FIDELITY IS NOT SELECTION: LOW-BIT OPTIMIZER STATES ACROSS CODEBOOKS AND BIT WIDTHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantizing Adam's persistent moment states reduces optimizer memory, but representations with the same nominal bit width can train differently, making exhaustive comparison expensive. A cheap selector is the error a representation introduces into the next Adam update. However, this score reads a reference state once, whereas training repeatedly applies the codec to states it has already encoded; whether its ranking remains valid for selection is unclear. We test this prospectively by locking each choice before candidate training across codebooks, bit widths, training working points, and an external low-bit optimizer. At a common 130M condition across 12 independent training units, the error retained the same ordering of our three representations, yet its choice was best by point estimate under a uniform map in all 12 units, intermediate under a floating-point map in all 12, and never best under a dynamic-exponent map, where it was worst in 11. In one paired 4+4-bit condition the ordering persisted and the mismatch did not recur. Similar fidelity values also accompanied sharply different stability outcomes at larger working points. In the external optimizer, the error rule beat the published default in all six units at the longer budget but missed the lowest-loss setting in five. Thus, the failure mode is not an unstable fidelity ranking, but a stable one whose training meaning changes across settings.

1 INTRODUCTION

Adam-style optimizers (Kingma & Ba, 2015; Loshchilov & Hutter, 2019) maintain persistent first- and second-moment states for every parameter, making optimizer state a substantial component of training memory. This memory can be reduced by quantizing optimizer states (Dettmers et al., 2022; Li et al., 2023; Xi et al., 2025), alongside complementary approaches such as sharding and offloading (Rajbhandari et al., 2020; Ren et al., 2021). At a fixed bit width, however, multiple representations can have the same nominal storage cost while differing in their codebooks, scaling rules, and treatment of the stored moments. These choices can lead to different training outcomes. Determining which representation trains best by training every candidate to the target budget is therefore expensive, especially as the candidate set or training budget grows. This creates a practical selection problem: how can one choose among low-bit optimizer-state representations before committing to the target training budget?

A natural low-cost signal is the error that a candidate representation introduces into the next Adam update. Numerical fidelity of quantized optimizer states and updates has been used to analyze and compare low-bit optimizer designs (Dettmers et al., 2022; Li et al., 2023; Xi et al., 2025). Using such a score for prospective selection, however, requires a stronger property: its ranking must remain informative about eventual training outcomes. A one-step score measures a local perturbation of a reference optimizer state, whereas training repeatedly applies the codec to states shaped by earlier optimizer and codec transitions. We therefore ask whether a stable one-step fidelity ranking provides sufficient evidence for prospective selection when the codebook, precision, or training setting changes.

We evaluate this question by treating one-step Adam update error explicitly as a prospective selection rule. In each test, we fix the lowest-error candidate before candidate training and then train every candidate on matched units, allowing direct comparison between the selected candidate and its alternatives. We evaluate the rule across three codebooks, two bit widths, multiple training work-

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

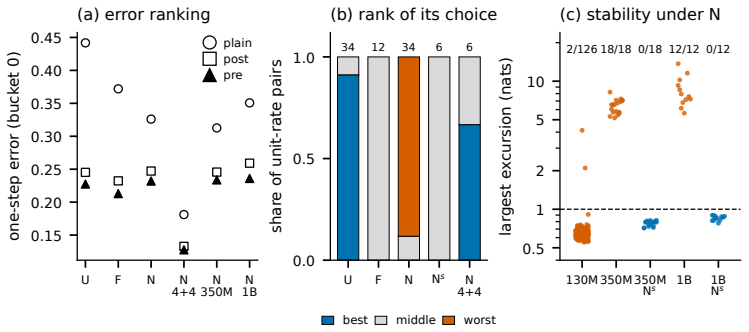

Figure 1: **Stable error rankings conceal changing training outcomes.** (a) Mean one-step error on AdamW reference states, 130M unless marked: pre $<$ post $<$ plain throughout. (b) Point-estimate training rank of the selected codec per qualified 130M pair; N^s denotes N with the signed-maximum scale. (c) Largest training-loss excursion of every N candidate, with the unstable count per group; the dashed line marks the 1-nat instability threshold. U/F/N: uniform, floating-point and dynamic-exponent first-moment maps; plain codes m, while post and pre code $m/(\sqrt{v}+\epsilon)$ formed with the quantized and exact v (§2).

ing points, and an unmodified external low-bit optimizer, and compare it with a fixed choice and a short-training selection rule. We distinguish numerical transfer, in which the error ordering is preserved, from decision transfer, in which the implications of that ordering for training quality and stability are preserved. Part of this ordering is expected from how each representation exposes the normalized Adam direction to quantization (§2), so its stability is not itself our finding; the question is whether such a reproducible ordering carries a reproducible training decision. Across our tests it did not: numerical transfer was substantially more consistent than decision transfer.

Three findings characterize this gap (Figure 1):

1. *A stable fidelity ranking can correspond to different training ranks of the selected codec.* At 3 bits, the one-step error ordered the three representations pre $<$ post $<$ plain in every qualified 130M condition (§3), and the same ordering persisted at the larger working points (§4). Yet the selected representation had different training ranks across codebooks: it was best by point estimate in 31 of 34 unit–learning-rate pairs under the uniform map, intermediate in all 12 pairs under the floating-point map, and never best in 34 pairs under the dynamic-exponent map. The same error ordering persisted at 4+4 bits, where this mismatch was not reproduced (§3).
2. *Similar one-step fidelity can coexist with sharply different training stability.* Under the dynamic-exponent map, the selected representation's one-step error changed little across working points, while every candidate became unstable in the two larger from-scratch tests (§4). Replacing absolute-maximum scaling with an existing signed-maximum rule (Orr et al., 2025) reduced the unstable count from 18 of 18 to 0 of 18 at one larger working point and from 12 of 12 to 0 of 12 at the other. At the smaller working point, however, the same scale change improved two representations and worsened the third (§5).
3. *Selection must be evaluated against the decision it replaces.* Among our codecs, neither the one-step rule nor selection based on a short training run established an advantage over a candidate chosen once on development units. In an external low-bit optimizer, the one-step rule beat the published default in all six units at the longer training budget, but missed the lowest-loss setting in five; a short pilot selected the lowest-loss setting in all six (§6).

2 Setting and Evaluation Protocol

Training units and controls. The main selection tests use a 130M-parameter LLaMA-style decoder (Touvron et al., 2023) trained for 4,000 steps (400 warmup steps, cosine decay, batch 32 of 256 tokens, weight decay 0.1, deterministic kernels) on a C4 token cache (Raffel et al., 2020), and evaluated on 512 C4 validation windows checked exhaustively to share no 64-token span with the

training cache. A run's *late loss* L is its mean held-out loss at steps 3,500, 3,540, 3,600, 3,700, 3,800, 3,900, and 4,000.

A *training unit* is a pair of model seed and data order, chosen by a rule fixed before any run: the smallest model seeds above those already used, with data order a fixed function of the seed. For each unit and learning rate, we first train an AdamW reference and two controls implementing the same update: an identity codec and a reordering of the same arithmetic. A unit is *invalid* if either control deviates from its reference by more than 0.0082 nats at any late point, a bound rounded up from the largest such deviation in the earlier runs. Let b_j denote the spread of the three controls' late losses. It enters the deterministic classification rule for the six unit outcomes; these classifications are not significance tests. Differences are classified against a practical threshold $\delta = 0.01$ nats fixed in advance: beyond $\delta + b_j$ they are *material*, within $\delta - b_j$ they are *within the threshold*, and otherwise *unresolved*; all boundaries are strict.

State codecs. Each codec stores 3-bit first and second moments in blocks of 128; the paired 4+4-bit test uses 4 bits for both moments (Appendix A.3). *Plain* codes the raw first moment, while *post* and *pre* code $m/(\sqrt{v} + \epsilon)$ using the quantized and exact second moment, respectively; both decode using the quantized second moment. Each update uses the decoded stored state and the new gradient, after which the codec encodes the state for the next step. Because post and pre code the normalized quantity from which the update is formed, whereas plain's error in m is divided coordinate-wise by $\sqrt{v}$ and grows where v is small, one-step update error is expected to rank plain last. Pre is likewise expected to precede post: it forms the normalized moment with the exact second moment and decodes it with the quantized one, so the second-moment coding error largely cancels in the next update, whereas post's remains.

The first moment uses one of three maps: a uniform signed grid (*U*), the dynamic-exponent map from the LPMM implementation of Li et al. (2023) (*N*), or the same library's floating-point map (*F*). At 3 bits, N has asymmetric endpoints -0.775 and $+1$, whereas U and F have symmetric endpoints; the exact codebooks are given in Appendix A.3. Under the original absolute-maximum rule, used by all three maps, each block is scaled by its largest absolute value. Section 5 additionally considers the signed-maximum scale (Orr et al., 2025), which instead uses the signed largest-magnitude entry. The second moment uses a logarithmic grid of the same bit width. The low-bit representations are simulated in fp32, so the bit widths are nominal and no realized memory saving is measured.

The update error. An AdamW reference run provides its exact state after update t and the clipped gradient of update $t + 1$. Let u be the bias-corrected Adam direction from the exact state and $\tilde{u}_c$ the corresponding direction after the state has passed through codec c. The direction excludes the learning rate and the decoupled weight-decay term, which are identical in the reference and codec updates. Over a coordinate set I, the error is $E_I(c) = \left(\sum_{i \in I} (\tilde{u}_{c,i} - u_i)^2 / \sum_{i \in I} u_i^2\right)^{1/2}$, the ratio of Euclidean norms, with sums accumulated over all tensors in double precision.

The primary coordinate set is *bucket 0*, where the gradient is at most twice its running root-mean-square, $|g|/(\sqrt{v} + \epsilon_t) \leq 2$ (96% of coordinates in the first recorded state); error over all scored coordinates (*global*) is a pre-declared auxiliary. The *error rule* selects the codec with the smallest bucket-0 error on the unit's own reference at the checkpoint-600 probe, which scores update 602 from the state after update 601 (Appendix A.4).

Evaluation axes. The tests ask four distinct questions: does the selected codec cost more than the simplest codec (the frozen endpoint $\Delta_j = L_j(\mathrm{sel}) - L_j(\mathrm{plain})$, §3); is it the best candidate (its rank and regret, the loss above the best candidate, §3, §6); does a costlier rule beat a fixed choice (the loss difference to a development-fixed candidate or default, §6); and does training remain stable (a rise of the training loss by more than 1 nat above its post-warmup running minimum, §5).

We say that the error transfers *numerically* between settings when it preserves the candidate ordering, and evaluate *decision transfer* through the latter three quantities. Numerical transfer is a property of the measurement; decision transfer is a property of the relation between that measurement and training outcomes.

3 One Error Ranking, Different Training Outcomes

Where the question came from. In earlier exploratory 130M runs, the one-step error gave the same codec ordering under the uniform map U and the dynamic-exponent map N, yet the lowest-error codec trained best under U and worst under N. Because this reversal motivated the prospective tests, we treat these earlier runs as exploratory evidence rather than independent confirmation (Appendix A.1). The tests below therefore fix each selection before observing the corresponding candidate outcomes.

Prospective tests. We apply the locked selection rule to two disjoint sets of six training units. Both sets are tested under U and N at 5×10^{-4}, 7×10^{-4}, and 10^{-3}, and under F at 7×10^{-4} (Appendix B). Each unit is first qualified by its controls and scored on its own AdamW reference; the selected codec is then fixed before any candidate is trained, after which every codec is trained on that unit. The primary endpoint is the selected codec's cost relative to plain, $\Delta_j = L_j(\mathrm{sel}) - L_j(\mathrm{plain})$, interpreted using the pre-specified δ and b_j rules of Section 2. All units qualified at the two lower learning rates; at 10^{-3}, two units in the second set failed qualification and their candidates were not trained.

The ranking transferred; the decision it implied did not. In every qualified unit and condition, at both probes and under both bucket-0 and global scoring, the error ordered the codecs pre $<$ post $<$ plain and therefore selected pre. Its training rank, however, depended strongly on the map. Under U, pre was best by point estimate in 31 of 34 qualified unit–learning-rate pairs; under F it was intermediate in all 12; under N it was never best in 34 pairs and was worst in 30. Accordingly, the frozen endpoint against plain ranged from repeatable material costs in some N conditions to below the preset cost under F (Table 1). The numerical ordering was reproducible, but the training meaning of its selected codec was not.

Table 1: Locked 130M selection outcomes by unit set, codebook, and learning rate. In every qualified condition, the step-600 bucket-0 error ordered pre<post<plain and selected *pre*. "Rank" reports the point-estimate training rank of the selected codec; $\Delta_j = L(\mathrm{sel}) - L(\mathrm{plain})$ is the frozen endpoint where specified. M/Unres./B count units classified as material loss, unresolved, and below the preset cost threshold. †The control-buffered secondary reading differs from the point-estimate summary: Set 1, U, 10^{-3} is beyond b_j in five of six units; Set 2, F, 7×10^{-4} in five of six; and Set 2, U, 10^{-3} in two of four qualified units. ‡Two set-2 units at 10^{-3} failed qualification; ranks and ranges use the remaining four. Per-unit outcomes are reported in Appendices B.1–B.3.

Condition	rank of selected pre	Δ_j range	M/Unres./B	frozen verdict
Set 1, N, 7×10^{-4}	worst 6	0.020–0.046	6/0/0	repeatable cost
Set 1, U, 7×10^{-4}	best 6	–	–	not an endpoint
Set 1, F, 7×10^{-4}	middle 6	-0.095 to -0.044	0/0/6	below preset cost
Set 1, N, 5×10^{-4}	worst 6	0.003–0.023	3/2/1	inconclusive
Set 1, U, 5×10^{-4}	best 6	–	–	secondary only
Set 1, N, 10^{-3}	worst 6	0.038–0.059	6/0/0	repeatable cost
Set 1, U, 10^{-3}	best 6†	–	–	secondary only
Set 2, N, 7×10^{-4}	middle 1, worst 5	-0.007 to 0.037	4/0/2	inconclusive
Set 2, U, 7×10^{-4}	best 6	–	–	secondary only
Set 2, F, 7×10^{-4}	middle 6†	-0.126 to -0.060	0/0/6	below preset cost
Set 2, N, 5×10^{-4}	middle 3, worst 3	-0.026 to 0.026	2/1/3	inconclusive
Set 2, U, 5×10^{-4}	best 6	–	–	secondary only
Set 2, N, 10^{-3}‡	worst 4	0.028–0.069	4/0/0	inconclusive
Set 2, U, 10^{-3}‡	best 1, middle 3†	–	–	secondary only

Across codebooks (post hoc). Expanding the candidate pool across codebooks did not rescue the error rule. F/pre had the lowest one-step error, yet F/post achieved lower late loss in all 12 units at 7×10^{-4}; F/pre remained 0.004–0.024 nats above the best of the nine candidates (Tables 6 and 7).

The same error ordering has different implications across bit widths. Repeating the set-1 test at 4+4 bits again gave pre $<$ post $<$ plain under both U and N, and the rule selected pre in every unit. Under N, however, the 3-bit mismatch was not reproduced: Δ_j was negative in all six units

(-0.021 to -0.008 nats), every unit was below the preset cost threshold, and pre was the lowest-loss codec or within control spread of it in every unit (Appendix B.4). Thus, the same numerical ordering that corresponded to a poor selection at 3 bits was compatible with a substantially different training outcome at 4+4 bits. The 4-bit post and pre codecs also finished below their AdamW references in every unit, an effect we did not investigate (Appendix B.4).

4 What Transfers with a Fidelity Ranking?

We next ask which properties of the fidelity ranking persist beyond the 130M selection tests, across working points, scoring inputs, and times of entry into training. We refer to the three model configurations as the 130M, 350M, and 1B working points; their exact parameter counts are 134.1M, 369.1M, and 1.35B, respectively.

The ranking transfers across working points. On AdamW reference states at the 130M, 350M, and 1B working points, the one-step error ordered the codecs pre $<$ post $<$ plain under all three maps in all 36 unit–probe–map cells at each working point. The error values themselves were also similar: the mean error of N/pre was 0.232, 0.234, and 0.236, respectively. Extending the reference-state replay to the next 8 or 32 coded updates did not change the selected codec in any cell. At $H = 128$, the earlier 130M probe selected plain instead of pre in four of six units, but this change did not recur at the 350M working point (Appendix C.2). These replays use the same reference gradient stream.

Training at the 350M working point. At the 350M working point, built from the 130M recipe by fixed rules (Appendix C.3), all six units qualified and the rule again selected pre. Under N, the frozen endpoint showed a *repeatable material cost*: pre was worse than plain beyond the preset margin in all six units ($+0.081$ to $+1.992$ nats) and was never the lowest-loss codec. Separately, a post-hoc stability reading found large training-loss excursions for every N candidate, ranging from 5.2 to 8.2 nats above its running minimum; none of the 36 U, reference, or control runs exceeded 0.83 nats (Table 11). Because every N candidate was unstable, we do not read this condition as a clean comparison of candidate quality. Its role is different: the reference-state fidelity ranking was unchanged while training under N moved from stable to unstable.

Training at the 1B working point. At the 1B working point, four codec candidates and two same-update controls per unit were trained from initialization. In every unit, at least one control exceeded the 0.0082 qualification bound inherited from the 130M tests. The frozen condition is therefore *inconclusive* and yields no selection verdict (Appendix C.4); the 1B runs enter only the fidelity-ranking and stability analyses. Descriptively, all 12 N candidates were unstable and no U candidate was, although two of the 18 codec-free runs briefly crossed the instability threshold and recovered.

Reading the evidence together. The fidelity ordering transferred more consistently than the training decision it was meant to support. Changing the scoring horizon (Appendix C.2), the state on which fidelity was measured (Appendix C.5.1), or the point at which the codec entered training (Appendix C.5.2) changed some selections, but none produced a decision rule that tracked training outcomes consistently across the tested settings.

5 A Second Blind Spot: Stability

Repeated encoding exposes behavior beyond one-step fidelity. Training applies a codec repeatedly to states that it has itself decoded. Under N with absolute-maximum scaling, this operation is not a fixed point: a block of equal negative entries $-a$ decodes to $-0.775a$, and encoding it again gives $-0.775^2 a$, because N has a $+1$ endpoint but no -1 endpoint. U and F have both endpoints. On stored states from the second-set runs at 7×10^{-4}, re-encoding produced a 3.5–6.2% change in update space under N (median by codec) and no change under U or F (Appendix D.3). The signed-maximum scale (Orr et al., 2025) instead maps the largest-magnitude entry to $+1$ regardless of its sign, so a second encoding leaves the state unchanged in exact arithmetic. Yet this scale lowered the first-encoding bucket-0 error by only 0.3–1.0% of its absolute-maximum value. Low-bit quantization is already known to alter the dynamics of stored moving averages (Xu et al., 2025).

Table 2: **Stability and late held-out loss at the 350M working point.** Each scale includes six paired training units with three representations per unit. Excursions are measured above each run's post-warm-up running minimum; reference gaps use the corresponding unit's AdamW run. Ranges are over runs.

Quantity	Absolute-maximum	Signed-maximum
Unstable or diverged runs	18 / 18	0 / 18
Largest excursion per run (nats)	5.168–8.219	0.714–0.820
Late held-out loss (nats)	3.915–7.395	3.790–3.824
Late loss minus own reference (nats)	+0.134–+3.616	+0.010–+0.043

Changing the scale rule removes the measured instability. Section 4 showed that similar one-step fidelity could accompany sharply different stability across working points. After the absolute-scale candidates had become unstable at the 350M working point, we prospectively tested the same three representations under N with signed-maximum scaling. All 18 absolute-scale candidates were unstable, whereas none of their 18 signed-scale counterparts were; the largest training-loss excursions fell from 5.17–8.22 to 0.71–0.82 nats (Table 2). The signed-scale runs also had lower late held-out loss, although they remained above their own AdamW references. A corresponding follow-up at the 1B working point reduced the unstable count from 12 of 12 to 0 of 12 (Appendix D.2). These paired interventions establish a substantial effect of the scale rule on the measured instability.

The same scale change does not uniformly improve training quality. We next tested the scale intervention prospectively at 130M. Two development units were used to choose signed-scale post as the representation to confirm and F/post as a strong fixed candidate of the same budget; six new units with their own controls and locked error-rule selections were then used for confirmation (Appendix D.3). Signed-maximum scaling lowered the late loss of post by 0.012–0.025 nats and of pre by 0.008–0.024 in all six confirmation units, with four material gains for each; both signed-versus-absolute verdicts were therefore inconclusive. In contrast, it raised the loss of plain by 0.003–0.067 in all six units, materially in four (Table 3). Under the signed scale, training ordered the representations post $<$ pre $<$ plain, while the one-step error still ranked them pre $<$ post $<$ plain; the selected codec therefore moved from worst to middle rather than becoming the best. Against F/post, signed-scale post showed no gain beyond the preset threshold.

Table 3: Six confirmation units under N at 7×10^{-4}. ^{s} denotes signed-maximum scaling. For a signed loss difference d_j, $^{\bullet}$ marks $|d_j| > \delta + b_j$ and $^{\circ}$ marks $|d_j| < \delta - b_j$; unmarked entries are unresolved. Regret is the loss of a rule's choice above the lowest-loss candidate. The final three columns are used in Section 6; complete candidate losses are reported in Table 12.

		signed minus absolute scale			post^{s}	regret of each rule's choice		
Unit	b_j	plain	post	pre	− post (F)	error: pre^{s}	1,000-step: plain	fixed: post (F)
(56, 14)	0.0020	$+0.0333^{\bullet}$	$-0.0158^{\bullet}$	$-0.0231^{\bullet}$	$+0.0091$	0.0313	0.0043	0.0000
(57, 15)	0.0039	$+0.0641^{\bullet}$	-0.0119	-0.0093	$+0.0124$	0.0384	0.0000	0.0045
(58, 16)	0.0064	$+0.0033^{\circ}$	-0.0162	$-0.0207^{\bullet}$	$+0.0029^{\circ}$	0.0230	0.0271	0.0000
(59, 17)	0.0063	$+0.0667^{\bullet}$	$-0.0168^{\bullet}$	-0.0075	$+0.0002^{\circ}$	0.0221	0.0000	0.0010
(60, 18)	0.0048	$+0.0610^{\bullet}$	$-0.0246^{\bullet}$	$-0.0237^{\bullet}$	-0.0001°	0.0217	0.0000	0.0043
(61, 19)	0.0034	$+0.0112$	$-0.0208^{\bullet}$	$-0.0204^{\bullet}$	$+0.0007^{\circ}$	0.0126	0.0106	0.0000

What the intervention establishes. The intervention establishes a clear limitation of one-step fidelity: training stability can change sharply while the local fidelity score changes little, and the effect of the scale rule on training quality depends on the representation. The 130M intervention was motivated by the registered hypothesis that repeated absolute-maximum encoding adds extra forgetting of the first moment that a one-step measurement cannot see. The diagnostic did not support this explanation: extra decay exceeded the pre-specified negligibility threshold only for plain (median 0.0025, versus 0.0008 for post and 0.0003 for pre), the representation that the signed scale made worse.

A second idempotent control further constrains the explanation. For post, its first-encoding bucket-0 error was higher than under absolute-maximum scaling (0.268 versus 0.247), yet it lowered late

loss by 0.012–0.028 nats in all six units, remaining within -0.005 to $+0.004$ nats of the signed scale. At the 350M working point, however, this idempotent control still left 3 of 6 runs unstable at 5×10^{-4}, while lowering the learning rate under absolute-maximum scaling left 2 of 6 unstable at 3×10^{-4} (Appendix D.1). Thus, neither lower first-encoding error nor idempotence alone explains the observed stability change. The intervention identifies a training-relevant effect beyond local fidelity, while the property of the scale rule that mediates this effect remains an open mechanistic question.

6 Evaluating Selection Against a Strong Fixed Choice

A selector must justify its extra cost. A more expensive selector can observe training behavior that one-step fidelity misses, but that information is useful only if it improves the decision over a strong fixed choice. On the six confirmation units, we compare three rules over seven candidates of the same budget: the three codecs under each scale and post under F. The *error rule* selects the candidate with the smallest bucket-0 error on the unit's own AdamW reference. The *fixed rule* uses post under F, chosen once for all confirmation units by its mean loss on two development units. The *1,000-step rule* briefly trains every candidate and keeps the fixed choice unless another candidate has lower held-out loss by more than the spread of the unit's controls. Its horizon was chosen from 600 and 1,000 steps on earlier second-set runs. All three rules were fixed before any confirmation candidate was trained. The corresponding prefix-equivalent cost of the 1,000-step rule is 3.5 A100-hours over the six confirmation units (Appendix E.1).

Short training improves on one-step fidelity, but does not establish an advantage over the fixed choice. The error rule selected signed-scale pre in all six units and never selected the lowest-loss candidate, with regret of 0.013–0.038 nats (Table 3). The 1,000-step rule instead selected absolute-scale plain in all six units; it was optimal in three and had regret up to 0.027 nats. It improved over the error rule by more than $\delta + b_j$ in four units. The fixed choice, however, had regret at most 0.0045 nats. Relative to that fixed choice, the 1,000-step rule was within the threshold in four units, unresolved in one, and materially worse in one; no unit showed a material gain. Anchored on the near-tied development alternative, signed-scale post, it was within the threshold in three units, a material gain in one, a material loss in one, and unresolved in one (post hoc). Thus, short training improved on one-step fidelity but did not establish value over a development choice. Admitting F/pre post hoc as an even lower-error candidate did not change this conclusion (Appendix D.3).

External optimizer: improvement over a default does not establish selection value. We test the same question in the unmodified SOLO optimizer (Xu et al., 2025), where the error rule selects among five quantile settings on six new 130M units at 4,000 and 40,000 training steps, with 40,000 steps as the primary budget. At the primary budget, the error rule beat the published default by 0.012–0.028 nats in all six units, but selected the lowest-loss setting in only one; a short pilot selected the lowest-loss setting in all six. The numerical ranking itself changed little across budgets, ordering the five settings identically in 11 of 12 unit–budget pairs, while the preferred training decision changed. The 4,000-step comparison is descriptive; there the error rule was mixed relative to the default (Appendix E.2). In a post-hoc target-budget development check, a single setting chosen from two development units matched the pilot's choice in all six confirmation units and beat the error rule by 0.015–0.017 nats in five. Thus, beating the published default shows that the error signal can be useful, but does not by itself establish the value of per-unit selection over a strong fixed choice.

Decision unit. In practice a representation is chosen once per workload rather than once per seed; our matched units estimate how robust that choice is. Because the one-step ordering was identical across units, scoring at the workload level would induce the same choice as the per-unit rule.

7 Related Work

Memory-efficient optimizers. Optimizer-state memory can be reduced in several ways. Adam's state can be factored (Shazeer & Stern, 2018), quantized (Dettmers et al., 2022; Li et al., 2023; Xi et al., 2025), or replaced in part by low-rank gradient statistics (Zhao et al., 2024; Hao et al., 2024; Zhang et al., 2025b; He et al., 2025; Xiao et al., 2025; Rajabi et al., 2025). Other methods use error

feedback (Robert et al., 2025; Modoranu et al., 2024), rescale low-rank updates toward full-rank behavior (Chen et al., 2025; Zhu et al., 2025), share or reduce optimizer state (Zhang et al., 2025a; Zmushko et al., 2025), adapt the retained rank (Refael et al., 2025), or simplify second-moment state selectively (Kalra et al., 2025). These methods primarily address how to reduce the memory or state cost of optimization. Our problem is orthogonal: once several low-bit representations have the same nominal precision and storage budget, their training outcomes can still differ, leaving the question of how to choose among them without training every candidate to the target budget.

Fidelity measures for quantized optimizer states. Numerical fidelity is already used to analyze and guide low-bit optimizer design. Dettmers et al. (2022) study errors between quantized and full-precision Adam updates, show that low mean quantization error alone is insufficient, and validate quantization components through training ablations. Li et al. (2023) use quantized-moment distributions to motivate smaller blocks and a zero-free second-moment map and confirm their choices in training. Xi et al. (2025) compare FP8 formats using mean-squared error in the normalized Adam moment $m/(\sqrt{v}+\epsilon)$. For low-bit Muon, Su et al. (2026) instead optimize directional fidelity of the quantized momentum, motivated by a perturbation bound on the polar update. These studies establish fidelity as a useful diagnostic or design signal, and prior work already shows that lower numerical error need not imply better training. What remains distinct in our setting is the stronger decision question: whether a fidelity ranking that remains numerically stable can itself be used prospectively to choose among candidates before they train. We test that use directly by locking the lowest-error choice before candidate training, training every candidate, and measuring the consequence of the resulting decision across codebooks, precisions, and training settings.

Dynamics of low-bit optimizer states. Quantized optimizer states have recurrent dynamics that are not captured by an instantaneous reconstruction error alone. Xu et al. (2025) show that low-bit quantization changes the dynamics of exponential moving averages and propose logarithmic quantization and precision-specific momentum settings for two- and three-bit states. Topollai & Choromanska (2026) analyze re-quantized moving averages through state staleness and propose state resets. The signed-maximum scale introduced by Orr et al. (2025) provides the scale intervention used in Section 5. These works establish that state evolution under repeated quantization can matter for optimization. They leave open a different question relevant to selection: whether training-relevant recurrent behavior can change substantially while the one-step fidelity ranking used to make the selection remains nearly unchanged. Our experiments show that such a separation can occur, while the specific property of the scale rule mediating the observed stability change remains an open mechanistic question.

Optimizer evaluation and decision baselines. Optimizer comparisons are sensitive to tuning, training budget, scale, and experimental control. Careful tuning can substantially narrow apparent differences between optimizers (Schmidt et al., 2021; Zhao et al., 2025), reported gains can change with model scale (Wen et al., 2026), and the preferred optimizer can depend on model size, batch size, and training duration (Semenov et al., 2025). Well-tuned full-rank baselines can also change conclusions about memory-efficient methods (Glentis et al., 2025), while Kaddour et al. (2023) show that apparent gains of efficient-training methods can disappear when the baseline is matched to the training budget. Training stability itself depends on choices such as learning rate and warmup (Wortsman et al., 2024; Kalra & Barkeshli, 2024), and numerical or geometric optimizer properties need not identify where training progress originates (Song et al., 2025). More generally, single runs can hide substantial variation (Bouthillier et al., 2021), uncontrolled factors can reverse benchmark conclusions (Mytkowicz et al., 2009), and time-to-result benchmarks control workloads and hardware for this reason (Dahl et al., 2023). These studies motivate strong experimental baselines, but do not directly answer whether an additional per-unit selection procedure is worthwhile relative to choosing a strong configuration once. We therefore evaluate selectors not only against cheaper rules or published defaults, but also against a configuration chosen from development data.

8 Discussion

Stable fidelity is not stable decision evidence. Across our tests, the one-step fidelity ordering was substantially more reproducible than the training decision it implied. On AdamW reference states, the representations repeatedly ordered as pre $<$ post $<$ plain, yet the selected codec moved

among best, intermediate, and worst point-estimate training ranks as the codebook changed, and the 3-bit mismatch was not reproduced at 4+4 bits. Across working points, similar one-step errors also accompanied sharply different stability outcomes. These results do not make fidelity uninformative: it remains a direct measure of local update distortion. They instead separate two questions that are easy to conflate. Numerical fidelity characterizes a local perturbation of a reference optimizer state; selection requires that the ordering remain informative about the consequences of repeatedly using the codec during training. Reproducibility of the former therefore does not by itself establish validity of the latter.

Implications for evaluating selection. The experiments support three principles for using fidelity as a selection signal.

1. *Reproducing a ranking is not validating a selection.* A fidelity ordering should be evaluated through the training outcome of the choice it induces in the target setting. In our tests, the same numerical ordering corresponded to different training ranks across codebooks and bit widths (§3).
2. *A fidelity ranking cannot certify its own transfer.* The ordering remained stable across working points even when its implications for training quality and stability changed. The scale intervention further showed that measured stability can change sharply while first-encoding fidelity changes little (§4, §5).
3. *Selection should be compared with the decision it replaces.* Additional per-unit information can improve on one-step fidelity, but this does not establish the value of selection unless it also improves on a strong fixed alternative. In our codec tests, short training did not establish an advantage over a candidate chosen prospectively from development data; in the external optimizer, beating the published default did not imply selection of the lowest-loss setting (§6).

Scope and open questions. The evidence is strongest for the tested LLaMA-style decoder and data pipeline, the three representation families and codebooks studied here, and the external SOLO configuration. Every internal working point trains well below compute-optimal budgets, and the external optimizer shows that the preferred setting can change with the budget; our claims therefore concern selection at the tested target budgets, not budget-invariant codec quality. The 130M, 350M, and 1B conditions are distinct working points rather than a controlled scaling experiment: model size, schedule, data, and learning rate change together, so our claims concern transfer across these settings rather than an isolated effect of parameter count. At the 1B working point, the fidelity ranking remains measurable, but the formal selection test yields no verdict because the inherited control-qualification rule fails; its stability results are therefore descriptive. The scale intervention establishes a substantial effect of the scale rule on stability, but the property mediating that effect remains an open mechanistic question. Finally, our codecs simulate low-bit representations in fp32, so the experiments test selection validity rather than realized memory savings. Locked, paired, descriptive, and post-hoc analyses retain their separate evidence status throughout rather than being pooled into a single inferential claim (Appendix A).

9 Conclusion

Selecting among low-bit optimizer-state representations requires more than a reproducible numerical ranking. Across codebooks, bit widths, and training working points, the same one-step fidelity ordering accompanied different training ranks and stability outcomes. The scale intervention further showed that training-relevant behavior can change in ways not captured by first-encoding fidelity alone. In our codec tests, richer per-unit selection improved on one-step fidelity but did not establish an advantage over a strong choice fixed from development data; in an external optimizer, the error rule improved over the published default while still missing the lowest-loss setting in most units. One-step fidelity is therefore useful as a design signal, but its use for selection should be validated as a decision in the target setting and against the decision policy it is intended to replace.

AI USE STATEMENT

Large language models were used to assist in writing and editing this paper. The authors prompted the model with text from the paper and received suggestions for rewording, grammar, and clarity. The authors reviewed and edited the suggestions before incorporating them into the final text. The use of AI was limited to language assistance; all scientific content, analysis, and conclusions were generated by the authors.

REPRODUCIBILITY STATEMENT

Appendices A–E describe the evidence status, exact codecs and protocols, per-unit outcomes, transfer tests, scale interventions, and selection-rule evaluations. Appendix F reports independent implementations sharing no code with the read-outs that recomputed every verdict and every locked error. The training, scoring, and read-out code is provided as anonymized supplementary material.

REFERENCES

Xavier Bouthillier, Pierre Delaunay, Mirko Bronzi, Assya Trofimov, Brennan Nichyporuk, Justin Szeto, Nazanin Mohammadi Sepahvand, Edward Raff, Kanika Madan, Vikram Voleti, Samira Ebrahimi Kahou, Vincent Michalski, Tal Arbel, Chris Pal, Gael Varoquaux, and Pascal Vincent. Accounting for variance in machine learning benchmarks. In A. Smola, A. Dimakis, and I. Stoica (eds.), *Proceedings of Machine Learning and Systems*, volume 3, pp. 747–769, 2021. URL https://proceedings.mlsys.org/paper_files/paper/2021/file/0184b0cd3cfb185989f858a1d9f5c1eb-Paper.pdf.

Xi Chen, Kaituo Feng, Changsheng Li, Xunhao Lai, Xiangyu Yue, Ye Yuan, and Guoren Wang. Fira: Can we achieve full-rank training of LLMs under low-rank constraint? In *Advances in Neural Information Processing Systems 38 (NeurIPS)*, 2025.

George E. Dahl, Frank Schneider, Zachary Nado, Naman Agarwal, Chandramouli Shama Sastry, Philipp Hennig, Sourabh Medapati, Runa Eschenhagen, Priya Kasimbeg, Daniel Suo, Juhan Bae, Justin Gilmer, Abel L. Peirson, Bilal Khan, Rohan Anil, Mike Rabbat, Shankar Krishnan, Daniel Snider, Ehsan Amid, Kongtao Chen, Chris J. Maddison, Rakshith Vasudev, Michal Badura, Ankush Garg, and Peter Mattson. Benchmarking neural network training algorithms. *arXiv preprint arXiv:2306.07179*, 2023.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 8-bit optimizers via block-wise quantization. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=shpkpVXzo3h.

Athanasios Glentis, Jiaxiang Li, Qiulin Shang, Andi Han, Ioannis Tsaknakis, Quan Wei, and Mingyi Hong. Scalable parameter and memory efficient pretraining for LLM: Recent algorithmic advances and benchmarking. *arXiv preprint arXiv:2505.22922*, 2025.

Yongchang Hao, Yanshuai Cao, and Lili Mou. Flora: Low-rank adapters are secretly gradient compressors. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

Yutong He, Pengrui Li, Yipeng Hu, Chuyan Chen, and Kun Yuan. Subspace optimization for large language models with convergence guarantees. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Jean Kaddour, Oscar Key, Piotr Nawrot, Pasquale Minervini, and Matt J. Kusner. No train no gain: Revisiting efficient training algorithms for transformer-based language models. In *Advances in Neural Information Processing Systems 36 (NeurIPS)*, 2023.

Dayal Singh Kalra and Maissam Barkeshli. Why warmup the learning rate? Underlying mechanisms and improvements. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Dayal Singh Kalra, John Kirchenbauer, Maissam Barkeshli, and Tom Goldstein. When can you get away with low memory Adam? *arXiv preprint arXiv:2503.01843*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1412.6980.

Bingrui Li, Jianfei Chen, and Jun Zhu. Memory efficient optimizers with 4-bit states. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/3122aaa22b2fe83f9cead1a696f65ceb-Abstract-Conference.html.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Ionuţ-Vlad Modoranu, Mher Safaryan, Grigory Malinovsky, Eldar Kurtic, Thomas Robert, Peter Richtárik, and Dan Alistarh. MicroAdam: Accurate adaptive optimization with low space overhead and provable convergence. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

Todd Mytkowicz, Amer Diwan, Matthias Hauswirth, and Peter F. Sweeney. Producing wrong data without doing anything obviously wrong! In *Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 265–276, 2009. doi: 10.1145/1508244.1508275.

Douglas Orr, Luka Ribar, and Carlo Luschi. Optimal formats for weight quantisation. *arXiv preprint arXiv:2505.12988*, 2025.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020.

Sahar Rajabi, Nayeema Nonta, and Sirisha Rambhatla. Subtrack++ : Gradient subspace tracking for scalable LLM training. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=6geRIdlFWJ.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. ZeRO: Memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2020.

Yehonathan Refael, Jonathan Svirsky, Boris Shustin, Wasim Huleihel, and Ofir Lindenbaum. AdaRankGrad: Adaptive gradient rank and moments for memory-efficient LLMs training and fine-tuning. In *13th International Conference on Learning Representations (ICLR)*, 2025.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. ZeRO-Offload: Democratizing Billion-Scale model training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pp. 551–564. USENIX Association, July 2021. ISBN 978-1-939133-23-6. URL https://www.usenix.org/conference/atc21/presentation/ren-jie.

Thomas Robert, Mher Safaryan, Ionut-Vlad Modoranu, and Dan Alistarh. LDAdam: Adaptive optimization from low-dimensional gradient statistics. In *13th International Conference on Learning Representations (ICLR)*, 2025.

Robin M Schmidt, Frank Schneider, and Philipp Hennig. Descending through a crowded valley - benchmarking deep learning optimizers. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, volume 139 of *Proceedings of Machine Learning Research*, pp. 9367–9376, 2021.

Andrei Semenov, Matteo Pagliardini, and Martin Jaggi. Benchmarking optimizers for large language model pretraining. *arXiv preprint arXiv:2509.01440*, 2025.

Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, volume 80 of *Proceedings of Machine Learning Research*, pp. 4596–4604, 2018.

Minhak Song, Kwangjun Ahn, and Chulhee Yun. Does SGD really happen in tiny subspaces? In *13th International Conference on Learning Representations (ICLR)*, 2025.

Yupeng Su, Ruijie Zhang, Ziyue Liu, Yequan Zhao, and Zheng Zhang. Muonq: Enhancing low-bit muon quantization via directional fidelity optimization. In *Conference on Language Modeling (CoLM)*, 2026. URL https://arxiv.org/abs/2605.11396.

Kristi Topollai and Anna Choromanska. Understanding quantization of optimizer states in LLM pre-training: Dynamics of state staleness and effectiveness of state resets. *arXiv preprint arXiv:2603.16731*, 2026.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Kaiyue Wen, David Hall, Tengyu Ma, and Percy Liang. Fantastic pretraining optimizers and where to find them. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 144731–144838, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/ea4f0a6ad7088c9b93f22d74c0c9e8c2-Paper-Conference.pdf.

Mitchell Wortsman, Peter Liu, Lechao Xiao, Katie Everett, Alexander Alemi, Ben Adlam, JD Co-Reyes, Izzeddin Gur, Abhishek Kumar, Roman Novak, Jeffrey Pennington, Jascha Sohl-Dickstein, Kelvin Xu, Jaehoon Lee, Justin Gilmer, and Simon Kornblith. Small-scale proxies for large-scale transformer training instabilities. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 49844–49869, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/d848cb2c84f0bba7f1f73cf232734c40-Paper-Conference.pdf.

Haocheng Xi, Han Cai, Ligeng Zhu, Yao Lu, Kurt Keutzer, Jianfei Chen, and Song Han. COAT: Compressing optimizer states and activations for memory-efficient FP8 training. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 42989–43011, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/6ac807c9b296964409b277369e55621a-Paper-Conference.pdf.

Jinqi Xiao, Shen Sang, Tiancheng Zhi, Jing Liu, Qing Yan, Linjie Luo, and Bo Yuan. COAP: memory-efficient training with correlation-aware gradient projection. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2025, Nashville, TN, USA, June 11-15, 2025*, pp. 30116–30126. Computer Vision Foundation / IEEE, 2025. doi: 10.1109/CVPR52734.2025.02803. URL https://openaccess.thecvf.com/content/CVPR2025/html/Xiao_COAP_Memory-Efficient_Training_with_Correlation-Aware_Gradient_Projection_CVPR_2025_paper.html.

Cong Xu, Wenbin Liang, Mo Yu, Anan Liu, Ke-Yue Zhang, Shunli Wang, Lizhuang Ma, Jianyong Wang, Jun Wang, and Wei Zhang. Pushing the limits of low-bit optimizers: A focus on EMA dynamics. *arXiv preprint arXiv:2505.00347*, 2025.

Yushun Zhang, Congliang Chen, Ziniu Li, Tian Ding, Chenwei Wu, Diederik (Durk) Kingma, Yinyu Ye, Zhi-Quan Luo, and Ruoyu Sun. Adam-mini: Use fewer learning rates to gain more. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28033–28063, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/45ae878717399e6f62d57c65f052cd46-Paper-Conference.pdf.

Zhenyu Zhang, Ajay Kumar Jaiswal, Lu Yin, Shiwei Liu, Jiawei Zhao, Yuandong Tian, and Zhangyang Wang. Q-GaLore: Quantized GaLore with INT4 projection and layer-adaptive low-rank gradients. In *Proceedings of the 2nd Conference on Parsimony and Learning (CPAL)*, volume 280 of *Proceedings of Machine Learning Research*, pp. 1035–1050, 2025b.

Jiawei Zhao, Zhenyu Zhang, Beidi Chen, Zhangyang Wang, Anima Anandkumar, and Yuandong Tian. GaLore: Memory-efficient LLM training by gradient low-rank projection. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. Oral.

Rosie Zhao, Depen Morwani, David Brandfonbrener, Nikhil Vyas, and Sham Kakade. Deconstructing what makes a good optimizer for autoregressive language models. In *13th International Conference on Learning Representations (ICLR)*, 2025.

Hanqing Zhu, Zhenyu Zhang, Wenyan Cong, Xi Liu, Sem Park, Vikas Chandra, Bo Long, David Z. Pan, Zhangyang Wang, and Jinwon Lee. Apollo: Sgd-like memory, adamw-level performance. In M. Zaharia, G. Joshi, and Y. Lin (eds.), *Proceedings of Machine Learning and Systems*, volume 7. MLSys, 2025. URL https://proceedings.mlsys.org/paper_files/paper/2025/file/437bc4ccafd3fc6d4289bd10940be42b-Paper-Conference.pdf.

Philip Zmushko, Aleksandr Beznosikov, Martin Takáč, and Samuel Horváth. FRUGAL: Memory-efficient optimization by reducing state overhead for scalable training. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, volume 267 of *Proceedings of Machine Learning Research*, pp. 80708–80739, 2025.

A Protocol, Evidence Status, and Codec Definitions

Table 4 records the evidence status and frozen readout of the experiments and auxiliary analyses used in the paper. Evidence classes are kept separate rather than pooled into a single verdict.

A.1 The recorded codec study

The question of Section 3 came from a codec study in earlier, unpublished work on this codebase; it motivated the locked tests and is not independent confirmation. Its errors were computed once, on one recorded reference state (seed 42, step 600), and its codecs were trained in two seed cells at each of the three learning rates; Section 3 summarizes the reversal it showed, which motivated the prospective tests.

A.2 Units, recipe and runs

A unit is a (model seed, data order) pair; each new set takes the six smallest unused model seeds, with data order equal to the seed minus 42: $(44, 2)$ to $(49, 7)$ (set 1), $(50, 8)$ to $(55, 13)$ (set 2) and $(56, 14)$ to $(61, 19)$ (Appendix D.3). Every 130M run of our own codecs uses one recipe (the external-optimizer test differs in β, Appendix E.2): 12 layers, width 768, 12 heads, vocabulary 32,000, sequence length 256, batch 32, 4,000 steps with 400 warmup steps and cosine decay, $\beta = (0.9, 0.95)$, $\epsilon = 10^{-8}$, weight decay 0.1, clipping at 1.0 and deterministic kernels. Held-out loss is evaluated on 512 windows; the late loss L is its mean over the seven points 3,500, 3,540 and 3,600 to 4,000 in steps of 100. Each unit and learning rate has an AdamW reference, an identity control (the codec path with exact moments) and an op-order control (AdamW with a different arithmetic

Table 4: Evidence registry for the experiments and analyses reported in the paper. *Units*: training units, each a (model seed, data order) pair; a test that reuses an earlier test's units is paired with it. *Locked* means the protocol, endpoint, and readout rule were committed before the corresponding test runs; *paired extension* is a locked condition on previously established units; *post hoc* denotes an analysis defined after the relevant outcomes were known. Verdicts are read from the frozen readout files; M/Unres./B counts material loss, unresolved, and below-threshold units.

Test	Units	Evidence	Frozen verdict	Where
Replication (set 1, 7×10^{-4}; N, with U descriptive)	6, new	locked	N: repeatable material cost (M/Unres./B 6/0/0); under U the choice trained best in all six	B.1
Set-1 extension (5×10^{-4}, 10^{-3} under U, N; F at 7×10^{-4})	6, set 1	paired extension	N 5×10^{-4}: inconclusive (3/2/1); N 10^{-3}: repeatable material cost (6/0/0); F: below preset cost (0/0/6), the choice middle in all six	B.2
Set-2 test (7×10^{-4} under U, N, F; 5×10^{-4} under U, N)	6, new	locked	N 7×10^{-4}: inconclusive (4/0/2); F: below preset cost (0/0/6); N 5×10^{-4}: inconclusive (2/1/3)	B.3
Set 2 at 10^{-3}	6, set 2	paired extension	inconclusive: two units failed qualification; the four qualified all material	B.3
4+4 bits (set 1, 7×10^{-4})	6, set 1	paired extension	below preset cost (0/0/6)	B.4
130M scale intervention	6 new; development on two set-2 units	locked; one control added after development	registered explanation (extra forgetting) not supported; primary endpoint: no gain beyond the threshold; second: inconclusive	D.3
130M selector comparison	6 new; development on two set-2 units	locked	the 1,000-step rule showed no material gain over the fixed candidate (four units within threshold, one unresolved, one material loss)	E.1
F/pre on the confirmation units	6, the confirmation units	post hoc, after the read-out	lowest error in all six, so the rule would have chosen it; regret 0.010 to 0.020; above post under F in every unit (material in two)	D.3
Fixed-choice sensitivity	6, the confirmation units	post hoc, after the read-out	with signed-scale post, the near-tied development alternative, as the fixed choice, the 1,000-step rule was within the threshold in three units, a material gain in one, a material loss in one, and unresolved in one	§6
External optimizer, development-fixed choice at 40,000 steps	2 development units	post hoc, after the read-out	$p = 0.5$ lowest at 40,000 steps; on the confirmation units it equals the pilot's choice in all six and beats the error's choice by 0.015 to 0.017 in five (mixed); the 4,000-step choice costs 0.028 to 0.037 (consistent material cost)	E.2
350M selection	6, new	locked	N: repeatable material cost (6/0/0); every N candidate had a training-loss excursion (post hoc)	C.3
350M scale intervention	6 (18 runs)	locked, after the instability was seen	prediction met: 0 of 18 signed-scale runs unstable, 18 of 18 twins	D.1
350M learning rate and idempotence	6 (12 runs), paired	locked, after the intervention	both partial: absolute scale at 3×10^{-4} unstable in 2 of 6 (6 of 6 at 5×10^{-4}); idempotent control at 5×10^{-4} unstable in 3 of 6, predicted stable	D.1
Error at 350M and 1B	6 per size	locked	ranking invariant at 130M, 350M and 1B	C.2
1B selection	6, new	qualification failed	inconclusive: every unit failed the inherited 0.0082 bound; no selection verdict	C.4
1B scale intervention	6 (12 runs)	locked	prediction met: 0 of 12 signed-scale runs unstable, 12 of 12 twins	D.2
External optimizer (SOLO)	6, new ($\times$ 2 budgets)	locked	40,000 steps: consistent material benefit; 4,000 steps: mixed	E.2
Own-trajectory scoring	6, set 2 checkpoints	post hoc; rules committed before any tensor was read	selection changes, not toward training (primary grid and 13 of 19 auxiliary grids); trajectory invariant in 6	C.5.1
Role-weighted cross-scale error prediction	12 (130M)	locked	non-discriminating in 24 of 24 cells: the predicted move was smaller than the units' spread	stated in full in this row
Checkpoint forks (130M, 350M)	6 + 6	locked	130M known answer not reproduced at either fork step (15%: inconclusive; 50%: below preset cost), so the 1B forks were not run; 350M: at 15% not comparable (every N fork unstable), at 50% inconclusive; U below preset cost at both steps and sizes	C.5.2

order). On both sets at 7×10^{-4}, the LPMM library's own optimizer, unmodified and configured for 3-bit moments, is a public reference outside the selection set.

A.3 CODEBOOKS, SECOND-MOMENT GRID AND SCALE RULES

Every codec of ours stores each moment tensor in blocks of 128 consecutive elements (the last block zero-padded), with one code per element and per-block fp32 metadata: one scale for the first moment, and the grid's $\ell_{\min}$ and Δ (below) for the second; the bit widths are nominal, since every moment is kept as an fp32 tensor during training. The values below are read from the optimizer module and match the hash each training run recorded.

First moment. Under the original, absolute-maximum rule a block is scaled by $a = \max_i |x_i|$ and each entry is coded to the level nearest to x_i/a; an exact tie takes the lower index, that is, the more negative level; a block of zeros decodes to zero; the decoded value is a times the level. The maps are:

- **N**, the library's dynamic-exponent map at the pinned revision (its signed map with $b-1$ exponent bits and b bits in total). At 3 bits its 8 levels are $\{-0.775, -0.325, -0.055, 0, 0.055, 0.325, 0.775, 1\}$; at 4 bits its 16 levels are $\{-0.8875, -0.6625, -0.4375, -0.2125, -0.0775, -0.0325, -0.0055, 0, 0.0055, 0.0325, 0.0775, 0.2125, 0.4375, 0.6625, 0.8875, 1\}$. The negative end is -0.775 (-0.8875 at 4 bits) while the positive end is 1, which is what makes the absolute-maximum rule non-idempotent (Section 5).
- **F**, the same library's floating-point map at 3 bits (one sign bit, two exponent bits): the codes $\{-1, -0.5, -0.25, 0, -0, 0.25, 0.5, 1\}$, where the two zero codes decode alike and a tie between them takes $+0$. F was trained at 3 bits only.
- **U**, a uniform signed grid with $m_{\max} = 2^{b-1} - 1$ levels on each side of zero: the levels $k/m_{\max}$, $k = -m_{\max}, \ldots, m_{\max}$, so 7 levels ($k/3$) at 3 bits and 15 ($k/7$) at 4 bits, using $2^b - 1$ of the 2^b codes. The code is $x_i m_{\max}/a$ rounded to the nearest integer, with midpoints rounded toward $+\infty$, then clamped to $[-m_{\max}, m_{\max}]$.

The signed-maximum scale of Section 5 replaces a by $s = x_{\hat{\imath}}$, with $\hat{\imath}$ the first index attaining $\max_i |x_i|$, and codes x_i/s with the same nearest-level rule, so the block's extreme entry decodes to itself whatever its sign. The idempotent control scale of Appendix D.3 uses $a/0.775$ when the block's most negative entry equals $-a$ and a otherwise.

Second moment. Every codec, at both bit widths, codes the second moment on a per-block logarithmic grid: with $\ell_i = \log_{10} v_i$ over the block's valid lanes (v floored at 10^{-30}), the 2^b levels are $10^{\ell_{\min} + k\,\Delta}$, $k = 0, \ldots, 2^b - 1$, $\Delta = (\ell_{\max} - \ell_{\min})/(2^b - 1)$ floored at 10^{-7}: 8 levels at 3 bits and 16 at 4 bits, with the block's smallest and largest values on the grid. Each entry takes the lower or the upper neighbouring level, whichever brings $1/(\sqrt{v_{\text{level}}/c_2} + \epsilon)$ closer to $1/(\sqrt{v/c_2} + \epsilon)$, with $c_2 = 1 - \beta_2^t$ the bias correction.

Order within a step. From the decoded stored state $(\hat{m}, \hat{v})$ and the new clipped gradient g the optimizer forms the raw moments $m = \beta_1 \hat{m} + (1 - \beta_1) g$ and $v = \beta_2 \hat{v} + (1 - \beta_2) g^2$ and applies the bias-corrected AdamW update from them. It then codes v on its grid, giving v_q; forms the first-moment target, m for plain, $m/(\sqrt{v_q} + \epsilon_t)$ for post and $m/(\sqrt{v} + \epsilon_t)$ for pre, with $\epsilon_t = \epsilon\sqrt{c_2}$; codes the target with the map; and stores the decoded target (plain) or the decoded target times $\sqrt{v_q} + \epsilon_t$ (post, pre) as $\hat{m}$, and v_q as $\hat{v}$.

A.4 SCORING INPUTS AND SELECTION

From each reference run's checkpoint after update 600, the recorder saved the exact Adam state after update 601 and the clipped gradients of updates 601 and 602 (the checkpoint-600 probe; the checkpoint-2,000 probe is the same with updates 2,001 and 2,002). The reference direction is the bias-corrected Adam direction of update 602 from that state; a candidate's direction repeats it after encoding and decoding the state with the codec, with $\epsilon_t = \epsilon\sqrt{1 - \beta_2^{601}}$ in the normalized moment. Bucket 0 holds the coordinates with $|g|/(\sqrt{v} + \epsilon_t) \le 2$; over it (or over all coordinates, for the global error), squared errors and squared reference directions are summed over all tensors in double precision, and the error is the square root of their ratio. Selection takes the smallest error per codebook, with ties broken in the order plain, post, pre (none occurred).

A.5 Qualification, decision rule, secondary reading and locks

A unit is invalid if its identity or op-order control differs from its AdamW reference by more than 0.0082 nats at any of the seven late points, the largest such deviation in the recorded study (0.00815) rounded up; a diverged or missing outcome would also make it invalid (none occurred). b_j is the maximum minus the minimum of the three controls' late losses. Unit states and the six-unit verdict are as in Section 3, with strict boundaries. $\delta = 0.01$ nats, about 1% in perplexity, was fixed with the comparator and the tie rule before the first locked test.

After the replication, a pre-declared secondary reading asks whether the selected codec was the best of the three: with Γ_j its loss minus the lowest loss of the other two, a unit is *selected best* if $\Gamma_j < -b_j$ and *selected not best* if $\Gamma_j > b_j$. The same classification rules are used throughout.

Each unit's selections were committed before its candidates were submitted; from the set-1 extension on, a second implementation recomputed them before the commit (Appendix F).

B Locked Selection Across Codebooks and Bit Widths

Four 3+3-bit protocols and one 4+4-bit protocol applied this rule at 130M, each committed before any of its own runs (Table 4); one that reuses earlier units is a paired condition, not an independent replication.

Reading the per-unit tables. Tables 5, 6, 7 and 8 have one row per unit and condition. *Dev.* is the largest late-point deviation of a same-math control from the AdamW reference (bound 0.0082), b_j the control spread, and the loss columns the late loss L of each codec in the primary map. $\Delta_j = L(\mathrm{sel}) - L(\mathrm{plain})$ is the frozen endpoint and *state* its classification. Γ_j is the selected codec's loss minus the lowest of the other two, *best* if $\Gamma_j < -b_j$ and *n.b.* (not best) if $\Gamma_j > b_j$; Γ_j^U is the same under U.

B.1 The locked replication (set 1, 7×10^{-4})

All six units qualified and every unit showed a material loss; under U, a descriptive reading here, the selection had the lowest loss in every unit. The error order plain $>$ post $>$ pre held in all 24 combinations of unit, step and aggregation.

Table 5: Per-unit outcomes of the replication (set 1, 7×10^{-4}); the primary map is N. The locked rule selected *pre* under both codebooks in every unit. The replication declared no secondary reading, so Γ_j and Γ_j^U are descriptive. LPMM: the library's native 3-bit optimizer, a public reference outside the selection set.

			L, map N			N: endpoint		descriptive		
Unit	dev.	b_j	plain	post	pre	Δ_j	state	Γ_j	Γ_j^U	LPMM
(44, 2)	0.0017	0.0023	4.2465	4.2665	4.2889	$+0.0424$	material	$+0.0424$	-0.0161	4.2661
(45, 3)	0.0014	0.0013	4.2555	4.2704	4.2758	$+0.0203$	material	$+0.0203$	-0.0151	4.2768
(46, 4)	0.0030	0.0028	4.2505	4.2723	4.2961	$+0.0455$	material	$+0.0455$	-0.0207	4.2850
(47, 5)	0.0037	0.0036	4.2518	4.2682	4.2793	$+0.0274$	material	$+0.0274$	-0.0162	4.2741
(48, 6)	0.0024	0.0035	4.2502	4.2740	4.2871	$+0.0369$	material	$+0.0369$	-0.0098	4.2710
(49, 7)	0.0030	0.0027	4.2418	4.2692	4.2847	$+0.0430$	material	$+0.0430$	-0.0111	4.2655

B.2 Set 1 at two further learning rates and a third first-moment map

This paired extension adds 5×10^{-4} and 10^{-3} under U and N, and the map F at 7×10^{-4}, on set 1; F is the public library's only signed 3-bit map with a zero level, symmetric end points and non-uniform spacing. All twelve new (learning rate, unit) pairs qualified.

B.3 The second set of units

The set-2 test applies the replication's procedure to set 2 in three conditions judged separately: N and F at 7×10^{-4} and N at 5×10^{-4}; its candidates saved the checkpoints read in Appendices D.3

Table 6: Per-unit outcomes on set 1 at the two further learning rates and under F; the primary map is N (F rows: F). The F condition reuses the replication's references, controls and b_j. The locked rule selected *pre* in every qualified unit.

				L, primary map			frozen endpoint		secondary reading	
Cond.	Unit	dev.	b_j	plain	post	pre	Δ_j	state	Γ_j	Γ_j^U
5×10^{-4}	(44, 2)	0.0024	0.0026	4.2875	4.2946	4.3003	$+0.0128$	material	$+0.0128$ n.b.	-0.0110 best
	(45, 3)	0.0015	0.0020	4.2963	4.2897	4.2989	$+0.0025$	below	$+0.0092$ n.b.	-0.0190 best
	(46, 4)	0.0009	0.0007	4.2960	4.3023	4.3191	$+0.0232$	material	$+0.0232$ n.b.	-0.0153 best
	(47, 5)	0.0015	0.0018	4.2928	4.2945	4.3018	$+0.0090$	unres.	$+0.0090$ n.b.	-0.0191 best
	(48, 6)	0.0022	0.0021	4.3019	4.3073	4.3108	$+0.0090$	unres.	$+0.0090$ n.b.	-0.0078 best
	(49, 7)	0.0009	0.0006	4.2872	4.2937	4.3054	$+0.0181$	material	$+0.0181$ n.b.	-0.0173 best
10^{-3}	(44, 2)	0.0028	0.0028	4.2605	4.2774	4.2980	$+0.0375$	material	$+0.0375$ n.b.	-0.0165 best
	(45, 3)	0.0036	0.0033	4.2692	4.2772	4.3101	$+0.0408$	material	$+0.0408$ n.b.	-0.0117 best
	(46, 4)	0.0031	0.0055	4.2533	4.2971	4.3111	$+0.0578$	material	$+0.0578$ n.b.	-0.0092 best
	(47, 5)	0.0052	0.0095	4.2331	4.2774	4.2856	$+0.0525$	material	$+0.0525$ n.b.	-0.0057 unres.
	(48, 6)	0.0026	0.0027	4.2542	4.2810	4.3063	$+0.0521$	material	$+0.0521$ n.b.	-0.0165 best
	(49, 7)	0.0050	0.0046	4.2362	4.2763	4.2955	$+0.0593$	material	$+0.0593$ n.b.	-0.0064 best
F, 7×10^{-4}	(44, 2)	0.0017	0.0023	4.3238	4.2383	4.2586	-0.0652	below	$+0.0203$ n.b.	–
	(45, 3)	0.0014	0.0013	4.3393	4.2495	4.2569	-0.0824	below	$+0.0075$ n.b.	–
	(46, 4)	0.0030	0.0028	4.3404	4.2532	4.2683	-0.0721	below	$+0.0151$ n.b.	–
	(47, 5)	0.0037	0.0036	4.2996	4.2410	4.2558	-0.0437	below	$+0.0148$ n.b.	–
	(48, 6)	0.0024	0.0035	4.3578	4.2487	4.2625	-0.0953	below	$+0.0138$ n.b.	–
	(49, 7)	0.0030	0.0027	4.3290	4.2484	4.2581	-0.0709	below	$+0.0097$ n.b.	–

and C.5.1. All twelve pairs qualified, and the step-600 bucket-0 error ordered pre $<$ post $<$ plain in all 30 combinations of unit, learning rate and codebook. U candidates were trained on every qualified unit but serve only the pre-declared secondary reading; U defines no frozen endpoint in this set.

Table 7: Per-unit outcomes on set 2 at 7×10^{-4} and 5×10^{-4}; the primary map is N (F rows: F). F uses the 7×10^{-4} controls and b_j. LPMM: the library's own optimizer configured for 3-bit moments at 7×10^{-4}, a public reference outside the selection set. The locked rule selected *pre* in every qualified unit.

				L, primary map			frozen endpoint		secondary reading		
Cond.	Unit	dev.	b_j	plain	post	pre	Δ_j	state	Γ_j	Γ_j^U	LPMM
7×10^{-4}	(50, 8)	0.0010	0.0007	4.2698	4.2610	4.2845	$+0.0148$	material	$+0.0235$ n.b.	-0.0051 best	4.2812
	(51, 9)	0.0021	0.0031	4.2430	4.2681	4.2799	$+0.0369$	material	$+0.0369$ n.b.	-0.0067 best	4.2693
	(52, 10)	0.0024	0.0026	4.2861	4.2779	4.2905	$+0.0044$	below	$+0.0126$ n.b.	-0.0183 best	4.2817
	(53, 11)	0.0023	0.0019	4.2570	4.2652	4.2802	$+0.0232$	material	$+0.0232$ n.b.	-0.0085 best	4.2671
	(54, 12)	0.0040	0.0039	4.2803	4.2644	4.2732	-0.0071	below	$+0.0088$ n.b.	-0.0111 best	4.2698
	(55, 13)	0.0046	0.0040	4.2490	4.2728	4.2772	$+0.0283$	material	$+0.0283$ n.b.	-0.0102 best	4.2727
F, 7×10^{-4}	(50, 8)	0.0010	0.0007	4.3328	4.2401	4.2637	-0.0691	below	$+0.0236$ n.b.	–	–
	(51, 9)	0.0021	0.0031	4.3790	4.2411	4.2527	-0.1263	below	$+0.0116$ n.b.	–	–
	(52, 10)	0.0024	0.0026	4.3209	4.2446	4.2579	-0.0630	below	$+0.0133$ n.b.	–	–
	(53, 11)	0.0023	0.0019	4.3136	4.2405	4.2541	-0.0596	below	$+0.0135$ n.b.	–	–
	(54, 12)	0.0040	0.0039	4.3171	4.2480	4.2571	-0.0600	below	$+0.0091$ n.b.	–	–
	(55, 13)	0.0046	0.0040	4.3459	4.2498	4.2529	-0.0930	below	$+0.0031$ unres.	–	–
5×10^{-4}	(50, 8)	0.0022	0.0020	4.3154	4.2956	4.3143	-0.0011	below	$+0.0187$ n.b.	-0.0129 best	–
	(51, 9)	0.0006	0.0004	4.2861	4.2999	4.3099	$+0.0237$	material	$+0.0237$ n.b.	-0.0100 best	–
	(52, 10)	0.0031	0.0030	4.3061	4.2924	4.3060	-0.0001	below	$+0.0136$ n.b.	-0.0110 best	–
	(53, 11)	0.0022	0.0015	4.2818	4.2918	4.3080	$+0.0262$	material	$+0.0262$ n.b.	-0.0109 best	–
	(54, 12)	0.0017	0.0016	4.3295	4.2980	4.3038	-0.0257	below	$+0.0057$ n.b.	-0.0149 best	–
	(55, 13)	0.0050	0.0048	4.2936	4.2940	4.3074	$+0.0138$	unres.	$+0.0138$ n.b.	-0.0161 best	–

At 10^{-3} the frozen endpoint is defined under N only; U candidates were trained on the same qualified units and read only through the pre-declared secondary reading. In $(52, 10)$ the identity and op-order controls deviated from the reference by 0.0140 and 0.0117 nats and in $(53, 11)$ the identity control by 0.0088; the two units failed qualification, were not replaced, and make the verdict inconclusive. The four qualified units all showed material loss.

B.4 THE SAME RULE AT 4+4 BITS

The 4+4-bit test changes only the bit width of the replication (Appendix A.3), on the same units, references, controls and captures. The 4-bit error ordered pre $<$ post $<$ plain under U and N, with

Table 8: Per-unit outcomes on set 2 at 10^{-3}; the primary map is N. The locked rule selected *pre* in every qualified unit. (52, 10) and (53, 11) failed qualification (*invalid*), so they were not locked and their candidates were not trained.

				L, primary map			frozen endpoint		secondary reading	
Cond.	Unit	dev.	b_j	plain	post	pre	Δ_j	state	Γ_j	Γ_j^U
10^{-3}	(50, 8)	0.0035	0.0034	4.2660	4.2711	4.3008	$+0.0348$	material	$+0.0348$ n.b.	-0.0085 best
	(51, 9)	0.0061	0.0058	4.2343	4.2717	4.2823	$+0.0480$	material	$+0.0480$ n.b.	$+0.0082$ n.b.
	(52, 10)	0.0140	0.0136	–	–	–	–	invalid	–	–
	(53, 11)	0.0088	0.0096	–	–	–	–	invalid	–	–
	(54, 12)	0.0062	0.0058	4.2657	4.2789	4.2941	$+0.0284$	material	$+0.0284$ n.b.	$+0.0018$ unres.
	(55, 13)	0.0061	0.0059	4.2283	4.2756	4.2970	$+0.0687$	material	$+0.0687$ n.b.	$+0.0016$ unres.

smaller values: under N, 0.177–0.184 for plain, 0.131–0.135 for post and 0.126–0.129 for pre at step 600, against 0.318–0.333, 0.246–0.249 and 0.231–0.235 at 3 bits. The rule selected pre in every unit.

Under N the endpoint Δ_j was negative in all six units, -0.021 to -0.008 nats, every unit below the margin: *below the preset cost*.

Against the references. Under both maps the 4-bit post and pre codecs finished below the AdamW reference of their unit in all six units, by 0.024 to 0.038 nats, and 4-bit plain within -0.025 to $+0.010$; every 3+3-bit codec finished above its reference, by 0.016 nats or more (references 4.220 to 4.234, identity controls within ± 0.003 of them). The 4-bit quantizer agrees with an independent transcription on the reference states (Appendix F) and the identity control matches AdamW, so the effect lies in the specified 4-bit codec rather than in the training loop; its cause was not investigated. The 4+4-bit verdict compares codecs of that bit width with each other.

Table 9: Every 4+4-bit candidate's late held-out loss on set 1 at 7×10^{-4} (the 3+3-bit values for the same units under N are in Table 5), the unit's control spread b_j (from the replication), the frozen endpoint $\Delta_j = L_j(\text{sel}) - L_j(\text{plain})$ under N (the error rule selected pre in every unit, codebook and probe) and its state at $\delta = 0.01$ nats. Six-unit verdict: below preset cost.

Unit	b_j	N, 4+4 bits			U, 4+4 bits			Δ_j	state
		plain	post	pre	plain	post	pre		
(44, 2)	0.0023	4.2058	4.1952	4.1883	4.2131	4.1988	4.1937	-0.0176	below margin
(45, 3)	0.0013	4.2046	4.1957	4.1951	4.2219	4.1961	4.1932	-0.0095	below margin
(46, 4)	0.0028	4.2102	4.1977	4.2000	4.2292	4.2053	4.1992	-0.0102	below margin
(47, 5)	0.0036	4.1997	4.1963	4.1914	4.2130	4.1877	4.1887	-0.0083	below margin
(48, 6)	0.0035	4.2188	4.1963	4.1978	4.2368	4.1926	4.1960	-0.0210	below margin
(49, 7)	0.0027	4.2050	4.1989	4.1919	4.2148	4.1984	4.1904	-0.0131	below margin

C Transfer Across Working Points and Scoring Conditions

C.1 Working-point construction

The 350M test and the scale intervention share one working point built from the 130M recipe by fixed rules: 369,148,928 parameters (24 layers, width 1,024, 16 heads) against 134,105,856 (12, 768, 12); $T = \text{round}(4{,}000 \cdot P_{350}/P_{130}) = 11{,}011$ steps with 1,101 warm-up; probes at 1,652 and 5,506, the 130M probes as fractions of T; and learning rate 5×10^{-4}, chosen by an AdamW-only pilot over $\{3, 5, 7, 10\} \times 10^{-4}$. To keep the 130M ratio of drawn tokens to cache size, a ratio $\lambda = 0.65536$ (not the fraction of distinct tokens covered), the 350M cache is $L = \text{round}(T_{350} \cdot 32 \cdot 256/\lambda) = 137{,}637{,}500$ tokens of the 500M-token cache, with every window around an exact 64-gram match with the held-out corpus removed. The candidates are the three codecs under U and N, and the controls, δ, b_j and the bound 0.0082 are inherited from 130M, the bound not re-measured.

C.2 Fidelity ranking across working points

This measurement reads the error only, on the AdamW reference states of set 1 at 130M (7×10^{-4}), of the six 350M units and of six 1B reference runs (1,345,423,360 parameters; 5×10^{-4}, trained to

the same 0.244 tokens per parameter), each at 0.15 and 0.5 of the schedule, under U, N and F. Every codec's stored state starts as the encoding of the state after the next update and then evolves as in training; E_H is the mean relative error of the next H updates against the exact AdamW updates, so E_1 is the locked rule, with $H \in \{1, 8, 32, 128\}$ at 130M and 350M and $\{1, 8, 32\}$ at 1B.

Table 10: Mean one-step error E_1 (bucket 0) over six units and two probes on the AdamW reference states, by map, codec and model size. At every size the order is pre $<$ post $<$ plain in all 36 unit–probe–map cells.

map	codec	130M	350M	1B
U	plain	0.442	0.412	0.468
U	post	0.245	0.243	0.259
U	pre	0.228	0.229	0.233
N	plain	0.326	0.313	0.351
N	post	0.247	0.246	0.259
N	pre	0.232	0.234	0.236
F	plain	0.372	0.350	0.409
F	post	0.232	0.227	0.241
F	pre	0.213	0.213	0.216

Reading. At every size the error ordered pre $<$ post $<$ plain under every map in all six units at both probes, in bucket 0 and globally: *ranking invariant* (Table 10); the value of N/pre barely moved: 0.232, 0.234 and 0.236 at 130M, 350M and 1B, against 0.326, 0.313 and 0.351 for N/plain. With $H = 8$ or 32 the pick equalled the one-step pick in every cell at every size. At 130M under N at the earlier probe with $H = 128$, the ratio $E_H(\text{plain})/E_H(\text{pre})$ crossed 1, where the pick moved to plain in four of six units; at 350M the same ratio stayed 1.06–1.15. The sizes also differ in learning rate, training length and data, so the invariance is not an effect of size alone.

C.3 Selection at the 350M working point

The two same-math controls deviated from their reference by 0.0020–0.0049 nats, below the inherited 0.0082 bound, so every unit qualified (b_j 0.0017–0.0046), and the rule selected pre under both maps at both probes in all six units.

Under N the frozen endpoint was positive and beyond $\delta + b_j$ in all six units, a *repeatable material cost*, and pre was never the lowest N candidate (Table 11).

Table 11: Every candidate's late held-out loss at the 350M working point (5×10^{-4}, 3+3 bits), the unit's control spread b_j, the frozen endpoint $\Delta_j = L_j(\text{pre}) - L_j(\text{plain})$ under N (the rule selected pre in every unit, map and probe) and its state at $\delta = 0.01$ nats, and, post hoc and descriptive, each N candidate's largest excursion of the training loss above its own running minimum after warm-up, in nats (no U candidate, reference or control at 350M exceeded 0.83). Six-unit verdict: repeatable material cost.

Unit	b_j	N			U			Δ_j	state	N excursion
		plain	post	pre	plain	post	pre			plain/post/pre
(63, 21)	0.0021	3.933	7.395	4.248	3.895	3.840	3.838	$+0.315$	material loss	5.3/5.7/6.6
(64, 22)	0.0035	3.915	4.399	4.021	3.901	3.840	3.836	$+0.105$	material loss	6.0/7.3/7.1
(65, 23)	0.0046	3.928	4.492	4.008	3.909	3.842	3.833	$+0.081$	material loss	5.7/7.1/6.8
(66, 24)	0.0025	3.942	4.027	4.748	3.898	3.836	3.833	$+0.806$	material loss	5.2/8.2/5.6
(67, 25)	0.0017	3.916	5.162	5.769	3.898	3.840	3.839	$+1.853$	material loss	5.8/6.5/6.9
(68, 26)	0.0028	5.186	3.970	7.178	3.899	3.841	3.838	$+1.992$	material loss	5.4/6.5/7.1

Training stability (post hoc, descriptive). Under an excursion rule written after these losses were seen (a logged training loss more than 1.0 nat above the run's running minimum after warm-up), all 18 N candidates had excursions of 5.2 to 8.2 nats, first between steps 2,182 and 4,575, none non-finite; no U candidate, reference or control at 350M exceeded 0.83 nats. The frozen endpoint includes these excursions.

C.4 Training and qualification at the 1B working point

The 1B test trained plain and pre under N and U and two same-math controls in each of six units at 1,345,423,360 parameters for 40,130 steps (0.244 tokens per parameter) at the 350M learning rate, against the six AdamW references of the checkpoint-fork test, which ran in a separate batch. Every unit failed the inherited 0.0082-nat qualification bound: the larger control-to-reference late-mean difference ranged from 0.0084 to 0.0216 nats (b_j 0.0084 to 0.0249), and no unit qualified under any of five late-window definitions, so the condition is inconclusive and yields no selection verdict.

Descriptively, all twelve N candidates had excursions and no U candidate did.

C.5 Scoring-input and entry-point diagnostics

C.5.1 Scoring on the candidates' own trajectories

This test is exploratory: it reads set-2 candidate checkpoints whose outcomes were known, and a candidate's own state exists only after it has trained, so it could not serve as a selection rule. Its predictions and reading rules were committed before any candidate tensor was read.

Input and grid. Encoding a candidate's already encoded state again would measure repeated-encoding damage, so each donor input is rebuilt: the checkpoint is resumed for two steps along the donor's own optimizer and data stream, and every codec of the donor's map is scored on the resulting raw state with the locked selector's functions; for the AdamW donor this is the reference reading. The grid is set 2 at 7×10^{-4} (primary: N, probe 600, bucket 0) and at 5×10^{-4}, with the AdamW reference and the three candidates of each map as donors, 204 inputs. The registered hypothesis predicted that on the N candidates' own inputs the selection moves from pre toward the unit's best-trained codec while U stays at pre; the alternative, trajectory invariance, predicts the same selection on every input.

Reading. On the post and pre donors the rule still selects pre in every unit, map, learning rate and probe. On the plain donor it selects plain at probe 600 in every unit under N and U and in five of six under F, and at probe 2,000 in five of six units under N but pre in every unit under U and F. The frozen six-unit state is therefore *selection changes, not toward training* in the primary grid and in thirteen of the nineteen auxiliary grids, and *trajectory invariant* in the other six. Under U, where pre trained best in every unit, this moved the choice away from the best-trained codec, outside both registered branches.

C.5.2 Checkpoint entry time

Every fork candidate continues the same stored AdamW reference checkpoint, with its moments and step count, instead of training from initialization. The protocol first asks whether this reproduces the known from-scratch verdict at 130M. Forking set 1 at 7×10^{-4} at 15% of the schedule, the N endpoint was four units below the margin and two material, *inconclusive*; at 50%, all six were below the margin, *below the preset cost*; from scratch the same units read a repeatable material cost. The known answer is therefore *not reproduced* at both fork steps, which by the protocol's rule stopped the 1B forks; the 350M forks are read in Table 4. A fork's state at the fork step is the reference's, and its suffix runs at the schedule's lower learning rates.

D Scale Interventions and Stability

D.1 Scale intervention at the 350M working point

The intervention was registered after the 350M test had shown every N candidate unstable and before any of its own runs; it predicted that the instability would disappear.

Design. The three codecs with the signed-maximum scale under N on the six 350M units, at the same working point, each paired with its absolute-scale twin from the 350M test; a pair shares levels, block size, layout and tie rule and differs in the per-block scale. The pairs also differ in checkpoint

writes; a 300-step control that changed only these writes reproduced all 300 training losses and four held-out evaluations bit for bit.

Stability criterion. A run with a non-finite training loss is diverged; a finite run is unstable if, after warm-up, a logged training loss exceeds its running minimum by more than 1.0 nat. With k the diverged or unstable signed-scale runs, the prediction is met if $k \leq 1$, rejected if $k \geq 14$, and partial otherwise. The allowance of one reflects the 2 of 105 runs under the absolute scale at 130M that were unstable when the test was written.

Result. $k = 0$ of 18: no signed-scale run diverged or had an excursion, whereas the same classifier marked all 18 absolute-scale runs unstable. The largest excursion per run ranged from 0.714 to 0.820 nats under the signed scale and from 5.168 to 8.219 under the absolute scale (Table 2).

Learning rate and idempotence. A later test, registered after this read-out, trained post under N with the absolute scale at 3×10^{-4} and with the idempotent control scale (Appendix A.3) at 5×10^{-4}, six runs each; per arm of six, $k \leq 1$ reads stable, $k \geq 5$ unstable, and any count between partial, and stability of the idempotent control was predicted. Both arms read partial: the absolute scale at 3×10^{-4} left 2 of 6 runs unstable (6 of 6 at 5×10^{-4}), and the idempotent control left 3 of 6 unstable.

Stability census (post hoc, descriptive). Applied to the 780 training runs of the 130M tests, the 350M test and intervention, the 1B test, the checkpoint-fork test and the six 1B references, the same rule marked as unstable: at 130M, 2 of 126 from-scratch N runs (including five byte-identical reproductions) and 1 of the 18 LPMM runs; at 350M, 18 of 18 from-scratch N runs and 18 of 36 N forks; at 1B, 12 of 12 N runs and 2 of the 18 codec-free runs, as brief transients that recovered. No run under U, F or the signed scale crossed the threshold.

D.2 Scale intervention at the 1B working point

Registered after the 1B test had shown every N candidate unstable, this intervention trained plain and pre under N with the signed-maximum scale on the six 1B units, each paired with its absolute-scale twin, which differs only in the scale rule. Because two of the 18 codec-free 1B runs had crossed the threshold as recovering transients, the prediction is met when at most 2 of the 12 signed-scale runs are diverged or unstable and is rejected when at least 10 are.

$k = 0$ of 12: no signed-scale run diverged or had an excursion, while the same classifier marked all twelve absolute-scale twins unstable. The largest excursion per run was 0.778 to 0.898 nats under the signed scale and 5.627 to 13.697 under the absolute scale.

D.3 Scale intervention and mechanism diagnostics at 130M

This test was committed before any of its runs and before any saved codec state was read; the signed-maximum and idempotent control scales were added to the codec module as new, default-off codebooks.

Repeated-encoding diagnostics. On stored states from the second-set runs at 7×10^{-4}, re-encoding produced a 3.5–6.2% change in update space under N (median by codec) and no change under U or F. The registered explanation predicted that extra decay under N would be larger for post and pre than for plain; instead, measured extra decay was largest for plain (median 0.0025, versus 0.0008 for post and 0.0003 for pre). The endpoints of these runs were already known when this diagnostic was specified, so it is used to test the proposed mechanism rather than as independent confirmation of the training outcome.

Development. On the development units $(50, 8)$ and $(51, 9)$ the signed scale raised the loss of plain and lowered that of post and pre, all materially in both units. The frozen rules chose post as the signed-scale codec to confirm and post under F, lower than the signed-scale post by only 0.0002 nats in mean late loss, as the fixed candidate. The drift-matched controls did not run because the drift of post was negligible (the median ratio of realized to nominal decay was 0.99923 against a threshold

of 0.999). The idempotent control was added by a dated amendment after the development read-out and before any confirmation candidate ran.

Confirmation. Six new units, $(56, 14)$ to $(61, 19)$, all qualified (largest control deviation 0.0014 to 0.0055 nats; b_j 0.0020 to 0.0064), and the error-rule selections were locked before the 54 candidate runs. Primary endpoint: no gain beyond the threshold. Second endpoint: inconclusive.

F/pre, post hoc. F/pre, the lowest-error codec across codebooks (Section 3), was not in the frozen menu. After the confirmation read-out it was trained on the same six units with the same recipe, references and controls, from a later revision of the codec module that adds default-off codebooks and leaves the F levels unchanged (same recorded levels hash). Its bucket-0 error on each unit's reference state, by both implementations of the lock, was 0.211 to 0.216 against 0.225 to 0.230 for the signed-scale pre, so the error rule would have chosen it in every unit. Its late loss was 0.010 to 0.020 nats above post under F in every unit (a material loss in two, unresolved in four) and 0.002 to 0.020 below the signed-scale pre (a material gain in one, unresolved in three, within threshold in two); the best of the eight candidates was unchanged, post under F in three units and plain under the absolute scale in three. No frozen verdict changes. A rule that first rejects codecs whose re-encoding changes the stored state and then takes the lowest error chooses the signed-scale pre on this menu, as the error rule did, and F/pre when admitted, so its regret is the error rule's.

Table 12: Every frozen-menu run of the six confirmation units: late held-out loss (the post-hoc F/pre runs are reported separately in this appendix). The idempotent control (idem. ctrl.) is a scale added after development and before confirmation training; it fits a negative extreme to the codebook's negative endpoint. LPMM is the library's own optimizer configured for 3-bit moments. The last three columns are the same-math controls.

Unit	N, absolute max			N, signed max			F	N, idem. ctrl.	LPMM	AdamW	identity	op-order
	plain	post	pre	plain	post	pre	post	post		ref.		null
(56, 14)	4.2488	4.2695	4.2989	4.2821	4.2536	4.2757	4.2445	4.2541	4.2741	4.2275	4.2266	4.2286
(57, 15)	4.2435	4.2723	4.2912	4.3076	4.2604	4.2819	4.2480	4.2561	4.2719	4.2272	4.2282	4.2243
(58, 16)	4.2777	4.2697	4.2943	4.2810	4.2536	4.2737	4.2506	4.2577	4.2806	4.2288	4.2312	4.2248
(59, 17)	4.2409	4.2589	4.2705	4.3077	4.2421	4.2630	4.2419	4.2403	4.2615	4.2238	4.2227	4.2291
(60, 18)	4.2406	4.2694	4.2860	4.3016	4.2448	4.2622	4.2449	4.2411	4.2739	4.2219	4.2252	4.2267
(61, 19)	4.2578	4.2687	4.2802	4.2690	4.2479	4.2598	4.2472	4.2424	4.2629	4.2210	4.2176	4.2182

E Selection Rules and Decision Baselines

E.1 Selection rules on the 130M confirmation units

Three rules were fixed before any confirmation candidate was trained. The *error rule* selected the candidate with the smallest bucket-0 error on the unit's own AdamW reference. The *fixed rule* used post under F, chosen once from the two development units. The *1,000-step rule* ranked candidates by the mean held-out loss at steps 900 and 1,000 and retained the fixed choice unless another candidate was lower by more than the spread of the unit's controls. The 1,000-step horizon was chosen over 600 steps on the 18 second-set unit–map cells at 7×10^{-4}: its choice matched the late ordering in 15 of 18 cells, compared with 8 of 18 for the 600-step rule. The corresponding prefix-equivalent cost over the six confirmation units is 3.5 A100-hours. Table 3 reports the regret of all three rules, and Table 12 gives every candidate loss.

E.2 Selection inside the external SOLO optimizer

Design. The external-optimizer test applies the same rule, a one-step update error fixed before training, inside SOLO (Xu et al., 2025), run unmodified from its published code. Every run of this test, the AdamW reference, both controls and the five candidates, uses $\beta = (0.3, 0.95)$, fixed before its runs, rather than the $(0.9, 0.95)$ of our own codecs' runs. The rule chooses among five settings of SOLO's quantile parameter, $p \in \{0.01, 0.05, 0.1, 0.3, 0.5\}$, which sets the base of its logarithmic second-moment quantizer; $p = 0.1$ is the default. Six new 130M units, $(77, 35)$ to $(82, 40)$, were trained from initialization at two paired budgets, 4,000 and 40,000 steps (0.244 and 2.443 tokens per parameter), the longer one primary; each unit and budget has an AdamW reference, an identity

control (SOLO with its quantizers disabled), an operation-order control and the five candidates. SOLO stores the first moment in 4 bits and the second in 2 bits, in blocks of 128.

Rules and endpoint. The learning rate, 7×10^{-4}, was chosen by an AdamW-only pilot on two development units. Before any confirmation candidate trained, the rules were fixed: the development choice (the lowest mean late loss on the development units at 4,000 steps, which was the default $p = 0.1$); the one-step error on the unit's own reference at the checkpoint-600 probe (6,000 at 40,000 steps) with SOLO's own encoding, averaged over 32 draws of its stochastic rounding and locked per unit; and a pilot rule, the lowest mean held-out loss at 0.225 and 0.25 of training. The endpoint $D_j = L_j(\text{error choice}) - L_j(\text{development choice})$ is compared with $\delta = 0.01$ directly, without qualification or b_j: a consistent material benefit if all six $D_j < -\delta$, a consistent material cost if all six exceed δ, all within the margin if every $|D_j| < \delta$, and otherwise mixed.

Table 13: The external-optimizer test, one row per unit and training budget. p_{f}: the quantile setting the one-step error selected on the unit's own reference; p^*: the lowest-loss of the five candidates; $D_j = L(p_{\mathrm{f}}) - L(0.1)$, the frozen endpoint against the development choice, which was the published default $p = 0.1$, chosen on the 4,000-step development runs; regret: loss above p^*; p_{pilot}: the pilot rule's choice at a quarter of training; spread: highest minus lowest candidate loss; ctrl.: the larger absolute difference between a same-math control's late loss and the unit's AdamW reference's, so the spread of the three late losses lies between ctrl. and twice it. Six-unit readings: consistent material benefit at 40,000 steps (primary), mixed or at the boundary at 4,000.

Steps	Unit	p_{f}	p^*	D_j	regret	p_{pilot}	regret	spread	ctrl.
40,000	(77, 35)	0.3	0.5	-0.0165	0.0155	0.5	0.0000	0.0459	0.0050
	(78, 36)	0.3	0.5	-0.0166	0.0163	0.5	0.0000	0.0436	0.0024
	(79, 37)	0.3	0.5	-0.0124	0.0171	0.5	0.0000	0.0457	0.0002
	(80, 38)	0.5	0.5	-0.0280	0.0000	0.5	0.0000	0.0469	0.0036
	(81, 39)	0.3	0.5	-0.0205	0.0163	0.5	0.0000	0.0506	0.0036
	(82, 40)	0.3	0.5	-0.0168	0.0148	0.5	0.0000	0.0429	0.0016
4,000	(77, 35)	0.3	0.3	-0.0058	0.0000	0.01	0.0038	0.0132	0.0055
	(78, 36)	0.3	0.01	$+0.0017$	0.0058	0.01	0.0000	0.0173	0.0017
	(79, 37)	0.3	0.1	$+0.0011$	0.0011	0.1	0.0000	0.0136	0.0133
	(80, 38)	0.3	0.1	$+0.0104$	0.0104	0.01	0.0015	0.0104	0.0185
	(81, 39)	0.3	0.1	$+0.0123$	0.0123	0.1	0.0000	0.0274	0.0051
	(82, 40)	0.3	0.1	$+0.0200$	0.0200	0.05	0.0083	0.0224	0.0105

Result. At 40,000 steps the reading is a *consistent material benefit*: the error's choice was lower than the default by 0.012 to 0.028 nats in all six units (Table 13). It was the lowest-loss candidate in one unit; in the other five it chose $p = 0.3$ while $p = 0.5$ trained best, with regret 0.015 to 0.017. At 4,000 steps the controls deviated from the reference by up to 0.019 nats, above the 130M qualification bound in three units and above the candidate spread in one, so the reading is descriptive. It is *mixed*: D_j ran from -0.006 to $+0.020$, with three units above δ, and the candidates spanned 0.010 to 0.027 nats while the controls deviated by up to 0.019. The error ranked the five settings identically in 11 of the 12 unit–budget pairs. In five of six units the error chose the same setting, $p = 0.3$, at both budgets while the training decision differed.

A development-fixed choice at the target budget (post hoc). After the read-out, the five settings were trained on the two development units at 40,000 steps with the same recipe, code and β. The lowest mean late loss was $p = 0.5$ (3.515; $p = 0.3$ 3.527, $p = 0.1$ 3.547). On the six confirmation units this choice was the lowest-loss setting in every unit, so the pilot rule's regret against it is 0 throughout; the error rule's choice was 0.015 to 0.017 nats above it in five units and equal in one (mixed under the six-unit rule); the 4,000-step development choice, $p = 0.1$, was 0.028 to 0.037 above it in every unit (a consistent material cost). The frozen verdict against the 4,000-step choice stands.

F Independent Recomputation

Implementations sharing no code with the read-outs recomputed every verdict and every locked error value and agreed, and a script checks every number quoted in the text against the recorded results. *Verdicts.* A second implementation re-parsed every run log and recomputed b_j, qualification,

states and verdicts with tolerances fixed beforehand; all fields agreed in every test, and the 130M scale test's two read-outs agreed in all 74 comparisons. *Locked errors.* An independent transcription of the scoring specification, importing neither the scoring code nor the codec and map modules, was compared with a relative tolerance of 10^{-6} fixed beforehand. Replication: 144 values within a relative 1.1×10^{-11}, 48 selections identical; 4+4 bits, at the locks, 144 values within a relative 2.5×10^{-11} and every selection identical; every later lock was recomputed before it was committed. *Recorded study.* A second implementation under the retrospective protocol and its tolerances recomputed it (Appendix A.1). The 72 codec error rows and the selections on all four recorded states match. *Error at 350M and 1B.* Every replay reproduced its reference's training losses bit for bit; at 350M, E_1 reproduced the 350M locked values to a relative 2.4×10^{-7}, and the identity pseudo-codec, the same recursion without encoding, stayed within 1.8×10^{-7} of the exact update at every H. *Stability at 350M and 1B.* Implementations written from the protocol without the read-out's code gave the same k, twin count and reading and the same largest excess per run (to four decimals at 350M).