POST-HOC PAIRED-COORDINATE REDUCTION IN LORA-RL REASONING ADAPTERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Parameter-efficient reinforcement learning makes reasoning post-training practical, yet completed low-rank adapters are typically retained at the rank chosen before or during optimization. Existing adaptive-rank methods mainly modify training itself, leaving open whether a completed LoRA-RL factorization contains removable capacity. We study this question through post-hoc paired-coordinate selection: checkpoint-derived scores rank shared LoRA factor coordinates, a fixed Top-k set is sliced consistently across modules, and the original scaling is preserved without retraining or using downstream task outcomes for selection. On a rank-32 LoRA-RL adapter, evaluations across MATH500, AIME25, and AMC23 at $k \in \{16, 20, 24\}$ reveal substantial but task- and budget-dependent retention relative to random selection and per-module SVD reconstruction. Validity diagnostics further show that successful endpoints do not imply intrinsic or representation-invariant coordinate importance. As a cross-family test, applying the Energy selector to a separately trained DoRA-r32 adapter retains MATH500 and AMC23 performance at $k = 16$ while AIME25 decreases, showing that post-hoc reduction transfers beyond the original LoRA parameterization but remains task dependent. These results identify usable post-hoc redundancy while bounding its interpretation to the stored factorization and evaluation protocol.

1 INTRODUCTION

Low-rank adaptation (LoRA) makes reasoning-oriented post-training feasible without updating an entire language model. It freezes the backbone and stores task adaptation in compact factors, while Group Relative Policy Optimization (GRPO) and LoRA-based reasoning models demonstrate that this training regime can change mathematical-reasoning behavior (Hu et al., 2021; Shao et al., 2024; Wang et al., 2025). Such an adapter is compact as a whole, yet its internal source-rank coordinates are normally retained at the rank selected before training. This leaves a practical analytical gap between the training-time choice of adapter rank and the behavior of a completed, stored factorization.

The gap matters because a LoRA rank is not only a storage budget after training. The completed factors provide a particular coordinate representation of the learned update, and retaining fewer coordinates changes that representation through a concrete selection and reconstruction rule. A post-hoc result is therefore interpretable only after specifying what is selected, whether the same indices are retained across modules, how the reduced factors are scaled, and how the reconstructed adapter is evaluated. Without these choices, a reported lower-rank outcome can conflate a changed coordinate set, a changed parameterization, and a changed evaluation protocol. This distinction is especially consequential for reasoning adapters: a small apparent change in aggregate accuracy can arise from different sampled generations, while a reconstruction can preserve a task endpoint without validating the score used to select it. The question is thus not whether one can remove coordinates in general, but what the completed factorization and a fully recorded post-hoc procedure allow one to conclude about the induced reconstruction. It is not a substitute for rerunning the RL optimization at a smaller training-time rank.

Existing parameter-efficient fine-tuning (PEFT) methods address adjacent but distinct objects. DyLoRA and AdaLoRA determine or allocate rank during optimization, whereas QLoRA reduces fine-tuning memory through a quantized frozen backbone (Valipour et al., 2022; Zhang et al., 2023; Dettmers et al., 2023). GeoRA uses a geometry-aware initialization and residual anchor for RLVR

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.

training (Zhang et al., 2026); it therefore changes the training parameterization rather than compressing a completed adapter. Recent post-optimization adaptive rank allocation uses a global singular-value threshold to assign non-uniform layer-wise ranks without retraining (Kumaravelu et al., 2026). A distinct question remains: after RL optimization has finished, how much of the realized LoRA adaptation must still be retained, and does preserving stored coordinates behave differently from reconstructing new module-local singular directions? Component energy, random subsets, and SVD-DeltaW provide matched post-hoc comparisons, but they do not define a universal coordinate semantics.

Our procedure reads saved adapter weights only after GRPO training. It introduces neither a policy update nor a new RL optimizer; in particular, rewards, verifiers, and task outputs are not pruning-score inputs. We analyze paired coordinates in a fixed stored LoRA factorization: a shared rank index pairs one row of every LoRA A factor with the matching column of its B factor. Post-hoc paired-coordinate selection ranks this factorization-specific object from checkpoint weights, selects one fixed-budget Top-k set, and slices paired rows and columns before separate evaluation. This framing is deliberately narrower than a claim that an individual coordinate is a model-intrinsic feature. Coordinate identity depends on the stored factorization, so neither basis invariance nor permutation invariance is assumed or established.

The central thesis is consequently descriptive rather than an optimization claim: post-hoc score-induced selection can be measured cleanly only when the selected object, reconstruction rule, denominator, and evaluation seeds are kept explicit. Our primary endpoint study evaluates one stored source-rank-$r = 32$ LoRA-RL adapter at checkpoint 360 across three mathematical-reasoning tasks and retained budgets $k \in \{16, 20, 24\}$. Its conclusions are deliberately scoped to this fixed factorization and recorded evaluation protocol. The observed paired outcomes include positive, zero, and negative differences, making task, budget, and generation seed part of the result rather than nuisance details to average away.

This paper makes four contributions. First, we formalize paired LoRA coordinates as factorization-specific bundles shared across adapted modules. Second, we give a fixed-budget reconstruction and evaluation protocol that separates score construction from task evaluation and compares stored-coordinate slicing with a matched per-module SVD reconstruction and random selection. Third, endpoint and validity analyses establish a bounded empirical finding: post-hoc capacity can be redundant, but coordinate importance is not shown to be stable or intrinsic. Fourth, an independently trained DoRA-r32 campaign provides cross-family transfer evidence under native DoRA reconstruction semantics, with task-dependent outcomes rather than a cross-family leaderboard. Training-time direct-rank, AdaLoRA, and GeoRA results remain methodological context rather than matched numerical baselines.

2 Related Work

Parameter-efficient adaptation and LoRA variants. Parameter-efficient fine-tuning (PEFT) updates a compact task-specific state while freezing most pretrained weights. LoRA uses a low-rank update and QLoRA reduces frozen-backbone memory through quantization (Houlsby et al., 2019; Hu et al., 2021; Dettmers et al., 2023); DoRA changes the low-rank parameterization by decomposing weight magnitude and direction (Liu et al., 2024). We instead hold one stored factorization fixed and examine a common coordinate budget selected only after training.

Training-time rank and post-hoc structure. DyLoRA and AdaLoRA choose or allocate rank during optimization, while AutoLoRA and SoRA learn related rank or sparsity structures (Valipour et al., 2022; Zhang et al., 2023; 2024; Ding et al., 2023). GeoLoRA adapts low-rank capacity during fine-tuning (Schotthöfer et al., 2024), whereas GeoRA changes RLVR initialization and parameterization with a geometry-aware structure and frozen residual anchor (Zhang et al., 2026). PARA is post-optimization but assigns non-uniform layer-wise ranks through a singular-value threshold (Kumaravelu et al., 2026). Our intervention begins only after training: it preserves one shared stored coordinate set, while SVD-DeltaW independently reconstructs each module's dense update.

RL reasoning and the open question. PPO, GRPO, and Tina contextualize RL reasoning and small LoRA-RL models (Schulman et al., 2017; Shao et al., 2024; Wang et al., 2025). Evalua-

tion depends on the evaluator and inference stack (Fourrier et al., 2023; Gao et al., 2024; Kwon et al., 2023); these works provide context rather than matched local baselines. Existing lines change training, rank allocation, parameterization, or reconstruction, but do not test whether one shared coordinate set in one completed LoRA-RL factorization has a stable ordering under paired reconstruction and matched evaluation. We therefore treat the selected set as an operational property of the stored representation and make no optimum, superiority, or intrinsic-coordinate claim.

3 METHOD

Motivated by this open question, our method analyzes rather than optimizes one stored LoRA-RL factorization. Its intervention point is strictly after RL optimization: it constructs checkpoint-derived scores for shared paired coordinates, selects one fixed-budget coordinate set, reconstructs the corresponding LoRA factors consistently, and evaluates the result under a matched protocol. This contribution provides a controlled measurement procedure for factorization-specific coordinate selection; it tests the behavior and limits of the resulting ordering without presuming that the ordering is stable or intrinsic. The measurement target is consequently an instantiated selection-and-reconstruction procedure, not an optimized pruning policy: a result is meaningful only together with its checkpoint, score family, retained budget, reconstruction scale, and evaluation protocol.

The three stages isolate three otherwise confounded choices. Stage 1 fixes the object of analysis: one global coordinate index denotes a paired row–column bundle across every adapted module. Stage 2 maps this stored object to alternative scalar score vectors using checkpoint weights only; it does not learn another policy or use downstream task outcomes. Stage 3 turns one score vector into an explicit index set, slices the same indices from all factors, restores the original scaling ratio, and evaluates the resulting adapter separately. This separation makes it possible to attribute a reported endpoint to a recorded selection-and-reconstruction procedure rather than to an unspecified notion of coordinate importance. In particular, applying the same index set to every adapted module preserves the meaning of a paired coordinate throughout reconstruction: the procedure cannot retain an A row in one module while discarding its matched B column or an identically indexed contribution elsewhere. The resulting object is thus a shared-coordinate slice of the stored adapter, not a collection of independently selected module ranks. Figure 1 organizes this workflow as a fixed trained LoRA factorization, checkpoint-derived coordinate scoring, and fixed-budget reconstruction followed by matched evaluation.

3.1 STAGE 1: FIXED TRAINED LORA FACTORIZATION

Let $\mathcal{M}$ denote the set of adapted modules. Let r be the source-adapter rank, and let d_{in} and d_{out} be a module's input and output dimensions, respectively. For $m \in \mathcal{M}$, let $A^{(m)}(t) \in \mathbf{R}^{r \times d_{\text{in}}}$ and $B^{(m)}(t) \in \mathbf{R}^{d_{\text{out}} \times r}$ be the LoRA factors saved at checkpoint step t, where $\mathbf{R}$ denotes the real numbers. Following the LoRA parameterization (Hu et al., 2021), with scaling parameter α, the scaling factor ρ and dense update are defined by Eq. 1:

$$\begin{aligned} \rho &= \tfrac{\alpha}{r}, \\ \Delta W^{(m)}(t) &= \rho B^{(m)}(t) A^{(m)}(t). \end{aligned} \tag{1}$$

We analyze the completed factorization $\mathcal{F}(t) = \{A^{(m)}(t), B^{(m)}(t)\}_{m \in \mathcal{M}}$ rather than modifying its training procedure. All quantities below are read from this stored artifact and, where required, its saved sibling checkpoints. Thus, the procedure does not attribute observed behavior to a new RL objective, a retraining schedule, or an adaptive optimization decision made after evaluation.

Paired coordinates. Write $B^{(m)} = [b_1^{(m)}, \ldots, b_r^{(m)}]$ and let ${a_i^{(m)}}^\top$ be row i of $A^{(m)}$. Equation 2 expands the update by source-factorization coordinate:

$$\Delta W^{(m)}(t) = \rho \sum_{i=1}^{r} b_i^{(m)}(t) {a_i^{(m)}(t)}^\top. \tag{2}$$

The implementation uses one global index across all adapted modules, so paired coordinate i is the bundle in Eq. 3:

$$z_i(t) = \{(b_i^{(m)}(t), a_i^{(m)}(t))\}_{m \in \mathcal{M}}. \tag{3}$$

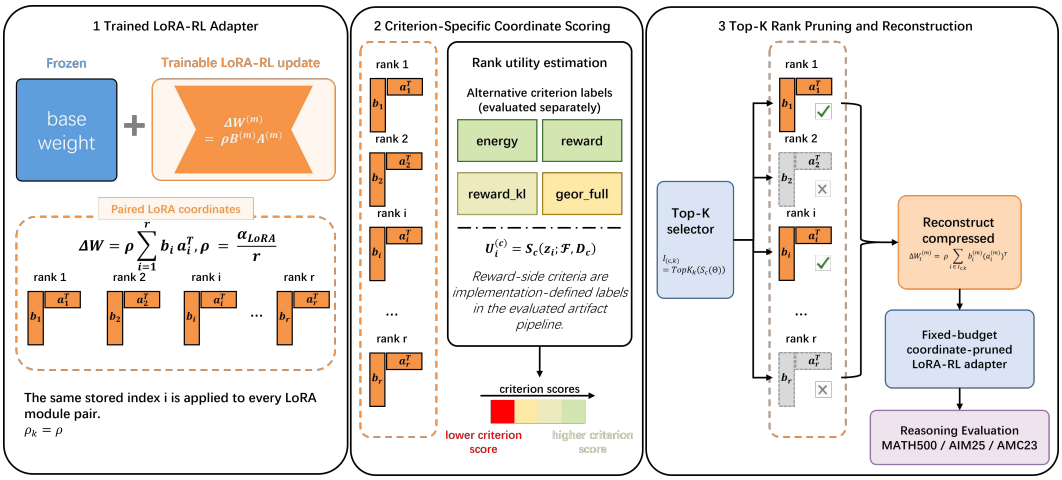

Figure 1: Post-hoc paired-coordinate reduction in a stored LoRA factorization. The rank-i tiles denote shared source-factorization coordinate indices $i \in \{1, \ldots, r\}$, pairing matching A rows and B columns across adapted modules. For each i, $e_i(t)$, $g_i(t)$, $h_i(t)$, and $q_i(t)$ are candidate checkpoint-derived score families; one selection run chooses one criterion c, rather than adding these scores, to rank indices and form a fixed-budget Top-k subset. Scores rank indices only and do not multiply or reweight retained factors, which are sliced consistently and reconstructed before matched evaluation. The schematic retains historical implementation labels in the scoring box for provenance; the formal score families used here are the checkpoint-derived e, g, h, and q quantities defined below. Historical labels should not be read as direct reward or verifier inputs.

Figure 1 depicts this bundle as the rank-i tile; the labels rank 1 through rank r are source-factorization coordinate indices, whereas k below denotes a target retained rank. This object belongs to the saved factorization. Selecting one global index means that every adapted module keeps or removes the same paired coordinate; it is therefore different from independently choosing a per-module rank or singular direction. Since $BA = (BQ)(Q^{-1}A)$ for any invertible $Q \in \mathbf{R}^{r\times r}$, equivalent dense updates can have different factor coordinates. The selected set is accordingly an operational property of the stored A/B representation, not a claim that the corresponding directions are unique in the dense update. We therefore make no coordinate-system-independent claim.

3.2 Stage 2: Checkpoint-Derived Coordinate Scoring

The implementation aggregates one energy per global coordinate as in Eq. 4:

$$E_i(t) = \sum_{m\in\mathcal{M}} \left\| A^{(m)}(t)[i,:] \right\|_2 \left\| B^{(m)}(t)[:,i] \right\|_2 . \tag{4}$$

Let $E(t) = (E_i(t))_{i=1}^r$ denote the resulting length-r energy vector; $\|\cdot\|_2$ is the Euclidean norm. There is no per-module normalization before this sum. For a length-r vector x, global coordinate-wise min–max normalization is defined in Eq. 5:

$$\mathrm{MM}(x)_i = \frac{x_i - \min_{j\in\{1,\ldots,r\}} x_j}{\max_{j\in\{1,\ldots,r\}} x_j - \min_{j\in\{1,\ldots,r\}} x_j}. \tag{5}$$

If the implementation's `math.isclose` test regards the minimum and maximum as equal, MM returns an all-ones vector.

Score families. The four score families deliberately separate a snapshot quantity from temporal changes in the stored factorization. Component energy asks which paired coordinates have large current aggregate magnitude; positive growth asks which have increased relative to the recorded predecessor; the two combined scores introduce fixed penalties or stability adjustments. They are

evaluated as alternative selectors, not estimated jointly from task outcomes. The *component-energy* score is defined in Eq. 6:

$$e_i(t) = \mathrm{MM}(E(t))_i. \tag{6}$$

Let $p(t)$ be the numerically largest available sibling checkpoint below t. If it exists, *positive energy growth* is defined in Eq. 7:

$$g_i(t) = \mathrm{MM}(\max\{0, E_i(t) - E_i(p(t))\}_{i=1}^{r})_i\,. \tag{7}$$

If no lower sibling checkpoint exists, the implementation instead sets $g_i(t) = \mathrm{MM}(E(t))_i$. The canonical M1 energy-penalty coefficient is $\lambda = 0.15$; it is also the frozen E2 default. Equation 8 defines the *growth-minus-energy* score:

$$h_i(t) = g_i(t) - \lambda e_i(t). \tag{8}$$

The recovered stability implementation enumerates eligible checkpoint paths up to t in lexicographic path order and uses the final up to five entries. This implementation-defined window can differ from numerical checkpoint order; numeric-versus-lexicographic comparisons are therefore diagnostics, not chronology or invariance tests. With at least three entries in this path-order window, the population variance is defined in Eq. 9:

$$v_i(t) = \mathrm{Var}_{\mathrm{pop}}\{E_i(u) : u \in \mathcal{L}(t)\}. \tag{9}$$

Equation 10 then defines the normalized stability term:

$$s_i(t) = \mathrm{MM}\left(\frac{1}{\sqrt{v_i(t)} + 10^{-8}}\right)_i. \tag{10}$$

Here $\mathrm{Var}_{\mathrm{pop}}$ denotes population variance, u is a checkpoint step, $\mathcal{L}(t)$ denotes the implementation's path-order window, and 10^{-8} is a numerical stabilizer. With fewer than three entries, $s_i(t) = 1$ for all i. The canonical M1 stability coefficient is $\beta = 0.20$. Equation 11 defines the supporting *growth-energy-stability* score:

$$q_i(t) = g_i(t) - \lambda e_i(t) + \beta s_i(t). \tag{11}$$

For use below, write $e(t) = (e_i(t))_{i=1}^{r}$, $g(t) = (g_i(t))_{i=1}^{r}$, $h(t) = (h_i(t))_{i=1}^{r}$, $s(t) = (s_i(t))_{i=1}^{r}$, and $q(t) = (q_i(t))_{i=1}^{r}$ for the corresponding length-r score vectors. Thus, the rank-i tile $z_i(t)$ is not multiplied by a score: its factors contribute to $E_i(t)$, and current, predecessor, and windowed checkpoint energies yield the scalar candidates $e_i(t)$, $g_i(t)$, $h_i(t)$, and $q_i(t)$.

Each run selects one of $e_i(t)$, $g_i(t)$, $h_i(t)$, or $q_i(t)$ to rank coordinate indices; the scores are not added together and do not reweight factor values. This separation prevents a score from silently becoming a new adapter parameterization: after selection, retained A/B entries are exactly those from the stored adapter. Random subsets use seeded uniform scores, while first-k and last-k follow stored coordinate order and serve only as implementation diagnostics; neither fixed-order control implies a semantic ordering of LoRA coordinates.

3.3 Stage 3: Fixed-Budget Selection, Reconstruction, and Evaluation

For criterion $c \in \{e, g, h, q\}$ and target rank k, Eq. 12 maps coordinate index i to its selected scalar score:

$$S_i^{(c)}(t) = \begin{cases} e_i(t), & c = e, \\ g_i(t), & c = g, \\ h_i(t), & c = h, \\ q_i(t), & c = q. \end{cases} \tag{12}$$

Let $\mathrm{TopKIndices}_k$ return the original coordinate indices of the k largest input scores. Scores are sorted in descending order with Python's stable sort over indices $1, \ldots, r$; equal scores therefore retain the smaller original index first. Equation 13 then selects the retained set:

$$I_{c,k}(t) = \mathrm{TopKIndices}_k\{S_i^{(c)}(t)\}_{i=1}^{r}. \tag{13}$$

Algorithm 1 Fixed-budget paired-coordinate pruning

Require: Current trained adapter at t, available sibling adapters, target rank k, criterion c

Load $A^{(m)}(t), B^{(m)}(t)$ and compute global coordinate energies $E(t)$.
if c uses growth **then**
    Find the largest numerical sibling step $p(t) < t$; compute $g(t)$, or use the no-previous fallback.
end if
if c uses stability **then**
    Select the final up to five eligible paths under lexicographic path order and compute $s(t)$.
end if
Form the criterion score $e(t)$, $g(t)$, $h(t)$, or $q(t)$.
Stable-sort scores descending; retain the first k indices and store them ascending.
Slice matching A rows and B columns for every module.
Set $\alpha_k = \mathrm{round}(\alpha k/r)$ and save the post-hoc adapter.
Evaluate the saved adapter separately under the matched task protocol.

The selected indices are sorted ascending for storage, then sliced identically from every module as in Eq. 14. The stable tie rule is an implementation convention that makes the operation reproducible for equal scores; it does not resolve whether tied coordinates have distinct scientific importance.

$$\begin{aligned} A^{(m)}_{c,k}(t) &= A^{(m)}(t)[I_{c,k}(t), :], \\ B^{(m)}_{c,k}(t) &= B^{(m)}(t)[:, I_{c,k}(t)]. \end{aligned} \tag{14}$$

The implementation sets the reconstruction scaling parameter according to Eq. 15:

$$\alpha_k = \mathrm{round}\left(\alpha \frac{k}{r}\right). \tag{15}$$

For the evaluated source $r = 32$, $\alpha = 128$, and $k \in \{16, 20, 24\}$, this gives $\alpha_k \in \{64, 80, 96\}$ and preserves $\alpha_k/k = 4$ exactly. Writing $\rho_k = \alpha_k/k$ for the reconstructed adapter's scaling factor, the evaluated settings therefore satisfy $\rho_k = \alpha_k/k = \alpha/r = \rho$.

SVD-DeltaW comparator. As a matched post-hoc reconstruction baseline, SVD-DeltaW applies a rank-k truncated singular value decomposition (SVD) independently to each dense update $\Delta W^{(m)}(t)$. It reconstructs module-specific LoRA factors at effective rank k and rescales them to preserve $\alpha_k/k = 4$, but it does not select a shared global coordinate subset or preserve the stored factorization coordinates. Its reported cells come from the independent M1 SVD ledger, not from the paired-coordinate scorer above. Contrasting it with paired selection asks whether retaining original shared coordinates behaves differently from reconstructing each module's dense update independently. Random selection is instead a stochastic selection baseline: it preserves the paired-coordinate object and final budget while removing checkpoint-derived ordering. Neither baseline changes the original RL training trajectory.

4 Experiments

Our experiments separate two questions. Q-A asks how much of a completed LoRA-RL adapter can be removed after training while retaining useful reasoning behavior. Q-B asks how that post-hoc decision differs from deciding adaptation capacity before or during training. The local quantitative evidence addresses Q-A: M1 is a checkpoint-360 endpoint study of one source-rank-$r = 32$ DeepSeek-R1-Distill-Qwen-1.5B adapter, with retained budgets $k \in \{16, 20, 24\}$ and matched post-hoc alternatives. Q-B is addressed only by methodological positioning and external context; no matched direct-rank, AdaLoRA-RL, or GeoRA-RL result is claimed.

4.1 Experimental Setup

Model, source adapter, and tasks. The stored adapter uses DeepSeek-R1-Distill-Qwen-1.5B at checkpoint 360; local records identify a GRPO/GAIR-LIMR recipe (DeepSeek-AI, 2025; Shao

et al., 2024; Wang et al., 2025). This documents local lineage, not identity with an official Tina-LIMR weight, an upstream dataset revision, or an independent training replication. MATH500 (500 problems), AIME25 (30), and AMC23 (40) are evaluation benchmarks, not confirmed RL training data (Hendrycks et al., 2021; Lightman et al., 2023; Art of Problem Solving, n.d.a;n). We report percentage `extractive_match` separately by task.

Compared objects. The unpruned source is the full $r = 32$ reference. Energy and growth-minus-energy (G–E) select one shared paired-coordinate set from the stored factorization. SVD-DeltaW is a matched post-hoc reconstruction baseline: it forms each module's dense ΔW, applies an independent truncated SVD, and refactorizes the result at the same target rank and scale. Random is a selection baseline / stochastic control: it retains a seeded shared subset of the same stored coordinates. First-/last-k remain implementation diagnostics and are not method competitors.

Recorded protocol. For each deterministic method–task–budget cell, the mean and sample standard deviation (SD) summarize only recorded generation seeds (SD denominator $n - 1$). A paired comparison matches task, budget, and available generation seed. Random rows pool pruning-subset and generation-seed cells; these are not ordinary multi-seed SDs or training-seed replications. No matched wall-clock time, GPU hours, energy, or provider-cost record is available, so we make no efficiency or price claim.

4.2 Post-hoc Capacity Reduction

Starting from one completed source-rank-$r = 32$ adapter, we retain $k \in \{16, 20, 24\}$ coordinates only after training and preserve $\alpha_k/k = 4$. The corresponding final LoRA parameter counts are 18,464,768, 23,080,960, and 27,697,152, which are respectively 50.0%, 37.5%, and 25.0% below the $r = 32$ reference. These counts match a hypothetical direct LoRA adapter at the same rank when target modules and factor dimensions are identical, but they do not imply the same training trajectory or training-time capacity.

The Energy and G–E endpoints show useful retention at all three budgets, but the result is task- and budget-dependent. For example, the M1 MATH500 means remain near the full reference at $k = 16, 20, 24$, whereas AIME25 and AMC23 vary more visibly across selectors and budgets. The scientific object here is retained post-hoc capacity, not a universally best selector.

4.3 Comparison with Post-hoc Alternatives

Table 1 is the main quantitative comparison. It makes the roles explicit: Full is the uncompressed reference, SVD is a post-hoc reconstruction baseline, Random is a stochastic selection baseline, and Energy/G–E are the checkpoint-derived coordinate selectors. All local rows start from the same completed adapter and use matched final rank, scale, tasks, and evaluation conventions.

At matched final rank and scaling, no post-hoc alternative systematically dominates across tasks and budgets. SVD is strongest in several cells, while Random approaches or overlaps the checkpoint-derived selectors at the recorded $k = 24$ budget. This is neither a significance test nor a universal ranking. The defensible conclusion is that post-hoc capacity can be redundant, while the identity and advantage of retained stored coordinates remain conditional on the factorization, task, budget, and selector.

Why SVD is a method baseline. SVD-DeltaW operates on the same completed adapter and produces an ordinary rank-k PEFT adapter without retraining, but it changes the reconstruction object: each module receives its own top singular directions of the dense update. Post-hoc paired-coordinate selection instead preserves the stored factorization and applies one shared paired-coordinate set across modules. This is an object-level post-hoc comparison, not merely another score ablation.

4.4 Training-time Versus Post-hoc Capacity

Training-time rank methods answer a related but different question. Fixed LoRA-RL chooses capacity before optimization; AdaLoRA allocates effective rank during optimization using gradient-derived sensitivity and uncertainty; GeoRA changes RLVR initialization and parameterization with

Role / method	Budget	M500	A25	AMC23
Source: Full	$r = 32$	84.50 ± 1.84 [2]	26.67 ± 4.71 [5]	75.50 ± 6.22 [5]
Reconstruction: SVD	$k = 16$	84.84 ± 0.52 [5]	22.00 ± 3.80 [5]	72.00 ± 6.94 [5]
Reconstruction: SVD	$k = 20$	84.40 ± 0.87 [5]	24.67 ± 4.47 [5]	73.50 ± 4.87 [5]
Reconstruction: SVD	$k = 24$	84.10 ± 0.71 [2]	23.33 ± 4.08 [5]	75.00 ± 3.06 [5]
Selection: Random	$k = 24$	84.55 ± 1.21 [45]	22.40 ± 5.00 [50]	72.15 ± 4.63 [50]
Ours (Energy)	$k = 16$	83.96 ± 1.24 [5]	26.00 ± 6.41 [5]	72.50 ± 3.95 [5]
Ours: Energy	$k = 20$	84.00 ± 0.65 [5]	26.00 ± 2.79 [5]	74.50 ± 4.11 [5]
Ours: Energy	$k = 24$	84.60 ± 1.13 [2]	23.33 ± 4.08 [5]	72.50 ± 3.06 [5]
Ours: G–E	$k = 20$	84.04 ± 1.43 [5]	22.67 ± 1.49 [5]	71.50 ± 3.79 [5]
Ours: G–E	$k = 24$	84.50 ± 0.14 [2]	23.33 ± 2.36 [5]	71.00 ± 6.52 [5]

Table 1: Matched post-hoc low-rank alternatives on the completed M1 rank-32 adapter. Values are percentage `extractive_match` mean $\pm$ sample SD [cell n]. Deterministic rows vary recorded generation seeds. Random at $k = 24$ pools pruning-subset $\times$ generation-seed cells (45 for M500; 50 for each extension task), so its SD is not directly comparable to deterministic-row SDs. Random is a selection baseline; SVD-DeltaW is a reconstruction baseline. The locked Energy $k = 16$ row is the sole Ours row at that budget; G–E at $k = 16$ is omitted because the recovered M1 manifest gives the identical selected set and endpoint sequence. No row is a training-seed replication. The different [cell n] values are descriptive summaries; only seed-matched analyses support paired or inferential interpretation, with details in the Supplement.

Method	Intervention stage	Capacity signal	Retraining	Completed adapter	Stored coordinates
Fixed LoRA-RL	Before training	Fixed rank	Required	No	Not applicable
AdaLoRA-RL	During training	Gradient sensitivity / uncertainty	Required	No	No
GeoRA	Initialization + training	Weight geometry / SVD prior	Required	No	No
SVD-DeltaW	After training	Per-module singular directions	None	Yes	No
Post-hoc paired-coordinate	After training	Checkpoint weights / history	None	Yes	Yes

Table 2: Capacity interventions act on different objects and stages. The first three methods choose or allocate capacity before or during optimization. SVD-DeltaW and post-hoc paired-coordinate selection are matched post-hoc alternatives on the same completed adapter, differing in whether they reconstruct module-local singular directions or preserve shared stored coordinates. This is a methodological comparison, not a cross-paper score comparison.

a geometry-aware low-rank structure and frozen residual anchor (Zhang et al., 2023; 2026). Post-hoc paired-coordinate selection operates only after RL training has completed and does not read rewards, verifiers, gradients, or task outcomes during selection. These intervention stages are not interchangeable and are not treated as a score leaderboard.

Published training-time rank studies use different optimization and evaluation protocols and are therefore treated as context rather than matched numerical baselines. The Tina rank panel in the Supplement supports only the statement that training-time rank has been varied in prior LoRA-RL work; it cannot support a claim that local post-hoc $k = 16$ beats direct training-time $r = 16$. A matched direct-rank or training-time adaptive comparison remains open.

4.5 Cross-family Evidence and Runtime Boundaries

The cross-family evidence is asymmetric and we preserve that asymmetry. An independently trained DoRA source-rank-32 adapter (Liu et al., 2024) was stopped exactly at step 90, and its native post-hoc Energy reconstruction at $k = 16$ was evaluated with the same task names and generation seeds under DoRA-native semantics. The source was fixed before benchmark evaluation; downstream scores did not select it, and no step-91 update occurred. Table 3 reports task-dependent transfer separately from the LoRA table: MATH500 and AMC23 are retained while AIME25 decreases, not a cross-family leaderboard or superiority claim.

AdaLoRA training and native post-hoc construction were instantiated successfully, but formal benchmark evaluation could not be executed under the frozen LightEval/VLLM evaluation protocol without changing the validated loading path. Therefore no AdaLoRA benchmark score is reported;

Setting	MATH500	AIME25	AMC23
DoRA source $r = 32$, step 90	84.80 ± 1.51	27.78 ± 5.09	73.33 ± 2.89
Energy $k = 16$	84.93 ± 1.33	22.22 ± 1.92	73.33 ± 2.89
Delta (pp)	$+0.13$	-5.56	0.00

Table 3: Cross-family transfer on an independently trained DoRA-r32 adapter stopped exactly at step 90. The same transferred Energy scoring rule selects one shared physical-coordinate set, and reconstruction uses native DoRA semantics. Values are percentages, mean $\pm$ sample SD over generation seeds $11, 23, 47$ ($n = 3$). This table is not a LoRA–DoRA leaderboard and does not support universal transfer, lossless compression, or superiority claims.

this is a runtime boundary, not a negative performance result. Detailed source, native-artifact, and loader provenance is retained in the Supplement and audit files.

The resulting cross-family conclusion is procedural: the present paper establishes a matched post-hoc LoRA study, a separately audited DoRA transfer test, and an explicit AdaLoRA evaluation boundary. These states must not be collapsed into one leaderboard.

4.6 Validity of Coordinate Selection

The validity analyses constrain how the endpoint table should be read. Q1 finds that checkpoint-derived score order does not stably track leave-one-coordinate-out marginal utility. Q2 shows that stored cross-module coordinate alignment matters: relabeling paired factors changes the retained coordinate set while preserving the full module update. Q3 does not establish a stable Energy $>$ Random advantage; the M2 Energy–Random intervals include zero. Q4 shows that M1 behavior does not straightforwardly persist on M2. Full confidence intervals, permutation seeds, p-values, E1/E2 tables, and the bounded M2 details remain in the Supplement.

These diagnostics are a validity analysis, not a replacement method core. They rule out stronger interpretations of the score-induced orderings while leaving the narrower post-hoc compression finding intact.

Evidence boundary. The paper has a quantitative comparison among matched post-hoc alternatives, but not a matched training-time method comparison. We therefore do not claim that RL needs less rank than SFT, that post-hoc paired-coordinate selection outperforms AdaLoRA or GeoRA, or that the selected coordinates are intrinsic features of the dense update.

5 Conclusion and Future Work

We study post-hoc capacity reduction in one stored LoRA-RL factorization. At matched final budgets, shared-coordinate selection, per-module SVD reconstruction, and random selection produce task- and budget-dependent endpoints; no one post-hoc rule is uniformly preferable. The evidence supports a bounded conclusion: completed LoRA-RL adapters can exhibit substantial post-hoc capacity redundancy, while successful compression does not imply stable, intrinsic, or factorization-independent coordinate importance. The independently trained DoRA-r32 campaign shows that the Energy selector can transfer to another parameterization, with retention on two tasks and a decrease on AIME25; this is task-dependent transfer, not cross-family superiority. AdaLoRA training and native post-hoc construction were instantiated, but formal benchmark evaluation was unavailable under the frozen LightEval/VLLM loading protocol, so no AdaLoRA score is reported. Training-time rank requirements and post-hoc reduction therefore remain distinct questions.

AI Use Statement

We used large language model tools to assist with code and workflow debugging, manuscript language editing, and consistency and audit checks. All reported experiments, numerical results, citations, scientific interpretations, and final manuscript content were reviewed and verified by the authors, who take full responsibility for the work.

REFERENCES

Art of Problem Solving. AIME problems and solutions, n.d.a. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Art of Problem Solving. AMC 12 problems and solutions, n.d.b. URL https://artofproblemsolving.com/wiki/index.php/AMC_12_Problems_and_Solutions.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. *arXiv preprint arXiv:2305.14314*, 2023. URL https://arxiv.org/abs/2305.14314.

Ning Ding, Xingtai Lv, Qiaosen Wang, Yulin Chen, Bowen Zhou, Zhiyuan Liu, and Maosong Sun. Sparse low-rank adaptation of pre-trained language models, 2023.

Clémentine Fourrier, Nathan Habib, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. Lighteval: A lightweight framework for llm evaluation, 2023. URL https://github.com/huggingface/lighteval.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, 07 2024. URL https://zenodo.org/records/12608602.

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. *arXiv preprint arXiv:2103.03874*, 2021. URL https://arxiv.org/abs/2103.03874.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin de Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. Parameter-efficient transfer learning for NLP. *arXiv preprint arXiv:1902.00751*, 2019. URL https://arxiv.org/abs/1902.00751.

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. *arXiv preprint arXiv:2106.09685*, 2021. URL https://arxiv.org/abs/2106.09685.

Vishnuprasadh Kumaravelu, Sunil Gupta, and P. K. Srijith. Post-optimization adaptive rank allocation for LoRA. *arXiv preprint arXiv:2604.27796*, 2026. URL https://arxiv.org/abs/2604.27796.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the ACM Symposium on Operating Systems Principles*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023. URL https://arxiv.org/abs/2305.20050.

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. Dora: Weight-decomposed low-rank adaptation. *arXiv preprint arXiv:2402.09353*, 2024.

Steffen Schotthöfer, Emanuele Zangrando, Gianluca Ceruti, Francesco Tudisco, and Jonas Kusch. GeoLoRA: Geometric integration for parameter efficient fine-tuning. *arXiv preprint arXiv:2410.18720*, 2024. URL https://arxiv.org/abs/2410.18720.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. DyLoRA: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *arXiv preprint arXiv:2210.07558*, 2022. URL https://arxiv.org/abs/2210.07558.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. Tina: Tiny reasoning models via LoRA. *arXiv preprint arXiv:2504.15777*, 2025. URL https://arxiv.org/abs/2504.15777.

Jiaying Zhang, Lei Shi, Jiguo Li, Jun Xu, Jiuchong Gao, Jinghua Hao, and Renqing He. GeoRA: Geometry-aware low-rank adaptation for RLVR. *arXiv preprint arXiv:2601.09361*, 2026. URL https://arxiv.org/abs/2601.09361.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.10512*, 2023. URL https://arxiv.org/abs/2303.10512.

Ruiyi Zhang, Rushi Qiang, Sai Ashish Somayajula, and Pengtao Xie. Autolora: Automatically tuning matrix ranks in low-rank adaptation based on meta learning, 2024.