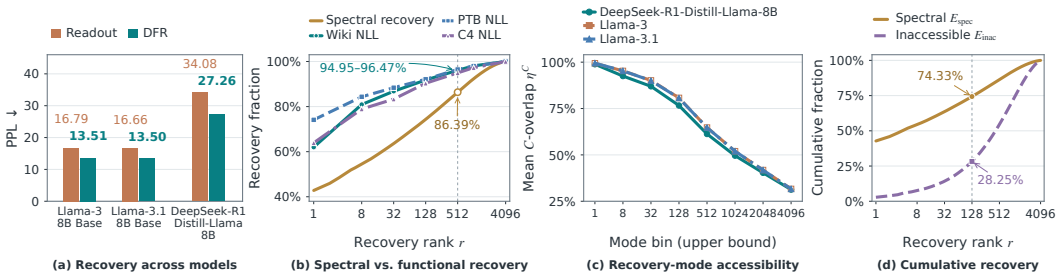


Figure 3: **Feature accessibility and recovery structure.** **(a)** DFR-Full consistently improves WikiText-2 perplexity over Readout across the three main models, showing recovery gain beyond the pretrained readout subspace. **(b)** Recovery as a function of rank on DeepSeek-R1-Distill-Llama-8B. Spectral recovery remains long-tailed, while functional NLL recovery on WikiText-2, PTB, and C4 saturates substantially earlier; at rank 512, 86.39% of the regularized recovery gain recovers 94.95–96.47% of the DFR-Full NLL improvement. **(c)** Mean C-weighted accessibility η^C to the pretrained readout subspace, grouped by recovery-mode index. Accessibility decreases along the spectrum, indicating that later modes increasingly rely on feature directions outside the readout subspace. **(d)** Cumulative total recovery gain $E_{\mathrm{spec}}(r)$ and cumulative Readout-inaccessible gain $E_{\mathrm{inac}}(r)$ on DeepSeek-R1-Distill-Llama-8B. The first 128 modes account for 74.33% of the total recovery gain but only 28.25% of the inaccessible gain.

on all three models, while using only 512 recovery modes at each boundary. Together, DFR-Full and DFR-512 show that broader feature access improves the recovery ceiling, while a moderate number of recovery modes is sufficient to capture a large fraction of the practical benefit.

3.3 Feature Accessibility and Recovery Structure

We next examine how feature accessibility and recovery rank contribute to DFR's recovery behavior. Figure 3(a) compares DFR-Full with Readout, which optimizes the same recovery objective within the pretrained readout subspace $\mathcal{S}_{\mathrm{read}}$. Across Llama-3-8B, Llama-3.1-8B, and DeepSeek-R1-Distill-Llama-8B, DFR-Full consistently achieves lower perplexity. The gap between the two therefore reflects recovery gain available beyond the pretrained readout subspace.

Table 2 provides a complementary view of this feature-space effect. An equal-dimensional random feature subspace gives recovery quality similar to Readout, while removing the component of the DFR correction outside the readout subspace substantially reduces performance. These results show that the additional recovery gain is associated with feature directions that are unavailable to a restricted subspace.

Table 2: Feature-space controls on DeepSeek-R1-Distill-Llama-8B. Random-Feature uses a random subspace with the same dimension as the pretrained readout subspace, while Drop-Perp removes the component of the DFR correction outside that subspace.

Method	Wiki ↓	PTB ↓	C4 ↓
Readout	34.08	74.98	66.28
Random-Feature	35.00	86.30	67.42
Drop-Perp	64.17	133.54	113.66
DFR-Full	**27.26**	**68.70**	**52.87**

The recovery spectrum reveals a different form of structure. As shown in Figure 3(b), spectral recovery remains long-tailed, whereas end-to-end model quality saturates much earlier. At rank 512, DFR retains 86.39–86.79% of the Full regularized recovery gain across the three models, while already recovering 94.06–96.47% of the DFR-Full NLL improvement on WikiText-2, PTB, and C4. This separation shows that most of the functional recovery can be captured with a moderate number of modes, even though the full optimum itself is not strongly low-rank.

Figure 3(c) shows how these recovery modes relate to the pretrained readout subspace. The leading mode is almost entirely aligned with this subspace, with C-weighted accessibility of 98.88–99.47% across the three models. Accessibility then decreases steadily along the spectrum, falling to 61–65% for modes 129–512 and to approximately 31–32% for modes 2049–4096. Thus, early recovery modes

Table 3: Iterative pruning with DFR on DeepSeek-R1-Distill-Llama-8B. We report perplexity on WikiText-2, PTB, and C4 together with the nine-task zero-shot average accuracy. Fixed Mask + DFR applies recovery to a pruning layout selected before recovery, while BI-Iterative + DFR recomputes Block Influence after each pruning-and-recovery step.

	37.5% Pruning				50% Pruning			
Method	Wiki ↓	PTB ↓	C4 ↓	QA ↑	Wiki ↓	PTB ↓	C4 ↓	QA ↑
Fixed Mask + DFR	47.18	280.72	125.19	50.82	179.40	2861.19	710.86	44.77
BI-Iterative + DFR	**44.48**	**233.41**	**111.40**	**50.98**	**84.79**	**800.95**	**316.18**	**45.67**

largely reuse feature directions already exposed by the pretrained readout, whereas later modes increasingly extend beyond them.

The distribution of recovery gain follows a complementary pattern. In Figure 3(d), the first 128 modes already account for 74.33% of the total DFR recovery gain, but only 28.25% of the gain inaccessible to Readout. Across the three models, 71.75–75.20% of the inaccessible gain lies beyond the first 128 modes. The total recovery gain is therefore concentrated toward the front of the spectrum, while the DFR–Readout gap is distributed more heavily among later modes.

Equation (11) also matches the measured DFR–Readout objective gap numerically. For DeepSeek-R1-Distill-Llama-8B, the relative error between the summed mode-wise inaccessible gain and the independently measured objective gap is 4.3×10^{-15}, with the same numerical agreement observed on the other two models. These results show that most functional recovery is concentrated in a moderate number of modes, while broader feature access determines the additional recovery available beyond the pretrained readout subspace.

3.4 Iterative Pruning with DFR

We further evaluate DFR in the iterative pruning procedure described in Section 2.5. After each block removal, the corresponding DFR correction is fitted and merged before the pruning criterion is recomputed on the updated model. We use the Block Influence (BI) criterion of ShortGPT (Men et al., 2025) for iterative layer selection and compare against applying DFR to the fixed pruning layout used in the main experiments. Both settings use the same calibration data and DFR recovery configuration.

Table 3 shows that recomputing the pruning criterion after each DFR recovery step consistently improves the final pruned model. At 37.5% pruning, BI-Iterative + DFR reduces WikiText-2 perplexity from 47.18 to 44.48, with corresponding improvements on PTB and C4, while the nine-task average increases slightly from 50.82 to 50.98. The benefit becomes much larger at 50% pruning: WikiText-2 perplexity drops from 179.40 to 84.79, PTB from 2861.19 to 800.95, and C4 from 710.86 to 316.18, while the zero-shot average improves from 44.77 to 45.67.

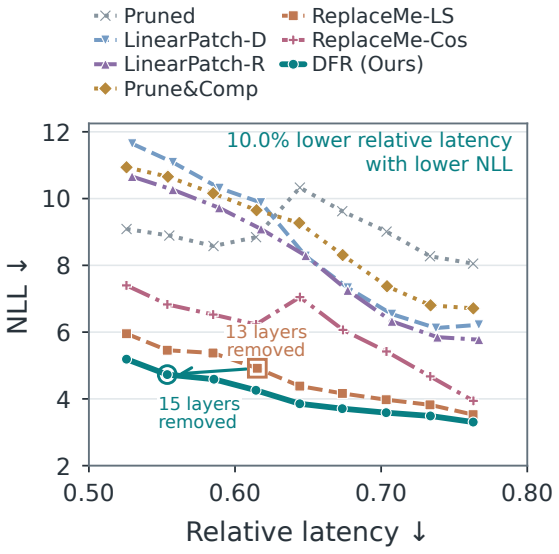


Figure 4: **Quality–latency trade-off.** WikiText-2 NLL versus relative inference latency over an 8–16 block pruning sweep on DeepSeek-R1-Distill-Llama-8B.

These results show that DFR can be naturally integrated into iterative depth pruning. By updating the model after each removal, subsequent

layer selection is performed on the recovered model rather than on the original dense model, which becomes particularly beneficial under more aggressive pruning.

3.5 Deployment Efficiency

DFR is merged into the surviving FFN projection and uses the same inference graph as the depth-pruned model. Across the 8–16 block pruning sweep, measured latency decreases from 2161.7 ms to 1489.2 ms, corresponding to speedups from $1.31\times$ to $1.90\times$ over the dense model. Measured DFR latency is effectively identical to that of the corresponding unrecovered pruned model.

Figure 4 shows the quality–latency trade-off over the full 8–16 block pruning sweep. DFR maintains lower NLL as latency decreases, with the gap becoming more pronounced under more aggressive pruning. For example, DFR with 15 blocks removed reaches a WikiText-2 perplexity of 113.06 at 1588.7 ms, compared with 136.70 at 1740.4 ms for ReplaceMe-LS with 13 blocks removed. Similarly, DFR with 16 blocks removed obtains 179.40 perplexity at 1489.2 ms, while ReplaceMe-LS with 14 blocks removed obtains 215.80 perplexity at 1644.7 ms. The complete 8–16 block sweep is reported in Appendix G.

4 Related Work

Depth Pruning and Layer Selection. Pruning has been widely explored for reducing the inference cost of large language models, ranging from weight-level sparsification (Sanh et al., 2020; Frantar & Alistarh, 2023; Sun et al., 2024; Yin et al., 2024) to structured removal of model components (Ma et al., 2023; Tao et al., 2023; An et al., 2024). Recent structured LLM pruning methods further consider joint removal of layers, heads, channels, and intermediate dimensions (Xia et al., 2024; Ashkboos et al., 2024; Zhang et al., 2024; Wei et al., 2024). Depth pruning focuses specifically on removing complete Transformer blocks, thereby reducing the number of sequential computations during inference. SLEB identifies redundant blocks through redundancy verification (Song et al., 2024), while ShortGPT introduces Block Influence (BI) to rank layers according to changes in their hidden representations (Men et al., 2025). LLM-Streamline considers consecutive layer removal based on hidden-state changes (Chen et al., 2025a), and PuDDing further makes depth selection input-dependent through prompt-routed pruning (Wee et al., 2025). Beyond whole-layer removal, BlockPruner considers finer-grained pruning of attention and MLP blocks (Zhong et al., 2025), while PIP and GISP explore iterative and globally informed structured selection procedures (Cao et al., 2025; Wang et al., 2026).

Post-Pruning Recovery. Removing Transformer blocks often requires additional recovery to restore the representations disrupted by pruning (Feng et al., 2025). LLM-Streamline learns a lightweight replacement module for the removed layers (Chen et al., 2025a). LinearPatch addresses activation mismatch at the pruning boundary through a lightweight linear transformation (Chen et al., 2025b), while ReplaceMe fits linear mappings from calibration data to approximate the removed computation and merges them into the surviving model (Shopkhoev et al., 2025). Prune&Comp instead uses magnitude compensation within an iterative pruning procedure (Chen et al., 2026). More recent work has explored representation reconstruction more directly: LaCo sequentially reconstructs hidden states through layer-wise compensation (Liu et al., 2026), while GRAIL studies post-hoc ridge reconstruction for structured compressed networks (Tang et al., 2026).

5 Conclusion

We introduced **Direct Feature Refit (DFR)** for recovering depth-pruned language models by directly refitting the surviving pretrained FFN feature space. Our analysis shows that effective recovery depends not only on the number of recovery modes, but also on which pretrained feature directions those modes can access. Across multiple 8B-scale models and pruning settings, DFR consistently improves post-pruning recovery, and can also be integrated into iterative pruning by updating the model before subsequent layer selection. Because the fitted correction is merged into the surviving FFN projection, DFR introduces no additional inference module and enables lower latency at comparable model quality, or better quality at the same inference cost.

AI USE STATEMENT

Generative AI tools were used for language editing, and LaTeX formatting assistance. All AI-assisted suggestions were reviewed by the authors, who take responsibility for the final content of the work.

REPRODUCIBILITY STATEMENT

All model, calibration, baseline, and evaluation details required to reproduce the experiments are provided in the main text and appendices.

REFERENCES

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 10865–10873, 2024.

Saleh Ashkboos, Maximilian 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*, volume 2024, pp. 11682–11701, 2024.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, et al. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024.

Yi Cao, Wei-Jie Xu, Yucheng Shen, Weijie Shi, Chi-Min Chan, Jianfeng Qu, and Jiajie Xu. Pip: Perturbation-based iterative pruning for large language models. *arXiv preprint arXiv:2501.15278*, 2025.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In *International Conference on Learning Representations*, volume 2025, pp. 30362–30383, 2025a.

Xinrui Chen, Haoli Bai, Tao Yuan, Kang Zhao, Xianzhi Yu, Lu Hou, Tian Guan, Yonghong He, Chun Yuan, et al. A simple linear patch revives layer-pruned large language models. *Advances in Neural Information Processing Systems*, 38:154484–154509, 2025b.

Xinrui Chen, Hongxing Zhang, Fanyi Zeng, Yongxian Wei, Yizhi Wang, Xitong Ling, Guanghao Li, and Chun Yuan. Prune&comp: Free lunch for layer-pruned llms via iterative pruning with magnitude compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20316–20324, 2026.

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 *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, 2019.

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.

Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1(3):211–218, 1936.

Zijian Feng, Hanzhang Zhou, Zixiao Zhu, Tianjiao Li, Chua Deryl, Lee Onn Mak, Gee Ng, and Kezhi Mao. Restoring pruned large language models via lost component compensation. *Advances in Neural Information Processing Systems*, 38:6462–6492, 2025.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International conference on machine learning*, pp. 10323–10337. PMLR, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

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*, volume 2025, pp. 81906–81920, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of multivariate analysis*, 5(2):248–264, 1975.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023.

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.

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.

Yingen Liu, Fan Wu, Ruihui Li, Zhuo Tang, Kenli Li, et al. Laco: Layer-wise compensation for pruned large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 29099–29113, 2026.

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

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

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 *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

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

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*, pp. 2381–2391, 2018.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

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

Victor Sanh, Thomas Wolf, and Alexander Rush. Movement pruning: Adaptive sparsity by fine-tuning. *Advances in neural information processing systems*, 33:20378–20389, 2020.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *2006 47th annual IEEE symposium on foundations of computer science (FOCS'06)*, pp. 143–152. IEEE, 2006.

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. *Advances in Neural Information Processing Systems*, 38: 19487–19517, 2025.

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. *arXiv preprint arXiv:2402.09025*, 2024.

Mingjie Sun, Zhuang Liu, Anna Bair, and Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 4942–4964, 2024.

Wenwu Tang, Dong Wang, Lothar Thiele, and Olga Saukh. Grail: Post-hoc compensation by linear reconstruction for compressed networks. *arXiv preprint arXiv:2602.23795*, 2026.

Chaofan Tao, Lu Hou, Haoli Bai, Jiansheng Wei, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. Structured pruning for efficient generative pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10880–10895, 2023.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pp. 353–355, 2018.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32, 2019.

Ziyan Wang, Enmao Diao, Qi Le, Pu Wang, Minwoo Lee, Shu-ping Yeh, Evgeny Stupachenko, Hao Feng, and Li Yang. From local to global: Revisiting structured pruning paradigms for large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 35720–35739, 2026.

Juyun Wee, Minjae Park, and Jaeho Lee. Prompt-based depth pruning of large language models. *arXiv preprint arXiv:2502.04348*, 2025.

Jiateng Wei, Quan Lu, Ning Jiang, Siqi Li, Jingyang Xiang, Jun Chen, and Yong Liu. Structured optimal brain pruning for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13991–14007, 2024.

David P Woodruff. Sketching as a tool for numerical linear algebra. *Foundations and Trends® in Theoretical Computer Science*, 10(1-2):1–157, 2014.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. Sheared llama: Accelerating language model pre-training via structured pruning. In *International Conference on Learning Representations*, volume 2024, pp. 5385–5409, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, et al. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. *arXiv preprint arXiv:2310.05175*, 2024.

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.

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

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. Blockpruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5065–5080, 2025.

A Mathematical Derivations

A.1 Full DFR Solution and Recovery Geometry

We begin from the ridge-regularized recovery objective

$$J(\Delta) = \|Z\Delta - D\|_F^2 + \lambda\|\Delta\|_F^2. \tag{12}$$

Expanding the first term gives

$$J(\Delta) = \operatorname{tr}\left[(Z\Delta - D)^\top (Z\Delta - D)\right] + \lambda \operatorname{tr}(\Delta^\top \Delta) \tag{13}$$

$$= \operatorname{tr}(\Delta^\top Z^\top Z\Delta) - 2\operatorname{tr}(\Delta^\top Z^\top D) + \operatorname{tr}(D^\top D) + \lambda \operatorname{tr}(\Delta^\top \Delta). \tag{14}$$

With

$$C = Z^\top Z + \lambda I, \qquad B = Z^\top D, \tag{15}$$

this becomes

$$J(\Delta) = \operatorname{tr}(\Delta^\top C\Delta) - 2\operatorname{tr}(\Delta^\top B) + \operatorname{tr}(D^\top D). \tag{16}$$

Taking the derivative with respect to Δ yields

$$\nabla_\Delta J = 2C\Delta - 2B. \tag{17}$$

Setting the gradient to zero gives

$$C\Delta_{\text{full}} = B. \tag{18}$$

Since $\lambda > 0$, $C = Z^\top Z + \lambda I \succ 0$ and is therefore invertible, so

$$\Delta_{\text{full}} = C^{-1}B. \tag{19}$$

We next derive the complete-square form used in the main text. Let

$$E = \Delta - \Delta_{\text{full}}, \qquad \Delta = \Delta_{\text{full}} + E. \tag{20}$$

Substituting into Eq. (16),

$$J(\Delta) = \operatorname{tr}\left[(\Delta_{\text{full}} + E)^\top C(\Delta_{\text{full}} + E)\right] - 2\operatorname{tr}\left[(\Delta_{\text{full}} + E)^\top B\right] + \operatorname{tr}(D^\top D). \tag{21}$$

Expanding the quadratic term gives

$$J(\Delta) = \operatorname{tr}(\Delta_{\text{full}}^\top C\Delta_{\text{full}}) + 2\operatorname{tr}(E^\top C\Delta_{\text{full}}) + \operatorname{tr}(E^\top CE) - 2\operatorname{tr}(\Delta_{\text{full}}^\top B) - 2\operatorname{tr}(E^\top B) + \operatorname{tr}(D^\top D). \tag{22}$$

Using $C\Delta_{\text{full}} = B$, the cross terms cancel:

$$2\operatorname{tr}(E^\top C\Delta_{\text{full}}) - 2\operatorname{tr}(E^\top B) = 0. \tag{23}$$

The remaining terms yield

$$J(\Delta) = J(\Delta_{\text{full}}) + \operatorname{tr}(E^\top CE). \tag{24}$$

Since $C \succ 0$,

$$\operatorname{tr}(E^\top CE) = \|C^{1/2}E\|_F^2, \tag{25}$$

and therefore

$$J(\Delta) = J(\Delta_{\text{full}}) + \|C^{1/2}(\Delta - \Delta_{\text{full}})\|_F^2. \tag{26}$$

Equation (26) shows that the natural geometry of the recovery problem is determined by C, rather than by the ordinary Frobenius distance between correction matrices.

A.2 Globally Optimal Rank-r Recovery

We consider

$$\min_{\mathrm{rank}(\Delta)\le r} J(\Delta). \tag{27}$$

Using Eq. (26), the constant $J(\Delta_{\mathrm{full}})$ can be dropped, giving

$$\min_{\mathrm{rank}(\Delta)\le r} \left\| C^{1/2}(\Delta - \Delta_{\mathrm{full}}) \right\|_F^2. \tag{28}$$

Define

$$A = C^{1/2}\Delta, \qquad A^\star = C^{1/2}\Delta_{\mathrm{full}}. \tag{29}$$

Because $C^{1/2}$ is invertible,

$$\mathrm{rank}(A) = \mathrm{rank}(C^{1/2}\Delta) = \mathrm{rank}(\Delta). \tag{30}$$

Therefore Eq. (28) is equivalent to

$$\min_{\mathrm{rank}(A)\le r} \|A - A^\star\|_F^2. \tag{31}$$

Let

$$A^\star = U\Sigma V^\top = \sum_i \sigma_i u_i v_i^\top, \tag{32}$$

with $\sigma_1 \ge \sigma_2 \ge \cdots \ge 0$. By the Eckart–Young theorem, an optimal rank-r solution is

$$A_r^\star = U_r \Sigma_r V_r^\top = \sum_{i=1}^{r} \sigma_i u_i v_i^\top. \tag{33}$$

Transforming back gives

$$\Delta_r^\star = C^{-1/2} U_r \Sigma_r V_r^\top. \tag{34}$$

We now derive the equivalent expression used in the main text. Since

$$\Delta_{\mathrm{full}} = C^{-1/2} U \Sigma V^\top, \tag{35}$$

we have

$$\begin{aligned} \Delta_{\mathrm{full}} V_r V_r^\top &= C^{-1/2} U \Sigma V^\top V_r V_r^\top && (36)\\ &= C^{-1/2} U_r \Sigma_r V_r^\top && (37)\\ &= \Delta_r^\star. && (38) \end{aligned}$$

Hence

$$\Delta_r^\star = \Delta_{\mathrm{full}} V_r V_r^\top. \tag{39}$$

The rank-r solution is therefore obtained by retaining the leading r recovery modes of the whitened optimum, rather than by truncating Δ_{full} directly in ordinary Euclidean weight space.

A.3 Recovery Spectrum and Cumulative Gain

Since

$$A^\star = C^{-1/2} B, \tag{40}$$

its right-singular Gram matrix is

$$\begin{aligned} K &= (A^\star)^\top A^\star && (41)\\ &= B^\top C^{-1/2} C^{-1/2} B && (42)\\ &= B^\top C^{-1} B. && (43) \end{aligned}$$

Using $A^\star = U\Sigma V^\top$,

$$K = V \Sigma^2 V^\top. \tag{44}$$

Thus

$$Kv_i = \sigma_i^2 v_i. \tag{45}$$

Defining

$$\mu_i = \sigma_i^2, \tag{46}$$

the eigenvalues of K are exactly the squared singular values of the whitened recovery optimum.

To derive the rank-r recovery gain, first note from Eq. (26) that

$$J(0) = J(\Delta_{\text{full}}) + \|A^\star\|_F^2. \tag{47}$$

Since

$$\|A^\star\|_F^2 = \sum_i \sigma_i^2 = \sum_i \mu_i, \tag{48}$$

Full DFR achieves total regularized recovery gain

$$J(0) - J(\Delta_{\text{full}}) = \sum_i \mu_i. \tag{49}$$

For the rank-r solution,

$$J(\Delta_r^\star) - J(\Delta_{\text{full}}) = \|A_r^\star - A^\star\|_F^2 \tag{50}$$

$$= \sum_{i>r} \sigma_i^2 \tag{51}$$

$$= \sum_{i>r} \mu_i. \tag{52}$$

Therefore

$$J(0) - J(\Delta_r^\star) = \sum_i \mu_i - \sum_{i>r} \mu_i \tag{53}$$

$$= \sum_{i=1}^{r} \mu_i. \tag{54}$$

Consequently,

$$E_{\text{spec}}(r) = \frac{J(0) - J(\Delta_r^\star)}{J(0) - J(\Delta_{\text{full}})} = \frac{\sum_{i=1}^r \mu_i}{\sum_i \mu_i}. \tag{55}$$

A.4 Mode Accessibility and Exact Gap Decomposition

Let

$$\mathcal{S}_{\text{read}} = \text{col}(W_d), \tag{56}$$

and let U_o be an ordinary Euclidean orthonormal basis of this subspace. The C-orthogonal projection of a feature vector x onto $\mathcal{S}_{\text{read}}$ is obtained by solving

$$\min_y \|x - U_o y\|_C^2. \tag{57}$$

Differentiating with respect to y gives

$$U_o^\top C U_o \, y = U_o^\top C x, \tag{58}$$

and hence

$$y = (U_o^\top C U_o)^{-1} U_o^\top C x. \tag{59}$$

Therefore

$$P_C = U_o (U_o^\top C U_o)^{-1} U_o^\top C. \tag{60}$$

Now consider recovery mode i. Its feature-side direction is

$$a_i = \Delta_{\text{full}} v_i. \tag{61}$$

Using $A^\star = C^{1/2}\Delta_{\text{full}}$,

$$\|a_i\|_C^2 = a_i^\top C a_i \tag{62}$$
$$= \|C^{1/2}\Delta_{\text{full}} v_i\|_2^2 \tag{63}$$
$$= \|A^\star v_i\|_2^2. \tag{64}$$

Since v_i is a right singular vector of $A^\star$,

$$A^\star v_i = \sigma_i u_i, \tag{65}$$

and therefore

$$\|a_i\|_C^2 = \sigma_i^2 = \mu_i. \tag{66}$$

The fraction of mode-i recovery gain expressible within the readout subspace is thus

$$\eta_i^C = \frac{\|P_C a_i\|_C^2}{\|a_i\|_C^2}. \tag{67}$$

Because P_C is a C-orthogonal projector,

$$\|a_i\|_C^2 = \|P_C a_i\|_C^2 + \|(I - P_C)a_i\|_C^2. \tag{68}$$

Combining this identity with Eq. (66),

$$\|(I - P_C)a_i\|_C^2 = \mu_i - \mu_i \eta_i^C \tag{69}$$
$$= \mu_i(1 - \eta_i^C). \tag{70}$$

Hence the inaccessible gain of mode i is

$$g_i^{\text{inac}} = \mu_i(1 - \eta_i^C). \tag{71}$$

It remains to connect these mode-wise quantities to the optimal readout-constrained solution. From Eq. (26),

$$\Delta_{\text{read}}^\star = \arg\min_{\Delta \in \mathcal{F}(\mathcal{S}_{\text{read}})} \|\Delta - \Delta_{\text{full}}\|_C^2. \tag{72}$$

Since the constraint acts independently on each column, the optimum is the C-orthogonal projection

$$\Delta_{\text{read}}^\star = P_C \Delta_{\text{full}}. \tag{73}$$

Therefore

$$J(\Delta_{\text{read}}^\star) - J(\Delta_{\text{full}}) = \|(I - P_C)\Delta_{\text{full}}\|_C^2. \tag{74}$$

Using the orthonormal basis $\{v_i\}$,

$$\Delta_{\text{full}} = \sum_i a_i v_i^\top. \tag{75}$$

Hence

$$\|(I - P_C)\Delta_{\text{full}}\|_C^2 = \left\| \sum_i (I - P_C) a_i v_i^\top \right\|_C^2 \tag{76}$$
$$= \sum_i \|(I - P_C)a_i\|_C^2 \tag{77}$$
$$= \sum_i \mu_i(1 - \eta_i^C), \tag{78}$$

where the cross terms vanish because the v_i are orthonormal.

Finally, defining

$$G_{\text{DFR}} = J(0) - J(\Delta_{\text{full}}) \tag{79}$$

and

$$G_{\text{Readout}} = J(0) - J(\Delta_{\text{read}}^\star), \tag{80}$$

we have

$$G_{\text{DFR}} - G_{\text{Readout}} = J(\Delta_{\text{read}}^\star) - J(\Delta_{\text{full}}) \tag{81}$$
$$= \sum_i \mu_i(1 - \eta_i^C). \tag{82}$$

Thus, the Full DFR–Readout recovery gap is exactly the sum of recovery gain carried by mode components inaccessible to the pretrained readout subspace.

Table 4: Pruning layouts used in the main experiments. Block indices are zero-based. "Both" denotes an identical mask produced by LLM-Streamline and ShortGPT for the same model and pruning ratio.

Model	Selector	Pruning Ratio	Removed Blocks
Llama-3-8B	Both	25%	$\{22, \ldots, 29\}$
	Both	37.5%	$\{18, \ldots, 29\}$
	LLM-Streamline	50%	$\{15, \ldots, 30\}$
	ShortGPT	50%	$\{11\} \cup \{16, \ldots, 30\}$
Llama-3.1-8B	Both	25%	$\{22, \ldots, 29\}$
	Both	37.5%	$\{18, \ldots, 29\}$
	LLM-Streamline	50%	$\{14, \ldots, 29\}$
	ShortGPT	50%	$\{11\} \cup \{16, \ldots, 30\}$
DeepSeek-R1-Distill-Llama-8B	LLM-Streamline	25%	$\{22, \ldots, 29\}$
	ShortGPT	25%	$\{20\} \cup \{22, \ldots, 28\}$
	Both	37.5%	$\{18, \ldots, 29\}$
	Both	50%	$\{15, \ldots, 30\}$

B Experimental Details

B.1 Models and Pruning Configurations

Our main experiments use three 8B-scale decoder-only language models: Llama-3-8B and Llama-3.1-8B (Grattafiori et al., 2024), and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025). For each model, we consider pruning layouts produced by LLM-Streamline (Chen et al., 2025a) and ShortGPT (Men et al., 2025).

For 32-layer models, we remove 8, 12, and 16 Transformer blocks, corresponding to pruning ratios of 25%, 37.5%, and 50%, respectively. Within each model, selector, and pruning ratio, all recovery methods are evaluated on exactly the same pruning layout.

Some LLM-Streamline and ShortGPT configurations produce identical pruning layouts. Such cases correspond to the same pruned model and therefore yield identical recovery results; we count them only once when reporting statistics over unique pruning layouts. The exact pruning layouts used in the main experiments are listed in Table 4.

Additional experiments on Qwen and Mistral models are reported in Appendix E.

B.2 Recovery Methods

We compare DFR with LinearPatch (Chen et al., 2025b), Prune&Comp (Chen et al., 2026), and ReplaceMe (Shopkhoev et al., 2025). For LinearPatch, we report its D and R variants, while for ReplaceMe we report the LS and cosine variants used in our experiments. All recovery methods within the same configuration operate on the same pruned model and calibration examples.

For DFR, we report two primary variants. **DFR-Full** uses the unconstrained solution Δ_{full} from Eq. (4). **DFR-512** uses the globally optimal rank-512 solution from Eq. (7). When a pruning layout contains multiple recovery boundaries, rank 512 is applied independently at each boundary rather than being divided across boundaries.

For the mechanism analyses, we additionally use **Readout**, the optimum of the same recovery objective restricted to the pretrained readout subspace. We also use auxiliary feature-space controls including **Random-Feature**, which replaces the pretrained readout subspace with an equal-dimensional random subspace, and **Drop-Perp**, which removes the component of the DFR correction outside the pretrained readout subspace. These controls are used only for analyzing recovery geometry and feature accessibility.

B.3 Calibration and Numerical Details

Recovery is fitted using 1,024 WikiText-2 sequences (Merity et al., 2016) with context length 2,048, corresponding to approximately 2.1 million calibration token positions. The calibration sequences are sampled from the training split, while WikiText-2 perplexity is evaluated on the test split. All recovery methods within the same experimental configuration use the same calibration examples.

For DFR and the readout-constrained controls, we accumulate the sufficient statistics $Z^\top Z$ and $Z^\top D$ over calibration batches rather than materializing the full calibration feature matrix. The accumulated statistics and linear solves used in the main analysis are computed in FP32, while recovered weights are stored and evaluated in the model's inference precision.

We use a ridge coefficient of $\lambda = 10^{-4}\,\mathrm{tr}(Z^\top Z)/m$ throughout the main experiments. The same regularization setting is used for DFR and the corresponding readout-constrained controls. For configurations containing multiple recovery boundaries, each boundary is fitted independently against the original dense teacher in increasing original-layer order. Previously fitted corrections are not deployed during fitting; after all boundaries have been fitted, the corrections are deployed jointly, with segment modifications applied in decreasing original-layer order.

DFR and its analytical controls require no gradient-based fine-tuning. The corrections are obtained directly from calibration activations and the closed-form systems described in Section 2. After fitting, the correction is merged into the retained FFN down projection as

$$W_d \leftarrow W_d + \Delta.$$

The recovered model therefore uses the same architecture and inference graph as the corresponding pruned model.

For the motivation analysis in Fig. 1, we evaluate the recoverability of pruning-induced boundary targets under different accessible feature spaces. For Fig. 1(a), the correction is fitted on the calibration split and the normalized reconstruction residual is evaluated on held-out sequences without refitting. We compare recovery restricted to the pretrained readout subspace with recovery using the full surviving FFN feature space over pruning spans of 1, 4, 8, 12, and 16 removed blocks. The normalized residual is defined as

$$R_{\mathcal{S}} = \frac{\left\| D_{\mathrm{val}} - Z_{\mathrm{val}}\widehat{\Delta}_{\mathcal{S}} \right\|_F^2}{\|D_{\mathrm{val}}\|_F^2}, \qquad \widehat{\Delta}_{\mathcal{S}} \in \operatorname*{arg\,min}_{\Delta:\,\mathrm{col}(\Delta)\subseteq\mathcal{S}} \|D_{\mathrm{cal}} - Z_{\mathrm{cal}}\Delta\|_F^2 .$$

Here $D = Y - X$ is the dense-model boundary difference across the removed interval and Z contains the surviving FFN features before the down projection. We use $\mathcal{S} = \mathrm{col}(W_d)$ for Readout and $\mathcal{S} = \mathbb{R}^m$ for Full. The denominator is the target energy on the evaluated split at the corresponding pruning span and is shared by all feature-space conditions. These unregularized fits use 2,097,152 calibration positions and are evaluated on 4,096 held-out positions.

For Fig. 1(b), we additionally replace the pretrained readout subspace with a dimension-matched random subspace at the representative 8-block pruning setting. Each random subspace is obtained by QR orthonormalization of a Gaussian matrix and has dimension 4,096, matching the pretrained readout subspace within the 14,336-dimensional FFN feature space. Results are averaged over three random-subspace seeds.

For Fig. 1(c), we construct targets with controlled misalignment γ to the pretrained readout subspace using centered real FFN features. Let Z contain 8,192 centered feature vectors from block 21, let U be an orthonormal basis of $\mathrm{col}(W_d)$, and define $H = ZU$ and $P_H = HH^\dagger$. We independently sample standard Gaussian matrices $A \in \mathbb{R}^{4096\times 32}$ and $G \in \mathbb{R}^{14336\times 32}$, and construct

$$V = (I - UU^\top)G, \qquad D_\parallel = \frac{HA}{\|HA\|_F}, \qquad D_\perp = \frac{(I - P_H)ZV}{\|(I - P_H)ZV\|_F}.$$

The mixed target is

$$D(\gamma) = \sqrt{1-\gamma^2}\, D_\parallel + \gamma D_\perp.$$

Here $D_\parallel$ lies in $\mathrm{col}(H)$, while $D_\perp$ is orthogonal to $\mathrm{col}(H)$ and remains representable using Z. Both components have unit Frobenius norm, so $\|D(\gamma)\|_F^2 = 1$. We vary γ from 0 to 1 and report the optimal normalized residual under Readout and Full feature access. Curves are averaged over three target seeds.

B.4 Evaluation Protocol

We evaluate language-model quality using perplexity on WikiText-2 (Merity et al., 2016), Penn Treebank (PTB) (Marcus et al., 1993), and C4 (Raffel et al., 2020). All recovery methods within a model are evaluated using the same tokenized examples and context-length protocol.

We additionally evaluate zero-shot downstream performance on nine tasks: ARC-Easy and ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), BoolQ (Clark et al., 2019), OpenBookQA (Mihaylov et al., 2018), RTE (Wang et al., 2018), COPA (Wang et al., 2019), and RACE (Lai et al., 2017). Zero-shot evaluation is implemented using the Language Model Evaluation Harness (Biderman et al., 2024).

ARC-E, ARC-C, HellaSwag, and OpenBookQA use normalized accuracy (`acc_norm`); the remaining tasks use standard accuracy. The nine-task score reported in the main text is the unweighted arithmetic mean of the corresponding task metrics.

The main tables report WikiText-2 perplexity together with the nine-task average as compact summaries of language-model and downstream performance. Complete perplexity results on WikiText-2, PTB, and C4 and the individual zero-shot task scores are reported in Appendix C. For mechanism analyses, boundary reconstruction error is additionally reported when needed as a diagnostic quantity.

B.5 Implementation Details

All experiments are implemented in PyTorch. Model weights are loaded and evaluated in BF16, while the DFR sufficient statistics and linear solves are computed in FP32. Recovery fitting is performed on a single NVIDIA H200 GPU. We use Transformers version 4.52.0 for model loading and lm-evaluation-harness version 0.4.4 for downstream evaluation.

Latency measurements use a separate controlled inference protocol, described in Appendix G.

C Complete Recovery Results

C.1 Complete Perplexity Results

Tables 5, 6, and 7 report the complete perplexity results on WikiText-2, PTB, and C4 for all main-model pruning configurations. The pruning layouts are identical to those used in the main text and are listed in Table 4. As in Table 4, "Both" denotes a pruning layout shared by LLM-Streamline and ShortGPT. Bold denotes the lowest perplexity for each dataset within a pruning configuration.

C.2 Complete Zero-Shot Results

Tables 8, 9, and 10 report the individual zero-shot task scores corresponding to the nine-task averages shown in the main text. As in Table 4, "Both" denotes a pruning layout shared by LLM-Streamline and ShortGPT. Scores are reported as percentages; bold denotes the highest nine-task average within each pruning configuration, and shading highlights DFR-Full.

D Additional Recovery Analyses

This section provides additional results supporting the recovery-structure analysis in Section 3.3. All analyses use 25% LLM-Streamline pruning and the same calibration protocol as the main experiments.

D.1 Rank Spectrum and Functional Recovery

Figure 5 reports the full rank-recovery curves for the three main models. For each rank r, we report the fraction of regularized recovery gain captured by the optimal rank-r solution, $E_{\mathrm{spec}}(r)$, together with the fraction of the DFR-Full NLL improvement recovered on WikiText-2, PTB, and C4. The curves consistently show that functional recovery saturates earlier than the full recovery spectrum.

Table 11 summarizes the rank required to capture 90%, 95%, and 99% of the Full DFR regularized recovery gain. Across models, the spectral thresholds are closely matched, while rank 512 already

Table 5: Complete perplexity results for Llama-3-8B

Selector	Pruning	Method	Wiki ↓	PTB ↓	C4 ↓
Both	25%	Pruned	3944.22	3952.08	2887.36
		LinearPatch-D	164.29	238.36	242.26
		LinearPatch-R	100.05	112.16	148.48
		Prune&Comp	305.31	388.27	351.87
		ReplaceMe-LS	16.79	41.78	38.57
		ReplaceMe-Cos	19.63	**37.43**	38.89
		DFR-512	17.55	52.71	40.44
		DFR-Full	**13.51**	40.56	**31.99**
Both	37.5%	Pruned	37397.24	41956.87	4870.98
		LinearPatch-D	316.49	673.59	312.44
		LinearPatch-R	439.71	756.34	400.18
		Prune&Comp	909.92	1842.23	727.81
		ReplaceMe-LS	48.93	327.51	137.90
		ReplaceMe-Cos	90.01	591.71	178.47
		DFR-512	41.14	266.95	143.62
		DFR-Full	**22.64**	**136.49**	**77.58**
LLM-Streamline	50%	Pruned	2758.91	4316.07	2101.66
		LinearPatch-D	59039.40	69530.40	48366.18
		LinearPatch-R	25568.81	35263.39	23687.82
		Prune&Comp	32600.45	48058.01	27392.96
		ReplaceMe-LS	111.29	1371.65	539.23
		ReplaceMe-Cos	4281.57	1139.12	2330.83
		DFR-512	98.65	1195.92	441.32
		DFR-Full	**64.53**	**879.52**	**300.56**
ShortGPT	50%	Pruned	1852.13	2547.13	1760.36
		LinearPatch-D	20955.21	28039.51	21802.10
		LinearPatch-R	9928.15	14183.41	10057.41
		Prune&Comp	13062.69	19019.26	12242.01
		ReplaceMe-LS	79.15	1027.00	395.97
		ReplaceMe-Cos	1924.21	850.16	998.13
		DFR-512	79.37	880.86	342.80
		DFR-Full	**46.12**	**595.14**	**215.81**

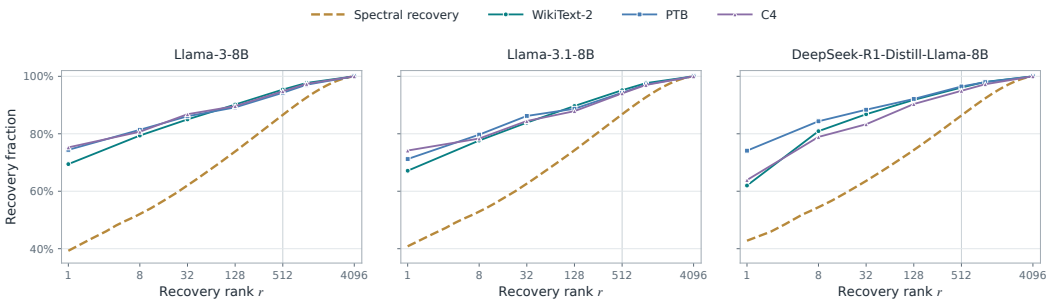


Figure 5: Full rank-recovery curves for Llama-3-8B, Llama-3.1-8B, and DeepSeek-R1-Distill-Llama-8B. Each panel reports spectral recovery $E_{\mathrm{spec}}(r)$ and functional NLL recovery on WikiText-2, PTB, and C4.

recovers substantially more of the DFR-Full NLL improvement than of the regularized recovery spectrum.

Table 6: Complete perplexity results for Llama-3.1-8B.

Selector	Pruning	Method	Wiki ↓	PTB ↓	C4 ↓
Both	25%	Pruned	2812.73	2472.16	1831.00
		LinearPatch-D	129.39	159.57	174.20
		LinearPatch-R	70.11	74.53	95.01
		Prune&Comp	215.74	223.07	225.12
		ReplaceMe-LS	16.66	42.73	40.29
		ReplaceMe-Cos	19.52	**36.02**	37.44
		DFR-512	17.47	52.34	41.89
		DFR-Full	**13.50**	41.44	**32.99**
Both	37.5%	Pruned	80347.86	73986.67	7654.58
		LinearPatch-D	314.96	634.99	293.02
		LinearPatch-R	388.39	602.23	329.05
		Prune&Comp	922.12	1542.48	665.36
		ReplaceMe-LS	39.39	341.61	148.68
		ReplaceMe-Cos	111.56	910.37	207.56
		DFR-512	42.14	287.20	154.65
		DFR-Full	**22.70**	**144.09**	**80.78**
LLM-Streamline	50%	Pruned	50416.95	36226.22	19305.19
		LinearPatch-D	7147.90	7344.80	3923.84
		LinearPatch-R	5232.53	6236.89	3750.18
		Prune&Comp	8810.33	10569.05	5530.42
		ReplaceMe-LS	162.97	2575.01	1062.93
		ReplaceMe-Cos	490.36	12015.28	1017.36
		DFR-512	125.56	1983.44	806.80
		DFR-Full	**75.64**	**1077.22**	**442.84**
ShortGPT	50%	Pruned	2163.20	1783.54	2004.04
		LinearPatch-D	21021.61	32529.92	22549.90
		LinearPatch-R	8405.80	12156.77	8700.12
		Prune&Comp	11974.34	17245.29	11164.45
		ReplaceMe-LS	86.96	1281.07	507.45
		ReplaceMe-Cos	1038.95	**684.99**	798.26
		DFR-512	84.37	1107.72	423.86
		DFR-Full	**47.87**	693.87	**246.23**

D.2 Feature-Accessibility Controls

We provide additional controls for the feature-accessibility analysis. **Readout** optimizes the same ridge objective as DFR while restricting the correction to the pretrained readout subspace. **Random-Feature** replaces this subspace with an equal-dimensional random feature subspace, obtained by sampling an orthonormal basis and solving the same constrained ridge problem within that basis. **Drop-Perp** removes the component of the DFR correction outside the pretrained readout subspace using the Euclidean orthogonal projection $\Delta_{\mathrm{Drop}} = UU^\top \Delta_{\mathrm{DFR}}^{\mathrm{deployed}}$, where U is an orthonormal basis for $\mathrm{col}(W_d)$ and $\Delta_{\mathrm{DFR}}^{\mathrm{deployed}}$ denotes the effective update after merging into the inference-precision weights. The projected correction is merged into the original down projection without refitting.

Table 12 reports the resulting perplexities. DFR-Full improves over Readout across all three models, while the additional controls on DeepSeek-R1-Distill-Llama-8B characterize the effect of restricting the accessible feature space.

Table 13 further separates recovery rank from feature accessibility. At rank 512, DFR outperforms Readout on all three datasets under the same mode budget. DFR-1024 also outperforms Readout-Full on all three datasets, despite using a moderate number of recovery modes.

D.3 Mode-Wise Accessibility and Inaccessible Gain

The main text summarizes how accessibility changes along the recovery spectrum. Table 14 reports the corresponding cross-model statistics. We denote the recovery gain inaccessible to Readout

Table 7: Complete perplexity results for DeepSeek-R1-Distill-Llama-8B.

Selector	Pruning	Method	Wiki ↓	PTB ↓	C4 ↓
LLM-Streamline	25%	Pruned	3136.70	3405.93	1803.02
		LinearPatch-D	502.42	711.96	391.54
		LinearPatch-R	321.63	499.05	222.54
		Prune&Comp	821.71	1123.09	373.98
		ReplaceMe-LS	34.08	74.98	66.27
		ReplaceMe-Cos	51.47	109.44	235.43
		DFR-512	32.81	78.85	63.18
		DFR-Full	**27.26**	**68.70**	**52.87**
ShortGPT	25%	Pruned	2088.56	2463.35	442.93
		LinearPatch-D	365.54	329.89	200.14
		LinearPatch-R	195.61	224.41	107.67
		Prune&Comp	529.48	519.01	166.70
		ReplaceMe-LS	26.37	42.50	41.49
		ReplaceMe-Cos	29.85	46.47	39.89
		DFR-512	26.24	44.73	41.88
		DFR-Full	**22.84**	**39.67**	**38.74**
Both	37.5%	Pruned	30616.01	15393.51	8448.06
		LinearPatch-D	3940.05	4130.44	1494.56
		LinearPatch-R	3982.29	4760.48	1026.43
		Prune&Comp	10617.61	11539.27	2279.31
		ReplaceMe-LS	80.12	475.67	212.11
		ReplaceMe-Cos	1151.88	67939.99	1186.51
		DFR-512	73.73	416.42	190.66
		DFR-Full	**47.18**	**280.72**	**125.19**
Both	50%	Pruned	8853.84	6908.69	3150.62
		LinearPatch-D	113961.81	132964.98	51983.73
		LinearPatch-R	42614.53	66296.01	15515.97
		Prune&Comp	56240.67	88842.74	20325.44
		ReplaceMe-LS	384.52	5140.84	1599.38
		ReplaceMe-Cos	1635.20	**2591.60**	2450.31
		DFR-512	279.52	4204.54	1215.36
		DFR-Full	**179.40**	2861.19	**710.86**

by $G_{\mathrm{inac}} = G_{\mathrm{DFR}} - G_{\mathrm{Readout}}$. The first recovery mode is almost entirely expressible within the pretrained readout subspace, while accessibility decreases progressively for later modes.

Table 15 provides a finer decomposition for DeepSeek-R1-Distill-Llama-8B. The overall recovery gain is concentrated toward the leading modes, while the inaccessible gain is distributed more heavily toward later parts of the spectrum.

D.4 Numerical Verification of the Gap Identity

Equation (11) states that the DFR–Readout objective gap equals the sum of mode-wise inaccessible gains. We verify this identity by computing the objective gap directly and comparing it with $\sum_i \mu_i(1-\eta_i^C)$.

Table 16 reports the relative difference between the two quantities. The errors are on the order of numerical precision for all three models.

E Additional Model Results

E.1 Additional Model Setup

We further evaluate DFR on Qwen and Mistral models to examine whether the recovery behavior observed extends to additional architectures. We use Qwen3-8B-Base (Yang et al., 2025) and Mistral-7B-v0.3 (Jiang et al., 2023). For each model, we construct pruning layouts using the same layer-selection procedures as in the main experiments.

Table 8: Task-wise zero-shot results for Llama-3-8B.

Selector	Pruning	Method	ARC-E	ARC-C	Hella	WinoG	BoolQ	OBQA	RTE	COPA	RACE	Avg.
Both	25%	Pruned	38.05	31.31	31.54	53.99	37.77	28.20	63.18	60.00	24.59	40.96
		LinearPatch-D	43.10	31.66	41.67	60.77	42.08	32.60	68.23	70.00	27.94	46.45
		LinearPatch-R	49.62	33.96	44.80	62.27	42.32	32.80	66.79	65.00	28.71	47.36
		Prune&Comp	44.87	32.25	40.10	57.06	40.52	31.60	66.06	70.00	27.27	45.53
		ReplaceMe-LS	60.45	40.00	60.50	69.96	59.58	33.00	66.51	73.00	33.62	55.18
		ReplaceMe-Cos	61.83	42.58	60.02	72.53	52.48	33.40	62.82	72.00	35.89	54.84
		DFR-512	60.86	40.61	60.10	72.14	54.98	34.00	67.51	76.00	34.83	55.67
		DFR-Full	64.27	42.32	63.53	71.67	47.95	36.80	67.51	76.00	35.98	**56.23**
Both	37.5%	Pruned	37.08	30.29	37.75	54.93	56.73	28.00	64.98	60.00	26.79	44.06
		LinearPatch-D	44.87	35.24	49.31	59.98	62.97	31.40	63.54	71.00	31.58	49.99
		LinearPatch-R	44.78	34.64	41.44	58.41	67.80	30.80	61.01	73.00	30.05	49.10
		Prune&Comp	43.81	33.79	39.56	56.04	65.41	30.00	60.29	67.00	30.33	47.36
		ReplaceMe-LS	44.65	33.87	43.39	65.90	73.00	30.00	59.21	65.00	30.14	49.46
		ReplaceMe-Cos	44.91	34.73	45.77	63.77	64.62	31.40	57.04	68.00	33.30	49.28
		DFR-512	43.27	33.28	42.36	67.64	75.54	30.80	57.04	68.00	31.10	49.89
		DFR-Full	51.18	35.15	48.85	67.88	74.04	33.00	65.34	76.00	32.82	**53.81**
LLM-Streamline	50%	Pruned	30.09	28.50	31.76	53.35	64.10	28.20	57.76	58.00	23.54	41.70
		LinearPatch-D	30.81	29.35	28.29	55.17	62.17	30.00	57.76	60.00	24.50	42.01
		LinearPatch-R	30.81	29.01	28.23	53.43	62.17	30.80	54.87	66.00	23.92	42.14
		Prune&Comp	31.36	28.84	29.88	54.46	62.17	29.80	57.40	65.00	24.59	42.61
		ReplaceMe-LS	31.10	28.58	31.08	60.06	60.89	29.40	54.15	59.00	24.69	42.11
		ReplaceMe-Cos	32.70	28.92	38.86	58.17	60.92	29.80	50.90	58.00	26.32	42.73
		DFR-512	32.27	29.16	32.60	64.38	56.57	28.40	53.71	63.00	25.47	42.84
		DFR-Full	33.71	28.16	33.76	64.25	62.51	29.20	55.96	66.00	24.98	**44.28**
ShortGPT	50%	Pruned	31.19	27.90	29.07	54.14	61.96	29.80	54.87	60.00	26.12	41.67
		LinearPatch-D	33.38	29.35	31.02	53.67	62.17	30.80	54.87	57.00	24.31	41.84
		LinearPatch-R	34.13	28.07	30.55	54.70	62.17	29.60	53.07	61.00	25.17	42.05
		Prune&Comp	33.71	29.01	31.89	54.70	62.20	30.00	54.15	61.00	24.50	42.35
		ReplaceMe-LS	35.35	28.41	32.99	60.30	38.87	29.80	57.04	64.00	26.22	41.44
		ReplaceMe-Cos	35.12	26.84	39.72	57.91	60.97	28.00	51.07	56.00	24.35	42.22
		DFR-512	35.98	29.18	33.48	62.98	38.38	30.60	59.21	63.00	26.41	42.14
		DFR-Full	39.86	31.06	36.41	62.83	38.75	31.40	60.65	73.00	27.18	**44.57**

Qwen3-8B-Base contains 36 Transformer blocks. We remove 9, 14, and 18 blocks for the three pruning settings, corresponding to actual pruning ratios of 25%, 38.89%, and 50%, respectively. Mistral-7B-v0.3 contains 32 Transformer blocks and uses the same 25%, 37.5%, and 50% pruning ratios as the main models.

Recovery uses the same calibration, regularization, and numerical protocol described in Appendix B. The exact pruning layouts are listed in Table 17.

E.2 Recovery Performance

Tables 18 and 19 report recovery performance on Qwen3-8B-Base and Mistral-7B-v0.3, respectively. We report perplexity on WikiText-2, PTB, and C4 together with the nine-task zero-shot average. The additional models show the same overall pattern as the main experiments. DFR-Full provides the strongest perplexity recovery in most settings, with particularly consistent improvements on WikiText-2 and C4, while downstream performance is also generally improved.

E.3 Recovery Structure on Additional Models

Table 20 compares DFR-Full with the optimal readout-constrained solution and summarizes recovery at rank 512. Both models exhibit the same qualitative structure: DFR-Full improves over Readout, while functional recovery at rank 512 exceeds the fraction of the regularized recovery spectrum retained at the same rank.

Table 9: Task-wise zero-shot results for Llama-3.1-8B.

Selector	Pruning	Method	ARC-E	ARC-C	Hella	WinoG	BoolQ	OBQA	RTE	COPA	RACE	Avg.
Both	25%	Pruned	41.50	31.74	30.87	54.30	37.58	29.20	64.62	64.00	25.26	42.12
		LinearPatch-D	47.64	33.79	44.72	61.17	43.91	34.40	68.95	68.00	27.56	47.79
		LinearPatch-R	55.13	36.09	49.85	62.75	43.15	32.80	69.68	68.00	30.05	49.72
		Prune&Comp	48.70	32.68	43.67	57.46	40.76	32.80	68.95	71.00	27.18	47.02
		ReplaceMe-LS	61.92	40.00	60.17	69.64	59.34	32.60	70.12	73.00	33.88	55.63
		ReplaceMe-Cos	63.05	42.24	59.88	71.27	51.74	34.60	65.34	74.00	37.51	55.51
		DFR-512	62.16	41.13	59.70	71.03	59.02	33.60	70.40	75.00	35.31	56.37
		DFR-Full	65.45	43.17	63.52	71.43	52.48	37.00	72.20	73.00	36.46	**57.19**
Both	37.5%	Pruned	39.18	30.63	35.14	55.01	55.99	29.80	67.87	63.00	26.79	44.83
		LinearPatch-D	48.95	35.24	48.84	59.83	62.75	31.00	68.95	67.00	30.33	50.32
		LinearPatch-R	49.41	33.96	43.04	58.25	66.18	31.00	65.70	65.00	31.00	49.28
		Prune&Comp	47.73	33.96	40.91	57.38	65.08	30.80	68.23	66.00	28.80	48.77
		ReplaceMe-LS	44.61	32.68	42.79	65.27	71.87	31.20	55.96	71.00	29.57	49.44
		ReplaceMe-Cos	45.41	33.53	44.23	64.01	65.17	33.00	61.73	62.00	32.92	49.11
		DFR-512	44.19	32.94	41.41	67.09	73.82	30.20	54.51	69.00	30.24	49.27
		DFR-Full	52.44	35.49	48.73	67.48	71.83	33.40	65.34	73.00	31.67	**53.27**
LLM-Streamline	50%	Pruned	28.37	26.54	27.35	50.83	51.47	27.20	48.01	55.00	24.50	37.70
		LinearPatch-D	31.48	28.07	26.42	51.54	62.32	29.20	57.40	57.00	23.92	40.82
		LinearPatch-R	31.19	26.02	27.34	52.25	62.26	28.20	57.76	58.00	24.98	40.89
		Prune&Comp	31.23	27.13	28.28	52.49	62.23	29.40	54.87	57.00	25.07	40.86
		ReplaceMe-LS	30.13	26.54	28.58	56.99	39.24	28.60	52.71	59.00	24.50	38.48
		ReplaceMe-Cos	30.26	24.57	32.83	56.83	54.25	28.80	52.35	54.00	25.84	39.97
		DFR-512	29.42	26.71	29.67	58.01	41.87	28.80	52.71	61.00	23.54	39.08
		DFR-Full	32.79	27.65	32.05	59.91	56.36	28.60	53.07	58.00	24.69	**41.46**
ShortGPT	50%	Pruned	32.03	29.10	30.36	53.99	62.48	31.00	57.04	57.00	25.07	42.01
		LinearPatch-D	33.84	30.03	30.75	54.85	62.20	32.00	55.23	56.00	24.11	42.11
		LinearPatch-R	35.14	28.50	30.63	53.67	62.20	30.20	52.71	62.00	26.22	42.36
		Prune&Comp	35.14	28.24	31.62	54.14	62.20	30.80	53.07	62.00	26.32	42.61
		ReplaceMe-LS	35.19	27.65	31.77	60.62	38.07	30.20	52.71	64.00	25.93	40.68
		ReplaceMe-Cos	38.47	30.20	38.33	62.43	63.85	29.40	52.71	65.00	27.08	**45.27**
		DFR-512	37.65	31.52	34.36	62.30	41.39	31.20	55.43	64.00	27.44	42.81
		DFR-Full	39.73	31.40	35.71	61.80	41.56	30.20	54.87	70.00	26.99	43.58

F Iterative Pruning Details

F.1 Iterative BI Selection

We implement the iterative pruning procedure from Section 2.5 using the Block Influence (BI) criterion of ShortGPT (Men et al., 2025). At pruning step t, let M_t denote the current pruned-and-recovered model. We recompute BI for every remaining candidate block as

$$\mathrm{BI}_t(\ell) = 1 - \frac{1}{NL} \sum_{n=1}^{N} \sum_{j=1}^{L} \cos\left(h_{\ell,t}^{\mathrm{in}}(n,j), h_{\ell,t}^{\mathrm{out}}(n,j)\right), \tag{83}$$

where $h_{\ell,t}^{\mathrm{in}}$ and $h_{\ell,t}^{\mathrm{out}}$ denote the hidden representations before and after block ℓ. We use all 1,024 calibration sequences for BI scoring.

The block with the smallest BI score is removed. DFR is then fitted at the newly created pruning boundary using the same 1,024-sequence calibration set as in the main recovery experiments, and the fitted correction is merged into the surviving FFN down projection. The resulting recovered model becomes M_{t+1}, on which BI is recomputed before the next pruning step. The procedure continues until the target number of blocks has been removed. No additional unified refit is applied after the final pruning step.

F.2 Selected Layers

Table 21 reports the pruning layouts obtained by the fixed-mask and iterative BI procedures on DeepSeek-R1-Distill-Llama-8B. Block indices are zero-based.

Table 10: Task-wise zero-shot results for DeepSeek-R1-Distill-Llama-8B.

Selector	Pruning	Method	ARC-E	ARC-C	Hella	WinoG	BoolQ	OBQA	RTE	COPA	RACE	Avg.
LLM-Streamline	25%	Pruned	45.83	33.79	45.47	61.56	37.46	30.40	64.98	72.00	28.80	46.70
		LinearPatch-D	50.04	35.07	48.67	62.27	37.34	33.60	74.73	77.00	29.86	49.84
		LinearPatch-R	51.81	36.43	52.49	64.09	37.46	32.20	71.84	73.00	34.07	50.38
		Prune&Comp	50.00	34.22	50.22	63.61	36.54	34.00	68.95	75.00	30.53	49.23
		ReplaceMe-LS	54.17	35.92	57.87	65.75	37.37	33.20	75.81	79.00	38.95	53.11
		ReplaceMe-Cos	54.55	36.95	56.29	64.88	45.63	33.00	75.45	74.00	37.51	53.14
		DFR-512	53.58	36.60	57.47	65.75	37.37	33.20	75.45	75.00	37.80	52.47
		DFR-Full	55.30	37.63	58.48	67.40	37.37	33.40	76.17	79.00	39.62	**53.82**
ShortGPT	25%	Pruned	44.95	34.64	49.58	62.83	72.42	32.80	74.37	65.00	31.67	52.03
		LinearPatch-D	51.18	34.04	51.97	63.61	81.04	32.40	72.92	72.00	31.20	54.49
		LinearPatch-R	50.29	33.62	56.06	64.64	79.97	35.80	73.65	75.00	34.83	55.99
		Prune&Comp	50.29	33.87	54.80	64.25	79.69	35.00	72.56	70.00	33.11	54.84
		ReplaceMe-LS	54.42	35.49	58.05	65.82	77.58	33.00	74.37	76.00	39.33	**57.12**
		ReplaceMe-Cos	54.42	36.18	58.54	67.17	77.00	32.80	67.51	77.00	37.99	56.51
		DFR-512	53.62	36.01	57.67	65.59	73.70	33.80	72.92	78.00	38.18	56.61
		DFR-Full	55.26	36.77	58.56	66.77	55.57	35.40	72.92	78.00	38.85	55.35
Both	37.5%	Pruned	35.40	31.06	36.67	53.04	67.95	26.60	56.68	57.00	24.98	43.26
		LinearPatch-D	41.96	32.51	38.87	56.75	68.84	32.00	67.51	64.00	28.13	47.84
		LinearPatch-R	42.30	32.17	40.74	58.56	67.16	29.40	66.06	63.00	30.05	47.72
		Prune&Comp	40.91	32.59	38.38	55.09	65.87	29.20	64.26	61.00	28.33	46.18
		ReplaceMe-LS	40.74	32.68	42.37	61.33	68.87	29.00	72.56	62.00	32.06	49.07
		ReplaceMe-Cos	38.34	32.00	41.29	58.64	71.47	29.20	68.23	63.00	31.10	48.14
		DFR-512	40.53	32.08	41.85	62.19	69.85	30.40	67.51	64.00	31.96	48.93
		DFR-Full	43.77	33.96	45.16	62.19	68.78	31.00	70.40	68.00	34.16	**50.82**
Both	50%	Pruned	28.87	26.28	29.76	53.51	69.63	29.40	61.01	60.00	23.73	42.47
		LinearPatch-D	30.30	29.27	29.79	54.85	62.20	30.60	57.76	63.00	23.92	42.41
		LinearPatch-R	31.44	27.82	28.90	53.35	62.75	31.20	55.96	69.00	23.83	42.69
		Prune&Comp	30.89	27.82	30.08	54.93	62.48	30.40	58.48	66.00	23.54	42.74
		ReplaceMe-LS	28.87	27.47	25.50	55.49	72.23	29.00	53.79	62.00	24.31	42.07
		ReplaceMe-Cos	30.47	25.85	32.35	55.56	58.20	29.60	59.93	67.00	25.07	42.67
		DFR-512	29.12	27.05	25.99	58.01	65.26	31.60	67.51	61.00	25.55	43.45
		DFR-Full	31.57	28.24	29.06	58.25	66.15	31.20	63.90	69.00	25.55	**44.77**

Table 11: Cross-model rank structure under 25% LLM-Streamline pruning. r_p denotes the smallest rank capturing $p\%$ of the Full DFR regularized recovery gain. $F_{\mathcal{D}}(512)$ denotes the fraction of the DFR-Full NLL improvement recovered at rank 512 on dataset $\mathcal{D}$.

Model	r_{90}	r_{95}	r_{99}	$E_{\text{spec}}(512)$	$F_{\text{Wiki}}(512)$	$F_{\text{PTB}}(512)$	$F_{\text{C4}}(512)$
Llama-3-8B	748	1396	2831	86.61%	95.39%	94.27%	94.80%
Llama-3.1-8B	737	1385	2824	86.79%	95.17%	94.28%	94.06%
DeepSeek-R1-Distill-Llama-8B	773	1447	2895	86.39%	96.10%	96.47%	94.95%

G Deployment Details

G.1 Latency Measurement Protocol

We measure inference latency using DeepSeek-R1-Distill-Llama-8B, with each benchmark run performed on a single NVIDIA H200 GPU. The base model weights and merged recovery weights are evaluated in BF16. Explicit boundary-patch baselines retain their FP32 patch computations, consistent with their quality-evaluation implementations. For each pruning level, we measure the dense model, the unrecovered pruned model, and the corresponding recovered models under identical settings within the same benchmark run.

Latency is measured with batch size 1, a prompt length of 2,048 tokens, and greedy autoregressive generation of exactly 128 new tokens with KV caching enabled. Generation does not terminate early upon encountering an EOS token. We perform 3 warm-up iterations followed by 10 timed iterations. CUDA execution is synchronized at the start of the prefill phase, at the prefill–decode boundary, and

Table 12: Feature-accessibility controls under 25% LLM-Streamline pruning.

Model	Method	Wiki ↓	PTB ↓	C4 ↓
Llama-3-8B	Readout	16.79	41.78	38.57
	DFR-Full	**13.51**	**40.56**	**31.99**
Llama-3.1-8B	Readout	16.66	42.73	40.29
	DFR-Full	**13.50**	**41.44**	**32.99**
DeepSeek-R1-Distill-Llama-8B	Readout	34.08	74.98	66.28
	Random-Feature	35.00	86.30	67.42
	Drop-Perp	64.17	133.54	113.66
	DFR-Full	**27.26**	**68.70**	**52.87**

Table 13: Rank and feature-accessibility controls on DeepSeek-R1-Distill-Llama-8B under 25% LLM-Streamline pruning.

Method	Wiki ↓	PTB ↓	C4 ↓
Readout-512	37.74	80.21	75.00
DFR-512	**32.81**	**78.85**	**63.18**
Readout-Full	34.08	74.98	66.28
DFR-1024	**29.89**	**74.35**	**58.27**

after decoding. For each repetition, total latency is the sum of the prefill and decode times, and we report the median total latency over the measured runs.

Measurements use wall-clock timing and include Python execution overhead, while excluding model loading, tokenization, input transfer to the GPU, and recovery fitting. The benchmark uses eager execution with SDPA attention.

Speedup is computed relative to the dense model measured in the same benchmark run as

$$\text{Speedup} = \frac{T_{\text{dense}}}{T_{\text{model}}},$$

where T_{model} is the median total latency of the corresponding pruned or recovered model.

G.2 Full Quality–Latency Sweep

Table 22 reports the complete 8–16 block pruning sweep used to construct Fig. 4. For each pruning level, we report measured latency, speedup relative to the dense model, and WikiText-2 perplexity for the main recovery methods.

G.3 Pruned and DFR Latency

Table 23 compares the measured latency of the unrecovered pruned model and DFR at matched pruning depths. Because the DFR correction is merged into the existing FFN projection, the two models share the same inference graph.

Table 14: Cross-model recovery-mode accessibility. η^C denotes C-weighted accessibility to the pretrained readout subspace. $G_{\mathrm{inac}}/G_{\mathrm{DFR}}$ is the fraction of Full DFR regularized recovery gain inaccessible to Readout, and $G_{\mathrm{inac},>128}/G_{\mathrm{inac}}$ is the fraction of that inaccessible gain contributed by modes after the first 128.

Model	η_1^C	$\eta_{129:512}^C$	$\eta_{2049:4096}^C$	$G_{\mathrm{inac}}/G_{\mathrm{DFR}}$	$G_{\mathrm{inac},>128}/G_{\mathrm{inac}}$
Llama-3-8B	99.42%	64.75%	31.67%	15.40%	75.07%
Llama-3.1-8B	99.47%	64.72%	31.66%	15.15%	75.20%
DeepSeek-R1-Distill-Llama-8B	98.88%	61.28%	31.29%	16.95%	71.75%

Table 15: Mode-wise decomposition for DeepSeek-R1-Distill-Llama-8B. Mean accessibility is the arithmetic mean within each mode range. Total gain share and inaccessible gain share are normalized by the Full DFR regularized recovery gain and the total inaccessible gain, respectively.

Mode range	Mean η^C	Share of total gain	Share of inaccessible gain
1	98.88%	42.79%	2.83%
2–8	92.51%	11.66%	4.74%
9–32	86.98%	9.15%	6.66%
33–128	76.62%	10.74%	14.02%
129–512	61.28%	12.06%	26.30%
513–1024	49.59%	5.96%	17.52%
1025–2048	40.35%	4.92%	17.10%
2049–4096	31.29%	2.72%	10.83%

Table 16: Numerical verification of the DFR–Readout gap decomposition. Relative error compares the directly computed objective gap with the sum of mode-wise inaccessible gains.

Model	Relative closure error
Llama-3-8B	4.3×10^{-15}
Llama-3.1-8B	7.8×10^{-16}
DeepSeek-R1-Distill-Llama-8B	4.3×10^{-15}

Table 17: Additional model configurations and pruning layouts. Block indices are zero-based.

Model	Selector	Pruning Ratio	Removed Blocks
Qwen3-8B-Base	Both	25%	$\{25, \ldots, 33\}$
	LLM-Streamline	38.89%	$\{13, \ldots, 26\}$
	ShortGPT	38.89%	$\{15, \ldots, 18\} \cup \{20\} \cup \{25, \ldots, 33\}$
	LLM-Streamline	50%	$\{13, \ldots, 30\}$
	ShortGPT	50%	$\{14, \ldots, 22\} \cup \{25, \ldots, 33\}$
Mistral-7B-v0.3	LLM-Streamline	25%	$\{22, \ldots, 29\}$
	ShortGPT	25%	$\{21, \ldots, 28\}$
	Both	37.5%	$\{19, \ldots, 30\}$
	LLM-Streamline	50%	$\{15, \ldots, 30\}$
	ShortGPT	50%	$\{13\} \cup \{16, \ldots, 30\}$

Table 18: Recovery results on Qwen3-8B-Base. Bold and underline denote the best and second-best recovery methods, respectively, excluding Pruned.

Selector	Pruning	Method	Wiki ↓	PTB ↓	C4 ↓	QA ↑
Both	25%	Pruned	721.81	1383.49	439.15	48.02
		LinearPatch-D	42.85	103.72	59.63	49.72
		LinearPatch-R	42.03	148.69	89.94	52.77
		Prune&Comp	39.39	<u>66.06</u>	<u>38.33</u>	<u>52.96</u>
		ReplaceMe-LS	<u>28.36</u>	134.95	81.51	51.00
		ReplaceMe-Cos	114.69	295.01	163.20	51.80
		DFR-512	30.03	67.39	38.73	52.23
		DFR-Full	**20.71**	**59.94**	**37.65**	**53.50**
LLM-Streamline	38.89%	Pruned	257.02	312.78	136.38	43.31
		LinearPatch-D	133.75	151.99	99.60	42.86
		LinearPatch-R	107.03	161.17	83.54	42.27
		Prune&Comp	93.27	<u>95.39</u>	77.50	<u>44.15</u>
		ReplaceMe-LS	36.56	394.85	101.18	41.05
		ReplaceMe-Cos	35.63	100.96	<u>59.69</u>	43.97
		DFR-512	<u>28.32</u>	138.21	82.24	43.05
		DFR-Full	**21.37**	**92.70**	**57.21**	**44.41**
ShortGPT	38.89%	Pruned	916.43	1549.77	1007.01	40.98
		LinearPatch-D	49.34	230.51	90.06	44.10
		LinearPatch-R	47.51	83.05	58.15	43.13
		Prune&Comp	47.26	<u>81.35</u>	51.53	43.31
		ReplaceMe-LS	38.87	209.21	127.88	42.13
		ReplaceMe-Cos	130.35	392.01	249.19	**45.80**
		DFR-512	<u>38.45</u>	159.28	<u>51.05</u>	42.89
		DFR-Full	**26.23**	**77.64**	**47.61**	<u>44.79</u>
LLM-Streamline	50%	Pruned	206422.09	100289.17	173487.89	37.79
		LinearPatch-D	514.35	1054.37	337.25	40.19
		LinearPatch-R	614.34	856.44	265.69	40.48
		Prune&Comp	269.36	**370.42**	<u>188.98</u>	<u>40.97</u>
		ReplaceMe-LS	80.50	1157.29	332.68	36.98
		ReplaceMe-Cos	1779.15	6570.45	14967.99	36.85
		DFR-512	<u>60.74</u>	555.58	268.89	39.57
		DFR-Full	**43.63**	<u>477.42</u>	**174.37**	**41.76**
ShortGPT	50%	Pruned	47472.80	129912.17	50419.25	41.89
		LinearPatch-D	124.57	469.17	132.44	43.04
		LinearPatch-R	118.84	297.84	<u>93.48</u>	**43.77**
		Prune&Comp	129.49	206.68	98.92	43.50
		ReplaceMe-LS	50.91	435.65	199.68	40.22
		ReplaceMe-Cos	223.32	708.00	573.12	43.32
		DFR-512	<u>50.22</u>	<u>180.75</u>	210.25	42.29
		DFR-Full	**33.74**	**180.44**	**93.01**	<u>43.64</u>

Table 19: Recovery results on Mistral-7B-v0.3. Bold and underline denote the best and second-best recovery methods, respectively, excluding Pruned.

Selector	Pruning	Method	Wiki ↓	PTB ↓	C4 ↓	QA ↑
LLM-Streamline	25%	Pruned	42.18	157.52	39.44	52.94
		LinearPatch-D	29.57	89.27	31.30	53.83
		LinearPatch-R	23.49	**72.47**	24.92	55.88
		Prune&Comp	29.05	101.01	34.28	54.74
		ReplaceMe-LS	<u>14.67</u>	109.95	27.78	<u>57.31</u>
		ReplaceMe-Cos	14.99	89.54	<u>24.01</u>	57.22
		DFR-512	14.88	110.11	27.61	57.08
		DFR-Full	**11.32**	<u>78.81</u>	**22.05**	**58.42**
ShortGPT	25%	Pruned	37.85	111.27	40.23	54.09
		LinearPatch-D	18.81	76.15	19.64	<u>58.62</u>
		LinearPatch-R	17.15	69.40	18.61	58.50
		Prune&Comp	20.48	86.08	25.30	57.19
		ReplaceMe-LS	11.06	75.08	20.79	58.20
		ReplaceMe-Cos	<u>10.93</u>	66.52	<u>18.45</u>	58.32
		DFR-512	11.33	<u>64.57</u>	21.19	57.47
		DFR-Full	**9.39**	**60.69**	**17.24**	**59.11**
Both	37.5%	Pruned	1833.05	1997.90	1199.78	41.06
		LinearPatch-D	445.33	914.56	410.84	46.89
		LinearPatch-R	240.79	412.38	213.43	48.25
		Prune&Comp	423.00	631.78	541.41	44.45
		ReplaceMe-LS	48.06	586.96	180.78	48.05
		ReplaceMe-Cos	61.96	<u>306.61</u>	<u>108.37</u>	<u>49.11</u>
		DFR-512	<u>47.32</u>	546.58	171.56	47.84
		DFR-Full	**25.84**	**292.26**	**92.49**	**49.62**
LLM-Streamline	50%	Pruned	1836.75	2841.37	1515.73	40.50
		LinearPatch-D	4526.88	7654.77	4112.69	41.92
		LinearPatch-R	1974.59	2187.90	1602.60	42.00
		Prune&Comp	6093.18	4101.47	5796.35	40.02
		ReplaceMe-LS	138.08	2247.05	652.54	39.26
		ReplaceMe-Cos	229.29	**985.18**	<u>389.14</u>	<u>43.04</u>
		DFR-512	<u>103.22</u>	1800.50	503.64	39.99
		DFR-Full	**71.55**	<u>1213.57</u>	**335.35**	**43.69**
ShortGPT	50%	Pruned	1010.52	2128.33	825.45	40.93
		LinearPatch-D	2899.02	7022.83	2878.06	42.23
		LinearPatch-R	1351.51	1780.79	1174.90	**42.96**
		Prune&Comp	4452.23	3224.16	5207.14	40.51
		ReplaceMe-LS	119.55	1856.26	554.08	38.53
		ReplaceMe-Cos	193.97	<u>1036.30</u>	466.05	42.57
		DFR-512	<u>104.17</u>	1699.85	<u>299.66</u>	41.28
		DFR-Full	**62.63**	**770.54**	**272.99**	<u>42.71</u>

Table 20: Recovery structure on additional models under 25% LLM-Streamline pruning. The DFR–Readout columns report WikiText-2 perplexity under full recovery. $E_{\mathrm{spec}}(512)$ is the fraction of regularized recovery gain retained at rank 512, while $F_{\mathrm{Wiki}}(512)$ is the corresponding fraction of DFR-Full NLL improvement.

Model	Readout PPL ↓	DFR-Full PPL ↓	$E_{\mathrm{spec}}(512)$	$F_{\mathrm{Wiki}}(512)$
Qwen3-8B-Base	28.37	**20.71**	89.53%	94.31%
Mistral-7B-v0.3	14.67	**11.32**	79.22%	89.53%

Table 21: Pruning layouts used in the iterative pruning experiments on DeepSeek-R1-Distill-Llama-8B. The fixed masks are shared by LLM-Streamline and ShortGPT at these pruning ratios.

Pruning Ratio	Method	Removed Blocks
37.5%	Fixed Mask + DFR	$\{18, \ldots, 29\}$
	BI-Iterative + DFR	$\{13, 16, 18\} \cup \{20, \ldots, 22\} \cup \{24, \ldots, 29\}$
50%	Fixed Mask + DFR	$\{15, \ldots, 30\}$
	BI-Iterative + DFR	$\{10, 11, 13, 14\} \cup \{16, \ldots, 18\} \cup \{20, \ldots, 22\} \cup \{24, \ldots, 29\}$

Table 22: Full deployment sweep on DeepSeek-R1-Distill-Llama-8B.

Removed	DFR Latency	Speedup	ReplaceMe-LS PPL	DFR PPL
8	2161.7	1.31×	34.08	27.26
9	2077.0	1.36×	45.70	32.74
10	2018.2	1.42×	53.27	36.12
11	1929.5	1.49×	64.31	40.75
12	1823.8	1.55×	80.12	47.18
13	1737.7	1.63×	136.70	70.67
14	1646.5	1.71×	215.80	98.41
15	1588.7	1.81×	234.49	113.06
16	1489.2	1.90×	384.52	179.40

Table 23: Measured latency before and after DFR recovery.

Removed Blocks	Pruned (ms)	DFR (ms)	DFR − Pruned (ms)
8	2161.6	2161.7	+0.1
12	1824.0	1823.8	-0.2
16	1489.3	1489.2	-0.1