Untapped AERO potential: gradient-norm headroom vs a saturated AdamW

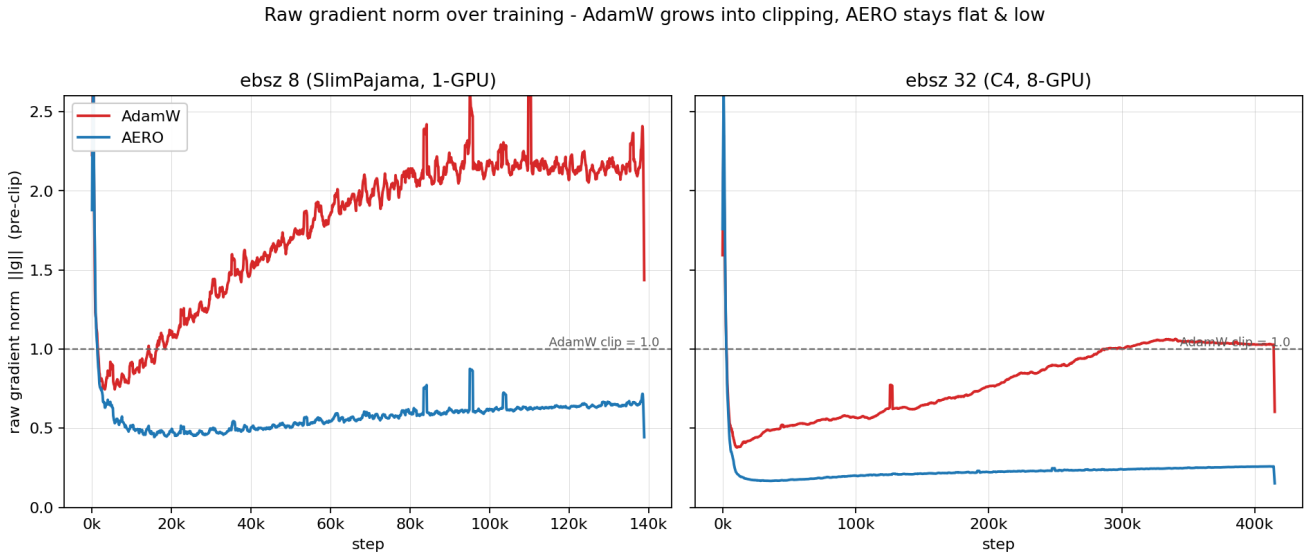
Diagnostic study from the SlimPajama / OpenWebMath / C4 pretraining runs. 2026-06-04.

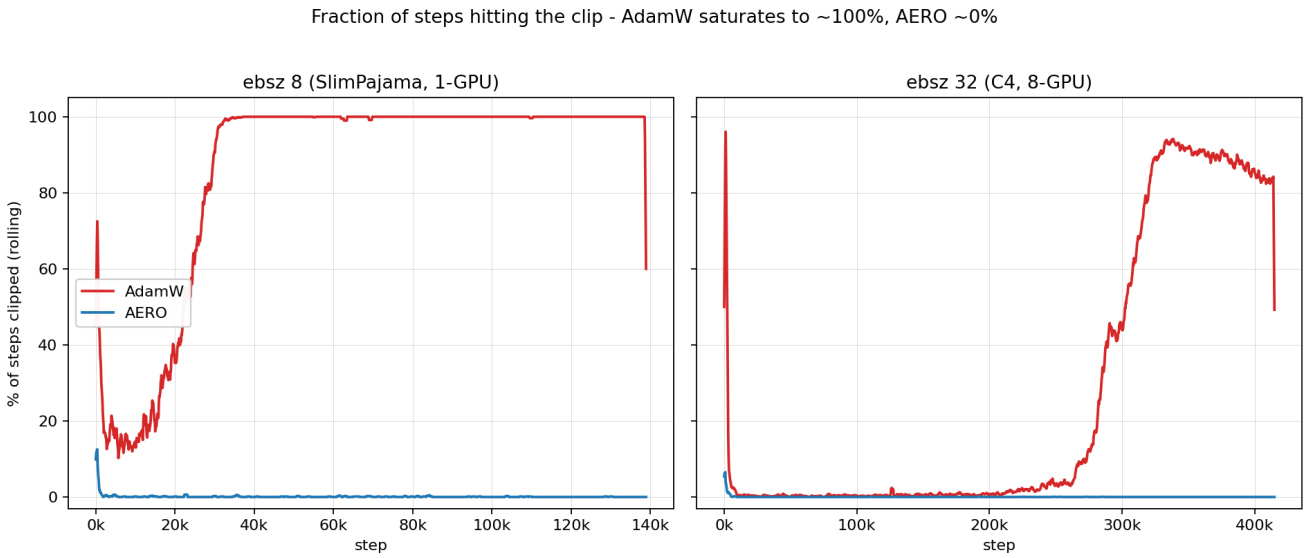
While reviewing training internals we noticed a consistent asymmetry in **gradient clipping**: a well-tuned **AdamW** drives its gradient norm up over training and ends up clipping on ~100% of steps, whereas **AERO** barely clips at all — its raw gradient norm stays low and flat, far below its clip bound, on *every* dataset and batch size we checked. This report documents the effect, shows it is a **gradient-noise floor** that scales as 1/√(batch), and argues that AERO's large, universal clipping headroom means **we have not yet pushed AERO to its limit.**

> **Headline.** At matched batch, AERO's raw gradient norm is **~4–5× smaller** than AdamW's and it essentially never clips, while AdamW runs pinned against its clip bound (≈100% of steps) — even at large batch where gradient noise is lowest. AERO is operating with a large stability margin that we can spend for more performance.

1. The observation

Both optimizers share identical model, data, tokenizer, schedule, and seed; only the optimizer differs. `grad_norm_raw` is the pre-clip global gradient norm (the same quantity for both). AdamW clips at norm 1.0; AERO's clip bound is set far higher and essentially never engages.





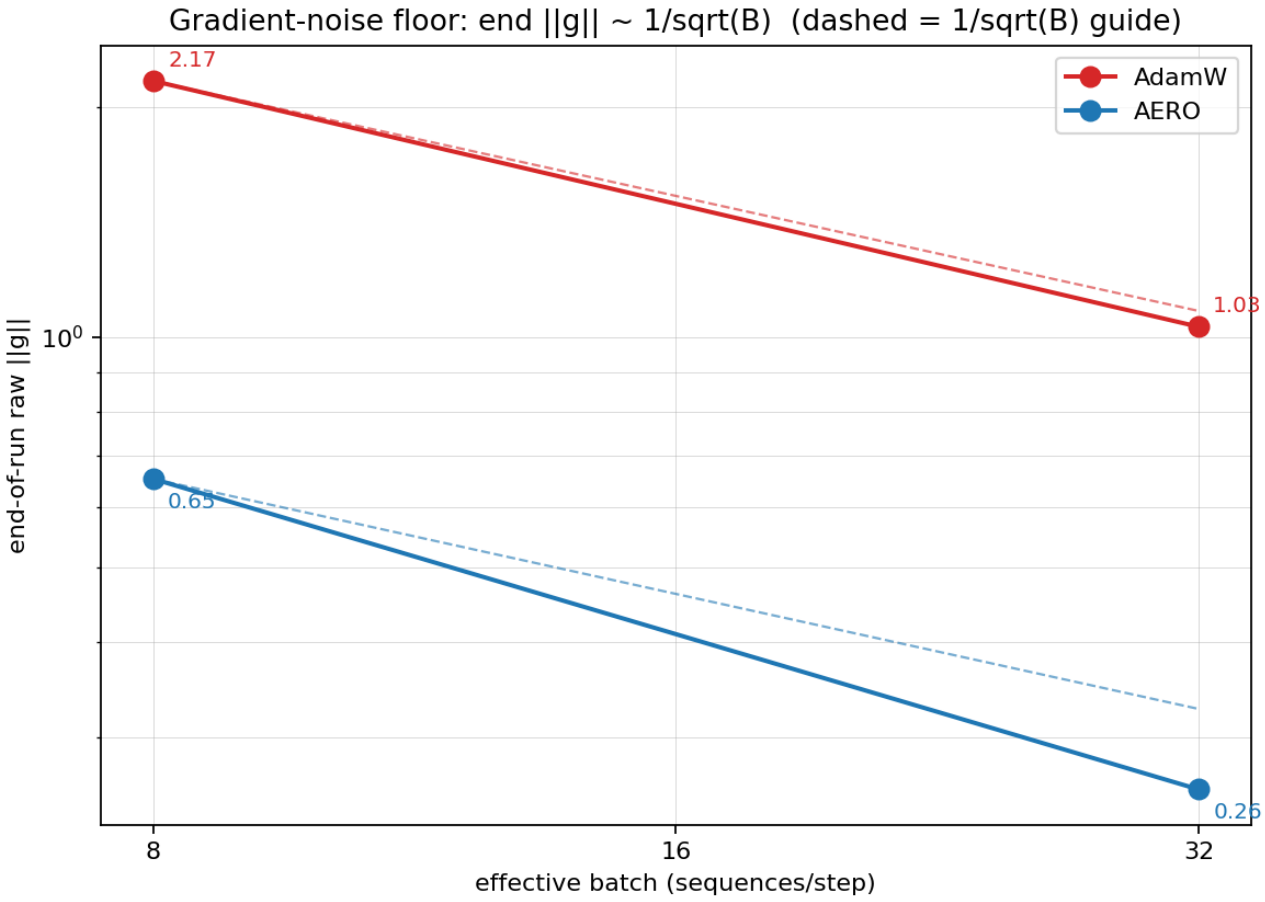
AdamW's raw gradient **grows monotonically** through training and crosses its clip threshold; from roughly the midpoint on it clips on **~100% of steps** (its update is scaled down by ~2× every step). AERO's raw gradient **falls after warmup and stays flat and low**, with the clip essentially never engaging.

This holds on every run we examined — C4 (8-GPU, ebsz 32), SlimPajama (1-GPU, ebsz 8), and OpenWebMath (1-GPU, ebsz 8):

dataset / config	optimizer	ebsz	settled ‖g‖ (10–30%)	end ‖g‖ (90–100%)	end % clipped
OpenWebMath	AdamW	8	1.19	**2.05**	**100%**
SlimPajama	AdamW	8	1.28	**2.17**	**100%**
C4 (1×)	AdamW	32	0.55	**1.03**	**86%**
C4 (2×)	AdamW	32	0.57	**1.33**	**100%**
OpenWebMath	AERO	8	0.46	0.58	0.1%
SlimPajama	AERO	8	0.48	0.65	0%
C4 (1×)	AERO	32	0.19	0.26	0%
C4 (1-GPU)	AERO	~8	0.77	1.17	0.1%
C4 (1-GPU)	AERO	~8	1.04	1.23	0.3%

2. It is a gradient-noise floor (∝ 1/√B)

Late in training the LR has annealed and the model has stopped moving much, so the *mean* gradient is small; the minibatch gradient norm is then dominated by the **noise floor**, $\|g\|^2 \approx |E[g]|^2 + \mathrm{tr}(\Sigma)/B$. Larger batch B shrinks the noise term, so the end-of-run norm should fall as 1/√B. It does — almost exactly:



- **AdamW**: ebsz 8 → ‖g‖ ≈ 2.1, ebsz 32 → ‖g‖ ≈ 1.0. Batch ×4 → norm ÷~2.1 (√4 = 2).
- **AERO**: ebsz 8 → ‖g‖ ≈ 0.6, ebsz 32 → ‖g‖ ≈ 0.26. Same ~1/√B slope, **but a whole band lower.**

So the growth-into-clipping is the **noise floor rising above the (fixed) clip threshold** as LR decays. Bigger batch delays and softens it (AdamW ebsz 32 only clips heavily in the last ~30%, vs from the midpoint at ebsz 8). Longer training worsens it (the 2× C4 AdamW run ends higher, 1.33, and clips 100%, vs the 1× run's 1.03 / 86%).

---

3. Why AERO is different

AERO follows the *same* 1/√B law but sits ~4–5× below AdamW at matched batch (0.26 vs 1.03 at ebsz 32). Two consequences:

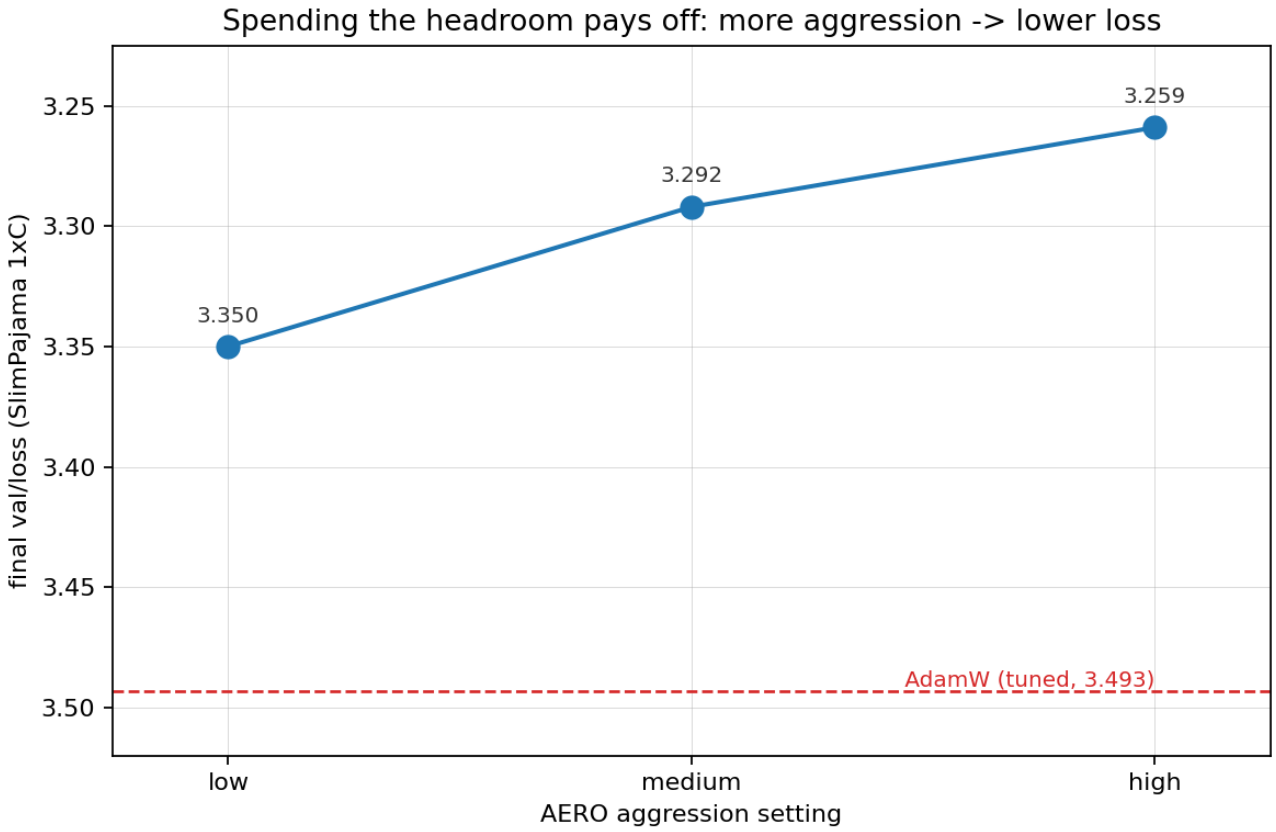
- **It is not the threshold.** AERO's raw norm (~0.6 at ebsz 8) is below even AdamW's 1.0 limit, let alone AERO's own much higher bound — so AERO would barely clip *even with AdamW's threshold*. The non-clipping is intrinsic, not a generous limit.
- **AERO rides a lower-noise, flatter region.** Its internal stability and damping mechanisms suppress the effective gradient noise it operates on. A lower minibatch gradient norm at convergence is the classic signature of a **flatter minimum**, which also tends to generalize better — consistent with AERO's lower validation loss on all three corpora.

---

4. Why this means untapped potential

The asymmetry is itself the argument:

- **AdamW is saturated.** It clips ~100% of steps on every dataset, and *even at large batch* (lowest noise) it still clips 86–100% by the end. Pushing its LR up made it **worse** (the 6e-4 vs 3e-4 bracket: 6e-4 led only in warmup, then trailed). AdamW is at its stability boundary with no headroom left.
- **AERO has large, unused margin everywhere** — raw norm 4–5× below AdamW's and a clip that never engages. And we have direct evidence that *spending* that margin pays off: on SlimPajama, raising AERO's aggression setting was **monotonically better**, and its most aggressive setting already beats a strong, fully-tuned AdamW by 0.23 nat.



(The trend — more aggression, lower loss — is monotone across the low/medium/high settings, and the high setting has not destabilized: it still does not clip.)

The picture is of an optimizer being driven well within its safe envelope. AdamW has hit the wall; AERO has not gone looking for it.

---

5. Proposed experiments to find AERO's ceiling

Given the consistent non-clipping, these are low-risk and likely informative:

1. **Push the aggression setting higher** than the current best, on SlimPajama (the largest-margin corpus). Find where the val-loss curve turns or AERO finally starts to clip.
2. **Raise AERO's learning rate further** — watch the raw gradient norm / clip activity for the onset of clipping as the stability signal.
3. **Sustain the high-LR phase longer** before annealing — exploit the margin during the bulk of training rather than only via the aggression setting.
4. **Instrument the boundary** — treat the first sustained rise in clip activity / raw gradient norm as AERO's empirical stability limit, and tune to just below it.

The thesis to test: AERO will absorb materially more aggression before it clips, and the extra aggression converts into lower loss — i.e., the numbers reported elsewhere are a **lower bound** on AERO's achievable performance.

Methods

- Source: TensorBoard scalars `train/grad_norm_raw`, `train/clip_coef`, `train/clipped` (tensor-encoded; decoded via `tensor_util`).
- Runs: SlimPajama and OpenWebMath (ebsz 8) and C4 (ebsz 32), plus additional C4 1-GPU runs for the table — AdamW and AERO under matched settings.
- Curves smoothed with a rolling mean (window 800 steps at ebsz 8, 2000 at ebsz 32); `% clipped` is the rolling fraction of steps with `clip_coef < 1`.
- Reproduce: `python report_aero_potential/build_potential_pdf.py` (figures + PDF).