WATER-FILLING LORA: DATA-FREE RANK ALLOCATION VIA CONDITIONAL ESTIMATION–APPROXIMATION BOUNDS

Anonymous authors
Paper under double-blind review

ABSTRACT

Choosing a LoRA rank for each layer is a resource-allocation problem when the total number of trainable parameters is fixed. We analyze this problem using a PAC-Bayes bound for Gibbs risk and an explicit spectral-tail approximation model. Under a common norm envelope and the stated approximation assumptions, the continuous optimum is proportional to pretrained stable rank, with clipping at the rank limits. The contribution is this conditional budgeted analysis, rather than the use of stable rank itself. We distinguish the continuous optimum, the proportionally rounded implementation, and a budget-feasible integer rule with a computable surrogate suboptimality certificate. An exact integer audit on Pythia-160M and Pythia-410M exposes a small budget overrun in the rounded 410M allocation and verifies feasibility of the budget-feasible rule. In four-seed adapter-only experiments, the proportionally rounded heuristic has lower mean validation perplexity than AdaLoRA on both models (28.353 versus 29.898 and 15.373 versus 16.078), while its advantage over uniform rank is not resolved. A separate four-seed comparison of the budget-feasible integer rule finds lower mean PPL than proportional rounding at 160M and lower mean PPL than uniform at 410M, but no consistent advantage across both comparators and models. Neither comparison verifies the assumptions linking the surrogate to downstream risk. The analysis identifies when a data-free allocation is optimal for a specified bound and where empirical validation is still necessary.

1 INTRODUCTION

Parameter-efficient fine-tuning (PEFT) methods have become essential for adapting large language models (LLMs) to downstream tasks without updating all parameters (Hu et al., 2022; Ding et al., 2023). Among these, LoRA (Hu et al., 2022) is widely adopted: for each target weight matrix $W_l^{(0)} \in \mathbb{R}^{d_l \times k_l}$, it injects a low-rank perturbation $\Delta W_l = B_l A_l$ where $A_l \in \mathbb{R}^{r_l \times k_l}$, $B_l \in \mathbb{R}^{d_l \times r_l}$, and keeps the pretrained weights frozen during training.

The rank allocation problem. LoRA introduces a key hyperparameter per layer: the rank r_l. In practice, a single global rank r is set for all layers—a default that is convenient but not generally optimal for a heterogeneous-layer surrogate. Recent empirical work has shown that different layers play qualitatively different roles during fine-tuning (Hu et al., 2022; Zhang et al., 2023b; Ding et al., 2023), while comparisons with full fine-tuning reveal an adaptation–retention tradeoff (Biderman et al., 2024).

Scope. We study *training-free, one-shot* rank allocation: the entire assignment is computed from pretrained weights before fine-tuning begins, requiring no labeled data and no modification to the training procedure. This differs from training-time methods (e.g., AdaLoRA (Zhang et al., 2023c)) that adapt ranks during training. Our goal is to understand how the structure of pretrained weights should inform the initial rank budget, providing a conditional alternative to uniform allocation. Its one-time spectral computation is separate from downstream training cost. SR-LoRA (Zhang et al., 2025) already uses pretrained stable rank for layer-wise allocation. ARD-LoRA (Shinwari & Usama,

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.

2025) instead learns rank allocation through scaling factors. Our claim concerns the stated budgeted surrogate and its integer implementation, not priority for heterogeneous ranks or stable-rank guidance.

Our contribution. We analyze rank allocation through three results:

1. **Generalization bound (Theorem 3).** Using the PAC-Bayes framework (Seeger, 2002), we derive a generalization bound for heterogeneous-rank LoRA of the form $\mathcal{O}\left((C\alpha/\tau)\sqrt{\sum_l \sigma_{\text{eff}}(W_l^{(0)})^2\,(d_l+k_l)/(nr_l)}\right)$, where $\sigma_{\text{eff}}(W) = \|W\|_F^2 \,/\, \|W\|_{\text{op}}^2$ is the *stable rank* of W. Equal-covariance product Gaussian prior and posterior distributions yield an exact additive KL divergence $\sum_l \mathrm{KL}(Q_l\|P_l)$. The statement explicitly controls Gibbs risk and does not use a zero-variance limit.
2. **Water-filling optimum (Theorem 8).** Under an explicit $1/r_l$ spectral-tail approximation condition and a fixed parameter budget $P = \sum_l r_l(d_l + k_l)$, the allocation-dependent part of the resulting risk bound is minimized by $r_l^* \propto \sigma_{\text{eff}}(W_l^{(0)})$—a *water-filling* formula (clipped when rank constraints are active).
3. **Quantified improvement (Corollary 10).** In the unclipped interior case, the relative reduction of the approximation surrogate is exactly $\mathrm{CV}_w^2[s]/(1+\mathrm{CV}_w^2[s])$, where CV_w is the width-weighted coefficient of variation of the stable ranks. This is not a direct prediction of PPL improvement.

We evaluate the method across model scales and include a controlled adapter-only comparison with AdaLoRA. The results support the data-free operating point but also expose regimes in which the WaterFill–Uniform difference remains unresolved.

2 Background and Related Work

LoRA and its variants. LoRA (Hu et al., 2022) initializes $B_l = 0$ and $A_l \sim \mathcal{N}(0, \sigma^2)$, so the fine-tuned model is $W_l = W_l^{(0)} + (\alpha/r_l)B_lA_l$. Several works improve on the vanilla recipe: LoRA+ (Hayou et al., 2024) uses different learning rates for A and B; AdaLoRA (Zhang et al., 2023c) prunes singular values during training; rsLoRA (Kalajdzievski, 2023) scales LoRA adapters by $1/\sqrt{r}$; GeLoRA (Ed-dib et al., 2025) allocates rank using intrinsic-dimension estimates derived from hidden-state representations of data. None of these studies the same data-free, fixed-budget allocation problem through the conditional estimation–approximation model developed here.

Adaptive rank methods and comparison with SR-LoRA. Several methods allocate or remove LoRA capacity during fine-tuning. ALoRA (Liu et al., 2024), DoRA (Mao et al., 2024), and IncreLoRA (Zhang et al., 2023a) prune, redistribute, or incrementally add rank components using task-dependent importance signals. LoRA-drop (Zhou et al., 2025) and WeightLoRA (Veprikov et al., 2026) select adapters using their outputs or optimization-time contribution, while Flexora (Wei et al., 2025) formulates layer selection as hyperparameter optimization. MELoRA (Ren et al., 2024) instead reorganizes a fixed parameter budget as an ensemble of small adapters. These are relevant training-time or data-dependent alternatives; WaterFill addresses the different setting in which all ranks must be chosen from pretrained weights before downstream data or gradients are available.

SR-LoRA (Zhang et al., 2025) already allocates ranks from the stable rank of pretrained weights and motivates that choice using generalization theory. We therefore do not claim that using pretrained stable rank is new or that SR-LoRA lacks theoretical motivation. Our narrower contribution is the conditional fixed-parameter-budget optimization of an explicit surrogate, including clipping and a comparison with uniform allocation. The assumptions linking this surrogate to downstream risk remain separate from the allocation calculation. Adaptive training-time methods such as AdaLoRA and ARD-LoRA address a different information constraint and provide relevant empirical comparators, rather than evidence that the conditional bound is tight.

Convergence theory for LoRA. Mu & Klabjan (2026) give an $\mathcal{O}(1/\log T)$ stationarity rate for LoRA gradient descent. They assume the original weight-space loss is smooth and lower bounded, but do not require global Lipschitz smoothness of the adapter-parameterized loss. Kim et al. (2025) distinguish low-rank global minima from a high-rank, large-magnitude regime and discuss the

implicit bias of initialization and weight decay. Neither work addresses per-layer rank allocation or generalization.

Generalization of PEFT. Matrix rank and norm constraints yield generalization bounds for matrix prediction (Srebro & Shraibman, 2005), while spectral complexity controls margin bounds for neural networks (Bartlett et al., 2017). Our Gibbs-risk analysis instead uses a PAC-Bayes bound with a rank-dependent update norm. Yaras et al. (2024) show that overparameterized low-rank learning stays within a compressible subspace, but does not derive explicit rank allocation rules.

Stable rank. The stable rank $\sigma_{\text{eff}}(W) = \|W\|_F^2 / \|W\|_{\text{op}}^2$ is a soft version of matrix rank that measures the effective dimensionality of the column space (Rudelson & Vershynin, 2007). Stable rank describes a pretrained matrix; relating it to a task-specific update requires additional assumptions, as made explicit below.

3 Theory

3.1 Setup

Consider an L-layer transformer with pretrained weights $\{W_l^{(0)}\}_{l=1}^L$, $W_l^{(0)} \in \mathbb{R}^{d_l \times k_l}$. Each layer receives a LoRA adapter: $\Delta W_l = B_l A_l$, $A_l \in \mathbb{R}^{r_l \times k_l}$, $B_l \in \mathbb{R}^{d_l \times r_l}$, initialized with $B_l = 0$. The full model computes $h = f_L \circ \cdots \circ f_1$ where $f_l(x) = \phi\Big((W_l^{(0)} + \frac{\alpha}{r_l}\Delta W_l)x\Big)$ and ϕ is the activation function. We assume the loss ℓ is $[0,1]$-bounded and that the downstream sample is independent of the data used to obtain the pretrained weights.

Definition 1 (Stable rank)**.** *For a nonzero matrix $W \in \mathbb{R}^{d\times k}$, the* stable rank *is*

$$\sigma_{\text{eff}}(W) \;=\; \frac{\|W\|_F^2}{\|W\|_{\text{op}}^2} \;=\; \frac{\sum_i \sigma_i(W)^2}{\sigma_{\max}(W)^2}.$$

$\sigma_{\text{eff}}(W)$ lies in $[1, \min(d,k)]$: it equals 1 for rank-1 matrices and $\min(d,k)$ for full-rank matrices with equal singular values.

3.2 Generalization Bound

Write $s_l = \sigma_{\text{eff}}(W_l^{(0)})$, $m_l = d_l + k_l$, and $c_l = s_l^2 m_l$.

Assumption 2 (LoRA perturbation energy)**.** *For a fixed fine-tuning protocol, the trained rank-r_l update satisfies*

$$\|B_l A_l\|_F^2 \le C^2 r_l s_l^2 m_l,$$

where $C > 0$ is shared across layers and all candidate rank allocations being compared. This is a structural condition on the training trajectory, not a claim about arbitrary gradient descent or a protocol-independent universal constant. Lemma 5 gives explicit sufficient conditions, and Section B.2 reports a direct empirical audit.

Theorem 3 (PAC-Bayes bound for the Gibbs predictor)**.** *Fix a data-independent $\tau > 0$ and define the product prior and posterior*

$$P = \bigotimes_l P_l, \quad Q = \bigotimes_l Q_l,$$
$$P_l = \mathcal{N}(\operatorname{vec} W_l^{(0)}, \tau^2 I),$$
$$Q_l = \mathcal{N}(\operatorname{vec}(W_l^{(0)} + (\alpha/r_l) B_l A_l), \tau^2 I).$$

Assume a loss in $[0,1]$ and pretrained weights independent of the n fine-tuning examples; raw unbounded language-model cross entropy does not meet this hypothesis. Let $R(Q) = \mathbb{E}_{\theta\sim Q} R(h_\theta)$ and define $\widehat{R}(Q)$ analogously. Under Assumption 2, with probability at least $1-\delta$ over $n \ge 8$ i.i.d. samples,

$$\boxed{R(Q) \le \widehat{R}(Q) + \sqrt{\frac{\frac{C^2\alpha^2}{2\tau^2}\sum_{l=1}^L c_l/r_l + \ln(2\sqrt{n}/\delta)}{2n}}.}$$

Proof. The PAC-Bayes-kl inequality with Maurer's refinement (Seeger, 2002; Maurer, 2004) and Pinsker's inequality give, for $n \geq 8$ and all posteriors Q,

$$R(Q) \leq \widehat{R}(Q) + \sqrt{\frac{\mathrm{KL}(Q\|P) + \ln(2\sqrt{n}/\delta)}{2n}}.$$

Because P and Q are chosen as product measures and have equal covariance,

$$\mathrm{KL}(Q\|P) = \sum_l \mathrm{KL}(Q_l\|P_l) = \frac{1}{2\tau^2}\sum_l (\alpha/r_l)^2 \left\|B_l A_l\right\|_F^2 .$$

Assumption 2 bounds the last display by $\frac{C^2\alpha^2}{2\tau^2}\sum_l c_l/r_l$ for positive ranks. Substitution proves the claim. An omitted layer has zero update and contributes zero KL; it is not evaluated by substituting $r_l = 0$ into this upper bound. □

Remark 4 (Scope of the PAC-Bayes statement)**.** *The rank-dependent factor α/r_l must match the implemented effective update. With fixed α, the complexity upper bound is proportional to $\mathcal{J}(\mathbf{r}) = \sum_l c_l/r_l$. Water-filling minimizes this complexity surrogate, but empirical risk can also change with allocation; the complete risk-bound claim below additionally assumes an approximation model. A common C must hold over the compared allocations; fitting it on WaterFill layers alone does not verify this premise for all alternatives. Theorem 3 directly controls the randomized Gibbs predictor. It does not invoke a zero-variance limit and does not by itself bound the deterministic posterior mean. Equal covariance is essential: if the prior covariance is I and the posterior covariance is $\sigma_Q^2 I$, the KL contains $\frac{d_l k_l}{2}(\sigma_Q^2 - 1 - \log\sigma_Q^2)$, which diverges as $\sigma_Q \to 0$.*

Lemma 5 (Sufficient gradient-flow conditions)**.** *Suppose $B_l(0) = 0$ and $\dot{B}_l = -\eta G_l A_l^\top$, where G_l includes the implemented α/r_l chain-rule factor. If for every $t \in [0, T]$,*

$$\left\|G_l(t)\right\|_{\mathrm{op}} \leq \gamma s_l\sqrt{m_l}, \qquad \left\|A_l(t)\right\|_F \leq a\sqrt{r_l}, \qquad \left\|A_l(t)\right\|_{\mathrm{op}} \leq a,$$

then Assumption 2 holds at time T with $C = \eta T\gamma a^2$.

Proof. Integration and submultiplicativity give $\left\|B_l(T)\right\|_F \leq \eta T\gamma a\sqrt{r_l}s_l\sqrt{m_l}$ and hence $\left\|B_l(T)A_l(T)\right\|_F \leq \left\|B_l(T)\right\|_F \left\|A_l(T)\right\|_{\mathrm{op}}$. Squaring proves the claim. □

3.3 Optimal Rank Allocation: Water-Filling

With fixed α and a common norm-envelope constant, the PAC-Bayes complexity term is increasing in $\mathcal{J}(\mathbf{r}) = \sum_l c_l/r_l$. Controlling the empirical-risk term still needs a complementary assumption. The source implementation uses one shared α and applies α/r_l in each layer; it does not set α_l proportional to every assigned rank.

Assumption 6 (Rank-limited approximation)**.** *Let U_l be a target full-rank update and $E_l(r) = \min_{\mathrm{rank}(X)\leq r}\left\|U_l - X\right\|_F^2$. For all allowed ranks, assume*

$$\sum_{j>r}\sigma_j(U_l)^2 \leq \frac{\kappa c_l}{r}, \qquad \widehat{R}(Q_{\mathbf{r}}) - \widehat{R}_* \leq K\sum_l E_l(r_l),$$

where $K, \kappa > 0$ are rank-independent and $\widehat{R}_*$ is a reference empirical Gibbs risk. The first inequality is an explicit $1/r$ spectral-tail condition; by Eckart–Young it implies $E_l(r_l) \leq \kappa c_l/r_l$.

Corollary 7 (Conditional budgeted risk bound)**.** *Under Assumptions 2–6 and the budget $\sum_l r_l m_l = P$, with the same probability as Theorem 3,*

$$R(Q_{\mathbf{r}}) \leq \widehat{R}_* + K\kappa\sum_l \frac{c_l}{r_l} + \sqrt{\frac{\frac{C^2\alpha^2}{2\tau^2}\sum_l c_l/r_l + \ln(2\sqrt{n}/\delta)}{2n}}.$$

Proof. Substitute Assumption 6 into Theorem 3, retaining the rank-dependent complexity term with the actual α/r_l scaling. □

Both allocation-dependent terms are increasing functions of the same $\mathcal{J}(\mathbf{r})$ when $C, K, \kappa, \alpha, \tau$ are held fixed. Thus minimizing this conditional upper bound reduces to minimizing $\mathcal{J}$, without replacing complexity by an allocation-independent term.

Theorem 8 (Water-filling rank allocation)**.** *Consider the optimization problem:*

$$\min_{r_{\min} \le r_l \le r_{\max}} \sum_{l=1}^{L} \frac{c_l}{r_l} \quad \textit{subject to} \quad \sum_{l=1}^{L} r_l(d_l + k_l) = P,$$

where $c_l = s_l^2 m_l > 0$, $0 < r_{\min} < r_{\max}$, and the budget is strictly between the all-floor and all-ceiling budgets. The unique optimum of the continuous relaxation is clipped water-filling:

$$\boxed{r_l^* = \mathrm{clip}\left(\frac{s_l}{\sqrt{\mu}}, r_{\min}, r_{\max}\right),}$$

where $\mu > 0$ is chosen to meet the budget. If neither bound is active, then

$$r_l^* = \frac{P}{\sum_{l'} s_{l'} m_{l'}} s_l,$$

so $r_l^* \propto \sigma_{\text{eff}}(W_l^{(0)})$.

Proof. Form the Lagrangian $\mathcal{L}(\mathbf{r}, \mu) = \sum_l c_l/r_l + \mu(\sum_l r_l(d_l + k_l) - P)$. For an interior coordinate, the KKT condition gives $c_l/r_l^2 = \mu m_l$ and hence $r_l = s_l/\sqrt{\mu}$. Complementary slackness clips this value at the lower and upper bounds. Strict convexity gives a unique rank vector. A positive multiplier can be chosen to meet the budget, but need not be unique when all coordinates are clipped. At the two budget endpoints the corresponding all-floor or all-ceiling vector is the unique feasible solution. Integer rounding need not preserve continuous optimality or exactly meet a weighted integer budget. □

Remark 9 (Scope of Theorem 8)**.** *Theorem 3 controls estimation; Assumption 6 controls rank-limited approximation. Theorem 8 minimizes the resulting bound in Corollary 7. Its optimality is conditional on the stated spectral-tail and empirical-risk model, not an algebraic consequence of the PAC-Bayes term alone.*

Corollary 10 (Quantified improvement over uniform allocation)**.** *Let $w_l = m_l / \sum_j m_j$ and let $\mathrm{CV}_w[s]$ denote the weighted coefficient of variation. If no rank bound is active, then*

$$\frac{\mathcal{J}^*}{\mathcal{J}^{\text{unif}}} = \frac{1}{1 + \mathrm{CV}_w^2[s]},$$

where $\mathcal{J}(\mathbf{r}) = \sum_l c_l/r_l$. The relative surrogate improvement is $\mathrm{CV}_w^2[s]/(1 + \mathrm{CV}_w^2[s])$.

This corollary concerns the approximation surrogate, not PPL directly, and does not apply unchanged when clipping is active.

4 Algorithm

Theorem 8 yields a continuous solution. Algorithm 1 gives a budget-feasible integer implementation; the training experiments compare this rule with proportional rounding and uniform ranks.

Continuous and integer allocation. The rounded allocator scales the unconstrained proportional ranks, rounds and clips them, and then greedily adjusts ranks toward the parameter budget. This procedure does not solve the clipped continuous KKT multiplier exactly, and the greedy weighted integer repair has no proven optimality guarantee. We therefore distinguish the conditional continuous theorem from the tested rounded heuristic; the empirical results do not certify exact constrained optimality of its ranks.

Algorithm 1 Water-Filling LoRA Rank Allocation

Require: Pretrained model $\{W_l^{(0)}\}$, total parameter budget P, minimum rank $r_{\min} = 1$, maximum rank $r_{\max}$

1: **for** each layer $l = 1, \ldots, L$ **do**
2: Compute all singular values $\{\sigma_i(W_l^{(0)})\}$ via full SVD (`torch.linalg.svdvals`) $\{\mathcal{O}(\min(d_l, k_l) \cdot d_l k_l)\}$
3: Compute $\sigma_{\text{eff}}(W_l^{(0)}) \leftarrow \left\|W_l^{(0)}\right\|_F^2 / \sigma_{\max}(W_l^{(0)})^2$
4: **end for**
5: Find $\mu > 0$ by bisection so that $\sum_l (d_l + k_l)\,\text{clip}(\sigma_{\text{eff}}(W_l^{(0)})/\sqrt{\mu}, r_{\min}, r_{\max}) = P$
6: **for** each layer l **do**
7: $r_l \leftarrow \lfloor \text{clip}(\sigma_{\text{eff}}(W_l^{(0)})/\sqrt{\mu}, r_{\min}, r_{\max}) \rfloor$
8: **end for**
9: **while** an increment fits the remaining budget and rank cap **do**
10: Among feasible layers, select the largest $c_l/[m_l r_l(r_l + 1)]$, where $m_l = d_l + k_l$, $c_l = \sigma_{\text{eff}}(W_l^{(0)})^2 m_l$
11: Increase that layer's rank by one; update the remaining budget
12: **end while**
13: **return** Rank vector $\mathbf{r} = (r_1, \ldots, r_L)$

A computable integer-allocation certificate. Let $F(r) = \sum_l c_l/r_l$ and let r^* minimize this objective over continuous ranks with the budget inequality. For integer rank limits and a feasible budget, flooring r^* is feasible. Each subsequent feasible increment reduces F by $c_l/[r_l(r_l + 1)]$. Thus the returned $\widehat{r}$ satisfies $F(r^*) \le F^*_{\text{integer}} \le F(\widehat{r})$. The observable difference $F(\widehat{r}) - F(r^*)$ bounds its integer suboptimality; it is a surrogate certificate, not a bound on measured PPL. Unused budget is reported explicitly. The marginal-benefit-per-cost rule is not generally an exact integer optimizer when layer costs differ.

Complexity. The dominant cost is computing the singular values of $W_l^{(0)}$ for each layer via full SVD (`torch.linalg.svdvals`), which costs $\mathcal{O}(\min(d_l, k_l) \cdot d_l k_l)$ per layer. Total cost: $\mathcal{O}(\sum_l \min(d_l, k_l) \cdot d_l k_l)$. No training data is required for allocation.

5 Experiments

5.1 Setup and comparison units

Our primary experiments use Pythia-160M and Pythia-410M (Biderman et al., 2023) on WikiText-2 language modeling (Merity et al., 2017). We distinguish a four-seed comparison of allocation methods, a separate four-seed comparison of integer budget rules, and an eight-run dense-update diagnostic. Base-model parameters are frozen in the two adapter experiments. Each experiment uses a fixed data split and reports its own seed variation; observations are not pooled across protocols.

WikiText-2 text is packed into 512-token sequences in batches of 1,000 source rows, discarding each packing batch's remainder. Training uses batch size four, AdamW with learning rate 2×10^{-4} and weight decay 0.01, a cosine schedule with 50-step warmup, and exactly 2,000 updates. The evaluation uses the fixed validation loader. WaterFill uses rank bounds 4–32 and a nominal rank-eight budget. Uniform assigns rank eight; Random uses a randomized rank vector; AdaLoRA starts at rank 16 and targets rank eight through training-time pruning (Zhang et al., 2023c). The AdaLoRA initialization has a larger trainable state, so these are not compute-matched methods.

Earlier broader-training measurements are reported separately in Appendix A. In that implementation, freezing wrapped linear layers left other parameters trainable; the Pythia runs stopped after 1,182 updates. They cannot isolate adapter allocation and are not 2,000-update replications of the primary comparison.

Table 1: Controlled adapter-only WikiText-2 validation PPL (mean ± sample standard deviation over four seeds; lower is better).

Model	WaterFill	Uniform	Random	AdaLoRA
Pythia-160M	$\mathbf{28.353 \pm 0.144}$	28.626 ± 0.421	28.377 ± 0.110	29.898 ± 0.103
Pythia-410M	15.373 ± 0.217	15.427 ± 0.326	$\mathbf{15.371 \pm 0.355}$	16.078 ± 0.145

WaterFill vs. AdaLoRA: paired $p = 4.21 \times 10^{-4}$ (160M) and $p = 0.0092$ (410M). WaterFill vs. Uniform: $p = 0.234$ and $p = 0.822$.

5.2 Controlled Adapter-Only Audit

To isolate rank allocation from other trainable parameters, we additionally freeze every base-model parameter and adapt only transformer projection matrices for exactly 2,000 updates. Table 1 reports four seed labels under the same data split, optimizer, schedule, and sequence length for every method. The earlier script constructs methods sequentially without resetting all random streams; seed-label pairing does not ensure identical initialization or minibatch order. The additional integer-rule comparison resets these streams explicitly.

WaterFill lowers mean PPL relative to AdaLoRA by 5.17% on 160M and 4.39% on 410M. Its differences from Uniform and Random are not statistically significant in this controlled audit. This establishes neither superiority nor parity: no equivalence margin or adequately powered equivalence design was specified. WaterFill and Uniform use exactly 1,179,648 adapter parameters on 160M. On 410M they use 3,146,752 and 3,145,728, respectively (a 0.033% rounding difference). AdaLoRA targets rank 8 but initializes at rank 16, storing 2,360,064 and 6,292,992 trainable parameters during optimization.

Four-seed uncertainty and test sensitivity. The paired WaterFill-minus-Uniform PPL differences are -0.273 on 160M (paired-t 95% interval $[-0.857, 0.312]$) and -0.054 on 410M ($[-0.761, 0.652]$). These intervals concern variation across four training seeds, not evaluation-example or cross-task uncertainty. Paired-t tests assume approximately normal seed differences, which four seeds cannot substantiate. For the six WaterFill-versus-baseline comparisons across these two models, Holm-adjusted paired-t p-values against AdaLoRA are 0.00253 and 0.0460. An exact two-sided sign-flip sensitivity calculation gives $p = 0.125$ in both cases, the smallest possible value with four nonzero paired differences. Its sign-exchangeability assumption is also explicit; it is not a substitute for a randomized experiment. We consequently report lower observed means and test-dependent evidence, not distribution-free superiority. AdaLoRA's larger initial state also prevents interpreting this as a compute-matched comparison.

We also audit Assumption 2 directly: one empirical layer-independent $C^2 = 2.72 \times 10^{-5}$ upper-bounds every layer in all eight controlled WaterFill runs. Appendix B.2 reports the per-model ranges and correlations; this is a post-training check, not a guarantee for arbitrary trajectories.

Additional earlier-protocol ablations and scaling observations appear in Appendix A. Their broader trainable-parameter setup differs from the adapter-only experiments, so the two protocols are analyzed separately. Neither establishes equivalence between WaterFill and Uniform.

5.3 Training comparison of integer allocation rules

We compare proportional rounding, the budget-feasible rule, and uniform rank with four additional paired seeds (20260920–20260923) for each model. Each condition uses 2,000 AdamW steps, sequence length 512, batch size 4, learning rate 2×10^{-4}, and the same WikiText-2 split. All base-model parameters are frozen. Random generators are reset before adapter construction and again before training, so adapter shapes do not change the subsequent data-order random stream. These runs are analyzed separately from the preceding seed set. The budget-feasible rule improves on proportional rounding in all four 160M seeds (mean paired difference -0.331 PPL), but its mean differs from uniform by $+0.007$ PPL and only one of four seeds favors it. At 410M, its mean paired difference is -0.132 relative to uniform (three of four seeds favor it) and $+0.021$ relative to proportional rounding. Three seeds favor the new rule in the latter comparison, but the remaining seed reverses the mean; the mean and win count therefore convey different information. These descriptive

Table 2: New allocation comparison: validation PPL, mean ± sample standard deviation over four paired seeds. Lower is better. No new AdaLoRA runs are included in this comparison.

Allocation	Pythia-160M	Pythia-410M
Proportional rounding	28.622 ± 0.321	15.297 ± 0.130
Budget-feasible rule	28.290 ± 0.038	15.318 ± 0.076
Uniform	28.283 ± 0.121	15.450 ± 0.117

Table 3: Task-update diagnostic over two seeds per cell. Ranges show both runs, not confidence intervals. Initial PPL is 34.37 for 160M and 20.23 for 410M. ρ_8 relates pretrained stable rank to the relative rank-8 residual of the learned dense projection update.

Model	Learning rate	Final PPL range	ρ_8 range
160M	2×10^{-4}	142.25–155.07	0.252–0.303
160M	2×10^{-5}	31.74–36.48	0.074–0.105
410M	2×10^{-4}	44.66–44.98	-0.136–-0.091
410M	2×10^{-5}	14.50–15.04	-0.054–0.081

comparisons do not establish consistent superiority or equivalence. At 160M all three allocations use 1,179,648 trainable parameters. At 410M proportional rounding uses 3,146,752, the budget-feasible rule 3,144,704, and uniform 3,145,728; the nominal budget is 3,145,728. Thus fixing the constraint violation and improving a surrogate does not ensure a downstream PPL improvement against every comparator.

Task-update spectrum check. A separate dense-projection diagnostic on both Pythia models does not establish that pretrained stable rank predicts the task update's approximation tail. The original learning rate degrades validation PPL; a lower-rate sensitivity has mixed outcomes on 160M and improves 410M, with weak spectral association. All eight runs and the update definition are reported in Appendix E. We therefore retain the approximation model as an explicit condition, not an empirically certified consequence of pretrained weight geometry.

6 Discussion

Scope and practical implications. Algorithm 1 is designed as a *training-free, one-shot* rank allocation method: the rank assignment is computed from the pretrained weights, requiring no labeled data, gradient computation, or additional training. This contrasts with *training-time* adaptive methods (e.g., AdaLoRA (Zhang et al., 2023c)) that perform SVD-based rank pruning during fine-tuning, incurring persistent additional overhead. WaterFill occupies a distinct point in the efficiency–performance tradeoff space.

Scale-dependent effects and open questions. The earlier broader-training comparison and the new adapter-only comparisons use different trainable parameter sets. The new four-seed, 2,000-step language-modeling comparison does not establish a general advantage: at 160M, mean perplexity is 28.290 for the new allocation and 28.283 for uniform allocation; at 410M, the corresponding means are 15.318 and 15.450. These measurements use a different endpoint from the earlier comparison and should not be pooled with it. Allocation benefits therefore remain dependent on the task, budget, and training protocol. Validation at 7B+ scale remains open.

Theoretical distinction from AdaLoRA. AdaLoRA (Zhang et al., 2023c) allocates rank by importance-scoring singular values *during* training via SVD reparameterization, making it a *data-driven, online* method. Our approach operates in a fundamentally different regime: it is *data-free* and *offline*. The entire rank allocation is determined before training begins, using only the pretrained weight matrices. The theoretical underpinnings also differ: our result separates a PAC-Bayes estimation term from an explicit spectral-tail approximation model, whereas AdaLoRA uses an

optimization-time importance heuristic. In practice, the two methods are complementary: our water-filling initialization could be combined with AdaLoRA's online pruning as a warm-start.

Connection to SR-LoRA and ARD-LoRA. GeLoRA (Ed-dib et al., 2025) uses intrinsic dimension estimates of hidden representations; our pretrained-weight stable rank enters our PAC-Bayes bound under Assumption 2. SR-LoRA (Zhang et al., 2025) and ARD-LoRA (Shinwari & Usama, 2025) are prior methods that also allocate ranks heterogeneously. Our distinction is the explicit budgeted estimation–approximation model: under its stated conditions it yields the same $r_l^* \propto \sigma_{\text{eff}}$ formula and the weighted-CV surrogate reduction in Corollary 10.

Limitations. WaterFill is a training-free initialization method that sets the rank budget before fine-tuning and does not adapt ranks during training (unlike AdaLoRA (Zhang et al., 2023c)); combining WaterFill initialization with online rank pruning is a natural extension. Assumption 2 requires stable-rank-scaled gradients and bounded factor trajectories; Lemma 5 does not establish these conditions for arbitrary GD. The water-filling guarantee additionally depends on Assumption 6; its spectral-tail and empirical risk conditions may fail for some tasks. The larger-model measurements in Appendix A use a different trainable-parameter setup and do not establish generality of the adapter-only result. The mechanism behind model-dependent allocation outcomes remains open. We validate primarily on WikiText-2 language modeling; downstream task evaluation on benchmarks such as MMLU and GSM8K is left to future work.

Interpreting the task-update diagnostic. Table 3 separates two questions that a downstream PPL comparison cannot answer by itself. The first is whether a dense projection update learned by the chosen optimizer improves the task objective. The second is whether the pretrained spectrum predicts how much of that update a low-rank approximation discards. At the larger learning rate, all four dense runs worsen validation PPL, so their update spectra are not credible estimates of a useful task solution. At the smaller rate, both 410M runs improve substantially, yet their rank-8 associations with pretrained stable rank are near zero and have different signs. Successful optimization alone therefore does not supply evidence for the spectral proxy.

This diagnostic also has a precise limit. It compares relative residual energy across layers, whereas the approximation model concerns a risk contribution under additional scale and norm conditions. A weak rank association neither proves that every absolute tail bound fails nor rules out useful allocation on another task. Conversely, an observed PPL improvement from one allocation would not verify those conditions: optimization dynamics and regularization can change when ranks change. The relevant empirical conclusion is narrower. On the measured updates, pretrained stable rank is not a reliable stand-alone ordering of relative approximation difficulty. The theorem remains a conditional statement about the supplied surrogate, and its integer certificate concerns that surrogate rather than task risk.

7 Conclusion

We combine a PAC-Bayes Gibbs-risk bound with an explicit rank-approximation model; under the stated conditions, clipped water-filling is the budgeted optimum. The resulting method selects all ranks from pretrained weights before fine-tuning. The controlled experiments show an unresolved difference from Uniform and lower mean PPL than AdaLoRA with test-dependent statistical evidence. Correcting the integer budget does not produce a consistent downstream gain against every comparator. These results distinguish exact optimization of a supplied surrogate from validation of that surrogate. The allocation rule is budget-feasible, but its downstream advantage remains dependent on the model and comparator.

Reproducibility Statement

The supplement provides allocation code, paired-seed training records, the dense projection diagnostic, and manuscript sources. The experiments use public Pythia checkpoints and WikiText-2. Rank budgets, targeted projections, data packing, random seeds, and learning rates are stated in the experimental sections and Appendix E. The dense updates are optimizer-dependent diagnostics and do not certify the unknown best task update.

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

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. arXiv:1706.08498.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John P Cunningham. Lora learns less and forgets less. *Transactions on Machine Learning Research*, 2024. URL https://openreview.net/forum?id=aloEru2qCG.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar Van Der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *Proceedings of the 40th International Conference on Machine Learning*, 2023. URL https://proceedings.mlr.press/v202/biderman23a.html.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv:1803.05457*, 2018.

Ning Ding, Yujia Qin, Guang Yang, Fuchao Wei, Zonghan Yang, Yusheng Su, Shengding Hu, Yulin Chen, Chi-Min Chan, Weize Chen, et al. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence*, 5:220–235, 2023. doi: 10.1038/s42256-023-00626-4.

Abdessalam Ed-dib, Zhanibek Datbayev, and Amine Mohamed Aboussalah. GeLoRA: Geometric adaptive ranks for efficient lora fine-tuning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2025. URL https://aclanthology.org/2025.findings-emnlp.1372/. arXiv:2412.09250.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. Lora+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning*, 2024. URL https://proceedings.mlr.press/v235/hayou24a.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*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with lora. *arXiv:2312.03732*, 2023.

Junsu Kim, Jaeyeon Kim, and Ernest K Ryu. Lora training provably converges to a low-rank global minimum or it fails loudly (but it probably won't fail). In *Proceedings of the 42nd International Conference on Machine Learning*, 2025. URL https://proceedings.mlr.press/v267/kim25n.html.

Zequan Liu, Jiawen Lyn, Wei Zhu, Xing Tian, and Yvette Graham. ALoRA: Allocating low-rank adaptation for fine-tuning large language models. In *Proceedings of NAACL-HLT*, 2024. URL https://aclanthology.org/2024.naacl-long.35/.

Yulong Mao, Kaiyu Huang, Changhao Guan, Ganglin Bao, Fengran Mo, and Jinan Xu. DoRA: Enhancing parameter-efficient fine-tuning with dynamic rank distribution. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024. URL https://aclanthology.org/2024.acl-long.626/.

Andreas Maurer. A note on the PAC bayesian theorem. *arXiv preprint cs/0411099*, 2004. URL https://arxiv.org/abs/cs/0411099.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Siqiao Mu and Diego Klabjan. On the convergence rate of lora gradient descent. In *International Conference on Machine Learning*, 2026. URL https://icml.cc/virtual/2026/poster/65870.

Pengjie Ren, Chengshun Shi, Shiguang Wu, Mengqi Zhang, Zhaochun Ren, Maarten de Rijke, Zhumin Chen, and Jiahuan Pei. MELoRA: Mini-ensemble low-rank adapters for parameter-efficient fine-tuning. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4), 2007.

Matthias Seeger. PAC-Bayesian generalisation error bounds for gaussian process classification. *Journal of Machine Learning Research*, 3:233–269, 2002.

Haseeb Ullah Khan Shinwari and Muhammad Usama. ARD-LoRA: Dynamic rank allocation for parameter-efficient fine-tuning of foundation models with heterogeneous adaptation needs. *arXiv:2506.18267*, 2025.

Nathan Srebro and Adi Shraibman. Rank, trace-norm and max-norm. In *Conference on Learning Theory (COLT)*, 2005.

Andrey Veprikov, Vladimir Solodkin, Alexander Zyl, Andrey Savchenko, and Aleksandr Beznosikov. Weightlora: Keep only necessary adapters. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026. URL https://aclanthology.org/2026.acl-long.566/.

Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Yu. Flexora: Flexible low rank adaptation for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*, 2025. URL https://aclanthology.org/2025.acl-long.713/.

Can Yaras, Peng Wang, Laura Balzano, and Qing Qu. Compressible dynamics in deep overparameterized low-rank learning and adaptation. In *International Conference on Machine Learning (ICML)*, 2024.

Chuyan Zhang, Kefan Wang, and Yun Gu. Beyond low-rank tuning: Model prior-guided rank allocation for effective transfer in low-data and large-gap regimes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3337–3345, 2025.

Feiyu Zhang, Liangzhi Li, Junhao Chen, Zhouqiang Jiang, Bowen Wang, and Yiming Qian. IncreLoRA: Incremental parameter allocation method for parameter-efficient fine-tuning. *arXiv:2308.12043*, 2023a.

Longteng Zhang, Lin Zhang, Shaohuai Shi, Xiaowen Chu, and Bo Li. Lora-fa: Memory-efficient low-rank adaptation for large language models fine-tuning. *arXiv:2308.03303v1*, 2023b.

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*, 2023c. URL https://openreview.net/forum?id=lq62uWRJjiY.

Hongyun Zhou, Xiangyu Lu, Wang Xu, Conghui Zhu, Tiejun Zhao, and Muyun Yang. Lora-drop: Efficient lora parameter pruning based on output evaluation. In *Proceedings of the 31st International Conference on Computational Linguistics*, 2025. URL https://aclanthology.org/2025.coling-main.371/.

A Additional broader-training measurements

These measurements use a broader trainable-parameter setup, with trainable embeddings and normalization parameters outside the wrapped linear projections. They do not isolate adapter rank allocation. The Pythia records contain 1,182 updates per condition, whereas Qwen-7B contains 2,000; the schedule's configured horizon alone does not determine the number of optimizer updates. PPL uses the first 40 validation batches. We recompute all summaries directly from per-run records and keep this evidence separate from the adapter-only experiments.

Table 4: Broader-training validation PPL, mean $\pm$ sample standard deviation. Four Pythia-160M runs share the same trainable-parameter counts; the unnumbered run does not identify its random seed. The other model blocks use seed labels 1–3. These are descriptive summaries, not a matched causal rank-allocation effect.

Model	Updates	Uniform	Random	WaterFill
Pythia-160M	1182	31.669 ± 1.207	30.048 ± 0.382	28.378 ± 0.543
Pythia-410M	1182	16.110 ± 0.110	16.094 ± 0.081	16.236 ± 0.374
Qwen2.5-7B	2000	7.251 ± 0.037	7.259 ± 0.038	7.203 ± 0.043

The observed ordering depends on model and configuration: WaterFill has lower mean PPL than Uniform on 160M and Qwen-7B, but higher mean PPL on 410M. Random is close to or better than WaterFill in several comparisons. Shared seed labels do not ensure identical initialization or data order across methods: the earlier code constructs methods sequentially without resetting all random streams. We therefore do not use a paired significance claim to interpret this table.

An additional 160M run labeled seed 42 has a different WaterFill parameter count, so it is reported separately: WaterFill 31.544, Uniform 30.631, and Random 29.416 PPL after 1,182 updates. This unfavorable comparison is retained rather than pooled with a different allocation. The Qwen2.5-1.5B language-modeling run gives 9.050, 9.022, and 9.049 PPL for those methods after 1,224 updates. Initial PPL values are 34.383, 20.227, 8.426, and 11.648 for the 160M, 410M, Qwen-7B, and Qwen-1.5B models, respectively.

Trainable counts further distinguish the regimes. In the first 160M block, WaterFill and Uniform train 40,291,968 and 40,260,096 parameters, rather than approximately 1.18 million adapter parameters. The seed-42 WaterFill run trains 40,307,712. For 410M the counts are 55,217,664 and 55,168,000; for Qwen-7B, 577,495,552 and 577,347,072. These are close but not identical budgets and include more than the primary adapter parameters.

The separate ARC-Easy measurement (Clark et al., 2018) scores 100 items per model by answer likelihood. Qwen2.5-1.5B changes from 71% to 69% under both WaterFill and Uniform. Pythia-160M changes from 40% to 36% and 37%, respectively. Equal aggregate endpoint scores do not establish equal item-level outcomes, absence of forgetting, or non-inferiority. These single-run downstream results do not supply a broad transfer evaluation.

The unclipped surrogate reduction in Corollary 10 is not a prediction of any of these PPL changes. The corollary holds for its supplied approximation model; connecting that objective to optimization dynamics and downstream risk requires additional assumptions.

B Additional Diagnostics

B.1 Practical notes and future work

Rank floors materially affect the allocation: by construction, $r_{\min} \geq 4$ avoids rank-1 allocations, while the budget must leave enough headroom above the floors for stable-rank heterogeneity to be expressed. Future work includes localized or data-dependent PAC-Bayes priors, a bridge from Gibbs to deterministic-predictor risk, combinations with online pruning, and broader downstream evaluation.

B.2 Direct audit of Assumption 2

For every trained WaterFill layer we compute

$$q_l = \frac{\|B_l A_l\|_F^2}{r_l s_l^2 m_l}.$$

Across four seeds, $\max_l q_l$ ranges from 2.87–3.35 $\times 10^{-6}$ on Pythia-160M and 1.19–2.72 $\times 10^{-5}$ on Pythia-410M; hence one layer-independent empirical value $C^2 = 2.72 \times 10^{-5}$ upper-bounds all layers in all eight runs. The layerwise Pearson correlation between stable rank and adapter energy ranges from 0.406–0.437 and 0.390–0.490, respectively. This is a post-training check of the inequality, not evidence that it holds for arbitrary optimization trajectories.

C Proofs

C.1 Interpretation of Assumption 2

Lemma 5 deliberately separates two statements. The norm integration is deterministic once the three trajectory bounds in the lemma hold. In contrast, the stable-rank scaling of $\|G_l(t)\|_{\mathrm{op}}$ is an additional structural condition; it is not implied by generic gradient descent. This makes the logical role of Assumption 2 explicit and avoids treating the empirical gradient concentration heuristic as a universal theorem. Section B.2 tests the resulting energy inequality directly in eight trained runs.

C.2 Proof of Corollary 10

Let $s_l = \sigma_{\mathrm{eff}}(W_l^{(0)})$ and define normalized weights $w_l = (d_l + k_l)/\Sigma_{\mathrm{total}}$ where $\Sigma_{\mathrm{total}} = \sum_{l'}(d_{l'} + k_{l'})$. Write $\mathbb{E}_w[f] = \sum_l w_l f(s_l)$ for the weighted expectation.

Water-filling objective value. From the proof of Theorem 8, the optimal solution satisfies $r_l^* = s_l/\sqrt{\mu}$ and $c_l = s_l^2(d_l + k_l)$. Therefore:

$$\mathcal{J}^* = \sum_l \frac{c_l}{r_l^*} = \sum_l \frac{s_l^2(d_l + k_l)}{s_l/\sqrt{\mu}} = \sqrt{\mu}\sum_l s_l(d_l + k_l) = \sqrt{\mu} \cdot \Sigma_{\mathrm{total}} \cdot \mathbb{E}_w[s].$$

The budget constraint $\sum_l r_l^*(d_l + k_l) = P$ gives:

$$\frac{1}{\sqrt{\mu}}\sum_l s_l(d_l + k_l) = P \implies \frac{1}{\sqrt{\mu}} = \frac{P}{\Sigma_{\mathrm{total}} \cdot \mathbb{E}_w[s]}.$$

Hence $\sqrt{\mu} = \Sigma_{\mathrm{total}} \cdot \mathbb{E}_w[s]/P$, and substituting back:

$$\mathcal{J}^* = \frac{\Sigma_{\mathrm{total}} \cdot \mathbb{E}_w[s]}{P} \cdot \Sigma_{\mathrm{total}} \cdot \mathbb{E}_w[s] = \frac{\Sigma_{\mathrm{total}}^2 \, (\mathbb{E}_w[s])^2}{P}.$$

Uniform allocation objective value. Let $r_0 = P/\Sigma_{\mathrm{total}}$ (uniform ranks per unit width). Then:

$$\mathcal{J}^{\mathrm{unif}} = \sum_l \frac{c_l}{r_0} = \frac{1}{r_0}\sum_l s_l^2(d_l + k_l) = \frac{\Sigma_{\mathrm{total}}}{r_0} \cdot \mathbb{E}_w[s^2] = \frac{\Sigma_{\mathrm{total}}^2}{P} \cdot \mathbb{E}_w[s^2].$$

Ratio.

$$\frac{\mathcal{J}^*}{\mathcal{J}^{\mathrm{unif}}} = \frac{\Sigma_{\mathrm{total}}^2 \, (\mathbb{E}_w[s])^2/P}{\Sigma_{\mathrm{total}}^2 \, \mathbb{E}_w[s^2]/P} = \frac{(\mathbb{E}_w[s])^2}{\mathbb{E}_w[s^2]}.$$

Using the identity $\mathrm{Var}_w[s] = \mathbb{E}_w[s^2] - (\mathbb{E}_w[s])^2$ and the definition $\mathrm{CV}_w^2[s] = \mathrm{Var}_w[s]/(\mathbb{E}_w[s])^2$:

$$\mathbb{E}_w[s^2] = (\mathbb{E}_w[s])^2(1 + \mathrm{CV}_w^2[s]).$$

Therefore:

$$\frac{\mathcal{J}^*}{\mathcal{J}^{\mathrm{unif}}} = \frac{(\mathbb{E}_w[s])^2}{(\mathbb{E}_w[s])^2(1 + \mathrm{CV}_w^2[s])} = \frac{1}{1 + \mathrm{CV}_w^2[s]},$$

which gives the relative improvement $1 - \mathcal{J}^*/\mathcal{J}^{\mathrm{unif}} = \mathrm{CV}_w^2[s]/(1 + \mathrm{CV}_w^2[s])$. □

D BUDGET AND SURROGATE AUDIT ON PRETRAINED WEIGHTS

The rank-allocation audit recomputes all projection spectra of Pythia-160M and Pythia-410M in float32 and compares the proportionally rounded heuristic, the budget-feasible algorithm, and an exact bounded-integer dynamic program. All use a nominal rank-eight parameter budget and rank limits 4–32. Dynamic programming minimizes the same conditional surrogate over all integer ranks within the budget; it does not optimize downstream PPL. The proportionally rounded rank vectors

Table 5: Allocation audit. Excess is used parameters minus the target budget; gap is the percentage excess surrogate objective over the exact integer optimum.

	Proportional rounding		Budget-feasible rule	
Model	Excess	Gap (%)	Excess	Gap (%)
pythia-160m	0	6.508	0	0.000
pythia-410m	1,024	2.040	-1,024	0.035

are reconstructed from the allocation code; their extrema and means agree with the saved summaries. The 410M heuristic exceeds its nominal budget by 1,024 parameters (0.033%); its reported objective is therefore not a feasible upper bound on the constrained integer optimum. The budget-feasible rule leaves 1,024 parameters unused for that model. Its small surrogate gap does not establish a PPL improvement. These spectra concern pretrained weights, not the unknown optimal task update, and therefore do not verify the task-update spectral-tail assumption.

E TASK-UPDATE SPECTRAL DIAGNOSTIC

We train the same projection weight matrices targeted by the adapters without a low-rank constraint; embeddings, normalization parameters and biases stay frozen. This provides an optimizer-dependent task update, not an oracle best update. The diagnostic uses Pythia-160M and Pythia-410M, two seeds, WikiText-2, 500 steps, batch size four, length 512, AdamW, 50 warmup steps, cosine decay and gradient clipping at one. The initial 2×10^{-4} rate, transferred from the adapter comparison, degrades validation perplexity in all four runs. A subsequent sensitivity comparison fixes 2×10^{-5} for both models and both seeds; all eight runs are reported, rather than selecting successful updates. For each projection we compute singular values of the trained weight change. At ranks 4, 8, 16 and 32 we retain its best rank-r squared residual divided by total squared update norm. We compare this residual with pretrained stable rank across projections within each run. No independence assumption across layers is used to assign a significance level. These observations do not validate the assumed

Table 6: Dense projection updates and spectral proxy diagnostic. ρ_8 is the within-run Spearman association between pretrained stable rank and relative rank-8 update residual. Seeds 1/2 denote 20260920/20260921.

Model	LR	Seed	Initial PPL	Final PPL	ρ_8
160m	2×10^{-4}	1	34.37	155.07	0.252
160m	2×10^{-4}	2	34.37	142.25	0.303
160m	2×10^{-5}	1	34.37	36.48	0.105
160m	2×10^{-5}	2	34.37	31.74	0.074
410m	2×10^{-4}	1	20.23	44.66	-0.091
410m	2×10^{-4}	2	20.23	44.98	-0.136
410m	2×10^{-5}	1	20.23	15.04	-0.054
410m	2×10^{-5}	2	20.23	14.50	0.081

approximation model. Even successful task training does not make pretrained stable rank a strong, universal predictor of relative update-tail error. The conditional allocation result remains a result about its supplied surrogate, and the surrogate's empirical adequacy must be checked separately from its optimizer. A larger or differently regularized dense update could produce different spectra; the current diagnostic establishes neither optimal ranks nor a universal counterexample across tasks.