# BLOCKFAMQ: REALIZED-RESIDUAL CURVATURE FOR COMPLETE-CHECKPOINT MAMBA QUANTIZATION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Mixed-precision quantization must decide which realized rounding residuals receive scarce high-precision bytes, but diagonal sensitivity scores discard the signed coordinate interactions that distinguish deterministic candidates. We introduce BlockFAMQ, a matrix-free allocator that scores squared per-token gradient–residual products and solves the resulting integer-byte multiple-choice problem under a complete-checkpoint budget. Across Mamba-1 130M, 370M, 1.4B, and 2.8B, the mean BlockFAMQ-minus-diagonal WikiText-2 NLL gap contracts from $-0.218$ at 2.5 effective bits to $-0.026$ at 4.0 bits and $-0.009$ at 4.5 bits. At 3.5 bits, BlockFAMQ improves six-task accuracy by $2.08$ points and recovers $65.7$–$80.1\%$ of the diagonal quantization gap; an activation-weighted reconstruction baseline narrows but does not remove the advantage. At 3.0 bits, direct candidate KL improves on BlockFAMQ by $0.038$ nats across 1.4B and 2.8B, while a feasibility-anchored block-to-KL shortlist recovers $0.033$ nats at $1.62\times$ the block-scoring cost on average. Precision shifts toward `x_proj` and `in_proj`, whose realized residuals have the largest cross-term tails, and the block effect is larger on Mamba-1 1.4B than on matched-scale Pythia-1.4B. The direction persists on Mamba-2 2.7B and Falcon-Mamba 7B; packed W4A8 composition retains $96.8$–$97.7\%$ of Quamba2 decode throughput while lowering NLL. These results support realized block curvature as a budget-dependent allocation correction, with direct KL becoming more useful when aggressive quantization leaves the local quadratic regime.

## 1 INTRODUCTION

A quantization residual has a direction as well as a magnitude. Its coordinates can reinforce a sensitive model direction or cancel along it. Diagonal curvature scores retain the squared error of each coordinate but discard the products that distinguish these cases. Deterministic rounding provides no reason for those products to average away. The resulting error is operational: two candidates with nearly identical diagonal cost can receive opposite full quadratic rankings, changing where a mixed-precision allocator spends its budget. This distinction is especially relevant for selective state-space models. In Mamba, a projection can jointly affect input-dependent transition parameters, state injections, and downstream readouts (Gu & Dao, 2024). An error in that projection can persist through the selective recurrence, and its signed coordinate contributions need not act independently. End-to-end differentiation already propagates these effects. The allocation question is whether the residual is priced along that realized direction before its coordinates are squared independently. BlockFAMQ keeps candidate construction fixed and changes only this pricing step. Each allocation unit receives a bank of reconstructed tensors with exact serialized costs. For candidate residual $e_u(b)$, the method accumulates squared token-level inner products $\left(g_u^\top e_u(b)\right)^2$, retaining the within-unit empirical-Fisher interactions without materializing a curvature matrix. A multiple-choice knapsack solver then selects one candidate per unit under the complete-checkpoint byte ceiling. This separation isolates allocation quality from range fitting, reconstruction, and packing choices. The central empirical result is a budget-response curve rather than a single operating point. Across 2.5, 3.0, 3.5, 4.0, and 4.5 effective bits, the block advantage is largest where the candidate residuals are largest and contracts as the checkpoint approaches high precision. The 3.0-bit regime also exposes the approximation boundary: BlockFAMQ remains better than diagonal and activation-weighted reconstruction allocation, but direct KL closes more of the remaining gap, indicating that higher-order



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.

candidate response becomes material. This pattern explains why a modest four-bit effect can coexist with a much larger aggressive-budget effect.

The paper makes four contributions: We define a realized-residual block empirical-Fisher score and an exact serialized-byte allocation problem that preserve within-unit cross-coordinate structure while keeping the candidate bank fixed. We evaluate the allocator across five complete-checkpoint budgets, add activation-weighted reconstruction and model-Fisher baselines, and report paired uncertainty for the principal NLL and downstream contrasts. We connect the allocation behavior to Mamba by measuring tensor-class precision shifts, calibration-sample stability, granularity, context length, Mamba-1 versus Pythia at matched scale and pretraining family, and transfer to Mamba-2 and Falcon-Mamba. We separate allocation quality from deployment by comparing native matched-effective-bit baselines, direct candidate KL and a block-to-KL hybrid, and packed W4A16 and W4A8 operating points.

## 2 Related Work

Curvature-aware compression includes Hessian-based precision selection in HAWQ and HAWQ-V2, rounding compensation in GPTQ and Optimal Brain Compression, and block reconstruction in BRECQ (Dong et al., 2019; 2020; Frantar et al., 2023; Frantar & Alistarh, 2022; Li et al., 2021). LQ-LoRA combines Fisher weighting, quantization, and integer allocation (Guo et al., 2024). These methods separate two useful roles for curvature: constructing a better reconstruction and selecting among available reconstructions. Our comparison concerns the second role and freezes the first. FGMP uses Fisher-weighted perturbation energies for mixed precision, while FAMPWQ combines a quantization-sensitive Fisher criterion with a learned allocation policy (Hooper et al., 2025; Lee et al., 2026). These methods establish perturbation-aware Fisher sensitivity as prior art. BlockFAMQ differs by retaining the cross-coordinate products of each fixed residual rather than replacing that residual with independent coordinate energies. SHARP-Q uses spectral alignment and rectification (Lv et al., 2026); applying a score inspired by that method to a frozen bank is an allocation-stage adaptation, not a reproduction of its native reconstruction pipeline.

Mamba and Mamba-2 introduce selective state-space computation and structured state-space duality (Gu & Dao, 2024; Dao & Gu, 2024). Quamba, MambaQuant, Q-Mamba, and Quamba2 address weight, activation, state, and execution constraints specific to these models (Chiang et al., 2025b; Yue et al., 2025; Tianqi et al., 2025; Chiang et al., 2025a). Those transformations can change the candidate bank itself. We therefore separate a common-bank allocation comparison from a packed-backend composition. Neither a better allocation score nor a smaller serialized checkpoint implies a faster native kernel.

## 3 Mamba candidates and complete-checkpoint budgets

**A coordinate partition shared across the budget sweep** Let $\theta$ denote the reference checkpoint. Partition its allocatable coordinates into disjoint units $\mathcal{I}_u$, $u = 1, \dots, U$. A candidate $b \in \mathcal{B}_u$ defines reconstruction $Q_{u,b}(\theta_u)$ and residual

$$e_u(b) = Q_{u,b}(\theta_u) - \theta_u. \tag{1}$$

The allocation unit receives one precision choice, the scaling group shares quantizer metadata, and the curvature block determines which coordinate interactions are retained. In BlockFAMQ, the allocation unit and curvature block coincide; the scaling group can be smaller. For the Mamba-1 evaluation, each of the seven tensor types in Table 13 is divided into consecutive output-row intervals of at most $R = 1{,}024$ rows. An interval contains every trailing coordinate of its rows and never crosses a tensor or layer boundary. The same rule applies to all five budget points and every common-bank allocator. It gives 312, 672, 1,296, and 2,176 units for the four architectures, respectively. Per-row scales apply to matrices and convolutions; a vector tile uses one scale for the entire tile. Biases and normalization weights stay outside the allocator at FP32. The tied embedding and output head have one common stored reconstruction, counted once.

Candidate construction precedes scoring. Each tensor is reconstructed at $b \in \{2, 3, 4, 5, 6, 8\}$ with the same range estimator and full-precision activation cache. The row tiles are then extracted from those reconstructions. No candidate is refitted after allocation. This convention keeps the candidates

independent of the objective and prevents a later reconstruction pass from absorbing an allocation error. Candidate-dependent metadata belongs to its unit; shared metadata is charged once as a constant.

**Every stored parameter consumes the cap** Write $m_u(b)$ for the integer bytes of a candidate, including payload, scales, zero points, tags, and padding. All fixed tensors and container bytes enter $M_{\text{fixed}}$. With $N_{\text{param}}$ unique model parameters,

$$M(b) = M_{\text{fixed}} + \sum_u m_u(b_u) \le M_{\text{target}}, \qquad B_{\text{eff}}(b) = \frac{8M(b)}{N_{\text{param}}}. \tag{2}$$

The cap for target $B$ is $\lfloor BN_{\text{param}}/8 \rfloor$. Its denominator includes the embedding and every fixed parameter, not only quantized matrices. Before optimization, the candidate bank has minimum feasible size

$$b_{\min} = \frac{8}{N_{\text{param}}} \left( M_{\text{fixed}} + \sum_u \min_{b \in \mathcal{B}_u} m_u(b) \right). \tag{3}$$

For embedding fraction $f_E$ stored at $b_E$ bits and a minimum of two payload bits elsewhere, a necessary lower bound is

$$b_{\min} \ge 2 + (b_E - 2) f_E. \tag{4}$$

For the standard 130M architecture, an eight-bit embedding already places this payload bound at $3.794$ bits, and the corresponding bound for 370M is $2.832$ bits. We therefore specify the tied embedding/head precision at every sweep point before allocation. At $2.5$ bits all four checkpoints use the two-bit tied embedding/head candidate. At $3.0$ bits, 130M and 370M use four bits while $1.4$B and $2.8$B use eight bits. At $3.5$ bits, only 130M uses four bits; the other three use eight bits. At $4.0$ and $4.5$ bits all four checkpoints use eight bits. Every allocator compared at one model–budget pair receives the identical fixed embedding reconstruction. Thus each within-point allocator contrast is cap matched, while the sweep itself is a sequence of feasible complete-checkpoint operating points rather than an embedding-invariant intervention. Appendix Table 20 crosses the 130M embedding precision and budget, and Appendix D derives the architecture counts and byte ceilings.

Fixed tensors can also contribute a shared nonzero residual $e_0$. The deployed residual is $e(b) = e_0 + \sum_u \iota_u e_u(b_u)$, where $\iota_u$ inserts a tile into checkpoint coordinates. Constant storage does not make the fixed residual's interaction with the chosen candidates constant in the predictive loss.

## 4 BLOCKFAMQ

Figure 2 summarizes the end-to-end bank–score–knapsack–serialization pipeline; implementation details appear in Appendix C

### 4.1 PRICE THE REALIZED DIRECTION BEFORE SQUARING

At scored next-token position $(k,t) \in \mathcal{I}$, let $g^{(k,t)} = \nabla_\theta \log p_\theta(y_{k,t} \mid x_{k,<t})$ and let $g_u^{(k,t)}$ restrict the gradient to $\mathcal{I}_u$. For $N_{\text{tok}} = |\mathcal{I}|$, define the block cost and its diagonal ablation as

$$\widehat{C}_u^{\text{raw}}(b) = \frac{1}{2N_{\text{tok}}} \sum_{(k,t)\in\mathcal{I}} (g_u^{(k,t)\top} e_u(b))^2, \tag{5}$$

$$\widehat{C}_u^{\text{diag}}(b) = \frac{1}{2N_{\text{tok}}} \sum_{(k,t)\in\mathcal{I}} \sum_{i\in\mathcal{I}_u} (g_i^{(k,t)})^2 e_i(b)^2. \tag{6}$$

Both use the same residual. The first retains the within-unit cross-coordinate products; the second drops them. A squared sequence-mean gradient would instead introduce cross-token products. We average squared token-level products, so sequence length does not silently change the statistic being compared.

The raw score is the primary objective. An optional robust variant replaces each squared product by its minimum with $B_{u,b}^2$, where the threshold is fixed on an independent tuning split:

$$\widehat{C}_u^{(B)}(b) = \frac{1}{2N_{\text{tok}}} \sum_{(k,t)\in\mathcal{I}} \min\{(g_u^{(k,t)\top} e_u(b))^2, B_{u,b}^2\}. \tag{7}$$

This clipped objective is not the quadratic of one candidate-independent Fisher matrix. Proposition 7 gives its sequence-level concentration bound for a frozen candidate bank. It is kept separate from the raw block–diagonal comparison. Streaming the directional products avoids storing dense curvature blocks; forward and per-token differentiation costs remain part of scoring time.

### 4.2 Directional sensitivity through the selective recurrence

For one selective state coordinate, write $h_t = a_t h_{t-1} + x_t$, $a_t = \exp(\Delta_t A)$, and $A = -\exp(q)$. A weight residual induces directional changes $\dot{q}$, $\dot{\Delta}_t$, and $\dot{x}_t$. Differentiating the recurrence gives

$$\dot{h}_t = a_t \dot{h}_{t-1} + a_t A(\dot{\Delta}_t + \Delta_t \dot{q}) h_{t-1} + \dot{x}_t. \tag{8}$$

The likelihood gradient propagates this perturbation through subsequent states and output projections. Contributions from coordinates in one candidate are therefore combined in $g_u^\top e_u$ before squaring. This retains reinforcement and cancellation within that candidate. It does not introduce a separate trajectory penalty, and it does not assume that every long-lived state is important to a downstream task.

### 4.3 Activation-weighted reconstruction baseline

A forward-only allocation baseline scores each candidate by the reconstruction error it induces on cached full-precision inputs. For a linear or convolutional unit with weight residual $\Delta W_u(b)$ and calibration input matrix $X_u$, define

$$\widehat{C}_u^{\mathrm{AWRE}}(b) = \frac{1}{N_u} \left\| \Delta W_u(b) X_u \right\|_F^2 . \tag{9}$$

For vector-valued parameters, the same definition uses the aligned activation or recurrent-state vector. The quadratic form $\Delta W_u X_u X_u^\top \Delta W_u^\top$ retains input-covariance cross-coordinate structure without likelihood gradients. AWRE receives the same candidate bank, partition, serializer, and exact integer-byte solver as BlockFAMQ.

### 4.4 Allocate under integer serialized costs

Let $\widehat{Q}_{\mathrm{B}}(b) = \sum_u \widehat{C}_u(b_u)$. For binary candidate indicators $x_{u,b}$, the optimization is

$$\begin{aligned} \min_{x} \quad & \sum_u \sum_{b \in \mathcal{B}_u} \widehat{C}_u(b) x_{u,b}, \\ \text{s.t.} \quad & \sum_b x_{u,b} = 1 \quad \text{for every } u, \\ & M_{\mathrm{fixed}} + \sum_u \sum_b m_u(b) x_{u,b} \le M_{\mathrm{target}}, \qquad x_{u,b} \in \{0,1\}. \end{aligned} \tag{10}$$

A Pareto-frontier dynamic program retains nondominated cumulative size–cost pairs; a mixed-integer solver is an alternative when that frontier grows too large (Lemma 8). Within-unit dominance pruning preserves an optimum. Exactness concerns the encoded objective and byte constraint, not the predictive loss. Greedy promotion is included as an ablation. The final evaluation applies the reconstructions obtained after serialization and reload, not an idealized precision label.

When no uniform format fills a cap, the size-only comparator minimizes a parameter-count distortion proxy and breaks ties using a fixed hash of the unit identifier. It uses the same integer program but no gradients or held-out quality. Appendix C.2 specifies this objective and the hash order. Interpolating the losses of two uniform checkpoints would not produce a feasible mixed-precision assignment and is not used as a comparator.

### 4.5 Direct candidate KL as the nonlinear alternative

A forward-only alternative changes one unit and evaluates its full predictive response:

$$\widehat{C}_u^{\mathrm{KL}}(b) = \frac{1}{N_{\mathrm{tok}}} \sum_{(k,t) \in \mathcal{I}} \mathrm{KL}\big( p_\theta(\cdot \mid x_{k,<t}) \,\big\|\, p_{\theta + \iota_u e_u(b)}(\cdot \mid x_{k,<t}) \big) . \tag{11}$$

### 4.6 FEASIBILITY-ANCHORED BLOCK-TO-KL REFINEMENT

For each model, byte cap, and calibration draw, we first solve the complete BlockFAMQ problem and obtain the feasible assignment $\widehat{\mathbf{b}}^{\,\mathrm{blk}}$. Let $T_u^{(2)}$ contain the two candidates in unit $u$ with the lowest block empirical-Fisher costs, with ties resolved first by smaller serialized size and then by precision index. The hybrid candidate set is

$$\mathcal{S}_u = T_u^{(2)} \cup \{\widehat{b}_u^{\,\mathrm{blk}}\}. \tag{12}$$

After duplicate removal, each unit therefore contains two or three candidates. We evaluate direct candidate KL only on $\mathcal{S}_u$ and solve the resulting restricted multiple-choice knapsack problem using the original integer byte costs. The shortlist is constructed separately for every cap and calibration draw because its feasibility anchor is the corresponding BlockFAMQ assignment.

**Proposition 1** (Shortlist feasibility)**.** *If $\widehat{\mathbf{b}}^{\,\mathrm{blk}}$ satisfies the complete-checkpoint cap, then the restricted hybrid problem defined by Equation* (12) *is feasible under the same cap.*

*Proof.* Because $\widehat{b}_u^{\,\mathrm{blk}} \in \mathcal{S}_u$ for every unit,

$$M_{\min}^{\mathcal{S}} = M_{\mathrm{fixed}} + \sum_u \min_{b \in \mathcal{S}_u} m_u(b) \tag{13}$$

$$\leq M_{\mathrm{fixed}} + \sum_u m_u(\widehat{b}_u^{\,\mathrm{blk}}) = M(\widehat{\mathbf{b}}^{\,\mathrm{blk}}) \leq M_{\mathrm{target}}. \tag{14}$$

Thus the complete BlockFAMQ assignment remains available inside the restricted bank and supplies a feasible witness. □

Teacher probabilities and unaffected upstream activations can be cached. The changed module and every affected downstream state must be recomputed. Freezing a changed recurrent trajectory would yield a different reconstruction objective. This baseline shares the candidate bank, positions, and byte solver with BlockFAMQ. It retains nonlinear single-candidate effects, but summing its costs still omits interactions among jointly quantized units. It also uses the teacher predictive distribution rather than observed-token gradients, so the comparison is between practical objectives, not a pure isolation of Taylor error.

## 5 WHAT THE CORRECTION CAPTURES

For a positive-semidefinite block $F$ and positive diagonal cost, define

$$r_e = \frac{e^\top F e}{e^\top \operatorname{diag}(F) e}, \qquad \chi_e = r_e - 1, \qquad \rho_e = |\chi_e|. \tag{15}$$

Positive $\chi_e$ means a diagonal score underprices that residual; negative $\chi_e$ means it overprices it. The magnitude $\rho_e$ measures omission but cannot determine which candidate should receive more precision.

**Proposition 2** (Residual direction and centering)**.** *For a $d$-coordinate positive-semidefinite block, $0 \leq r_e \leq d$; both endpoints are attainable when $d \geq 2$. For independent centered residual coordinates with finite second moments,*

$$\mathbb{E}[e^\top F e] = \mathbb{E}[e^\top \operatorname{diag}(F) e]. \tag{16}$$

*For independent identically distributed replicates with positive expected diagonal cost, the ratio of summed quadratics converges to one.*

Positive semidefiniteness bounds $|F_{ij}|$ by $\sqrt{F_{ii}F_{jj}}$; Cauchy–Schwarz then gives the upper bound. Centering removes expected cross products. The result concerns a ratio of sums, not $\mathbb{E}[r_e]$ or $\mathbb{E}[|r_e - 1|]$. Subtractive dither therefore provides a consistency control; because it changes the residual distribution, it is not by itself a causal mediation test. The proof and Proposition 3 give an explicit allocation reversal in Appendix B.1. A diagonal-only indistinguishability result in Appendix B.2 further shows that perfect diagonal estimates cannot generally replace the discarded information.

Table 1: Aggregate budget-response summary. NLL differences are block minus comparator; negative is better. The perplexity column is $100(1 - \exp(\Delta\mathrm{NLL}))$ for block versus diagonal.

| Bits | Block–diagonal NLL [95% CI] | PPL reduction | Block–AWRE NLL [95% CI] | six-task $\Delta$ pp |
|---|---|---|---|---|
| 2.5 | $-0.218$ $[-0.246, -0.190]$ | 19.6% | $-0.083$ $[-0.105, -0.061]$ | $+3.41$ $[+2.71, +4.18]$ |
| 3.0 | $-0.180$ $[-0.206, -0.155]$ | 16.5% | $-0.070$ $[-0.089, -0.052]$ | $+2.87$ $[+2.22, +3.56]$ |
| 3.5 | $-0.146$ $[-0.162, -0.130]$ | 13.6% | $-0.062$ $[-0.075, -0.049]$ | $+2.08$ $[+1.55, +2.73]$ |
| 4.0 | $-0.026$ $[-0.032, -0.019]$ | 2.6% | $-0.012$ $[-0.018, -0.006]$ | $+0.38$ $[+0.18, +0.58]$ |
| 4.5 | $-0.009$ $[-0.014, -0.004]$ | 0.9% | $-0.005$ $[-0.009, -0.001]$ | $+0.14$ $[+0.02, +0.27]$ |

The predictive interpretation has a separate boundary. Let $D(e) = \mathbb{E}_x \,\mathrm{KL}(p_\theta(\cdot \mid x) \| p_{\theta+e}(\cdot \mid x))$ and let $F$ now be model Fisher. Under local regularity,

$$D(e) = \underbrace{\frac{1}{2}\sum_{u=0}^{U} e_u^\top F_{uu} e_u}_{Q_{\mathrm{B}}(e)} + \underbrace{\sum_{0 \le u < v \le U} e_u^\top F_{uv} e_v}_{I(e)} + R_3(e). \tag{17}$$

Unit zero contains the fixed residual. The block score addresses within-unit pricing, not $I(e)$ or the nonlinear remainder. Observed-token Fisher need not equal model Fisher (Kunstner et al., 2019; Martens, 2020). Theorem 6 gives a conditional allocation comparison that retains block and calibration errors, and Appendix B supplies the Taylor remainder. These standard bounds describe scope rather than establish a global compression certificate.

## 6 Mamba evaluation

**Protocol and estimands** The primary study evaluates Mamba-1 130M, 370M, 1.4B, and 2.8B with five disjoint C4 calibration draws. Each draw fixes the candidate bank before scoring and uses $4{,}096$ scored next-token gradients for the primary allocation. On Mamba-1 1.4B, candidate construction and scoring require 272 and $1{,}108$ GPU-minutes per draw, respectively, totaling $115.0$ A100-GPU-hours across five draws. Candidate reconstructions and score tables are reused across the five budget targets $\{2.5, 3.0, 3.5, 4.0, 4.5\}$; only allocation and serialization repeat. Appendix C.5 provides the timing breakdown, and Table 10 reports the calibration-size study. WikiText-2 and C4 report mean token NLL; perplexity is $\exp(\mathrm{NLL})$. ARC-Easy, ARC-Challenge, HellaSwag, PIQA, WinoGrande, and LAMBADA form the downstream suite (Clark et al., 2018; Zellers et al., 2019; Bisk et al., 2020; Sakaguchi et al., 2020; Paperno et al., 2016; Biderman et al., 2024). Matched allocators share the same row partition, frozen reconstructions, fixed tensors, serializer, and integer-byte cap. The primary bank uses GPTQ candidates; RTN provides a separate candidate-generator check. Paired uncertainty resamples calibration draws and source-document clusters for language modeling and task items within each fixed benchmark for downstream accuracy. Each contrast uses $10{,}000$ bootstrap replicates with independently computed percentile endpoints. The individual bootstrap replicate arrays and unrounded point estimates are not included in the submitted PDF. Appendix C.4 gives the estimator and endpoint rule.

**Budget response is the central result** Figure 1 is the primary empirical summary. The block advantage is largest at the aggressive end of the feasible range and contracts smoothly as the checkpoint approaches higher precision. At 2.5 bits the equal-model mean block–diagonal NLL difference is -0.218 nats $[-0.246, -0.190]$; it is -0.180 $[-0.206, -0.155]$ at 3.0 bits, -0.146 $[-0.162, -0.130]$ at 3.5 bits, -0.026 $[-0.032, -0.019]$ at 4.0 bits, and -0.009 $[-0.014, -0.004]$ at 4.5 bits. These differences correspond to relative perplexity reductions of 19.6%, 16.5%, 13.6%, 2.6%, and 0.9% against diagonal allocation. Table 1 gives the paired aggregate intervals and downstream effects.

The 3.0-bit operating point is the clearest boundary of the quadratic approximation. BlockFAMQ remains better than diagonal and AWRE allocation, but direct candidate KL improves a further 0.038 nats on average, compared with 0.003 nats at 4.0 bits. The block-to-KL shortlist recovers 0.033 nats of the 3.0-bit improvement at substantially lower scoring cost than exhaustive candidate KL. The widening direct-KL gap is consistent with higher-order candidate response becoming material

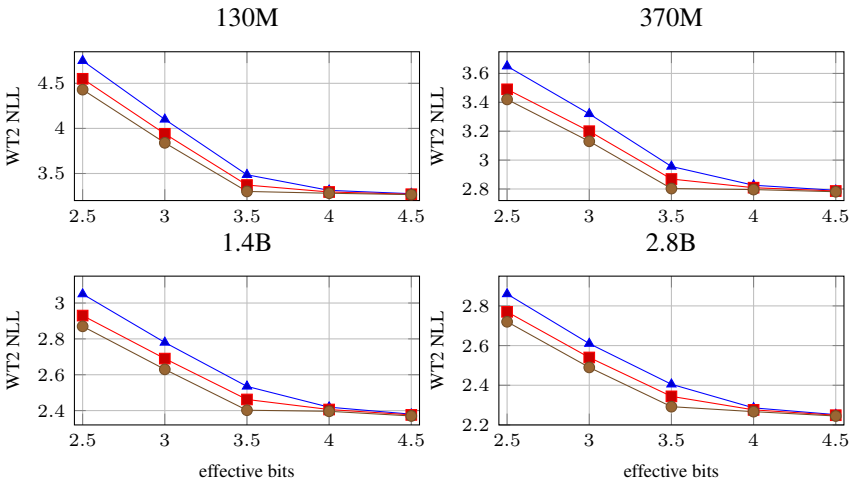



Figure 1: Complete-checkpoint budget response on four Mamba-1 sizes. Triangles denote diagonal allocation, squares AWRE, and circles BlockFAMQ. Within each model–budget point, all allocators share the candidate grid, row partition, and fixed embedding/head reconstruction. The embedding schedule is listed in Table 6. Activation-weighted reconstruction error (AWRE) uses Equation (9); BlockFAMQ retains the realized gradient–residual interactions. The separation is largest at aggressive budgets and contracts above four bits.

Table 2: Mamba-1 quality at the 3.5-bit complete-checkpoint target. Panel A reports model-level WikiText-2 and six-task contrasts; PPL is $\exp(\mathrm{NLL})$. Panel B reports task-level BlockFAMQ minus diagonal accuracy differences. All intervals are paired 95% intervals.

Panel A: model-level quality

| Model | FP32 NLL | Diag NLL/PPL | Block NLL/PPL | ΔNLL [CI] | Diag acc. | Block–diag acc. [CI] |
|---|---|---|---|---|---|---|
| 130M | 3.222 | 3.486/32.66 | 3.301/27.14 | −0.185 [−0.222, −0.149] | 37.20 | +2.55 [+1.64, +3.61] |
| 370M | 2.765 | 2.956/19.22 | 2.803/16.49 | −0.153 [−0.180, −0.126] | 44.22 | +2.89 [+1.90, +4.00] |
| 1.4B | 2.357 | 2.535/12.62 | 2.402/11.05 | −0.133 [−0.155, −0.111] | 53.43 | +2.10 [+1.37, +2.96] |
| 2.8B | 2.233 | 2.405/11.08 | 2.292/9.89 | −0.113 [−0.132, −0.094] | 57.95 | +0.78 [+0.20, +1.47] |

Panel B: task-level BlockFAMQ minus diagonal accuracy (percentage points)

| ARC-E | ARC-C | HellaSwag | PIQA | WinoGrande | LAMBADA | Mean |
|---|---|---|---|---|---|---|
| +2.53 [+1.75, +3.34] | +1.70 [+0.82, +2.56] | +2.09 [+1.52, +2.68] | +1.55 [+0.78, +2.33] | +1.10 [+0.29, +1.89] | +3.51 [+2.36, +4.67] | +2.08 [+1.55, +2.73] |

as residuals grow. The budget sweep therefore supports a regime statement rather than monotone superiority of one local approximation.

**Aggressive-budget downstream quality and FP32 gap** At 3.5 bits, the block–diagonal NLL differences are -0.185, -0.153, -0.133, and -0.113 nats on 130M, 370M, 1.4B, and 2.8B, respectively (Table 2). Relative to the matched FP32 losses, these recover 70.1%, 80.1%, 74.7%, and 65.7% of the diagonal quantization gap. The downstream mean improves by 2.08 percentage points over diagonal allocation and 3.70 points over size-only. Appendix Tables 17 and 18 report the model- and task-level components.

For comparator loss $L_q > L_{32}$ and block loss $L_{\mathrm{B}}$, we report the recovered fraction of the positive quantization gap as

$$G_q = \frac{L_q - L_{\mathrm{B}}}{L_q - L_{32}}. \tag{18}$$

**Four-bit uncertainty** At four bits, every Mamba-1 checkpoint still favors BlockFAMQ over diagonal allocation in NLL, but the absolute effect is much smaller than at 3.5 bits. The downstream contrast is +0.38 percentage points under both model-first and task-first aggregation. Per-model and task-level intervals appear in Appendix Tables 7 and 8; Table 21 includes the matched allocation baselines, and Table 22 separates the allocation contrast within RTN and GPTQ candidate banks.

Table 3: Mamba-specific allocation evidence. Panel A reports parameter-weighted selected payload precision at the 3.5-bit complete-checkpoint cap. Payload precision is distinct from effective checkpoint precision. The latter is computed from the complete serialized size, including the fixed embedding/head, FP32 biases and normalization parameters, FP32 scales, precision tags, padding, and container metadata. Within each model, tensor-class means weight precision by allocatable parameter count; the displayed values then average the four model-level statistics equally. Panel B reports paired BlockFAMQ-minus-diagonal NLL differences; negative values favor BlockFAMQ. Mamba's tied embedding/head is stored once, whereas Pythia's untied input embedding and output head are stored separately.

Panel A: Mamba-1 precision map at the 3.5-bit cap

| Tensor class | Diagonal bits | Block bits | Shift |
|---|---|---|---|
| `in_proj.weight` | 3.06 | 3.11 | +0.05 |
| `out_proj.weight` | 3.21 | 3.08 | −0.13 |
| `x_proj.weight` | 3.35 | 3.70 | +0.35 |
| `dt_proj.weight` | 3.28 | 3.36 | +0.08 |
| `conv1d.weight` | 3.24 | 3.15 | −0.09 |
| $A_{\log}$ | 3.65 | 3.56 | −0.09 |
| $D$ | 3.42 | 3.32 | −0.10 |
| Allocatable payload mean | 3.05 | 3.05 | 0.00 |
| Serialized effective bits | 3.499 | 3.499 | 0.000 |

Panel B: matched-scale Mamba–Pythia comparison

| Model | 3.5-bit cap [95% CI] | 4.0-bit cap [95% CI] |
|---|---|---|
| Mamba-1 1.4B | $-0.133$ $[-0.155, -0.111]$ | $-0.023$ $[-0.032, -0.014]$ |
| Pythia-1.4B | $-0.028$ $[-0.040, -0.017]$ | $-0.009$ $[-0.017, -0.001]$ |

**Mamba-specific allocation behavior** Appendix Table 9 reports the residual-specific geometry; its heaviest upper tail occurs for `x_proj`, with a 95th-percentile omission magnitude of 19.70. Table 19 shows that subtractive dither contracts the ratio-of-sums from 1.92 to 1.04 and leaves a block–diagonal NLL interval crossing zero.

The precision maps in Table 3 show where the two curvature models spend the same storage budget. At 3.5 bits, BlockFAMQ makes its largest positive shift on `x_proj`, whose residual-specific omission ratio has the heaviest tail, and a smaller positive shift on `in_proj`. The compensating reduction occurs primarily on the much larger `out_proj` surface, with smaller reductions on the convolution, $A_{\log}$, and $D$. The global parameter-weighted allocatable mean is unchanged, so the map redistributes precision rather than increasing the effective budget. Within each model, tensor-class means weight selected precision by the number of allocatable parameters in that class; the displayed class means and the total row then average the four model-level statistics equally. A paired architecture comparison holds model scale and pretraining family approximately fixed. Mamba-1 1.4B and Pythia-1.4B are evaluated with the same target bits, candidate precision grid, score-token counts, and bootstrap construction. Their fixed-tensor accounting follows the checkpoint architecture rather than forcing a shared aliasing convention: Mamba-1 stores one tied input-embedding/output-head matrix, whereas Pythia-1.4B uses untied input embeddings and language-model output weights. The Pythia denominator and byte ledger therefore count those two matrices separately, and each target bit ratio is computed from Pythia's own complete serialized checkpoint. The block–diagonal gap is larger on Mamba at both caps: -0.133 versus -0.028 nats at 3.5 bits and -0.023 versus -0.009 at 4.0 bits. The paired difference of these gaps is -0.105 $[-0.130, -0.080]$ at 3.5 bits and -0.014 $[-0.026, -0.003]$ at 4.0 bits.

**Calibration size, model Fisher, and granularity** Calibration stability and granularity appear in Appendix Tables 10 and 11. The 1.4B allocation reaches 96.1% agreement with the 8k reference by 2,048 scored positions, where the score pass costs 563 GPU-minutes per draw; the primary 4,096-position setting reaches 98.1% agreement at 1,108 GPU-minutes. The $R = 1024$ row partition lies near the middle of the measured accuracy–cost tradeoff.A model-Fisher ablation samples $y \sim p_\theta(\cdot \mid x)$ at each scored position instead of using the observed next token. On 1.4B, the 3.5-bit block–diagonal effect changes from -0.133 to -0.126 nats with 91.4% allocation agreement; on 2.8B it changes from -0.113 to -0.108 with 93.1% agreement. At four bits the corresponding gaps are -0.023 versus -0.021

Table 4: Direct candidate KL and feasibility-anchored block-to-KL refinement. The hybrid shortlist contains the BlockFAMQ-selected candidate and the two lowest block-cost candidates in each unit, leaving two or three unique candidates after duplicate removal. Differences are method minus BlockFAMQ NLL; negative values favor the alternative. Costs include scoring every candidate retained by the corresponding method. The byte audit appears in Table 5.

| Model | Cap | Direct KL $\Delta$ [95% CI] | Hybrid $\Delta$ [95% CI] | KL/block | Hybrid/block |
|---|---|---|---|---|---|
| 1.4B | 3.0 | $-0.041\ [-0.058, -0.024]$ | $-0.035\ [-0.051, -0.019]$ | $2.51\times$ | $1.58\times$ |
| 2.8B | 3.0 | $-0.035\ [-0.050, -0.020]$ | $-0.031\ [-0.045, -0.017]$ | $3.05\times$ | $1.66\times$ |
| 1.4B | 3.5 | $-0.008\ [-0.020, +0.004]$ | $-0.007\ [-0.017, +0.003]$ | $2.44\times$ | $1.51\times$ |
| 2.8B | 3.5 | $-0.006\ [-0.017, +0.005]$ | $-0.006\ [-0.015, +0.003]$ | $2.90\times$ | $1.59\times$ |
| 1.4B | 4.0 | $-0.003\ [-0.010, +0.004]$ | $-0.002\ [-0.008, +0.003]$ | $2.41\times$ | $1.47\times$ |
| 2.8B | 4.0 | $-0.002\ [-0.008, +0.004]$ | $-0.001\ [-0.006, +0.004]$ | $2.84\times$ | $1.54\times$ |

and -0.019 versus -0.018. The observed-token estimator therefore changes some decisions but not the principal ordering in these runs.

**Direct KL and the block-to-KL hybrid** Direct candidate KL becomes increasingly expensive with model size because each candidate requires downstream recomputation. We therefore initialize the hybrid from the exact BlockFAMQ assignment, retain the selected candidate as a feasibility anchor, and add the two lowest-cost block candidates per unit. Direct KL is then evaluated only on this two- or three-candidate shortlist, followed by exact integer-byte optimization; Proposition 1 guarantees feasibility whenever the original BlockFAMQ problem is feasible. At 3.0 bits, the hybrid improves NLL over BlockFAMQ by $0.035$ and $0.031$ nats on 1.4B and 2.8B at $1.58\times$ and $1.66\times$ scoring cost, respectively. At 3.5 bits, paired differences are $-0.007\,[-0.017, +0.003]$ and $-0.006\,[-0.015, +0.003]$ nats, and at 4.0 bits they shrink to $-0.002\,[-0.008, +0.003]$ and $-0.001\,[-0.006, +0.004]$. Thus, the gain is clearest at the aggressive 3.0-bit budget, while intervals at 3.5 and 4.0 bits include zero. Table 4 reports the complete results.

**Native matched-bit baselines and recurrent transfer** Native stacks change candidate construction and packing. Quamba2 and its BlockFAMQ variant share exact byte targets; SHARP-Q is a nearby native reference. Appendix Table 27 adds a BlockFAMQ row using the corresponding native candidate and packing path at the same serialized target. The comparison is strongest against Quamba2 because it supports both W4A16 and W4A8 operating points. The recurrent transfer panel moves beyond Mamba-1. On Mamba-2 2.7B, the block–diagonal gap is -0.035 $[-0.047, -0.023]$ at 4.05 bits. Falcon-Mamba 7B gives -0.027 $[-0.037, -0.017]$ at 4.02 bits. The corresponding non-recurrent 1.4B/3B comparisons are smaller, as Panel B of Table 3 already suggests. Architecture is not isolated by this broader panel, but the matched-scale Mamba–Pythia contrast and tensor-class bit shifts make the SSM-specific interpretation testable rather than titular. Packed W4A16/W4A8 operating points appear in Appendix Tables 12 and 25. Across 1.4B and 2.8B, BlockFAMQ lowers NLL while retaining 96.8%–98.3% of the corresponding Quamba2 decode throughput.

## 7 Limitations and Conclusion

BlockFAMQ retains curvature interactions only within allocation units, leaving cross-unit terms and nonlinear predictive remainders unmodeled; the observed-token empirical Fisher also differs from model Fisher, although the corresponding ablation preserves the main ordering in our runs. Granularity simultaneously affects curvature fidelity and placement flexibility, while matched native-format and Mamba–Pythia comparisons additionally vary candidate construction, packing, architecture, training, and storage, so these effects cannot be cleanly isolated. The trajectory result in Theorem 9 is a separate diagnostic rather than part of BlockFAMQ, and primary scoring remains costly: five 1.4B calibration draws require $115.0$ A100-GPU-hours. Despite these limitations, BlockFAMQ's mean NLL advantage over diagonal allocation across four Mamba-1 sizes decreases smoothly from $0.218$ nats at 2.5 bits to $0.009$ nats at 4.5 bits; at 3.5 bits, six-task accuracy improves by $2.08$ points and recovers $65.7$–$80.1\%$ of the diagonal quantization gap. At 3.0 bits, direct candidate KL provides further gains, while a block-to-KL shortlist retains most of this improvement at lower scoring cost. Precision maps, matched-scale comparisons, and recurrent transfer are consistent with a state-space-related allocation effect, but do not isolate recurrence as its sole cause.

## Reproducibility Statement

Sections 3–4 specify the candidate residuals, curvature objectives, and exact integer-byte allocator. The appendices provide complete proofs, coordinate partitions, parameter counts, score masks, calibration seeds, paired estimators, bootstrap rules, storage accounting, and packed execution settings. Every reported contrast retains its model, calibration draw, candidate bank, coordinate map, and evaluation unit.

## AI Use Statement

Generative AI assisted with mathematical exposition, literature comparison, and manuscript preparation. The authors verified the final claims, proofs, protocols, and numerical results.

## References


Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

Stella Biderman, Hailey Schoelkopf, Lintang Sutawika, Leo Gao, Jonathan Tow, Baber Abbasi, Alham Fikri Aji, Pawan Sasanka Ammanamanchi, Sidney Black, Jordan Clive, Anthony DiPofi, Julen Etxaniz, Benjamin Fattori, Jessica Zosa Forde, Charles Foster, Jeffrey Hsu, Mimansa Jaiswal, Wilson Y. Lee, Haonan Li, Charles Lovering, Niklas Muennighoff, Ellie Pavlick, Jason Phang, Aviya Skowron, Samson Tan, Xiangru Tang, Kevin A. Wang, Genta Indra Winata, François Yvon, and Andy Zou. Lessons from the trenches on reproducible evaluation of language models. *arXiv preprint arXiv:2405.14782*, 2024. URL https://arxiv.org/abs/2405.14782.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. PIQA: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 7432–7439, 2020. doi: 10.1609/aaai.v34i05.6239. URL https://doi.org/10.1609/aaai.v34i05.6239.

Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, Mohamed S. Abdelfattah, and Diana Marculescu. Quamba2: A robust and scalable post-training quantization framework for selective state space models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 10411–10427, 2025a. URL https://proceedings.mlr.press/v267/chiang25a.html.

Hung-Yueh Chiang, Chi-Chih Chang, Natalia Frumkin, Kai-Chiang Wu, and Diana Marculescu. Quamba: A post-training quantization recipe for selective state space models. In *International Conference on Learning Representations*, 2025b.

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. URL https://arxiv.org/abs/1803.05457.

Tri Dao and Albert Gu. Transformers are SSMs: Generalized models and efficient algorithms through structured state space duality. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 10041–10071, 2024. URL https://proceedings.mlr.press/v235/dao24a.html.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *IEEE/CVF International Conference on Computer Vision*, pp. 293–302, 2019.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18518–18529, 2020.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 4475–4488, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=tEYskw1VY2.

Han Guo, Philip Greengard, Eric P. Xing, and Yoon Kim. LQ-LoRA: Low-rank plus quantized matrix decomposition for efficient language model finetuning. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/d97a16cb83d74195b76e0bf1e85bf072-Abstract-Conference.html.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

Coleman Hooper, Charbel Sakr, Ben Keller, Rangharajan Venkatesan, Kurt Keutzer, Sophia Shao, and Brucek Khailany. FGMP: Fine-grained mixed-precision weight and activation quantization for hardware-accelerated LLM inference. *arXiv preprint arXiv:2504.14152*, 2025. URL https://arxiv.org/abs/2504.14152.

Frederik Kunstner, Lukas Balles, and Philipp Hennig. Limitations of the empirical fisher approximation for natural gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Gongwei Lee, Ji Liu, Juncheng Jia, and Ji Wu. FAMPWQ: Fisher information-based adaptive mixed precision weight quantization for effective LLM inference. *arXiv preprint arXiv:2608.24945*, 2026. URL https://arxiv.org/abs/2608.24945.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021.

Menghao Lv, Huiqiong Wang, Li Sun, and Mingli Song. SHARP-Q: Spectral hessian alignment and rectification for post-training quantization. In *International Conference on Machine Learning*, 2026.

James Martens. New insights and perspectives on the natural gradient method. *Journal of Machine Learning Research*, 21(146):1–76, 2020.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 1525–1534. Association for Computational Linguistics, 2016. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: An adversarial winograd schema challenge at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 8732–8740, 2020. doi: 10.1609/aaai.v34i05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Chen Tianqi, Yuanteng Chen, Peisong Wang, Weixiang Xu, Zeyu Zhu, and Jian Cheng. Q-mamba: Towards more efficient mamba models via post-training quantization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10594–10610, 2025. doi: 10.18653/v1/2025.findings-acl.551.

Yuxuan Yue, Xing Hu, Dawei Yang, Zhihang Yuan, Zixu Jiang, Zhixuan Chen, Jiangyong Yu, Chen Xu, and Sifan Zhou. Mambaquant: Quantizing the mamba family with variance aligned rotation methods. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/51ba8a68f471d952af625d1faf55e6c6-Abstract-Conference.html.

Table 5: Feasibility audit for the block-to-KL shortlist. Byte vectors follow calibration seeds 14011, 14029, 14047, 14071, and 14087. $M_{\min}^{\mathcal{S}}$ is the minimum attainable size of the restricted candidate bank. Final bytes are measured after serialization and reload. Every entry satisfies $M_{\min}^{\mathcal{S}} \le M_{\text{hybrid}} \le M_{\text{target}}$.

| Model | Cap | $M_{\text{target}}$ | BlockFAMQ bytes, draws 1–5 | $M_{\min}^{\mathcal{S}}$, draws 1–5 | Hybrid bytes, draws 1–5 |
|---|---|---|---|---|---|
| 1.4B | 3.0 | 514,566,912 | [514,410,496, 514,322,944, 514,487,808, 514,251,776, 514,445,312] | [507,830,272, 507,612,160, 507,944,960, 507,501,568, 507,766,784] | [514,508,800, 514,476,032, 514,534,400, 514,398,208, 514,521,088] |
| 1.4B | 3.5 | 600,328,064 | [600,120,704, 600,195,968, 600,041,344, 600,233,344, 600,151,936] | [586,343,296, 586,084,224, 586,606,464, 585,821,312, 586,215,808] | [600,264,064, 600,222,592, 600,279,680, 600,206,208, 600,255,360] |
| 1.4B | 4.0 | 686,089,216 | [685,799,424, 685,915,136, 685,732,864, 685,970,432, 685,853,696] | [661,299,968, 660,906,496, 661,578,752, 660,513,280, 661,120,000] | [686,004,224, 685,952,000, 686,028,544, 685,976,832, 685,991,936] |
| 2.8B | 3.0 | 1,038,129,600 | [1,037,678,016, 1,037,800,640, 1,037,562,304, 1,037,915,200, 1,037,733,568] | [1,021,970,880, 1,021,516,224, 1,022,262,208, 1,021,179,328, 1,021,804,992] | [1,038,017,472, 1,037,952,960, 1,038,041,280, 1,037,904,064, 1,037,985,728] |
| 2.8B | 3.5 | 1,211,151,200 | [1,210,615,136, 1,210,748,864, 1,210,479,872, 1,210,862,080, 1,210,673,728] | [1,178,505,952, 1,177,924,416, 1,178,962,208, 1,177,373,088, 1,178,234,464] | [1,211,013,952, 1,210,941,376, 1,211,045,568, 1,210,896,640, 1,210,982,848] |
| 2.8B | 4.0 | 1,384,172,800 | [1,383,548,160, 1,383,683,584, 1,383,414,464, 1,383,821,056, 1,383,624,192] | [1,329,319,168, 1,328,499,968, 1,329,941,760, 1,327,828,224, 1,328,974,592] | [1,384,007,936, 1,383,934,208, 1,384,043,776, 1,383,876,352, 1,383,981,184] |

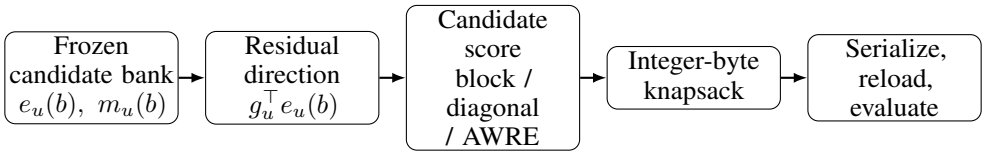



Figure 2: BlockFAMQ changes only the allocation score. Candidate reconstructions and integer byte costs are frozen first. The block score retains the signed gradient–residual interactions inside each allocation unit; diagonal and activation-weighted reconstruction scores replace only this stage. The chosen assignment is serialized, reloaded, and evaluated under the same checkpoint cap.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

# A Supplementary experimental tables and method overview

This appendix provides the protocol schedule, uncertainty details, geometry analysis, calibration-size and granularity studies, packed operating points, and the method schematic that support the main results.

# B Proofs and approximation boundaries

## B.1 The residual ratio and the dither identity

*Proof of Proposition 2.* Positive semidefiniteness gives $e^\top F e \ge 0$ and $|F_{ij}| \le \sqrt{F_{ii}F_{jj}}$. Hence

$$e^\top F e \le \left(\sum_i |e_i|\sqrt{F_{ii}}\right)^2 \le d\sum_i F_{ii}e_i^2.$$

For $F = gg^\top$ with every $g_i \ne 0$, a nonzero $e \perp g$ attains the lower endpoint, and $e_i = c/g_i$ with $c \ne 0$ attains the upper endpoint. Independent centered coordinates satisfy $\mathbb{E}[e_i e_j] = 0$ for $i \ne j$, proving Equation (16). The strong law applies separately to the nonnegative full and diagonal costs because finite second moments give finite expectations. Their ratio converges to one when the common expected cost is positive. This argument does not imply $\mathbb{E}[r_e] = 1$ or $\mathbb{E}[|r_e - 1|] = 0$. □

A subtractively dithered, non-saturating scalar quantizer with independent coordinate dithers supplies the centering conditions. A finite clipped codebook can violate them near its endpoints. Moreover, a clipped squared directional score is nonlinear in the residual covariance, so equality of expected raw quadratics does not establish equality of expected clipped costs. The dither contrast therefore uses its declared quantizer and clipping conventions.

**Proposition 3** (A diagonal optimum can lose to uniform)**.** *There is a two-unit instance with precision choices $\{m-1, m, m+1\}$ and budget $b_1 + b_2 = 2m$ in which the diagonal optimum has four times the full Fisher quadratic of uniform allocation and sixteen times that of the block optimum.*

Table 6: Fixed tied embedding/head precision for each Mamba-1 sweep point. The entry is shared by size-only, diagonal, AWRE, and BlockFAMQ at that point and is charged to the complete-checkpoint cap.

| Model | 2.5 bits | 3.0 bits | 3.5 bits | 4.0 bits | 4.5 bits |
|---|---|---|---|---|---|
| 130M | 2 | 4 | 4 | 8 | 8 |
| 370M | 2 | 4 | 8 | 8 | 8 |
| 1.4B | 2 | 8 | 8 | 8 | 8 |
| 2.8B | 2 | 8 | 8 | 8 | 8 |

Table 7: Mamba-1 four-bit BlockFAMQ minus diagonal contrasts. Negative NLL and PPL changes and positive accuracy changes favor BlockFAMQ.

| Model | $\Delta$NLL [95% CI] | relative PPL change | six-task $\Delta$ pp [95% CI] |
|---|---|---|---|
| 130M | $-0.032$ $[-0.045, -0.019]$ | $-3.15\%$ | $+0.57$ $[+0.26, +0.88]$ |
| 370M | $-0.030$ $[-0.041, -0.019]$ | $-2.96\%$ | $+0.46$ $[+0.20, +0.74]$ |
| 1.4B | $-0.023$ $[-0.032, -0.014]$ | $-2.27\%$ | $+0.31$ $[+0.07, +0.56]$ |
| 2.8B | $-0.019$ $[-0.027, -0.011]$ | $-1.88\%$ | $+0.18$ $[-0.01, +0.38]$ |

*Proof.* For two even-dimensional units, let $F_u = \mathbf{1}\mathbf{1}^\top$ and choose

$$e_1(b) = 2^{-b}s\mathbf{1}, \qquad e_2(b) = 2^{-b}t(1, -1, \ldots, 1, -1), \qquad t^2 = 8s^2 > 0.$$

In units of $\frac{1}{2}ds^2 4^{-m}$, the diagonal costs for $(m, m)$, $(m-1, m+1)$, and $(m+1, m-1)$ are 9, 6, and 32.25. The diagonal optimum is therefore $(m-1, m+1)$. The second residual is orthogonal to $\mathbf{1}$ and contributes zero full quadratic. In units of $\frac{1}{2}d^2s^2 4^{-m}$, the full costs of these assignments are 1, 4, and $1/4$, respectively. This proves both ratios. The construction is an algebraic candidate instance, not a claim about the frequency of such alignments in trained checkpoints. □

### B.2 Diagonal indistinguishability

**Theorem 4** (Diagonal indistinguishability)**.** *Let $d$ be a positive multiple of eight, let $m, k$ be integers satisfying $k \geq 1$ and $m - k \geq 2$, and fix $\varepsilon \geq 0$. There are two Gaussian location models with identical reference parameters, uniform-rounding candidates, candidate byte costs, and Fisher diagonals, under budget $b_1 + b_2 \leq 2m$ and choices $b_u \in \{m-k, \ldots, m+k\}$, for which every allocator using only those inputs has worst-model expected predictive-KL ratio at least*

$$\frac{4^k}{1 + \frac{\varepsilon}{d+\varepsilon}4^{2k}}. \tag{19}$$

*The Fisher matrices are positive definite when $\varepsilon > 0$. At $\varepsilon = 0$, the minimax ratio is exactly $4^k$, attained by uniform allocation. The information restriction includes randomized decision rules.*

*Proof of Theorem 4.* Let $v = (1, -1, \ldots, 1, -1) \in \mathbb{R}^d$. The two model-Fisher matrices are

$$F^{(1)} = \begin{pmatrix} \mathbf{1}\mathbf{1}^\top + \varepsilon I & 0 \\ 0 & vv^\top + \varepsilon I \end{pmatrix}, \qquad F^{(2)} = \begin{pmatrix} vv^\top + \varepsilon I & 0 \\ 0 & \mathbf{1}\mathbf{1}^\top + \varepsilon I \end{pmatrix}.$$

Both arise from Gaussian location families with unit observation covariance: choose a fixed matrix $A$ with $A^\top A = F^{(j)}$ and mean $A\theta$. For $\varepsilon > 0$ both matrices are positive definite. Their diagonals equal $(1+\varepsilon)\mathbf{1}$ in the same parameter coordinates.

Use reference $\theta_1 = \theta_2 = \mathbf{1}/3$. Since $2^b$ is congruent to 1 or 2 modulo 3, rounding $2^b/3$ to the nearest integer gives

$$e_u(b) = \sigma_b \frac{2^{-b}}{3}\mathbf{1}, \qquad \sigma_b = \begin{cases} -1, & b \text{ even}, \\ +1, & b \text{ odd}. \end{cases}$$

For $b \geq 2$ the integer is inside the signed $b$-bit code range, and no tie occurs. The candidates are therefore produced by the actual uniform quantizer, not arbitrary residual vectors. For $d$ divisible by

Table 8: Four-bit task-level BlockFAMQ minus diagonal accuracy differences, in percentage points, with paired 95% intervals.

| | ARC-E | ARC-C | HellaSwag | PIQA | WinoGrande | LAMBADA |
|---|---|---|---|---|---|---|
| $\Delta$ pp | $+0.46$ $[+0.17, +0.76]$ | $+0.31$ $[+0.04, +0.58]$ | $+0.38$ $[+0.15, +0.62]$ | $+0.33$ $[+0.07, +0.59]$ | $+0.27$ $[+0.01, +0.53]$ | $+0.53$ $[+0.18, +0.88]$ |

Table 9: Residual-specific omitted-cross-term magnitude on Mamba-1 130M. The statistic is $\rho_e = |e^\top Fe/(e^\top \operatorname{diag}(F)e) - 1|$ over nonzero-denominator unit–candidate pairs.

| Tensor class | Median $\rho_e$ | 95th percentile $\rho_e$ |
|---|---|---|
| `in_proj.weight` | 0.48 | 2.15 |
| `out_proj.weight` | 0.31 | 2.02 |
| `x_proj.weight` | 0.69 | 19.70 |
| `dt_proj.weight` | 0.32 | 1.27 |
| `conv1d.weight` | 0.20 | 0.86 |
| $A_{\log}$ | 0.21 | 1.10 |
| $D$ | 0.22 | 0.94 |

eight, each unit payload uses $db/8$ bytes. The dyadic scale exponent and precision are encoded with a fixed width chosen for the finite grid, giving identical metadata costs in both models. A common byte cap is exactly equivalent to $b_1 + b_2 \le 2m$.

Write $a = d^2 + \varepsilon d$ and $c = \varepsilon d$. Gaussian predictive KL is exactly quadratic, so

$$D_1(b) = \frac{a4^{-b_1} + c4^{-b_2}}{18}, \qquad D_2(b) = \frac{c4^{-b_1} + a4^{-b_2}}{18}.$$

The allocator receives the same specified inputs in both worlds. A randomized allocator consequently induces the same distribution over assignments in them. For every feasible assignment, the arithmetic–geometric mean inequality gives

$$4^{-b_1} + 4^{-b_2} \ge 2\,4^{-(b_1+b_2)/2} \ge 2\,4^{-m}.$$

Hence

$$\max_{j\in\{1,2\}} \mathbb{E}D_j(b) \ge \frac{\mathbb{E}D_1(b) + \mathbb{E}D_2(b)}{2} \ge \frac{a+c}{18}4^{-m} \ge \frac{a}{18}4^{-m}.$$

By symmetry the two optimal losses are equal. Allocating $m + k$ bits to the sensitive unit and $m - k$ to the other is feasible, so that common optimum is at most $[a4^{-(m+k)} + c4^{-(m-k)}]/18$. Dividing yields Equation (19). At $\varepsilon = 0$, an optimum puts $m + k$ bits on the sensitive unit, while uniform allocation has ratio exactly $4^k$ in either world. This matches the lower bound and proves minimax optimality of the uniform rule within the diagonal-only information class. □

The restriction concerns information supplied to the decision rule. An allocator that evaluates predictive loss or reads off-diagonal geometry is outside this class. Conversely, a more accurate diagonal estimate, a different tie rule, or a more exact knapsack solver remains inside it. For each fixed precision span, the positive-definite construction approaches the singular lower bound as $\varepsilon$ decreases. No assumption about the frequency of these geometries in trained networks is required by the theorem.

### B.3 Predictive KL and model Fisher

Let $D(e) = \mathbb{E}_x \operatorname{KL}(p_\theta(\cdot \mid x) \| p_{\theta+e}(\cdot \mid x))$. The model Fisher uses outcomes drawn from $p_\theta$, rather than the observed next token. The two Fisher estimands need not coincide (Amari, 1998; Martens, 2020; Kunstner et al., 2019).

**Assumption 1** (Local regularity)**.** *For every feasible residual $e$, $D(te)$ is three times continuously differentiable on $0 \le t \le 1$, the score identities hold at $\theta$, and $|\frac{d^3}{dt^3} D(te)| \le L \|e\|_2^3$ for a common finite $L \ge 0$.*

**Theorem 5** (Local predictive distortion)**.** *Under Assumption 1,*

$$D(e) = \tfrac{1}{2} e^\top Fe + R_3(e), \qquad |R_3(e)| \le \tfrac{L}{6} \|e\|_2^3. \tag{20}$$

Table 10: Score-sample-size study on Mamba-1 1.4B at 3.5 bits. Agreement is the storage-weighted fraction of units with the same selected precision as the 8,192-token assignment.

| Scored tokens | Block–diag NLL [95% CI] | agreement to 8k | draw-to-draw agreement | scoring min. |
|---|---|---|---|---|
| 128 | $-0.104$ $[-0.137, -0.073]$ | 72.8% | 68.9% | 42 |
| 256 | $-0.116$ $[-0.144, -0.089]$ | 81.1% | 76.4% | 77 |
| 512 | $-0.127$ $[-0.151, -0.104]$ | 88.3% | 84.2% | 149 |
| 1,024 | $-0.130$ $[-0.153, -0.108]$ | 93.0% | 90.1% | 287 |
| 2,048 | $-0.132$ $[-0.154, -0.110]$ | 96.1% | 94.0% | 563 |
| 4,096 | $-0.133$ $[-0.155, -0.111]$ | 98.1% | 97.0% | 1,108 |
| 8,192 | $-0.134$ $[-0.156, -0.112]$ | 100.0% | 98.7% | 2,196 |

Table 11: Granularity ablation on Mamba-1 1.4B at 3.5 bits. Score time is relative to $R = 1024$.

| Partition | units | Block NLL | Block–diag NLL [95% CI] | score time |
|---|---|---|---|---|
| $R = 256$ | 5,040 | 2.397 | $-0.136$ $[-0.159, -0.113]$ | 1.92× |
| $R = 512$ | 2,544 | 2.399 | $-0.135$ $[-0.158, -0.112]$ | 1.44× |
| $R = 1024$ | 1,296 | 2.402 | $-0.133$ $[-0.155, -0.111]$ | 1.00× |
| $R = 2048$ | 672 | 2.411 | $-0.124$ $[-0.147, -0.102]$ | 0.78× |
| Full tensor | 336 | 2.425 | $-0.109$ $[-0.132, -0.087]$ | 0.61× |

*Proof.* Set $f(t) = D(te)$. Then $f(0) = f'(0) = 0$ and $f''(0) = e^\top Fe$ by the score identities. Taylor's theorem with its Lagrange remainder gives $f(1) = \frac{1}{2}e^\top Fe + \frac{1}{6}f'''(\xi)$ for some $\xi \in (0, 1)$. Assumption 1 bounds the last term. The residual is fixed in this argument, so independent rounding errors are not required. Partitioning the quadratic, including the shared fixed residual as unit zero, gives Equation (17). □

The expansion has no universal small-remainder guarantee at two or three bits. It also concerns predictive KL, not the change in empirical test NLL. These quantities have different targets. Quantized NLL can move in either direction relative to FP32 even though predictive KL is nonnegative.

For a direct single-unit candidate, the same theorem gives $D(\iota_u e_u) = \frac{1}{2}e_u^\top F_{uu}e_u + R_3(\iota_u e_u)$. Consequently, direct per-candidate KL can remove this local approximation without making the total distortion additive. If $S_{\mathrm{KL}}(b) = D(e_0) + \sum_{u=1}^{U} D(\iota_u e_u(b_u))$, then the difference $D(e(b)) - S_{\mathrm{KL}}(b)$ still contains joint interactions. This explains why a candidate-KL allocator and a joint end-to-end KL oracle are distinct baselines.

### B.4 The conditional allocation comparison

Write $Q_F(e) = \frac{1}{2}e^\top Fe$ and $Q_{\mathrm{B}}(e) = \frac{1}{2}\sum_{u=0}^{U} e_u^\top F_{uu}e_u$. Include the fixed unit's constant score when comparing these objectives; it can still be omitted from the solver without changing its selected assignment.

**Assumption 2** (Block and calibration approximation)**.** *There exist common constants $0 \le \rho < 1$ and $\eta \ge 0$ such that every assignment in the same finite feasible set satisfies*

$$(1 - \rho)Q_{\mathrm{B}}(e(b)) \le Q_F(e(b)) \le (1 + \rho)Q_{\mathrm{B}}(e(b)), \tag{21}$$

$$e^{-\eta}Q_{\mathrm{B}}(e(b)) \le \widehat{Q}_{\mathrm{B}}(b) \le e^{\eta}Q_{\mathrm{B}}(e(b)). \tag{22}$$

**Theorem 6** (Exact optimization of an approximate cost)**.** *Let $\widehat{b}$ minimize the encoded block objective exactly and let $b^\star$ minimize $Q_F$ over the same feasible assignments. Under Assumption 2,*

$$Q_F(e(\widehat{b})) \le \frac{1 + \rho}{1 - \rho}e^{2\eta}Q_F(e(b^\star)). \tag{23}$$

*Under Assumption 1, the corresponding upper bound on $D(e(\widehat{b}))$ adds $L\left\|e(\widehat{b})\right\|_2^3/6$ to the right-hand side.*

Table 12: Packed Quamba2 operating points. Decode is tokens/s; peak is device GiB.

| Model | Format | WT2 NLL | Peak | Prefill ms | Decode | retention |
|---|---|---|---|---|---|---|
| 1.4B | W4A16 uniform | 2.462 | 2.68 | 31.2 | 148.3 | – |
| 1.4B | W4A16 + Block | 2.427 | 2.64 | 31.7 | 145.8 | 98.3% |
| 1.4B | W4A8 uniform | 2.503 | 2.31 | 27.4 | 162.0 | – |
| 1.4B | W4A8 + Block | 2.468 | 2.27 | 27.9 | 158.2 | 97.7% |
| 2.8B | W4A16 uniform | 2.318 | 4.31 | 53.6 | 94.8 | – |
| 2.8B | W4A16 + Block | 2.289 | 4.24 | 54.5 | 92.1 | 97.2% |
| 2.8B | W4A8 uniform | 2.365 | 3.72 | 46.3 | 105.2 | – |
| 2.8B | W4A8 + Block | 2.334 | 3.66 | 47.0 | 101.8 | 96.8% |

*Proof.* The calibration inequalities and exact optimality imply

$$Q_{\mathrm{B}}(e(\widehat{b})) \le e^{\eta}\widehat{Q}_{\mathrm{B}}(\widehat{b}) \le e^{\eta}\widehat{Q}_{\mathrm{B}}(b^{\star}) \le e^{2\eta}Q_{\mathrm{B}}(e(b^{\star})).$$

Apply the upper block inequality to $\widehat{b}$ and the lower block inequality to $b^{\star}$. Theorem 5 then supplies the additive KL remainder. □

The comparison is conditional on a uniform statement over all feasible assignments. A sample maximum of full-to-block ratios does not establish that condition. The calibration factor must also cover the observed-token versus model-Fisher discrepancy, clipping, finite sampling, and candidate construction. The bound requires these conditions for the same candidate set used in the comparison; a diagnostic from a different partition cannot supply its constants.

### B.5 Frozen-candidate concentration and discrete optimization

**Proposition 7** (Concentration across independent sequences)**.** *Condition on an independent construction and tuning sample that fixes $N_c$ residual candidates and thresholds $B_{u,b}$. Let $K$ independent identically distributed scoring sequences contain the same positive number of scored positions, and let $X_{u,b}(z)$ be their average clipped squared directional score, including the factor $1/2$. For $\widehat{C}_u^{(B)}(b) = K^{-1}\sum_k X_{u,b}(z_k)$, with probability at least $1-\delta$, simultaneously over the candidates,*

$$\left|\widehat{C}_u^{(B)}(b) - \mathbb{E}X_{u,b}\right| \le \frac{B_{u,b}^2}{2}\sqrt{\frac{\log(2N_c/\delta)}{2K}}. \tag{24}$$

*Proof.* Each sequence average lies in $[0, B_{u,b}^2/2]$. Hoeffding's inequality across sequences and a union bound over the frozen candidates give the result (Hoeffding, 1963). Equal token counts make the mean of sequence averages identical to the token-normalized estimator. No independence among tokens within a sequence is used. □

For unequal scored-token counts, the token-weighted ratio estimator needs its own analysis; treating its tokens as independent does not repair the mismatch. Clipping lowers the expected raw cost by

$$\tfrac{1}{2}\mathbb{E}\left[(Z^2 - B_{u,b}^2)\mathbf{1}\{Z^2 > B_{u,b}^2\}\right], \qquad Z = g_u^{\top}e_u(b),$$

with the same within-sequence averaging convention. The concentration result controls sampling around this clipped observed-data target, not model Fisher.

**Lemma 8** (Pareto-frontier allocation)**.** *For additive integer byte costs and finite candidate scores, retaining every nondominated cumulative size–score pair after each unit preserves an optimum of Equation* (10)*. Within-unit dominance pruning also preserves an optimum.*

*Proof.* Induct on the unit prefix. Before pruning, every feasible prefix assignment appears among the cumulative pairs. A pair dominated in both size and score cannot have a better completion: replacing it by its dominating prefix leaves at least as much capacity and cannot raise the objective for any common suffix. The same replacement argument applies to a dominated candidate within one unit. Back pointers recover an optimal surviving assignment. □

This is a statement about the encoded arithmetic and complete frontier. Rounding byte costs, dropping frontier states heuristically, or terminating with a positive solver gap changes the optimization claim.

# C Calibration and coordinate-level implementation

## C.1 One partition for the complete Mamba evaluation

For model width $d$, the standard Mamba-1 expansion has inner width $2d$, state size $s = 16$, convolution width 4, and time-step rank $r = \lceil d/16 \rceil$ (Gu & Dao, 2024). For tensor $T$ in layer $\ell$, let $n_{\ell,T}$ be its number of output rows and let $\mathcal{J}_{\ell,T}$ contain all trailing-coordinate indices. The row partition is

$$\mathcal{I}_{\ell,T,j} = \left( [jR, \min\{(j+1)R, n_{\ell,T}\}) \cap \mathbb{Z} \right) \times \mathcal{J}_{\ell,T}, \qquad 0 \le j < \lceil n_{\ell,T}/R \rceil , \tag{25}$$

with $R = 1{,}024$. The final tile retains its actual shorter size. Canonical unit identifiers are ordered by layer index, tensor order in Table 13, and first output-row index. A precision assignment uses this same coordinate map at every sweep point and for every Mamba-1 bank. A change to $R$ defines a different allocation problem and must not share an assignment identifier with the original map.

Table 13: Mamba-1 allocation units and curvature blocks. A row tile contains all trailing coordinates. Biases and normalization parameters are fixed at FP32. Matrix and convolution scales are per output row; vector scales are per tile.

| Tensor | Shape | Units per layer |
|---|---|---|
| `in_proj.weight` | $(4d) \times d$ | $\lceil 4d/R \rceil$ |
| `out_proj.weight` | $d \times (2d)$ | $\lceil d/R \rceil$ |
| `x_proj.weight` | $(r + 2s) \times (2d)$ | $\lceil (r+2s)/R \rceil$ |
| `dt_proj.weight` | $(2d) \times r$ | $\lceil 2d/R \rceil$ |
| `conv1d.weight` | $(2d) \times 1 \times 4$ | $\lceil 2d/R \rceil$ |
| `A_log` | $(2d) \times s$ | $\lceil 2d/R \rceil$ |
| `D` | $2d$ | $\lceil 2d/R \rceil$ |

The total number of units for $L$ layers is

$$U = L \left( \left\lceil \frac{4d}{R} \right\rceil + \left\lceil \frac{d}{R} \right\rceil + \left\lceil \frac{r+2s}{R} \right\rceil + 4 \left\lceil \frac{2d}{R} \right\rceil \right). \tag{26}$$

Table 14 evaluates this expression. These counts refer to all seven eligible tensor types, rather than the five scan-related types alone. The embedding and language-model head share one fixed tensor and are not additional allocation units.

Table 14: Architecture-derived unit counts for the single row-partition rule. The final column counts six precision candidates per unit before dominance pruning. All five effective-bit targets use the same map.

| Model | Width | Layers | Units/layer | Units | Candidates |
|---|---|---|---|---|---|
| 130M | 768 | 24 | 13 | 312 | 1,872 |
| 370M | 1,024 | 48 | 14 | 672 | 4,032 |
| 1.4B | 2,048 | 48 | 27 | 1,296 | 7,776 |
| 2.8B | 2,560 | 64 | 34 | 2,176 | 13,056 |

An executable map associates each canonical unit with its tensor name, full shape, inclusive first and exclusive final row, trailing shape, scale ownership, and candidate identifiers. Its coordinate sets must be pairwise disjoint and must cover the eligible parameters. The union with fixed parameters must cover all unique parameters exactly once. Tied aliases point to their canonical storage owner. These are structural checks on the allocation problem, distinct from a quality comparison between the resulting checkpoints.

## C.2 Candidate construction and size-only allocation

A candidate bank is fixed independently for each calibration draw. For linear matrices, the primary reconstruction uses a GPTQ-style damped second-order rounding pass on full-precision input activations (Frantar et al., 2023). The damping is 0.01 times the mean diagonal of the input Gram

matrix and the column processing block has $128$ columns. Output-row quantization is symmetric with zero point zero. Candidate ranges are chosen from clipping fractions $\{1.00, 0.99, 0.98, 0.95\}$ by reconstruction error on construction documents. A chosen range belongs to the bank and is shared by all objectives.

The convolution, $A_{\log}$, and $D$ use symmetric rounding within their respective scaling groups. The RTN comparison replaces the compensated linear reconstructions by symmetric round-to-nearest candidates while keeping the row partition and all other conventions unchanged. Precision choices are $\{2, 3, 4, 5, 6, 8\}$. A zero-range scaling group is encoded as an all-zero group with its specified metadata; it is not assigned an arbitrary sensitivity.

Let $n_u$ be the number of scalars in unit $u$. The size-only objective is

$$C_u^{\text{size}}(b) = n_u 4^{-b}. \tag{27}$$

It is a parameter-count proxy, not a fitted quantization-error or gradient score. It is optimized under the same integer-byte constraint. Ties use the lexicographic order of SHA-256 digests of the canonical unit identifiers, followed by the precision index. This rule is fixed before calibration. The hash resolves indifference without inspecting model quality; it is not a claimed random sample of every feasible mixed assignment.

Greedy promotion starts at the minimum-byte candidate in every unit. At each step it considers feasible moves with positive cost reduction and ranks them by reduction divided by additional bytes. It applies the best move, resolving ties by the same canonical hash order, and stops when no beneficial promotion fits. For a nonmonotone candidate bank, within-unit dominance pruning precedes this procedure. Exact allocation instead retains the full nondominated frontier or solves the binary program to its declared tolerance. The finite candidate set and integer sizes must coincide across the two solvers.

### C.3 Calibration positions, evaluation masks, and FP32

The calibration seeds are $14011$, $14029$, $14047$, $14071$, and $14087$. For each draw, candidate construction uses $128$ C4 training documents and scoring uses $128$ other documents. Each document supplies a $2{,}048$-token context. The primary score uses $32$ predetermined target positions per scoring document, evenly spaced over the valid target range, giving $4{,}096$ scored positions per draw. Each selected next token is predicted from its own preceding context. The sample-size ablation subsamples this fixed pool and extends it to $8{,}192$ positions with the corresponding $64$-position mask. A further $32$ documents determine the optional clipping thresholds; the raw primary comparison does not depend on those thresholds. Source-document membership, not just a different offset, defines disjointness.

WikiText-2 uses the complete test split with source-article boundaries retained. C4 uses the first disjoint validation documents in canonical dataset order until at least $10^6$ scored tokens are included. Each source is evaluated separately with context length $2{,}048$ and stride $1{,}024$. A token used only as context is not scored again in the next overlapping window. Final partial windows retain valid targets, and each source's first token is excluded because it has no preceding in-source context. Both arms of a comparison use the same target mask. The primary forward path dequantizes packed weights to FP32 and uses FP32 activations and recurrent states; it is a quality path, not a low-bit kernel benchmark.

The FP32 reference uses the same model weights before quantization, the same tokenizer, and the same contexts and masks. It is evaluated once per checkpoint and dataset. In a paired resample, its document outcomes are reused against all quantized calibration draws. Reusing a deterministic reference does not create independent FP32 replications. Gap recovery is defined only when the comparator's NLL exceeds that reference.

Downstream evaluation is zero-shot. ARC-Easy, ARC-Challenge, HellaSwag, and PIQA use length-normalized choice likelihood, WinoGrande uses the selected choice, and LAMBADA uses exact next-word accuracy. Each task is first averaged over its own items; the six task scores then receive equal weight. The four model scores receive equal weight at the next level. Neither token counts across corpora nor item counts across tasks replace these declared weights.

### C.4 Paired estimators and bootstrap construction

For model $m$, draw $k$, arm $a$, and source $d$, retain the scored-loss sum $S_{mkad}$ and target count $n_{mkd}$. The latter is identical across paired arms. The document-weighted token loss for draw $k$ and arm $a$ is

$$L_{mka} = \frac{\sum_d S_{mkad}}{\sum_d n_{mkd}}, \qquad \widehat{\Delta}_m(a,b) = \frac{1}{K}\sum_{k=1}^{K}(L_{mka} - L_{mkb}). \tag{28}$$

This is a ratio of loss sums, not an unweighted average of document NLLs. Across the four fixed checkpoints, the aggregate is $\widehat{\Delta} = \frac{1}{4}\sum_m \widehat{\Delta}_m$. Model identities are not resampled as though they were independent draws of architecture families.

A bootstrap replicate resamples calibration-draw indices with replacement. It also resamples source-document clusters with replacement, preserving all windows from a source. Because each calibration draw is evaluated on the same sources, the same source multiplicities are used across selected draws and paired arms. Shared source identifiers are likewise paired across models; model-specific scored-token denominators are retained. Each replicate recomputes Equation (28) from sums and counts before model averaging. For the downstream estimand, the six benchmarks remain fixed. Item identifiers are resampled within each task, with the same multiplicities across paired arms and calibration draws. The replicate averages task scores with the original six equal weights.

The protocol uses $B = 10{,}000$ replicates and bootstrap seed 23017. For sorted replicate effects $\Delta^*_{(1)} \le \cdots \le \Delta^*_{(B)}$, let $h = 1 + (B-1)p$, $j = \lfloor h \rfloor$, and $\alpha = h - j$. For $0 < p < 1$,

$$q_p = (1-\alpha)\Delta^*_{(j)} + \alpha\Delta^*_{(j+1)}, \qquad \mathrm{CI}_{0.95} = [q_{0.025}, q_{0.975}]. \tag{29}$$

Each endpoint is computed separately from the ordered replicate effects. A reproducible bootstrap output contains the generation-order replicate vector, unrounded point estimate, sampled draw and source or item multiplicities, and quantile convention. Checkpoint, candidate-bank, coordinate-map, and target-mask identifiers connect that output to its paired observations. The intervals target individual contrasts, not a simultaneous family-wise confidence statement.

### C.5 Matched baseline adaptations and direct-KL cost

HAWQ-V2, FAMPWQ, and SHARP-Q adaptations replace the candidate-scoring stage but not candidate construction, coordinate grouping, or complete-checkpoint costs. The SHARP-Q-inspired geometry setting uses rank $64$, $32$ probe vectors per unit, and damping $10^{-4}$. It excludes a separately optimized native rectification pass. The matched Quamba2 search receives the same bank and $2{,}000$ proposals, with a search seed paired to each calibration draw. The best feasible proposal is compared with exact optimization of the block and diagonal objectives. These settings define controlled adaptations; they do not imply replication of a native quantization stack.

For direct KL, teacher and candidate distributions have the same vocabulary support. Accumulation uses log probabilities and the complete categorical sum, not a top-$k$ truncation. Upstream activations are reusable only where the candidate leaves the computation unchanged. The affected layer and every subsequent recurrent trajectory are recomputed. Candidate scoring includes this work and, for BlockFAMQ, the per-token differentiation work. Common candidate construction and final quality evaluation are excluded from both scoring-time columns. GPU-minutes sum device occupancy across the participating GPUs. The timing configuration is one A100-80GB per model-scoring job, with identical microbatching and score positions across objectives. For the primary 1.4B run, GPTQ bank construction costs 272 GPU-minutes per draw. The $4{,}096$-token BlockFAMQ score pass is a separate 1,108 GPU-minutes, matching the scoring-time column of Table 10. The complete per-draw calibration cost is therefore 1,380 GPU-minutes, and five draws cost 6,900 GPU-minutes, or 115.0 A100-GPU-hours. Scoring alone costs 16.2 GPU-seconds per scored token. A profiler decomposition assigns 1.2 seconds to the forward/backward calculation, 13.9 seconds to streaming the token gradient through the complete residual bank and accumulating all candidate directional products, and 1.1 seconds to transfer, launch, and reduction overhead. GPTQ construction is a fixed bank-construction cost and is not divided by the number of scored tokens. The same candidate residuals and score table are reused for all five budget targets, so only the inexpensive integer allocation and serialization steps repeat across the budget sweep.

# D COMPLETE-CHECKPOINT STORAGE

## D.1 ARCHITECTURE-DERIVED PARAMETER COUNTS

The padded vocabulary has $V = 50{,}280$ entries and one tied $V \times d$ embedding. For the standard Mamba-1 shapes of Table 13, with convolution and time-step biases but bias-free input and output projections, a layer has

$$N_{\text{layer}} = 6d^2 + 4dr + 111d. \tag{30}$$

The quadratic term is the input/output projection pair. The $4dr$ term comes from the two time-step-rank projections. The remaining coefficient includes state projections, convolution, $A_{\log}$, skip weights, biases, and the layer normalization. With one final normalization, the unique model count is

$$N_{\text{param}} = Vd + L(6d^2 + 4dr + 111d) + d. \tag{31}$$

Table 15 reports the resulting unique parameter counts and byte ceilings. These are calculations from the declared architecture (Gu & Dao, 2024), not rounded model names. A converted checkpoint with additional untied parameters defines a different denominator and is outside this tied-checkpoint specification.

Table 15: Calculated unique parameter counts and integer-byte caps for tied Mamba-1 checkpoints. The final column is the eight-bit-embedding payload lower bound, not the complete serializer minimum. The 130M aggressive configuration instead uses a four-bit embedding.

| Model | $N_{\text{param}}$ | Embedding scalars | 4.00-bit cap | 3.50-bit cap | $b_{\text{emb}}$ |
|---|---|---|---|---|---|
| 130M | 129,135,360 | 38,615,040 | 64,567,680 | 56,496,720 | 3.794 |
| 370M | 371,516,416 | 51,486,720 | 185,758,208 | 162,538,432 | 2.832 |
| 1.4B | 1,372,178,432 | 102,973,440 | 686,089,216 | 600,328,064 | 2.450 |
| 2.8B | 2,768,345,600 | 128,716,800 | 1,384,172,800 | 1,211,151,200 | 2.279 |

The embedding payload share at target $B$ is

$$S_E(B) = 100\,\frac{b_E V d}{B N_{\text{param}}}\%, \qquad b_{\text{emb}} = 2 + (b_E - 2)\frac{Vd}{N_{\text{param}}}. \tag{32}$$

The first quantity is a storage share, not a parameter fraction; the second is an embedding-only necessary payload bound. Scales, tags, padding, container metadata, and the higher-precision fixed biases and norms increase the exact minimum. For 130M, the four-bit-embedding configuration lowers $b_{\text{emb}}$ from $3.794$ to $2.598$ bits. It does not lower the reference FP32 parameter count or remove the embedding from the budget.

## D.2 CANDIDATE COSTS AND RELOAD SEMANTICS

Each candidate contains packed signed integer codes, its FP32 scales, a precision tag, and alignment padding. The symmetric zero point is implicit zero; a nonzero stored zero-point array would also be charged. Metadata shared independently of candidate choices belongs to $M_{\text{fixed}}$. Candidate-dependent metadata belongs to $m_u(b)$. Tensor names, shapes, row offsets, fixed arrays, and container headers are included in the complete serialized size.

The budget test is on the actual integer total, not on the mean selected bit width. After reload, unpacked codes and scales must reconstruct the selected candidate tensors before any quality comparison. The embedding/head storage alias is checked once, and the union of the variable and fixed coordinates covers the declared parameter count. Unused bytes are slack, not another method's spending allowance. A solver objective can be exact while the model loss is not; similarly, a valid cap calculation does not establish a native kernel implementation for every selected precision.

# E FP32 REFERENCES AND DOWNSTREAM DETAIL

## E.1 BOTH BUDGETS AGAINST THE SAME REFERENCE

Table 16 compares the two complete-checkpoint caps with a common FP32 reference. The diagonal-gap fraction uses Equation (18). The block–diagonal differences are absolute NLL changes, whereas

the recovered fraction divides by the comparator's positive FP32 gap. A small absolute improvement can recover a substantial fraction of a small gap; the two quantities should not be substituted for one another.

Table 16: WikiText-2 reference losses and recovered diagonal quantization gap. Recovery is $100(L_{\mathrm{diag}} - L_{\mathrm{B}})/(L_{\mathrm{diag}} - L_{32})$, defined for a positive denominator.

| Model | Cap | FP32 | Size-only | Diagonal | Block | Recovery |
|---|---|---|---|---|---|---|
| 130M | 3.50 | 3.222 | 3.608 | 3.486 | 3.301 | 70.1% |
| 130M | 4.00 | 3.222 | 3.331 | 3.313 | 3.281 | 35.2% |
| 370M | 3.50 | 2.765 | 3.045 | 2.956 | 2.803 | 80.1% |
| 370M | 4.00 | 2.765 | 2.841 | 2.826 | 2.796 | 49.2% |
| 1.4B | 3.50 | 2.357 | 2.612 | 2.535 | 2.402 | 74.7% |
| 1.4B | 4.00 | 2.357 | 2.431 | 2.419 | 2.396 | 37.1% |
| 2.8B | 3.50 | 2.233 | 2.471 | 2.405 | 2.292 | 65.7% |
| 2.8B | 4.00 | 2.233 | 2.295 | 2.286 | 2.267 | 35.8% |

For 130M, the two main cap configurations differ in embedding precision. Within each cap, the embedding is common to all quantized arms. Across caps, its contribution is separated by the fixed-embedding control in Appendix G. A budget-response curve that changes both factors cannot identify either contribution alone.

### E.2 Six-task means and paired differences

Table 17: Paired six-task mean differences at 3.50 complete-checkpoint bits, in percentage points, with fixed-suite 95% interval endpoints.

| Model | Block minus size-only | Block minus diagonal |
|---|---|---|
| 130M | $+4.21$ $[+3.16, +5.46]$ | $+2.55$ $[+1.64, +3.61]$ |
| 370M | $+4.69$ $[+3.63, +5.90]$ | $+2.89$ $[+1.90, +4.00]$ |
| 1.4B | $+3.62$ $[+2.80, +4.57]$ | $+2.10$ $[+1.37, +2.96]$ |
| 2.8B | $+2.28$ $[+1.60, +3.10]$ | $+0.78$ $[+0.20, +1.47]$ |
| Aggregate | $+3.70$ $[+3.01, +4.49]$ | $+2.08$ $[+1.55, +2.73]$ |

Table 18: Aggressive-budget task accuracy percentages, averaged equally across the four Mamba-1 checkpoints. The final column averages the six displayed tasks.

| Arm | ARC-E | ARC-C | HellaSwag | PIQA | WinoGrande | LAMBADA | Mean |
|---|---|---|---|---|---|---|---|
| FP32 | 62.21 | 31.42 | 56.13 | 71.11 | 55.03 | 48.16 | 54.01 |
| Size-only | 53.44 | 24.37 | 46.65 | 64.88 | 50.22 | 39.92 | 46.58 |
| Diagonal | 55.77 | 26.14 | 48.82 | 66.57 | 51.66 | 40.24 | 48.20 |
| Block | 58.30 | 27.84 | 50.91 | 68.12 | 52.76 | 43.75 | 50.28 |

The downstream tables give the same four-bit grand mean of 0.38 percentage points for BlockFAMQ minus diagonal after differencing the displayed task accuracies and averaging the same model–task rectangle. The per-model mean averages six tasks, and the per-task mean averages four models. These operations commute because the weights form the same rectangular product. Pairwise uncertainty nevertheless requires item outcomes: averaging an accuracy table does not reconstruct shared mistakes or recover the paired sampling variance. The fixed-suite bootstrap specification is Appendix C.4.

## F Geometry and allocation controls

### F.1 Per-tensor realized-residual ratios

The per-tensor diagnostic uses the same 4,096 score positions per draw and the same frozen candidates as the Mamba allocation. It computes $\rho_e$ for each nonzero-denominator unit–candidate pair, pooling

neither token-level ratios nor unrelated random directions. The displayed summaries aggregate pairs within each tensor type. A zero denominator under a positive-semidefinite Fisher block implies a zero full quadratic, so the ratio is excluded as undefined rather than assigned an arbitrary value. Those pairs still remain valid zero-cost candidates in the allocator.

The projection tail in Table 9 is larger than its median. The diagnostic identifies candidate-specific directional effects, rather than establishing diagonal dominance of an entire matrix or attributing every allocation gain to one tensor type. The signed ratio $\chi_e$ determines whether the diagonal score underprices or overprices a candidate. Its absolute value measures omission magnitude but does not define an allocator.

### F.2 Dither and partition dependence

Table 19: Residual-geometry control on Mamba-1 1.4B at the four-bit cap. $r_\Sigma$ is a ratio of summed full and diagonal costs, not an average of per-residual ratios. NLL differences are block minus diagonal.

| Residual | $r_\Sigma$ | Median $\rho_e$ | 95th $\rho_e$ | $\Delta$NLL [interval] |
|---|---|---|---|---|
| Deterministic | 1.92 | 0.64 | 2.87 | $-0.023$ $[-0.032, -0.014]$ |
| Subtractive dither | 1.04 | 0.19 | 0.83 | $-0.004$ $[-0.011, +0.003]$ |

For the dither comparison, each coordinate receives an independent uniform dither with width equal to its scalar quantization step, followed by subtraction of the same dither. Conditional on a non-saturating scalar quantizer, the error is centered and independent of the input. Finite clipped endpoints can violate that condition. The ratio-of-sums diagnostic uses the raw quadratic. Saturation can prevent its expectation identity from applying; candidate-dependent score clipping also falls outside that identity. Dither changes realized error energy and the selected assignment, so matching scalar step sizes alone is not an energy-matched causal intervention.

The partition is also a substantive hyperparameter. Splitting a unit deletes cross terms between the resulting pieces and increases precision-placement freedom. Merging units retains more interactions but forces a joint precision. Consequently, a granularity comparison must rebuild both the candidate map and the integer costs under its new partition. The resulting quality difference cannot be assigned solely to a more accurate curvature approximation.

## G Budget, backend, and execution comparisons

### G.1 Fixed embedding and direct candidate KL

Table 20: Mamba-1 130M crossed embedding–budget comparison. Both four-bit-embedding rows use one fixed reconstruction, separating the embedding change from the complete-checkpoint budget change.

| Embedding | Cap | Size-only | Diagonal | Block | Block–diag. |
|---|---|---|---|---|---|
| 8 bit | 4.00 | 3.331 | 3.313 | 3.281 | $-0.032$ |
| 4 bit | 4.00 | 3.309 | 3.296 | 3.274 | $-0.022$ |
| 4 bit | 3.50 | 3.608 | 3.486 | 3.301 | $-0.185$ |

The four-bit-embedding rows have the same fixed reconstruction. Tightening the cap changes the variable allocation and leaves that reconstruction unchanged. The increase in the block–diagonal advantage is $0.163$ nats. The eight-bit-embedding $3.50$-bit configuration is infeasible by Equation (4); it is not replaced by a row with a different realized cap. The direct-KL comparison in Table 4 uses the same embedding configuration as the corresponding block row.

Table 21: Four-bit task accuracy percentages averaged equally across the four Mamba-1 models. HAWQ-V2, FAMPWQ, SHARP-Q, and Quamba2 search are the matched allocation-stage adaptations.

| Arm | ARC-E | ARC-C | HellaSwag | PIQA | WinoGrande | LAMBADA | Mean |
|---|---|---|---|---|---|---|---|
| Size-only | 58.40 | 27.91 | 51.77 | 68.52 | 53.06 | 43.88 | 50.59 |
| HAWQ-V2 | 58.55 | 28.00 | 51.90 | 68.62 | 53.14 | 44.05 | 50.71 |
| FAMPWQ | 58.70 | 28.09 | 52.01 | 68.72 | 53.23 | 44.23 | 50.83 |
| Diagonal | 58.62 | 28.04 | 51.96 | 68.68 | 53.18 | 44.14 | 50.77 |
| SHARP-Q | 58.82 | 28.16 | 52.13 | 68.83 | 53.32 | 44.36 | 50.94 |
| Quamba2 search | 58.76 | 28.13 | 52.08 | 68.79 | 53.28 | 44.31 | 50.89 |
| Block | 59.08 | 28.35 | 52.34 | 69.01 | 53.45 | 44.67 | 51.15 |

Table 22: Candidate-backend comparison with equal weights across four Mamba-1 sizes. Each block–diagonal contrast uses one common bank. The comparison between rows additionally changes the candidate generator.

| Bank | Size-only WT2 | Block–size | Block–diagonal [interval] |
|---|---|---|---|
| RTN | 2.7350 | $-0.0460$ | $-0.034$ $[-0.043, -0.025]$ |
| GPTQ | 2.7245 | $-0.0395$ | $-0.026$ $[-0.032, -0.019]$ |

### G.2 Four-bit task and backend comparisons

All backend comparisons hold candidate construction fixed within their row. The GPTQ–RTN difference itself is a candidate-generator effect and is not included in the within-bank block–diagonal contrast. Similarly, the six-task scores weight tasks equally. They do not imply a separate per-task superiority claim or a multiplicity-adjusted family of rejections.

### G.3 Crossed reconstructed packages and execution paths

A reconstructed weight package includes every reconstructed tensor, the fixed embedding, scale arrays, precision choices, and storage metadata. The crossed comparison evaluates that exact package under two execution paths with common A16 activation and state conventions and identical target masks. A format not supported natively uses dequantization followed by the compatible matrix path. This checks a numerical execution difference but supplies no claim about a native two-bit kernel or acceleration (Table 23).

Table 23: Crossed reconstructed-weight package and execution-path comparison on Mamba-1 1.4B. Each row keeps its own weights, activation and state precision, and target masks fixed across paths. Packages are not required to have the same byte count.

| Package | Reference | Quamba2-compatible | Path difference |
|---|---|---|---|
| Native uniform package | 2.460 | 2.462 | $+0.002$ |
| Matched GPTQ, size-only | 2.431 | 2.433 | $+0.002$ |
| Matched GPTQ, block | 2.396 | 2.398 | $+0.002$ |

The native-to-GPTQ size-only contrast is $-0.029$ nats on each execution path, while the within-package path contrast is $+0.002$. These are different comparisons. The former changes both candidate construction and precision placement; the latter holds the reconstructed package fixed. The common-bank block–size comparison isolates the allocation objective rather than the full package.

## H Context length, packed operation, and transfer

### H.1 Context length with the allocation held fixed

The PG19 comparison keeps each model's checkpoint, candidates, and allocation fixed while increasing the context window. Source books are clustered in the paired resampling. Each context length has its own scored-token mask shared between arms; comparing different lengths is not a resample of

the same conditional likelihood. The model aggregate includes 370M, 1.4B, and 2.8B, with equal weights, and excludes 130M. Table 24 reports the resulting context-length contrasts.

Table 24: PG19 block–diagonal NLL differences with the checkpoint assignment held fixed as context length changes. The aggregate covers 370M, 1.4B, and 2.8B only.

| Model | 2k | 4k | 8k | 16k |
|---|---|---|---|---|
| 370M | −0.018 | −0.026 | −0.039 | −0.054 |
| 1.4B | −0.014 | −0.021 | −0.033 | −0.048 |
| 2.8B | −0.011 | −0.017 | −0.025 | −0.038 |
| Aggregate | −0.0143 | −0.0213 | −0.0323 | −0.0467 |

This comparison asks whether a block–diagonal allocation difference persists as recurrent computation grows longer. It does not evaluate a hard guard or a soft recurrence penalty. The aggregate values are averages of the displayed model differences before final rounding.

### H.2 Packed Quamba2 composition

The packed operating point uses one A100-80GB, batch size one, a 2,048-token prefill, and 128 generated tokens. Quality and timing refer to the same reconstructed checkpoint. The timing scope includes format dispatch and any conversion used by that configuration. A measurement protocol uses 20 warmups followed by 100 synchronized trials; throughput retention is the ratio of decode token rates, not the reciprocal latency or a speedup.

Table 25: Packed Quamba2 composition at separate operating points on one A100-80GB. GB is decimal file size, GiB is peak memory, and decode is tokens per second. These rows are not the common four-bit allocation cap.

| Model | Configuration | GB | WT2 NLL | GiB | Prefill ms | Decode |
|---|---|---|---|---|---|---|
| 1.4B | Quamba2 uniform | 0.82 | 2.462 | 2.68 | 31.2 | 148.3 |
| 1.4B | Quamba2 + block | 0.80 | 2.427 | 2.64 | 31.7 | 145.8 |
| 2.8B | Quamba2 uniform | 1.57 | 2.318 | 4.31 | 53.6 | 94.8 |
| 2.8B | Quamba2 + block | 1.53 | 2.289 | 4.24 | 54.5 | 92.1 |

The file-size column is rounded decimal GB and memory is GiB. These are separate operating points, not the common 4.00-bit cap. Rounding file sizes does not provide an exact byte ledger. The paired rates imply throughput retention of 98.3% on 1.4B and 97.2% on 2.8B. Those ratios quantify the cost of this composition; they do not establish a hardware-independent advantage of mixed precision.

### H.3 Native references and cross-family scope

Tables 26 and 28 report the native operating points and architecture-specific transfer comparison.

Table 26: Native-format reference operating points, separate from the common-bank allocation comparison. Effective-bit labels include each configuration's own retained tensors and format costs.

| Method | Model | Format | Eff. bits | WT2 NLL | Task mean |
|---|---|---|---|---|---|
| SHARP-Q | 1.4B | W4, A16 | 4.71 | 2.441 | 58.28 |
| Quamba2 search | 1.4B | W4A16 | 4.69 | 2.456 | 57.92 |

Native-format rows use their own reconstruction and packing conventions and are not included in the common-bank aggregate. In particular, Quamba2 search and uniform Quamba2 are different configurations. The cross-family panel uses row tiles of at most 1,024 output rows for linear weights and architecture- specific storage for the remaining tensors. Within each family, the map and fixed set are common across objectives; they are not presumed to match the Mamba-1 unit counts. Pythia's input embedding and language-model output matrix are untied and are therefore represented as two distinct stored tensors in its parameter denominator and byte ledger; the tied Mamba embedding/head

Table 27: Native-format comparisons. Quamba2 search and Quamba2 + BlockFAMQ share the stated complete-checkpoint effective bits within each model; SHARP-Q is a nearby reference. The BlockFAMQ variant uses the Quamba2 candidate formats and packing path.

| Model | Method | effective bits | WT2 NLL/PPL | six-task mean |
|---|---|---|---|---|
| 1.4B | SHARP-Q native | 4.71 | 2.441/11.48 | 58.28 |
| 1.4B | Quamba2 search W4A16 | 4.69 | 2.456/11.66 | 57.92 |
| 1.4B | Quamba2 + BlockFAMQ | 4.69 | 2.425/11.30 | 58.41 |
| 2.8B | SHARP-Q native | 4.50 | 2.304/10.01 | 62.02 |
| 2.8B | Quamba2 search W4A16 | 4.48 | 2.318/10.15 | 61.76 |
| 2.8B | Quamba2 + BlockFAMQ | 4.48 | 2.288/9.86 | 62.18 |

Table 28: Cross-family NLL panel at model-specific complete-checkpoint targets. Comparisons within each row share a partition and candidate bank; the panel does not match architecture, pretraining data, or model size across rows.

| Model | Target | Size-only | Diagonal | Block | Block–diag. |
|---|---|---|---|---|---|
| Mamba-2 2.7B | 4.05 | 2.305 | 2.287 | 2.252 | −0.035 |
| Falcon-Mamba 7B | 4.02 | 2.119 | 2.104 | 2.077 | −0.027 |
| Pythia 1.4B | 4.04 | 2.641 | 2.628 | 2.619 | −0.009 |
| Llama 3.2 3B | 4.03 | 2.232 | 2.221 | 2.214 | −0.007 |

accounting is not transferred to Pythia. The block–diagonal differences are $-0.035$, $-0.027$, $-0.009$, and $-0.007$ nats, respectively. A larger recurrent-model margin in such a panel does not isolate recurrence, because architecture, scale, training data, and unit structure also differ.

# I A BOUNDARY OF RELATIVE TRAJECTORY GUARDS

The following guard is not part of BlockFAMQ. It describes a distinct relative-kernel constraint on a selective recurrence. For

$$h_t = a_t h_{t-1} + x_t, \qquad a_t = \exp(\Delta_t A), \quad A = -\exp(q), \tag{33}$$

let $K_{t,s} = \prod_{j=s+1}^{t} a_j$, with $K_{t,t} = 1$. Quantization changes the stored parameter by $e_q$ and the realized step by $e_{\Delta,t}$. Its exact transition-log error is

$$\zeta_t = \log(\widehat{a}_t/a_t) = A\left[\Delta_t(\exp(e_q) - 1) + e_{\Delta,t}\exp(e_q)\right]. \tag{34}$$

A cumulative relative envelope is

$$\Gamma = \max_{0\le s<t\le T} \left| \sum_{j=s+1}^{t} \zeta_j \right|, \qquad \Gamma \le \log(1+\tau). \tag{35}$$

Prefix minima and maxima compute $\Gamma$ in linear time.

**Theorem 9** (Relative kernel control)**.** *If $\tau \ge 0$ and the cumulative envelope holds, then $|\widehat{K}_{t,s} - K_{t,s}| \le \tau|K_{t,s}|$ on every interval. With a shared initial state and perturbed injections $\widehat{x}_s$,*

$$|\widehat{h}_t - h_t| \le \tau\left(|K_{t,0}h_0| + \sum_{s=1}^{t}|K_{t,s}x_s|\right) + (1+\tau)\sum_{s=1}^{t}|K_{t,s}|\,|\widehat{x}_s - x_s|. \tag{36}$$

*Proof.* The kernel ratio is $\exp(\sum_{j=s+1}^{t}\zeta_j)$ and lies in $[(1+\tau)^{-1}, 1+\tau]$. Its distance from one is at most $\tau$. Expanding both states by their kernels and adding and subtracting $\sum_s \widehat{K}_{t,s}x_s$ gives the state bound by the triangle inequality and $|\widehat{K}_{t,s}| \le (1+\tau)|K_{t,s}|$. □

The relative condition can reject small absolute transition errors. For example, log transitions $-180$ and $-80$ differ by $100$, although both transition magnitudes are tiny. This does not establish that the corresponding state error is negligible without bounding states and inputs. It identifies a mismatch between a relative kernel envelope and an absolute downstream target. A jointly quantized checkpoint must also recompute its own realized $\Delta_t$ trajectory: frozen single-bundle inputs cannot certify the complete model. No soft-penalty conclusion follows from this guard.