SIGNS BEAT FLOATS: LOW-RANK DOUBLE-BINARY ADAPTATION FOR STORAGE-EFFICIENT DEPLOYMENT

Anonymous authors
Paper under double-blind review

ABSTRACT

Task-specific LoRA adapters can be stored and exchanged independently of a shared frozen language model, but their floating-point factors remain a substantial deployment cost. We introduce LoRDBA, a low-rank double-binary adapter that represents both factors with binary signs and lightweight channel-wise scales. A post-training factorization initializes the signs, after which our default QAT Freeze procedure adapts only the scales; QAT Full additionally updates latent carriers. We provide a conditional finite-sample reconstruction bound for a sign–noise model. Across three-seed math-reasoning evaluations, LoRDBA achieves favorable compact accuracy–storage trade-offs on LLaMA-2-7B and 13B, with model-dependent margins on Mistral-7B. At approximately 9.54 MiB logical payload on 7B, Freeze improves GSM8K/Minerva means by $6.24/2.24$ points over compact fp16 LoRA with equal total dataset passes. On 13B, it exceeds an SVD rank-2 control initialized from the same source and given the same two-epoch recovery by $3.61/3.34$ points at approximately 14.93 MiB. Separately, packed execution reduces resident adapter memory by $6.90\times$, and adapter-local recomputation brings Freeze training to the fp16-LoRA peak-memory level at higher step time. These findings support storage-efficient adapters produced on server GPUs, not inference speedup or demonstrated on-device training.

1 INTRODUCTION

Large language models are increasingly adapted using parameter-efficient updates rather than full fine-tuning. LoRA enables this paradigm by freezing pretrained weights and injecting low-rank adapter matrices into Transformer projections, reducing the number of trainable parameters (Hu et al., 2022). QLoRA extends this approach to low-bit settings by backpropagating through a frozen quantized base model into LoRA adapters (Dettmers et al., 2023). This base-plus-adapter design is attractive for communication-efficient update distribution and adapter hot-swapping: a shared base can remain fixed while task-specific adapters are trained, exchanged, or selected. Recent multi-adapter serving systems, including S-LoRA and Punica, demonstrate that practical deployment often treats adapters as swappable runtime objects rather than as weights that are always merged into the base model (Sheng et al., 2024; Chen et al., 2024).

In practical low-bit deployments, merging a trained adapter into a quantized base requires materializing full-precision weights or requantizing per task, forfeiting hot-swap capability; it can therefore be useful to serve the adapter as an unmerged side branch (Frantar et al., 2022; Lin et al., 2024). Compression changes the adapter's storage and resident footprint, but does not by itself guarantee lower wall-clock latency. We therefore evaluate accuracy, size, runtime, and training resource cost as separate quantities.

Throughout this paper, *deployment* refers to storing, transmitting, hot-swapping, and executing an adapter after it has been produced. We target adapter banks and repeated distribution over a shared frozen base, where the size of each unmerged adapter matters independently of its one-time production cost. This is an accuracy–storage trade-off with explicitly reported training costs, not a claim that compression reduces total compute. Our experiments produce adapters on server-class GPUs; neither Freeze nor Full is demonstrated as end-to-end constrained-device training. Repeated use motivates the setting but does not establish a measured time or energy break-even point.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

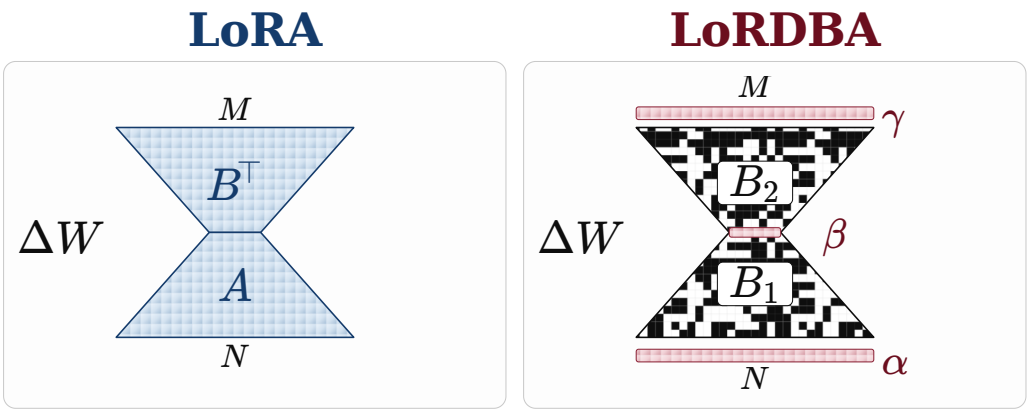


Figure 1: Adapter-mode LoRA versus LoRDBA. Both serve an unmerged update over a shared frozen base. LoRA uses two dense factors; LoRDBA uses two sign matrices with input-, rank-, and output-axis scales. The adapter representation is independent of base precision: a low-bit base is possible, but the accuracy experiments here use bf16.

We introduce LoRDBA, a low-rank double-binary adapter designed for unmerged adapter-mode deployment. LoRDBA replaces both LoRA factors with binary sign carriers and augments them with a compact scale envelope along the input, rank, and output axes. Binary signs encode directional information while lightweight channel-wise scales capture magnitudes; the base model remains frozen and unchanged. Our default pipeline fits a binary representation to a task-trained LoRA adapter and then refines its scales while freezing the signs. An optional Full variant also updates signs through latent carriers and a straight-through estimator. At inference, binary-weight accumulation is interleaved with scaling. Unlike dense-delta binarization (Liu et al., 2024a) or mixed-precision sub-LoRA compression (Mirzaei et al., 2025), both of our low-rank factors use binary carriers.

Our contributions are a compact double-binary adapter representation, a PTQ-initialized scale-refinement procedure, and a conditional reconstruction analysis. Experiments separate accuracy–storage trade-offs from implementation-dependent memory and latency. Three-seed comparisons show clear gains at compact operating points on LLaMA-2-7B and 13B, while Mistral-7B remains close to its compact fp16 control. We report the different adapter-production budgets explicitly, rather than attributing these pipeline-level differences to binarization alone.

2 Preliminaries

Notation. Uppercase Roman letters denote matrices; lowercase boldface denotes vectors. $\{\pm 1\}^{p\times q}$ is the set of sign matrices. $\|\cdot\|_F$ and $\|\cdot\|_{\text{op}}$ are the Frobenius and operator norms. $\text{diag}(\boldsymbol{v})$ constructs a diagonal matrix; $\text{sign}(U)$ is entry-wise sign with $\text{sign}(0)=+1$. The sub-Gaussian Orlicz norm is $\|X\|_{\psi_2} \coloneqq \inf\{t>0 : \mathbb{E}[\exp(X^2/t^2)] \le 2\}$ (Vershynin, 2018).

LoRA. Given a frozen effective weight matrix $W_0 \in \mathbb{R}^{N\times M}$, LoRA reparameterizes the fine-tuning increment as

$$\Delta W = AB^\top \in \mathbb{R}^{N\times M}, \quad A \in \mathbb{R}^{N\times r}, \; B \in \mathbb{R}^{M\times r}, \tag{1}$$

where $r \ll \min(N, M)$ is the adapter rank. Following the column-vector convention, the adapted forward pass for an input $\boldsymbol{x} \in \mathbb{R}^N$ is $\boldsymbol{y} = W_0^\top \boldsymbol{x} + B(A^\top \boldsymbol{x}) \in \mathbb{R}^M$. Thus, each adapted projection incurs two adapter GEMMs with an aggregate arithmetic complexity of $\mathcal{O}(r(N+M))$ per token and stores $16r(N+M)$ bits when both factors are kept in fp16. The base weights remain fixed throughout adaptation.

Adapter-mode deployment. We assess LoRDBA in an unmerged adapter-mode regime (Dettmers et al., 2023; Sheng et al., 2024; Chen et al., 2024). Let W_{fp} denote the original full-precision pretrained weight, and let W_Q be its stored low-bit codebook representation. The effective deployed

base weight is

$$\widetilde{W_0} \coloneqq \mathrm{dequant}(W_Q),$$

When a low-bit W_Q is used, the base need not be materialized in fp16; the trained adapter is served in an un-merged state. The adapted forward pass is

$$\boldsymbol{y} = \widetilde{W_0}^\top \boldsymbol{x} + \Delta W^\top \boldsymbol{x} \;=\; \mathrm{dequant}(W_Q)^\top \boldsymbol{x} + \Delta W^\top \boldsymbol{x}. \tag{2}$$

This factorizes the operation into a base GEMM (executed by a fused dequantize-matmul kernel when the base is quantized) and an independent adapter side branch. The deployment regime is characterized by the following four conditions:

(A1) **Frozen base:** the base model is held fixed and is not updated. It may be stored in a low-bit format (e.g., NF4, GPTQ), but the conditions do not mandate a specific base precision.

(A2) **Adapter-only training:** optimizer states for the base weights are never materialized.

(A3) **Un-merged serving:** the adapter is evaluated as a side branch rather than being merged into a task-specific base weight.

(A4) **Hot-swappability:** multiple task-specific adapters can share the same deployed base (Sheng et al., 2024; Chen et al., 2024).

The un-merged regime decouples base and adapter kernels: the base GEMM runs at the codebook's native precision while the adapter GEMM runs at its own bit width. Reducing adapter bit width directly reduces storage and resident bytes without violating the no-merging constraint; its effect on wall-clock latency depends on the implementation and is measured separately.

Bits per weight (BPW). Following standard practice (Mirzaei et al., 2025), we quantify adapter compression in *bits per weight* (BPW) relative to the reference LoRA parameter count $r_0(N+M)$. An fp16 LoRA adapter of rank r thus has BPW $= 16r/r_0$, and the full-size reference at rank r_0 has BPW $= 16$. The LoRDBA-specific BPW metrics are defined in Section 3.1 once the adapter parameterization is introduced.

3 METHOD

3.1 LOW RANK DOUBLE BINARY ADAPTER

LoRDBA parameterizes the LoRA delta with two binary factors and channel-wise scales across the input, rank, and output axes. The measured pipeline uses a single scale triple ($\ell = 1$); logical storage assumes fp16 scales. We also define a mathematical diagonal-sum extension with $\ell \geq 1$ terms for the reconstruction analysis. This extension is not an evaluated multi-envelope implementation.

Definition 3.1 (Low Rank Double Binary Adapter)**.** *Fix a reference fp16 LoRA rank $r_0 \geq 1$ used only for normalizing storage. A* LoRDBA adapter *of binary carrier rank $R \in \mathbb{N}$ and envelope rank $\ell \geq 1$ for a host projection of shape $N \times M$ is represented as the tuple $\theta = \big(B_1, B_2, (\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)})_{i=1}^{\ell}\big)$ with*

$$B_1 \in \{\pm 1\}^{N \times R},\; B_2 \in \{\pm 1\}^{R \times M},\; \boldsymbol{\alpha}^{(i)} \in \mathbb{R}^N,\; \boldsymbol{\beta}^{(i)} \in \mathbb{R}^R,\; \boldsymbol{\gamma}^{(i)} \in \mathbb{R}^M, \tag{3}$$

whose induced weight update is

$$\Delta W(\theta) = \sum_{i=1}^{\ell} \mathrm{diag}(\boldsymbol{\alpha}^{(i)}) B_1 \,\mathrm{diag}(\boldsymbol{\beta}^{(i)}) B_2 \,\mathrm{diag}(\boldsymbol{\gamma}^{(i)}) \in \mathbb{R}^{N \times M}. \tag{4}$$

The adapted forward pass is $\boldsymbol{y} = W_0^\top \boldsymbol{x} + \Delta W(\theta)^\top \boldsymbol{x}$ and the logical payload cost, with bit-packed carriers and fp16 scales, is

$$\mathrm{bits}(\theta) = R(N+M) + 16\ell(N+R+M). \tag{5}$$

We refer to R as the binary carrier rank *(the inner dimension of the binary matmul), reserving r_0 for the fixed reference LoRA rank used only in BPW normalization.*

The logical payload cost of Eq. (5) induces two BPW metrics:

$$\mathrm{BPW}_{\mathrm{bc}} \coloneqq \frac{R}{r_0}, \qquad \mathrm{BPW}_{\mathrm{tot}} \coloneqq \frac{R(N+M)+16\ell(N+R+M)}{r_0(N+M)}. \tag{6}$$

$\mathrm{BPW}_{\mathrm{bc}}$ counts only the binary carriers; $\mathrm{BPW}_{\mathrm{tot}}$ includes all fp16 scale vectors. At the default operating point ($R{=}r_0, \ell{=}1$), $\mathrm{BPW}_{\mathrm{tot}} \to 1$ as $r_0, N, M \to \infty$. This payload excludes file headers, alignment, and runtime buffers. Logical size in MiB is distinct from serialized and resident size. For the current $r_0 = 16$, $\ell = 1$ experiments, $R = 8$ and $R = 16$ give approximately 1.5007 and 2.0014 total BPW, respectively, after summing over projections. Historical configuration names 1 BPW and 2 BPW are budget labels, not exact total BPW. Main accuracy tables therefore identify the carrier rank and size directly.

The canonical $\ell = 1$ case is $\Delta W = \mathrm{diag}(\boldsymbol{\alpha}) B_1 \,\mathrm{diag}(\boldsymbol{\beta}) B_2 \,\mathrm{diag}(\boldsymbol{\gamma})$ (Figure 1). For $\ell \geq 2$, Eq. (4) sums ℓ rank-R terms; inference cost grows linearly in ℓ, while binary-carrier storage is unchanged. Here ℓ counts diagonal-sum terms, not the magnitude-envelope rank in MDBF (Ichikawa et al., 2025), whose product of two rank-ℓ envelopes is a different parameterization. They coincide at $\ell = 1$ after combining the inner-axis scales; we do not transfer multi-envelope storage or rank claims between them.

3.2 LoRDBA Training

Source adapter and PTQ initialization. We first train a task-specific fp16 LoRA adapter with the base frozen (two epochs in the current accuracy experiments). PTQ-LoRDBA then fits Eq. (4) to its update in Frobenius norm without additional task data. The single-envelope DBF implementation starts from seeded random continuous factors, uses 100 outer iterations, and fits magnitude scales. Algorithm 1 specifies the complete construction pipeline, while Appendix D gives the internal factor-update equations. Its task-adapted signs and scales initialize Freeze. The implementation factorizes the unscaled factor product and preserves the source adapter's LoRA multiplier exactly once, outside the factorization; source rank r_0 need not equal carrier rank R. The appendix also distinguishes this implementation from the reference SVD initialization and consensus-ADMM analysis.

Default: QAT Freeze. Starting from the PTQ output, we fix B_1, B_2 and optimize only $(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)})_{i=1}^{\ell}$ on the downstream loss. The trainable count per projection is $\ell(N + R + M)$. Current math experiments use two epochs of scale refinement with cross-entropy (CE) on the tokenized training sequence. Scales are optimized in fp32. Logical payload assumes fp16 scales and packed carriers, whereas audit checkpoints may retain fp32 or unpacked tensors. Binary carriers can be packed during training; adapter-local recomputation reduces saved intermediate activations.

Optional extension: QAT Full. Full also maintains real-valued latent carriers H_1, H_2 whose signs give B_1, B_2, and uses a smooth-sign STE for their backward pass (Leng et al., 2018; Liu et al., 2020; Boža & Macko, 2025a). Its trainable count is $R(N + M) + \ell(N + R + M)$, although its exported representation is unchanged. It is a higher-state extension, not the procedure behind the main Freeze accuracy rows. Previous Full/Freeze/Scratch observations use differing schedules and are reported separately in Appendix G; they do not establish equivalence between these modes.

The DBF-Factorize call is the executed randomized-start routine detailed in Appendix D: it alternates the two factor updates for 100 outer iterations, uses three projection/dual steps per factor update, and extracts the three scale vectors from the final factors. The source multiplier c is not absorbed into the fitting target and is applied exactly once in the adapter wrapper. Optimizer, scheduler, and checkpoint details are given in Appendix K.

3.3 Adapter-Mode Inference Kernel

For an input batch $X \in \mathbb{R}^{T \times N}$ and a canonical ($\ell{=}1$) LoRDBA adapter, the evaluation order is

$$X\Delta W(\theta) = \Big(\Big(\Big(\big(XD_{\boldsymbol{\alpha}}\big)B_1\Big)D_{\boldsymbol{\beta}}\Big)B_2\Big)D_{\boldsymbol{\gamma}}, \tag{7}$$

Algorithm 1 Default LoRDBA construction: single-envelope PTQ initialization followed by QAT Freeze

Require: Frozen base W_0; trained source factors A_0, B_0; multiplier $c = \alpha_0/r_0$
Require: Carrier rank R; downstream data $\mathcal{D}$; random seed s
Ensure: Deployable adapter $\Theta = (B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, c)$

$\overline{\Delta W}_0 \leftarrow A_0 B_0^\top$ ▷ unscaled source update
$(B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \leftarrow \textsc{DBFFactorize}(\overline{\Delta W}_0, R, s, 100)$ ▷ Appendix D
$\boldsymbol{u} \leftarrow (\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$; freeze W_0, B_1, B_2
for $e = 1, 2$ **do**
  for minibatch $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{D}$ **do**
    $\Delta W_{\boldsymbol{u}} \leftarrow \mathrm{diag}(\boldsymbol{\alpha}) B_1 \,\mathrm{diag}(\boldsymbol{\beta}) B_2 \,\mathrm{diag}(\boldsymbol{\gamma})$
    $\mathcal{L} \leftarrow \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}; W_0, c\Delta W_{\boldsymbol{u}}), \boldsymbol{y})$ ▷ full-sequence CE
    $\boldsymbol{u} \leftarrow \textsc{AdamW}(\boldsymbol{u}, \nabla_{\boldsymbol{u}} \mathcal{L}, \eta_t)$ ▷ $\eta_0 = 5 \times 10^{-5}$
  end for
end for
return $\Theta = (B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, c)$

where $D_{\boldsymbol{v}} \coloneqq \mathrm{diag}(\boldsymbol{v})$. The diagonal products are element-wise floating-point rescalings; the inner matmuls $(\cdot)B_1$ and $(\cdot)B_2$ are binary-weight GEMMs. Each column reduces to sign-accumulation:

$$(XB)_{tj} = \sum_{i:B_{ij}=+1} X_{ti} - \sum_{i:B_{ij}=-1} X_{ti}, \tag{8}$$

The inner products use additions and subtractions, while the scales remain floating-point. Our custom CUDA prototype reads packed carriers directly. The two adapter products and intermediate rescalings still incur implementation-dependent overhead; fewer stored bits do not imply lower latency. Appendix J details the prototype and the earlier PyTorch fallback, and Section 5.2 reports their evidence separately.

4 Theoretical Analysis

We study a sufficient probabilistic condition under which a double-binary representation approximates a low-rank update. This is a reconstruction statement for the specified factor model, not a downstream-accuracy or optimizer guarantee. The formal model and proof are in Appendix C. Under the independence and tail assumptions below, small residuals relative to factor magnitudes control reconstruction error; separation from zero alone does not establish these assumptions.

Theorem 4.1 (Informal LoRDBA expressivity)**.** *Let $A \in \mathbb{R}^{N\times r}$ and $B \in \mathbb{R}^{M\times r}$ be fp16 LoRA factors. Suppose that, after fixing the usual positive column-rescaling gauge of LoRA, their entries can be written as*

$$A_{ik} = \mu_A \sigma^A_{ik} + \xi^A_{ik}, \qquad B_{jk} = \mu_B \sigma^B_{jk} + \xi^B_{jk},$$

where $\mu_A, \mu_B > 0$, the sign arrays are in $\{\pm 1\}$, and the residuals are independent, mean-zero, sub-Gaussian with scale at most ζ, and independent of the signs. For the relative bound, assume the signs are jointly i.i.d. Rademacher, $NM > 8$, $\delta \in (0, 1 - 8/(NM))$, $\zeta \le \max(\mu_A, \mu_B)$, and $\log(6NM/\delta) \le c_1 r$ for a universal constant $c_1 > 0$.

Let $\theta^\star$ be the canonical single-envelope LoRDBA adapter with binary carriers σ^A and $(\sigma^B)^\top$, unit input/output scales, and rank-axis scale $\mu_A\mu_B\mathbf{1}_r$. Then, for a universal constant $C' > 0$, with probability at least $1 - \delta - 8/(NM)$,

$$\frac{\|AB^\top - \Delta W(\theta^\star)\|_F}{\|\Delta W(\theta^\star)\|_F} \le C' \frac{\zeta}{\min(\mu_A, \mu_B)} \sqrt{\log(2NM/\delta)}. \tag{9}$$

The full statement, including the absolute Frobenius bound, the observed-sign version using $\mathrm{sign}(A)$ and $\mathrm{sign}(B)$, and the monotonic extension to envelope rank $\ell \ge 1$, is given in Appendix C.

Scope and relation to training. The bound concerns the canonical single-envelope construction and normalizes error by its reconstructed update, not by $\|AB^\top\|_F$. Its sub-Gaussian scale ζ is a

distributional tail parameter, not a sample standard deviation. Learned SVD factors need not satisfy the required independence or Rademacher assumptions; empirical summaries (Appendix C.6) are descriptive, not calibrated bounds. We do not infer a numerical reconstruction error, downstream-accuracy plateau, or PTQ optimizer outcome from this theorem. Increasing ℓ enlarges a nested feasible set and cannot worsen its *minimum* reconstruction objective, but does not guarantee improved optimization or task accuracy. Freeze and Full are empirical refinement procedures.

5 EXPERIMENTS

Under the adapter-mode conditions (A1)–(A4), we evaluate LoRDBA along three distinct axes:

- **RQ1 Accuracy** (Section 5.1): Does LoRDBA match or exceed fp16 LoRA at matched adapter size?
- **RQ2 Residency and decode latency** (Section 5.2): How much resident adapter memory does packing save, and what is the pure-decode latency cost?
- **RQ3 Training cost** (Section 5.3): What is the peak memory and wall-clock overhead relative to the corresponding fp16 LoRA loop?

Models, data, and metrics. The replicated comparisons use LLaMA-2-7B, LLaMA-2-13B, and Mistral-7B. The base is frozen in bf16 for accuracy and training measurements, and adapters remain unmerged. Source LoRA targets all seven Transformer projections ($q, k, v, o, \mathrm{gate}, \mathrm{up}, \mathrm{down}$), with rank $r_0 = 16$ and LoRA $\alpha = 32$ (multiplicative scale 2). The reference $\alpha = 2r$ rule also applies to direct compact controls. Current math adapters use MetaMathQA (395,000 examples), maximum sequence length 1,024, and full-sequence CE. We evaluate GSM8K (8-shot exact match, 1,319 items) and Minerva Math (4-shot, seven categories weighted by item counts, 5,000 items) with `lm-evaluation-harness` (Gao et al., 2023). Scores are percentages, reported as mean $\pm$ sample standard deviation over three seeds (0–2). Appendix K separates these settings from earlier 3B and non-math experiments.

Comparators and training budgets. Compact fp16 LoRA and LoRAQuant (Mirzaei et al., 2025) are the main reported comparators. The 7B ~ 9.54-MiB comparison gives direct rank-2 LoRA four epochs, versus two source-training epochs plus two Freeze epochs. This matches dataset passes, not FLOPs or wall-clock time. The 13B/Mistral direct rank-2 controls use two epochs. An additional 13B control truncates the same seed-matched rank-16 source to rank 2 by SVD, then trains its floating factors with the same two-epoch recovery schedule and learning rate as Freeze. This controls source and recovery exposure, but not trainable parameter count or optimization geometry. The 7.152-MiB LoRAQuant comparison matches source and logical packed-code-plus-quantizer-parameter size, but supplies no additional two-epoch CE recovery. Historical factor-wise BitDelta-style records remain separate in Appendix G; Appendix M summarizes which comparison axes are controlled.

Hardware and accounting. Experiments use server-class NVIDIA H100, H100 NVL, and B200 GPUs. Logical adapter payload (MiB), resident adapter memory (MiB), and peak training allocation (GiB) measure different objects. The serving diagnostic uses an fp16 base and fixed decoding choices. Appendix K gives additional protocol details.

Accuracy uses saved checkpoints; logical sizes assume packed signs and fp16 scales/factors, not the checkpoint format. Freeze accuracy uses a bf16 dense side branch materialized from fp32 scales, separately from the packed CUDA timing path. Casting each checkpoint's scales to the counted fp16 deployment format has not been separately accuracy-tested.

5.1 ADAPTER ACCURACY (RQ1)

At the 9.54-MiB tier, QAT Freeze has seed-averaged gains of $+6.24$ GSM8K and $+2.24$ Minerva points over the fresh four-epoch fp16 $r{=}2$ control. At the 7.152-MiB packed-formula tier, its gains over LoRAQuant are $+4.02$ and $+0.69$ points. The paired descriptive interval excludes zero on GSM8K and overlaps zero on Minerva. The smaller $R = 8$ Freeze adapter also has means $+2.27/+0.73$ points above the larger four-epoch fp16 control. Per-seed scores, bootstrap scope, and provenance are reported in Appendix G.

Table 1: Fresh three-seed LLaMA-2-7B compact comparison. Values are mean ± sample standard deviation; task metrics are reported separately.

Method	Training / compression	MiB	GSM8K	Minerva	n
LoRAQuant r=16	ρ=0.6763, post-hoc	7.152	39.85±1.52	8.25±0.61	3
QAT Freeze R=8	2+2 epochs	7.152	43.87±1.54	8.93±0.08	3
fp16 LoRA r=2	4 epochs	9.53	41.60±1.92	8.20±0.53	3
QAT Freeze R=16	2+2 epochs	9.54	47.84±1.20	10.44±0.68	3

GSM8K uses 8-shot `exact_match,strict-match`×100 over 1,319 items; Minerva uses 4-shot `math_verify,none`×100, sample-weighted over seven categories and 5,000 items. Freeze starts from a two-epoch seed-matched r=16 source and runs two QAT epochs. The $R = 8$/LoRAQuant pair is equal only under the preregistered implementable packed-code-plus-quantizer-parameter formula (difference 0.0028%), not serialized file size, memory, or latency; the $R = 16$/fp16 pair uses logical storage. The latter pair has four total dataset passes but does not match FLOPs, energy, or wall-clock time. For the $R = 8$ pair, paired hierarchical descriptive intervals are $[2.02, 5.86]$ on GSM8K and $[-0.09, 1.52]$ on Minerva.

Table 2: Source and compact-adapter comparisons across model families. Mean ± sample standard deviation over seeds 0–2. SVD+recovery and Freeze on 13B share the source and recovery schedule; direct rank-2 LoRA has only two training epochs. Sizes are logical payloads, not serialized checkpoint sizes.

Model	Method	MiB	Epochs	GSM8K	Minerva
LLaMA-2-13B	Source fp16 $r = 16$	119.38	2	62.22±0.68	17.04±0.11
	Direct fp16 $r = 2$	14.92	2	53.65±0.90	12.37±1.00
	SVD $r = 2$ + recovery	14.92	$2 + 2$	54.64±1.09	11.31±0.78
	QAT Freeze $R = 16$	14.93	$2 + 2$	58.25±1.36	14.65±0.43
Mistral-7B	Source fp16 $r = 16$	80.00	2	63.94±0.92	19.89±0.11
	Direct fp16 $r = 2$	10.00	2	60.55±0.77	17.90±0.46
	QAT Freeze $R = 16$	10.01	$2 + 2$	60.02±0.70	17.55±0.34

On 13B, Freeze exceeds direct rank-2 LoRA by $4.60/2.27$ points and the same-source SVD+recovery control by $3.61/3.34$ points (GSM8K/Minerva). The latter result shows that the observed advantage over this floating-point control remains after matching source and recovery schedule; it is not a wall-clock-matched or universal representation comparison. Relative to the much larger rank-16 source, Freeze loses $3.97/2.39$ points while using approximately one eighth of the logical payload. On Mistral, its gaps are $-0.53/-0.35$ points to direct rank-2 LoRA and $-3.92/-2.35$ to the source. Thus the compact trade-off is model-dependent. We have no completed same-source SVD+recovery comparison on Mistral. Appendix G provides per-seed results and the scope of the controls.

Additional task evidence. Across five additional multiple-choice tasks on LLaMA-2-7B/13B and Qwen2.5-7B, Freeze stays within 1.32 points of the rank-16 source means at approximately one eighth of the logical payload. Separately, on Magicoder-trained LLaMA-2-7B, scale refinement raises HumanEval pass@1 from 1.69 (PTQ) to 15.68 at 9.538 MiB. Appendix L reports the complete comparator tables, variability, and scope of these supplementary results.

5.2 Adapter Residency and Pure-Decode Latency (RQ2)

We measure unmerged LLaMA-2-7B decoding on H100 NVL at batch 1, context length 512, and 128 generated tokens. Both methods use the same base model and decoding workload, with separate method-specific adapter checkpoints. Each implementation passed token-output and logit checks against its own dequantized-dense reference before timing; all 224 LoRDBA adapters were converted. These checks establish representation fidelity, not equality between the two trained adapters.

The runtime representation reduces resident adapter memory by $6.90\times$ (Table 3). The median and p95 latency ratios are 1.129 and 1.123. An fp16 sample of 316.944 ms raises its mean (127 of 128 samples are below 30 ms); we retain it and do not use the mean ratio 1.030 as an overhead guarantee. The one-pair diagnostic supports compact residency, not latency equivalence or speedup. The earlier

Table 3: Custom-CUDA pure-decode diagnostic: one process pair. TPOT is time per output token. Percentiles summarize tokens within a process, not independent repeated runs.

Adapter	Resident MiB	Mean ms	p50 ms	p95 ms
fp16 LoRA $r = 16$	76.25	25.361	23.040	23.210
LoRDBA custom CUDA	11.05	26.128	26.002	26.070

Table 4: Controlled H100 training calibration. Trainable parameters are fp32; fp16 LoRA denotes its exported adapter format. The three-step diagnostic is not a full-run timing estimate.

Training representation	Peak allocated GiB	ms/step
fp16 LoRA	36.638	477.6
Freeze, float carriers	46.867	–
Freeze, packed carriers	46.795	–
Freeze, packed + adapter-local recomputation	36.365	680.9
Full, latent carriers + STE	55.792	599.4

PyTorch fallback was much slower (413.446 ms vs. 27.266 ms); its separate repeated measurement is preserved in Appendix J, not pooled with the custom path. Fusion and projection-shape specialization remain optimization opportunities, not measured improvements.

5.3 Training Resource Cost (RQ3)

Freeze optimizes few parameters, but saved adapter intermediates can still dominate peak memory. We isolate this effect with a bf16 LLaMA-2-7B base on H100 NVL, batch 4, sequence length $1{,}024$, and AdamW. After one warm-up step we reset CUDA peak statistics and measure three steps; the memory statistic is the maximum allocation, not the mean.

Packing alone has little effect on the measured peak. Adding adapter-local recomputation brings Freeze to approximately the fp16-LoRA memory level, at $1.43\times$ step time (Table 4). Full maintains latent carriers and optimizer state and has higher peak allocation. These are refinement-loop measurements, not the cost of producing a trained adapter from scratch. That cost includes source LoRA training, PTQ, and recovery; when reusing an existing source, only the latter two are incremental (Appendix M). Neither few trainable scales nor equal dataset passes implies equal total training time. Earlier B200 whole-run records use a different schedule and do not supply a matched end-to-end timing comparison for the current Freeze results.

5.4 Ablation Scope

Appendix G retains previous mode, rank, and bit-budget observations separately from the replicated main comparisons. Differences in source training and refinement schedules prevent treating those observations as controlled evidence that Full and Freeze are equivalent, or that changing envelope rank alone improves memory or accuracy. The present main result concerns the PTQ-initialized Freeze pipeline; the theory does not substitute for these empirical controls.

6 Related Work

Low-rank adaptation. Beyond LoRA (Hu et al., 2022), DoRA separates weight magnitude and direction (Liu et al., 2024b), PiSSA initializes from principal components of the base weight (Meng et al., 2024), and LoRA+ changes factor learning-rate ratios (Hayou et al., 2024). VeRA shares frozen random factors and trains scaling vectors (Kopiczko et al., 2024); it is particularly relevant to both training state and adapter storage. Freeze instead retains task-fitted binary factors from a trained source. These methods are relevant alternatives, not excluded by our deployment setting, but are not empirically ranked here. Our same-source SVD control compresses a trained update; it is not PiSSA. Appendix M details these distinctions.

Base-model quantization. QLoRA trains adapters through a frozen quantized base (Dettmers et al., 2023); LoftQ jointly selects quantization and adapter initialization (Li et al., 2024). QA-LoRA uses group-wise adaptation that can be integrated into a quantized model (Xu et al., 2024a). These differ from compressing an unmerged adapter over an unchanged shared base. Base PTQ (Frantar et al., 2022; Lin et al., 2024) is conceptually compatible with LoRDBA, but our main accuracy measurements use a bf16 base.

Adapter compression. BitDelta (Liu et al., 2024a) binarizes the full fine-tuning delta ΔW with a learned scalar per weight matrix, producing a $\Theta(NM)$-bit delta rather than preserving its low-rank factors. LoRAQuant (Mirzaei et al., 2025) is a post-training method that re-decomposes $AB^\top$ via SVD, optimizes reconstruction, and quantizes sub-LoRA factors with mixed 1-/2-bit codes and floating-point quantizer parameters. LoRDBA uses binary carriers throughout both low-rank factors and refines channel-wise scales. The historical factor-wise BitDelta-style variants in the appendix are not the original full-delta method.

Binary matrix factorization. The ± 1 parameterization connects to binary networks (Courbariaux et al., 2016; Rastegari et al., 2016; Liu et al., 2020) and full-LLM binarization (BitNet (Wang et al., 2023), OneBit (Xu et al., 2024b)). DBF (Boža & Macko, 2025b) already combines two sign matrices, channel-wise scales, and scale-only refinement; MDBF (Ichikawa et al., 2025) extends the magnitude envelopes. LoRDBA adapts this factorization to task-trained low-rank updates and unmerged adapter serving, with a conditional reconstruction analysis (Theorem 4.1) and measurements that separate accuracy, payload, residency, and training cost. Scale-only refinement itself is not a new optimization principle. Smooth-sign STE training is an optional extension (Leng et al., 2018; Boža & Macko, 2025a).

7 Conclusion

LoRDBA represents both low-rank factors with binary signs and compact scales. PTQ-initialized Freeze yields favorable compact trade-offs on LLaMA-2-7B and 13B; the 13B advantage persists against a same-source SVD rank-2 control with matched recovery schedule. Mistral remains within 0.6 points of direct compact fp16 LoRA. Compression does incur accuracy losses relative to the much larger rank-16 sources. Separately, packed execution reduces resident adapter memory, and local recomputation lowers Freeze's training peak. The representation, conditional reconstruction analysis, and measured trade-offs support storage-efficient deployment after off-device adapter production.

Limitations. The theorem assumes independent sub-Gaussian residuals and, for the relative bound, independent Rademacher signs; learned factors are not established to obey that model. The primary accuracy evidence covers two math-reasoning evaluations, three seeds, and a small set of storage points; supplementary coverage is detailed in Appendix L. Production paths and recovery budgets differ across comparators, and superiority at equal measured training time or over newer PEFT methods is not established. The CUDA timing evidence is a single process pair, and the controlled memory experiment is a short calibration. Neither establishes inference speedup, multi-adapter throughput, or constrained-device training feasibility.

AI USE STATEMENT

Generative AI tools assisted with experiment design and critique, implementation and debugging of experiment and monitoring code, interpretation and cross-checking of results, review of theoretical claims and their scope, manuscript drafting and revision, translation, and formatting. AI-generated reviews were used to identify issues for investigation. Numerical results reported as measurements come from experiment artifacts rather than generated estimates. The authors are responsible for the final content, claims, and artifacts.

ETHICS STATEMENT

Compact adapters can reduce storage and communication costs, but compression does not itself establish privacy, energy, or latency benefits. The method may also be applied to harmful task adaptations and should retain the safety controls of the underlying models. We use existing datasets and model checkpoints; no new human-subject data collection is reported.

REPRODUCIBILITY STATEMENT

The appendix specifies the adapter representation, mathematical assumptions and proofs, evaluation metrics, seed-level results, and training and measurement conditions. The experiment workflow records configuration and artifact identities and uses training checkpoints and incremental evaluation outputs. Logical payload, runtime residency, and peak training allocation are reported separately. Historical and current-protocol results are distinguished to avoid pooling incompatible measurements.

REFERENCES

H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka–łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

H. Attouch, J. Bolte, and B. F. Svaiter. Convergence of descent methods for semi-algebraic and tame problems: Proximal algorithms, forward–backward splitting, and regularized Gauss–Seidel methods. *Mathematical Programming*, 137(1–2):91–129, 2013.

Hicham Badri and Appu Shaji. gemlite: Cuda kernels for low-bit matrix multiplication, 2023. `https://github.com/mobiusml/gemlite`.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *AAAI Conference on Artificial Intelligence*, 2020.

J. Bolte, A. Daniilidis, and A. Lewis. The Łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM Journal on Optimization*, 17(4):1205–1223, 2007.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

Vladimír Boža and Vladimír Macko. Two sparse matrices are better than one: Sparsifying neural networks with double sparse factorization. In *International Conference on Learning Representations (ICLR)*, 2025a.

Vladimír Boža and Vladimír Macko. Addition is almost all you need: Compressing large language models with double binary factorization. In *International Conference on Machine Learning (ICML)*, 2025b.

Lequn Chen, Zihao Ye, Yongji Wu, Danyang Zhuo, Luis Ceze, and Arvind Krishnamurthy. Punica: Multi-tenant LoRA serving. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

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

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to $+1$ or -1. *arXiv preprint arXiv:1602.02830*, 2016.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*, 2022.

Leo Gao, Jonathan Tow, Stella Biderman, et al. A framework for few-shot language model evaluation. `https://github.com/EleutherAI/lm-evaluation-harness`, 2023.

Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *International Conference on Machine Learning (ICML)*, 2024.

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.

Yuma Ichikawa, Yoshihiko Fujisawa, Yudai Fujimoto, Akira Sakai, and Katsuki Fujisawa. More than bits: Multi-envelope double binary factorization for extreme quantization. *arXiv preprint arXiv:2512.24545*, 2025.

Dawid J. Kopiczko, Tijmen Blankevoort, and Yuki M. Asano. VeRA: Vector-based random matrix adaptation. In *International Conference on Learning Representations (ICLR)*, 2024.

Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. Extremely low bit neural network: Squeeze the last bit out with ADMM. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Yixiao Li, Yifan Yu, Chen Liang, Pengcheng He, Nikos Karampatziakis, Weizhu Chen, and Tuo Zhao. LoftQ: LoRA-fine-tuning-aware quantization for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *Proceedings of Machine Learning and Systems*, 6, 2024.

James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. BitDelta: Your fine-tune may only be worth one bit. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024a.

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

Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. ReActNet: Towards precise binary neural network with generalized activation functions. In *European Conference on Computer Vision (ECCV)*, 2020.

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.

Amir Reza Mirzaei, Yuqiao Wen, Yanshuai Cao, and Lili Mou. LoRAQuant: Mixed-precision quantization of LoRA to ultra-low bits. *arXiv preprint arXiv:2510.26690*, 2025.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. XNOR-Net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, 2016.

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

Ying Sheng, Shiyi Cao, Dacheng Li, Coleman Hooper, Nicholas Lee, Shuo Yang, Christopher Chou, Banghua Zhu, Lianmin Zheng, Kurt Keutzer, Joseph Gonzalez, and Ion Stoica. S-LoRA: Serving thousands of concurrent LoRA adapters. In *Proceedings of Machine Learning and Systems (MLSys)*, 2024.

Andreas Themelis and Panagiotis Patrinos. Douglas–Rachford splitting and ADMM for nonconvex optimization: Tight convergence results. *SIAM Journal on Optimization*, 30(1):149–181, 2020.

Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge University Press, 2018.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of ADMM in nonconvex nonsmooth optimization. *Journal of Scientific Computing*, 78(1):29–63, 2019.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. QA-LoRA: Quantization-aware low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2024a.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. OneBit: Towards extremely low-bit large language models. *arXiv preprint arXiv:2402.11295*, 2024b.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2019.

APPENDIX CONTENTS

A Symbol Glossary **13**

B Self-contained probabilistic background **13**

C Conditional Reconstruction: Proof of Theorem 4.1 **15**
C.1 Entry-wise expansion . . . 16
C.2 Sub-exponential tail of each entry . . . 17
C.3 Frobenius bound by union . . . 18
C.4 Chebyshev lower bound on the signal . . . 19
C.5 Sign consistency and observed-sign reconstruction . . . 21
C.6 Gauge fixing and descriptive magnitude dispersion . . . 23

D PTQ Initialization and Consensus-ADMM Analysis **23**

E Convergence of PTQ-LoRDBA **25**
E.1 Finite stabilisation under a sign margin . . . 26
E.2 KL convergence of a regular fixed-penalty tail . . . 27

F QAT Full: Smooth-Sign STE Extension **28**

G Training Mode Comparison **28**

H Extended Numerical Results **31**

I Pseudocode **32**
I.1 Closed-form scale updates for PTQ-LoRDBA . . . 32

J Inference-Kernel Details **36**

K Implementation Details and Compute **38**

L Additional Task Evidence **38**
L.1 Scale-only recovery on HumanEval . . . 39
L.2 Preservation on additional multiple-choice tasks . . . 40

M Adapter-Production Budgets and Baseline Scope **41**

N Additional Figures **42**

A SYMBOL GLOSSARY

Table 5 collects the principal symbols used throughout the paper.

B SELF-CONTAINED PROBABILISTIC BACKGROUND

This appendix collects the elementary concentration facts used by the expressivity proof. Apart from Markov's inequality, every estimate below is proved from scratch using only Cauchy–Schwarz and series manipulations, so the proof of Theorem 4.1 does not depend on any external reference. Throughout we work with real-valued random variables defined on a common probability space, write $C, c, c_1, c_2, \ldots$ for finite, strictly positive numerical constants whose value may change from line to line and which are independent of all model parameters, and denote

$$\|X\|_{\psi_2} \coloneqq \inf\big\{t > 0 : \mathbb{E}\exp(X^2/t^2) \le 2\big\}, \qquad \|X\|_{\psi_1} \coloneqq \inf\big\{s > 0 : \mathbb{E}\exp(|X|/s) \le 2\big\}. \tag{10}$$

Lemma B.1 (Sub-Gaussian tail and second moment)**.** *Let X be a real random variable with $\|X\|_{\psi_2} \le K$. Then*

$$\mathbb{E}[X^2] \;\le\; K^2, \qquad \Pr[|X| \ge t] \;\le\; 2\exp(-t^2/K^2),\ t \ge 0, \tag{11}$$

and for every integer $p \ge 1$, $(\mathbb{E}|X|^p)^{1/p} \le CK\sqrt{p}$ with a numerical constant C.

Table 5: Glossary of the principal symbols used in the paper.

Symbol	Meaning	First used
N, M	in-/out-features of the host projection	Section 2
r_0	reference fp16 LoRA rank used for BPW normalisation	Section 2
R	binary carrier rank of LoRDBA (inner dim of B_1, B_2)	Definition 3.1
ℓ	envelope rank of LoRDBA	Definition 3.1
$W_0 \in \mathbb{R}^{N\times M}$	frozen base-model weight matrix	Section 2
A, B	fp16 LoRA factors of rank r_0	Eq. (1)
ΔW	fine-tuning weight update (either LoRA or LoRDBA)	Section 2
$B_1 \in \{\pm1\}^{N\times R}$, $B_2 \in \{\pm1\}^{R\times M}$	binary factors of LoRDBA	Definition 3.1
$\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)}$	channel-wise fp16 scale vectors ($i \in [\ell]$)	Definition 3.1
$\mathrm{sign}(\cdot), \mathrm{diag}(\cdot)$	element-wise sign and diagonal operators	Section 2
$\odot$	Hadamard (element-wise) product	Section 2
$\mathrm{BPW}_{\mathrm{bc}}, \mathrm{BPW}_{\mathrm{tot}}$	binary-carrier and total bits per weight	Eq. (6)
U_k, M_k, Y_k	continuous, discrete, scaled dual iterates of PTQ-LoRDBA	App. D
$n_1 = NR, n_2 = RM$	number of entries in block $k \in \{1, 2\}$	App. D
$\rho, \widetilde{\rho}_k = \rho/n_k$	ADMM penalty, per-block scaling	Eq. (44)
κ	smooth-sign STE temperature	Eq. (64)
$\|\cdot\|_F, \|\cdot\|_{\psi_2}, \|\cdot\|_{\psi_1}$	Frobenius, sub-Gaussian and sub-exponential Orlicz norms	Sections 4, B
σ^A, σ^B	latent signs in the factor decomposition	App. C
μ_A, μ_B	mean magnitudes of fp16 LoRA factors A, B	App. C
ξ^A, ξ^B	zero-mean sub-Gaussian residuals ($\|\xi\|_{\psi_2} \le \zeta$)	App. C
$\theta^\star$	canonical LoRDBA reconstruction (Eq. (15))	App. C

Proof. The inequality $1 + u \le e^u$ at $u = X^2/K^2$ and the definition of $\|\cdot\|_{\psi_2}$ give $1 + \mathbb{E}[X^2]/K^2 \le \mathbb{E}\exp(X^2/K^2) \le 2$, hence $\mathbb{E}[X^2] \le K^2$. Markov's inequality applied to $\exp(X^2/K^2)$ at level $\exp(t^2/K^2)$ yields the tail bound. For the moment bound, integrate the tail:

$$\mathbb{E}|X|^p = p\int_0^\infty t^{p-1}\Pr[|X| \ge t]dt \le 2p\int_0^\infty t^{p-1}e^{-t^2/K^2}dt = pK^p\Gamma(p/2) \le (CK)^p p^{p/2},$$

where the last step uses Stirling. Taking the p-th root proves the moment bound. □

Lemma B.2 (MGF bound and sub-Gaussian sum)**.** *If $\mathbb{E}X = 0$ and $\|X\|_{\psi_2} \le K$, then $\mathbb{E}\exp(\lambda X) \le \exp(C\lambda^2K^2)$ for every $\lambda \in \mathbb{R}$. If furthermore $X_1, \dots, X_n$ are mutually independent, mean-zero with $\|X_i\|_{\psi_2} \le K$, and $a_1, \dots, a_n \in \mathbb{R}$ are deterministic, then for every $u \ge 0$,*

$$\Pr\left[\left|\sum_{i=1}^n a_iX_i\right| \ge CK\|a\|_2\sqrt{u}\right] \;\le\; 2e^{-u}. \tag{12}$$

Proof. Expanding the exponential as a Taylor series and using $\mathbb{E}X = 0$ gives

$$\mathbb{E}e^{\lambda X} = 1 + \sum_{p\ge2}\frac{\lambda^p\mathbb{E}X^p}{p!}.$$

By Lemma B.1, $|\mathbb{E}X^p| \le \mathbb{E}|X|^p \le (CK\sqrt{p})^p$. Hence

$$\sum_{p\ge2}\frac{|\lambda|^p|\mathbb{E}X^p|}{p!} \le \sum_{p\ge2}\frac{(C|\lambda|K\sqrt{p})^p}{p!} \le C_1\lambda^2K^2\exp(C_2\lambda^2K^2),$$

where the last inequality is the standard consequence of $p! \ge (p/e)^p$ and splitting the series at $p \simeq \lambda^2K^2$. Since $1 + s\exp(s') \le \exp(C(s + s'))$ for $s, s' \ge 0$, this yields $\mathbb{E}e^{\lambda X} \le \exp(C'''\lambda^2K^2)$ for a universal constant C'''. By independence, $\mathbb{E}\exp(\lambda\sum_i a_iX_i) = \prod_i \mathbb{E}e^{\lambda a_iX_i} \le \exp(C\lambda^2K^2\|a\|_2^2)$. Chernoff's inequality $\Pr[\sum a_iX_i > t] \le \inf_{\lambda>0} e^{-\lambda t}\mathbb{E}e^{\lambda\sum a_iX_i}$ optimised at $\lambda = t/(2CK^2\|a\|_2^2)$ gives $\Pr[\sum a_iX_i > t] \le \exp(-ct^2/(K^2\|a\|_2^2))$. A symmetric bound on the negative tail and adjusting constants yields Eq. (12). □

Lemma B.3 (Sub-exponential product and Bernstein bound)**.** *If X, Y are independent real random variables with $\|X\|_{\psi_2} \le K$ and $\|Y\|_{\psi_2} \le L$, then $\mathbb{E}[XY] = \mathbb{E}X\mathbb{E}Y$ and the product XY is sub-exponential with $\|XY\|_{\psi_1} \le KL$. Moreover, if $W_1, \dots, W_n$ are mutually independent, mean-zero, and $\|W_i\|_{\psi_1} \le L_0$, then for every $u \ge 0$,*

$$\Pr\left[\Big|\sum_{i=1}^{n} W_i\Big| \ge t\right] \;\le\; 2\exp\Big(-c\min\Big(\frac{t^2}{nL_0^2}, \frac{t}{L_0}\Big)\Big), \quad t \ge 0. \tag{13}$$

Proof. For the product part, the Young inequality $|xy|/(KL) \le x^2/(2K^2) + y^2/(2L^2)$ combined with the AM–GM and Cauchy–Schwarz inequalities yields

$$\begin{aligned}\mathbb{E}\exp(|XY|/(KL)) &\le \mathbb{E}\exp(X^2/(2K^2) + Y^2/(2L^2)) \\ &= \mathbb{E}\exp(X^2/(2K^2))\mathbb{E}\exp(Y^2/(2L^2)) \le 2,\end{aligned}$$

where we used independence in the equality and the ψ_2-defining bound twice in the last step (after noting that $\exp(X^2/(2K^2)) \le \sqrt{\exp(X^2/K^2)}$, so its expectation is at most $\sqrt{2}$). Hence $\|XY\|_{\psi_1} \le KL$. For the mean, $\mathbb{E}|XY| \le \sqrt{\mathbb{E}X^2\mathbb{E}Y^2} < \infty$ by Cauchy–Schwarz and Lemma B.1, so by Fubini $\mathbb{E}[XY] = \mathbb{E}X\mathbb{E}Y$.

For the Bernstein bound, an argument identical to that of Lemma B.2 (now using $(\mathbb{E}|W|^p)^{1/p} \le CL_0p$ instead of $CK\sqrt{p}$, which is the sub-exponential moment bound proved by integrating the tail $\Pr[|W| \ge t] \le 2e^{-t/L_0}$) yields, for $|\lambda| \le c/L_0$,

$$\mathbb{E}\exp\Big(\lambda\sum_{i=1}^{n} W_i\Big) \le \exp(Cn\lambda^2 L_0^2).$$

Optimising the resulting Chernoff bound at $\lambda = \min\big(t/(2CnL_0^2), c/(2L_0)\big)$ gives the two-regime bound Eq. (13). □

Lemma B.4 (Sign-flip bound from a ψ_2 residual)**.** *If $\|X\|_{\psi_2} \le K$ and $\mu > 0$, then* $\Pr[|X| \ge \mu] \le 2\exp(-\mu^2/K^2)$.

Proof. Markov's inequality applied to $\exp(X^2/K^2)$ at level $\exp(\mu^2/K^2)$ gives $\Pr[|X| \ge \mu] = \Pr[\exp(X^2/K^2) \ge \exp(\mu^2/K^2)] \le 2\exp(-\mu^2/K^2)$. □

C Conditional Reconstruction: Proof of Theorem 4.1

The following argument preserves the original finite-sample result under its stated probabilistic assumptions. Learned or SVD-derived signs have not been shown to satisfy joint Rademacher independence. The canonical feasible point is not necessarily attained by the implemented PTQ optimizer. Neither downstream accuracy nor a carrier-budget saturation point follows from the theorem. Envelope-count monotonicity concerns nested feasible sets and their best reconstruction error, not the solution reached by training.

This appendix gives a full, self-contained proof of Theorem 4.1 from Section 4. The argument is entirely a matter of entry-wise sub-Gaussian and sub-exponential concentration, and no sketching, Rademacher compression, or Johnson–Lindenstrauss reduction is used. All underlying probabilistic facts are proved from scratch in Appendix B (sub-Gaussian sum tail Lemma B.2, sub-exponential product and Bernstein inequality Lemma B.3, and the sign-flip bound Lemma B.4). In particular, the sub-Gaussian Orlicz norm and the second-moment bound $\mathbb{E}[X^2] \le \|X\|_{\psi_2}^2$ used freely below are recorded in Lemma B.1.

Assumption C.1 (Sign-plus-sub-Gaussian-noise factor model)**.** *The LoRA factors $A \in \mathbb{R}^{N\times r}$ and $B \in \mathbb{R}^{M\times r}$ are random matrices defined on a common probability space $(\Omega, \mathcal{A}, \Pr)$ and admit the entry-wise decomposition*

$$A_{ik} = \mu_A\sigma_{ik}^A + \xi_{ik}^A, \;\; B_{jk} = \mu_B\sigma_{jk}^B + \xi_{jk}^B, \;\; i \in [N],\; j \in [M],\; k \in [r], \tag{14}$$

where $\mu_A, \mu_B > 0$ and $\zeta > 0$ are deterministic scalars and the sign arrays $\sigma^A \in \{\pm1\}^{N\times r}$, $\sigma^B \in \{\pm1\}^{M\times r}$ and residual arrays $\xi^A \in \mathbb{R}^{N\times r}$, $\xi^B \in \mathbb{R}^{M\times r}$ satisfy:

1. *The families $\{\xi^A_{ik}\}_{(i,k)\in[N]\times[r]}$ and $\{\xi^B_{jk}\}_{(j,k)\in[M]\times[r]}$ consist of mutually independent random variables, the two families are mutually independent, and each entry has zero mean.*

2. *The residuals are uniformly sub-Gaussian:*
$$\max\Big(\max_{(i,k)\in[N]\times[r]}\|\xi^A_{ik}\|_{\psi_2},\ \max_{(j,k)\in[M]\times[r]}\|\xi^B_{jk}\|_{\psi_2}\Big)\le\zeta.$$

3. *The sign σ-algebra $\mathcal{F}\coloneqq\sigma(\sigma^A,\sigma^B)$ is independent of $\sigma(\xi^A,\xi^B)$.*

We allow the latent sign arrays to be either fixed or random. When σ^A,σ^B are fixed, all probability statements are taken only over the residual arrays (ξ^A,ξ^B); in this case Assumption C.1(iii) is vacuous. For the relative bound in Theorem 4.1, the entries of σ^A and σ^B are additionally assumed to be jointly i.i.d. Rademacher.

The canonical reconstruction used in Theorem 4.1 is
$$\theta^\star\coloneqq\big(\sigma^A,(\sigma^B)^\top,\mathbf{1}_N,\mu_A\mu_B\mathbf{1}_r,\mathbf{1}_M\big),\tag{15}$$
which is a single-envelope LoRDBA adapter with the latent signs as the binary carriers. For each envelope rank $\ell\ge1$, let Θ_ℓ denote the corresponding LoRDBA class of Definition 3.1. The zero-padding embedding
$$\iota_\ell:\Theta_{\ell-1}\hookrightarrow\Theta_\ell,\ \ \big(B_1,B_2,\{(\boldsymbol{\alpha}^{(i)},\boldsymbol{\beta}^{(i)},\boldsymbol{\gamma}^{(i)})\}_{i=1}^{\ell-1}\big)\mapsto\big(B_1,B_2,\{(\boldsymbol{\alpha}^{(i)},\boldsymbol{\beta}^{(i)},\boldsymbol{\gamma}^{(i)})\}_{i=1}^{\ell-1},(\mathbf{0},\mathbf{0},\mathbf{0})\big)\tag{16}$$
preserves ΔW, so the best achievable reconstruction error is non-increasing in the envelope rank.

Let $E\coloneqq AB^\top-\Delta W(\theta^\star)\in\mathbb{R}^{N\times M}$ denote the reconstruction residual. We first expand E entry-wise (Lemma C.2), then bound each entry by sub-exponential concentration (Lemma C.3), aggregate by a union bound to obtain (25), and finally combine the Frobenius upper bound with a Chebyshev lower bound on $\|\Delta W(\theta^\star)\|_F^2$ (Lemma C.4) to obtain (9).

For the absolute bound (25) the signs σ^A,σ^B are deterministic, so all probability statements below are taken with respect to the residuals (ξ^A,ξ^B). For the relative bound (9) the signs are additionally i.i.d. Rademacher and jointly independent of the residuals; every statement proved for deterministic signs therefore also holds *conditionally* on (σ^A,σ^B), and by Fubini the residual event of probability at least $1-\delta$ holds on the product space.

C.1 Entry-wise expansion

The heart of the argument is the following entry-wise expansion of the residual into three pieces with distinct concentration behaviour. Its pairwise L^2-orthogonality keeps the final variance computation additive.

Lemma C.2 (Residual expansion)**.** *Under Assumption C.1, for every $i\in[N]$ and $j\in[M]$,*
$$E_{ij}\ =\ \underbrace{\mu_A\sum_{k=1}^r\sigma^A_{ik}\xi^B_{jk}}_{=:T^{(1)}_{ij}}\ +\ \underbrace{\mu_B\sum_{k=1}^r\xi^A_{ik}\sigma^B_{jk}}_{=:T^{(2)}_{ij}}\ +\ \underbrace{\sum_{k=1}^r\xi^A_{ik}\xi^B_{jk}}_{=:T^{(3)}_{ij}}.\tag{17}$$
Let $\mathcal{F}\coloneqq\sigma(\sigma^A,\sigma^B)$ denote the σ-algebra generated by the signs (which reduces to the trivial σ-algebra when the signs are deterministic). Then the three summands are pairwise L^2-orthogonal conditionally on $\mathcal{F}$:
$$\mathbb{E}\big[T^{(a)}_{ij}T^{(b)}_{ij}\big|\mathcal{F}\big]\ =\ 0\quad\text{for every distinct } a,b\in\{1,2,3\}.\tag{18}$$
In particular, taking total expectation, the same orthogonality holds unconditionally.

Proof. Step 1: entry-wise identity. Substituting Eq. (14) into $(AB^\top)_{ij}=\sum_{k=1}^r A_{ik}B_{jk}$ and expanding the scalar product $(\mu_A\sigma^A_{ik}+\xi^A_{ik})(\mu_B\sigma^B_{jk}+\xi^B_{jk})$ termwise,
$$(AB^\top)_{ij}\ =\ \underbrace{\mu_A\mu_B\sum_{k=1}^r\sigma^A_{ik}\sigma^B_{jk}}_{=[\Delta W(\theta^\star)]_{ij}}+T^{(1)}_{ij}+T^{(2)}_{ij}+T^{(3)}_{ij},$$

where the identification of the leading term with $[\Delta W(\theta^\star)]_{ij}$ follows from Eq. (15) together with $[\mathrm{diag}(\mathbf{1}_N)\sigma^A \,\mathrm{diag}(\mu_A\mu_B\mathbf{1}_r)(\sigma^B)^\top \,\mathrm{diag}(\mathbf{1}_M)]_{ij} = \mu_A\mu_B \sum_k \sigma^A_{ik}\sigma^B_{jk}$. Subtracting this leading term yields Eq. (17).

Step 2: conditional L^2-orthogonality. By Assumption C.1(iii) the sign σ-algebra $\mathcal{F} = \sigma(\sigma^A, \sigma^B)$ is independent of $\sigma(\xi^A, \xi^B)$, which implies that the conditional law of (ξ^A, ξ^B) given $\mathcal{F}$ equals its unconditional law; all residual moments appearing below therefore equal their unconditional counterparts. Fix (i, j) and recall $T^{(1)}_{ij}T^{(2)}_{ij} = \mu_A\mu_B \sum_{k,l} \sigma^A_{ik}\sigma^B_{jl}\xi^B_{jk}\xi^A_{il}$. The sign factors are $\mathcal{F}$-measurable and can be pulled out of the conditional expectation; Assumption C.1(i) ensures that ξ^A_{il} and ξ^B_{jk} are independent (they belong to different independent families) with $\mathbb{E}[\xi^A_{il}] = \mathbb{E}[\xi^B_{jk}] = 0$, so

$$\mathbb{E}[T^{(1)}_{ij}T^{(2)}_{ij} \mid \mathcal{F}] = \mu_A\mu_B \sum_{k,l=1}^{r} \sigma^A_{ik}\sigma^B_{jl}\mathbb{E}[\xi^A_{il}]\mathbb{E}[\xi^B_{jk}] = 0.$$

For $T^{(1)}_{ij}T^{(3)}_{ij} = \mu_A \sum_{k,l} \sigma^A_{ik}\xi^B_{jk}\xi^A_{il}\xi^B_{jl}$, the factor ξ^A_{il} is independent of (ξ^B_{jk}, ξ^B_{jl}) (different families, cf. (i)), so its expectation factorises and vanishes:

$$\mathbb{E}[T^{(1)}_{ij}T^{(3)}_{ij} \mid \mathcal{F}] = \mu_A \sum_{k,l=1}^{r} \sigma^A_{ik}\mathbb{E}[\xi^A_{il}]\mathbb{E}[\xi^B_{jk}\xi^B_{jl}] = 0.$$

The identity $\mathbb{E}[T^{(2)}_{ij}T^{(3)}_{ij} \mid \mathcal{F}] = 0$ is symmetric, replacing the roles of A and B:

$$\mathbb{E}[T^{(2)}_{ij}T^{(3)}_{ij} \mid \mathcal{F}] = \mu_B \sum_{k,l=1}^{r} \sigma^B_{jk}\mathbb{E}[\xi^B_{jl}]\mathbb{E}[\xi^A_{ik}\xi^A_{il}] = 0.$$

This establishes Eq. (18). Taking expectations and using the tower property yields the unconditional orthogonality. □

C.2 Sub-exponential tail of each entry

With Lemma C.2 in hand, each summand in Eq. (17) is either a sum of independent sub-Gaussians or a sum of independent products of sub-Gaussians; the latter is sub-exponential. A union bound over the three summands therefore yields a sub-exponential tail for E_{ij}.

Lemma C.3 (Sub-exponential bound on E_{ij})**.** *Let $V := r\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)$. There exists a universal constant $c_1 > 0$ such that, for every $(i, j) \in [N] \times [M]$ and every $t > 0$,*

$$\Pr\big[|E_{ij}| > t\big] \;\le\; 6\exp\Big(-c_1 \min\Big(\frac{t^2}{V}, \frac{t}{\zeta^2}\Big)\Big). \tag{19}$$

Moreover $\mathbb{E}[E^2_{ij}] \le V$, hence $\mathbb{E}\|E\|_F^2 \le NMV$.

Proof. Fix $(i, j) \in [N] \times [M]$ and condition on the σ-algebra $\mathcal{F} = \sigma(\sigma^A, \sigma^B)$. By Assumption C.1(iii) the residuals (ξ^A, ξ^B) are independent of $\mathcal{F}$, so the conditional joint law of the residuals given $\mathcal{F}$ coincides with their unconditional law. All probability statements in parts (a)–(c) below are understood conditionally on $\mathcal{F}$; the bounds are uniform over realisations of $\mathcal{F}$ and therefore also hold unconditionally. Throughout we write $c_a, c_b > 0$ for the universal constants produced by Lemma B.2 (sub-Gaussian sum tail) and Lemma B.3 (Bernstein) of Appendix B.

(a) $T^{(1)}_{ij}$ is sub-Gaussian. For each $k \in [r]$, $|\sigma^A_{ik}| = 1$ and the positive-homogeneity of $\|\cdot\|_{\psi_2}$ give $\|\sigma^A_{ik}\xi^B_{jk}\|_{\psi_2} = \|\xi^B_{jk}\|_{\psi_2} \le \zeta$. By Assumption C.1(i) the family $\{\xi^B_{jk}\}_{k=1}^r$ is mutually independent and the $\mathcal{F}$-measurable signs σ^A_{ik} preserve (conditional) independence and the zero mean. Applying Lemma B.2 conditionally on $\mathcal{F}$ to $\{X_k = \sigma^A_{ik}\xi^B_{jk}\}_{k=1}^r$ with $a_k \equiv 1$, $K = \zeta$, scaled by the deterministic constant μ_A, yields

$$\Pr\big[|T^{(1)}_{ij}| > t\big|\mathcal{F}\big] \;\le\; 2\exp\big(-c_a t^2/(\mu_A^2 r\zeta^2)\big) \qquad \forall t \ge 0. \tag{20}$$

(b) $T^{(2)}_{ij}$ is sub-Gaussian. Exchanging the roles of the two families in (a),

$$\Pr\big[|T^{(2)}_{ij}| > t\big|\mathcal{F}\big] \;\le\; 2\exp\big(-c_a t^2/(\mu_B^2 r\zeta^2)\big) \qquad \forall t \ge 0. \tag{21}$$

(c) $T_{ij}^{(3)}$ is sub-exponential. By Assumption C.1(i), ξ_{ik}^A and ξ_{jk}^B are independent with zero mean, so Lemma B.3 gives $\mathbb{E}[\xi_{ik}^A \xi_{jk}^B \mid \mathcal{F}] = \mathbb{E}[\xi_{ik}^A]\mathbb{E}[\xi_{jk}^B] = 0$ and $\|\xi_{ik}^A \xi_{jk}^B\|_{\psi_1} \le \|\xi_{ik}^A\|_{\psi_2} \|\xi_{jk}^B\|_{\psi_2} \le \zeta^2$. The summands $\{\xi_{ik}^A \xi_{jk}^B\}_{k=1}^r$ are independent across k: indeed, $\sigma(\xi_{ik}^A \xi_{jk}^B) \subseteq \sigma(\xi_{ik}^A, \xi_{jk}^B)$, and the pairs (ξ_{ik}^A, ξ_{jk}^B) for distinct $k \in [r]$ involve disjoint index sets in the mutually independent families $\{\xi_{\cdot\cdot}^A\}, \{\xi_{\cdot\cdot}^B\}$. The Bernstein bound Eq. (13) of Lemma B.3 therefore yields, with $c_b > 0$ the universal constant of that lemma,

$$\Pr\big[|T_{ij}^{(3)}| > t \big| \mathcal{F}\big] \;\le\; 2\exp\Big(-c_b \min\big(t^2/(r\zeta^4), t/\zeta^2\big)\Big) \qquad \forall t \ge 0. \tag{22}$$

Combination. Triangle inequality gives $|E_{ij}| \le |T_{ij}^{(1)}| + |T_{ij}^{(2)}| + |T_{ij}^{(3)}|$, so

$$\{|E_{ij}| > t\} \;\subseteq\; \{|T_{ij}^{(1)}| > t/3\} \cup \{|T_{ij}^{(2)}| > t/3\} \cup \{|T_{ij}^{(3)}| > t/3\}.$$

Since $V = r\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)$ dominates each of $\mu_A^2 r\zeta^2$, $\mu_B^2 r\zeta^2$, and $r\zeta^4$, Eqs. (20)–(21) at $t/3$ are bounded by $2\exp(-c_a t^2/(9V))$, and Eq. (22) at $t/3$ is bounded by $2\exp(-c_b \min(t^2/(9V), t/(3\zeta^2)))$. Defining

$$c_1 \;\coloneqq\; \frac{1}{9}\min(c_a, c_b), \tag{23}$$

each of the three bounds is dominated by $2\exp(-c_1 \min(t^2/V, t/\zeta^2))$: indeed $c_a/9 \ge c_1$, $c_b/9 \ge c_1$, and $c_b/3 \ge c_1$. Taking the union over the three events yields Eq. (19) conditionally on $\mathcal{F}$; uniformity over $\mathcal{F}$ gives the unconditional bound.

Second moment. Lemma B.1 gives $\mathbb{E}[(\xi_{ik}^A)^2] \le \zeta^2$ and $\mathbb{E}[(\xi_{jk}^B)^2] \le \zeta^2$. Using the mutual independence of Assumption C.1(i), zero means, and $(\sigma_{\cdot\cdot}^{\cdot})^2 = 1$,

$$\begin{aligned}
\mathbb{E}[(T_{ij}^{(1)})^2 \mid \mathcal{F}] &= \mu_A^2 \sum_{k,l=1}^r \sigma_{ik}^A \sigma_{il}^A \mathbb{E}[\xi_{jk}^B \xi_{jl}^B] = \mu_A^2 \sum_{k=1}^r \mathbb{E}[(\xi_{jk}^B)^2] \;\le\; \mu_A^2 r\zeta^2, \\
\mathbb{E}[(T_{ij}^{(2)})^2 \mid \mathcal{F}] &\le\; \mu_B^2 r\zeta^2, \\
\mathbb{E}[(T_{ij}^{(3)})^2 \mid \mathcal{F}] &= \sum_{k,l=1}^r \mathbb{E}[\xi_{ik}^A \xi_{il}^A]\mathbb{E}[\xi_{jk}^B \xi_{jl}^B] = \sum_{k=1}^r \mathbb{E}[(\xi_{ik}^A)^2]\mathbb{E}[(\xi_{jk}^B)^2] \;\le\; r\zeta^4,
\end{aligned}$$

where the off-diagonal ($k \ne l$) cross terms vanish because the residuals within each family are mean zero and mutually independent. Combined with the conditional orthogonality of Lemma C.2, $\mathbb{E}[E_{ij}^2 \mid \mathcal{F}] = \sum_{a=1}^3 \mathbb{E}[(T_{ij}^{(a)})^2 \mid \mathcal{F}] \le V$. Taking total expectation and summing over $(i,j) \in [N] \times [M]$ yields $\mathbb{E}\|E\|_F^2 \le NMV$. □

C.3 Frobenius bound by union

The per-entry bound of Lemma C.3 transforms into a Frobenius bound by a union argument over the NM entries of E.

Proof of Theorem 4.1, Eq. (25). Fix $N, M \ge 2$, $r \ge 1$, and $\delta \in (0,1)$ with $u \coloneqq \log(6NM/\delta) \le c_1 r$, where c_1 is the constant defined in Eq. (23). Since $NM \ge 4$ and $\delta < 1$, $u \ge \log 24 > 0$. Set $t_\star \coloneqq \sqrt{Vu/c_1} > 0$.

Step 1: the sub-Gaussian branch binds at $t = t_\star$. In Eq. (19) the inner minimum satisfies $\min(c_1 t^2/V, c_1 t/\zeta^2) = c_1 t^2/V$ iff $t^2/V \le t/\zeta^2$, iff $t \le V/\zeta^2$. At $t = t_\star$ this becomes $\sqrt{Vu/c_1} \le V/\zeta^2$, equivalently $u \le c_1 V/\zeta^4$. From $V = r\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)$ and $\zeta > 0$,

$$\frac{V}{\zeta^4} \;=\; r\Big(\frac{\mu_A^2}{\zeta^2} + \frac{\mu_B^2}{\zeta^2} + 1\Big) \;\ge\; r,$$

so the standing hypothesis $u \le c_1 r$ entails $u \le c_1 V/\zeta^4$, and the sub-Gaussian branch binds:

$$c_1 \min(t_\star^2/V, t_\star/\zeta^2) \;=\; c_1 t_\star^2/V \;=\; u.$$

Step 2: per-entry bound. Substituting $t = t_\star$ into Eq. (19) and using Step 1,

$$\Pr\big[E_{ij}^2 > Vu/c_1\big] \;=\; \Pr\big[|E_{ij}| > t_\star\big] \;\le\; 6e^{-u} \;=\; \frac{\delta}{NM}, \qquad (i,j) \in [N] \times [M].$$

Step 3: union bound. If $\max_{(i,j)} E_{ij}^2 \le Vu/c_1$, then $\|E\|_F^2 = \sum_{i,j} E_{ij}^2 \le NM \cdot Vu/c_1$. Contrapositively, $\{\|E\|_F^2 > NMVu/c_1\} \subseteq \bigcup_{i,j}\{E_{ij}^2 > Vu/c_1\}$. A union bound over the NM entries gives $\Pr[\|E\|_F^2 > NMVu/c_1] \le NM \cdot \delta/(NM) = \delta$. Hence, with probability at least $1-\delta$,

$$\|E\|_F^2 \;\le\; \frac{NMr\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)}{c_1} \log(6NM/\delta). \tag{24}$$

Step 4: conversion $\log(6NM/\delta) \to \log(2NM/\delta)$. For $N, M \ge 2$ and $\delta \in (0,1)$, $2NM/\delta \ge 8$, so $\log(2NM/\delta) \ge \log 8 > \log 3$. Therefore

$$\log(6NM/\delta) \;=\; \log 3 + \log(2NM/\delta) \;\le\; \log(2NM/\delta) + \log(2NM/\delta) \;=\; 2\log(2NM/\delta).$$

Substituting into Eq. (24) and setting $C \coloneqq 2/c_1 > 0$ yields the absolute bound

$$\big\|AB^\top - \Delta W(\theta^\star)\big\|_F^2 \;\le\; C \cdot NMr\zeta^2\big(\mu_A^2 + \mu_B^2 + \zeta^2\big)\log(2NM/\delta). \tag{25}$$

The constant C depends only on the universal constants c_a, c_b produced by Lemma B.2 and Lemma B.3 of Appendix B through Eq. (23), and is otherwise independent of all model parameters. □

C.4 Chebyshev lower bound on the signal

The relative bound requires a lower bound on $\|\Delta W(\theta^\star)\|_F^2$. This is supplied by a second-moment argument on the Rademacher bilinear form of the signs.

Lemma C.4 (Lower bound on $\|\Delta W(\theta^\star)\|_F^2$)**.** *Assume $\{\sigma_{ik}^A\} \cup \{\sigma_{jk}^B\}$ are i.i.d. Rademacher with $\sigma^A \perp \sigma^B$. Let $W_{ij} \coloneqq \sum_{k=1}^r \sigma_{ik}^A \sigma_{jk}^B$. Then for every $N, M \ge 1$, $r \ge 1$,*

$$\Pr\Big[\textstyle\sum_{i,j} W_{ij}^2 \ge \frac{1}{2}NMr\Big] \;\ge\; 1 - \frac{8}{NM}. \tag{26}$$

In particular, on this event $\|\Delta W(\theta^\star)\|_F^2 = \mu_A^2\mu_B^2 \sum_{ij} W_{ij}^2 \ge \frac{1}{2}\mu_A^2\mu_B^2 NMr$.

Proof. Define $Z \coloneqq \sum_{i,j} W_{ij}^2$ and, for $k, l \in [r]$,

$$G_{kl}^A \;\coloneqq\; \sum_{i=1}^N \sigma_{ik}^A \sigma_{il}^A, \qquad G_{kl}^B \;\coloneqq\; \sum_{j=1}^M \sigma_{jk}^B \sigma_{jl}^B,$$

so that $G_{kl}^A = G_{lk}^A$ and $G_{kl}^B = G_{lk}^B$. Expanding $W_{ij}^2 = \sum_{k,l} \sigma_{ik}^A \sigma_{il}^A \sigma_{jk}^B \sigma_{jl}^B$ and interchanging the finite sums over (i,j) and (k,l),

$$Z \;=\; \sum_{k,l=1}^r G_{kl}^A G_{kl}^B \;=\; \underbrace{\sum_{k=1}^r G_{kk}^A G_{kk}^B}_{=rNM} \;+\; \sum_{k \ne l} G_{kl}^A G_{kl}^B, \tag{27}$$

where the first sum is deterministic because $G_{kk}^A = \sum_i (\sigma_{ik}^A)^2 = N$ and $G_{kk}^B = M$. If $r = 1$ the off-diagonal sum is empty, $Z = NM = rNM \ge rNM/2$ deterministically, and Eq. (26) is trivial. We henceforth assume $r \ge 2$.

Mean. For $k \ne l$ and fixed $i \in [N]$, σ_{ik}^A and σ_{il}^A are independent Rademachers, hence $\mathbb{E}[\sigma_{ik}^A \sigma_{il}^A] = \mathbb{E}[\sigma_{ik}^A]\mathbb{E}[\sigma_{il}^A] = 0$, so $\mathbb{E}[G_{kl}^A] = 0$ and similarly $\mathbb{E}[G_{kl}^B] = 0$. By $\sigma^A \perp \sigma^B$, $\mathbb{E}[G_{kl}^A G_{kl}^B] = \mathbb{E}[G_{kl}^A]\mathbb{E}[G_{kl}^B] = 0$. Hence $\mathbb{E}[Z] = rNM$.

Variance. By symmetry, $\sum_{k \ne l} G_{kl}^A G_{kl}^B = 2S$ with $S \coloneqq \sum_{1 \le k < l \le r} G_{kl}^A G_{kl}^B$. We claim

$$\mathbb{E}\big[G_{kl}^A G_{k'l'}^A\big] \;=\; N \cdot \mathbf{1}\{(k,l) = (k',l')\} \qquad \text{for } k < l,\; k' < l'. \tag{28}$$

Indeed, $G_{kl}^A G_{k'l'}^A = \sum_{i,i'=1}^N \sigma_{ik}^A \sigma_{il}^A \sigma_{i'k'}^A \sigma_{i'l'}^A$. By the joint i.i.d. Rademacher hypothesis, all $\{\sigma_{pq}^A\}_{(p,q) \in [N] \times [r]}$ are mutually independent, so the expectation of any finite product factorises across distinct (p,q) pairs. We split by the coincidence pattern of i and i':

- $i \neq i'$. The four indices $(i,k),(i,l),(i',k'),(i',l')$ are pairwise distinct (as $k \neq l$ and $k' \neq l'$ and $i \neq i'$), so the expectation factorises into four separate factors $\mathbb{E}[\sigma^A_{ik}] \cdots \mathbb{E}[\sigma^A_{i'l'}] = 0$.
- $i = i'$. The product becomes $\sigma^A_{ik}\sigma^A_{il}\sigma^A_{ik'}\sigma^A_{il'}$, whose expectation equals 1 iff each column index $q \in \{k,l,k',l'\}$ appears with even multiplicity in the multiset and equals 0 otherwise. Under $k < l$ and $k' < l'$, even multiplicity forces $\{k,l\} = \{k',l'\}$ as multisets, hence $(k,l) = (k',l')$. In the diagonal case each index appears exactly twice, $(\sigma^A_{i\cdot})^2 = 1$, and the expectation is 1; summing over $i \in [N]$ contributes N.

This proves Eq. (28); the identical argument with B in place of A gives $\mathbb{E}[G^B_{kl}G^B_{k'l'}] = M \cdot \mathbf{1}\{(k,l) = (k',l')\}$ for $k < l$, $k' < l'$. By $\sigma^A \perp \sigma^B$,

$$\mathbb{E}\big[G^A_{kl}G^B_{kl}G^A_{k'l'}G^B_{k'l'}\big] = \mathbb{E}\big[G^A_{kl}G^A_{k'l'}\big]\mathbb{E}\big[G^B_{kl}G^B_{k'l'}\big] = NM \cdot \mathbf{1}\{(k,l) = (k',l')\}.$$

Summing over the $\binom{r}{2}$ pairs with $k < l$,

$$\mathbb{E}[S^2] \;=\; \sum_{\substack{k<l,\\ k'<l'}} \mathbb{E}\big[G^A_{kl}G^B_{kl}G^A_{k'l'}G^B_{k'l'}\big] \;=\; \binom{r}{2}NM.$$

Combined with Eq. (27) and $Z - rNM = 2S$,

$$\mathrm{Var}(Z) \;=\; \mathbb{E}[(Z - \mathbb{E}[Z])^2] \;=\; 4\mathbb{E}[S^2] \;=\; 2r(r-1)NM \;\le\; 2r^2NM.$$

Chebyshev. Since $Z \geq 0$, $\{Z < rNM/2\} = \{rNM - Z > rNM/2\} \subseteq \{|Z - rNM| > rNM/2\}$. Chebyshev's inequality therefore gives

$$\Pr\big[Z < rNM/2\big] \;\le\; \Pr\big[|Z - rNM| > rNM/2\big] \;\le\; \frac{\mathrm{Var}(Z)}{(rNM/2)^2} \;\le\; \frac{2r^2NM}{r^2N^2M^2/4} \;=\; \frac{8}{NM},$$

which is the complementary form of Eq. (26). □

Proof of Theorem 4.1, Eq. (9)*. Transfer of the absolute bound to the joint space.* In the relative statement of Theorem 4.1 we assume $NM > 8$ and $\delta \in (0, 1 - 8/(NM))$, so the final probability lower bound is positive. Under the Rademacher hypothesis on the signs, the arguments in Lemmas C.2–C.3 and in the preceding proof were performed conditionally on $\mathcal{F} = \sigma(\sigma^A, \sigma^B)$. In particular, for every realisation (s^A, s^B) of (σ^A, σ^B) the conditional probability

$$\Pr\big[\mathcal{E}_1 \big| \sigma^A = s^A, \sigma^B = s^B\big] \;\ge\; 1 - \delta,$$
$$\mathcal{E}_1 \;:=\; \big\{\|E\|_F^2 \le CNMr\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)\log(2NM/\delta)\big\}.$$

Taking expectation over the signs (tower property) gives $\Pr[\mathcal{E}_1] \geq 1 - \delta$ on the joint space. Let $\mathcal{E}_2 := \{\sum_{i,j} W_{ij}^2 \geq NMr/2\}$ be the signal event of Lemma C.4; it depends only on the signs and has $\Pr[\mathcal{E}_2] \geq 1 - 8/(NM)$. The union bound $\Pr[\mathcal{E}_1^c \cup \mathcal{E}_2^c] \leq \delta + 8/(NM)$ yields

$$\Pr[\mathcal{E}_1 \cap \mathcal{E}_2] \;\ge\; 1 - \delta - \frac{8}{NM}.$$

Ratio on $\mathcal{E}_1 \cap \mathcal{E}_2$. By definition of $\mathcal{E}_2$,

$$\|\Delta W(\theta^\star)\|_F^2 = \mu_A^2\mu_B^2 \sum_{i,j} W_{ij}^2 \;\ge\; \tfrac{1}{2}\mu_A^2\mu_B^2 NMr.$$

Combining with the formal upper bound on $\|E\|_F^2$ provided by $\mathcal{E}_1$,

$$\begin{aligned}\frac{\|E\|_F^2}{\|\Delta W(\theta^\star)\|_F^2} &\le \frac{CNMr\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)\log(2NM/\delta)}{\frac{1}{2}\mu_A^2\mu_B^2 NMr} \\ &= 2C \cdot \frac{\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)}{\mu_A^2\mu_B^2}\log(2NM/\delta).\end{aligned} \tag{29}$$

Let $\mu_{\max} := \max(\mu_A, \mu_B)$ and $\mu_{\min} := \min(\mu_A, \mu_B)$; the hypothesis $\zeta \leq \mu_{\max}$ yields $\mu_A^2 + \mu_B^2 + \zeta^2 \leq 3\mu_{\max}^2$. Using $\mu_A\mu_B = \mu_{\max}\mu_{\min}$,

$$\frac{\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)}{\mu_A^2\mu_B^2} \;\le\; \frac{3\zeta^2\mu_{\max}^2}{\mu_{\max}^2\mu_{\min}^2} \;=\; \frac{3\zeta^2}{\mu_{\min}^2}.$$

Substituting into Eq. (29) yields

$$\frac{\|E\|_F^2}{\|\Delta W(\theta^\star)\|_F^2} \;\le\; \frac{6C\zeta^2}{\mu_{\min}^2}\log(2NM/\delta).$$

Taking square roots (both sides are non-negative) and setting $C' \coloneqq \sqrt{6C} > 0$ proves Eq. (9) on the event $\mathcal{E} \coloneqq \mathcal{E}_1 \cap \mathcal{E}_2$, whose probability is at least $1 - \delta - 8/(NM)$.

Small-noise transfer. Put $\tau \coloneqq C'\zeta\sqrt{\log(2NM/\delta)}/\mu_{\min}$ and assume $\tau \le 1/2$. On $\mathcal{E}$, Eq. (9) gives $\|E\|_F \le \tau\|\Delta W(\theta^\star)\|_F$, and the reverse triangle inequality $\big|\|AB^\top\|_F - \|\Delta W(\theta^\star)\|_F\big| \le \|AB^\top - \Delta W(\theta^\star)\|_F = \|E\|_F$ implies

$$\|AB^\top\|_F \;\ge\; \|\Delta W(\theta^\star)\|_F - \|E\|_F \;\ge\; (1-\tau)\|\Delta W(\theta^\star)\|_F \;\ge\; \tfrac{1}{2}\|\Delta W(\theta^\star)\|_F.$$

Dividing the bound Eq. (9) by a denominator at least $\frac{1}{2}\|\Delta W(\theta^\star)\|_F$ costs at most a factor of 2, which establishes the transferred bound $\|E\|_F/\|AB^\top\|_F \le 2\tau$ on $\mathcal{E}$. □

The theorem gives a sufficient reconstruction condition through the distributional ratio $\zeta/\min(\mu_A, \mu_B)$ under its stated sign and residual assumptions. The sub-Gaussian tail parameter ζ is not directly observable from a single trained adapter and is not identified by an empirical standard deviation. The descriptive statistics below therefore neither validate those assumptions nor calibrate a numerical theorem guarantee. A vacuous upper bound does not establish that quantization must fail.

C.5 Sign consistency and observed-sign reconstruction

The canonical reconstruction $\theta^\star$ uses the latent sign arrays σ^A, σ^B, which are not in general equal to the observed entrywise signs $\operatorname{sign}(A), \operatorname{sign}(B)$. The next lemma shows that the two coincide with high probability whenever the residual is small relative to the latent mean, so the observed-sign LoRDBA adapter inherits the same Theorem 4.1 guarantees up to an explicit additional failure probability.

Lemma C.5 (Sign consistency of the canonical reconstruction)**.** *Under Assumption C.1, the latent sign arrays σ^A, σ^B and the observed entrywise sign patterns $\operatorname{sign}(A), \operatorname{sign}(B)$ agree simultaneously with probability at least*

$$1 \;-\; 2Nr\exp\!\big(-\mu_A^2/\zeta^2\big) \;-\; 2Mr\exp\!\big(-\mu_B^2/\zeta^2\big). \tag{30}$$

Equivalently, on the event of probability (30)*,*

$$\operatorname{sign}(A) \;=\; \sigma^A, \qquad \operatorname{sign}(B) \;=\; \sigma^B. \tag{31}$$

Consequently, on the same event, the observed-sign choice $B_1 = \operatorname{sign}(A)$, $B_2 = \operatorname{sign}(B^\top)$ equals the latent-sign binary factors of the canonical reconstruction in Eq. (15)*.*

Proof. Fix an entry $(i,k) \in [N] \times [r]$ of A. By Assumption C.1(i)–(ii),

$$A_{ik} \;=\; \mu_A\sigma_{ik}^A + \xi_{ik}^A, \qquad \sigma_{ik}^A \in \{\pm 1\}, \qquad \mu_A > 0, \qquad \|\xi_{ik}^A\|_{\psi_2} \le \zeta.$$

We claim the implication

$$\operatorname{sign}(A_{ik}) \neq \sigma_{ik}^A \implies |\xi_{ik}^A| \ge \mu_A. \tag{32}$$

Indeed, multiplying A_{ik} by $\sigma_{ik}^A \in \{\pm 1\}$ gives

$$\sigma_{ik}^A A_{ik} \;=\; \mu_A + \sigma_{ik}^A\xi_{ik}^A.$$

Under the convention $\operatorname{sign}(0) = +1$, a mismatch implies $\sigma_{ik}^A A_{ik} \le 0$: if $\sigma_{ik}^A = +1$ then the mismatch forces $A_{ik} < 0$, while if $\sigma_{ik}^A = -1$ it forces $A_{ik} \ge 0$. Therefore

$$\sigma_{ik}^A A_{ik} = \mu_A + \sigma_{ik}^A\xi_{ik}^A \le 0,$$

which implies $\sigma_{ik}^A\xi_{ik}^A \le -\mu_A$ and hence $|\xi_{ik}^A| \ge \mu_A$. This proves Eq. (32).

By Eq. (32) and Lemma B.4 of Appendix B applied to ξ_{ik}^A with $K = \zeta$ and $\mu = \mu_A$,

$$\Pr\big[\operatorname{sign}(A_{ik}) \neq \sigma_{ik}^A\big] \;\le\; \Pr\big[|\xi_{ik}^A| \ge \mu_A\big] \;\le\; 2\exp\!\big(-\mu_A^2/\zeta^2\big). \tag{33}$$

The identical argument with A replaced by B and μ_A by μ_B gives, for every $(j,k) \in [M] \times [r]$,

$$\Pr\big[\operatorname{sign}(B_{jk}) \neq \sigma^B_{jk}\big] \;\le\; 2\exp\big(-\mu_B^2/\zeta^2\big). \tag{34}$$

A union bound over the Nr entries of A and the Mr entries of B, using Eqs. (33)–(34), yields

$$\Pr\big[\operatorname{sign}(A) \neq \sigma^A \text{ or } \operatorname{sign}(B) \neq \sigma^B\big] \;\le\; 2Nr\exp\big(-\mu_A^2/\zeta^2\big) + 2Mr\exp\big(-\mu_B^2/\zeta^2\big). \tag{35}$$

The complement is exactly Eq. (31) and has probability at least the value displayed in Eq. (30).

On the event Eq. (31), the observed-sign binary factors satisfy

$$B_1 \;=\; \operatorname{sign}(A) \;=\; \sigma^A, \qquad B_2 \;=\; \operatorname{sign}(B^\top) \;=\; \operatorname{sign}(B)^\top \;=\; (\sigma^B)^\top,$$

because sign is applied entrywise and is therefore preserved under transposition. These coincide with the binary factors of the canonical reconstruction in Eq. (15), completing the proof. □

Corollary C.6 (LoRDBA envelope-rank monotonicity)**.** *Under the hypotheses of Theorem 4.1, for every $\ell \ge 1$, $\inf_{\theta\in\Theta_\ell} \|AB^\top - \Delta W(\theta)\|_F$ is bounded by the absolute bound in Eq.* (25) *and, in the small-noise regime $\tau \le 1/2$, by a factor of two times the right-hand side of Eq.* (9) *relative to $\|AB^\top\|_F$.*

Proof. Apply ι_ℓ iteratively to $\theta^\star \in \Theta_1$ to obtain a feasible point with the same ΔW in Θ_ℓ, then invoke Theorem 4.1; the small-noise transfer follows from $\|AB^\top\|_F \ge \frac{1}{2}\|\Delta W(\theta^\star)\|_F$ under the standing condition $\tau \le 1/2$. □

Corollary C.7 (Observed-sign reconstruction)**.** *Assume the hypotheses of Theorem 4.1 and define the observed-sign LoRDBA adapter*

$$\widehat{\theta} \;\coloneqq\; \big(\operatorname{sign}(A), \operatorname{sign}(B)^\top, \mathbf{1}_N, \mu_A\mu_B\mathbf{1}_r, \mathbf{1}_M\big). \tag{36}$$

On the sign-consistency event of Lemma C.5,

$$\Delta W(\widehat{\theta}) \;=\; \Delta W(\theta^\star). \tag{37}$$

Consequently, every bound in Theorem 4.1 and Corollary C.6 applies to $\widehat{\theta}$, with the failure probability of the corresponding event increased by at most

$$p_{\text{flip}} \;\coloneqq\; 2Nr\exp\big(-\mu_A^2/\zeta^2\big) + 2Mr\exp\big(-\mu_B^2/\zeta^2\big). \tag{38}$$

Proof. By Lemma C.5, on the event Eq. (31) the binary factors of $\widehat{\theta}$ and $\theta^\star$ coincide,

$$\operatorname{sign}(A) \;=\; \sigma^A, \qquad \operatorname{sign}(B)^\top \;=\; (\sigma^B)^\top.$$

The scale vectors of $\widehat{\theta}$ are the same as those of $\theta^\star$, namely $\mathbf{1}_N, \mu_A\mu_B\mathbf{1}_r, \mathbf{1}_M$ in Eq. (15). Substituting both equalities into Eq. (4) yields $\Delta W(\widehat{\theta}) = \Delta W(\theta^\star)$ on this event, proving Eq. (37).

For each event $\mathcal{E}'$ produced by Theorem 4.1 or Corollary C.6, the corresponding inequality bounds the residual $\|AB^\top - \Delta W(\theta^\star)\|_F$. On the intersection $\mathcal{E}' \cap \{\operatorname{sign}(A) = \sigma^A, \operatorname{sign}(B) = \sigma^B\}$, this residual equals $\|AB^\top - \Delta W(\widehat{\theta})\|_F$ by Eq. (37), so the inequality also bounds the observed-sign residual. The complement bound follows from

$$\Pr\big[(\mathcal{E}')^c \cup \{\operatorname{sign}(A) \neq \sigma^A \text{ or } \operatorname{sign}(B) \neq \sigma^B\}\big] \;\le\; \Pr[(\mathcal{E}')^c] \;+\; p_{\text{flip}},$$

which adds at most p_{flip} of Eq. (38) to the original failure probability of $\mathcal{E}'$. □

Corollary C.8 (Operator-norm consequence)**.** *Under the hypotheses used for the absolute bound in Theorem 4.1, $E \coloneqq AB^\top - \Delta W(\theta^\star)$ satisfies, with probability at least $1-\delta$ over the residuals,*

$$\|E\|_{\text{op}} \;\le\; \|E\|_F \;\le\; \sqrt{CNMr\zeta^2(\mu_A^2 + \mu_B^2 + \zeta^2)\log(2NM/\delta)}, \tag{39}$$

where $C > 0$ is the universal constant of Eq. (25)*.*

Proof. For every $X \in \mathbb{R}^{N\times M}$, the singular value decomposition gives

$$\|X\|_{\text{op}}^2 \;=\; \sigma_1(X)^2 \;\le\; \sum_{k=1}^{\min(N,M)} \sigma_k(X)^2 \;=\; \|X\|_F^2,$$

where $\sigma_1(X) \ge \sigma_2(X) \ge \cdots \ge 0$ are the singular values of X. Hence $\|E\|_{\text{op}} \le \|E\|_F$ deterministically. The second inequality of Eq. (39) is exactly the square root of Eq. (25), which holds with probability at least $1-\delta$. □

C.6 GAUGE FIXING AND DESCRIPTIVE MAGNITUDE DISPERSION

LoRA factor gauge. The product $AB^\top$ is invariant under the per-column positive rescaling

$$A_{:k} \mapsto d_k A_{:k}, \qquad B_{:k} \mapsto d_k^{-1} B_{:k}, \qquad d_k > 0,\ k \in [r_0], \tag{40}$$

since $(d_k A_{:k})(d_k^{-1} B_{:k})^\top = A_{:k} B_{:k}^\top$. The descriptive magnitude statistics depend on the particular gauge. We therefore compute every diagnostic in the following *column-balancing gauge*: for every nonzero column pair $(A_{:k}, B_{:k})$, set

$$d_k \;=\; \big(\|B_{:k}\|_2 / \|A_{:k}\|_2\big)^{1/2}, \qquad A_{:k} \leftarrow d_k A_{:k}, \qquad B_{:k} \leftarrow d_k^{-1} B_{:k}; \tag{41}$$

zero-contribution column pairs (if any) are left unchanged and excluded from the diagnostic. By construction this transformation preserves $AB^\top$ and the entrywise sign patterns of both columns (since $d_k > 0$), and equalises the column norms $\|A_{:k}\|_2 = \|B_{:k}\|_2 = \sqrt{\|A_{:k}^{\mathrm{old}}\|_2 \|B_{:k}^{\mathrm{old}}\|_2}$ for every nonzero pair. Thus the LoRA update and the canonical LoRDBA sign factors are unchanged, while the plug-in magnitude diagnostics are computed in a fixed, reproducible gauge. Eq. (41) specifies the historical diagnostic gauge, not a verification of the theorem's probabilistic assumptions.

Descriptive proxies. For trained fp16 LoRA factors A, B in the gauge of Eq. (41), define the following sample magnitude statistics. They are not estimators of the uniform sub-Gaussian ψ_2 bound in Assumption C.1:

$$\widehat{\mu}_A \coloneqq \tfrac{1}{Nr_0} \sum_{i,k} |A_{ik}|,\ \ \widehat{\mu}_B \coloneqq \tfrac{1}{Mr_0} \sum_{j,k} |B_{jk}|,\ \ \widehat{\zeta}_A \coloneqq \big(\tfrac{1}{Nr_0} \sum_{i,k} (|A_{ik}| - \widehat{\mu}_A)^2\big)^{1/2},$$

$\widehat{\zeta}_B$ is analogous, and $\widehat{\zeta} \coloneqq \max(\widehat{\zeta}_A, \widehat{\zeta}_B)$. We retain the historical notation for the descriptive ratio $\widehat{\zeta}/\min(\widehat{\mu}_A, \widehat{\mu}_B)$ in Table 6. A sample standard deviation does not verify residual independence, independence from signs, a mean-zero residual model, or a uniform sub-Gaussian bound. Consequently these values neither establish the theorem's small-noise and rank conditions nor give a numerical prediction of reconstruction error. For example, with $N = M = 4096$ and $\delta = 0.05$, the logarithmic factor alone makes $0.19\sqrt{\log(2NM/\delta)} \simeq 0.857$, before the unknown constant; it is not a prediction of 20% error. The relative theorem uses the canonical reconstruction in its denominator, whereas Table 16 uses the original update.

Table 6: Historical descriptive magnitude-dispersion ratios of fp16 LoRA factors; these are not estimated theorem parameters (adapter rank $r_0 = 64$, averaged over all attention and MLP projections), computed in the column-balancing gauge of Eq. (41).

Model	$\widehat{\mu}_A$	$\widehat{\mu}_B$	$\widehat{\zeta}$	ratio
LLaMA-3.2-3B	0.019	0.021	0.0044	0.23
LLaMA-2-7B	0.016	0.018	0.0030	0.19

D PTQ INITIALIZATION AND CONSENSUS-ADMM ANALYSIS

This appendix records post-training fitting of an existing fp16 LoRA update to the representation in Definition 3.1. It uses no task-loss recovery pass. The fitted signs and scales can be exported directly, or used to initialize the default QAT Freeze procedure in Section 3.2. QAT Full additionally updates latent sign variables. All three routes produce the same representation family; the reconstruction theorem does not determine which optimization route gives the best task accuracy.

Implementation used by the current DBF artifacts. The current single-envelope path starts the continuous factorization variables from seeded random values, then performs 100 DBF fitting iterations against the task-trained LoRA factor product. These are task-derived optimized carriers, not random carriers that remain unchanged throughout fitting. Within the factorization, sign projections are combined with rank-one magnitude-envelope fits; final separable magnitude scales are extracted with power iteration. This path does not initialize its directions with the truncated SVD of the source

update. The separately implemented multi-envelope path and the mathematical construction below use SVD, so they must not be conflated with the current single-envelope implementation. The source LoRA alpha/rank multiplier is preserved by the adapter wrapper outside the factorized product, rather than multiplied into the matrix before fitting.

Scope of the solver description. The equations below specify a consensus-ADMM formulation and its conditional analysis. They provide a mathematical reference for the constrained objective, not a claim that every internal update or initialization equals the current DBF code. In particular, the current DBF implementation uses randomized initialization and a cubic penalty ramp, whereas the reference algorithm below illustrates an SVD initialization and residual balancing. Neither formulation is given an unconditional convergence or task-accuracy guarantee.

Executed factor-update routine. For normalized fixed factor C, target T, current factor Z, and scaled dual U, form

$$G = C^\top C + 0.03\ \mathrm{mean}(\mathrm{diag}(C^\top C))I, \quad H = C^\top T, \quad Q = (G + \rho_t I)^{-1}\{H + \rho_t(Z - U)\}.$$

Repeat $Z \leftarrow \mathcal{P}_5(Q + U)$ and $U \leftarrow U + Q - Z$ three times, with $Q \leftarrow (G + I)^{-1}(H + Z - U)$ between consecutive repetitions. Thus the cubic penalty applies only to the first solve of each factor update; the next two use penalty one. Here $\mathcal{P}_5(V) = \mathrm{sign}(V) \odot \sigma uv^\top$, where five power iterations on $|V|$ approximate its leading singular triplet and $\mathrm{sign}(0) = +1$. Power iteration also uses seeded random initialization. Factor norms include 10^{-12} stabilization; default input/output weighting is uniform with 10^{-8} stabilization. The implementation transposes projections when appropriate and restores their orientation. It uses neither SVD initialization, residual balancing, nor a sign-stability stopping rule. Logical size assumes fp16 scales and packed signs; ordinary audit checkpoints need not use that serialized format. Algorithm 1 gives the end-to-end construction and calls this executed routine explicitly; the reference algorithms below analyze different mathematical solvers.

For the left-factor update, the routine receives the transpose of the row-normalized right factor and the transposed working target, returning the transposed left factor. The right-factor update receives the column-normalized left factor and the working target. Normalization applies to each solve's fixed-factor input, not in-place to the stored factor and dual variables.

Setup. Let an fp16 LoRA adapter $\Delta W^\star = A^\star(B^\star)^\top$ of rank r_0 be derived using any standard fine-tuning recipe, and fix a target carrier rank $R \in \mathbb{N}$. The $\ell = 1$ PTQ-LoRDBA problem is the Frobenius least-squares fitting problem

$$\min_{\theta=(B_1,B_2,\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})} \frac{1}{2}\left\|\Delta W^\star - \Delta W(\theta)\right\|_F^2 \text{ s.t. } B_1 \in \{\pm 1\}^{N\times R}, B_2 \in \{\pm 1\}^{R\times M}. \tag{42}$$

The general $\ell \geq 2$ case replaces the single scale triple by a sum of ℓ scale triples; all updates below generalise block-coordinate-wise by viewing $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ as stacked matrices (Algorithm 4). Eq. (42) is non-convex, non-smooth, and combinatorial in (B_1, B_2) but decouples along the three continuous variables for fixed binaries and along each binary for fixed other variables.

Consensus ADMM. To exploit this separability while respecting the discrete constraint, we introduce continuous relaxations $U_1 \in \mathbb{R}^{N\times R}$, $U_2 \in \mathbb{R}^{R\times M}$ of (B_1, B_2) together with binary copies M_1, M_2, and minimise

$$f(U; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) + I_{\{\pm 1\}}(M_1) + I_{\{\pm 1\}}(M_2), \text{ s.t. } U_k - M_k = 0, k \in \{1, 2\}, \tag{43}$$

with $f(U; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{1}{2}\|\Delta W^\star - \mathrm{diag}(\boldsymbol{\alpha})U_1\,\mathrm{diag}(\boldsymbol{\beta})U_2\,\mathrm{diag}(\boldsymbol{\gamma})\|_F^2$ and $I_{\{\pm 1\}}$ as the indicator of the binary set. Using per-block normalisers $n_1 = NR$, $n_2 = RM$, scaled penalties $\widetilde{\rho}_k = \rho/n_k$, and scaled duals Y_1, Y_2, the (Gauss–Seidel) sweep over the consensus-scaled augmented Lagrangian (Boyd et al., 2011; Themelis & Patrinos, 2020) produces the following ordered updates,

where $\widehat{U}^{(t,1)} \coloneqq (U_1, U_2^{(t)})$ and $\widehat{U}^{(t,2)} \coloneqq (U_1^{(t+1)}, U_2)$ make the within-sweep dependency explicit:

$$U_1^{(t+1)} = \arg\min_{U_1} f\big(U_1, U_2^{(t)}; \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) + \frac{\widetilde{\rho}_1}{2}\|U_1 - M_1^{(t)} + Y_1^{(t)}\|_F^2, \tag{44}$$

$$U_2^{(t+1)} = \arg\min_{U_2} f\big(U_1^{(t+1)}, U_2; \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) + \frac{\widetilde{\rho}_2}{2}\|U_2 - M_2^{(t)} + Y_2^{(t)}\|_F^2, \tag{45}$$

$$(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})^{(t+1)} \leftarrow \text{one block-coordinate sweep of } \arg\min_{\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma}} f\big(U^{(t+1)}; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}\big), \tag{46}$$

$$M_k^{(t+1)} = \operatorname{sign}\big(U_k^{(t+1)} + Y_k^{(t)}\big),\ Y_k^{(t+1)} = Y_k^{(t)} + U_k^{(t+1)} - M_k^{(t+1)},\ k \in \{1,2\}. \tag{47}$$

We collectively refer to Eqs. (44)–(45) as the *U-step*. Each U_k-subproblem is a strongly convex Tikhonov least-squares problem in U_k (its Hessian is the sum of a positive-semidefinite normal-equation Gram matrix and $\widetilde{\rho}_k I$), and hence admits a unique minimiser regardless of the conditioning of the cross-covariance. Eq. (46) is a block-coordinate sweep in the three scale variables: with the other two fixed, the subproblem in $\boldsymbol{\alpha}$ decouples into N scalar least-squares (one per coordinate), the subproblem in $\boldsymbol{\gamma}$ into M scalar least-squares, and the subproblem in $\boldsymbol{\beta}$ is an R-dimensional positive semidefinite least-squares system whose normal matrix is the Hadamard product of two Gram matrices and is hence positive semidefinite by the Schur product theorem; the exact normal equations and coordinate-wise minimisers are recorded in Appendix I.1. Eq. (47) is the sign-projection step modulated by the dual Y_k, which biases the projection by the accumulated discretisation residual.

One possible initialization of this mathematical reference solver uses a rank-R truncated SVD:

$$\Delta W^\star \approx U_R S_R V_R^\top, \qquad B_1^{(0)} = \operatorname{sign}(U_R), \quad B_2^{(0)} = \operatorname{sign}(V_R^\top), \quad \boldsymbol{\beta}^{(0)} = \operatorname{diag}(S_R). \tag{48}$$

This is an optional reference initialization, not the randomized-start DBF path used for the current accuracy tables. For the reference solver, set $(M^{(0)}, Y^{(0)}) = (U^{(0)}, 0)$ after binary initialization. A Boyd-style residual-balancing schedule with $(\tau, \mu) = (2, 10)$ may be truncated at $K/2$ to create a fixed-penalty tail. A fixed penalty alone does not establish the additional convergence hypotheses of Appendix E.

Conditional finite stabilisation. Define $Z_k^{(t)} \coloneqq U_k^{(t+1)} + Y_k^{(t)}$, so $M_k^{(t+1)} = \operatorname{sign}(Z_k^{(t)})$. Theorem E.3 assumes that the fixed-penalty tail converges and that its limiting sign tests have a positive margin from zero. Under those conditions the signs eventually stabilize. Theorem E.6 supplies convergence only under the additional bounded-tail, sufficient-decrease, and relative-error hypotheses stated there. These inequalities are not established for every implemented run. In particular, a single pair of equal consecutive binary iterates is not a valid stopping certificate: scales and dual variables may still change the signs later. Algorithm 2 therefore uses a fixed outer-iteration budget.

E Convergence of PTQ-LoRDBA

Throughout this appendix we adopt the shorthands

$$n_1 \coloneqq NR, \qquad n_2 \coloneqq RM, \qquad \widetilde{\rho}_k \coloneqq \rho/n_k, \qquad k \in \{1,2\}, \tag{49}$$

and recall the PTQ-LoRDBA objective

$$f(U; \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \coloneqq \tfrac{1}{2}\big\|\Delta W^\star - \operatorname{diag}(\boldsymbol{\alpha}) U_1 \operatorname{diag}(\boldsymbol{\beta}) U_2 \operatorname{diag}(\boldsymbol{\gamma})\big\|_F^2. \tag{50}$$

For fixed scale vectors and one of the two U-blocks, f is a convex quadratic because the map from that block to the reconstructed matrix is linear. The binary projection step, however, is discontinuous at zero sign-test entries. Consequently, boundedness and Lipschitz smoothness alone do not imply finite binary stabilisation. We therefore state the convergence facts in the exact form used by the paper: a penalty-agnostic sign-margin theorem, and a standard KL-tail theorem whose sufficient-decrease and relative-error hypotheses must hold on the fixed-penalty tail.

Assumption E.1 (Bounded fixed-penalty tail)**.** *There exists an iteration T_ρ after which the penalty is fixed, $\rho^{(t)} \equiv \rho > 0$. On this tail, all iterates $(U^{(t)}, M^{(t)}, Y^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)})$ belong to a compact set $\mathcal{C}$. The U_1- and U_2-subproblems in Eqs.* (44)–(45) *are solved exactly, the scale update is well defined, and the partial gradients $\nabla_{U_1} f$ and $\nabla_{U_2} f$ are Lipschitz on $\mathcal{C}$ with respect to all their arguments.*

This assumption is the bounded-tail regularity used in the convergence statements below. It is not a consequence of discreteness alone. Numerical safeguards may help control continuous least-squares and scale variables, but their presence is not by itself a proof that every hypothesis holds for an implemented run.

Lemma E.2 (Gradient–dual identity on a fixed-penalty tail)**.** *For $t \geq T_\rho$ define the Gauss–Seidel arguments*

$$\widehat{U}^{(t,1)} \;\coloneqq\; (U_1^{(t+1)}, U_2^{(t)}), \qquad \widehat{U}^{(t,2)} \;\coloneqq\; (U_1^{(t+1)}, U_2^{(t+1)}). \tag{51}$$

Then, for every $k \in \{1,2\}$,

$$\rho Y_k^{(t+1)} \;+\; n_k \nabla_{U_k} f\big(\widehat{U}^{(t,k)}; \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) \;+\; \rho\big(M_k^{(t+1)} - M_k^{(t)}\big) \;=\; 0. \tag{52}$$

Proof. The first-order condition of the exact U_k-subproblem is

$$\nabla_{U_k} f\big(\widehat{U}^{(t,k)}; \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) + \widetilde{\rho}_k\big(U_k^{(t+1)} - M_k^{(t)} + Y_k^{(t)}\big) = 0.$$

The dual update gives

$$U_k^{(t+1)} - M_k^{(t)} + Y_k^{(t)} = Y_k^{(t+1)} + \big(M_k^{(t+1)} - M_k^{(t)}\big).$$

Multiplying by n_k and using $n_k \widetilde{\rho}_k = \rho$ yields Eq. (52). □

E.1 Finite stabilisation under a sign margin

The next theorem is the stabilisation statement invoked in the main text. It assumes convergence of the fixed-penalty tail and proves that a positive limiting sign margin forces the binary iterate to freeze.

Theorem E.3 (Finite stabilisation on a fixed-penalty tail)**.** *Fix the penalty parameters $\widetilde{\rho}_1, \widetilde{\rho}_2 > 0$ on the tail $t \geq T_\rho$, and assume that the resulting PTQ-LoRDBA iterates $(U_k^{(t)}, M_k^{(t)}, Y_k^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)})$ converge to a limit $(U_k^\infty, M_k^\infty, Y_k^\infty, \boldsymbol{\alpha}^\infty, \boldsymbol{\beta}^\infty, \boldsymbol{\gamma}^\infty)$ for every $k \in \{1,2\}$. Define*

$$Z_k^{(t)} \;\coloneqq\; U_k^{(t+1)} + Y_k^{(t)}, \qquad Z_k^\infty \;\coloneqq\; U_k^\infty + Y_k^\infty, \qquad k \in \{1,2\}, \tag{53}$$

and suppose the strict sign-margin condition

$$\eta \;\coloneqq\; \min_{k\in\{1,2\}} \min_{a,b} \big|(Z_k^\infty)_{ab}\big| \;>\; 0. \tag{54}$$

Then:

(i) *There exists a finite iteration $T_\star \geq T_\rho$ such that, for every $t \geq T_\star$ and $k \in \{1,2\}$,*

$$M_k^{(t+1)} = \operatorname{sign}(Z_k^\infty) =: M_k^\infty. \tag{55}$$

(ii) *If the U_k-subproblems are solved exactly and f is continuously differentiable in U, the limit satisfies*

$$U_k^\infty = M_k^\infty, \qquad M_k^\infty = \operatorname{sign}(M_k^\infty + Y_k^\infty), \qquad \rho Y_k^\infty + n_k \nabla_{U_k} f(U^\infty; \boldsymbol{\alpha}^\infty, \boldsymbol{\beta}^\infty, \boldsymbol{\gamma}^\infty) = 0. \tag{56}$$

(iii) *Equivalently,*

$$M_k^\infty = \operatorname{sign}\Big(M_k^\infty - \tfrac{1}{\widetilde{\rho}_k} \nabla_{U_k} f(U^\infty; \boldsymbol{\alpha}^\infty, \boldsymbol{\beta}^\infty, \boldsymbol{\gamma}^\infty)\Big). \tag{57}$$

Proof. Since $U_k^{(t+1)} \to U_k^\infty$ and $Y_k^{(t)} \to Y_k^\infty$, we have $Z_k^{(t)} \to Z_k^\infty$ entrywise. Choose $T_\star$ so that $\|Z_k^{(t)} - Z_k^\infty\|_{\max} < \eta/2$ for both k and all $t \geq T_\star$. Then every entry of $Z_k^{(t)}$ has the same sign as the corresponding entry of Z_k^∞, proving Eq. (55) from $M_k^{(t+1)} = \operatorname{sign}(Z_k^{(t)})$.

For $t \geq T_\star$ the dual update becomes $Y_k^{(t+1)} - Y_k^{(t)} = U_k^{(t+1)} - M_k^\infty$. Passing to the limit gives $U_k^\infty = M_k^\infty$. Substituting this identity into $M_k^\infty = \operatorname{sign}(Z_k^\infty)$ gives the second relation in Eq. (56). Finally, for $t \geq T_\star + 1$, the term $M_k^{(t+1)} - M_k^{(t)}$ in Eq. (52) is zero; passing to the limit in Eq. (52) yields the gradient–dual stationarity relation. Solving it for Y_k^∞ and using $n_k/\rho = 1/\widetilde{\rho}_k$ gives Eq. (57). □

Remark E.4. *The sign-margin condition is checkable on a converged fixed-penalty tail. If an entry of Z_k^∞ is exactly zero, the entrywise binary projection is not unique; in that boundary case the robust statement is the argmin form used in Theorem E.6 below rather than the single-valued sign equation.*

E.2 KL CONVERGENCE OF A REGULAR FIXED-PENALTY TAIL

The following theorem is the rigorous convergence statement for the ADMM tail. It is deliberately conditional: the nonconvex binary projection does not yield an unconditional descent estimate without an additional margin or regularisation condition. The assumptions below are the standard sufficient-decrease and relative-error hypotheses used in KL analyses of nonconvex ADMM and block-coordinate methods.

Let

$$\mathcal{X}^{(t)} := \big(U^{(t)}, M^{(t)}, Y^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}, M^{(t-1)}\big), \qquad t \geq T_\rho + 1, \tag{58}$$

where the last component records the one-step memory needed by the usual ADMM Lyapunov functions.

Assumption E.5 (KL-regular ADMM tail)**.** *On the fixed-penalty tail of Assumption E.1, there exist a proper lower-semicontinuous semi-algebraic Lyapunov function Φ on the state variable $\mathcal{X}$, constants $a, b > 0$, and an iteration $T \geq T_\rho + 1$ such that, for every $t \geq T$,*

$$\Phi(\mathcal{X}^{(t)}) - \Phi(\mathcal{X}^{(t+1)}) \geq a\|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|^2, \tag{59}$$

$$\operatorname{dist}\big(0, \partial\Phi(\mathcal{X}^{(t+1)})\big) \leq b\|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\|. \tag{60}$$

The sequence $\{\Phi(\mathcal{X}^{(t)})\}_{t\geq T}$ is bounded below.

The semi-algebraicity requirement is natural here: after introducing the finite binary indicators, all remaining terms in the augmented Lagrangian are polynomial functions of the continuous variables. The substantive algorithm-dependent requirements are the two inequalities (59)–(60); they are the precise conditions under which the KL convergence conclusion is valid.

Theorem E.6 (KL convergence of a regular fixed-penalty tail)**.** *Suppose Assumptions E.1 and E.5 hold. Then the PTQ-LoRDBA fixed-penalty tail has finite length and converges to a single limit*

$$\big(U^{(t)}, M^{(t)}, Y^{(t)}, \boldsymbol{\alpha}^{(t)}, \boldsymbol{\beta}^{(t)}, \boldsymbol{\gamma}^{(t)}\big) \longrightarrow \big(U^{\infty}, M^{\star}, Y^{\infty}, \boldsymbol{\alpha}^{\infty}, \boldsymbol{\beta}^{\infty}, \boldsymbol{\gamma}^{\infty}\big). \tag{61}$$

Moreover:

(i) *The increments of all tail variables are summable, hence square-summable.*

(ii) *The binary iterate terminates in finitely many steps: there is a finite $K^\star$ such that $M^{(t)} = M^\star$ for every $t \geq K^\star$.*

(iii) *The limit satisfies, for each $k \in \{1, 2\}$,*

$$\begin{aligned} U_k^{\infty} &= M_k^{\star}, \\ M_k^{\star} &\in \operatorname*{arg\,min}_{m\in\{\pm 1\}^{n_k}} \|m - (U_k^{\infty} + Y_k^{\infty})\|_F^2, \\ \rho Y_k^{\infty} &+ n_k \nabla_{U_k} f(U^{\infty}; \boldsymbol{\alpha}^{\infty}, \boldsymbol{\beta}^{\infty}, \boldsymbol{\gamma}^{\infty}) = 0. \end{aligned} \tag{62}$$

If every entry of $U_k^{\infty} + Y_k^{\infty}$ is nonzero, the argmin is the singleton $\{\operatorname{sign}(U_k^{\infty} + Y_k^{\infty})\}$.

Proof. By Assumption E.1, the tail sequence has a cluster point. Since Φ is semi-algebraic, it satisfies the Kurdyka–Łojasiewicz inequality at every point of its domain (Bolte et al., 2007; Attouch et al., 2010). Combining the KL inequality at a cluster point with the sufficient-decrease and relative-error estimates (59)–(60) gives the standard finite-length conclusion for descent methods (Attouch et al., 2013; Wang et al., 2019):

$$\sum_{t\geq T} \|\mathcal{X}^{(t+1)} - \mathcal{X}^{(t)}\| < \infty. \tag{63}$$

Hence $\{\mathcal{X}^{(t)}\}_{t\geq T}$ is Cauchy and converges to a single limit. Projecting this convergence onto the variables $(U, M, Y, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ proves Eq. (61), and claim (i) follows immediately from Eq. (63).

For claim (ii), each entry of $\Delta M_k^{(t)} = M_k^{(t+1)} - M_k^{(t)}$ belongs to $\{-2, 0, 2\}$. Therefore, whenever $\Delta M_k^{(t)} \neq 0$, $\|\Delta M_k^{(t)}\|_F \geq 2$. The summability in claim (i) implies that only finitely many such nonzero jumps can occur. Thus the two binary blocks are eventually constant; denote the common tail value by $M^\star$.

It remains to verify stationarity. The dual update gives $Y_k^{(t+1)} - Y_k^{(t)} = U_k^{(t+1)} - M_k^{(t+1)}$. Since the left-hand side tends to zero by claim (i), passing to the limit gives $U_k^\infty = M_k^\star$. The binary update is the exact entrywise projection

$$M_k^{(t+1)} \in \operatorname*{arg\,min}_{m \in \{\pm 1\}^{n_k}} \|m - (U_k^{(t+1)} + Y_k^{(t)})\|_F^2.$$

Taking limits in this finite-dimensional inequality for every $m \in \{\pm 1\}^{n_k}$ yields the argmin condition in Eq. (62). Finally, by claim (ii) the term $M_k^{(t+1)} - M_k^{(t)}$ in Eq. (52) is zero for all sufficiently large t. Passing to the limit in Eq. (52), using the continuity of $\nabla_{U_k} f$ on $\mathcal{C}$, gives the gradient–dual stationarity equation in Eq. (62). If $U_k^\infty + Y_k^\infty$ has no zero entries, the binary projection decouples entrywise and is uniquely given by the sign of that vector, proving the final assertion. □

F QAT FULL: SMOOTH-SIGN STE EXTENSION

QAT Full optimizes real-valued latent carrier matrices U_1, U_2 together with the scale vectors $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$. In contrast, QAT Freeze fixes the initialized binary carriers and optimizes only scales, requiring no sign gradient. In Full's forward pass the binary carriers are $B_1 = \operatorname{sign}(U_1)$ and $B_2 = \operatorname{sign}(U_2)$ inside Eq. (4); its backward pass can use the *smooth-sign* straight-through estimator (Leng et al., 2018; Liu et al., 2020):

$$\frac{\partial}{\partial u} \operatorname{sign}(u) \;\longleftarrow\; \frac{\partial}{\partial u} \tanh(\kappa u) \;=\; \kappa\big(1 - \tanh^2(\kappa u)\big), \tag{64}$$

with $\kappa = 100$ in the historical smooth-sign configuration. The scales $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ are trained in fp32 and do not require an STE. The source adapter in the current accuracy pipeline is trained for two full epochs before compression; it is not a 10% warm-up. Algorithm 3 summarizes the optional Full update, not the default Freeze update. Historical Full schedules below must not be substituted for the current two-epoch Freeze recipe.

G TRAINING MODE COMPARISON

Completed three-seed comparison. Tables 7 and 8 give the seed-level values used to generate the main-text means and sample standard deviations. All runs use seeds 0–2, the same pinned MetaMathQA revision, GSM8K 8-shot strict exact match, and Minerva 4-shot `math_verify`; the Minerva aggregate weights the seven category scores by their evaluated sample counts (5,000 examples total). The displayed cells use completed evaluations with artifact identity, expected item counts, and complete per-task JSON checked before aggregation.

Table 7: Per-seed fresh LLaMA-2-7B records underlying Table 1.

Method	Comparison size (MiB)	Seed	GSM8K	Minerva	Training
LoRAQuant r=16, ρ=0.6763	7.152	0	38.13	8.62	post-hoc from 2-epoch source
		1	41.02	8.58	post-hoc from 2-epoch source
		2	40.41	7.54	post-hoc from 2-epoch source
QAT Freeze r_0=16, R=8	7.152	0	43.14	8.96	2+2 epochs
		1	45.64	8.84	2+2 epochs
		2	42.84	9.00	2+2 epochs
fp16 LoRA r=2	9.53	0	39.42	8.80	4 epochs
		1	42.30	7.98	4 epochs
		2	43.06	7.82	4 epochs
QAT Freeze r_0=16, R=16	9.54	0	48.75	11.22	2+2 epochs
		1	48.29	10.08	2+2 epochs
		2	46.47	10.02	2+2 epochs

The LoRAQuant parameter $\rho = 0.6762906015$ was selected before evaluation by solving the implementable packed-code storage formula against the Freeze $R = 8$ target. The sizes are 59,992,460 and 59,994,112 bits, respectively, a 0.0028% difference. The filename tag "1 BPW" denotes approximately 1.5007 total BPW, not one bit including scales. This logical size excludes framework serialization overhead; audit checkpoints can contain unpacked or floating tensors. LoRAQuant is

itself a post-training method. We use a port of its compression procedure, including the 300-step reconstruction optimization before mixed-bit quantization, on the task-trained source. A small fixed-factor fixture matched the pinned upstream implementation bitwise; this check does not establish identity for every layer and setting. Freeze additionally receives two epochs of task-loss recovery, which LoRAQuant does not receive in this comparison.

For this packed-formula-matched pair, we paired item scores by task and document hash and drew 10,000 within-seed bootstrap samples; a hierarchical descriptive aggregate additionally resampled the three training seeds. The pooled Freeze-minus-LoRAQuant mean differences and 95% descriptive intervals are 4.018 points $[2.022, 5.863]$ on GSM8K and 0.687 points $[-0.087, 1.520]$ on Minerva. Thus the observed GSM8K difference is resolved at the item level, whereas the Minerva interval overlaps zero. With only three training seeds, these intervals describe the observed artifacts rather than asymptotic method-level certainty.

Table 8: Per-seed multi-model records underlying Table 2. Direct fp16 controls use two epochs. The 13B SVD control shares the source and two-epoch recovery schedule with Freeze, not measured compute.

Model	Method	Seed	GSM8K	Minerva
LLaMA-2-13B	Source fp16 LoRA r=16	0	62.93	17.12
		1	62.17	16.92
		2	61.56	17.08
	fp16 LoRA r=2	0	53.37	13.52
		1	54.66	11.94
		2	52.92	11.66
	Same-source SVD r=2 + recovery	0	55.88	11.36
		1	53.83	12.06
		2	54.21	10.50
	QAT Freeze r_0=16, R=16	0	56.71	15.02
		1	58.76	14.74
		2	59.29	14.18
Mistral-7B	Source fp16 LoRA r=16	0	63.61	19.82
		1	64.97	19.84
		2	63.23	20.02
	fp16 LoRA r=2	0	60.96	17.56
		1	61.03	17.72
		2	59.67	18.42
	QAT Freeze r_0=16, R=16	0	59.21	17.46
		1	60.35	17.26
		2	60.50	17.92

Same-source SVD control and completed-evaluation checks. The 13B SVD control starts from the exact seed-matched rank-16 source used by Freeze. A truncated SVD of its scaled update is computed via reduced QR and a small core decomposition; square-root singular values are split across the two rank-2 factors while preserving the target multiplier $\alpha/r = 2$. These floating-point factors receive two recovery epochs using learning rate 5×10^{-5}, the same batches, tokenization, full-sequence CE objective, and scheduler as Freeze. All three training runs completed 98,750 optimizer steps. The comparison does not match trainable parameter counts, FLOPs, or wall-clock time, and it does not establish that this schedule is optimal for rank-2 LoRA.

For the new 13B control, all 24 evaluation cells (three seeds times GSM8K and seven Minerva categories) completed with output checksums and expected item counts. The six source-adapter evaluations comprise 48 analogous cells across 13B and Mistral. Input-document, few-shot-prompt, and generation-argument hashes agree with the corresponding Freeze evaluations. Source checkpoint identities also agree across the source evaluation, compression, and SVD initialization paths. Means and sample standard deviations are calculated from unrounded per-seed scores, with Minerva category counts summing to 5,000 rather than an unweighted category mean. Checkpoint recovery and incremental evaluation were tested separately, including a forced-interruption/restart check; passing that engineering check is not an accuracy result.

Logical payload versus saved tensors and accuracy path. Shape inspection gives logical fp16 source/rank-2 payloads of $119.375/14.921875$ MiB for 13B and $80/10$ MiB for Mistral. With one-bit signs and fp16 scales, Freeze has 14.930420 and 10.006836 MiB logical payload, respectively. The saved Freeze tensors instead contain packed uint8 signs and fp32 scales: their actual tensor payloads are 22.399902 and 15.013672 MiB, excluding file headers and metadata. Accuracy evaluation reconstructs the dense adapter in fp32 and casts it to the bf16 base dtype while keeping it as an unmerged side branch; it does not use the custom packed CUDA timing kernel. Dense LoRA factors are also cast to the activation dtype in this evaluation. The normalized size columns therefore compare a specified 16-bit-coefficient deployment format, not measured file bytes; score preservation after casting each checkpoint's scales to that format remains untested.

Historical training-mode audit. Table 9 reports older records whose source JSON was directly audited. These records are historical candidate evidence: QAT Full and QAT Freeze used different step budgets, and Freeze and Scratch have only one run per displayed setting. They are not pooled with the fresh tables and support neither a Full–Freeze equivalence claim nor a causal claim about Scratch initialization.

Table 9: Audited candidate LLaMA-2-7B math records. Metrics are reported separately; no average is formed across tasks. Steps are per run. The unequal schedules and run counts make cross-mode differences descriptive rather than confirmatory.

Mode	BPW	Size (MB)	GSM8K	Minerva	n	Steps
QAT Full, r=16	1.0	7.15	43.82	7.93	3	6,250
QAT Freeze, r=16	1.0	7.15	44.43	8.58	1	49,374
QAT Scratch, r=16	1.0	7.15	38.59	6.28	1	49,374
QAT Full, r=16	2.0	9.54	46.70	9.17	3	6,250
QAT Freeze, r=16	2.0	9.54	48.07	10.40	1	49,374

GSM8K is extracted as `exact_match,strict-match`$\times 100$ and Minerva as `math_verify,none`$\times 100$. Full uses the arithmetic mean of the per-seed scalar scores stored under `output_seedpaper/s{0,1,2}_bpw{1.0,2.0}`; no example-level pooling is performed. Freeze uses the direct seed-0 records under `out_7b_freeze1` and `out_7bf2`, and Scratch uses the direct seed-0 record under `output/qat_scratch_bpw1`. No confidence interval is reported. Code and dataset revisions are not completely recoverable for every arm; these values are therefore excluded from formal accuracy claims.

Submission-era comparator context. The external values removed from the active ranking remain useful for interpreting which comparisons were intended. The archived table reported the following records; they are not pooled with Table 9 and do not restore a cross-method ranking.

Table 10: Submission-era external comparator records. The source revisions, seed policies, and training budgets are not matched to the audited LoRDBA candidates. Values are therefore descriptive context only.

Method	Size (MB)	GSM8K	Minerva	XSum
fp16 LoRA r=16	76.3	54.38	13.17	18.87
fp16 LoRA r=2	9.5	43.52	8.26	17.85
LoRAQuant ρ=0.9	8.0	42.30	8.94	18.27
LoRAQuant ρ=0.5	5.7	36.09	7.18	17.67
fp16 LoRA r=4	19.1	48.07	10.28	18.23
BD-style column+distill	4.8	31.92	6.46	18.70

Attributing a cross-method difference to the representation alone requires controlling source adapter, recovery budget, and evaluation conditions. These controls are not established for Table 10, whose values therefore provide historical context rather than that causal comparison. BD-style denotes historical factor-wise BitDelta-style variants, not the published full-delta method. The originally reported LoRDBA Minerva value 9.68 is preserved in Table 13 as historical evidence, not promoted to a current result.

Controlled QAT Freeze memory calibration. The H100 NVL calibration used a bf16 LLaMA-2-7B base, batch 4, sequence length 1024, AdamW, one warm-up step, and three measured steps after resetting `torch.cuda.max_memory_allocated`. The reported memory is the maximum allocation over the measured steps, not their average.

Table 11: Controlled training-memory calibration. Packing changes carrier residency, while adapter-local recomputation removes the dominant saved intermediates.

Configuration	Peak (GiB)	ms/step
fp16 LoRA	36.638	477.6
QAT Freeze, float carriers	46.867	573.3
QAT Freeze, packed carriers	46.795	601.2
QAT Freeze, packed + local recomputation	36.365	680.9
QAT Full	55.792	599.4

Bit-packing reduces the resident Freeze adapter representation from 85.79 to 11.92 MiB, but by itself changes peak allocated memory by only 0.072 GiB because saved adapter intermediates dominate. Adapter-local recomputation returns Freeze to approximately the fp16-LoRA peak at a $1.43\times$ step-time cost. The training implementation maintains trainable adapter parameters in fp32 (hence a 152.5 MiB in-memory fp16-LoRA parameter payload); the exported fp16 serving object is 76.25 MiB. These calibration measurements isolate implementation components but are not substitutes for the unmatched accuracy comparison above.

Historical factor-wise BitDelta-style comparison. The 4.8 MB column+distill record quantizes the two LoRA factors separately; it is a BitDelta-style adaptation, not the original full-ΔW method (Liu et al., 2024a). Its archived GSM8K, Minerva, and XSum scores are 31.92, 6.46, and 18.70. These records do not match the current source, recovery, and evaluation protocol, and the exact historical code-to-artifact lineage is incomplete. We retain their numerical values for traceability but do not use them to rank the published BitDelta method, establish a distillation ceiling, or average heterogeneous task metrics.

H Extended Numerical Results

Table 12: Historical B200 training summary retained from the original main table. The frozen base is bf16; fp16 LoRA trains for two epochs and Full adds one QAT epoch. Legacy MB/GB labels and times are reproduced, not treated as a current Freeze or equal-compute measurement. Total time includes source training, PTQ, and QAT for Full.

Method	Size (MB)	Peak (GB)	ms/step	Total min
fp16 LoRA, $r_0 = 16$	76.3	36.3	468	385
QAT Full, nominal 4 BPW	19.1	58.0	899	744
QAT Full, nominal 1 BPW	7.2	55.6	804	718

The following tables preserve submission-era measurements for provenance and broader task context. They are not pooled with the completed three-seed tables. Rank, nominal BPW, "MB", and timing columns retain the original reporting conventions rather than freshly audited total bits, physical-file sizes, or resident memory. Source lineage, schedules, and evaluator revisions are not matched across every row, so these tables do not establish a current cross-method ranking. The originally named Freeze $r16$ row with $44.20/8.18$ was traced to an incomplete Full run: its method label is corrected below and it is excluded from Freeze evidence.

Reconstruction error. Table 16 preserves the historical PTQ measurements normalized by $\|\Delta W^\star\|_F$. Their decrease over the displayed budgets is empirical. The theorem treats one canonical construction, under additional assumptions and with a different denominator in its primary relative bound; it does not compare this implemented carrier-rank sweep or predict task accuracy.

Table 13: Historical LLaMA-2-7B records, not a matched current comparison. Metrics are GSM8K exact match (originally reported as 8-shot), Minerva Math (4-shot), and XSum ROUGE-L F1. Nominal BPW, sizes, and schedules retain the original reporting convention; see the scope note above.

				Math		Summ	
Method	**Rank**	**BPW**	**Size (MB)**	**GSM8K**	**Minerva**	**XSum**	**Time**[‡]
Reference (full-precision LoRA)							
LoRA FP16	r16	16.0	76.3	54.38	13.17	18.87	–
LoRA FP16	r64	16.0	305.0	58.68	15.52	–	–
Existing baselines (post-hoc, compressing LoRA FP16 r=16)							
LoRAQuant ρ=0.5	r16	1.20	5.7	36.09	7.18	17.67	+36 min
LoRAQuant ρ=0.9	r16	1.68	8.0	42.30	8.94	18.27	+55 min
BD-style scalar	r16	1.00	4.8	30.71	6.38	–	+0 min
BD-style column	r16	1.00	4.8	31.84	6.28	–	+0 min
BD-style col+distill	r16	1.00	4.8	31.92	6.46	18.70	+17 min
BD-style row	r16	2.00	9.5	32.83	6.72	–	+0 min
PTQ-LoRDBA (post-hoc, training-free, $\sim$2 min compression)							
PTQ-LoRDBA	r16	1.0	7.2	27.14	5.74	7.24	+2 min
PTQ-LoRDBA	r16	2.0	9.5	34.04	6.36	–	+2 min
PTQ-LoRDBA	r16	4.0	19.1	45.34	10.04	–	+2 min
PTQ-LoRDBA	r64	1.0	19.1	35.18	7.88	–	+2 min
QAT Full (LoRA FP16 $\rightarrow$ PTQ $\rightarrow$ 1-epoch QAT refinement)							
QAT Full	r16	1.0	7.2	44.43	9.68	18.32	+11h
QAT Full	r16	2.0	9.5	48.60	9.68	18.49	+11h
QAT Full	r16	4.0	19.1	53.45	12.28	18.49	+11h
QAT Full	r64	1.0	19.1	53.60	12.04	18.60	+15h
QAT Freeze (PTQ $\rightarrow$ freeze binary signs, train scaling vectors only)							
Full, incomplete[†]	r16	1.0	7.2	44.20	8.18	–	+11h
QAT Freeze	r64	1.0	19.1	50.80	11.38	–	+15h
QAT Scratch (end-to-end from random init, no FP16 required)							
QAT Scratch	r16	2.0	9.5	39.42	7.32	–	8h
QAT Scratch	r16	4.0	19.1	45.94	8.50	–	9h
QAT Scratch	r64	1.0	19.1	38.51	5.74	–	8h

"–" = not evaluated. [†]Originally mislabeled Freeze; the log records an unfinished Full trajectory (21,500/24,687 steps). This row is not Freeze evidence.
[‡]Time = additional cost after LoRA FP16 training (Math$\sim$5h, Summ$\sim$3–6h on H100). QAT Scratch does not require a pre-trained FP16 adapter. All runs use a single NVIDIA H100 (80 GB) or B200.

I PSEUDOCODE

I.1 CLOSED-FORM SCALE UPDATES FOR PTQ-LORDBA

This subsection derives the per-axis least-squares updates used by Eq. (46). Throughout, let R denote the LoRDBA binary carrier rank of Definition 3.1 (so that $B_1 \in \{\pm 1\}^{N\times R}$ and $B_2 \in \{\pm 1\}^{R\times M}$); to avoid the notation clash, we use the symbol T for the regression target, defined below. We treat the canonical single-envelope case $\ell = 1$; the multi-envelope case applies the same updates block-coordinate-wise to each envelope while holding the remaining envelopes fixed (see the residual definition at the end of this subsection).

Setup. Fix the binary carriers B_1, B_2 and define the regression target

$$T \;\coloneqq\; \Delta W^{\star} \;\in\; \mathbb{R}^{N\times M}. \tag{65}$$

Table 14: Historical LLaMA-3.2-3B records, not a matched current comparison. GSM8K, Minerva Math, and XSum use the original task-specific reporting. Nominal BPW and "MB" are legacy conventions, not freshly audited total BPW and physical bytes.

				Math		Summ	
Method	**Rank**	**BPW**	**Size (MB)**	**GSM8K**	**Minerva**	**XSum**	**Time**[‡]
Reference (full-precision LoRA)							
LoRA FP16	r16	16.0	46.4	59.36	17.56	15.38	–
LoRA FP16	r64	16.0	185.5	61.56	20.60	–	–
Existing baselines (post-hoc, compressing LoRA FP16 r=16)							
LoRAQuant ρ=0.5	r16	1.22	3.6	45.26	14.10	15.02	+8 min
LoRAQuant ρ=0.7	r16	1.42	4.1	47.31	14.64	–	+11 min
LoRAQuant ρ=0.9	r16	1.71	5.0	51.63	15.20	15.49	+17 min
BD-style scalar	r16	1.00	2.9	42.46	13.00	–	+0 min
BD-style column	r16	1.00	2.9	42.68	12.66	–	+0 min
BD-style col+distill	r16	1.00	2.9	42.61	12.68	16.78	+15 min
BD-style row	r16	2.00	5.8	43.29	13.18	–	+0 min
PTQ-LoRDBA (post-hoc, training-free, ∼2 min compression)							
PTQ-LoRDBA	r16	1.5	4.4	41.32	12.20	15.52	+2 min
PTQ-LoRDBA	r16	2.0	5.8	48.22	12.86	15.55	+2 min
PTQ-LoRDBA	r16	4.0	11.6	55.04	16.50	15.19	+2 min
PTQ-LoRDBA	r64	1.0	11.6	49.81	14.60	–	+2 min
QAT Full (LoRA FP16 → PTQ → 1-epoch QAT refinement)							
QAT Full	r16	1.5	4.4	52.77	13.86	15.34	+7h
QAT Full	r16	2.0	5.8	55.42	15.34	15.49	+7h
QAT Full	r16	4.0	11.6	56.63	17.44	15.35	+7h
QAT Full	r64	1.0	11.6	56.71	16.32	–	+7h
QAT Freeze (PTQ → freeze binary signs, train scaling vectors only)							
QAT Freeze	r16	1.5	4.4	50.27	13.38	15.54	+4h
QAT Freeze	r64	1.0	11.6	57.16	16.46	–	+4h
QAT Freeze	r16	2.0	5.8	57.01	15.20	15.18	+4h
QAT Freeze	r16	4.0	11.6	58.98	17.10	15.38	+4h
QAT Scratch (end-to-end from random init, no FP16 required)							
QAT Scratch	r16	1.5	4.4	45.56	12.42	14.72	7h
QAT Scratch	r16	2.0	5.8	50.19	12.76	15.05	7h
QAT Scratch	r16	4.0	11.6	54.66	14.14	15.07	7h
QAT Scratch	r64	1.0	11.6	48.29	13.04	14.54	7h

"–" = not evaluated. **Bold** = best among compressed methods at the corresponding size tier.
[‡]Time = additional cost after LoRA FP16 training (∼5h on H100). QAT Scratch does not require a pre-trained FP16 adapter. All runs use a single NVIDIA H100 (80 GB) or B200.

The PTQ-LoRDBA scale subproblem is the joint minimisation

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^N,\boldsymbol{\beta}\in\mathbb{R}^R,\boldsymbol{\gamma}\in\mathbb{R}^M} \frac{1}{2}\big\|T - \operatorname{diag}(\boldsymbol{\alpha})B_1\operatorname{diag}(\boldsymbol{\beta})B_2\operatorname{diag}(\boldsymbol{\gamma})\big\|_F^2, \tag{66}$$

which we solve by per-axis block-coordinate descent. Each per-axis subproblem admits a closed-form least-squares solution, derived next.

$\boldsymbol{\alpha}$-update (rowwise). For fixed $B_1, B_2, \boldsymbol{\beta}, \boldsymbol{\gamma}$, set

$$D_\alpha \;:=\; B_1\operatorname{diag}(\boldsymbol{\beta})B_2\operatorname{diag}(\boldsymbol{\gamma}) \;\in\; \mathbb{R}^{N\times M}. \tag{67}$$

Substituting Eq. (67) into Eq. (66) and expanding the Frobenius norm rowwise,

$$\frac{1}{2}\big\|T - \operatorname{diag}(\boldsymbol{\alpha})D_\alpha\big\|_F^2 \;=\; \sum_{i=1}^{N}\frac{1}{2}\big\|T_{i,:} - \alpha_i(D_\alpha)_{i,:}\big\|_2^2,$$

Table 15: Historical adapter-mode QAT Full training statistics on LLAMA-2-7B (MetaMathQA). LoRA pre-trained for 2 epochs; QAT for 1 epoch (bf16 base, single NVIDIA B200). "Peak" is the device-side QAT training peak memory; "ms/step" is the average wall-clock per QAT training step.

Setting	Mode	Steps	ms/step	Time (min)	Peak (GB)	Hardware
$r = 16$, LoRDBA @1BPW	QAT Full	24687	804	331	55.6	B200
$r = 16$, LoRDBA @2BPW	QAT Full	24687	856	352	56.1	B200
$r = 16$, LoRDBA @4BPW	QAT Full	24687	899	370	58.0	B200
$r = 64$, LoRDBA @1BPW	QAT Full	24687	913	376	58.0	B200

Table 16: Historical PTQ relative Frobenius reconstruction error $\|\Delta W^\star - \Delta W(\theta)\|_F / \|\Delta W^\star\|_F$ on LLAMA-2-7B (averaged across projections).

BPW	0.5	1.0	2.0	4.0	8.0
effective carrier rank R	~ 24	~ 47	~ 94	~ 190	~ 380
Relative error ratio	0.358	0.198	0.099	0.040	0.015

which decouples over the row index i. Each scalar problem

$$\min_{\alpha_i \in \mathbb{R}} \ \frac{1}{2}\big\|T_{i,:} - \alpha_i (D_\alpha)_{i,:}\big\|_2^2 \tag{68}$$

has the unique minimum-norm least-squares solution

$$\alpha_i \ = \ \begin{cases} \dfrac{\langle T_{i,:}, (D_\alpha)_{i,:}\rangle}{\|(D_\alpha)_{i,:}\|_2^2}, & \|(D_\alpha)_{i,:}\|_2 > 0, \\ 0, & \|(D_\alpha)_{i,:}\|_2 = 0, \end{cases} \qquad i \in [N]. \tag{69}$$

$\boldsymbol{\gamma}$-update (columnwise). For fixed $B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\beta}$, set

$$D_\gamma \ \coloneqq \ \mathrm{diag}(\boldsymbol{\alpha}) B_1 \, \mathrm{diag}(\boldsymbol{\beta}) B_2 \ \in \ \mathbb{R}^{N \times M}. \tag{70}$$

The objective Eq. (66) expands columnwise as

$$\frac{1}{2}\big\|T - D_\gamma \, \mathrm{diag}(\boldsymbol{\gamma})\big\|_F^2 \ = \ \sum_{j=1}^{M} \frac{1}{2}\big\|T_{:,j} - \gamma_j (D_\gamma)_{:,j}\big\|_2^2,$$

and analogously decouples over $j \in [M]$. The unique minimum-norm least-squares solution is

$$\gamma_j \ = \ \begin{cases} \dfrac{\langle T_{:,j}, (D_\gamma)_{:,j}\rangle}{\|(D_\gamma)_{:,j}\|_2^2}, & \|(D_\gamma)_{:,j}\|_2 > 0, \\ 0, & \|(D_\gamma)_{:,j}\|_2 = 0, \end{cases} \qquad j \in [M]. \tag{71}$$

$\boldsymbol{\beta}$-update (R-dimensional). The $\boldsymbol{\beta}$-update is an R-dimensional least-squares problem. For fixed $B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\gamma}$, define

$$P \ \coloneqq \ \mathrm{diag}(\boldsymbol{\alpha}) B_1 \ \in \ \mathbb{R}^{N \times R}, \qquad Q \ \coloneqq \ B_2 \, \mathrm{diag}(\boldsymbol{\gamma}) \ \in \ \mathbb{R}^{R \times M}. \tag{72}$$

Expanding the bilinear identity

$$\mathrm{diag}(\boldsymbol{\alpha}) B_1 \, \mathrm{diag}(\boldsymbol{\beta}) B_2 \, \mathrm{diag}(\boldsymbol{\gamma}) \ = \ P \, \mathrm{diag}(\boldsymbol{\beta}) Q \ = \ \sum_{k=1}^{R} \beta_k P_{:,k} Q_{k,:}, \tag{73}$$

where the second equality uses the column–row decomposition $\mathrm{diag}(\boldsymbol{\beta}) = \sum_k \beta_k \boldsymbol{e}_k \boldsymbol{e}_k^\top$ and $P \boldsymbol{e}_k \boldsymbol{e}_k^\top Q = P_{:,k} Q_{k,:}$. Vectorising Eq. (73) gives $\mathrm{vec}(P \, \mathrm{diag}(\boldsymbol{\beta}) Q) = H \boldsymbol{\beta}$ with $H \in \mathbb{R}^{NM \times R}$ defined by

$$H_{:,k} \ \coloneqq \ \mathrm{vec}(P_{:,k} Q_{k,:}), \qquad k \in [R]. \tag{74}$$

Hence Eq. (66) reduces, in the $\boldsymbol{\beta}$-direction, to the standard linear least-squares problem

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^R} \ \frac{1}{2}\big\|\mathrm{vec}(T) - H\boldsymbol{\beta}\big\|_2^2, \tag{75}$$

with normal equations $H^\top H\boldsymbol{\beta} = H^\top \mathrm{vec}(T)$. We record these in our notation as

$$G_\beta \boldsymbol{\beta} \ = \ \boldsymbol{h}_\beta, \tag{76}$$

where, for $k, l \in [R]$,

$$(G_\beta)_{kl} \ = \ \langle P_{:,k}, P_{:,l}\rangle\langle Q_{k,:}, Q_{l,:}\rangle, \qquad (\boldsymbol{h}_\beta)_k \ = \ \langle T, P_{:,k}Q_{k,:}\rangle_F. \tag{77}$$

Equivalently, in matrix form,

$$G_\beta \ = \ (P^\top P) \odot (QQ^\top), \tag{78}$$

where $\odot$ is the Hadamard product. Since $P^\top P$ and $QQ^\top$ are positive semidefinite, G_β is positive semidefinite by the Schur product theorem. When G_β is nonsingular, the update is $\boldsymbol{\beta} = G_\beta^{-1}\boldsymbol{h}_\beta$; when singular, we use the unique minimum-norm least-squares solution

$$\boldsymbol{\beta} \ = \ G_\beta^{\dagger}\boldsymbol{h}_\beta, \tag{79}$$

where $G_\beta^{\dagger}$ denotes the Moore–Penrose pseudoinverse.

Generalisation to envelope rank $\ell \geq 2$. For envelope rank $\ell \geq 2$, the same per-axis updates Eqs. (69), (71), and (79) apply to one envelope $i \in [\ell]$ at a time after replacing the target T by the per-envelope residual

$$T^{(i)} \ \coloneqq \ \Delta W^\star - \sum_{m\neq i} \mathrm{diag}(\boldsymbol{\alpha}^{(m)})B_1\,\mathrm{diag}(\boldsymbol{\beta}^{(m)})B_2\,\mathrm{diag}(\boldsymbol{\gamma}^{(m)}), \qquad i \in [\ell]. \tag{80}$$

Cycling the envelope index $i \in [\ell]$ in any order produces a block-coordinate descent on Eq. (66) extended to ℓ envelopes, and each per-envelope update inherits the closed form above with T replaced by $T^{(i)}$.

Algorithm 2 Mathematical reference PTQ solver at $\ell = 1$ (scaled consensus ADMM with optional SVD start). This is not the current randomized-start DBF implementation.

Require: fp16 LoRA factors $A^\star, B^\star$, carrier rank R, outer iterations K, penalty schedule $\{\rho^{(t)}\}$
Ensure: LoRDBA adapter $\theta^\star = (B_1^\star, B_2^\star, \boldsymbol{\alpha}^\star, \boldsymbol{\beta}^\star, \boldsymbol{\gamma}^\star)$

$(U_\Sigma, S_\Sigma, V_\Sigma) \leftarrow \mathrm{thinSVD}(A^\star(B^\star)^\top, \mathrm{rank} = R)$ ▷ Eq. (48)
$U_1^{(0)} \leftarrow \mathrm{sign}(U_\Sigma)$, $U_2^{(0)} \leftarrow \mathrm{sign}(V_\Sigma^\top)$, $\boldsymbol{\beta}^{(0)} \leftarrow \mathrm{diag}(S_\Sigma)$
$M_k^{(0)} \leftarrow U_k^{(0)}$, $Y_k^{(0)} \leftarrow 0$ for $k \in \{1, 2\}$
Initialise $(\boldsymbol{\alpha}^{(0)}, \boldsymbol{\gamma}^{(0)})$ by one block-coordinate sweep of Eq. (46) at $(U^{(0)}, \boldsymbol{\beta}^{(0)})$
for $t = 0, 1, \ldots, K-1$ **do**
    Update $U_1^{(t+1)}$, $U_2^{(t+1)}$ by Tikhonov least-squares (Eqs. (44)–(45))
    Update $(\boldsymbol{\alpha}^{(t+1)}, \boldsymbol{\beta}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)})$ by one block-coordinate sweep of Eq. (46) (per-axis closed-form least-squares)
    $M_k^{(t+1)} \leftarrow \mathrm{sign}(U_k^{(t+1)} + Y_k^{(t)})$ for $k \in \{1, 2\}$
    $Y_k^{(t+1)} \leftarrow Y_k^{(t)} + U_k^{(t+1)} - M_k^{(t+1)}$ for $k \in \{1, 2\}$
    Residual-balance $\rho^{(t+1)}$ (Boyd, truncated at $K/2$)
    Continue to the fixed budget K; equal consecutive signs alone do not terminate the loop
end for
return $(M_1^{(K)}, M_2^{(K)}, \boldsymbol{\alpha}^{(K)}, \boldsymbol{\beta}^{(K)}, \boldsymbol{\gamma}^{(K)})$

Algorithm 3 Optional QAT Full (smooth-sign STE), not QAT Freeze.

Require: base model W_0, stream $\{(\boldsymbol{x}_t, \boldsymbol{y}_t)\}$, carrier rank R, STE temperature κ, optimiser opt

Initialize (U_1, U_2) from the task-trained source adapter and its PTQ factors
Initialise scales $(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$ by one block-coordinate sweep of Eq. (46) (per-axis closed-form least-squares)
for each training step t **do**
    Forward: $\Delta W \leftarrow \mathrm{diag}(\boldsymbol{\alpha})\,\mathrm{sign}(U_1)\,\mathrm{diag}(\boldsymbol{\beta})\,\mathrm{sign}(U_2)\,\mathrm{diag}(\boldsymbol{\gamma})$
    Compute task loss $\ell_t(W_0 + \Delta W; \boldsymbol{x}_t, \boldsymbol{y}_t)$
    Backward: replace $\partial\,\mathrm{sign}(U)/\partial U$ by $\kappa(1 - \tanh^2(\kappa U))$ in autograd
    Step $(U_1, U_2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \leftarrow \mathsf{opt}(\cdot, \nabla \ell_t)$
end for
return $(B_1, B_2, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) \leftarrow (\mathrm{sign}(U_1), \mathrm{sign}(U_2), \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})$

Algorithm 4 Mathematical reference PTQ extension at envelope rank $\ell \geq 1$, not the current single-envelope DBF execution path.

Require: fp16 LoRA factors $A^\star, B^\star$, carrier rank R, envelope rank $\ell \geq 1$, outer iterations K

$(U_\Sigma, S_\Sigma, V_\Sigma) \leftarrow \mathrm{thinSVD}(A^\star(B^\star)^\top, \mathrm{rank} = R)$
Split singular spectrum into ℓ disjoint index sets $\mathcal{I}_1, \ldots, \mathcal{I}_\ell \subseteq [R]$; initialise $\boldsymbol{\beta}^{(i)}$ from $\{(S_\Sigma)_{jj}\}_{j \in \mathcal{I}_i}$ (extended by zeros to $\mathbb{R}^R$)
Initialise (B_1, B_2) by Eq. (48), and $\{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\gamma}^{(i)})\}_{i=1}^{\ell}$ by solving the per-envelope scale step
for $t = 0, 1, \ldots, K-1$ **do**
    (U_1, U_2)-update: outer-loop sign-ADMM step (Algorithm 2, continuous, binary, and dual updates) with f replaced by the envelope-sum objective $\frac{1}{2}\|\Delta W^\star - \sum_i \mathrm{diag}(\boldsymbol{\alpha}^{(i)}) U_1 \,\mathrm{diag}(\boldsymbol{\beta}^{(i)}) U_2 \,\mathrm{diag}(\boldsymbol{\gamma}^{(i)})\|_F^2$
    Update $\{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)})\}_{i=1}^{\ell}$ by block-coordinate closed-form least-squares, cycling over envelope index i
end for
return $\big(B_1, B_2, \{(\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)})\}_{i=1}^{\ell}\big)$

J Inference-Kernel Details

Equation (7) expresses the LoRDBA forward pass as three element-wise scale fusions and two binary matmuls. Each binary matmul $\boldsymbol{z} = B^\top \boldsymbol{w}$ with $B \in \{\pm 1\}^{p \times q}$ and fp16 input $\boldsymbol{w} \in \mathbb{R}^p$ computes, for every $j \in [q]$, the exact sign-accumulation

$$z_j \;=\; \sum_{i=1}^{p} B_{ij} w_i \;=\; \sum_{i : B_{ij} = +1} w_i \;-\; \sum_{i : B_{ij} = -1} w_i, \tag{81}$$

i.e. a signed sum of fp16 values with no floating-point multiplications. On contemporary GPUs this primitive could be implemented with a specialized binary-weight kernel such as `gemlite` (Badri & Shaji, 2023). Neither the current custom-CUDA diagnostic nor the historical PyTorch fallback uses GemLite. The frozen `FusedDBFLinear` source dispatches CUDA inputs to `_fused_dbf_torch`, whose two binary matmuls iterate over output chunks in Python, unpack and cast bit masks, and invoke `torch.addmm`. The native extension is selected only for CPU inputs. The formal harness verified that all 224 target adapters were packed `FusedDBFLinear` modules and that the dense correctness reference was removed before timing, so the reference result is valid for this packed fallback path; it is not a measurement of the custom CUDA prototype in the latest block or of GemLite. Repeated unpacking, casts, Python chunk loops, and small matrix launches are plausible sources of the observed slowdown, but no GPU profile was collected and causal attribution is therefore not claimed.

Latest custom-CUDA diagnostic. After the formal run, we implemented a separate CUDA extension that repacks each row-major uint8 carrier once into 32-bit words and keeps the carrier packed throughout inference. For the ℓ=1 checkpoint, the five factors are mapped as $D_{\boldsymbol{\alpha}} \to B_1 \to D_{\boldsymbol{\beta}} \to B_2 \to D_{\boldsymbol{\gamma}}$; temporary dense weights are used only by the correctness oracle and removed before timing. On an idle H100 NVL, a frozen one-pair B=1, C=512, D=128

diagnostic passed 128/128 generated-token agreement, the selected-logit tolerance, and 224/224 custom-adapter coverage. Custom/fp16 mean TPOT was $26.128/25.361$ ms ($1.030\times$), but the final fp16 sample was 316.944 ms while 127/128 fp16 samples were below 30 ms. We retain the outlier; descriptive p50 and p95 ratios were 1.129 and 1.123. The isolated q-, gate-, and down-projection adapter matmuls were $1.729\times$, $1.765\times$, and $4.194\times$ slower than fp16, while being $35.03\times$, $82.75\times$, and $14.40\times$ faster than the PyTorch fallback. This single unbalanced pair has no process-level interval and is reported as exploratory evidence, not as a latency-equivalence or speedup result. Table 17 summarises the arithmetic intensity independently of this implementation result.

Historical fallback measurement. The earlier six-pair H100 benchmark measured a PyTorch packed fallback, not the current custom CUDA kernel. Its mean TPOT was 27.266 ms for fp16 LoRA and 413.446 ms for LoRDBA; the geometric-mean paired ratio was 15.164 with interval $[14.955, 15.383]$. The base-only reference was 17.138 ms, and adapter residency was 76.25 MiB for fp16 versus 11.92 MiB for the fallback. These values are retained to distinguish an implementation bottleneck from representation storage. The prototype diagnostic above replaces this fallback path, but its single process pair is not a precision-matched repetition of the six-pair benchmark.

Table 17: Arithmetic-intensity breakdown of the adapter matmul on a host projection of shape $N \times M$. The fp16/INT4 LoRA baselines use rank r_0; LoRDBA uses binary carrier rank R and envelope rank ℓ (the table is for $\ell = 1$). One fp16 multiplication and one fp16 addition count as two separate ops. LoRDBA trades two fp16 GEMMs for two sign-accumulation matmuls (zero floating-point multiplications on the inner edges) plus three channel-wise vector scales that may be fused into the residual-stream epilogue with the scale traffic already accounted for in the bytes column.

Method	Bytes loaded (per adapter application)	Arithmetic (ops/application)	Dominant kernel
LoRA fp16	$2r_0(N+M)$	$2r_0(N+M)$, half muls / half adds	fp16 GEMM
Hypothetical INT4 factors	$0.5r_0(N+M)$	$2r_0(N+M)$, half muls / half adds	INT4 GEMM
BitDelta (full-Δ)	$NM/8 + O(1)$	NM fp16 adds	addition-only GEMM
LoRDBA ($\ell = 1$)	$R(N+M)/8 + 2(N+R+M)$	$R(N+M)$ fp16 adds $+O(N+R+M)$ muls	sign-accumulation GEMM

K Implementation Details and Compute

Current accuracy recipe. The completed three-seed math tables use seeds $0, 1, 2$, a frozen bf16 base, and all seven attention/MLP projection types. The source LoRA has rank $r_0 = 16$, alpha 32, and two epochs of MetaMathQA training. Direct rank-2 controls use alpha 4, preserving the multiplier 2. Source and direct training use AdamW with learning rate 2×10^{-4}. Compression uses 100 DBF outer iterations. Freeze fixes the carriers and trains scales for two epochs at 5×10^{-5} on full-sequence cross-entropy; the 13B SVD control uses the same recovery learning rate and schedule. AdamW uses betas $(0.9, 0.95)$, zero weight decay, gradient clipping at 1, and 3% warm-up followed by cosine decay with minimum learning-rate multiplier 0.01. Maximum sequence length is 1,024. The 13B runs use microbatch 2 and effective batch 8 (98,750 steps per two-epoch stage); 7B and Mistral use microbatch 4 and effective batch 16 (49,374 steps). The 7B direct-rank-2 comparison uses four epochs; 13B and Mistral direct controls use two. Matching epoch counts does not match FLOPs or wall-clock time. MetaMathQA contains 395,000 training examples at revision `aa4f34d3d2d3231299b5b03d9b3e5a20da45aa18`; the current evaluator uses `lm-evaluation-harness` 0.4.11.

Evaluation uses `lm_eval` 0.4.11, greedy generation, GSM8K 8-shot strict exact match on 1,319 examples, and Minerva 4-shot `math_verify` on 5,000 examples across seven categories. The Minerva aggregate is sample-count weighted, not an unweighted mean of category means. Source adapters, dataset revisions, evaluator configuration, per-seed outputs, and checkpoint identities are kept separate from historical records.

Storage accounting for current 7B artifacts. With $39{,}976{,}960$ source-rank-16 parameters across 224 projections, the $R = 8$, $\ell = 1$ packed representation plus fp16 scales requires $59{,}994{,}112$ logical bits: 1.500717 BPW and 7.151855 MiB. The corresponding $R = 16$ representation requires 2.001434 BPW and 9.538086 MiB. These are approximately 1.50 and 2.00 total BPW, although historical filenames use "1 BPW" and "2 BPW". This formula does not include framework checkpoint headers or resident-kernel padding; measured files and runtime allocations are separate quantities.

Historical solver and Full configuration. The original solver description used $K = 100$ outer iterations, a Boyd residual-balancing schedule with $(\tau, \mu) = (2, 10)$, and warm penalty $\rho^{(0)} = \|\Delta W^\star\|_F^2/(NR + RM)$. The consensus-ADMM pseudocode describes this mathematical solver; it is not a byte-for-byte specification of every archived DBF implementation. In particular, the fixed-iteration stopping correction does not establish identity of all internal update rules. The historical Full experiments used one-epoch refinement and a smooth-sign parameter $\kappa = 100$ in the B200 block. They must not be read as the current two-epoch Freeze experiment.

Compute budget. Table 18 preserves the original approximate H100 timing records. Its coarse times and historical schedule labels are not matched wall-clock measurements for the current accuracy tables. The separately measured controlled H100 step-time and memory results appear in Table 11; neither table should be used to infer equal end-to-end compute from equal epoch counts.

Table 18: Historical approximate adapter-production times on a single NVIDIA H100 (80GB), retaining original schedule labels. These are not current matched-compute measurements.

Stage	LLaMA-2-7B	LLaMA-3.2-3B
fp16 LoRA fine-tuning	∼5 h	∼5 h
PTQ-LoRDBA ($\ell = 1$)	∼2 min	∼2 min
QAT Full ($\ell = 1$, 2 ep)	∼11 h	∼7 h
QAT Freeze ($\ell = 1$, 2 ep)	∼11–15 h	∼7 h
QAT Scratch ($\ell = 1$, 2 ep)	∼8–9 h	∼7 h

L Additional Task Evidence

These supplementary evaluations test two useful aspects of compact adapters: recovery of a task-trained code adapter after compression, and preservation of a math-trained source adapter's behavior

on additional tasks. We report the completed comparison arms with their logical sizes and seed variability. The main GSM8K/Minerva comparisons retain their original checkpoints and evaluation protocol.

L.1 SCALE-ONLY RECOVERY ON HUMANEVAL

Training and evaluation. We train fresh LLaMA-2-7B source adapters on Magicoder-Evol-Instruct-110K (dataset revision `b0079beaa0361d82412520b873715bee59cc7dd4`), with source rank $r_0 = 16$, $\alpha = 32$, two epochs, maximum sequence length $4{,}096$, effective batch size 16, and learning rate 2×10^{-4}. PTQ fits carrier rank $R = 16$ to the corresponding source. Freeze then updates only scales for two epochs at learning rate 5×10^{-5}. Direct fp16 LoRA starts at rank 2, $\alpha = 4$, and trains for four epochs. This matches its dataset-pass budget to the two source plus two Freeze epochs. LoRAQuant uses the same seed-matched source, with a common $\rho = 0.9799053$ selected from seed-0 spectra to match the 9.538-MiB packed formula; selection uses size only. PTQ and LoRAQuant receive no additional CE recovery.

HumanEval uses all 164 problems, 20 sampled completions per problem, temperature 0.2, top-$p = 0.95$, and at most 512 new tokens. The fixed prompt, stopping, and code-extraction implementation is shared across arms. A pinned BigCode evaluator runs the generated programs in an isolated sandbox. We compute pass@1 from the successful-completion fraction over all $3{,}280$ completions per seed, and then report the mean and sample standard deviation across complete seeds. Four disjoint 41-problem shards must all have validated score receipts before a seed enters the aggregate. The direct rank-2 seed-2 run lacks two final shard receipts following the reservation time limit; its partial score is not pooled with complete seeds.

Table 19: Magicoder-trained LLaMA-2-7B on HumanEval (pass@1, percent). Mean ± sample standard deviation over complete training seeds.

Method	MiB	Epochs	HumanEval	Seeds
Source fp16 $r = 16$	76.250	2	19.10±4.30	3
Direct fp16 $r = 2$	9.531	4	16.10±1.12	2
PTQ $R = 16$	9.538	2 + PTQ	1.69±0.39	3
LoRAQuant	9.537	2 + post-hoc	4.19±2.36	3
QAT Freeze $R = 16$	9.538	2 + 2	15.68±5.31	3

Source, PTQ, LoRAQuant, and Freeze use seeds 0–2. Direct rank-2 LoRA has complete scores for seeds 0 and 1; seed 2 lacks two final shard receipts and is excluded from its mean. Source training plus Freeze recovery and direct rank-2 training each use four dataset passes; PTQ and LoRAQuant have no additional CE recovery. Code LoRAQuant is size-calibrated independently of its task score: $\rho = 0.9799053$, sizes $9.5373/9.5432/9.5310$ MiB across seeds. Logical sizes use the same convention as Table 21.

Recovery at compact size. Freeze improves HumanEval by 13.99 points over its PTQ initialization, from 1.69 ± 0.39 to 15.68 ± 5.31, with an improvement in every paired seed. At 9.538 MiB it retains 82.12% of the rank-16 source's mean pass@1 while using 12.51% of its 76.250-MiB logical payload (approximately $8\times$ smaller). This percentage is a ratio of scores, not the fraction of an improvement over an unadapted base. The result extends the evidence for scale-only recovery to task-specific code generation. Table 19 also places this recovery alongside the size-calibrated post-hoc comparator and the directly trained compact fp16 control. The latter has two complete seeds and mean 16.10 ± 1.12; the observed means do not establish equivalence or a general ranking of compact training methods. Per-seed values are in Table 20.

Table 20: HumanEval seed-level scores underlying Table 19.

Method	Seed 0	Seed 1	Seed 2
Source fp16 $r = 16$	23.41	14.82	19.05
Direct fp16 $r = 2$	15.30	16.89	–
PTQ $R = 16$	1.40	1.52	2.13
LoRAQuant	4.12	1.86	6.59
QAT Freeze $R = 16$	20.70	10.12	16.22

L.2 PRESERVATION ON ADDITIONAL MULTIPLE-CHOICE TASKS

We reuse the LLaMA-2-7B and 13B MetaMathQA adapters from the primary comparisons, and evaluate independently trained Qwen2.5-7B source, direct rank-2, and Freeze adapters over seeds 0–2. Qwen source and direct training each use two epochs; Freeze uses two additional recovery epochs from the source, with the same rank, α, sequence-length, and learning-rate rules as the math protocol. Thus Qwen's direct control and Freeze have different production budgets. The unadapted Qwen base provides a separate reference. These adapters are not trained on the additional evaluation tasks.

We use zero-shot MathQA, ARC-Challenge (Clark et al., 2018), HellaSwag (Zellers et al., 2019), PIQA (Bisk et al., 2020), and WinoGrande (Sakaguchi et al., 2021). The fixed `lm-evaluation-harness` 0.4.11 protocol uses normalized accuracy for the first four tasks and accuracy for WinoGrande. All task items are evaluated. The source, compact fp16, post-hoc, and Freeze rows in Table 21 share task configurations within each model. We keep task metrics separate rather than constructing a heterogeneous overall average.

Table 21: Additional multiple-choice evaluation of the MetaMathQA adapters. Scores are percentages, mean ± sample standard deviation over seeds 0–2. The unadapted Qwen base is a single deterministic reference. Each completed model/arm is shown, including compact baselines.

Method	MiB	MathQA	ARC-C	HellaSwag	PIQA	WinoGrande
LLaMA-2-7B						
Source fp16 $r = 16$	76.250	25.01±0.72	35.78±0.74	69.42±0.14	73.74±0.17	63.61±0.75
Direct fp16 $r = 2$	9.531	24.99±0.89	35.47±0.79	71.36±0.19	72.74±0.38	66.01±1.28
LoRAQuant	7.151	26.18±0.21	37.97±0.73	73.59±0.37	75.99±0.84	68.09±0.46
PTQ $R = 8$	7.152	26.62±0.43	39.11±0.90	75.69±0.11	77.68±0.65	70.61±0.68
PTQ $R = 16$	9.538	26.05±0.15	38.82±0.39	74.87±0.34	76.90±0.38	68.17±0.89
QAT Freeze $R = 8$	7.152	25.57±0.40	35.69±0.90	70.50±0.14	73.32±0.84	64.54±1.87
QAT Freeze $R = 16$	9.538	24.97±0.33	36.12±1.63	69.47±0.14	73.65±0.95	64.11±0.64
LLaMA-2-13B						
Source fp16 $r = 16$	119.375	25.66±0.39	37.88±0.60	72.86±0.43	75.28±0.88	66.80±0.75
Direct fp16 $r = 2$	14.922	26.58±0.25	38.48±1.24	76.14±0.22	75.99±0.96	69.22±1.11
SVD $r = 2$ + recovery	14.922	26.41±0.56	39.79±1.02	75.88±0.12	75.19±0.39	66.69±1.45
QAT Freeze $R = 16$	14.930	26.30±0.41	38.05±1.03	73.05±0.24	74.65±0.42	65.48±2.37
Qwen2.5-7B						
Base (no adapter)	0.000	43.58	51.19	78.84	79.87	73.09
Source fp16 $r = 16$	77.000	34.77±0.73	45.53±0.52	78.10±0.06	79.07±0.40	68.59±0.34
Direct fp16 $r = 2$	9.625	36.05±0.67	45.39±0.70	78.62±0.22	79.45±0.59	70.51±1.19
QAT Freeze $R = 16$	9.631	34.81±0.36	45.85±0.87	77.87±0.22	78.82±0.31	67.98±0.48

All five tasks use zero-shot scoring; MathQA, ARC-C, HellaSwag, and PIQA use normalized accuracy, and WinoGrande uses accuracy. Sizes count packed signs and fp16 scales, or fp16 factors; LoRAQuant uses its packed-code-plus-quantizer-metadata formula. The math LoRAQuant size is 7.151 ± 0.015 MiB across seeds. These are descriptive comparisons of retained source behavior, with production schedules as specified in the text.

At carrier rank $R = 16$, the maximum absolute difference between Freeze and its rank-16 source means across these five tasks is 0.50 points on LLaMA-2-7B, 1.32 on LLaMA-2-13B, and 0.61 on Qwen2.5-7B. Their logical payloads shrink from 76.250 to 9.538, 119.375 to 14.930, and 77.000 to 9.631 MiB, respectively. These descriptive differences support preservation of the evaluated source behavior at approximately one eighth of its payload. The compact baselines in the table provide context for this source-retention comparison; preservation of a task-trained source is not a claim that every compact baseline or the unadapted base is outperformed.

Accounting and evaluation coverage. All payloads follow the logical deployment convention used in the main paper: fp16 dense factors, or packed signs plus fp16 scales; LoRAQuant additionally counts its quantizer parameters and group padding. Accuracy evaluation uses the saved checkpoints and their materialized side branches, with the scale-precision qualification stated in Section 5. These supplementary evaluations do not measure peak GPU memory or packed-kernel latency. MBPP returned zero for every arm and seed and is withheld from performance claims pending evaluator validation. ASDiv's answer-extraction behavior also requires validation, and the Qwen ASDiv/primary-math seed grid is incomplete. Consequently, those results are not used to extend the main accuracy claims. The retained evidence records their status and preserves all completed outputs.

M ADAPTER-PRODUCTION BUDGETS AND BASELINE SCOPE

Two distinct production questions. Producing an adapter starting from a base model and compressing an already-trained adapter are different workloads. For the current Freeze pipeline, the wall-clock accounting is

$$T_{\text{from base}} = T_{\text{source}} + T_{\text{PTQ}} + T_{\text{recovery}}, \tag{82}$$

$$T_{\text{from existing source}} = T_{\text{PTQ}} + T_{\text{recovery}}. \tag{83}$$

The same boundary must be used for both arms of a comparison. Source training cannot be omitted from only the LoRDBA arm when comparing against a directly trained compact adapter. Conversely, when both compression methods receive the same existing source, its shared cost can be reported separately. LoRAQuant has its own reconstruction-optimization and quantization costs, even though it receives no additional task-loss recovery in our comparison. Peak training memory is the maximum across sequential production stages, not the sum of their peaks. Distributed GPU-hours additionally require accounting for the devices assigned to each stage; queueing and evaluation time should be identified separately from training.

Table 22: Production stages behind the current math accuracy comparisons. Epochs specify dataset exposure, not equal elapsed time. No current row pair is presented as an end-to-end wall-clock-matched comparison.

Method	Initial training	Additional processing
Source fp16 $r = 16$	Two epochs	None
Direct fp16 $r = 2$	Four epochs (7B); two (13B/Mistral)	None; no rank-16 source is used
LoRAQuant (7B)	Two-epoch source	Reconstruction optimization and mixed-bit quantization; no CE recovery
Freeze	Two-epoch source	PTQ plus two epochs of scale-only CE recovery
SVD $r = 2$ (13B)	Same two-epoch source as Freeze	Truncated SVD plus two epochs of floating-factor CE recovery

What the controlled comparisons establish. The 7B four-epoch direct control tests whether the compact advantage survives giving direct LoRA the same total number of dataset passes; Freeze exceeds it by $6.24/2.24$ points on GSM8K/Minerva. The 13B same-source SVD control additionally fixes the source checkpoint and recovery schedule; Freeze exceeds it by $3.61/3.34$ points at nearly equal logical payload. These controls address specific budget and initialization confounds, without holding elapsed time or optimization geometry fixed. The 7B LoRAQuant comparison fixes source and logical size, but Freeze's additional recovery remains part of the pipeline difference. Thus the results support measured accuracy–storage trade-offs for specified production recipes, not an accuracy-per-GPU-hour ranking.

Training-memory evidence. Table 4 separates carrier packing from adapter-local recomputation. The latter reduces the measured Freeze-loop peak to 36.365 GiB versus 36.638 GiB for LoRA, while its step time is 680.9 versus 477.6 ms. These short controlled measurements identify a memory–time trade-off; they neither give the maximum over all production stages nor predict the full-run timing of the main accuracy artifacts. Historical B200 totals of 718–744 minutes for Full versus 385 for source LoRA (Table 12) remain visible as evidence of production overhead under that older recipe, not a timing estimate for current Freeze.

Alternatives beyond standard LoRA. DoRA (Liu et al., 2024b) changes magnitude–direction adaptation, PiSSA (Meng et al., 2024) initializes trainable factors from principal components of the base weight, and LoRA+ (Hayou et al., 2024) changes their learning-rate ratio. These are relevant alternatives when adapter-training effectiveness is the question. Our SVD control instead truncates a task-trained update and therefore is not a reproduction of PiSSA. VeRA (Kopiczko et al., 2024) shares frozen random factors and learns small scale vectors; sharing or regenerating those factors gives a different storage accounting from task-specific packed signs. Its small trainable state makes it a relevant comparator for Freeze rather than a method ruled out by the unmerged setting. We

provide no controlled cost or accuracy measurements against these alternatives, so the comparisons in this paper do not resolve their relative performance. QLoRA, LoftQ, and QA-LoRA address base quantization or its interaction with adaptation (Dettmers et al., 2023; Li et al., 2024; Xu et al., 2024a); their training-memory claims cannot be transferred to our bf16-base measurements.

When the trade-off is relevant. The intended use is to retain and distribute multiple task-specific adapters over a common frozen base after off-device production. Adapter storage and transfer requirements then remain relevant even when extra training is acceptable. This is a deployment motivation, not evidence that repeated use automatically offsets production time, energy, or cost. The experiments directly establish adapter sizes, sampled accuracy trade-offs, and a packed-residency reduction; they do not measure a multi-tenant serving workload or its break-even point. The additional HumanEval and multiple-choice results in Appendix L broaden the observed task coverage without making those unmeasured systems claims.

N Additional Figures

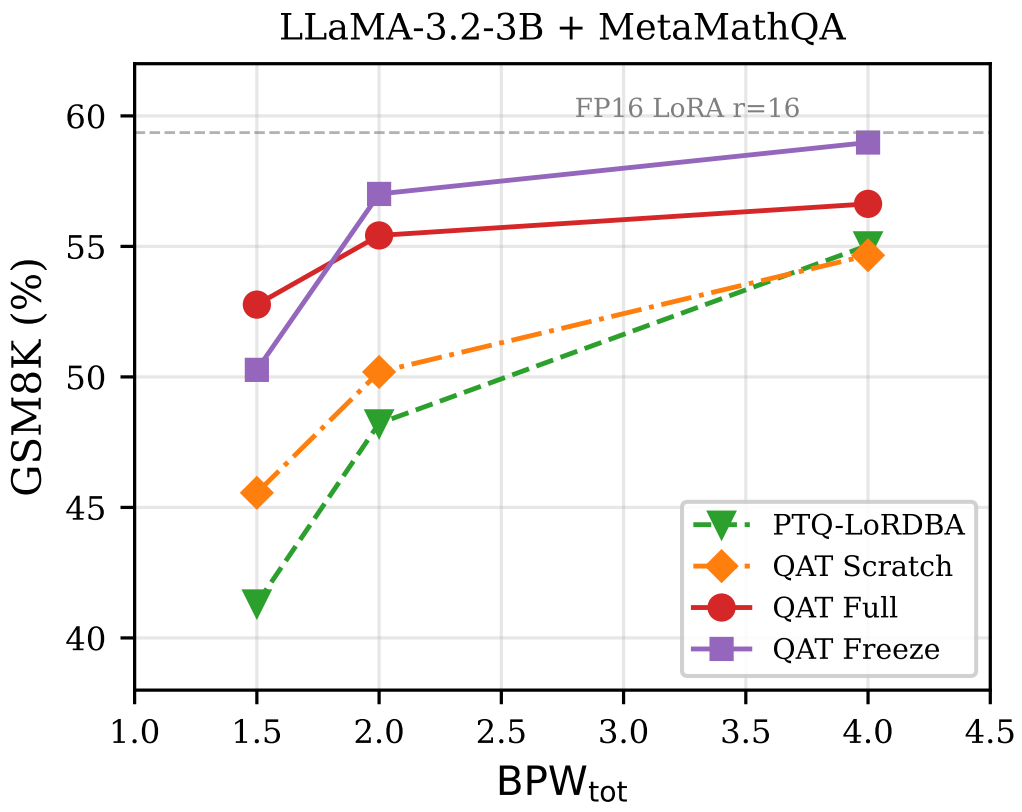


Figure 2: Historical BPW sweep on LLaMA-3.2-3B+MetaMathQA (GSM8K, source $r_0 = 16$), retained in its original graphical form. The axis uses legacy nominal BPW labels. These unmatched mode curves do not establish a theoretical saturation point or a current Full–Freeze ranking.