PROJECTION-LEVEL LORA PLACEMENT: AN EQUAL-PARAMETER AUDIT ACROSS MODELS AND TASKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing which transformer projections receive LoRA adapters changes both parameter budget and training behavior. We audit placement under matched trainable-parameter budgets using Llama-2-7b-chat-hf and Llama-3.1-8B-Instruct, Alpaca and GSM8K, and three seeds per setting. The resulting 36-run matrix compares up-only, down-only and a training-data-based PLoP allocation. Up-only has the lowest mean response NLL in one of four model–task combinations; down-only wins the other three. These comparisons measure teacher-forced response likelihood. We also audit generated numeric answers at all 18 GSM8K adapters and two base checkpoints on separate internal and official-test subsets; prompt compatibility and termination limit interpretation of raw base scores. An earlier 25-configuration sweep favored up-only on reused WikiText-2 validation of candidate choices, but did not match parameter budgets. On the official-test subset, up-minus-down accuracy is +3.57 percentage points for Llama-2 (pointwise 95% interval [1.30,5.87]) and -0.50 for Llama-3 ([-3.00,1.97]). Historical and matched protocols differ in objective and holdout construction, so their ordering changes are not a placement-only intervention. Our results support task- and metric-dependent evaluation of placement, not a universally best projection or a new module-selection algorithm.

1 INTRODUCTION

LoRA (Hu et al., 2022) makes large-language-model fine-tuning practical by restricting each weight update to a low-rank factorization. Its performance depends not only on rank and parameterization, but also on where adapters are placed. PLoP (Hayou et al., 2026) directly studies projection-level placement with a data-aware signal, while AlphaLoRA (Qing et al., 2024) uses HT-SR statistics to allocate LoRA-MoE experts across depth. Recent safety-alignment experiments also report strong low-rank `up_proj` results (Xue & Mirzasoleiman, 2025). These studies make target placement an active design question rather than an unexplored one.

Our primary evidence is the frozen 36-run, equal-parameter comparison on two models and two tasks. It tests up-only, down-only and the same training-derived PLoP rule across three seeds, retaining all runs. Earlier sweeps are reported as historical exploration: their held-out metric was reused for configuration selection, and their parameter budgets differed. The historical spectral slope is descriptive and is not a standard HT-SR exponent or a validated selection rule.

Contributions. We provide (i) the complete equal-budget cross-model, cross-task comparison, (ii) an explicit account of changes in objective and evaluation protocol, and (iii) reproducible, limited placement conclusions rather than a universal up-only recommendation. The primary endpoint remains response NLL; the separate saved-checkpoint generation audit measures answer correctness and termination explicitly.

2 BACKGROUND AND RELATED WORK

LoRA and its variants. For a pretrained weight $W_0 \in \mathbb{R}^{d\times k}$, LoRA (Hu et al., 2022) writes the fine-tuning update as $\Delta W = BA$ with $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, $r \ll \min(d,k)$. The original paper adapted only `q_proj` and `v_proj`; a common instruction-tuning recipe adapts all seven projections. AdaLoRA (Zhang et al., 2023) adjusts per-layer rank during training, DoRA (Liu et al.,

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.

Table 1: Actual PLoP allocations in the matched matrix. Each chosen projection type is adapted in all 32 layers. Up-only and down-only use rank 4 throughout; the PLoP arm uses the integer ranks below and $\alpha/r = 2$. Counts refer to projection instances.

Model/task	Selected types	PLoP rank counts	Parameters
Llama-2 / alpaca	v, down, up	37 at 1, 59 at 2	1,933,312
Llama-2 / gsm	v, down, o	10 at 1, 86 at 2	1,933,312
Llama-3 / alpaca	v, o, down	68 at 2, 28 at 3	2,359,296
Llama-3 / gsm	o, v, down	68 at 2, 28 at 3	2,359,296

2024) separates magnitude and direction, and PiSSA (Meng et al., 2024) initializes the low-rank factors from principal components. PLoP (Hayou et al., 2026) is the closest work to ours: it treats projection-type placement directly and ranks candidates using a data-aware normalized feature norm (NFN). We reproduce that signal in §A. Xue and Mirzasoleiman (Xue & Mirzasoleiman, 2025) independently find that low-rank `up_proj` adaptation is effective for safety alignment of reasoning models. The following PLoP comparison adapts ranks to a common budget rather than reproducing all original hyperparameters. Our study is not a first or universal `up_proj` finding; its matched comparisons instead delimit which model, objective, and evaluation regime favor a placement.

Spectral statistics and LoRA allocation. HT-SR (Martin & Mahoney, 2021; 2020) analyzes the empirical spectral density of $\lambda_i = \sigma_i^2$ and conventionally uses a positive exponent in $p(\lambda) \propto \lambda^{-\alpha_{\mathrm{pl}}}$, with smaller exponents indicating heavier tails. AlphaLoRA (Qing et al., 2024) previously uses a standard positive HT-SR/Hill statistic as a training-free signal for allocating LoRA-MoE expert counts across transformer depths. Our experiment concerns ordinary, non-MoE LoRA placement across projection types. More importantly, our implementation does not estimate the standard HT-SR exponent: it fits a log-rank/log-singular-value slope, defined precisely in §C. We therefore use it only as a descriptive spectral diagnostic.

Knowledge localization in MLP. ROME (Meng et al., 2022) and MEMIT (Meng et al., 2023) show that factual associations in decoder-only transformers can be edited through specific MLP weights. This motivates hypotheses that MLPs store associative information while attention participates in routing. Our experiments do not include the activation-level interventions needed to establish that mechanism, so we treat this connection as an interpretation to test rather than an explanation of our results.

3 Primary matched-budget cross-model experiment

We train Llama-2-7b-chat-hf and Llama-3.1-8B-Instruct on Alpaca and GSM8K, with up-only, down-only, and an official-NFN-selected PLoP placement under three paired seeds (36 runs). The PLoP implementation selects the three lowest module scores from 100 training prompts, with integer ranks adapted to our common parameter budget. This is a matched-budget adaptation, not a claim to reproduce the original PLoP hyperparameters. The PLoP arm adapts only the published training-derived module score and its lowest-three selection rule to our budget; it does not reproduce the original method's complete tuning recipe. The trainable budgets are 1,933,312 and 2,359,296 parameters for Llama-2 and Llama-3, respectively. Actual selected modules and integer ranks are recorded in each frozen configuration. Each arm uses the same frozen split, three epochs, logical batch 32, AdamW learning rate 2×10^{-4}, weight decay 0.01, and dropout 0.05. There are 10,000 Alpaca or 6,000 GSM8K training examples and 1,000 held-out examples per task. Each model/task/seed comparison runs on the same GPU and software environment.

3.1 What the budget adaptation changes

The normalized feature norm is averaged by projection type over the 100 training-only prompts and all layers. The lowest three types define the PLoP subset. The budget adaptation then assigns one of two adjacent integer ranks to each selected projection instance; a deterministic name-hash ordering and subset-sum rule reach the exact target count. It does not optimize those ranks against held-out

Table 2: New 36-run matrix: mean held-out response-token NLL over three seeds. Lower is better. These are teacher-forced losses, not generated-answer accuracy.

Model / task	Up	Down	PLoP
llama2 / alpaca	1.16730	1.14480	1.16409
llama2 / gsm	0.42406	0.43166	0.42749
llama3 / alpaca	1.31077	1.28407	1.28741
llama3 / gsm	0.42913	0.42481	0.42544

loss. All arms keep $\alpha/r = 2$, so this implementation avoids changing the explicit adapter multiplier while matching parameter count. It still compares different update subspaces and factor shapes.

The selected types themselves vary with the model and training task (Table 1). In particular, Llama-2/Alpaca includes both MLP directions, while Llama-2/GSM8K replaces up with the attention output projection. This makes the data-aware allocation a distinct competitor to each single-type arm, rather than a renamed up/down mixture. The evidence concerns this declared adaptation of PLoP; it does not compare the best achievable performance under separately tuned selection and rank budgets.

3.2 Training and held-out scoring contract

The primary runs use response-only training loss. The saved input serialization contains Instruction, optional Input, and Response delimiters; the prompt has BOS plus at most 255 prompt tokens. The reference response and EOS are then appended, and the combined sequence is right-truncated to 512 tokens. Prompt and padding labels are masked. Training batches are right-padded; each logical 32-example gradient aggregates two-example microbatches in proportion to their scored response-token counts. Global gradient clipping uses threshold one. Three epochs give 939 updates on Alpaca and 564 on GSM8K. This accounting matters because equal numbers of examples need not provide equal numbers of scored response tokens.

Evaluation uses the same retained tokenization and target masking, with loss sums and response-token counts saved separately for all 1,000 cases. The held-out NLL is their total ratio, not an average of per-example means. Seeds 21604–21606 are set before adapter creation. Within each model/task, the frozen base checkpoint, data split, logical batch order rule and training schedule are shared; differing adapter shapes do not make every stochastic operation identical across placements. The recorded per-seed results retain this optimization variability instead of treating tokens as independent runs.

The primary endpoint is response-token NLL, and three seeds provide a sensitivity check rather than evidence of equivalence. A common training schedule controls the protocol but does not establish that every placement is optimally tuned.

Up-only has the lowest mean in one of four combinations (LLaMA-2/GSM8K); down-only wins the other three. Thus the earlier WikiText-2 ordering does not establish a generally preferred placement. The audit reconstructs every NLL from saved per-example loss sums and response-token counts, uses all model–task–seed cells under equal parameter budgets. This NLL reconstruction does not itself reevaluate checkpoints; the separate generation audit in §4 does. Three seeds provide a sensitivity check, not a broad statistical guarantee; full per-seed values and paired differences are retained in the accompanying analysis JSON.

Magnitude and task dependence. Down-only lowers mean response NLL by 0.02250 and 0.02670 nats on the two Alpaca settings. On GSM8K, up-only is lower by 0.00760 on Llama-2, whereas down-only is lower by 0.00432 on Llama-3. These paired, within-setting contrasts are the interpretable quantities; different tokenizers and targets prevent pooling absolute NLL across models or tasks.

3.3 Paired uncertainty and case-level heterogeneity

The aggregate loss can conceal different orderings across examples. We therefore reanalyze the retained per-example loss sums and response-token counts for all 36 runs. For each model and task, examples are aligned by identifier across placements and seeds; every aligned record has the same positive token count. This checks the denominator as well as the numerator of the paired comparison.

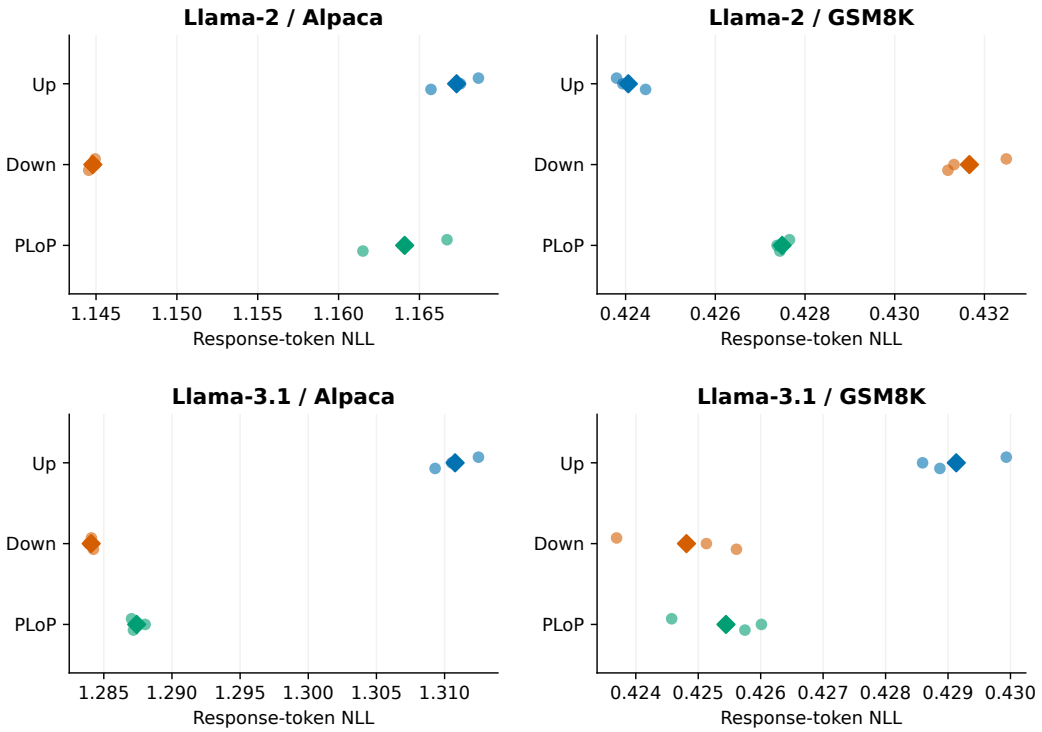

Figure 1: Complete 36-run matched-parameter placement comparison. Small points retain all three seeds per placement; diamonds show mean response-token NLL. PLoP ranks are adapted to the common budget. Axes are task-specific; these teacher-forced losses are not generated-answer accuracy or a proof of a universal placement ranking.

Table 3: Retrospective uncertainty for the primary up-minus-down response NLL. The token-weighted ratio is recomputed in 10,000 paired question-and-seed bootstrap draws. The last column counts cases with lower up-only NLL after averaging over the three seeds; it is a different, equally weighted descriptive endpoint.

Model / task	Mean (nats)	95% interval	Up lower / 1,000
Llama-2 / alpaca	+0.02250	$[+0.01957, +0.02551]$	338
Llama-2 / gsm	-0.00760	$[-0.00942, -0.00570]$	631
Llama-3 / alpaca	+0.02670	$[+0.02332, +0.03022]$	316
Llama-3 / gsm	+0.00432	$[+0.00243, +0.00635]$	423

Let $\ell_{s,a,i}$ be the summed response loss for seed s, placement a, and case i, and let n_i be its scored token count. The reported contrast is

$$\widehat{\Delta}_{a,b} = \frac{1}{3}\sum_{s=1}^{3} \frac{\sum_i (\ell_{s,a,i} - \ell_{s,b,i})}{\sum_i n_i}.$$

In each of 10,000 bootstrap draws, we resample cases and the three paired training seeds with replacement, using the same sampled indices for both placements. The token-weighted ratio is recomputed inside the resample. This preserves the shared-example comparison and does not treat individual tokens as independent experimental runs. The resulting intervals are retrospective, pointwise descriptions conditional on these datasets, checkpoints and training seeds; they are not simultaneous intervals over all placement comparisons or a measure of variation across architectures.

All four up-minus-down intervals in Table 3 retain the sign of their corresponding mean. That stability is specific to the saved response-NLL evaluation. The last column answers a different question: it counts examples on which the seed-averaged per-token loss favors up, giving each example equal

Table 4: Generated numeric-answer accuracy (%), three-seed adapter means. Base is one unchanged checkpoint, not three independent replicas. Cap-hit rates must accompany the raw base score; the Llama-3 base exhibits repeated answers under this training-format prompt.

Split	Model	Base	Up	Down	PLoP
Internal train holdout	llama2	12.60	32.87	32.67	33.00
Internal train holdout	llama3	92.60	77.37	79.27	77.87
Official test subset	llama2	12.40	32.53	28.97	30.40
Official test subset	llama3	82.70	72.97	73.47	73.17

weight. These counts need not agree with the ordering of a token-weighted aggregate, because response lengths and effect magnitudes vary. Neither endpoint identifies a placement that is uniformly better for every input. The supplement retains all three pairwise contrasts, the per-seed differences and the aligned arrays used for this calculation.

4 Generated answers under the saved prompt contract

We additionally evaluate the 18 final GSM8K adapters and two unchanged base models, without training or selecting checkpoints. Each uses greedy decoding, batch eight, a 512-token output cap, the original training-format instruction prompt, and BOS plus the first 255 prompt tokens. The unchanged numeric parser extracts the final marked numeric answer. This comparison controls the input format across placements; it does not optimize each base model's chat template.

We distinguish two 1,000-question sets. The original held-out set comes from the official GSM8K *training* split, disjoint from our 6,000 training questions. An additional subset of the official 1,319-question test split is selected by normalized-question SHA256 order, excluding all own train, internal-evaluation and smoke questions. All 1,000 question IDs are frozen before these additional evaluations. This is a retrospective benchmark extension, not a claim that the test data were historically unseen by the project or absent from model pretraining. There is no question overlap between these own splits. The two evaluations are reported separately.

Paired uncertainty and metric dependence. The uncertainty analysis resamples the same questions across arms and the three paired training seeds, using 10,000 joint bootstrap replicates. The base checkpoint is repeated only algebraically for paired differences. These are exploratory pointwise intervals, not simultaneous or broad generalization guarantees. On the official-test subset, the up-minus-down accuracy difference is +3.57 percentage points for llama2 (95% interval [+1.30, +5.87]) and -0.50 percentage points for llama3 (95% interval [-3.00, +1.97]). Full per-seed scores and all arm comparisons are retained in the artifact. The response-NLL ordering and generated-answer ordering are distinct endpoints.

Interpreting the paired endpoints. On Llama-2, the GSM8K likelihood and official-test answer endpoints both favor up over down, whereas Alpaca likelihood favors down. On Llama-3, the mean likelihood endpoint favors down on both tasks, but its official-test answer difference is uncertain. Figure 2 consequently distinguishes a model/task-specific ordering from a confidence statement about generated answers. It also shows why the larger Alpaca NLL gaps should not be converted into percentage-point accuracy gains: they concern different targets and evaluation examples.

Termination and prompt compatibility. The initial evaluation passed an incomplete EOS list for Llama-3. We audit all 20,000 saved token sequences against the full native EOS configuration and repeat 16 engineering queries with native stopping. All prefixes agree exactly, and none of the saved sequences changes: the EOS omission does not explain the repetition. The official-test extension uses the full native EOS list. Every selected checkpoint is evaluated on the same question subset without outcome-based checkpoint selection. On the internal holdout, the Llama-3 base reaches the output cap on 99.9% of questions; the fraction both correct and terminated is 0.1%. On the official-test subset, the Llama-3 base reaches the output cap on 99.9% of questions; the fraction both correct and terminated is 0.0%. Correct-and-terminated is a retrospective operational diagnostic,

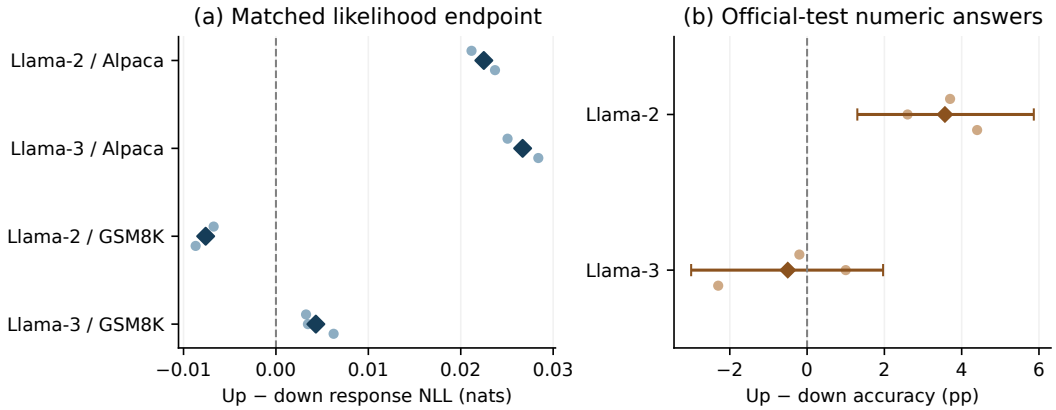

Figure 2: Paired up-minus-down contrasts from completed experiments. Left: response NLL, with all three seed differences and their mean; negative favors up. Right: official-test numeric-answer accuracy, with three seed differences, means and the saved pointwise 95% question-and-seed bootstrap intervals; positive favors up. The axes have different units and favorable directions. No interval is added to the three-seed NLL summaries.

Table 5: New held-out response-only NLL (lower is better). Seeds are 20260911/12/13; the last column is mean ± sample SD. Only the two rank-4 rows are parameter-matched.

Configuration	Seed 11	Seed 12	Seed 13	Mean ± SD
`up_proj`, rank 4	1.092936	1.093358	1.093561	1.093285 ± 0.000319
`down_proj`, rank 4	1.094704	1.094650	1.094828	1.094728 ± 0.000091
All modules, rank 16	1.147376	1.150784	1.151647	1.149936 ± 0.002259

not a replacement benchmark score. Repeated or capped base answers and the shared non-native prompt preclude interpreting raw base–adapter differences as intrinsic model capability or a clean training-benefit estimate. We retain all raw scores, missing-answer rates, repeated-prompt markers and truncation rates.

The prompt-cap audit records llama2: 0, llama3: 0 truncated official-test inputs (out of 1,000 each); no internal-holdout prompt is truncated. Any input truncation remains part of this fixed protocol.

Exploratory comparisons. The earlier WikiText-2 sweep reused its evaluation metric for selecting configurations and changed parameter budgets. A separate response-only holdout found a small up/down gap under its own objective. These experiments are reported separately in §5 and Appendix A; neither supplies additional independent replicates of the primary matched matrix.

5 A SEPARATE TRAINING-OBJECTIVE COMPARISON

We train `up_proj`/rank 4, `down_proj`/rank 4, and all-module/rank 16 with three fixed seeds and score 1,000 prompt-disjoint Alpaca records selected before evaluation (Section 5.1). The two rank-4 choices each have 1,933,312 trainable parameters, while all-module rank 16 has 39,976,960. All runs finish 939 optimizer steps and score the same 61,283 response tokens. One case is truncated at length 512; no case has zero scored response tokens. Training scores the full formatted sequence, whereas this evaluation masks prompt tokens and aggregates only response next-token losses.

Table 5 gives every seed. The paired up-minus-down NLL differences are -0.001769, -0.001292, and -0.001268, with mean -0.001443: about 0.13% of the down-only mean. This is a small observed difference, not a large practical advantage or a significance claim. The larger all-module gap still changes rank and budget. This holdout adds evidence under a frozen same-dataset protocol; it neither removes the original WikiText selection bias nor establishes downstream instruction quality, robustness, or cross-model transfer.

5.1 HOLDOUT CONSTRUCTION AND RESPONSE SCORING

This comparison fixes three configurations (`up_proj`/rank 4, `down_proj`/rank 4, and all seven projections/rank 16) and three new seeds 20260911/20260912/20260913 before examining their evaluation scores. Each uses the same first 10,000 Alpaca training records and the schedule in Section D: three epochs, effective batch size 32, maximum length 512, and peak learning rate 2×10^{-4}. The implementation completes 939 optimizer steps. Seeds are set before adapter initialization. Preliminary attempts that set the seed only after adapter construction are retained as diagnostics and excluded, as are short smoke tests.

We select the first 1,000 eligible records from the remaining Alpaca records in source order, excluding any instruction-plus-input key already present in the training or selected held-out records. Keys use case folding and collapsed whitespace. The selected indices are fixed before scoring. This is a same-dataset prompt-disjoint holdout, not an external benchmark, a semantic near-duplicate audit, or a guarantee against base-model pretraining overlap. It does not undo selection on the historical WikiText-2 evaluation.

Training uses the full formatted prompt-plus-response language-model objective. The response-only evaluation instead measures response-only NLL using the same Alpaca formatting. A fast tokenizer supplies character offsets: only tokens starting at or after the response boundary and having nonempty spans are scored; boundary-straddling tokens, prompt tokens, and padding are excluded. Sequences are right-truncated to 512 tokens and right-padded within batches of four. The causal next-token shift is applied before computing cross-entropy. We retain per-case summed losses and effective token counts, and divide their totals to obtain token-weighted NLL. Truncated and zero-response cases are reported rather than silently removed. All configurations use identical cases and denominators.

The two single-projection rank-4 configurations have exactly equal trainable parameter counts; the all-module rank-16 reference has a larger budget and is not an equal-parameter placement comparison. All seeds are retained without held-out-based tuning. Reported variability is the sample standard deviation over the three seeds, not evidence of significance or cross-model transfer.

6 PARAMETER MATCHING AND EVALUATION CONTRACTS

6.1 WHY EQUAL RANK IS NOT EQUAL BUDGET

For a weight of shape $d_j \times k_j$, a rank-r_j adapter has $r_j(d_j + k_j)$ trainable entries when both factors are trained. For a selected set S of projections across layers, the exact budget is

$$B(S, r) = \sum_{j \in S} r_j(d_j + k_j).$$

Giving every projection the same rank changes this sum when the selected set changes. Even at one fixed model, an all-projection rank-16 configuration and an up-only rank-4 configuration change placement, rank and total capacity together. Their performance difference is valid as a comparison of configurations, but it cannot isolate placement at a matched budget. The primary matrix instead fixes the total number of trainable parameters separately within each model and adapts the selected integer ranks to that budget.

Up and down projections have transposed dimensions within the same MLP, so their single-projection parameter counts match at equal rank. This makes them a direct paired comparison under the common schedule. A mixed PLoP subset can include projections of other shapes and requires counting actual trainable tensors rather than matching nominal rank. The two model-specific budgets also differ: equal parameters within Llama-2 does not mean equal parameters between Llama-2 and Llama-3. The cross-model result concerns the ordering under each matched experiment, not a head-to-head comparison at identical architecture or compute.

Matching this count does not match optimizer dynamics. The factorization scale, update directions, activation statistics and conditioning of the selected projections can differ. A single learning rate and training duration hold the recipe fixed; they do not establish that every arm is optimally tuned. Conversely, tuning one arm more extensively would add a search-budget difference. A follow-up that targets the best achievable placement should first define a common tuning budget, select using

Table 6: Evidence units and what is controlled. Each row retains its own objective, data split and denominator.

Evidence	Controlled comparison	Main boundary
36-run matrix	Parameter count, split, schedule, paired seeds	Response NLL under one recipe
Saved-adapter generation	Prompt, parser and stopping within each audit	Accuracy under the saved training format
Historical sweeps	Shared recipe across selected configurations	Reused selection metric and unequal budgets
Native-EOS check	Retained checkpoint and questions	Stopping sensitivity only

training-side validation, and score a separate untouched test. That follow-up is not part of the completed matrix.

6.2 Likelihood and generated-answer endpoints

For response loss sum s_i and scored-token count n_i, the primary endpoint is $L = \sum_i s_i / \sum_i n_i$. This weights tokens rather than examples and requires identical prompt masking, causal shift, truncation and padding within each comparison. The saved numerators and denominators reproduce every table entry. Teacher-forced likelihood and greedy numeric-answer accuracy measure different behavior, so neither replaces the other.

The generated-answer contrast averages question-level up-minus-down correctness over three paired seeds. The bootstrap draws common questions and common seed indices across arms, retaining both sources of observed variation. It is a pointwise sensitivity analysis conditional on these models, prompts and three seeds. An unchanged base model has no adapter-training seed: repeating its correctness vector in paired arithmetic does not create independent replicas.

A native chat-template evaluation would test a different inference contract at the same checkpoint; training with that template would be a separate intervention. Neither has been completed here. The 99.9% base-model cap-hit rate therefore remains a concrete limitation of the saved-format comparison, rather than evidence about intrinsic model capability or adapter training gain.

7 Discussion

What the spectral result does and does not show. The fitted slopes separate attention and MLP projection types on this LLaMA-2-7B checkpoint. They do not identify a standard heavy-tail cluster, provide a formal link to adaptation quality, distinguish `up_proj` from `down_proj`, or choose rank 4. The latter two choices are empirical outcomes of the module and rank ablations. Accordingly, the slope is best viewed as a descriptive screening diagnostic rather than a target-selection method.

Task dependence. PLoP (Hayou et al., 2026) reports that preferred module subsets vary across architectures, and Xue and Mirzasoleiman (Xue & Mirzasoleiman, 2025) study `up_proj` in a different safety-alignment regime. Our own WikiText-2 and GSM8K orderings differ. Taken together, this evidence argues against an architecture-independent recommendation. The useful conclusion is that target placement can materially change the parameter/PPL trade-off and should be validated against the deployment metric.

Relation to prior placement work. PLoP is the closest prior target-placement method, while AlphaLoRA (Qing et al., 2024) establishes prior use of HT-SR for LoRA capacity allocation. Our primary contribution is a complete matched-budget comparison across two models and two tasks, accompanied by a saved-checkpoint generation audit. The historical sweeps provide context; the PLoP arm adapts the published selection signal to a common budget and is not an exact reproduction of the original training recipe.

Limitations and future work. The original sweep uses LLaMA-2-7B/Alpaca and primarily WikiText-2 PPL. The separate follow-up (§3) adds LLaMA-3 and two held-out response-NLL

tasks, and contradicts a universal up-only preference. It does not establish that `up_proj`-only transfers to Mistral, Qwen, larger scales, QLoRA, long-context retrieval, or other instruction datasets. The historical generated-answer GSM8K comparison used one trained seed per configuration and remains descriptive. The current matched matrix and generated-answer audit use three seeds per placement. The latter adds an official-test subset, while the original 1,000-question GSM8K holdout is drawn from the official training split. Neither removes possible pretraining exposure. The shared schedule controls comparisons but does not optimize each placement separately. The historical PLoP subset is one seed and is not parameter-matched to either Uniform or `up_proj`-only. Finally, the spectral fitting protocol is descriptive and should be replaced by a standard estimator before making any HT-SR claim. Broader architectures, datasets, metrics, and placement-specific hyperparameter sweeps are required for general recommendations.

Protocol distinction. The matched-budget matrix trains on response tokens only, whereas the historical runs used full prompt-plus-response loss. Holdout construction also changed. Therefore an old-to-new rank reversal is not a controlled intervention on the model or adapter placement. The actual checkpoints are Llama-2-7b-chat-hf and Llama-3.1-8B-Instruct.

7.1 Seed Sensitivity under a Shared Training Schedule

We repeated the three central configurations with seeds {42, 2027, 2028}, keeping the data, optimizer, and evaluation code fixed. Table 7 reports per-seed PPL rounded to three decimals. The seed-42 values are retained from the archived experiment summary; full-precision evaluation JSON is available for the two additional seeds. Thus this audit checks aggregation but does not independently rerun all three seeds. The main comparison replicates descriptively: the worst `up_proj`-only seed (10.934) is about 2.24 PPL below the best Uniform seed (13.172). Attention-only and Uniform are close at rank 16, which does not establish that attention adaptation is intrinsically harmful.

Table 7: Three-seed WikiText-2 PPL (mean $\pm$ sample standard deviation).

Configuration	Seed 42	Seed 2027	Seed 2028	Mean $\pm$ std
Uniform r=16	13.351	13.172	13.438	13.320 ± 0.136
Attention-only r=16	13.246	13.285	13.285	13.272 ± 0.023
`up_proj`-only r=4	10.934	10.776	10.808	$\mathbf{10.840 \pm 0.083}$

Reading repeated-seed evidence. Repeating the same selected configuration tests sensitivity to initialization and training order under the retained data and recipe. It cannot turn a metric used during configuration selection into an untouched test. The historical seed contrast is consequently useful as a stability check of that configuration comparison, while the parameter-matched matrix addresses a different question. The two should retain separate denominators, budgets and uncertainty summaries. Averaging them would confound the protocol differences with seed variability.

In particular, small seed variation within the historical sweep does not resolve its unequal-budget comparison or validate the reused selection metric.

Joint resampling preserves the shared seeds and evaluation examples across placements. The resulting intervals describe these trained models and evaluation sets; they do not include uncertainty from changing the training objective, hyperparameter search or deployment distribution.

8 Conclusion

The full matched-budget matrix finds up-only best in only one of four model–task comparisons. Placement therefore needs evaluation for the target model, data and metric. Historical unmatched and selection-reused results cannot establish a general up-only advantage. The PLoP comparison is a declared equal-budget adaptation, and NLL is not a substitute for generated-answer accuracy. We retain the old experiments as exploratory context rather than combine their scores with the new protocol.

Reproducibility Statement

The manuscript and supplementary protocols describe the target modules, parameter budgets, paired seeds, data partitioning, and response-token scoring used in the controlled matrix. Retained training and analysis records preserve the complete model–task comparisons. Historical spectral fits and test-reusing comparisons are labeled separately from the held-out evaluation. The saved controls establish reproducibility of those specified comparisons, not a universal module-selection rule; native-EOS and prompt-termination diagnostics retain their stated limitations.

AI Use Statement

OpenAI Codex was used to aid and polish manuscript writing and to assist with literature retrieval and discovery, including finding related work. The authors are responsible for the final manuscript and citations.

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. PLoP: Precise LoRA placement for efficient finetuning of large models. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/6ace8b8873adf5af379e8b5e0dcd8664-Abstract-Conference.html.

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 *International Conference on Learning Representations (ICLR)*, 2022.

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. In *International Conference on Machine Learning (ICML)*, 2024.

Charles H. Martin and Michael W. Mahoney. Heavy-tailed universality predicts trends in test accuracies for very large pre-trained deep neural networks. In *Proceedings of the SIAM International Conference on Data Mining*, 2020. doi: 10.1137/1.9781611976236.57. URL https://epubs.siam.org/doi/10.1137/1.9781611976236.57.

Charles H. Martin and Michael W. Mahoney. Implicit self-regularization in deep neural networks: Evidence from random matrix theory and implications for learning. *Journal of Machine Learning Research*, 22(165): 1–73, 2021.

Fanxu Meng, Zhaohui Wang, and Muhan Zhang. PiSSA: Principal singular values and singular vectors adaptation of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations (ICLR)*, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations (ICLR)*, 2017.

Peijun Qing, Chongyang Gao, Yefan Zhou, Xingjian Diao, Yaoqing Yang, and Soroush Vosoughi. AlphaLoRA: Assigning LoRA experts based on layer training quality. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20511–20523, 2024. doi: 10.18653/v1/2024.emnlp-main.1141. URL https://aclanthology.org/2024.emnlp-main.1141/.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Yihao Xue and Baharan Mirzasoleiman. LoRA is all you need for safety alignment of reasoning LLMs. *arXiv preprint arXiv:2507.17075*, 2025.

Qingru Zhang, Minshuo Chen, Alexander Bukharin, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=lq62uWRJjiY.

Table 8: Mean per-module ranks for the two slope-weighted allocations at total budget $B = 3584$ ($= 16 \times 224$ modules). The labels are retained from the original sweep.

	q	k	v	o	gate	up	down
Prop	26.3	25.7	22.5	22.4	7.2	3.8	4.1
Uniform	16	16	16	16	16	16	16
Inverse	3.6	3.9	8.2	8.4	26.5	30.8	30.7

Table 9: GSM8K accuracy on the full 1319-example test set with greedy decoding. Uniform and `up_proj`-only reverse their PPL ordering. We report this comparison descriptively.

Method	**Params (M)**	**PPL ↓**	**GSM8K (%) ↑**	**correct / 1319**
Uniform LoRA r=16	40.0	13.35	**7.66**	101
Attention-only r=16	16.8	13.25	6.82	90
MLP-only r=16 (all 3 MLP)	23.2	11.90	5.91	78
`up_proj`-only r=4	1.9	**10.93**	5.00	66

A Same-Schedule Comparison with PLoP

We reproduce PLoP's public NFN calculation (Hayou et al., 2026) on the same LLaMA-2-7B base model using 100 randomly sampled Alpaca examples (seed 2026, maximum length 256). From lowest to highest, the module-type NFN scores are v 1.003, down 1.069, up 1.194, o 1.315, gate 1.447, q 2.516, and k 2.606. Selecting the lowest three gives {`v_proj`, `down_proj`, `up_proj`}. Training this subset with the same optimizer, data, epochs, and per-module rank as our sweep gives PPL 12.749 with 19.66M trainable parameters in one seed. It improves over Uniform r=16's three-seed mean of 13.320, but does not match `up_proj`-only r=4's mean of 10.840. This is a same-schedule comparison, not an equal-parameter or multi-seed comparison. PLoP is task-aware and uses a forward pass; the slope diagnostic in §C is data-free but is not a complete selector.

Rank trends in the original sweep. Within the Uniform family, PPL increases from 12.54 at rank 4 to 13.67 at rank 20. Attention-only likewise has higher PPL at rank 32 than rank 16. These are observations under one shared optimization schedule; they do not establish that attention adaptation is intrinsically harmful or that every placement has been individually tuned.

Slope-weighted allocations. The Prop allocation has higher PPL than Uniform at each reported nominal rank, while Inverse has lower PPL. Table 8 shows that the two allocations put capacity on different module groups. This comparison is part of the empirical sweep, but it does not show that the fitted slope alone selects the best module subset: neither allocation identifies `up_proj`-only, and both are outperformed by configurations found through explicit ablation.

MLP-only and the parameter-efficiency frontier. In the original sweep, MLP-only LoRA has lower PPL than every Uniform or slope-weighted row. The supplementary single-projection and rank ablations further locate the lowest-PPL tested configuration.

B Metric-Dependent Reversal on GSM8K

To check whether the PPL ordering transfers to a reasoning benchmark, we evaluate four configurations on all 1319 GSM8K test examples. We use greedy decoding (temperature 0, `max_new_tokens=256`) with the prompt "Question: {q}\n\nLet's solve this step by step.\n\nAnswer:". Answer extraction follows the `####` convention, using the final number after `####` if present and otherwise the last numeric literal in the generation. Table 9 reports the result.

Uniform obtains 101 correct answers (7.66%), while `up_proj`-only obtains 66 (5.00%). The ordering is opposite to WikiText-2 PPL. Because the models are evaluated on the same questions and the retained aggregate results do not provide the paired correctness table required for an exact paired

Table 10: Fitted spectral-decay slope s at selected LLaMA-2-7B layer indices. Attention modules are in the left four columns and MLP modules in the right three. These values are not HT-SR exponents.

Layer	q	k	v	o	gate	up	down
0	-0.71	-0.69	-1.30	-1.37	-2.43	-2.50	-3.02
8	-1.45	-1.47	-1.76	-1.78	-3.14	-3.62	-3.21
16	-1.52	-1.55	-1.92	-1.88	-3.20	-3.59	-3.79
24	-1.61	-1.62	-2.04	-2.17	-3.44	-3.67	-3.73
31	-1.67	-1.68	-2.06	-2.03	-3.41	-3.44	-3.59

test, we make no new significance claim. One reasoning benchmark is also insufficient to establish a general perplexity–reasoning trade-off. The supported statement is narrower: the PPL-best placement is not the GSM8K-best placement among these four configurations.

C Descriptive Spectral Analysis

What was computed. For every linear module ℓ in the pretrained LLaMA-2-7B, we sort its singular values $\sigma_1 \geq \cdots \geq \sigma_n$. For each retained fraction in $\{0.95, 0.90, 0.85, 0.80, 0.70\}$, we fit

$$\log(\text{rank}_i) = b + s_\ell \log(\sigma_i) \tag{1}$$

by linear regression and retain the fit with the smallest normalized rank-space discrepancy. The reported quantity is the fitted slope s_ℓ, which is negative. It is not the standard positive HT-SR exponent.

Under the idealized density convention $p(\lambda) \propto \lambda^{-\alpha_{\text{pl}}}$ with $\lambda = \sigma^2$, the corresponding CCDF relation gives $\text{rank} \propto \sigma^{2(1-\alpha_{\text{pl}})}$ and hence $\alpha_{\text{pl}} = 1 - s/2$. A more negative fitted slope therefore corresponds to a larger density exponent under this convention. Because our cutoff selection and regression are not a standard Hill or Clauset estimator, we do not report transformed slopes as corrected HT-SR estimates. The analysis below is descriptive.

Observed separation. Across the 224 linear projections, the mean fitted slopes by type are $q = -1.520$, $k = -1.531$, $v = -1.898$, $o = -1.913$, $\text{gate} = -3.248$, $\text{up} = -3.565$, and $\text{down} = -3.583$. Thus the four attention types and three MLP types occupy distinct slope ranges on this model. Table 10 shows that the separation is also visible at representative depths. This observation does not order `up_proj` ahead of `down_proj`; their means are nearly identical. It also contains no information about the optimal LoRA rank.

Role in the placement study. The original sweep included two slope-weighted rank allocations, which we call Slope-Prop and Slope-Inverse, together with uniform, attention-only, MLP-only, and single-projection configurations. Their comparison shows that different slope-weighted allocations produce different PPL values under the shared schedule. It does not establish the slope as an optimal target-selection method. In particular, `up_proj` and rank 4 are identified by separate empirical ablations in §D.1 and §D.2.

D Exploratory Placement Sweeps

Setup. Base model: LLaMA-2-7B (Touvron et al., 2023). Fine-tuning data: Alpaca (Taori et al., 2023) with a 10K-sample subset (one 50K-sample run is also reported as *Full Alpaca*). All LoRA runs use the HuggingFace PEFT library with `task_type="CAUSAL_LM"`, `lora_dropout=0.05`, `bias="none"`, and $\alpha_{\text{LoRA}} = 2r$. Models are loaded in bfloat16. We train for three epochs with AdamW (weight decay 0.01, cosine schedule, 3% warmup, peak learning rate 2×10^{-4}), effective batch size 32, maximum sequence length 512. Gradient checkpointing is enabled. All runs execute on a single NVIDIA H100-80GB.

Primary evaluation: WikiText-2 (Merity et al., 2017) perplexity on 500 held-out passages of the `wikitext-2-raw-v1` test split, filtered for length >50 characters, with maximum context 2048 and batch size 1. For instruction-tuned models we merge the LoRA adapter into the base weights

Table 11: Single-seed WikiText-2 perplexity after LLaMA-2-7B/Alpaca fine-tuning under a shared schedule. Lower is better. `up_proj`-only at rank 4 is the best of the 25 tested configurations.

Method	**Target modules**	**Rank**	**Params (M)**	**PPL ↓**
Uniform LoRA	Q/K/V/O + gate/up/down	4	10.0	12.54
Uniform LoRA	Q/K/V/O + gate/up/down	8	20.0	13.04
Uniform LoRA	Q/K/V/O + gate/up/down	16	40.0	13.35
Uniform LoRA	Q/K/V/O + gate/up/down	20	49.9	13.67
Slope-Prop	all 7	4	8.3	13.01
Slope-Prop	all 7	8	16.3	13.37
Slope-Prop	all 7	16	32.7	13.76
Slope-Inverse	all 7	4	12.1	12.27
Slope-Inverse	all 7	8	24.4	12.39
Slope-Inverse	all 7	16	48.8	12.49
Attention-only (control)	Q/K/V/O	16	16.8	13.25
Attention-only (control)	Q/K/V/O	32	33.6	13.90
MLP-only	gate/up/down	2	2.9	11.55
MLP-only	gate/up/down	4	5.8	11.63
MLP-only	gate/up/down	8	11.6	11.95
MLP-only	gate/up/down	16	23.2	11.90
MLP-only	gate/up/down	32	46.4	12.11
MLP-only (50K Alpaca)	gate/up/down	4	5.8	11.42
Single MLP projection at r=16:				
`gate_proj` only	gate_proj	16	7.7	11.42
`down_proj` only	down_proj	16	7.7	11.40
`up_proj` only	up_proj	16	7.7	11.13
`up_proj`-only rank sweep (r=16 row above):				
`up_proj` only	**up_proj**	**4**	**1.9**	**10.93**
`up_proj` only	up_proj	8	3.9	11.20
`up_proj` only	up_proj	32	15.5	11.61
Original LoRA (Hu et al., 2022)	q_proj, v_proj	16	8.4	12.57

before evaluation. Secondary evaluation: GSM8K (Cobbe et al., 2021) accuracy with greedy decoding on the full 1319-example test set (prompt and extraction details in §B).

Configurations. Table 11 reports the original 25 target-module and rank combinations. All rows use the same training schedule, which controls the comparison under that schedule but does not individually optimize hyperparameters for each placement. Slope-Prop and Slope-Inverse refer to the two allocations described in §C; they should not be read as validated HT-SR methods.

D.1 Which MLP projection matters most?

LLaMA-2 uses a SwiGLU MLP of the form $\mathrm{MLP}(x) = \mathrm{down}\big(\mathrm{SiLU}(\mathrm{gate}(x)) \odot \mathrm{up}(x)\big)$. Fixing rank 16 on a single MLP projection at a time, we adapt one of `gate_proj`, `up_proj`, `down_proj`, and keep the other MLP weights and all attention weights frozen.

- `up_proj` only: PPL 11.13
- `down_proj` only: PPL 11.40
- `gate_proj` only: PPL 11.42
- All three (MLP-only, rank 16): PPL 11.90

The ranking differs from the `down_proj`-focused interventions in ROME (Meng et al., 2022), but the objectives are also different: ROME performs localized factual editing, whereas our experiment performs Alpaca instruction tuning. The present data do not establish why `up_proj` is better here. They show only that it is the best of the three single-MLP targets under this model, dataset, and schedule. The descriptive slopes in §C do not distinguish it from `down_proj`.

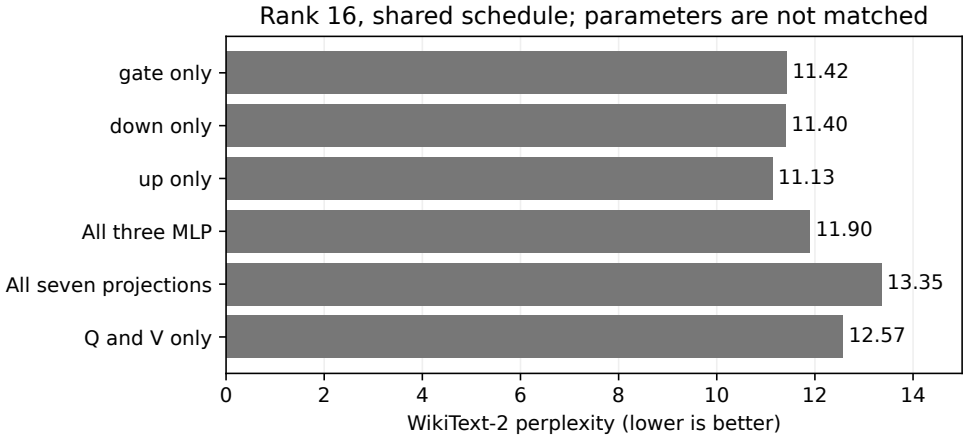

Figure 3: Single-seed perplexity for target choices at rank 16. Under the shared schedule, `up_proj`-only has lower PPL than the other displayed targets.

Figure 3 visualizes this ablation. Two observations are worth flagging. First, `up_proj` alone beats all three MLP projections together by 0.77 PPL while using a third of the parameters. Second, QV-only LoRA, the recipe in the original LoRA paper (Hu et al., 2022), scores 12.57, worse than every MLP-only variant we tested.

D.2 Rank sweep on `up_proj`-only

With the target module fixed to `up_proj`, rank 4 obtains PPL 10.93 with 1.9M parameters. The single-seed values at ranks 8, 16, and 32 are 11.20, 11.13, and 11.61. Thus rank 4 is the best tested rank for this target and schedule. We do not infer from four ranks that low rank is universally optimal.

D.3 Shared hyperparameters and their limitation

Using the same learning rate, dropout, LoRA scaling, epochs, and effective batch size controls the placement comparison under one schedule. It does not show that every placement is individually optimized. The three-seed results address initialization and data-order variability for the central configurations, but not placement-by-hyperparameter interactions. A per-placement hyperparameter sweep is therefore needed before attributing every difference solely to target choice.

Selection and resource matching. WikiText-2 test perplexity was used to compare the 25 configurations and identify the reported target/rank choice. It therefore serves as a selection metric in this study, not an untouched confirmatory test. Repeating selected configurations with new training seeds does not remove this evaluation-set reuse. An independent validation split for selection followed by a fresh held-out test is needed to estimate post-selection generalization. Moreover, fixed rank across different target sets does not fix trainable parameter count, and the central rank-4 versus rank-16 comparison changes both rank and placement. Shared training steps do not match optimization capacity, update scaling, or compute. These results are comparisons of complete configurations under one schedule, not isolated causal effects of target placement at a matched resource budget. The new up/down comparison in Section 5 matches rank and parameter count, but remains one schedule and one same-dataset holdout metric.

Comparison with the original LoRA configuration. The original LoRA recipe (Hu et al., 2022) adapts only `q_proj` and `v_proj`. At rank 16 this configuration has 8.4M parameters and scores PPL 12.57, compared with 10.93 for `up_proj`-only at rank 4. Under this model, data, schedule, and metric, the QV-only configuration is not the best tested placement.

Reproducibility. The supplementary archive includes the matched-budget training and evaluation scripts, rank-allocation records, saved outputs and analysis code. Section 5.1 specifies the separate holdout experiment, including fixed sample indices, per-seed outputs and per-case loss denominators. The training scripts use the HuggingFace stack. Integrity checks and saved-result reproduction commands are provided in the archive README; reproducing model training additionally requires the stated checkpoints and compute.

E ADDITIONAL EVIDENCE BOUNDARY

Correcting a generation template on an already trained adapter does not reproduce training with that template. Native chat serialization, response-only labels, padding masks, and termination tokens jointly define the protocol. An official test subset first inspected after model selection remains retrospective, and successful termination alone does not demonstrate downstream accuracy.