LAYER-PROPAGATION EQUIVALENCE COMPRESSION FOR GRADIENT-BASED DATA SELECTION IN LLM FINE-TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-based data selection identifies useful training examples by comparing their gradients with those of a target validation set, which is crucial to large language model (LLM) supervised fine-tuning (SFT). Computing these gradients with an LLM scales poorly, making it impractical for multi-billion-parameter models, motivating the construction of smaller proxy models through compression for gradient-influence equivalence to preserve the gradient information used for data selection. However, existing compression objectives evaluate each weight matrix separately, without accounting for how the resulting errors propagate across layers; we find that these errors can persist and become amplified, potentially altering the gradient alignment scores used for data selection. To address this issue, we introduce Layer-propagation Equivalence Compression (LPEC), a method that accounts for error propagation when constructing a low-rank compressed proxy. Specifically, we perform propagation-aware direction ranking by considering both singular values and how removal errors propagate to other layers. We then perform joint rank allocation across weight matrices under a shared parameter budget, giving more rank to matrices where retaining additional components is expected to reduce compression error more effectively. Experimental results across different LLM families and evaluation tasks demonstrate that LPEC improves downstream SFT performance over existing influence-preserving compression methods. Anonymous code is available at https://anonymous.4open.science/r/LEPC-119E/.

1 INTRODUCTION

Supervised fine-tuning (SFT) adapts a large language model (LLM) to a target task. Its final quality often depends on the training examples used for adaptation. Large candidate datasets are useful, but they also contain repeated, mismatched, and low-value examples. Simply combining more datasets can therefore hurt downstream performance (Wang et al., 2023b; 2024). The practical goal is to select a small subset that gives the greatest improvement on a target set.

Gradient-based data selection offers a direct way to make this choice. It compares a candidate gradient with gradients from the target set, so examples that push the model in a useful direction receive higher scores. TracIn (Pruthi et al., 2020; Xia et al., 2024) accumulates these comparisons over training checkpoints. Influence functions (Koh & Liang, 2017; Kwon et al., 2024; Zhang et al., 2024; Wang et al., 2025a) additionally account for local loss curvature. Both are expensive for the models and datasets where selection matters most. TracIn requires repeated target-model backward passes at many checkpoints. Influence functions require inverse-Hessian vector products in addition to gradients. In either case, the per-candidate cost grows with both model size and the size of the candidate dataset (Grosse et al., 2023).

A smaller proxy model can instead move all repeated candidate passes away from the target model. Small models are already used to estimate properties of larger ones, including through scaling relations (Kaplan et al., 2020; Shum et al., 2025; Lin et al., 2025a; Zou et al., 2025). An off-the-shelf proxy is convenient, yet its training history, size, and behavior need not match the target model (Yang et al., 2024). Its size is fixed before the selection problem is specified. A more direct alternative de-

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.

rives a proxy from the target weights and chooses its size through compression. Influence-preserving compression follows this route by replacing target matrices with low-rank approximations that retain a local gradient signal (Chen et al., 2026).

The low-rank compression objective in existing influence-preserving work evaluates each weight matrix separately (Chen et al., 2026). This captures the local gradient effect of replacing one matrix, but does not account for the errors introduced when many matrices are compressed together. An error at one weight matrix changes the forward activations consumed by later layers and the backward gradients returned through earlier layers. Our measurements reveal two consequences. First, a single layer's compression error persists across much of the network, with amplification at large layer distances in Qwen3-4B (Figure 1). Second, the errors from jointly compressed matrices cannot be explained by adding their individual errors: the unexplained residual energy grows with sparsity and exceeds the joint error energy at high sparsity. These observations expose the limitation of evaluating truncations independently: a locally accurate approximation can still change the gradient alignment used for data selection through its effects on other layers.

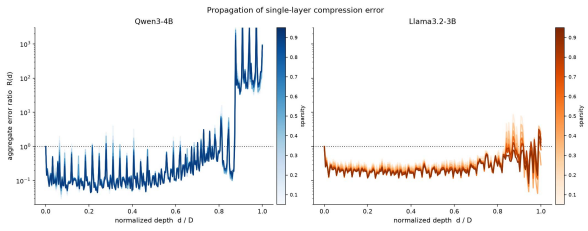

Figure 1: Single-layer SVD truncation error ratio $R(d)$ versus relative depth (Qwen3-4B left, Llama3.2-3B right): plateaus near one fifth of the source level; far-tail amplification is specific to Qwen3-4B.

To address this problem, we introduce Layer-Propagation Equivalence Compression (LPEC), which compresses the proxy's weight matrices using a criterion for the propagated first-order effects of their errors on gradient alignment across layers. Here, "equivalence" refers to this selection-relevant criterion, not exact functional equivalence to the target model. Under a simplified sequential model, our analysis bounds the first-order change in gradient alignment over the compressed matrices by compression-error terms weighted by their propagation through the network. Within a matrix, LPEC ranks low-rank directions by their error energy and estimated forward and backward propagation strength. Across matrices, it reallocates one shared rank budget using both propagation strength and measured interactions between compression errors. A per-matrix cap keeps every selected matrix compressed. These two decisions account for how a truncation affects other layers when constructing the proxy.

We evaluate LPEC under matched selection budgets and fine-tuning protocols across target models, candidate datasets, and evaluation tasks. In the evaluated settings, it improves downstream SFT performance over existing influence-preserving compression methods. Our contributions are threefold.

Alignment-equivalence analysis under compression. We characterize how simultaneous low-rank compression changes a restricted gradient alignment through forward and backward error propagation.

Propagation-aware direction ranking. We retain low-rank directions using their error energy and estimated propagation strength.

Joint rank allocation. We distribute rank jointly across selected weight matrices under one parameter budget, per-matrix compression caps, and measured error interactions.

2 Related Work

Gradient-Based Data Selection. Data selection methods are either model-agnostic, curating data by diversity, redundancy, or distributional similarity (Xie et al., 2024; Liu et al., 2024), or model-aware, scoring each candidate by its effect on training. Among the model-aware ones, influence functions (Koh & Liang, 2017; Kwon et al., 2024; Grosse et al., 2023) approximate the leave-one-out effect of a training sample, and TracIn-style methods (Pruthi et al., 2020; Han et al., 2023) accumulate gradient alignments along the optimization trajectory (Bae et al., 2024). Efficient variants reduce the second-order computation (Zhang et al., 2024; Kwon et al., 2024), score a subsample and extrapolate (Xia et al., 2024; Yu et al., 2024; Gu et al., 2024; Lin et al., 2025b), or use parameter-efficient gradient representations such as LoRA adapter gradients (Hu et al., 2022; Xia et al., 2024).

These approaches differ in how they represent or estimate the selection signal, but they still rely on target-model gradients or target-derived supervision.

Proxy Models for Data Selection. A complementary line scores candidates with a smaller proxy rather than the target. Off-the-shelf small models are common (Yang et al., 2024; Shum et al., 2025), but their dynamics can differ from the target's. Target-derived proxies instead truncate each weight matrix to retain a local influence signal (Chen et al., 2026), controlling proxy size from the target weights. We extend this line by designing the compression criterion for joint compression of all selected matrices.

LLM Compression via Decomposition. Decomposition methods exploit low-rank structure through standard SVD, activation-aware factorization, low-rank/quantization hybrids, or block removal (Ganesh et al., 2021; Yuan et al., 2025; Saha et al., 2024; Men et al., 2024). Error-aware variants weight task-sensitive directions or distribute compression by estimated truncation loss (Hsu et al., 2022; Wang et al., 2025c;b), while SAES-SVD couples local and accumulated errors (Hu et al., 2026). Unlike objectives that target reconstruction or output quality, LPEC evaluates direction-specific forward and backward propagation and jointly allocates rank under empirical interactions to preserve gradient-based selection signals.

3 Preliminaries and Problem Definition

Problem Setup. Let $\mathcal{D}_{\text{train}}$ be a dataset of candidate training examples and $\mathcal{D}_{\text{target}}$ be a target set. Model-aware data selection seeks a subset $\mathcal{D}^* \subseteq \mathcal{D}_{\text{train}}$ of selection budget B_{sel} that gives the best target performance after fine-tuning:

$$\mathcal{D}^* = \underset{\mathcal{D}\subseteq\mathcal{D}_{\text{train}},\ |\mathcal{D}|=B_{\text{sel}}}{\arg\max} \ \mathbb{E}_{z'\sim\mathcal{D}_{\text{target}}}\big[\mathcal{U}(f_{\theta(\mathcal{D})}; z')\big]. \tag{1}$$

Here $\mathcal{U}$ is a task utility and $\theta(\mathcal{D})$ denotes the parameters after fine-tuning on $\mathcal{D}$. Directly solving equation 1 is impractical, so gradient-based methods score each candidate by how it aligns with the target set and select the top-B_{sel} examples.

Gradient Alignment. TracIn (Pruthi et al., 2020) compares gradients over training checkpoints,

$$I_{\text{TracIn}}(z, z') = \sum_{t=1}^{T} \eta_t \big\langle \nabla_\theta \mathcal{L}(z;\theta_t), \nabla_\theta \mathcal{L}(z';\theta_t) \big\rangle, \tag{2}$$

where $\mathcal{L}$ is the training loss and η_t is the learning rate at checkpoint t. Influence functions (Koh & Liang, 2017) use a related alignment weighted by loss curvature. Our analysis uses a single-checkpoint alignment evaluated over the matrices compressed by the proxy. With $\theta_{\mathcal{T}} = (W_1, \ldots, W_L)$, define

$$I_{\mathcal{T}}(z, z') = \big\langle \nabla_{\theta_{\mathcal{T}}} \mathcal{L}(z;\theta), \nabla_{\theta_{\mathcal{T}}} \mathcal{L}(z';\theta) \big\rangle. \tag{3}$$

This is the analytical score whose change under compression we study.

A Compressed Proxy. Scoring every candidate with the target model requires a backward pass and a large gradient for each example, so the cost grows with both model size and dataset size (Grosse et al., 2023). We instead score candidates with a low-rank proxy built from the target model, whose smaller weights reduce the cost of its forward and backward passes.

Joint Compression Setting. Let $\mathcal{T} = \{W_k \in \mathbb{R}^{m_k \times n_k}\}_{k=1}^{L}$ be the selected attention and MLP matrices, where L is the number of selected linear matrices rather than the number of Transformer blocks. The embedding, LM head, biases, normalization parameters, and other unselected parameters lie outside $\theta_{\mathcal{T}}$. In the simplified sequential model used for analysis, $h_{k-1}(z)$ is the input of W_k, $u_k = W_k h_{k-1}$ is its pre-activation, and $\delta_k(z)$ is the derivative of the loss with respect to u_k. Thus $\nabla_{W_k} \mathcal{L}(z;\theta) = \delta_k(z) h_{k-1}(z)^\top$. Joint compression replaces every selected matrix at once:

$$W_k \mapsto \widehat{W}_k = W_k - E_k, \qquad k = 1, \ldots, L. \tag{4}$$

The next section studies how the errors $\{E_k\}$ change equation 3 and uses that analysis to build the compression proxy.

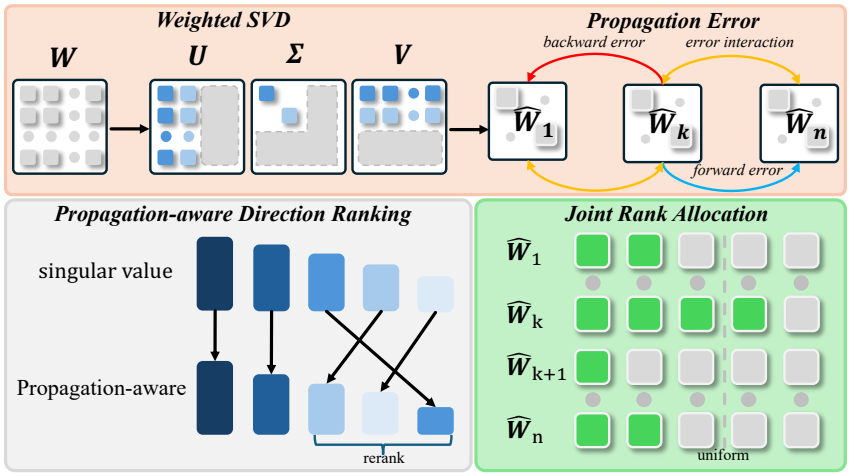

Figure 2: Overview of LPEC. Weighted low-rank components are constructed for selected matrices; their forward and backward propagation through the network ranks directions by propagated error. Joint rank allocation then retains the highlighted components under a shared factor-storage budget and per-matrix caps, producing the compressed matrices $\widehat{W}_k$.

4 METHOD

LPEC constructs a compressed proxy by considering both the local error caused by removing a weight component and how that error affects other layers. For each $W_k \in \mathbb{R}^{m_k \times n_k}$, we construct

$$\widehat{W}_k(r_k) = \sum_{a \in \mathcal{S}_k(r_k)} \sigma_{k,a} p_{k,a} q_{k,a}^\top, \qquad |\mathcal{S}_k(r_k)| = r_k. \tag{5}$$

A weighted reconstruction problem supplies the rank-one components $\sigma_{k,a} p_{k,a} q_{k,a}^\top$. We then use their propagated effects to decide which components to retain, $\mathcal{S}_k$, and how many to retain, r_k, under a shared storage budget. The analysis below motivates these three steps; Figure 2 summarizes the resulting construction.

4.1 HOW COMPRESSION AFFECTS GRADIENT ALIGNMENT

Before constructing $\widehat{W}_k$, we need to understand which compression errors change gradient alignment. A small reconstruction error at one matrix need not have a small effect on the selection score, because it also changes inputs and loss gradients at other matrices. We separate the local error from its propagation to identify what the compression criterion should account for.

In the sequential model $h_k = \phi_k(W_k h_{k-1})$, a matrix gradient is $\delta_k h_{k-1}^\top$; it depends on inputs from earlier layers and loss gradients from later layers. Writing $\Delta h_k = \widehat{h}_k - h_k$ and $E_k = W_k - \widehat{W}_k$, we have

$$\Delta u_k = W_k \Delta h_{k-1} - E_k h_{k-1} - E_k \Delta h_{k-1}. \tag{6}$$

The first two terms carry earlier errors forward and introduce the current local error; the last term couples them. Under **(H1)**, we hold the input, layer derivatives, and final loss gradient at their uncompressed values. Retaining terms linear in E then traces $E_k h_{k-1}$ to later inputs through path operators $\mathcal{F}_{\ell-1 \leftarrow k}$, and $E_k^\top \delta_k$ to earlier loss gradients through $\mathcal{B}_{\ell \leftarrow k}$ (Appendix D).

Let $\overline{C}_{h,k}$ and $\overline{C}_{\Delta,k}$ denote the input and loss-gradient second moments, averaged equally over the candidate and target distributions. With finite data and propagation-gain moments **(H2)**, these paths yield:

Proposition 4.1 (Conditional first-order bound)**.** *Under (H1)–(H2), let $\Delta I_{\mathcal{T}}^{[1]}$ denote the terms linear in the compression errors in the change of the alignment $I_{\mathcal{T}}$ defined in equation 3. There exist finite*

nonnegative coefficients A_k, D_k, determined by the uncompressed model and the candidate and target distributions, such that

$$\mathbb{E}_{z,z'}\left|\Delta I_{\mathcal{T}}^{[1]}\right| \;\le\; \sum_{k=1}^{L}\left[A_k\sqrt{\operatorname{tr}(E_k^\top E_k \overline{C}_{h,k})} + D_k\sqrt{\operatorname{tr}(E_k E_k^\top \overline{C}_{\Delta,k})}\right]. \tag{7}$$

The square roots measure local forward and backward errors, while A_k, D_k combine path norms with activation and gradient amplitudes. Thus, equally large local errors can receive different weights depending on where they occur. This gives two design insights. First, reconstruction should account for input and gradient statistics. Second, the scalar path bounds discard direction-specific propagation, so choosing components requires returning to the paths themselves. We use the local terms to construct a weighted basis, then evaluate propagation within that basis and across matrices.

4.2 Constructing the Low-Rank Components

This step supplies the candidate rank-one components for $\widehat{W}_k$. The local terms in Proposition 4.1 suggest preserving directions with large input or loss-gradient magnitudes, since compression in these directions produces larger local errors. We therefore build a decomposition using both statistics.

To obtain a closed-form construction, we combine the two one-sided weighted norms. Adding a positive ridge λ bounds their sum by a constant times a two-sided weighted norm (Corollary D.2). This gives an auxiliary reconstruction problem with an SVD solution, as in (Chen et al., 2026). The ridge also stabilizes the inverse weights needed to map components back to the original coordinates.

In practice, collect N probe inputs and loss gradients as columns of $H_k \in \mathbb{R}^{n_k\times N}$ and $\Delta_k \in \mathbb{R}^{m_k\times N}$. Their empirical moments are $\widehat{C}_{h,k} = H_k H_k^\top/N$ and $\widehat{C}_{\Delta,k} = \Delta_k\Delta_k^\top/N$. Let U_H, U_Δ contain the orthonormal eigenvectors for their nonzero eigenvalues, and let D_H, D_Δ be diagonal matrices with entries $\sqrt{\nu+\lambda}$ for the corresponding eigenvalues ν. Thus, U specifies the sampled subspaces and D weights their directions by observed magnitudes. Within these subspaces, we solve

$$\min_{\operatorname{rank}(X)\le r}\left\|D_\Delta\big(U_\Delta^\top W_k U_H - X\big)D_H\right\|_F^2, \qquad \widehat{W}_k = U_\Delta X U_H^\top. \tag{8}$$

The rank-preserving change of variables $Y = D_\Delta X D_H$ reduces this to an ordinary low-rank approximation of

$$S_{\mathrm{core},k} = D_\Delta(U_\Delta^\top W_k U_H)D_H = P\,\mathrm{diag}(\sigma_k)R^\top. \tag{9}$$

Keeping the largest r singular values minimizes equation 8, with residual $\sum_{a=r}^{R_k-1}\sigma_{k,a}^2$, where $R_k = \operatorname{rank}(S_{\mathrm{core},k})$. Undoing the weighting gives the components in the original coordinates:

$$p_{k,a} = U_\Delta D_\Delta^{-1}P_{:,a}, \qquad q_{k,a} = U_H D_H^{-1}R_{:,a}, \qquad a\in\{0,\dots,R_k-1\}. \tag{10}$$

The directions need not have unit Euclidean norm. These components supply equation 5, but their singular-value order optimizes only the auxiliary empirical reconstruction objective. We therefore reconsider which to retain using propagation. All components together recover the projection of W_k onto the sampled subspaces; error outside them remains fixed. Appendix F gives the derivation and distinguishes this empirical construction from the population bound.

4.3 Selecting Components by Their Effects on Other Layers

Given the candidate components, we now determine which r_k to retain in each matrix. The singular values quantify local weighted reconstruction error, but components with similar singular values can have very different effects on other layers. We therefore rank components by the propagated error caused by discarding them.

To assess this effect, consider discarding $E_{k,a} = \sigma_{k,a}p_{k,a}q_{k,a}^\top$. Its forward error is $\sigma_{k,a}p_{k,a}(q_{k,a}^\top h_{k-1})$: the input sets its magnitude along $p_{k,a}$, and the paths determine how it propagates. Hence,

$$\mathbb{E}_z\sum_{\ell>k}\|\mathcal{F}_{\ell-1\leftarrow k}E_{k,a}h_{k-1}\|^2 = \sigma_{k,a}^2\mathbb{E}_z\left[(q_{k,a}^\top h_{k-1})^2\sum_{\ell>k}\|\mathcal{F}_{\ell-1\leftarrow k}p_{k,a}\|^2\right]. \tag{11}$$

The backward expression is analogous. This separates the data-dependent size of the error from the propagation of its direction, explaining why singular values alone are insufficient.

We summarize these factors by propagation strengths d and input/gradient moments u:

$$\begin{aligned} d_{k,a}^{\mathrm{fwd}} &= \mathbb{E}_z \sum_{\ell>k} \|\mathcal{F}_{\ell-1\leftarrow k} p_{k,a}\|^2, \quad & u_{k,a}^{\mathrm{fwd}} &= q_{k,a}^\top \widehat{C}_{h,k} q_{k,a}, \\ d_{k,a}^{\mathrm{bwd}} &= \mathbb{E}_z \sum_{\ell<k} \|\mathcal{B}_{\ell\leftarrow k} q_{k,a}\|^2, & u_{k,a}^{\mathrm{bwd}} &= p_{k,a}^\top \widehat{C}_{\Delta,k} p_{k,a}. \end{aligned} \tag{12}$$

Random-sign probes and vector–Jacobian products estimate d without forming path matrices (Appendix G). Path and data factors depend on the same example; approximation **(H3a)** replaces their joint expectation by the product of their moments. To assign fixed component costs, **(H3b)** omits cross-terms between discarded components. This gives costs $\sigma_{k,a}^2(d_{k,a}^{\mathrm{fwd}} u_{k,a}^{\mathrm{fwd}} + d_{k,a}^{\mathrm{bwd}} u_{k,a}^{\mathrm{bwd}})$. To balance the two channels, divide each d and u by its positive mean over all components within matrix k, giving

$$\mathrm{key}_{k,a} = \sigma_{k,a}^2 \left(\widetilde{d}_{k,a}^{\mathrm{fwd}} \widetilde{u}_{k,a}^{\mathrm{fwd}} + \widetilde{d}_{k,a}^{\mathrm{bwd}} \widetilde{u}_{k,a}^{\mathrm{bwd}} \right). \tag{13}$$

We retain the r_k largest costs, which minimizes their discarded sum. This can favor a smaller singular component whose error propagates more strongly. The criterion approximates propagated error energy under the stated assumptions; it does not estimate the scalar coefficients of Proposition 4.1 or the full alignment deviation. Appendix H derives the criterion, including the omitted interactions with error outside the sampled subspaces.

4.4 Joint Rank Allocation

The component order tells us what to retain at any rank; we now decide how many components each matrix should receive. Under a shared budget, we compare the estimated benefit of the next component with its storage cost. That benefit depends on how much error is removed, where it propagates, and how it interacts with errors still present in other matrices. We therefore allocate ranks using both individual and pairwise error terms.

Reindex components by decreasing equation 13. Adding component r_k costs $c_k = m_k + n_k$ stored parameters. With total budget B_{rank} matched to IProX and per-matrix caps $\bar{r}_k$, the feasible ranks satisfy

$$\mathcal{R} = \left\{ \mathbf{r} \in \mathbb{Z}_{\geq 0}^L : \quad \sum_k c_k r_k \leq B_{\mathrm{rank}}, \quad r_k \leq \min\{\bar{r}_k, R_k\} \right\}. \tag{14}$$

Remaining error and propagation. The component costs are normalized within each matrix, so they cannot directly compare rank gains across matrices. Following the local-error and propagation structure of Proposition 4.1, define

$$\omega_{k,a} = \sigma_{k,a}^2 \big(u_{k,a}^{\mathrm{fwd}} + u_{k,a}^{\mathrm{bwd}}\big), \qquad \widetilde{Q}_k(r) = \sum_{a=r}^{R_k-1} \omega_{k,a}. \tag{15}$$

Adding the next component removes ω_{k,r_k} from the remaining-error measure $\widetilde{Q}_k$. We weight its square root by the matrix's mean propagation strength:

$$\widehat{a}_k = \frac{\overline{d_{k,\cdot}^{\mathrm{fwd}} + d_{k,\cdot}^{\mathrm{bwd}}}}{\mathrm{med}_j \; \overline{d_{j,\cdot}^{\mathrm{fwd}} + d_{j,\cdot}^{\mathrm{bwd}}}}, \qquad \Phi_1(\mathbf{r}) = \sum_k \widehat{a}_k \sqrt{\widetilde{Q}_k(r_k)}. \tag{16}$$

The overline averages components and med_j takes the median across matrices. The same error reduction is therefore more valuable at a matrix with greater propagation weight. These are empirical proxies: $\widehat{a}_k$ does not estimate the full A_k, D_k, and $\widetilde{Q}_k$ omits direction-norm factors and cross-terms (Appendix I).

Interactions between errors. The term $-E_k \Delta h_{k-1}$ in equation 6 shows why the additive objective is incomplete: an earlier error changes the input on which a later compression error acts. Substituting

the first-order paths yields products of errors, with analogous backward terms. Under (H1)–(H2), their contribution satisfies

$$\mathbb{E}_{z,z'}\left|\Delta I^{[2]}_{\mathcal{T},\mathrm{prop}}\right| \;\le\; \sum_{i<k} \Gamma_{ik}\,\|E_i\|_F \|E_k\|_F. \tag{17}$$

Here $\Gamma_{ik} \ge 0$ combines path gains and data amplitudes. This bounds propagated pair interactions, rather than all second-order score terms (Appendix E).

To include these interactions, let $x_k = \sqrt{\widetilde{Q}_k(r_k)}$ and use an empirical table $\widehat{\Gamma}$ measured by projection type and block distance. With $\overline{\Gamma} = (\widehat{\Gamma} + \widehat{\Gamma}^\top)/2$, we evaluate allocations by

$$\Phi_2^{\mathrm{eval}}(\mathbf{r}) = \sum_k \widehat{a}_k x_k + \frac{1}{2}\sum_{i\neq k} \overline{\Gamma}_{ik} x_i x_k. \tag{18}$$

With other errors fixed, reducing x_k by $\Delta x_k > 0$ decreases the objective by $\Delta x_k(\widehat{a}_k + \sum_{j\neq k} \overline{\Gamma}_{kj} x_j)$. Thus, rank can be more valuable where it also reduces interactions with large remaining errors elsewhere. This motivates updating priorities as ranks change. The objective follows the structure of the bounds, but its empirical quantities do not make it a bound on full alignment error.

Completing the proxy. We propose rank increments using a Taylor estimate of error reduction per storage cost, weighted by propagation and current interactions. Eight fixed-point passes generate feasible allocations; we retain the one with the lowest Φ_2^{eval}. These proposals are heuristic because the normalized interaction table can be asymmetric. Per-matrix caps keep factor storage within the specified compression limits. Appendices J–K give the cap, normalization, and update formulas. Finally, retain the first r_k components and assemble equation 5 using equation 10. Storing each $\widehat{W}_k$ as $m_k \times r_k$ and $r_k \times n_k$ factors completes the compressed proxy.

5 Experiments

We first describe the common experimental protocol, then compare LPEC with matched-budget proxies and baselines. We follow with analyses of probe construction, efficiency, and component ablations.

5.1 Experimental Setup

Datasets. We use three candidate training datasets: Dolly (Conover et al., 2023), Math (Hendrycks et al., 2021), and Biomed (Tran et al., 2024). Following (Xia et al., 2024), we evaluate on MMLU (Hendrycks et al., 2020), BBH (Suzgun et al., 2022), and TyDiQA (Clark et al., 2020) and report the three scores and their average. Table 6 in Appendix A.2 summarizes the datasets and the validation sets.

Baselines and Models. The target models are Qwen2.5-7B (Team, 2024), Qwen3-4B (Yang et al., 2025), and Llama3.2-3B (Dubey et al., 2024); the target-direct row is a reference that scores candidates with the full target model. We compare with a same-family off-the-shelf proxy, the Influence Scorer that trains a smaller model to predict influence (Gu et al., 2024; Yu et al., 2024), Layer Extraction (Men et al., 2024), IProX (Chen et al., 2026), and LPEC; more details are provided in Appendix A.1.

Implementation Details. We follow the training and data selection settings of (Chen et al., 2026): each target model is warmed up on a randomly selected 5% subset of $\mathcal{D}_{\mathrm{train}}$, with the embedding and LM head frozen and excluded from the influence calculation; candidates are then scored by their influence values and the top 5% is selected (scoring protocols are given in Appendix B), on which the target model is fully fine-tuned for four epochs. LPEC is built from the warmed-up target model using 1% of the data source. At each sparsity, IProX and LPEC use the same factor-storage budget, and we vary the selected-matrix sparsity $s \in \{0.3, 0.5, 0.7\}$. More details are provided in Appendix A.3.

Table 1: Main results. Each method selects from the three candidate datasets under the same budget. s is the sparsity applied to each selected weight matrix; Params is the approximate scoring-model size. The gray row is a target-direct reference. **Bold** and underline mark the best and second-best proxy-based results per column within each block.

Method	Params	Dolly				Math				Biomed			
		MMLU	BBH	TyDiQA	Avg.	MMLU	BBH	TyDiQA	Avg.	MMLU	BBH	TyDiQA	Avg.
Target model: Qwen2.5-7B													
Target model	7B	72.00	69.80	33.27	58.36	72.13	67.94	23.28	54.45	72.12	70.74	27.46	56.77
Off-the-shelf proxy	0.5B	72.03	69.68	31.73	57.81	72.03	**68.00**	18.59	52.87	72.22	**70.59**	26.41	56.41
Influence Scorer	0.5B	72.01	68.80	28.41	56.41	72.20	66.82	18.12	52.38	72.30	64.53	27.07	54.63
Layer Extraction	5.7B	72.67	68.85	25.84	55.79	72.24	65.90	18.36	52.17	72.42	62.11	21.09	51.87
IProX, s=0.3	5.7B	72.62	68.87	33.34	58.28	72.24	66.00	16.94	51.73	72.56	69.30	22.17	54.68
IProX, s=0.5	4.4B	72.68	68.78	30.95	57.47	72.20	65.45	17.36	51.67	72.55	68.88	27.29	56.24
IProX, s=0.7	3.0B	72.53	68.72	24.11	55.12	72.11	66.82	18.09	52.34	72.50	68.42	24.00	54.97
LPEC (Ours), s=0.3	5.7B	72.48	**69.92**	30.81	57.74	**72.30**	67.33	**18.91**	52.85	72.23	65.85	27.47	55.18
LPEC (Ours), s=0.5	4.4B	**72.71**	68.93	**34.27**	**58.64**	72.24	67.02	18.33	52.53	72.38	68.25	28.26	56.30
LPEC (Ours), s=0.7	3.0B	72.58	69.78	29.13	57.16	72.10	67.71	18.87	**52.89**	**72.65**	67.41	**29.64**	**56.57**
Target model: Qwen3-4B													
Target model	4B	68.79	73.38	49.86	64.01	68.29	71.40	42.13	60.60	68.84	73.56	47.06	63.16
Off-the-shelf proxy	1.7B	68.78	69.45	**51.80**	**63.34**	68.37	71.06	42.95	60.79	69.06	**73.46**	47.51	63.34
Influence Scorer	1.7B	68.59	69.32	48.24	62.05	68.38	71.35	42.35	60.69	68.96	69.90	47.15	62.00
Layer Extraction	2.9B	68.72	70.40	49.70	62.94	68.52	70.89	42.41	60.61	68.67	72.65	47.61	62.98
IProX, s=0.3	2.9B	68.66	70.48	44.18	61.11	68.34	70.00	42.81	60.39	68.79	71.88	47.86	62.84
IProX, s=0.5	2.2B	**68.95**	70.77	44.55	61.42	68.36	70.53	43.12	60.67	68.88	71.19	47.66	62.58
IProX, s=0.7	1.5B	68.78	**71.42**	48.53	62.91	68.42	70.86	43.48	60.92	68.88	70.25	48.24	62.46
LPEC (Ours), s=0.3	2.9B	68.41	70.83	46.58	61.94	**68.59**	71.32	43.05	60.99	68.92	72.04	47.95	62.97
LPEC (Ours), s=0.5	2.2B	68.56	71.19	46.98	62.24	68.48	71.87	**43.65**	**61.33**	**69.09**	72.08	48.53	63.23
LPEC (Ours), s=0.7	1.5B	68.72	71.15	47.12	62.33	68.49	**71.96**	43.54	**61.33**	69.02	73.03	**48.58**	**63.54**
Target model: Llama3.2-3B													
Target model	3B	53.34	46.41	30.36	43.37	51.92	45.10	27.54	41.52	53.12	42.29	36.00	43.80
Off-the-shelf proxy	1B	51.62	44.83	29.65	42.03	**52.52**	46.40	28.68	42.53	53.47	42.24	35.93	43.88
Influence Scorer	1B	53.56	41.26	**33.15**	42.66	51.32	46.12	29.61	42.35	53.39	44.20	32.23	43.27
Layer Extraction	2.4B	53.23	43.90	31.06	42.73	51.77	47.16	27.53	42.15	53.19	43.90	32.03	43.04
IProX, s=0.3	2.4B	53.75	45.02	16.87	38.55	51.78	46.29	29.41	42.49	53.94	44.62	32.29	43.62
IProX, s=0.5	1.8B	53.50	44.33	20.03	39.29	51.66	46.16	29.20	42.34	**54.02**	**45.26**	30.66	43.31
IProX, s=0.7	1.2B	53.77	43.95	16.09	37.94	51.68	46.84	29.50	42.67	53.82	44.53	27.95	42.10
LPEC (Ours), s=0.3	2.4B	**53.97**	45.09	28.77	42.61	51.13	46.83	30.01	42.66	53.67	43.18	34.51	43.79
LPEC (Ours), s=0.5	1.8B	53.66	45.25	26.81	41.91	51.57	**47.30**	**31.18**	**43.35**	53.24	43.93	**39.62**	**45.60**
LPEC (Ours), s=0.7	1.2B	53.72	**45.53**	31.88	**43.71**	51.40	46.59	30.51	42.83	53.45	44.00	39.12	45.52

5.2 Main Results

Table 1 reports the main comparison under matched selection and storage budgets.

Strong overall selection. Under matched selection and budgets, LPEC delivers strong average performance across the candidate datasets. Its results remain competitive across target models and evaluation tasks, rather than relying on a particular model or dataset. The task-level scores are broadly balanced across the evaluated settings.

Robust to tight budgets. At s=0.7, where the proxy retains roughly one third of the compressible parameters, LPEC outperforms rate-matched IProX in eight of nine settings and can even outperform its own lower-sparsity variants. The Params column shows that the preferred sparsity varies with the target model, candidate dataset, and task.

Table 2: Extreme compression rates on Math. **Bold** marks the better method per column.

s	Method	Params	MMLU	BBH	TyDiQA	Avg.
Qwen3-4B						
0.9	IProX	0.8B	68.52	71.35	43.38	61.08
	LPEC	0.8B	**68.65**	**71.50**	**43.50**	**61.22**
0.99	IProX	0.4B	68.50	71.30	42.25	60.68
	LPEC	0.4B	**68.58**	**72.42**	**43.12**	**61.37**
Llama3.2-3B						
0.9	IProX	0.7B	51.09	46.13	28.92	42.04
	LPEC	0.7B	**51.31**	**47.31**	**30.71**	**43.11**
0.99	IProX	0.4B	**51.96**	46.12	**30.34**	42.80
	LPEC	0.4B	51.66	**47.35**	30.30	**43.11**

Proxy-selected data can approach target-selected performance. Compared with the gray reference rows, where the target model selects data for itself, LPEC approaches the reference performance on all nine combinations and exceeds it on six, although its scoring proxy is only a fraction of the target's size. This echoes prior observations that smaller models

can identify more generalizable training data (Xie et al., 2023; Engstrom, 2024; Xia et al., 2024; Wang et al., 2023a).

Extreme compression. Table 2 further compares with rate-matched IProX on Math at s=0.9 and s=0.99. Even under such extreme compression, LPEC preserves its selection quality and can even achieve better results than at lower sparsity. On Qwen3-4B, the smallest proxy attains its best Math average among all sparsity levels while retaining only about 0.4B parameters, roughly a tenth of the target model. This further demonstrates that extremely compressed models can serve as scoring proxies, improving the efficiency of gradient-based data selection.

5.3 Analysis

We analyze probe construction, efficiency, and component ablations; Appendix C reports a scoring-aggregation control, cap sensitivity, and a realized allocation.

Probe construction. The probe diagnostics use Llama3.2-3B on Dolly at s=0.5; each setting rebuilds the proxy, reselects data, and keeps the downstream protocol fixed. In the top of Table 3, the budget response is non-monotonic: the 3× setting has the highest observed average, whereas halving or quintupling the budget does not improve on the default. In the bottom, replacing part of the probe set with SMOTE interpolations lowers the average at all three tested fractions, with 20% replacement the strongest among them. The construction sequences drawn from the candidate dataset therefore already provide sufficient diversity for proxy construction.

Table 3: Probe diagnostics on Llama3.2-3B (Dolly, s=0.5). Top: probe budget; bottom: probe diversity.

Setting	MMLU	BBH	TyDiQA	Avg.
Probe budget				
0.5×	52.99	45.25	23.68	40.64
1×	53.66	45.25	26.81	41.91
3×	53.06	45.63	30.39	43.02
5×	53.26	45.02	26.54	41.61
Probe diversity				
0%	53.66	45.25	26.81	41.91
10%	52.61	44.90	23.23	40.25
20%	52.81	44.63	26.16	41.20
30%	53.54	43.70	23.98	40.41

Efficiency. Table 4 reports the wall-clock cost of proxy construction and influence scoring, measured on a single PPU-ZW810 GPU. Both compressed scorers price influence at about 75 minutes, roughly a quarter below the dense target, in exchange for about 19–25 minutes of one-off construction. LPEC trades roughly six additional construction minutes for propagation and interaction statistics for improved selection quality, while keeping scoring time comparable to IProX through an equally sized proxy. Once constructed, the proxy can be reused for repeated candidate scoring, avoiding repeated target-model gradients.

Ablation Studies. In Table 5, removing propagation-aware direction ranking, removing the weighted SVD, replacing the interaction-aware proposal with the first-order one, or removing joint rank allocation each lowers the average score at the same storage budget. The largest loss follows removing propagation-aware direction ranking. Together, the ablations show that propagation-aware component selection and joint rank coordination are both needed for the full method.

Table 4: Wall-clock time on Llama3.2-3B (Dolly$s$=0.5): proxy construction and influence scoring (Infl. Calc.).

Model	Compress	Infl. Calc.
Llama3.2-3B	–	∼100 mins
IProX (1.8B)	∼19 mins	∼75 mins
LPEC (1.8B)	∼25 mins	∼76 mins

Table 5: Component ablation on Llama3.2-3B (Math, s=0.5); Δ is the average-score change from the full method.

Configuration	MMLU	BBH	TyDiQA	Avg.	Δ
Full method	51.57	47.30	31.18	43.35	–
w/o propagation-aware direction ranking	51.45	47.17	28.82	42.48	−0.87
w/o weighted SVD	51.86	46.90	28.82	42.53	−0.82
first-order proposal	51.40	46.52	30.11	42.68	−0.67
w/o joint rank allocation	51.54	47.09	30.38	43.00	−0.35

6 Conclusion

We introduced LPEC, a low-rank proxy construction for gradient-based data selection that distinguishes local reconstruction error from the propagated effects of weight-matrix compression across layers. A conditional first-order analysis motivates a weighted basis and propagation-aware direction ranking, while propagated pair interactions motivate joint rank allocation under a shared storage budget and per-matrix caps.

AI USE STATEMENT

We used generative AI tools to assist with language editing, code review, and checking mathematical derivations. We did not use generative AI to generate, modify, or augment experimental data or results. All AI-assisted text, code, and derivations were reviewed and verified by the authors. We take full responsibility for the final content, including the methods, experiments, code, and claims.

REPRODUCIBILITY STATEMENT

We provide the complete method specification, including the weighted basis construction, propagation estimators, rank-allocation objective, and scoring protocol, in Sections 4 and B. The experimental setup, datasets, model configurations, training hyperparameters, and implementation details are specified in Sections 5 and A. The appendices provide proofs of the theoretical results, derivations of the optimization components, and additional experimental results. We will release the code, configurations, and processed data-selection outputs upon publication.

REFERENCES

Juhan Bae, Wu Lin, Jonathan Lorraine, and Roger B Grosse. Training data attribution via approximate unrolling. *Advances in Neural Information Processing Systems*, 37:66647–66686, 2024.

Sirui Chen, Yunzhe Qi, Mengting Ai, Yifan Sun, Ruizhong Qiu, Jiaru Zou, and Jingrui He. Influence-preserving proxies for gradient-based data selection in LLM fine-tuning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=PDNpRLxDlI.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instructiontuned llm. 2023.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Logan Engstrom. Dsdm: Model-aware dataset selection with datamodels. In *Forty-first International Conference on Machine Learning*, 2024.

Prakhar Ganesh, Yao Chen, Xin Lou, Mohammad Ali Khan, Yin Yang, Hassan Sajjad, Preslav Nakov, Deming Chen, and Marianne Winslett. Compressing large-scale transformer-based models: A case study on bert. *Transactions of the Association for Computational Linguistics*, 9: 1061–1080, 2021. ISSN 2307-387X. doi: 10.1162/tacl_a_00413. URL http://dx.doi.org/10.1162/tacl_a_00413.

Roger Grosse, Juhan Bae, Cem Anil, Nelson Elhage, Alex Tamkin, Amirhossein Tajdini, Benoit Steiner, Dustin Li, Esin Durmus, Ethan Perez, et al. Studying large language model generalization with influence functions. *arXiv preprint arXiv:2308.03296*, 2023.

Yuxian Gu, Li Dong, Hongning Wang, Yaru Hao, Qingxiu Dong, Furu Wei, and Minlie Huang. Data selection via optimal control for language models. *arXiv preprint arXiv:2410.07064*, 2024.

Xiaochuang Han, Daniel Simig, Todor Mihaylov, Yulia Tsvetkov, Asli Celikyilmaz, and Tianlu Wang. Understanding in-context learning via supportive pretraining data. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12660–12673, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization, 2022. URL https://arxiv.org/abs/2207.00112.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Xing Hu, Dawei Yang, Yuan Cheng, Zhixuan Chen, and Zukang Xu. Saes-svd: Self-adaptive suppression of accumulated and local errors for svd-based llm compression, 2026. URL https://arxiv.org/abs/2602.03051.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International conference on machine learning*, pp. 1885–1894. PMLR, 2017.

Yongchan Kwon, Eric Wu, Kevin Wu, and James Zou. Datainf: Efficiently estimating data influence in lora-tuned llms and diffusion models. In *The Twelfth International Conference on Learning Representations*, 2024.

Haowei Lin, Yacine Jernite, HT Kung, and Andrew Gordon Wilson. Evosld: Automated neural scaling law discovery with large language models. *Preprint*, 2025a.

Xiaoqiang Lin, Xinyi Xu, See-Kiong Ng, and Bryan Kian Hsiang Low. Efficient top-m data values identification for data selection. In *The Thirteenth International Conference on Learning Representations*, 2025b.

Zifan Liu, Amin Karbasi, and Theodoros Rekatsinas. Tsds: Data selection for task-specific model finetuning. *arXiv preprint arXiv:2410.11303*, 2024.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. Shortgpt: Layers in large language models are more redundant than you expect, 2024. URL https://arxiv.org/abs/2403.03853.

Garima Pruthi, Frederick Liu, Satyen Kale, and Mukund Sundararajan. Estimating training data influence by tracing gradient descent. *Advances in Neural Information Processing Systems*, 33: 19920–19930, 2020.

Rajarshi Saha, Naomi Sagan, Varun Srivastava, Andrea J. Goldsmith, and Mert Pilanci. Compressing large language models using low rank and low precision decomposition, 2024. URL https://arxiv.org/abs/2405.18886.

Kashun Shum, Yuzhen Huang, Hongjian Zou, Qi Ding, Yixuan Liao, Xiaoxin Chen, Qian Liu, and Junxian He. Predictive data selection: The data that predicts is the data that teaches. *arXiv preprint arXiv:2503.00808*, 2025.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V Le, Ed H Chi, Denny Zhou, et al. Challenging big-bench tasks and whether chain-of-thought can solve them. *arXiv preprint arXiv:2210.09261*, 2022.

Qwen Team. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Hieu Tran, Zhichao Yang, Zonghai Yao, and Hong Yu. Bioinstruct: instruction tuning of large language models for biomedical natural language processing. *Journal of the American Medical Informatics Association*, 31(9):1821–1832, 2024.

Andrew Wang, Elisa Nguyen, Runshi Yang, Juhan Bae, Sheila A McIlraith, and Roger Grosse. Better training data attribution via better inverse hessian-vector products. *arXiv preprint arXiv:2507.14740*, 2025a.

Renxi Wang, Haonan Li, Minghao Wu, Yuxia Wang, Xudong Han, Chiyu Zhang, and Timothy Baldwin. Demystifying instruction mixing for fine-tuning large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 4: Student Research Workshop)*, pp. 68–75, 2024.

Xin Wang, Samiul Alam, Zhongwei Wan, Hui Shen, and Mi Zhang. Svd-llm v2: Optimizing singular value truncation for large language model compression, 2025b. URL https://arxiv.org/abs/2503.12340.

Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression, 2025c. URL https://arxiv.org/abs/2403.07378.

Xinyi Wang, Wanrong Zhu, Michael Saxon, Mark Steyvers, and William Yang Wang. Large language models are latent variable models: Explaining and finding good demonstrations for incontext learning. *Advances in Neural Information Processing Systems*, 36:15614–15638, 2023a.

Yizhong Wang, Hamish Ivison, Pradeep Dasigi, Jack Hessel, Tushar Khot, Khyathi Chandu, David Wadden, Kelsey MacMillan, Noah A Smith, Iz Beltagy, et al. How far can camels go? exploring the state of instruction tuning on open resources. *Advances in Neural Information Processing Systems*, 36:74764–74786, 2023b.

Mengzhou Xia, Sadhika Malladi, Suchin Gururangan, Sanjeev Arora, and Danqi Chen. Less: selecting influential data for targeted instruction tuning. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 54104–54132, 2024.

Sang Michael Xie, Hieu Pham, Xuanyi Dong, Nan Du, Hanxiao Liu, Yifeng Lu, Percy S Liang, Quoc V Le, Tengyu Ma, and Adams Wei Yu. Doremi: Optimizing data mixtures speeds up language model pretraining. *Advances in Neural Information Processing Systems*, 36:69798–69818, 2023.

Tong Xie, Haoyu Li, Andrew Bai, and Cho-Jui Hsieh. Data attribution for diffusion models: Timestep-induced bias in influence estimation. *arXiv preprint arXiv:2401.09031*, 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.

Yu Yang, Siddhartha Mishra, Jeffrey N Chiang, and Baharan Mirzasoleiman. Smalltolarge (s2l): Scalable data selection for fine-tuning large language models by summarizing training trajectories of small models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=K9IGlMQpif.

Zichun Yu, Spandan Das, and Chenyan Xiong. MATES: Model-aware data selection for efficient pretraining with data influence models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=6gzPSMUAz2.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. Asvd: Activation-aware singular value decomposition for compressing large language models, 2025. URL https://arxiv.org/abs/2312.05821.

Chi Zhang, Huaping Zhong, Kuan Zhang, Chengliang Chai, Rui Wang, Xinlin Zhuang, Tianyi Bai, Jiantao Qiu, Lei Cao, Ju Fan, et al. Harnessing diversity for important data selection in pretraining large language models. *arXiv preprint arXiv:2409.16986*, 2024.

Jiaru Zou, Yikun Ban, Zihao Li, Yunzhe Qi, Ruizhong Qiu, Ling Yang, and Jingrui He. Transformer copilot: Learning from the mistake log in llm fine-tuning. *arXiv preprint arXiv:2505.16270*, 2025.

Appendix

Appendix Contents

A Experimental Details **15**
A.1 Baselines 15
A.2 Datasets and Evaluation 15
A.3 Implementation Details 16

B Scoring Criteria **17**

C Additional Experiment Results **17**

D Proof of Proposition 4.1 **18**
D.1 A Two-Sided Weighted Upper Bound 21

E Pairwise Interaction from Propagated Errors **22**

F Construction of the Regularized Basis **24**

G Unbiasedness of the Random-Sign Path Estimator **25**

H From Propagated Error to Component Selection **26**

I The Remaining Truncation-Error Proxy $\widetilde{Q}_k$ **28**

J Accuracy of the Taylor Approximation **28**

K Per-Matrix Allocation under Dense Interactions **29**

A EXPERIMENTAL DETAILS

This appendix documents the experimental setup for all main-text results: the baselines, the data and evaluation protocol, and the implementation of LPEC.

A.1 BASELINES

Here we provide implementation details of the baselines. Our baseline configuration follows (Chen et al., 2026). The joint protocol of equation 19 is the conventional aggregation for gradient-based selection, so every baseline follows it. LPEC instead uses the block-balanced protocol of equation 20, which we introduce for its non-uniformly rank-reallocated proxy; each protocol is fixed across datasets and sparsities.

- **IProX:** The official implementation of (Chen et al., 2026), run at sparsity $s \in \{0.3, 0.5, 0.7\}$. It builds the layer-local whitened basis and scores candidates with the resulting proxy.
- **Layer Extraction (LE):** We extract layers from the warmed-up target models. Following (Men et al., 2024), each block (attention plus MLP) is scored with an influence
$$I_{\mathrm{LE}} = 1 - \mathbb{E}_x \frac{h_i^\top h_{i+1}}{\|h_i\| \, \|h_{i+1}\|},$$
where h_i and h_{i+1} denote the hidden states before and after the i-th block; larger values indicate that the block changes the representation more. For a fair comparison, the sparsity is set to s=0.3 and the top 70% of blocks ranked by I_{LE} are kept. The influence is computed on a 1% random sample of $\mathcal{D}_{\mathrm{train}}$.
- **Influence Scorer (IS):** Prior work (Gu et al., 2024; Yu et al., 2024) formulates this task as a regression problem, where a smaller model is trained to predict influence scores from a limited set of annotated data. Concretely, the target model first computes influence on a hold-out set, and these values supervise the smaller model. Once trained, the smaller model generates influence scores for the entire candidate dataset. This approach still requires influence computation with the target model to produce annotations, and its generalizability can be affected when data preferences shift during training. In our implementation, we adopt the off-the-shelf model of Table 1 as the backbone, attach a regression head, and freeze all other layers during training. For a fair comparison, we use 1% of $\mathcal{D}_{\mathrm{train}}$ as the hold-out set and train only once. We optimize with Adam for five epochs at learning rate $1\mathrm{e}{-5}$ and weight decay $1\mathrm{e}{-2}$.
- **Off-the-shelf proxy (direct):** A smaller pretrained model from the same family scores every candidate directly by its own gradient alignment with the validation set.
- **Target model (direct):** The target model scores candidates for itself. This row is a reference only; it bypasses the proxy bottleneck that the paper studies.

A.2 DATASETS AND EVALUATION

Target models. Qwen2.5-7B, Qwen3-4B, and Llama3.2-3B.

Candidate datasets. Three instruction-tuning datasets, Dolly, Math, and Biomed, are disjoint from the evaluation datasets. Table 6 lists candidate-pool sizes and the evaluation splits.

Construction probes. For each candidate dataset, 14 probe sequences are drawn disjointly from the scored candidates, yielding a budget of 2048 accumulated directions per selected matrix ($N \approx 2044$ realized).

Selection budget. Each method selects the top 5% of its dataset under the same budget.

Downstream protocol. We follow (Xia et al., 2024) to evaluate the target tasks. MMLU is 5-shot accuracy on the test set averaged across 57 subtasks. TyDiQA is 1-shot macro-averaged exact match across 9 languages under the gold-passage setting, where a passage containing the reference answer is provided. BBH is the average 3-shot exact match across tasks. The target model is fine-tuned on each selected subset, and tables report final scores on a 0–100 scale with their average. Table 7 gives the training hyperparameters.

Table 6: Candidate datasets and evaluation splits. Candidate datasets supply the data ranked by each method, from which the top 5% is selected.

Dataset	Role	Task	Candidate pool	$\lvert\mathcal{D}_{\text{test}}\rvert$	$\lvert\mathcal{D}_{\text{val}}\rvert$	# Shots	Metric
Dolly	Candidate	General knowledge	15,011	–	–	–	–
Math	Candidate	Mathematics	12,500	–	–	–	–
Biomed	Candidate	Biomedicine	25,005	–	–	–	–
MMLU	Evaluation	Multiple choice	–	18,721	285	5	Accuracy
BBH	Evaluation	Reasoning	–	920	81	3	Accuracy
TyDiQA	Evaluation	Multilingual QA	–	1,713	9	1	Exact match

Table 7: Warm-up and downstream SFT hyperparameters.

Hyperparameter	Warm-up	Downstream SFT
Tuner type	Full	Full
Epochs	1	4
Learning rate	1e−5	1e−5
Weight decay	1e−2	1e−2
Optimizer	AdamW	AdamW
Scheduler	Cosine	Cosine
Warm-up ratio	0	0
Per-device batch size	4	4
Gradient accumulation	1	4
Effective batch per GPU	4	16
Max sequence length	2048	2048
Frozen modules	Embedding, LM head	Embedding, LM head

A.3 Implementation Details

Whitened basis. We collect input activations and loss-output gradients at shared probe-token positions in one forward/backward pass. The 14 probe sequences supply a budget of 2048 columns ($N{\approx}2044$ realized). For each selected matrix, we form the empirical input and gradient moments, compute economy SVDs, and floor singular values at 10^{-6}. With ridge $\lambda = 10^{-3}$, we factor the weighted core $D_\Delta(U_\Delta^\top W_k U_H)D_H$ and map the singular vectors back through D_Δ^{-1} and D_H^{-1}, as in Appendix F.

Propagation statistics. We estimate direction-specific propagation without forming path matrices. Forward strengths use reverse-mode VJPs with independent Rademacher signs at later layer-input boundaries, whereas backward strengths use forward-mode JVPs with signs at output boundaries. Each repetition covers all selected matrices, and squared projections are averaged over valid probe tokens. Both propagation and interaction stages use all 14 probes truncated to 256 tokens. Interactions are measured separately: for four depth-spread consumers per projection type, a VJP of the derived input loss gradient gives the directed squared response, which is aggregated by projection type and block distance. The reach variant scales each directed value by the consumer's mean forward strength.

Rank reallocation. We match IProX's factor-storage budget and impose the per-matrix caps of equation 41. Increments are proposed using the remaining propagated error and the directed interaction score of equation 43 with $\beta = 1$ and linear Taylor gains; eight fixed-point passes generate feasible candidates, and we retain the one with the lowest fixed-scale Φ_2^{eval}. Appendix K gives the normalization and update formulas. Unless otherwise noted, we sweep the cap parameter over $\tau \in \{0, 0.3, 0.5, 0.7, 1\}$ for each model–dataset–sparsity setting; Figure 3 shows the Math sweep.

Scoring. Following the conventional joint aggregation, all baselines score candidates with the proxy alone using equation 19. LPEC uses the block-balanced aggregation of Appendix B, introduced for its non-uniformly rank-reallocated proxy; the top 5% is selected in every case.

Table 8: Scoring-aggregation control on Dolly at selected-matrix sparsity s=0.7 (single run). The implementation's `per_layer` aggregation is denoted block-balanced. LPEC uses τ=0.2; Δ is block-balanced minus joint Avg.

Proxy	Joint				Block-balanced				Δ Avg.
	MMLU	BBH	TyDiQA	Avg.	MMLU	BBH	TyDiQA	Avg.	
Target model: Llama3.2-3B									
LPEC	52.44	45.24	30.15	42.61	52.21	45.26	33.00	43.49	+0.88
IProX	52.96	43.69	25.03	40.56	52.96	44.44	27.71	41.70	+1.14
Target model: Qwen3-4B									
LPEC	68.27	72.69	44.16	61.70	68.46	70.72	47.00	62.06	+0.36
IProX	68.67	71.65	45.59	61.97	68.37	70.64	45.38	61.46	−0.51

Hardware. Experiments used six PPU-ZW810 accelerators (96 GiB each). Each warm-up, proxy-construction, influence-scoring, and downstream SFT job used one accelerator; independent jobs were scheduled in parallel across available accelerators.

B Scoring Criteria

The deployed score is distinct from the analytical alignment equation 3. Let $\mathcal{P}$ be the parameter blocks exposed by the proxy scorer, let $g_p(z)$ be the candidate gradient for block p, and let v_p be the corresponding anchor gradient. The conventional gradient-selection protocol uses one cosine over their concatenation,

$$\text{score}^{\text{joint}}(z) = \frac{\sum_{p\in\mathcal{P}} \langle g_p(z), v_p\rangle}{\sqrt{\sum_{p\in\mathcal{P}} \|g_p(z)\|^2}\sqrt{\sum_{p\in\mathcal{P}} \|v_p\|^2} + \epsilon}. \tag{19}$$

For a low-rank proxy, $\mathcal{P}$ contains factor parameter blocks whose norms can differ after rank reallocation. To score these non-uniformly rank-reallocated proxies, we introduce a block-balanced protocol that normalizes each available parameter block before averaging:

$$\text{score}^{\text{block}}(z) = \frac{1}{\sum_{p\in\mathcal{P}} w_p} \sum_{p\in\mathcal{P}} w_p \frac{\langle g_p(z), v_p\rangle}{\|g_p(z)\| \, \|v_p\| + \epsilon}. \tag{20}$$

Here $\epsilon = 10^{-12}$ follows the implementation, and $w_p \geq 0$ with $\sum_p w_p > 0$; by default $w_p = 1$. This protocol is a deployment choice for the rank-reallocated proxy, not a consequence of the first-order perturbation result. All scoring comparisons therefore report the aggregation protocol explicitly.

C Additional Experiment Results

Scoring-aggregation control. Table 8 compares the joint cosine with the implementation's `per_layer` aggregation, denoted block-balanced here, on Dolly at s=0.7. Block-balanced aggregation raises the average for LPEC on both target models. In contrast, its effect on the uniformly allocated IProX proxy is inconsistent across model families, improving the Llama3.2-3B result but reducing the Qwen3-4B result.

Cap sensitivity. Figure 3 sweeps the cap parameter τ (Appendix K): a larger τ raises the per-matrix rank cap $\bar{r}_k$, letting the joint allocation deviate further from the uniform storage ratio. On Llama3.2-3B, the average peaks at τ=0.3; on Qwen3-4B, it peaks at τ=1. MMLU remains comparatively flat, while BBH and TyDiQA trade off. The response is mild: every tested cap stays within about one point of its family's best average, and a moderate cap (τ=0.5) comes within half a point in both families. LPEC therefore tolerates a wide range of caps, even though the peak location differs across model families.

A realized joint allocation. Figure 4 shows one Qwen3-4B/Math/s=0.3 allocation. Relative to the uniform baseline, value, key, and output projections receive 83%, 68%, and 41% more rank on average, while gate projections receive 47% less. The allocation also varies across Transformer layers rather than following one static sublayer ratio. This realization shows that joint rank allocation produces non-uniform matrix budgets; within each matrix, the retained directions are the top-ranked prefix under the propagation-aware key, not a raw singular-value prefix.

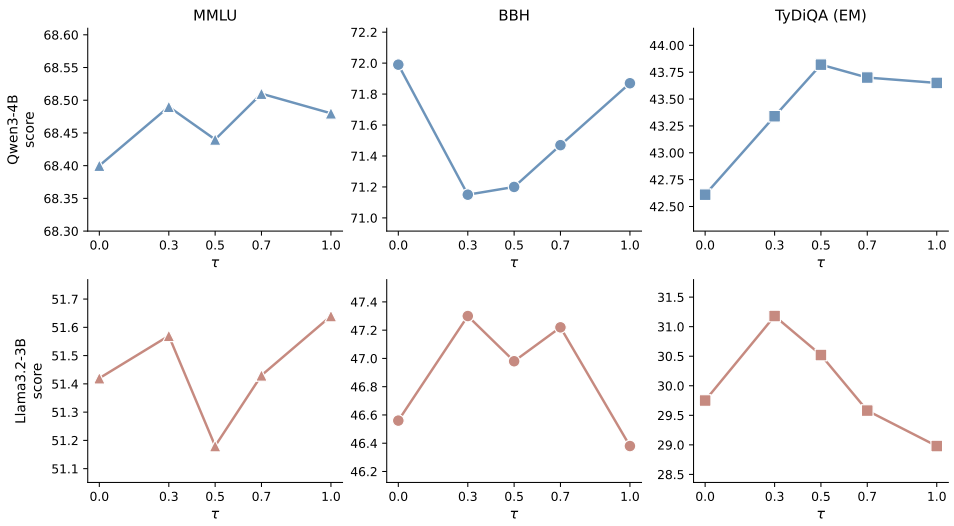

Figure 3: Cap-sensitivity sweep on Math at s=0.5; columns: tasks, rows: Qwen3-4B, Llama3.2-3B; triangles: tested $\tau \in \{0, 0.3, 0.5, 0.7, 1\}$.

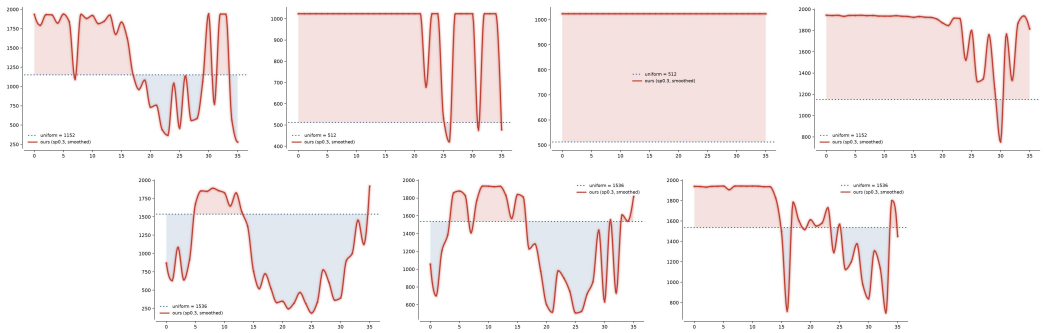

Figure 4: Realized joint rank allocation per sublayer type on Qwen3-4B (Math dataset, selected-matrix sparsity s=0.3), shown per Transformer layer against the uniform baseline allocation (dashed). Red and blue shading mark allocations above and below uniform, respectively.

D Proof of Proposition 4.1

This appendix proves the conditional expected first-order statement in Proposition 4.1. It concerns gradient alignment evaluated over the selected matrices and the simplified sequential model of §4.1; it does not establish a bound for the full Transformer graph or the deployed factor-gradient cosine score.

Analytical compression goal. Let $\widehat{I}_{\mathcal{T}}$ be the alignment in equation 3 evaluated in the compressed model, using gradients in the original matrix coordinates. With $c_k = m_k + n_k$, total storage budget B_{rank}, and per-matrix rank caps $\bar{r}_k$, the design goal is

$$\begin{aligned} \min_{\{\widehat{W}_k, r_k\}} \quad & \mathbb{E}_{z,z'} \left| \widehat{I}_{\mathcal{T}}(z, z') - I_{\mathcal{T}}(z, z') \right| \\ \text{subject to} \quad & \operatorname{rank}(\widehat{W}_k) \le r_k, \quad \sum_k c_k r_k \le B_{\text{rank}}, \quad r_k \in \{0, \dots, \bar{r}_k\}. \end{aligned} \tag{21}$$

The construction in §4 uses tractable approximations to this goal. The deployed factor-gradient score is specified separately in Appendix B.

Assumptions. The formal result uses the following conditions.

(H1) *(Simplified sequential model).* The selected matrices occur in a sequential composition $h_k = \phi_k(W_k h_{k-1})$. Along the perturbation, each layer derivative $J_k = D\phi_k(u_k)$ and the final loss gradient are fixed, $\Delta\delta_L^{[1]} = 0$.

(H2) *(Finite propagation gains).* All products of uncompressed amplitudes and products of layer-derivative norms that define A_k, D_k, and $\Gamma_{ik}^{\mathrm{fwd/bwd}}$ below are integrable under (z, z'); the first-order gain products have finite second moments. The candidate- and target-side moments are finite and defined by $C_{h,k} = \mathbb{E}_z[h_{k-1}h_{k-1}^\top]$, $C'_{h,k} = \mathbb{E}_{z'}[h'_{k-1}{h'_{k-1}}^\top]$, $C_{\Delta,k} = \mathbb{E}_z[\delta_k\delta_k^\top]$, and $C'_{\Delta,k} = \mathbb{E}_{z'}[\delta'_k{\delta'_k}^\top]$.

(H3a) *(Separable propagation and data-use statistics).* The propagation-error surrogate replaces joint propagation–activation and propagation–gradient moments by products of their separate moments.

(H3b) *(Diagonal approximation across components).* The direction-level score ignores cross-terms between distinct regularized low-rank directions.

Only (H1)–(H2) enter Proposition 4.1; (H3a)–(H3b) are the approximations used for component selection in §4.3 and Appendix H. The proof uses vector Cauchy–Schwarz and therefore does not require a directional-coherence assumption.

Proposition D.1 (Restatement of Proposition 4.1)**.** *Under (H1)–(H2),*

$$\mathbb{E}_{z,z'}\left|\Delta I_{\mathcal{T}}^{[1]}\right| \le \sum_{k=1}^{L}\left[A_k\sqrt{\mathrm{tr}(E_k^\top E_k \overline{C}_{h,k})} + D_k\sqrt{\mathrm{tr}(E_k E_k^\top \overline{C}_{\Delta,k})}\right].$$

Proof. **Step 1 (gradient-alignment decomposition).** For each selected matrix, $\nabla_{W_\ell}\mathcal{L}(z) = \delta_\ell h_{\ell-1}^\top$. Hence

$$\begin{aligned}\left\langle \delta_\ell h_{\ell-1}^\top, \delta'_\ell {h'_{\ell-1}}^\top\right\rangle_F &= \mathrm{tr}\left[(\delta_\ell h_{\ell-1}^\top)^\top(\delta'_\ell {h'_{\ell-1}}^\top)\right]\\ &= \mathrm{tr}\left[h_{\ell-1}\delta_\ell^\top\delta'_\ell {h'_{\ell-1}}^\top\right]\\ &= (\delta_\ell^\top\delta'_\ell)\,\mathrm{tr}\left[h_{\ell-1}{h'_{\ell-1}}^\top\right]\\ &= \langle\delta_\ell,\delta'_\ell\rangle\langle h_{\ell-1}, h'_{\ell-1}\rangle.\end{aligned}$$

Summing exactly over $\mathcal{T}$ gives

$$I_{\mathcal{T}}(z,z') = \sum_{\ell=1}^{L}\langle\delta_\ell,\delta'_\ell\rangle\langle h_{\ell-1}, h'_{\ell-1}\rangle. \tag{22}$$

Step 2 (extract the coefficient linear in the errors). Write $\widehat{\delta}_\ell = \delta_\ell + \Delta\delta_\ell$ and $\widehat{h}_{\ell-1} = h_{\ell-1} + \Delta h_{\ell-1}$. The exact product at layer ℓ is

$$\begin{aligned}\langle\widehat{\delta}_\ell,\widehat{\delta}'_\ell\rangle\langle\widehat{h}_{\ell-1},\widehat{h}'_{\ell-1}\rangle = &\langle\delta_\ell,\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle\\ &+\langle\Delta\delta_\ell,\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle + \langle\delta_\ell,\Delta\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle\\ &+\langle\delta_\ell,\delta'_\ell\rangle\langle\Delta h_{\ell-1},h'_{\ell-1}\rangle + \langle\delta_\ell,\delta'_\ell\rangle\langle h_{\ell-1},\Delta h'_{\ell-1}\rangle + O(\|E\|^2).\end{aligned}$$

The first term cancels in the score difference. Taking the coefficient linear in E and grouping the primed terms as the swapped copy gives

$$\Delta I_{\mathcal{T}}^{[1]} = \sum_{\ell=1}^{L}\left[\langle\Delta\delta_\ell^{[1]},\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle + \langle\delta_\ell,\delta'_\ell\rangle\langle\Delta h_{\ell-1}^{[1]},h'_{\ell-1}\rangle\right] + \mathrm{sym}(z\leftrightarrow z'). \tag{23}$$

Here $\mathrm{sym}(z\leftrightarrow z')$ means addition of the preceding expression with z and z' exchanged. The terms not displayed here are not included in $\Delta I_{\mathcal{T}}^{[1]}$.

Step 3 (linear recursions). Write $\Delta h_k = \widehat{h}_k - h_k$ and $\Delta u_k = \widehat{u}_k - u_k$. With $u_k = W_k h_{k-1}$ and $E_k = W_k - \widehat{W}_k$, the exact pre-activation recursion is

$$\Delta u_k = W_k\Delta h_{k-1} - E_k h_{k-1} - E_k\Delta h_{k-1}. \tag{24}$$

Retain its linear terms and apply the fixed Jacobian in (H1):

$$\begin{aligned}\Delta h_k^{[1]} &= J_k\big(W_k\Delta h_{k-1}^{[1]} - E_k h_{k-1}\big)\\ &= P_k\Delta h_{k-1}^{[1]} - J_k\varepsilon_k, \qquad P_k = J_k W_k, \quad \varepsilon_k = E_k h_{k-1}.\end{aligned}$$

The adjoint recursion is

$$\begin{aligned}\Delta\delta_{k-1}^{[1]} &= J_{k-1}^\top W_k^\top \Delta\delta_k^{[1]} - J_{k-1}^\top E_k^\top \delta_k\\ &= G_k^{\mathrm{bwd}}\Delta\delta_k^{[1]} - J_{k-1}^\top\eta_k, \qquad G_k^{\mathrm{bwd}} = J_{k-1}^\top W_k^\top\ (2\le k\le L), \quad \eta_k = E_k^\top\delta_k,\end{aligned}$$

with $\Delta h_0^{[1]} = 0$ and $\Delta\delta_L^{[1]} = 0$.

Step 4 (expand the propagation paths). Repeated substitution, starting from the fixed input and final loss gradient, gives

$$\begin{aligned}\Delta h_{\ell-1}^{[1]} &= -\sum_{j<\ell}\mathcal{F}_{\ell-1\leftarrow j}\varepsilon_j, \quad \mathcal{F}_{\ell-1\leftarrow j} = P_{\ell-1}\cdots P_{j+1}J_j,\\ \Delta\delta_\ell^{[1]} &= -\sum_{j>\ell}\mathcal{B}_{\ell\leftarrow j}\eta_j, \qquad \mathcal{B}_{\ell\leftarrow j} = G_{\ell+1}^{\mathrm{bwd}}\cdots G_{j-1}^{\mathrm{bwd}}J_{j-1}^\top.\end{aligned} \tag{25}$$

An empty product is the identity. Thus $\mathcal{F}_{\ell-1\leftarrow j}$ carries the local forward error $\varepsilon_j = E_j h_{j-1}$ to a later matrix's input, and $\mathcal{B}_{\ell\leftarrow j}$ carries the local backward error $\eta_j = E_j^\top\delta_j$ to an earlier matrix's loss gradient.

Step 5 (regroup by source layer). Substitute these chains into equation 23. For the forward channel,

$$\begin{aligned}\sum_\ell\langle\delta_\ell,\delta'_\ell\rangle\langle\Delta h_{\ell-1}^{[1]},h'_{\ell-1}\rangle &= -\sum_\ell\sum_{j<\ell}\langle\delta_\ell,\delta'_\ell\rangle\langle\mathcal{F}_{\ell-1\leftarrow j}\varepsilon_j,h'_{\ell-1}\rangle\\ &= -\sum_j\sum_{\ell>j}\langle\varepsilon_j,\mathcal{F}_{\ell-1\leftarrow j}^\top h'_{\ell-1}\rangle\langle\delta_\ell,\delta'_\ell\rangle.\end{aligned}$$

For the backward channel, substitute the second line of equation 25 and transpose the inner product:

$$\begin{aligned}\sum_\ell\langle\Delta\delta_\ell^{[1]},\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle &= -\sum_\ell\sum_{j>\ell}\langle\mathcal{B}_{\ell\leftarrow j}\eta_j,\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle\\ &= -\sum_j\sum_{\ell<j}\langle\eta_j,\mathcal{B}_{\ell\leftarrow j}^\top\delta'_\ell\rangle\langle h_{\ell-1},h'_{\ell-1}\rangle.\end{aligned}$$

Step 6 (pointwise Cauchy–Schwarz bound). Define

$$\begin{aligned}a_j(z,z') &= \sum_{\ell>j}\|\mathcal{F}_{\ell-1\leftarrow j}\|_{\mathrm{op}}\|h'_{\ell-1}\|\,\|\delta_\ell\|\,\|\delta'_\ell\|,\\ b_j(z,z') &= \sum_{\ell<j}\|\mathcal{B}_{\ell\leftarrow j}\|_{\mathrm{op}}\|\delta'_\ell\|\,\|h_{\ell-1}\|\,\|h'_{\ell-1}\|.\end{aligned}$$

For example,

$$\begin{aligned}\big|\langle\varepsilon_j,\mathcal{F}_{\ell-1\leftarrow j}^\top h'_{\ell-1}\rangle\langle\delta_\ell,\delta'_\ell\rangle\big| &\le \|\varepsilon_j\|\,\|\mathcal{F}_{\ell-1\leftarrow j}^\top h'_{\ell-1}\|\,|\langle\delta_\ell,\delta'_\ell\rangle|\\ &\le \|\varepsilon_j\|\,\|\mathcal{F}_{\ell-1\leftarrow j}\|_{\mathrm{op}}\|h'_{\ell-1}\|\,\|\delta_\ell\|\,\|\delta'_\ell\|.\end{aligned}$$

Applying this to every term in the two finite sums and using the triangle inequality gives

$$|\Delta I_{\mathcal{T}}^{[1]}| \le \sum_j\left[a_j(z,z')\|\varepsilon_j\| + b_j(z,z')\|\eta_j\|\right] + \left|\mathrm{sym}(z\leftrightarrow z')\right|.$$

Step 7 (take expectations). By Cauchy–Schwarz over (z,z'),

$$\begin{aligned}\mathbb{E}[a_j\|\varepsilon_j\|] &\le \sqrt{\mathbb{E}[a_j^2]}\sqrt{\mathbb{E}\|\varepsilon_j\|^2},\\ \mathbb{E}[b_j\|\eta_j\|] &\le \sqrt{\mathbb{E}[b_j^2]}\sqrt{\mathbb{E}\|\eta_j\|^2}.\end{aligned}$$

For the forward error, the required second moment expands as

$$\begin{aligned}\mathbb{E}\|\varepsilon_j\|^2 &= \mathbb{E}\big[h_{j-1}^\top E_j^\top E_j h_{j-1}\big] \\ &= \mathbb{E}\,\mathrm{tr}\big[E_j^\top E_j h_{j-1} h_{j-1}^\top\big] \\ &= \mathrm{tr}(E_j^\top E_j C_{h,j}).\end{aligned}$$

The swapped copy similarly gives

$$\mathbb{E}\|E_j h'_{j-1}\|^2 = \mathrm{tr}(E_j^\top E_j C'_{h,j}).$$

Let $x = \mathrm{tr}(E_j^\top E_j C_{h,j})$ and $x' = \mathrm{tr}(E_j^\top E_j C'_{h,j})$. If $\widetilde{A}_j$ and $\widetilde{A}'_j$ are the finite coefficients from the two copies, then

$$\begin{aligned}\widetilde{A}_j\sqrt{x} + \widetilde{A}'_j\sqrt{x'} &\le \max\{\widetilde{A}_j, \widetilde{A}'_j\}(\sqrt{x} + \sqrt{x'}) \\ &\le 2\max\{\widetilde{A}_j, \widetilde{A}'_j\}\sqrt{\frac{x+x'}{2}} \\ &= 2\max\{\widetilde{A}_j, \widetilde{A}'_j\}\sqrt{\mathrm{tr}(E_j^\top E_j \overline{C}_{h,j})}.\end{aligned}$$

For the backward error,

$$\begin{aligned}\mathbb{E}\|\eta_j\|^2 &= \mathbb{E}\big[\delta_j^\top E_j E_j^\top \delta_j\big] \\ &= \mathrm{tr}(E_j E_j^\top C_{\Delta,j}),\end{aligned}$$

and the swapped copy replaces $C_{\Delta,j}$ by $C'_{\Delta,j}$. Applying the same two-root bound to the backward channel yields the term with $\overline{C}_{\Delta,j}$. Absorbing the finite factors into A_j, D_j and summing over j proves equation 7.

Curvature, terminal-adjoint, graph-branch, and products of two perturbed score factors are excluded by the definition of $\Delta I^{[1]}_{\mathcal{T}}$. Bounding the full deviation would require additional assumptions. □

What the bound retains and discards. The trace terms retain the dependence of local errors on input and loss-gradient statistics. The operator-norm inequalities in Step 6, however, replace the action of each path on a particular error vector by a direction-independent upper bound. This explains the two uses of the analysis in the construction: the local norms support the reconstruction metric below, while the path expressions before this relaxation motivate evaluating individual component directions in §4.3. The latter construction returns to the propagation paths; it is not a componentwise estimate of the final scalar coefficients.

D.1 A Two-Sided Weighted Upper Bound

Proposition 4.1 controls the first-order alignment by a sum of one-sided weighted norms. The following consequence explains why a two-sided weighted reconstruction problem can provide a candidate low-rank basis. For $\lambda > 0$, define $C^\lambda_{h,k} = \overline{C}_{h,k} + \lambda I$ and $C^\lambda_{\Delta,k} = \overline{C}_{\Delta,k} + \lambda I$. Then

$$\begin{aligned}&A_k\|E_k\overline{C}_{h,k}^{1/2}\|_F + D_k\|\overline{C}_{\Delta,k}^{1/2}E_k\|_F \\ &\quad\le \frac{A_k + D_k}{\sqrt{\lambda}}\,\|(C^\lambda_{\Delta,k})^{1/2}E_k(C^\lambda_{h,k})^{1/2}\|_F.\end{aligned} \tag{26}$$

Corollary D.2 (Regularized reconstruction upper bound)**.** *For $\lambda > 0$, let $C^\lambda_{h,k} = \overline{C}_{h,k} + \lambda I$ and $C^\lambda_{\Delta,k} = \overline{C}_{\Delta,k} + \lambda I$. For every compression error E_k, equation 26 holds. Consequently, under (H1)–(H2),*

$$\mathbb{E}_{z,z'}|\Delta I^{[1]}_{\mathcal{T}}| \le \sum_k \frac{A_k + D_k}{\sqrt{\lambda}}\,\|(C^\lambda_{\Delta,k})^{1/2}E_k(C^\lambda_{h,k})^{1/2}\|_F.$$

Proof. Write $T_k = \|(C^\lambda_{\Delta,k})^{1/2}E_k(C^\lambda_{h,k})^{1/2}\|_F^2$. Since $C^\lambda_{\Delta,k} \succeq \lambda I$,

$$T_k \ge \lambda\|E_k(C^\lambda_{h,k})^{1/2}\|_F^2 \ge \lambda\|E_k\overline{C}_{h,k}^{1/2}\|_F^2.$$

The second inequality follows from $C^{\lambda}_{h,k} - \overline{C}_{h,k} = \lambda I \succeq 0$. Similarly, using $C^{\lambda}_{h,k} \succeq \lambda I$,

$$T_k \geq \lambda \|(C^{\lambda}_{\Delta,k})^{1/2} E_k\|_F^2 \geq \lambda \|\overline{C}^{1/2}_{\Delta,k} E_k\|_F^2.$$

Taking square roots, multiplying by A_k, D_k, and adding proves equation 26. Summing over k and applying Proposition 4.1 proves the stated consequence. □

For fixed rank and $A_k + D_k > 0$, the multiplicative coefficient does not affect the minimizer of the single-matrix weighted norm; the norm and its square have the same minimizers. If $A_k + D_k = 0$, any feasible matrix minimizes that term. This is an upper-bound justification for the auxiliary reconstruction problem, not an assertion that its singular-value truncation minimizes the original sum of norms. The construction further replaces population moments by empirical probe moments and restricts the matrix to their sampled subspaces, as specified in Appendix F.

E Pairwise Interaction from Propagated Errors

This appendix isolates one specific $O(E^2)$ component: the interaction created when the errors $-J_k E_k \Delta h_{k-1}$ and $-J_{k-1}^{\top} E_k^{\top} \Delta\delta_k$ propagate through the simplified sequential model. It does not bound every second-order term in the full score deviation.

Proof of the propagated pair-interaction bound. **Step 1 (forward interaction).** Insert the first-order forward chain into the interaction at layer k:

$$\begin{aligned} -J_k E_k \Delta h^{[1]}_{k-1} &= -J_k E_k \left(-\sum_{i<k} \mathcal{F}_{k-1\leftarrow i} \varepsilon_i \right) \\ &= J_k E_k \sum_{i<k} \mathcal{F}_{k-1\leftarrow i} \varepsilon_i. \end{aligned}$$

Propagating this injection to $a \geq k$ gives

$$\begin{aligned} \Delta h^{[2,\mathrm{fwd}]}_a &= \sum_{k\leq a} P_a \cdots P_{k+1} \left(-J_k E_k \Delta h^{[1]}_{k-1} \right) \\ &= \sum_{i<k\leq a} P_a \cdots P_{k+1} J_k E_k \mathcal{F}_{k-1\leftarrow i} \varepsilon_i \\ &= \sum_{i<k\leq a} \mathcal{F}_{a\leftarrow k} E_k \mathcal{F}_{k-1\leftarrow i} \varepsilon_i. \end{aligned}$$

Step 2 (backward interaction). The analogous backward component is

$$\begin{aligned} \Delta\delta^{[2,\mathrm{bwd}]}_a &= \sum_{k>a} G^{\mathrm{bwd}}_{a+1} \cdots G^{\mathrm{bwd}}_{k-1} \left(-J_{k-1}^{\top} E_k^{\top} \Delta\delta^{[1]}_k \right) \\ &= \sum_{a<k<i} G^{\mathrm{bwd}}_{a+1} \cdots G^{\mathrm{bwd}}_{k-1} J_{k-1}^{\top} E_k^{\top} \mathcal{B}_{k\leftarrow i} \eta_i \\ &= \sum_{a<k<i} \mathcal{B}_{a\leftarrow k} E_k^{\top} \mathcal{B}_{k\leftarrow i} \eta_i. \end{aligned}$$

Thus every displayed term contains two compression errors.

Step 3 (forward pair coefficient). The forward propagation component of the second-order score is defined by

$$\begin{aligned} \Delta I^{[2,\mathrm{fwd}]}_{\mathcal{T},\mathrm{prop}} = \sum_{\ell=1}^{L} &\langle \delta_\ell, \delta'_\ell \rangle \langle \Delta h^{[2,\mathrm{fwd}]}_{\ell-1}, h'_{\ell-1} \rangle \\ &+ \mathrm{sym}(z \leftrightarrow z'). \end{aligned}$$

For one consumer ℓ, substitute the forward expansion from Step 1:

$$\begin{aligned}
&\left|\langle \Delta h_{\ell-1}^{[2,\mathrm{fwd}]}, h'_{\ell-1}\rangle\langle \delta_\ell, \delta'_\ell\rangle\right| \\
&= \left|\sum_{i<k\le \ell-1} \langle \mathcal{F}_{\ell-1\leftarrow k} E_k \mathcal{F}_{k-1\leftarrow i}\varepsilon_i, h'_{\ell-1}\rangle\langle \delta_\ell, \delta'_\ell\rangle\right| \\
&\le \sum_{i<k\le \ell-1} \left|\langle \mathcal{F}_{\ell-1\leftarrow k} E_k \mathcal{F}_{k-1\leftarrow i}\varepsilon_i, h'_{\ell-1}\rangle\right| \, |\langle \delta_\ell, \delta'_\ell\rangle| \\
&\le \sum_{i<k\le \ell-1} \|\mathcal{F}_{\ell-1\leftarrow k}\|_{\mathrm{op}}\|E_k\|_F\|\mathcal{F}_{k-1\leftarrow i}\|_{\mathrm{op}}\|E_i\|_F\|h_{i-1}\|\|h'_{\ell-1}\| \, |\langle \delta_\ell, \delta'_\ell\rangle|.
\end{aligned}$$

After summing consumers, let $G_{ik}^{\mathrm{fwd}}(z, z')$ denote the displayed amplitude sum,

$$\begin{aligned}
G_{ik}^{\mathrm{fwd}}(z, z') = \sum_{\ell>k} &\|\mathcal{F}_{\ell-1\leftarrow k}\|_{\mathrm{op}}\|\mathcal{F}_{k-1\leftarrow i}\|_{\mathrm{op}} \\
&\cdot \|h_{i-1}\|\|h'_{\ell-1}\||\langle \delta_\ell, \delta'_\ell\rangle|.
\end{aligned}$$

The complete coefficient includes the sample-swapped score half:

$$\Gamma_{ik}^{\mathrm{fwd}} = \mathbb{E}\left[G_{ik}^{\mathrm{fwd}}(z, z') + G_{ik}^{\mathrm{fwd}}(z', z)\right]. \tag{27}$$

Summing the preceding consumer bound and its swapped copy gives

$$\left|\Delta I_{\mathcal{T}}^{[2,\mathrm{fwd}]}\right| \le \sum_{i<k} \|E_i\|_F\|E_k\|_F \left[G_{ik}^{\mathrm{fwd}}(z, z') + G_{ik}^{\mathrm{fwd}}(z', z)\right].$$

Taking expectations and using the definition above yields

$$\mathbb{E}\left|\Delta I_{\mathcal{T}}^{[2,\mathrm{fwd}]}\right| \le \sum_{i<k} \Gamma_{ik}^{\mathrm{fwd}}\|E_i\|_F\|E_k\|_F.$$

Step 4 (backward pair coefficient). Applying the same substitution and triangle inequality to the backward expansion defines

$$\begin{aligned}
G_{ik}^{\mathrm{bwd}}(z, z') &= \sum_{a<i} \|\mathcal{B}_{a\leftarrow i}\|_{\mathrm{op}}\|\mathcal{B}_{i\leftarrow k}\|_{\mathrm{op}} \\
&\quad \cdot \|\delta_k\|\|\delta'_a\||\langle h_{a-1}, h'_{a-1}\rangle|, \\
\Gamma_{ik}^{\mathrm{bwd}} &= \mathbb{E}\left[G_{ik}^{\mathrm{bwd}}(z, z') + G_{ik}^{\mathrm{bwd}}(z', z)\right], \qquad i<k.
\end{aligned}$$

Therefore

$$\begin{aligned}
\left|\Delta I_{\mathcal{T}}^{[2,\mathrm{bwd}]}\right| &\le \sum_{i<k} \|E_i\|_F\|E_k\|_F \left[G_{ik}^{\mathrm{bwd}}(z, z') + G_{ik}^{\mathrm{bwd}}(z', z)\right], \\
\mathbb{E}\left|\Delta I_{\mathcal{T}}^{[2,\mathrm{bwd}]}\right| &\le \sum_{i<k} \Gamma_{ik}^{\mathrm{bwd}}\|E_i\|_F\|E_k\|_F.
\end{aligned}$$

Define $\Delta I_{\mathcal{T},\mathrm{prop}}^{[2]} = \Delta I_{\mathcal{T}}^{[2,\mathrm{fwd}]} + \Delta I_{\mathcal{T}}^{[2,\mathrm{bwd}]}$ and $\Gamma_{ik} = \Gamma_{ik}^{\mathrm{fwd}} + \Gamma_{ik}^{\mathrm{bwd}}$. Applying the triangle inequality to these two channel bounds proves equation 17.

Step 5 (from the bound to the allocator). The production allocator does not evaluate the theoretical coefficient Γ_{ik} from the preceding proof. Instead, it measures a source–consumer interaction table by projection type and block distance. Before the table is aggregated into the symmetric empirical matrix Γ^{raw}, the reach correction multiplies each directed measurement by the consumer's mean forward amplification. The allocator then replaces the Frobenius errors by $x_k = \sqrt{\widetilde{Q}_k(r_k)}$, normalizes each interaction row by the mean initial interaction strength of its projection type, and obtains generally asymmetric $\widehat{\Gamma}$. These are heuristic allocation choices, not steps in the preceding inequality. The fixed-scale evaluation objective and the beta-bearing per-matrix proposal scores are defined in §4.4 and Appendix K. □

F CONSTRUCTION OF THE REGULARIZED BASIS

We derive the solution of the empirical weighted reconstruction problem equation 8, then express its components in the original weight coordinates. The computational construction is inherited from (Chen et al., 2026). Throughout this appendix, D_H, D_Δ denote positive diagonal matrices, not vectors.

Step 1 (sampled second moments). Let $H_k \in \mathbb{R}^{n_k \times N}$ and $\Delta_k \in \mathbb{R}^{m_k \times N}$ contain the N sampled activation and gradient columns. Suppressing k on the basis factors, compact SVDs over the nonzero singular subspaces give

$$H_k = U_H S_H V_H^\top, \qquad \Delta_k = U_\Delta S_\Delta V_\Delta^\top,$$

where $U_H, V_H, U_\Delta, V_\Delta$ have orthonormal columns and S_H, S_Δ are diagonal matrices of positive singular values. Therefore

$$\widehat{C}_{h,k} = \frac{H_k H_k^\top}{N} = U_H \frac{S_H^2}{N} U_H^\top,$$
$$\widehat{C}_{\Delta,k} = \frac{\Delta_k \Delta_k^\top}{N} = U_\Delta \frac{S_\Delta^2}{N} U_\Delta^\top.$$

This computes their nonzero eigenspaces without forming the full moment matrices. The sampled moments estimate statistics of the construction data; they are not assumed equal to the candidate–target averages in Corollary D.2.

Step 2 (regularized scales and subspaces). With ridge $\lambda > 0$, define

$$D_H = (S_H^2/N + \lambda I)^{1/2}, \qquad D_\Delta = (S_\Delta^2/N + \lambda I)^{1/2}. \tag{28}$$

Their diagonal entries are positive. Define the orthogonal projectors $\Pi_{h,k} = \Pi_H = U_H U_H^\top$ and $\Pi_{\Delta,k} = \Pi_\Delta = U_\Delta U_\Delta^\top$. For example,

$$\Pi_H(\widehat{C}_{h,k} + \lambda I)\Pi_H = U_H D_H^2 U_H^\top.$$

The ridge prevents inversion of tiny empirical eigenvalues; the median eigenvalue reported for a query-projection input moment is approximately 10^{-34}.

Step 3 (the optimization problem and its SVD solution). Any compressed matrix confined to these subspaces has the form $\widehat{W}_k = U_\Delta X U_H^\top$, with $\operatorname{rank}(\widehat{W}_k) = \operatorname{rank}(X)$. Write $W_{\text{core}} = U_\Delta^\top W_k U_H$. The fixed-rank reconstruction problem is

$$\min_{\operatorname{rank}(X) \le r} \|D_\Delta (W_{\text{core}} - X) D_H\|_F^2.$$

The empirical moments preserve these subspaces and their orthogonal complements. Thus this objective differs from the corresponding full-space regularized reconstruction error only by a constant independent of X. The invertible change of variables $Y = D_\Delta X D_H$ preserves rank and turns it into

$$\min_{\operatorname{rank}(Y) \le r} \|S_{\text{core},k} - Y\|_F^2, \qquad S_{\text{core},k} = D_\Delta W_{\text{core}} D_H.$$

Let $S_{\text{core},k} = P \operatorname{diag}(\sigma_k) R^\top$ be its compact SVD, with the singular values in nonincreasing order for this reconstruction problem. By the best low-rank approximation property of SVD, a minimizer is $Y^* = [S_{\text{core},k}]_r$, the sum of its largest r singular components. Mapping back gives

$$X^* = D_\Delta^{-1} [S_{\text{core},k}]_r D_H^{-1}, \qquad \widehat{W}_{k,r}^{\text{rec}} = U_\Delta X^* U_H^\top. \tag{29}$$

The optimal objective value is $\sum_{a=r}^{R_k - 1} \sigma_{k,a}^2$, using zero-based indexing. Repeated singular values may give multiple minimizers.

This establishes optimality for the auxiliary empirical reconstruction objective on the sampled subspaces. The complete basis is subsequently reused for the propagation-based selection problem; retaining a different set need not remain optimal for reconstruction.

Step 4 (components in the original coordinates). The mapped directions are

$$p_a = U_\Delta D_\Delta^{-1} P_{:,a}, \qquad q_a = U_H D_H^{-1} R_{:,a}, \tag{30}$$

where the leading index k is suppressed. They need not be Euclidean unit vectors. Their weighted orthonormality follows from

$$\begin{aligned} p_a^\top (U_\Delta D_\Delta^2 U_\Delta^\top) p_b &= P_{:,a}^\top D_\Delta^{-1} U_\Delta^\top U_\Delta D_\Delta^2 U_\Delta^\top U_\Delta D_\Delta^{-1} P_{:,b} \\ &= P_{:,a}^\top P_{:,b} = \delta_{ab}. \end{aligned}$$

Here δ_{ab} is the Kronecker delta; the input-side identity for q_a, q_b follows in the same way. Thus, in the identity-path ablation, $d_a^{\mathrm{fwd}} = \|p_a\|^2$ need not equal one.

Step 5 (retained components and truncation error). For any retained set $\mathcal{S}_k$,

$$\begin{aligned} \widehat{W}_k &= U_\Delta D_\Delta^{-1} \left[\sum_{a \in \mathcal{S}_k} \sigma_{k,a} P_{:,a} R_{:,a}^\top \right] D_H^{-1} U_H^\top \\ &= \sum_{a \in \mathcal{S}_k} \sigma_{k,a} p_a q_a^\top. \end{aligned}$$

Keeping every component yields

$$U_\Delta D_\Delta^{-1} S_{\mathrm{core},k} D_H^{-1} U_H^\top = \Pi_\Delta W_k \Pi_H.$$

Subtracting the retained sum therefore gives

$$E_k^{\mathrm{span}} = \Pi_{\Delta,k} W_k \Pi_{h,k} - \widehat{W}_k = \sum_{a \notin \mathcal{S}_k} \sigma_{k,a} p_{k,a} q_{k,a}^\top. \tag{31}$$

The contribution $\sigma_{k,a}^2$ is the squared error of discarding component a in weighted core coordinates. In the original coordinates, its squared Frobenius norm is $\sigma_{k,a}^2 \|p_a\|^2 \|q_a\|^2$.

What the sampled basis can control. Let $E_k^{\mathrm{out}} = W_k - \Pi_\Delta W_k \Pi_H$ denote the error outside the basis, so $E_k = E_k^{\mathrm{out}} + E_k^{\mathrm{span}}$. The decomposition

$$E_k^{\mathrm{out}} = (I - \Pi_\Delta) W_k + \Pi_\Delta W_k (I - \Pi_H)$$

shows that each of its two terms is Frobenius-orthogonal to E_k^{span}, whose columns lie in $\mathrm{range}(\Pi_\Delta)$ and rows in $\mathrm{range}(\Pi_H)$. Hence

$$\|W_k - \widehat{W}_k\|_F^2 = \|E_k^{\mathrm{out}}\|_F^2 + \|E_k^{\mathrm{span}}\|_F^2. \tag{32}$$

Only the second term can be changed by selecting components. This orthogonality is specific to the Frobenius metric; propagation matrices can mix these subspaces, so it does not imply that the cross-term between E_k^{out} and E_k^{span} vanishes in equation 37. The ranking surrogate applies to E_k^{span} and omits that cross-term.

G Unbiasedness of the Random-Sign Path Estimator

The propagation scores equation 12 are quadratic forms in the matrices equation 36, which are never formed. Let r be a Rademacher vector whose entries are independent signs taking $+1$ and -1 with equal probability, so that $\mathbb{E}[rr^\top] = I$. For any fixed product of layer derivatives $\mathcal{F}$ and vector x, the random-sign estimator satisfies

$$\begin{aligned} \mathbb{E}_r\left[(x^\top \mathcal{F}^\top r)^2\right] &= \mathbb{E}_r\left[r^\top \mathcal{F} x x^\top \mathcal{F}^\top r\right] \\ &= \mathrm{tr}\left[\mathcal{F} x x^\top \mathcal{F}^\top \mathbb{E}(rr^\top)\right] \\ &= \mathrm{tr}[\mathcal{F} x x^\top \mathcal{F}^\top] \\ &= x^\top \mathcal{F}^\top \mathcal{F} x \\ &= \|\mathcal{F} x\|^2. \end{aligned} \tag{33}$$

The second equality writes the scalar as a trace and uses cyclicity; the third pulls the expectation inside because $\mathcal{F} x x^\top \mathcal{F}^\top$ does not depend on r. For a sum of paths with possibly different output

spaces, use independent random sign vectors r_ℓ and define $g_k = \sum_{\ell>k} \mathcal{F}_{\ell-1\leftarrow k}^\top r_\ell$. Then

$$\begin{aligned}\mathbb{E}_r[g_k g_k^\top \mid z] &= \sum_{\ell,m>k} \mathcal{F}_{\ell-1\leftarrow k}^\top \mathbb{E}_r[r_\ell r_m^\top] \mathcal{F}_{m-1\leftarrow k} \\ &= \sum_{\ell>k} \mathcal{F}_{\ell-1\leftarrow k}^\top \mathcal{F}_{\ell-1\leftarrow k},\end{aligned}$$

because $\mathbb{E}_r[r_\ell r_m^\top] = 0$ for $\ell \neq m$ and equals I for $\ell = m$. Averaging independent random-sign collections and independently sampled inputs is therefore unbiased for the joint (z, r) expectation defining M_k^{fwd}, with the usual Monte Carlo variance decay.

H From Propagated Error to Component Selection

This appendix starts from the same propagation paths used to prove Proposition 4.1, before they are reduced to scalar operator-norm bounds. It connects the path error in equation 35, the matrix objective in equation 37, and the retained-set problem in equation 39. The expression in equation 11 is the single-component specialization of its forward channel. The derivation identifies the additional approximations separately. All moments below are assumed finite; their construction estimates are computed on the probe set.

For one discarded component $E_{k,a} = \sigma_{k,a} p_{k,a} q_{k,a}^\top$, the local forward and backward errors are

$$\begin{aligned}E_{k,a} h_{k-1} &= \sigma_{k,a} p_{k,a} (q_{k,a}^\top h_{k-1}), \\ E_{k,a}^\top \delta_k &= \sigma_{k,a} q_{k,a} (p_{k,a}^\top \delta_k).\end{aligned} \tag{34}$$

Propagating the forward error yields equation 11; the backward expression follows by applying $\mathcal{B}$ to the second line.

Path error and accumulated propagation matrices. Before the separability approximation, the error attributable to one source matrix is measured by

$$\begin{aligned}\mathcal{J}_k^{\mathrm{path}}(E_k) = \mathbb{E}_z \sum_{\ell>k} \|\mathcal{F}_{\ell-1\leftarrow k} E_k h_{k-1}\|^2 \\ + \mathbb{E}_z \sum_{\ell<k} \|\mathcal{B}_{\ell\leftarrow k} E_k^\top \delta_k\|^2.\end{aligned} \tag{35}$$

This is the energy of individual first-order propagation paths, not the complete joint score deviation. Define

$$\begin{aligned}M_k^{\mathrm{fwd}} &= \mathbb{E}_z \sum_{\ell>k} \mathcal{F}_{\ell-1\leftarrow k}^\top \mathcal{F}_{\ell-1\leftarrow k}, \\ M_k^{\mathrm{bwd}} &= \mathbb{E}_z \sum_{\ell<k} \mathcal{B}_{\ell\leftarrow k}^\top \mathcal{B}_{\ell\leftarrow k}.\end{aligned} \tag{36}$$

Then the path definitions in equation 12 are equivalently $d_{k,a}^{\mathrm{fwd}} = p_{k,a}^\top M_k^{\mathrm{fwd}} p_{k,a}$ and $d_{k,a}^{\mathrm{bwd}} = q_{k,a}^\top M_k^{\mathrm{bwd}} q_{k,a}$.

Step 1 (a quadratic form before separating statistics). Define the sample-dependent path matrices

$$K_k^{\mathrm{fwd}}(z) = \sum_{\ell>k} \mathcal{F}_{\ell-1\leftarrow k}(z)^\top \mathcal{F}_{\ell-1\leftarrow k}(z), \qquad K_k^{\mathrm{bwd}}(z) = \sum_{\ell<k} \mathcal{B}_{\ell\leftarrow k}(z)^\top \mathcal{B}_{\ell\leftarrow k}(z).$$

Their expectations are $M_k^{\mathrm{fwd}}, M_k^{\mathrm{bwd}}$. The two terms in equation 35 are exactly

$$\begin{aligned}\mathbb{E}_z \sum_{\ell>k} \|\mathcal{F}_{\ell-1\leftarrow k} E_k h_{k-1}\|^2 &= \mathbb{E}_z \operatorname{tr}\left[E_k^\top K_k^{\mathrm{fwd}}(z) E_k h_{k-1} h_{k-1}^\top\right], \\ \mathbb{E}_z \sum_{\ell<k} \|\mathcal{B}_{\ell\leftarrow k} E_k^\top \delta_k\|^2 &= \mathbb{E}_z \operatorname{tr}\left[E_k K_k^{\mathrm{bwd}}(z) E_k^\top \delta_k \delta_k^\top\right].\end{aligned}$$

The propagation matrices and the data vectors depend on the same example, so their expectations do not generally factor.

Step 2 (separability and empirical moments). Approximation (H3a) replaces each joint moment by the product of its path and data moments. For example,

$$\mathbb{E}_z \operatorname{tr}\big[E_k^\top K_k^{\mathrm{fwd}}(z) E_k h_{k-1} h_{k-1}^\top\big] \approx \operatorname{tr}(E_k^\top M_k^{\mathrm{fwd}} E_k C_{h,k}).$$

Replacing $C_{h,k}, C_{\Delta,k}$ by the empirical probe moments then yields

$$\widehat{\mathcal{J}}_k(E_k) = \operatorname{tr}(E_k^\top M_k^{\mathrm{fwd}} E_k \widehat{C}_{h,k}) + \operatorname{tr}(E_k M_k^{\mathrm{bwd}} E_k^\top \widehat{C}_{\Delta,k}),$$

or, equivalently,

$$\begin{aligned}\widehat{\mathcal{J}}_k(E_k) = \|(M_k^{\mathrm{fwd}})^{1/2} E_k \widehat{C}_{h,k}^{1/2}\|_F^2 \\ + \|\widehat{C}_{\Delta,k}^{1/2} E_k (M_k^{\mathrm{bwd}})^{1/2}\|_F^2.\end{aligned} \tag{37}$$

The path matrices are positive semidefinite, so their square roots exist. In computation, only their directional quadratic forms are estimated, using Appendix G. This surrogate measures propagated error energy; it is not an algebraic reformulation of the alignment bound.

Step 3 (restriction to the basis and diagonal approximation). Fix matrix k and suppress k on component quantities. Substitute $E_k^{\mathrm{span}} = \sum_{a \notin \mathcal{S}_k} \sigma_a p_a q_a^\top$. The forward part expands exactly as

$$\operatorname{tr}\Big[(E_k^{\mathrm{span}})^\top M_k^{\mathrm{fwd}} E_k^{\mathrm{span}} \widehat{C}_{h,k}\Big] = \sum_{a,b \notin \mathcal{S}_k} \sigma_a \sigma_b (p_a^\top M_k^{\mathrm{fwd}} p_b)(q_a^\top \widehat{C}_{h,k} q_b).$$

The backward part similarly contains

$$\sum_{a,b \notin \mathcal{S}_k} \sigma_a \sigma_b (q_a^\top M_k^{\mathrm{bwd}} q_b)(p_a^\top \widehat{C}_{\Delta,k} p_b).$$

Approximation (H3b) discards the $a \neq b$ terms. Keeping the $a = b$ contributions gives

$$\widehat{\mathcal{J}}_k(E_k^{\mathrm{span}}) \approx \sum_{a \notin \mathcal{S}_k} \sigma_{k,a}^2 \big(d_{k,a}^{\mathrm{fwd}} u_{k,a}^{\mathrm{fwd}} + d_{k,a}^{\mathrm{bwd}} u_{k,a}^{\mathrm{bwd}}\big). \tag{38}$$

Here d, u are defined in equation 12. Weighted orthonormality of the SVD basis alone does not make the off-diagonal propagation terms vanish. The expression applies to E_k^{span} and omits its cross-terms with E_k^{out}. Thus the retained-set construction uses an approximation within the fixed basis, rather than solving the matrix objective over all rank-r matrices.

Step 4 (the objective after channel balancing). Let $\bar{d}^c = R_k^{-1} \sum_a d_a^c$ and $\bar{u}^c = R_k^{-1} \sum_a u_a^c$, where $c \in \{\mathrm{fwd}, \mathrm{bwd}\}$ denotes a channel. For positive channel means, normalization by the two means is exactly a fixed channel reweighting:

$$\kappa^c = \frac{1}{\bar{d}^c \bar{u}^c}, \qquad \sigma_a^2 \widetilde{d}_a^c \widetilde{u}_a^c = \kappa^c \sigma_a^2 d_a^c u_a^c.$$

The balanced diagonal objective is consequently

$$\sum_{a \notin \mathcal{S}_k} \sum_{c \in \{\mathrm{fwd}, \mathrm{bwd}\}} \kappa^c \sigma_a^2 d_a^c u_a^c = \sum_{a \notin \mathcal{S}_k} \mathrm{key}_{k,a}.$$

The means are computed over all available components, before choosing the retained set, so these costs are fixed during selection. This reweighting can change the optimizer of the unbalanced diagonal objective.

For a fixed rank r_k, we therefore solve

$$\mathcal{S}_k^*(r_k) \in \operatorname*{arg\,min}_{\mathcal{S} \subseteq \{0,\dots,R_k-1\},\ |\mathcal{S}| = r_k} \sum_{a \notin \mathcal{S}} \mathrm{key}_{k,a}. \tag{39}$$

Lemma H.1 (Optimal selection for fixed component costs)**.** *For any fixed finite component costs* $\mathrm{key}_{k,a}$ *and integer* $0 \le r \le R_k$*, a set containing the* r *largest costs solves equation 39.*

Proof. The sum of all costs is constant, so minimizing the discarded sum is equivalent to maximizing the retained sum. If a retained component has lower cost than a discarded component, exchanging them decreases the discarded sum. Repeating such exchanges yields the r largest costs. Ties permit multiple optimal sets. □

The lemma is exact for the stated selection problem. It does not assert global optimality over all rank-r matrices, account for error outside the basis, or remove approximations (H3a)–(H3b). The rank allocator uses a further simplified error tail, described next.

I THE REMAINING TRUNCATION-ERROR PROXY $\widetilde{Q}_k$

Direction ranking and rank allocation use different proxies. The ranking score equation 13 includes measured propagation. After sorting directions by that score, allocation uses

$$\omega_{k,a} = \sigma_{k,a}^2\big(u_{k,a}^{\mathrm{fwd}} + u_{k,a}^{\mathrm{bwd}}\big),$$
$$\widetilde{Q}_k(r) = \sum_{a=r}^{R_k-1} \omega_{k,a}.$$

with zero-based direction index a. Thus propagation and data use determine the retained order, while $\widetilde{Q}_k(r)$ records the data-use-weighted truncation error remaining after rank r is retained.

For one discarded component $E_{k,a} = \sigma_{k,a} p_{k,a} q_{k,a}^\top$, the two diagonal error contributions are

$$\mathbb{E}\|E_{k,a}h_{k-1}\|^2 = \sigma_{k,a}^2\|p_{k,a}\|^2\mathbb{E}(q_{k,a}^\top h_{k-1})^2 = \sigma_{k,a}^2\|p_{k,a}\|^2 u_{k,a}^{\mathrm{fwd}},$$
$$\mathbb{E}\|E_{k,a}^\top\delta_k\|^2 = \sigma_{k,a}^2\|q_{k,a}\|^2\mathbb{E}(p_{k,a}^\top\delta_k)^2 = \sigma_{k,a}^2\|q_{k,a}\|^2 u_{k,a}^{\mathrm{bwd}}.$$

After summing diagonal discarded-direction terms, the identity-path quantity contains

$$\sum_{a\ge r}\sigma_{k,a}^2\left(\|p_{k,a}\|^2 u_{k,a}^{\mathrm{fwd}} + \|q_{k,a}\|^2 u_{k,a}^{\mathrm{bwd}}\right).$$

The production allocation proxy deliberately drops the original-coordinate norm factors and uses $\widetilde{Q}_k(r)$ instead. It is therefore not a transformed theorem bound. The fixed error outside the sampled subspaces is also not controlled by rank allocation. The allocator combines this remaining-error proxy with the matrix weight $\widehat{a}_k$ and the rank constraint equation 14.

J ACCURACY OF THE TAYLOR APPROXIMATION

The first-order priority used when $\beta = 0$ is

$$\pi_k = \frac{\omega_{k,r_k}}{c_k}\frac{\widehat{a}_k}{2\sqrt{\widetilde{Q}_k(r_k)}}. \tag{40}$$

It is the Taylor decrease in Φ_1 per unit of storage cost. The following calculation compares its error-reduction factor with the exact marginal.

The per-matrix allocation procedure may use a Taylor approximation to the decrease in remaining truncation error. Let $Q = \widetilde{Q}_k(r) > 0$ and $v = \widetilde{Q}_k(r) - \widetilde{Q}_k(r{+}1) = \omega_{k,r}$, so $0 \le v \le Q$. The exact decrease is $\sqrt{Q} - \sqrt{Q-v}$ and the Taylor decrease is $v/(2\sqrt{Q})$. Since $f(y) = \sqrt{y}$ is concave, its tangent lies above the graph, $f(y) \le f(Q) + f'(Q)(y-Q)$; taking $y = Q - v$ and $f'(Q) = 1/(2\sqrt{Q})$,

$$\sqrt{Q-v} \le \sqrt{Q} - \frac{v}{2\sqrt{Q}} \quad\Longrightarrow\quad \underbrace{\sqrt{Q} - \sqrt{Q-v}}_{\text{true}} \ge \underbrace{\frac{v}{2\sqrt{Q}}}_{\text{Taylor}},$$

with equality if and only if $v = 0$. For $0 < v \le Q$, the ratio between the two simplifies exactly,

$$\frac{\sqrt{Q} - \sqrt{Q-v}}{v/(2\sqrt{Q})} = \frac{2\sqrt{Q}\left(\sqrt{Q} - \sqrt{Q-v}\right)}{\left(\sqrt{Q} - \sqrt{Q-v}\right)\left(\sqrt{Q} + \sqrt{Q-v}\right)}$$
$$= \frac{2\sqrt{Q}}{\sqrt{Q} + \sqrt{Q-v}}.$$

using $v = Q - (Q-v) = \left(\sqrt{Q} - \sqrt{Q-v}\right)\left(\sqrt{Q} + \sqrt{Q-v}\right)$. For $0 < v \le Q$,

$$\sqrt{Q} \le \sqrt{Q} + \sqrt{Q-v} \le 2\sqrt{Q} \quad\Longrightarrow\quad 1 \le \frac{2\sqrt{Q}}{\sqrt{Q} + \sqrt{Q-v}} \le 2.$$

For $v = 0$, both gains vanish and no ratio is needed. Thus the Taylor gain never exceeds the true gain and differs by at most a factor of two. It is used because, when $\beta = 0$, it reproduces the first-order priority equation 40; the exact decrease is retained as an ablation.

K PER-MATRIX ALLOCATION UNDER DENSE INTERACTIONS

Storage budget and per-matrix caps. If r_k^{base} is the IProX rank, use $B_{\text{rank}} = \sum_k c_k r_k^{\text{base}}$ with $c_k = m_k + n_k$, as in equation 14. To set the cap $\bar{r}_k$, define the storage break-even rank $r_k^* = m_k n_k / c_k$, at which low-rank factors cost as much as the dense matrix. Let $\rho_k = r_k / r_k^*$ be the resulting storage ratio, and let $\rho_{\text{unif}} = \max_k (r_k^{\text{base}} / r_k^*)$ be the smallest uniform ratio containing the baseline allocation. The cap satisfies

$$\rho_k \le \rho_{\max}(\tau) = \rho_{\text{unif}} + \tau(1 - \rho_{\text{unif}}), \qquad \bar{r}_k \le \lfloor \rho_{\max}(\tau) r_k^* \rfloor, \tag{41}$$

where $\tau \in [0, 1]$ controls how far a matrix may exceed the uniform ratio. At $\tau = 0$, the common cap is ρ_{unif}; the actual ratios need not be equal. At $\tau = 1$, a matrix may use storage up to break-even. If $\rho_{\text{unif}} < 1$, any $\tau < 1$ keeps every selected matrix strictly compressed.

Constructing the empirical interaction table. The allocator uses empirical interactions instead of evaluating the theoretical coefficients in equation 17. It groups measured source–consumer interactions by projection type and Transformer-block distance. A directed measurement $\gamma_{i \to k}^{\text{raw}}$ is multiplied by the consumer's mean forward propagation strength, giving $\gamma_{i \to k}^{\text{reach}} = \gamma_{i \to k}^{\text{raw}} \bar{d}_{k,\cdot}^{\text{fwd}}$, before aggregation into a symmetric table Γ^{raw}. This adjustment accounts approximately for propagation beyond the measured consumer. It does not identify the empirical table with the coefficients in equation 17.

Let $x_k = \sqrt{\widetilde{Q}_k(r_k)}$ be the remaining error scale, and let $\text{kind}(k)$ denote the projection type of matrix k (for example, query or value). We normalize the empirical table by the mean initial interaction strength of each projection type:

$$\begin{aligned} c_{0,k} &= \sum_{j \ne k} \Gamma_{kj}^{\text{raw}} \sqrt{\widetilde{Q}_j(0)}, \\ \bar{c}_k &= \operatorname*{mean}_{i:\, \text{kind}(i) = \text{kind}(k)} c_{0,i}, \quad \widehat{\Gamma}_{kj} = \frac{\Gamma_{kj}^{\text{raw}}}{\bar{c}_k}. \end{aligned} \tag{42}$$

Here $c_{0,k}$ is the interaction strength before allocating any rank and $\bar{c}_k$ is its mean within the projection type. Row normalization generally makes $\widehat{\Gamma}$ asymmetric.

Updating rank proposals.

The normalized interaction matrix $\widehat{\Gamma}$ is dense. Changing $x_j = \sqrt{\widetilde{Q}_j}$ changes every other selected matrix's proposal score b_k^{row} through the column $\widehat{\Gamma}_{:j}$, so a lazy priority queue becomes stale after every allocation. If one allocation changes only x_j by Δx_j, the implementation updates

$$\widehat{\Gamma} x \;\leftarrow\; \widehat{\Gamma} x + \Delta x_j \, \widehat{\Gamma}_{:j}.$$

It scans all feasible selected matrices and periodically recomputes this product to measure numerical drift.

Evaluation and proposal are distinct. The scalar value reported for an allocation is equation 18. Since $x^\top \widehat{\Gamma} x$ is a scalar,

$$\begin{aligned} \frac{1}{2} x^\top \widehat{\Gamma} x &= \frac{1}{4} x^\top (\widehat{\Gamma} + \widehat{\Gamma}^\top) x = \frac{1}{2} x^\top \overline{\Gamma} x, \\ \nabla_x \Phi_2^{\text{eval}} &= \widehat{a} + \overline{\Gamma} x. \end{aligned}$$

To propose a rank increment, the allocation procedure multiplies the Taylor error reduction per storage cost, $\omega_{k,r_k} / (2 c_k x_k)$, by the directed score

$$b_k^{\text{row}} = \widehat{a}_k + \beta \sum_{j \ne k} \widehat{\Gamma}_{kj} x_j. \tag{43}$$

We use $\beta = 1$; the $\beta = 0$ ablation removes interactions from the proposals while leaving the evaluation criterion equation 18 unchanged. In vector form, $b^{\text{row}} = \widehat{a} + \beta \widehat{\Gamma} x$. Its difference from the fixed-scale evaluation gradient is

$$b^{\text{row}} - \nabla_x \Phi_2^{\text{eval}} = (\beta \widehat{\Gamma} - \overline{\Gamma}) x.$$

Thus at $\beta = 1$ the difference is the effect of the asymmetric row normalization, while at $\beta = 0$ the proposal score reduces to $\widehat{a}$. Eight fixed-point passes freeze the previous proposal scores and generate feasible candidate allocations. We retain the candidate with the lowest fixed-scale evaluation value. The procedure is therefore not exact greedy minimization of equation 18; it is an asymmetric candidate generator evaluated by a symmetrized scalar proxy.