ANO: ADAPTIVE OPTIMIZATION FOR TRACKING UNDER NOISE AND DRIFT

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep reinforcement learning asks optimizers to track moving objectives from persistently noisy gradients. We show that, in this regime, Adam's momentum improves the reliability of its update direction but not its progress. Because Adam's update is linear in the momentum, the variance reduction it buys is spent on shorter steps rather than on progress along the signal: at a fixed learning rate, Adam's signal-aligned progress per step matches that of RMSprop, which uses no momentum at all. This trade-off is sound when the objective is fixed and more steps can compensate, but costly when the objective moves. We introduce Ano, which uses momentum only to choose the direction of each coordinate, not the size of the step: the update takes the sign of the momentum and the magnitude of the current normalized gradient. The variance reduction then goes into direction rather than into shorter steps, roughly tripling signal-aligned progress per step in the high-noise limit. Applying the same logic to the second-moment estimate yields an Active Contraction Accumulator, which returns to the post-burst gradient scale in finite time where Adam's estimate returns only asymptotically. Ano keeps Adam's state and hyperparameter set. On real SAC gradient streams, it produces larger signal-aligned updates than Adam. Across 10 MuJoCo and Atari environments with 25 seeds, it improves aggregate return over Adam and four other adaptive and sign-based optimizers, while remaining competitive rather than superior on stationary supervised benchmarks, where shorter steps act as useful annealing.

1 INTRODUCTION

Adaptive optimizers are usually judged by how well they stabilize noisy stochastic gradients. Adam, RMSprop, and related methods maintain coordinatewise second-moment estimates and use them to normalize updates, making them robust workhorses across deep learning (Kingma & Ba, 2015; Tieleman & Hinton, 2012). This view is natural when the objective is fixed: noisy updates are repeatedly taken against the same target, so stability and long-horizon averaging are valuable.

Deep reinforcement learning changes the optimization problem. The target being optimized moves during training, through bootstrapped value estimates, replay, policy drift, and target-network updates (Henderson et al., 2018; Mnih et al., 2016; van Hasselt et al., 2018). Stability is no longer sufficient: what matters is how much progress each update makes along the current signal, because a moving target leaves only a limited number of steps to follow it before it changes again.

We show that, in this regime, Adam's momentum improves the reliability of its update direction but not its progress. Momentum averages recent gradients, which makes the sign of each coordinate update more reliable under noise. But Adam's update is linear in the momentum, and averaging does not change the mean gradient: the variance reduction therefore shows up as a shorter step rather than as more progress along the signal. At a fixed learning rate, Adam's signal-aligned progress per step matches that of RMSprop, which uses no momentum at all (Theorem 2.3). This is the right trade-off when the objective is fixed, since shorter steps under noise act as annealing and more steps compensate; it is the wrong one when the objective moves.

We introduce Ano (Adaptive Noise-aware Optimizer), which uses momentum only to choose the direction of each coordinate, not the size of the step. Ano moves each coordinate in the direction of the momentum's sign, as Adam does, but by the magnitude of the current normalized gradient. Because the sign is a nonlinear function of the momentum, the variance reduction now goes into

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.

direction, and signal-aligned progress per step roughly triples in the high-noise limit (Section 3). The same logic applies to the second-moment estimate. After a variance burst, Adam's estimate returns to the new gradient scale only asymptotically and keeps suppressing updates in the meantime; Ano's Active Contraction Accumulator matches Adam's estimate whenever the gradient scale grows, and returns to the post-burst scale in finite time (Proposition 3.2). Ano keeps Adam's state and hyperparameter set.

We test whether this mechanism carries over to deep RL. On identical SAC critic-gradient streams, Ano produces signal-aligned updates $4.94\times$ larger than Adam's (median over seeds). Across 10 MuJoCo/SAC and Atari/PPO environments with 25 seeds, it improves normalized aggregate return over Adam, RMSprop, Adan, Lion, and Grams, with stratified-bootstrap confidence intervals excluding zero for every baseline. On supervised benchmarks, where the objective is fixed and shorter steps are appropriate, Ano is competitive rather than superior.

Contributions.

- **We show that, under persistent gradient noise, Adam's momentum makes the update direction more reliable but not more productive.** At a fixed learning rate, Adam's signal-aligned progress per step matches that of RMSprop; because the update is linear in the momentum, the variance reduction shows up as a shorter step instead (Section 2).
- **We introduce Ano, which uses momentum to decide the direction of each coordinate but not the size of the step**, roughly tripling signal-aligned progress per step in the high-noise limit. Applied to the second-moment estimate, the same logic gives the Active Contraction Accumulator, which returns to the post-burst gradient scale in finite time (Section 3).
- **We show that this mechanism carries over to deep RL**, from identical gradient streams to full benchmarks over 10 environments and 25 seeds (Section 4).

2 Where Adaptive Smoothing Costs Progress

Adam-style optimizers damp their updates in two ways that suit a fixed objective: they shorten steps when gradients are noisy, and they forget past gradient variance slowly. This section characterizes each form of damping and when it becomes costly.

Notation. g_k, m_k, and v_k denote the stochastic gradient, the first-moment estimate, and the second-moment estimate at iteration k; coordinates are indexed by subscripts, e.g. $g_{k,i}$, and operations and inequalities between vectors are coordinatewise. β_1, β_2 are the decay parameters and η the learning rate. When accumulators are compared, v_k^{EMA} denotes Adam's exponential moving average.

2.1 Momentum Buys Reliability, Not Progress

Momentum is the main tool Adam-style optimizers have against gradient noise. Averaging recent gradients with weights $(1-\beta_1)\beta_1^j$ reduces the noise standard deviation by a factor $\tau := \sqrt{(1-\beta_1)/(1+\beta_1)}$ ($\tau \approx 0.23$ at $\beta_1 = 0.9$), so the sign of $m_{k,i}$ is a more reliable estimate of the sign of the true gradient than the sign of $g_{k,i}$. We ask how much of this gain reaches the parameters.

Assumption 2.1 (Local persistent-noise model)**.** Fix $x \in \mathbb{R}^d$ and suppose $\nabla F(x) = \mu \mathbf{1}_d$ with $\mu \neq 0$. The observed gradient is $g_k = \mu \mathbf{1}_d + \sigma \xi_k$, where $\sigma \geq 0$, ξ_k has i.i.d. standard Gaussian coordinates, and $\beta_1, \beta_2 \in [0, 1)$ with $\beta_1^2 < \beta_2$.

Definition 2.2 (Signal-aligned progress)**.** Under Assumption 2.1, define

$$S_\sigma^{\mathrm{alg}} := \mathbb{E}\left[-\frac{\langle \Delta x_k^{\mathrm{alg}},\, \nabla F(x) \rangle}{\eta \, \|\nabla F(x)\|_2^2} \right]. \tag{1}$$

S_σ^{alg} is the progress an update makes along $\nabla F(x)$ per step, in units of an ideal gradient step ($S_\sigma = 1$ for $\Delta x = -\eta \nabla F(x)$). It depends on both the size of the update and its agreement with $\nabla F(x)$; we track the size separately through the raw magnitude $U_\sigma^{\mathrm{alg}} := \|\Delta x_k^{\mathrm{alg}}\|_2 / (\eta\sqrt{d})$.

Theorem 2.3 (Signal-aligned progress under persistent noise)**.** *Under Assumption 2.1, as $\sigma \to \infty$,*

$$S_\sigma^{\mathrm{Adam}} = \frac{C_{\mathrm{Adam}}(\beta_1, \beta_2)}{\sigma} + o(\sigma^{-1}), \qquad S_\sigma^{\mathrm{RMSprop}} = \frac{C_{\mathrm{RMS}}(\beta_2)}{\sigma} + o(\sigma^{-1}), \tag{2}$$

where $C_{\mathrm{Adam}}, C_{\mathrm{RMS}} > 0$ are explicit constants (Appendix B.1).

Why momentum does not help Adam. At a fixed learning rate, Adam makes the same signal-aligned progress per step as RMSprop, which uses no momentum: C_{Adam} and C_{RMS} differ by less than 5% across all tested hyperparameters (Table 5). The reason is linearity. Adam's update is proportional to m_k, and averaging does not change the mean: $\mathbb{E}[m_k] = \mathbb{E}[g_k]$. To leading order in μ/σ, the signal therefore enters Adam's update exactly as it enters RMSprop's, and momentum reduces only the noise. This reduction shows up as a shorter step: Adam's raw magnitude plateaus at $U_\sigma \approx \tau$, against $U_\sigma \approx 1$ for RMSprop (Figure 1b; Theorem B.7). Adam's update is better directed than RMSprop's but correspondingly shorter, and per step both make the same progress along the signal.

When shorter steps are costly. Under a fixed objective, this trade-off is sensible: shortening steps as noise grows acts as annealing, and additional steps compensate. Under a moving objective, the number of steps available before the target changes is limited, so per-step progress becomes the relevant quantity. Theorem 2.3 is a local statement under fixed gradient statistics: non-stationarity does not create the effect, but it determines its cost.

2.2 The Second-Moment Estimate Recovers Only Asymptotically

In deep RL, exploration and bootstrapping can induce spikes in gradient magnitude (Ellis et al., 2024). After a spike, the second-moment estimate remains above the current gradient scale and keeps shortening updates by a factor $\sqrt{g_k^2/v_k} < 1$, which reduces signal-aligned progress by the same factor, until it recovers. We isolate this recovery in a constant-gradient model.

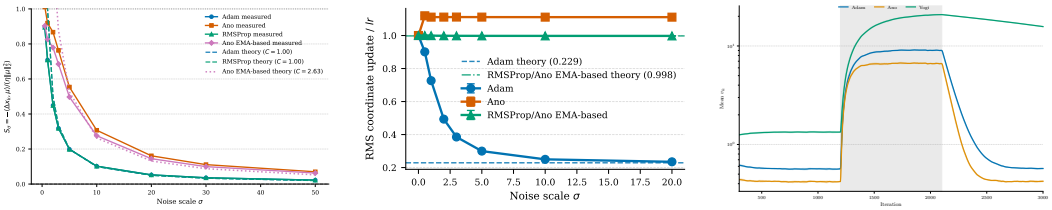
Assumption 2.4 (Burst recovery regime)**.** After a burst, the squared gradient is constant, $g_k^2 = g_\infty^2 > 0$; $\beta_2 \in [1/2, 1)$, and the estimate starts from $v_b > (1+\delta)\, g_\infty^2$ for a tolerance $\delta > 0$.

Lemma 2.5 (EMA recovery)**.** *Under Assumption 2.4, let $T^*_{\mathrm{ALG}} := \min\{k \geq 0 : v_k^{\mathrm{ALG}} \leq (1+\delta)\, g_\infty^2\}$. For Adam's EMA,*

$$T^*_{\mathrm{Adam}} = \left\lceil \frac{\log\big((v_b - g_\infty^2)/(\delta\, g_\infty^2)\big)}{\log(1/\beta_2)} \right\rceil. \tag{3}$$

In particular, $T^*_{\mathrm{Adam}} \to \infty$ as $\delta \to 0$: the EMA approaches the post-burst scale only asymptotically (Appendix B.6).

Both forms of damping are sensible for a fixed objective and costly for a moving one: the first shortens steps as noise grows, the second keeps them short after the noise has receded. Section 3 removes each without adding state or hyperparameters.



(a) Signal-aligned progress S_σ (b) Raw magnitude U_σ (c) $\hat{v}_k$ around a burst

Figure 1: Mechanisms under the local models of Section 2; Ano (Section 3) is shown for reference. (a) Adam and RMSprop make the same signal-aligned progress per step (dashed: C/σ with $C \approx 1$); Ano makes more. (b) Adam's steps are shorter than RMSprop's ($U_\sigma \approx \tau \approx 0.23$ vs. ≈ 1), while Ano's are not. (c) After a variance burst (shaded) at $\beta_2 = 0.99$, Adam's estimate approaches the new scale asymptotically, while Ano's crosses it in finite time.

Algorithm 1 Ano

Require: Learning rate η, β_1, $\beta_2 \in [1/2, 1)$, $\varepsilon > 0$, weight decay λ, initial parameters x_0

$m_0 \leftarrow 0$, $v_0 \leftarrow 0$
for $k = 1, 2, \ldots$ **do**
  $g_k \leftarrow$ stochastic gradient at x_{k-1}
  $m_k \leftarrow \beta_1 m_{k-1} + (1-\beta_1) g_k$
  $v_k \leftarrow \beta_2 v_{k-1} - (1-\beta_2)\,\mathrm{sign}(v_{k-1} - g_k^2)\, g_k^2$ → ACA, Section 3.2
  $\hat{v}_k \leftarrow v_k / (1-\beta_2^k)$
  $x_k \leftarrow x_{k-1} - \eta\,|g_k| \cdot \mathrm{sign}(m_k)/(\sqrt{\hat{v}_k} + \varepsilon) - \eta\lambda\, x_{k-1}$
end for

3 Ano: Momentum for Direction, Not Step Size

Section 2 suggests a simple principle: momentum should decide where each coordinate moves, not how far. Ano (Algorithm 1) applies this principle to the update (Section 3.1) and the same logic to the second-moment estimate (Section 3.2).

Defaults. Ano has the same state (m_k, v_k) and hyperparameter set as Adam. Bias correction is applied to v_k only, since $\mathrm{sign}(m_k)$ is invariant to it (Chen et al., 2023); weight decay follows AdamW (Loshchilov & Hutter, 2019). The recommended defaults, $\beta_1 = 0.92$, $\beta_2 = 0.99$, and $\varepsilon = 10^{-8}$, differ from Adam's; sensitivity is reported in Appendix G.

3.1 Spending Momentum on Direction

Ano moves each coordinate in the direction of the momentum's sign, as Adam does, but by the magnitude of the current normalized gradient:

$$x_{k+1,i} = x_{k,i} - \frac{\eta}{\sqrt{\hat{v}_{k,i}} + \varepsilon} \underbrace{|g_{k,i}|}_{\text{how far}} \cdot \underbrace{\mathrm{sign}(m_{k,i})}_{\text{where}}.$$

Adam and Ano therefore agree on the direction of every coordinate and differ only in how far they move it. Equivalently, since $|g_{k,i}|\,\mathrm{sign}(m_{k,i}) = g_{k,i}\,\mathrm{sign}(g_{k,i} m_{k,i})$, Ano applies the current gradient where it agrees with the momentum and reverses it where it does not.

Because the sign is a nonlinear function of m_k, the reliability gained by averaging now translates into progress, while the step no longer shrinks with the noise level. To isolate this effect from the accumulator, let Ano-EMA denote Ano with Adam's EMA in place of the ACA.

Proposition 3.1 (Signal-aligned progress of Ano)**.** *Under Assumption 2.1, as $\sigma \to \infty$,*

$$S_\sigma^{\text{Ano-EMA}} = \frac{C_{\text{Ano-EMA}}(\beta_1, \beta_2)}{\sigma} + o(\sigma^{-1}), \tag{4}$$

with an explicit constant $C_{\text{Ano-EMA}} > 0$ (Lemma B.5). Moreover, $U_\sigma^{\text{Ano-EMA}} = U_\sigma^{\text{RMSprop}}$, since both updates have magnitude $|g_{k,i}|/\sqrt{v_{k,i}^{\text{EMA}}}$.

Monte Carlo evaluation gives $C_{\text{Ano-EMA}} \approx 2.7$–$4.0$ for $\beta_1 \in [0.9, 0.95]$, against $C_{\text{Adam}} \approx C_{\text{RMS}} \approx 1$ (Table 5). At a fixed learning rate, Ano thus roughly triples signal-aligned progress per step relative to Adam at the default $\beta_1 = 0.92$, while its raw magnitude stays at RMSprop's level instead of shrinking to Adam's (Figure 1a,b).

3.2 Active Contraction Accumulator

The second form of damping acts through the denominator: every Adam-style update is divided by $\sqrt{v_k}$, so an estimate that stays above the current gradient scale reduces signal-aligned progress by the same factor, whatever the direction. We therefore want a second-moment estimate that grows like Adam's when the gradient scale increases, but contracts actively when it exceeds the current scale. Yogi (Zaheer et al., 2018) already separates the two regimes through $\mathrm{sign}(g_{k,i}^2 - v_{k-1,i})$, but

its contraction is arithmetic, so its recovery time grows linearly with the burst size (Appendix B.6). The ACA keeps Adam's geometric factor and adopts Yogi's regime switch:

$$v_{k,i} = \beta_2\, v_{k-1,i} - (1-\beta_2)\,\mathrm{sign}(v_{k-1,i} - g_{k,i}^2)\, g_{k,i}^2. \tag{5}$$

When $v_{k-1,i} < g_{k,i}^2$, this is exactly Adam's EMA. When $v_{k-1,i} > g_{k,i}^2$, the stored value is both decayed and actively reduced.

Proposition 3.2 (ACA recovery)**.** *Under Assumption 2.4,*

$$T^*_{Ano} = \left\lceil \frac{\log\big((v_b + g_\infty^2)/((2+\delta)\, g_\infty^2)\big)}{\log(1/\beta_2)} \right\rceil \le T^*_{\mathrm{Adam}}. \tag{6}$$

Unlike T^*_{Adam}, T^*_{Ano} remains bounded as $\delta \to 0$: the ACA returns to the post-burst scale in finite time.

At $\beta_2 = 0.99$, the ACA recovers in ≈ 215 steps against ≈ 730 for Adam's EMA (Figure 1c). At every step, the ACA never exceeds Adam's estimate and stays nonnegative for $\beta_2 \ge 1/2$ (Lemma B.10). Since Ano's numerator is unchanged, replacing the EMA by the ACA can therefore only increase signal-aligned progress and step size: $S_\sigma^{Ano} \ge S_\sigma^{\text{Ano-EMA}}$ and $U_\sigma^{Ano} \ge U_\sigma^{\text{RMSprop}}$ (Corollaries B.6 and B.8). This reduction is state-dependent rather than a rescaling: the ACA coincides with Adam's EMA whenever the gradient scale grows. Finally, $v_{k,i} \ge c_{\beta_2}\, g_{k,i}^2$ bounds each coordinate update by $\eta/\sqrt{c_{\beta_2}}$ (Lemma B.11).

4 Experiments

The two forms of damping identified in Section 2 matter most when persistent gradient noise and moving targets occur together. Deep RL provides both (Appendix D.1), whereas the supervised benchmarks serve as stationary sanity checks. We ask whether the mechanism appears on real gradients, whether it improves aggregate performance, and whether the gains are robust to hyperparameters.

4.1 Signal alignment on real SAC gradients

To connect Section 2 to real RL gradients, we collect critic-gradient streams from Adam SAC runs on HalfCheetah and replay optimizer states offline on the same mini-batch gradients. Large-batch critic gradients from the same replay buffer provide a reference signal $\bar{g}_t$. For each optimizer, we measure the signal-aligned progress per step (Definition 2.2), with the large-batch gradient as reference.

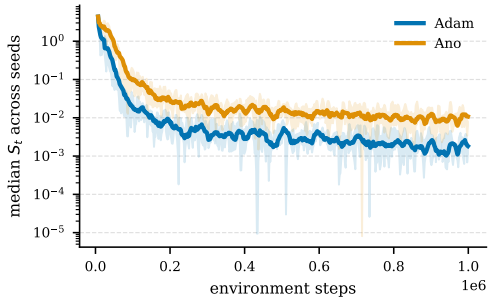


Figure 2: SAC critic-gradient replay on HalfCheetah. Adam and Ano are replayed offline on identical mini-batch gradients, with large-batch gradients from the same replay buffer as reference. Median signal-aligned progress over 5 seeds and 1M environment steps (log scale; faint lines: seeds).

Ano produces larger signal-aligned update components than Adam on identical SAC critic-gradient streams. Across 5 seeds, the median per-seed ratio $\mathrm{median}_t S_t^{Ano}/\mathrm{median}_t S_t^{\mathrm{Adam}}$ is $4.94\times$, and $S_t^{Ano} > S_t^{\mathrm{Adam}}$ on 82.17% of the 2485 captured updates. Because both optimizers are replayed on the same captured mini-batch gradients and evaluated against the same large-batch reference gradient (batch size 8192), differences in S_t isolate optimizer-state dynamics rather than differences in the gradient stream or reference estimate. Reward performance is evaluated separately below.

4.2 Does Ano improve aggregate performance?

We evaluate on 10 environments: 5 MuJoCo continuous-control tasks under SAC (Haarnoja et al., 2018; Todorov et al., 2012) and the fixed Atari-5 PPO suite from Aitchison et al. (2023) (BattleZone, DoubleDunk, Name This Game, Phoenix, Q*bert), which predicts Atari-57 aggregate performance with reported $R^2 = 0.984$. All comparisons use 25 seeds and stratified-bootstrap 95% CIs (Agarwal et al., 2021). Full metric definitions, per-domain results, learning curves, and implementation details are reported in Appendix D.

Aggregate results. Table 1 reports default-configuration pairwise head-to-head comparisons across all 10 environments. Normalized averages are computed from per-environment IQM final returns. Ano improves normalized aggregate performance over every baseline, with confidence intervals excluding zero in all comparisons; the probability-of-improvement metric is also above 0.5 for every baseline.

Baseline	Δ norm. avg. [95% CI]	$P(Ano > B)$ [95% CI]
Adam	$+6.27$ $[+1.09, +11.13]$	0.661 $[0.539, 0.766]$
RMSprop	$+7.05$ $[+2.37, +12.06]$	0.664 $[0.559, 0.761]$
Adan	$+22.32$ $[+12.68, +32.52]$	0.854 $[0.748, 0.939]$
Grams	$+22.98$ $[+14.12, +32.15]$	0.858 $[0.744, 0.947]$
Lion	$+36.23$ $[+18.20, +56.92]$	0.879 $[0.805, 0.944]$

Table 1: Pairwise head-to-head comparisons between Ano and each baseline across 10 environments under default optimizer configurations. Δ norm. avg. is the bootstrap difference in normalized aggregate performance, in percentage points, computed from per-environment IQM final returns. $P(Ano > B)$ is the stratified-bootstrap probability of improvement over baseline B.

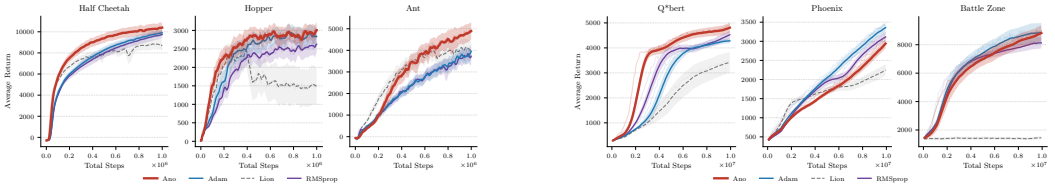


Figure 3: Representative MuJoCo/SAC and Atari/PPO learning curves at 25 seeds. Shaded regions show 95% stratified-bootstrap CIs; full curves are in Appendix D.7.

Supervised sanity check. On stationary supervised benchmarks, Ano remains competitive: GLUE average 82.92 versus 82.64 for Adam, and CIFAR-100 accuracy 70.31 versus 69.57. These are out-of-regime sanity checks, not evidence that Ano should replace Adam for supervised learning; full results are in Appendix E.

4.3 Are the gains robust to hyperparameters and protocol?

Hyperparameter sensitivity. Figure 4 shows a HalfCheetah/SAC grid on the 100k-step tuning proxy over three learning rates and β_1, β_2 values. At the lowest learning rate, Adam does not improve within the budget in any of its 9 configurations, consistent with its shorter steps (Section 2.1), whereas Ano makes progress in all of them; at $\mathrm{lr} = 3 \times 10^{-4}$, Ano's minimum exceeds Adam's maximum.

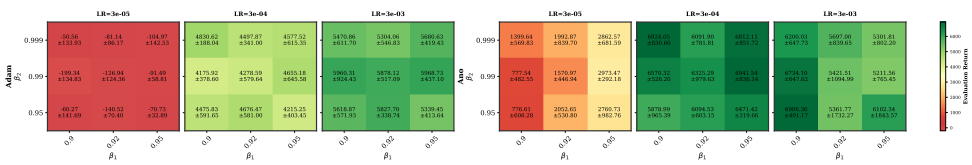


Figure 4: HalfCheetah/SAC hyperparameter grid on the 100k-step tuning proxy (5 seeds per cell).

Default versus tuned configurations. Table 1 uses default optimizer configurations. As a robustness check, Appendix D also reports tuned and best-of-default-or-tuned results under the homogeneous proxy-task protocol described in Appendix G; these tuned results do not define the main claim.

5 Ablation

Each component of Ano is tested in isolation on HalfCheetah/SAC (500k steps, deep RL), CIFAR-100 (vision), and MRPC (NLP). The three benchmarks span the noise spectrum: high variance and non-stationary in RL, low variance and stationary in supervised settings.

Variant	HalfCheetah ↑	CIFAR-100 ↑	MRPC ↑
Adam ($\lvert m\rvert/\sqrt{v}$, $\mathrm{sign}(m)$, Adam v_k)	$7481_{\pm 1323}$	$69.84_{\pm 0.22}$	$85.93_{\pm 0.92}$
SignumGrad ($\lvert g\rvert \cdot \mathrm{sign}(m)$, no v_k)	*div.*	$53.93_{\pm 0.68}$	*fail*
Grams ($\lvert m\rvert/\sqrt{v}$, $\mathrm{sign}(g)$, Adam v_k)	$5567_{\pm 782}$	$70.20_{\pm 0.17}$	$82.25_{\pm 0.74}$
Ano-EMA ($\lvert g\rvert/\sqrt{v}$, $\mathrm{sign}(m)$, Adam v_k)	$9855_{\pm 1173}$	$70.30_{\pm 0.38}$	$86.96_{\pm 0.85}$
AnoYogi ($\lvert g\rvert/\sqrt{v}$, $\mathrm{sign}(m)$, Yogi v_k)	$9053_{\pm 792}$	$\mathbf{70.32}_{\pm 1.20}$	$\mathbf{87.06}_{\pm 0.69}$
Ano ($\lvert g\rvert/\sqrt{v}$, $\mathrm{sign}(m)$, ACA v_k)	$\mathbf{10520}_{\pm 416}$	$69.74_{\pm 0.45}$	$86.76_{\pm 0.63}$

Table 2: Ablation results (mean ± 95% CI). *div.*: diverged (NaN). *fail*: collapsed to near-chance. Each variant isolates one design choice relative to Ano.

Three conclusions follow.

Normalization is required, and the ACA matters in RL. SignumGrad — $|g| \cdot \mathrm{sign}(m)$ without any second-moment estimate — diverges in deep RL. AnoYogi, which replaces the ACA with Yogi's accumulator, reaches 9053 versus 10520 for Ano ($+16\%$), while CIFAR-100 and MRPC are essentially unchanged: an arithmetic contraction does not substitute for the ACA's geometric one (Proposition 3.2).

Direction must come from the momentum. Grams takes its direction from $\mathrm{sign}(g)$ and its magnitude from Adam's $|m|/\sqrt{v}$, the opposite of Ano. It scores 5567 in RL, the lowest among non-diverging variants: it keeps Adam's noise-dependent step shrinkage while using the noisier instantaneous sign.

Both components contribute; the magnitude contributes most. Replacing Adam's magnitude $|m|$ by $|g|$ (Ano-EMA) gains $+32\%$ over Adam in RL; replacing Adam's EMA by the ACA adds a further $+6.7\%$. On this single environment the Ano-EMA and Ano intervals overlap, so this ablation is consistent with, but does not by itself establish, the accumulator's contribution. Neither change moves supervised scores by more than about one point.

6 Stationarity Guarantee

The experiments above evaluate Ano in the regime it is designed for: noisy, non-stationary objectives. Here we ask a narrower question: whether the same optimizer family admits a stationarity guarantee in the stationary stochastic nonconvex setting.

We analyze *Scheduled Capped Ano*, a proof-oriented variant of the practical optimizer. It adds a finite cap $V_{\max}$ on the accumulator and follows decaying schedules. These devices are used only for the analysis; empirically, capped and uncapped variants produce indistinguishable learning curves in our diagnostics (Appendix F.1).

Theorem 6.1 (Averaged stationarity for Scheduled Capped Ano)**.** *Assume that f is L-smooth and lower bounded, stochastic gradients are unbiased, and each coordinate has bounded conditional variance. Run Scheduled Capped Ano with current denominator, cap $V_{\max}$, $\eta_k = \eta/(k+2)^{3/4}$, $\beta_{1,k} = 1 - 1/\sqrt{k+1}$, and fixed $\beta_2 \in [0.5, 1)$. Then*

$$\frac{1}{K}\sum_{k=0}^{K-1} \mathbb{E}\big[\|\nabla f(x_k)\|^2\big] \le \tilde{\mathcal{O}}(K^{-1/4}).$$

Result	Object	Status
Theorem 2.3	Adam, RMSprop (EMA)	proved
Proposition 3.1	Ano-EMA	proved; $C_{\text{Ano-EMA}} > C_{\text{RMS}}$ numerical
Corollary B.6	exact ACA	proved
Lemma 2.5, Proposition 3.2	exact EMA / ACA, burst model	proved
Lemmas B.10–B.11	exact ACA	proved
Theorem 6.1	Scheduled Capped Ano	proved
Convergence	Algorithm 1	open
Replay, benchmarks	Algorithm 1	empirical

Table 3: Scope of the theoretical and empirical results.

The proof follows a smoothness descent argument. The main additional step is a coupled momentum-tracking estimate: the error between the momentum m_k and the true gradient $\nabla f(x_k)$ is controlled cumulatively and absorbed into the descent budget. This yields the $\beta_{1,k} = 1 - 1/\sqrt{k+1}$ schedule and the logarithmic factor. Full algorithmic details, assumptions, and proof are in Appendix C.

The theorem is a stationary sanity guarantee for the optimizer family, not a convergence analysis of the exact constant-hyperparameter optimizer used in the RL experiments. Extending it to the fully uncapped variant remains open: the cap is needed to convert self-normalized descent into Euclidean stationarity under bounded conditional variance. Appendix C.5 isolates this obstruction. Table 3 states which object each result applies to.

7 Related Work

Adaptive optimizers. Adam (Kingma & Ba, 2015; Reddi et al., 2018), RMSprop (Tieleman & Hinton, 2012), and AdaGrad (Duchi et al., 2011) are standard coordinatewise adaptive methods; AdamW (Loshchilov & Hutter, 2019) decouples weight decay. Yogi (Zaheer et al., 2018) uses sign-based second-moment updates to limit runaway accumulator growth. Ano targets the complementary post-burst regime: once the accumulator is too large, it should contract quickly.

Sign-based optimizers. SignSGD and Signum update with element-wise signs (Bernstein et al., 2018), and Lion (Chen et al., 2023) takes the sign of an interpolated momentum with a uniform step. Like Ano, these methods use momentum only for direction, but they discard gradient magnitudes and normalization; Lion underperforms in our RL experiments (Table 1). Balles & Hennig (2018) interpret Adam as sign descent with a variance-adapted magnitude that shrinks steps on noisy coordinates; our analysis quantifies the cost of this shrinkage when the objective moves (Section 2.1). Grams (Cao et al., 2025) also separates direction and magnitude, but assigns them the opposite way: $\operatorname{sign}(g_k)$ for direction and Adam's $|m_k|/\sqrt{v_k}$ for magnitude. It therefore keeps Adam's noise-dependent step shrinkage while taking its direction from the noisier instantaneous sign; Ano does the reverse.

Optimization in non-stationary settings. Ellis et al. (2024) address non-stationarity in RL by replacing Adam's global bias-correction timestep with a local timestep reset after target changes, while keeping the moment estimates themselves. This is designed to prevent large Adam updates after abrupt increases in gradient magnitude. In contrast, Ano changes how momentum enters the update and lets the second-moment estimate contract actively after variance bursts. Lyle et al. (2024) show that normalization and projection (NaP) can recover plasticity in RL by addressing dead units. These approaches operate at the architecture level; Ano instead operates at the optimizer level and requires no architectural change.

Positioning. Table 4 summarizes how Ano relates to the most closely connected optimizers along five axes.

8 Limitations

Removing damping is not free. Adam's noise-dependent step shrinkage is useful when the objective is fixed: near convergence, it acts as annealing. Ano removes it, so on stationary objectives it can keep moving where Adam would settle. For example, in Rosenbrock diagnostics, Ano reaches

	Direction	Magnitude	v_k rule	Active contraction	Memory
Adam	$\mathrm{sign}(m_k)$	$\lvert m_k\rvert/\sqrt{v_k}$	EMA(g^2)	✗	$2p$
RMSprop	$\mathrm{sign}(g_k)$	$\lvert g_k\rvert/\sqrt{v_k}$	EMA(g^2)	✗	p
Lion	$\mathrm{sign}(m_k)$	uniform	–	✗	p
Grams	$\mathrm{sign}(g_k)$	$\lvert m_k\rvert/\sqrt{v_k}$	EMA(g^2)	✗	$2p$
Yogi	$\mathrm{sign}(m_k)$	$\lvert m_k\rvert/\sqrt{v_k}$	one-sided	✗	$2p$
Ano	$\mathrm{sign}(m_k)$	$\lvert g_k\rvert/\sqrt{v_k}$	ACA	✓	$2p$

Table 4: Comparison with closely related optimizers (p: number of parameters). Adam, Yogi, and Ano take their direction from the sign of the momentum; Ano differs from Adam only in the magnitude attached to that sign and in how its second-moment estimate contracts.

the optimum basin faster from most initializations but can show residual near-optimum oscillations under a constant learning rate (Appendix F.2); learning-rate decay reduces them. We therefore do not expect Ano to outperform Adam in late-stage fine-tuning of stationary objectives, and view it as targeted at noisy tracking regimes.

Controlled diagnostics do not fully model deep RL. The persistent-noise and burst analyses in Section 2 isolate optimizer-level mechanisms, but they are not complete models of deep RL training. Real RL adds bootstrapping, replay, exploration, policy drift, target-network updates, and architecture-dependent effects. Offline replay diagnostics on real SAC gradients reduce this gap by testing the mechanism on realistic gradient streams while holding the gradients fixed across optimizers, but they still do not capture the closed-loop interaction between optimizer, policy, replay buffer, and data distribution. End-to-end RL results therefore remain necessary for evaluating the practical effect.

Theory for the fully practical variant remains open. Our convergence result applies to a scheduled capped variant, not to the exact constant-hyperparameter uncapped optimizer used in experiments. The finite cap is used to convert self-normalized descent into Euclidean stationarity; without it, bounded conditional variance alone does not rule out rare large stochastic gradients that are saturated by the adaptive denominator. Appendix C.5 isolates this obstruction. Extending the guarantee to the fully uncapped, constant-hyperparameter optimizer remains open. More broadly, our mechanistic results are local statements under fixed gradient statistics, and the ordering $C_{\text{Ano-EMA}} > C_{\text{RMS}}$ is established numerically.

9 Conclusion

Adam-style optimizers are rational under stationarity: they spend the variance reduction bought by momentum on shorter steps, and they forget past variance slowly, both of which are harmless when more steps can compensate. Deep RL removes that compensation. Ano keeps Adam's coordinatewise directions, state, and hyperparameter set, but uses momentum to decide where to move rather than how far, and lets its second-moment estimate return to the current gradient scale in finite time. Its gains are largest where persistent noise and target drift interact, and smallest where the objective is stationary.[1]

AI use statement

Generative AI tools assisted us with translation, with writing proofs, with improving the readability of the manuscript, and with finding related literature; they were not used for any other task requiring disclosure. The authors checked all AI-assisted content, including every proof and reference, and take full responsibility for this work.

[1]The source code for reproducing the main experiments is available at `https://anonymous.4open.science/r/ano-experiments-D769/README.md`.

REPRODUCIBILITY STATEMENT

The source code for reproducing all main experiments is available in anonymized form at `https://anonymous.4open.science/r/ano-experiments-D769/README.md`; the repository includes the optimizer implementation, experiment scripts, and reproduction instructions. The optimizer's exact specification and default hyperparameters are given in Algorithm 1, and its design is fully described in Section 3. All experimental details needed for reproduction are provided in the appendix: the implementation and evaluation protocol (D.2), the default, tuned, and best-configuration conventions (D.3), the aggregation metrics with stratified-bootstrap confidence intervals (D.4), hyperparameter ranges (D.8), the full per-domain results and learning curves (D.6, D.7), and details of the supervised sanity-check experiments (E). Complete proofs of all theoretical claims appear in Appendices B and C.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems 34*, pp. 2244–2261, 2021.

Matthew Aitchison, Penny Sweetser, and Marcus Hutter. Atari-5: Distilling the arcade learning environment down to five games. In *International Conference on Machine Learning*, pp. 421–438, 2023.

Krizhevsky Alex. Learning multiple layers of features from tiny images. Technical report, Toronto, ON, Canada, 2009.

Lukas Balles and Philipp Hennig. Dissecting Adam: The sign, magnitude and variance of stochastic gradients. In *International Conference on Machine Learning*, pp. 404–413, 2018.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47: 253–279, 2013.

Jeremy Bernstein, Yu-Xiang Wang, Kamyar Azizzadenesheli, and Animashree Anandkumar. signSGD: Compressed optimisation for non-convex problems. In *International Conference on Machine Learning*, pp. 560–569, 2018.

Yang Cao, Xiaoyu Li, and Zhao Song. Grams: Gradient descent with adaptive momentum scaling. In *ICLR Workshop on Scalable Optimization for Efficient and Adaptive Foundation Models*, 2025.

Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, et al. Symbolic discovery of optimization algorithms. In *Advances in Neural Information Processing Systems 36*, pp. 49205–49233, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1*, pp. 4171–4186, 2019.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(7), 2011.

Benjamin Ellis, Matthew T Jackson, Andrei Lupu, Alexander D Goldie, Mattie Fellows, Shimon Whiteson, and Jakob N Foerster. Adam on local time: Addressing nonstationarity in rl with relative Adam timesteps. In *Advances in Neural Information Processing Systems 37*, pp. 134567–134590, 2024.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pp. 1861–1870, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Shengyi Huang, Rousslan Fernand Julien Dossa, Chang Ye, and Jeff Braga. Cleanrl: High-quality single-file implementations of deep reinforcement learning algorithms. *Journal of Machine Learning Research*, 23(274):1–18, 2022.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Clare Lyle, Zeyu Zheng, Khimya Khetarpal, James Martens, Hado van Hasselt, Razvan Pascanu, and Will Dabney. Normalization and effective learning rates in reinforcement learning. In *Advances in Neural Information Processing Systems 37*, pp. 106440–106473, 2024.

Marlos C Machado, Marc G Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. Revisiting the arcade learning environment: Evaluation protocols and open problems for general agents. *Journal of Artificial Intelligence Research*, 61:523–562, 2018.

Sam McCandlish, Jared Kaplan, Dario Amodei, and OpenAI Dota Team. An empirical model of large-batch training. *arXiv preprint arXiv:1812.06162*, 2018.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International Conference on Machine Learning*, pp. 1928–1937, 2016.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. In *International Conference on Learning Representations*, 2018.

Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. Coursera: Neural Networks for Machine Learning, 2012. Lecture 6.5, slide 29.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Mark Towers, Ariel Kwiatkowski, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Kallinteris Andreas, Markus Krimmel, Arjun KG, Rodrigo De Lazcano Perez-Vicente, J K Terry, Andrea Pierré, Sander V Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2025.

Hado van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. Deep reinforcement learning and the deadly triad. *arXiv preprint arXiv:1812.02648*, 2018.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019.

Jiayi Weng, Min Lin, Shengyi Huang, Bo Liu, Denys Makoviichuk, Viktor Makoviychuk, Zichen Liu, Yufan Song, Ting Luo, Yukun Jiang, et al. EnvPool: A highly parallel reinforcement learning environment execution engine. In *Advances in Neural Information Processing Systems 35*, pp. 22409–22421, 2022.

Ross Wightman, Hugo Touvron, and Hervé Jégou. ResNet strikes back: An improved training procedure in timm. In *NeurIPS Workshop on ImageNet: Past, Present, and Future*, 2021.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Manzil Zaheer, Sashank J. Reddi, Devendra Sachan, Satyen Kale, and Sanjiv Kumar. Adaptive methods for nonconvex optimization. In *Advances in Neural Information Processing Systems 31*, pp. 9793–9803, 2018.

Juntang Zhuang, Tommy Tang, Yifan Ding, Sekhar C Tatikonda, Nicha Dvornek, Xenophon Papademetris, and James Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In *Advances in Neural Information Processing Systems 33*, pp. 18795–18806, 2020.

A Appendix Roadmap

The appendix is organized as follows. Appendix B contains the complete proofs for the two optimizer-level mechanisms in Section 2. Appendix C gives the full averaged-stationarity proof for the scheduled capped current-denominator variant and explains why removing the cap is nontrivial under bounded conditional variance. Appendix D provides deep RL implementation details, aggregate metrics, full per-environment tables, and learning curves. Appendix E reports supervised sanity checks. Appendix F contains additional controlled diagnostics, and Appendix G describes the hyperparameter selection protocol.

B Mechanistic Proofs

This appendix contains complete proofs for the results of Section 2. Appendix B.1 proves Theorem 2.3. Appendix B.4 proves the accumulator upper bound used in Corollary B.8. Appendix B.6 proves Lemma 2.5 and Proposition 3.2.

Additional notation. For Adam-style methods, $\hat{v}_k$ denotes the bias-corrected second-moment estimate. We write $\cdot$ for element-wise multiplication. For RMSprop, we write the second-moment decay as β_2 for consistency, although it is often denoted α in implementations. At equilibrium under gradients with mean μ_i and variance σ_i^2, Adam's estimates satisfy $\hat{v}_{k,i} \to \mu_i^2 + \sigma_i^2$ and $\hat{m}_{k,i} \to \mu_i$ (Zhuang et al., 2020).

B.1 Signal-Alignment Collapse: Proof of Theorem 2.3

Optimizer convention. Throughout this appendix, Adam and RMSprop are first analyzed in their stationary EMA forms. In one coordinate,

$$m_k = \beta_1 m_{k-1} + (1-\beta_1) g_k, \tag{7}$$

$$v_k^{\mathrm{EMA}} = \beta_2 v_{k-1}^{\mathrm{EMA}} + (1-\beta_2) g_k^2. \tag{8}$$

The corresponding uncorrected updates are

$$\Delta x_k^{\mathrm{RMSprop}} = -\eta \, \frac{g_k}{\sqrt{v_k^{\mathrm{EMA}}} + \varepsilon}, \tag{9}$$

$$\Delta x_k^{\mathrm{Adam}} = -\eta \, \frac{m_k}{\sqrt{v_k^{\mathrm{EMA}}} + \varepsilon}. \tag{10}$$

Ano-EMA update considered below is

$$\Delta x_k^{\text{Ano-EMA}} = -\eta \, \frac{|g_k| \operatorname{sign}(m_k)}{\sqrt{v_k^{\mathrm{EMA}}} + \varepsilon}. \tag{11}$$

Here $\varepsilon \geq 0$ is fixed independently of σ. All EMAs are understood in the stationary regime obtained by unrolling the recursions to the infinite past.

The usual bias-corrected implementation is recovered, at any fixed iteration k, by replacing

$$m_k \mapsto \hat{m}_k := \frac{m_k}{1-\beta_1^k}, \qquad v_k^{\mathrm{EMA}} \mapsto \hat{v}_k^{\mathrm{EMA}} := \frac{v_k^{\mathrm{EMA}}}{1-\beta_2^k}.$$

These factors are deterministic for fixed k and therefore only rescale the leading high-noise constants. They do not change the σ^{-1} collapse.

Notation. By symmetry across coordinates, all computations below are for a single coordinate i; we drop the subscript i throughout. Define

$$X := \xi_k, \tag{12}$$

$$Z := (1-\beta_1)\sum_{j=0}^{\infty}\beta_1^j\,\xi_{k-j}, \tag{13}$$

$$A := (1-\beta_2)\sum_{j=0}^{\infty}\beta_2^j\,\xi_{k-j}, \tag{14}$$

$$Q := (1-\beta_2)\sum_{j=0}^{\infty}\beta_2^j\,\xi_{k-j}^2. \tag{15}$$

We also write

$$a_j := (1-\beta_1)\beta_1^j, \qquad c_j := (1-\beta_2)\beta_2^j.$$

The series defining Z and A converge in L^2 and almost surely, because their deterministic weights are square-summable. Moreover, $Q < \infty$ almost surely since

$$\mathbb{E}Q = (1-\beta_2)\sum_{j=0}^{\infty}\beta_2^j\mathbb{E}\xi_{k-j}^2 = 1.$$

Also $Q > 0$ almost surely, since $Q \geq c_0\xi_k^2$ and $c_0 = 1-\beta_2 > 0$.

Under Assumption 2.1, unrolling the EMA recursions from stationarity gives

$$g_k = \mu + \sigma X, \qquad m_k = \mu + \sigma Z, \qquad v_k^{\mathrm{EMA}} = \mu^2 + 2\mu\sigma A + \sigma^2 Q. \tag{16}$$

For $\sigma > 0$, set $t := \mu/\sigma$. Then

$$\frac{1}{\sqrt{v_k^{\mathrm{EMA}}}} = \frac{1}{\sigma\sqrt{Q + 2tA + t^2}}. \tag{17}$$

The signal-aligned step-size (1) reduces to a one-coordinate expectation. Indeed, since $\nabla F(x) = \mu\mathbf{1}_d$ and all coordinates are identically distributed,

$$S_\sigma^{\mathrm{alg}} = \frac{1}{\mu}\,\mathbb{E}\left[\frac{\mathrm{numerator}_k}{\sqrt{v_k^{\mathrm{EMA}}}}\right], \tag{18}$$

where the numerator is g_k for RMSprop, m_k for Adam, and $|g_k|\,\mathrm{sign}(m_k)$ for Ano-EMA.

A negative-moment bound. We shall use the following elementary consequence of the infinite weighted chi-square structure of Q.

Lemma B.1 (Negative moments of Q)**.** *For every $p > 0$,*

$$\mathbb{E}[Q^{-p}] < \infty.$$

Moreover, for every finite Gaussian polynomial P in the variables $\{\xi_{k-j}\}_{j\geq 0}$ and every $p > 0$,

$$\mathbb{E}\big[|P|\,Q^{-p}\big] < \infty.$$

Proof. Fix $p > 0$. Choose an integer $N > 2p$. Since

$$Q \geq \sum_{j=0}^{N-1} c_j\xi_{k-j}^2 \geq c_{N-1}\sum_{j=0}^{N-1}\xi_{k-j}^2,$$

we have

$$Q^{-p} \leq c_{N-1}^{-p}\left(\sum_{j=0}^{N-1}\xi_{k-j}^2\right)^{-p}.$$

The sum is a χ^2_N random variable, whose negative p-th moment is finite whenever $N > 2p$. This proves $\mathbb{E}[Q^{-p}] < \infty$.

For the second claim, apply Cauchy–Schwarz:

$$\mathbb{E}[|P|Q^{-p}] \le \left(\mathbb{E}P^2\right)^{1/2} \left(\mathbb{E}Q^{-2p}\right)^{1/2}.$$

Gaussian polynomials have finite moments of all orders, and the first part gives $\mathbb{E}Q^{-2p} < \infty$. □

Differentiating through the expectation. For RMSprop and Adam, the only nontrivial analytic point is differentiation under the expectation at $t = 0$.

Lemma B.2 (Differentiation lemma)**.** *Let Y be either X or Z, and define*

$$F_Y(t) := \mathbb{E}\left[\frac{Y + t}{\sqrt{Q + 2tA + t^2}}\right].$$

Then F_Y is differentiable at $t = 0$, and

$$F_Y'(0) = \mathbb{E}\Big[Q^{-1/2} - YAQ^{-3/2}\Big]. \tag{19}$$

Proof. For fixed realization of the Gaussian sequence, the map

$$t \mapsto \frac{Y + t}{\sqrt{Q + 2tA + t^2}}$$

is differentiable in a neighborhood of 0, because

$$Q + 2tA + t^2 = (1 - \beta_2) \sum_{j=0}^{\infty} \beta_2^j (\xi_{k-j} + t)^2$$

is positive almost surely for all sufficiently small t. Its pointwise derivative is

$$(Q + 2tA + t^2)^{-1/2} - (Y + t)(A + t)(Q + 2tA + t^2)^{-3/2}.$$

At $t = 0$, this gives the integrand in (19).

It remains to justify passing the derivative through the expectation. For $|t| \le t_0$ sufficiently small, the denominator can be bounded locally by the same type of weighted chi-square quantity. More precisely, using

$$Q + 2tA + t^2 = (1 - \beta_2) \sum_{j=0}^{\infty} \beta_2^j (\xi_{k-j} + t)^2,$$

and restricting to a finite block of N indices, one obtains lower bounds by noncentral Gaussian quadratic forms. These have finite negative moments of any prescribed order once N is chosen large enough. Combined with the fact that X, Z, A are Gaussian linear forms, Lemma B.1 gives an integrable envelope for the difference quotients in a neighborhood of 0. Hence dominated convergence applies, yielding (19). □

General strategy. For each optimizer, after factoring out σ, we obtain an expectation of the form $F(t)$ with $t = \mu/\sigma$. The global symmetry

$$(\xi_k, \xi_{k-1}, \ldots) \mapsto (-\xi_k, -\xi_{k-1}, \ldots)$$

implies that all odd integrands have zero expectation. Thus, for RMSprop and Adam, $F(0) = 0$. By Lemma B.2,

$$F(t) = tF'(0) + o(t).$$

Since $t = \mu/\sigma$ and $\mu \neq 0$ is fixed,

$$S_\sigma^{\mathrm{alg}} = \frac{1}{\mu} F(\mu/\sigma) = \frac{F'(0)}{\sigma} + o(\sigma^{-1}). \tag{20}$$

The asymptotic constant is therefore $C_{\mathrm{alg}} = F'(0)$.

Lemma B.3 (RMSprop signal-alignment constant)**.** *Under Assumption 2.1, as $\sigma \to \infty$,*

$$S_\sigma^{\mathrm{RMSprop}} = \frac{C_{\mathrm{RMS}}(\beta_2)}{\sigma} + o(\sigma^{-1}), \qquad C_{\mathrm{RMS}}(\beta_2) := \mathbb{E}\Big[Q^{-1/2} - XAQ^{-3/2}\Big]. \tag{21}$$

Moreover $C_{\mathrm{RMS}}(\beta_2) > 0$.

Proof. For RMSprop,

$$\frac{g_k}{\sqrt{v_k^{\mathrm{EMA}}}} = \frac{X+t}{\sqrt{Q+2tA+t^2}}.$$

Define

$$F_{\mathrm{RMS}}(t) := \mathbb{E}\left[\frac{X+t}{\sqrt{Q+2tA+t^2}}\right].$$

By global symmetry, $F_{\mathrm{RMS}}(0) = \mathbb{E}[XQ^{-1/2}] = 0$. Applying Lemma B.2 with $Y = X$ gives

$$F'_{\mathrm{RMS}}(0) = \mathbb{E}[Q^{-1/2}] - \mathbb{E}[XAQ^{-3/2}],$$

which gives (21) through (20).

It remains to prove positivity. Since

$$Q = \sum_{j=0}^{\infty} c_j \xi_{k-j}^2,$$

we have pointwise

$$Q^{-1/2} = \sum_{j=0}^{\infty} c_j \xi_{k-j}^2 Q^{-3/2}.$$

By monotone convergence,

$$\mathbb{E}[Q^{-1/2}] = \sum_{j=0}^{\infty} c_j \mathbb{E}[\xi_{k-j}^2 Q^{-3/2}].$$

Also,

$$XA = \sum_{j=0}^{\infty} c_j X \xi_{k-j}.$$

For $j \neq 0$, the integrand $X\xi_{k-j}Q^{-3/2}$ is odd under flipping only ξ_{k-j} while holding all other coordinates fixed. Therefore its expectation is zero. Thus

$$\mathbb{E}[XAQ^{-3/2}] = c_0 \mathbb{E}[X^2 Q^{-3/2}].$$

Consequently,

$$C_{\mathrm{RMS}}(\beta_2) = \sum_{j=1}^{\infty} c_j \mathbb{E}[\xi_{k-j}^2 Q^{-3/2}] > 0, \tag{22}$$

because $\beta_2 > 0$ follows from $\beta_1^2 < \beta_2$, hence at least one coefficient c_j with $j \geq 1$ is strictly positive. □

Lemma B.4 (Adam signal-alignment constant)**.** *Under Assumption 2.1, as $\sigma \to \infty$,*

$$S_\sigma^{\mathrm{Adam}} = \frac{C_{\mathrm{Adam}}(\beta_1, \beta_2)}{\sigma} + o(\sigma^{-1}), \qquad C_{\mathrm{Adam}}(\beta_1, \beta_2) := \mathbb{E}\Big[Q^{-1/2} - ZAQ^{-3/2}\Big]. \tag{23}$$

Equivalently,

$$C_{\mathrm{Adam}}(\beta_1, \beta_2) = \sum_{j=0}^{\infty} c_j (1 - a_j)\, \mathbb{E}[\xi_{k-j}^2 Q^{-3/2}] > 0. \tag{24}$$

Proof. For Adam,

$$\frac{m_k}{\sqrt{v_k^{\mathrm{EMA}}}} = \frac{Z+t}{\sqrt{Q+2tA+t^2}}.$$

Define

$$F_{\mathrm{Adam}}(t) := \mathbb{E}\left[\frac{Z+t}{\sqrt{Q+2tA+t^2}}\right].$$

By global symmetry, $F_{\mathrm{Adam}}(0) = \mathbb{E}[ZQ^{-1/2}] = 0$. Applying Lemma B.2 with $Y = Z$ gives

$$F'_{\mathrm{Adam}}(0) = \mathbb{E}[Q^{-1/2}] - \mathbb{E}[ZAQ^{-3/2}],$$

which proves (23) through (20).

We now derive the series representation. Since

$$Z = \sum_{j=0}^{\infty} a_j \xi_{k-j}, \qquad A = \sum_{j=0}^{\infty} c_j \xi_{k-j},$$

we have

$$ZA = \sum_{\ell,j\geq 0} a_\ell c_j \xi_{k-\ell}\xi_{k-j}.$$

For $\ell \neq j$, the integrand $\xi_{k-\ell}\xi_{k-j}Q^{-3/2}$ is odd under flipping only $\xi_{k-\ell}$, while Q is unchanged. Hence all off-diagonal terms have zero expectation. Therefore

$$\mathbb{E}[ZAQ^{-3/2}] = \sum_{j=0}^{\infty} a_j c_j \mathbb{E}[\xi_{k-j}^2 Q^{-3/2}].$$

As in the RMSprop proof,

$$\mathbb{E}[Q^{-1/2}] = \sum_{j=0}^{\infty} c_j \mathbb{E}[\xi_{k-j}^2 Q^{-3/2}].$$

Subtracting gives

$$C_{\mathrm{Adam}}(\beta_1,\beta_2) = \sum_{j=0}^{\infty} c_j(1-a_j)\mathbb{E}[\xi_{k-j}^2 Q^{-3/2}].$$

Finally, $0 \leq a_j \leq 1$ for all j, so every term in the sum is nonnegative. Moreover, since $\beta_1^2 < \beta_2$, we have $\beta_2 > 0$, hence $c_j > 0$ for some $j \geq 1$. For such an index, $a_j = (1-\beta_1)\beta_1^j < 1$, so $1 - a_j > 0$. Therefore the sum is strictly positive. □

Proof of Theorem 2.3. The RMSprop statement follows from Lemma B.3, and the Adam statement follows from Lemma B.4. In both cases, the constants are finite by Lemma B.1, strictly positive by the series representations above, and independent of σ. Therefore

$$S_\sigma^{\mathrm{RMSprop}} = \frac{C_{\mathrm{RMS}}(\beta_2)}{\sigma} + o(\sigma^{-1}), \qquad S_\sigma^{\mathrm{Adam}} = \frac{C_{\mathrm{Adam}}(\beta_1,\beta_2)}{\sigma} + o(\sigma^{-1}).$$

Since $\mu \neq 0$ is fixed, both signal-aligned effective step-sizes vanish as $\sigma \to \infty$. □

Lemma B.5 (Ano-EMA signal-alignment constant)**.** *Under the same stationary convention, as $\sigma \to \infty$,*

$$S_\sigma^{\text{Ano-EMA}} = \frac{C_{\text{Ano-EMA}}(\beta_1,\beta_2)}{\sigma} + o(\sigma^{-1}), \tag{25}$$

where

$$C_{\text{Ano-EMA}}(\beta_1,\beta_2) := \mathbb{E}\Big[\mathrm{sign}(X)\,\mathrm{sign}(Z)\,Q^{-1/2}\Big] + 2p_Z(0)\,\mathbb{E}\Big[|X|Q^{-1/2} \mid Z=0\Big] - \mathbb{E}\Big[|X|\,\mathrm{sign}(Z)AQ^{-3/2}\Big]. \tag{26}$$

Here p_Z denotes the density of the centered Gaussian variable Z. For $\beta_1 = 0.9$, $\beta_2 = 0.999$, Monte Carlo evaluation gives $C_{\text{Ano-EMA}} \approx 2.73$ and $C_{\mathrm{RMS}} \approx 1.00$. Whether $C_{\text{Ano-EMA}} > C_{\mathrm{RMS}}$ holds analytically for all $\beta_1 \in (0,1)$ remains open; numerical evidence across tested hyperparameters is given in Table 5.

Proof. For Ano-EMA,

$$\frac{|g_k|\,\mathrm{sign}(m_k)}{\sqrt{v_k^{\mathrm{EMA}}}} = \frac{|X+t|\,\mathrm{sign}(Z+t)}{\sqrt{Q+2tA+t^2}}, \qquad t := \mu/\sigma.$$

Define

$$F_{\mathrm{Ano}}(t) := \mathbb{E}\left[\frac{|X+t|\,\mathrm{sign}(Z+t)}{\sqrt{Q+2tA+t^2}}\right].$$

By the global symmetry $(\xi_k, \xi_{k-1}, \ldots) \mapsto (-\xi_k, -\xi_{k-1}, \ldots)$, we have

$$F_{\mathrm{Ano}}(0) = \mathbb{E}\Big[|X|\,\mathrm{sign}(Z)Q^{-1/2}\Big] = 0.$$

We now compute the derivative at zero. Let

$$H(t,z) := \mathbb{E}\left[\frac{|X+t|}{\sqrt{Q+2tA+t^2}}\,\middle|\, Z=z\right].$$

Since Z is a centered Gaussian with variance

$$\tau^2 = (1-\beta_1)^2\sum_{j=0}^{\infty}\beta_1^{2j} = \frac{1-\beta_1}{1+\beta_1},$$

it has density

$$p_Z(z) = \frac{1}{\tau\sqrt{2\pi}}\exp\left(-\frac{z^2}{2\tau^2}\right).$$

Therefore

$$F_{\mathrm{Ano}}(t) = \int_{\mathbb{R}} H(t,z)\,\mathrm{sign}(z+t)p_Z(z)\,dz.$$

Equivalently,

$$F_{\mathrm{Ano}}(t) = \int_{-t}^{\infty} H(t,z)p_Z(z)\,dz - \int_{-\infty}^{-t} H(t,z)p_Z(z)\,dz.$$

Differentiating this expression at $t=0$ gives

$$F'_{\mathrm{Ano}}(0) = \int_{\mathbb{R}} \partial_t H(0,z)\,\mathrm{sign}(z)p_Z(z)\,dz + 2H(0,0)p_Z(0). \tag{27}$$

The boundary term appears because the discontinuity of $\mathrm{sign}(z+t)$ crosses the point $z=-t$.

For the first term, differentiating the smooth part of the integrand gives

$$\partial_t\left[\frac{|X+t|}{\sqrt{Q+2tA+t^2}}\right]_{t=0} = \mathrm{sign}(X)Q^{-1/2} - |X|AQ^{-3/2}.$$

Hence

$$\begin{aligned}\int_{\mathbb{R}} \partial_t H(0,z)\,\mathrm{sign}(z)p_Z(z)\,dz = \mathbb{E}\Big[\mathrm{sign}(X)\,\mathrm{sign}(Z)Q^{-1/2}\Big] \\ - \mathbb{E}\Big[|X|\,\mathrm{sign}(Z)AQ^{-3/2}\Big].\end{aligned} \tag{28}$$

The boundary term is

$$2H(0,0)p_Z(0) = 2p_Z(0)\mathbb{E}\Big[|X|Q^{-1/2} \mid Z=0\Big].$$

Combining this identity with (27) and (28) yields (26).

The differentiations above are justified as follows. Conditional on $Z=z$ in a compact neighborhood of zero, the remaining Gaussian sequence is still Gaussian on the corresponding affine hyperplane. The quadratic form

$$Q+2tA+t^2 = (1-\beta_2)\sum_{j=0}^{\infty}\beta_2^j(\xi_{k-j}+t)^2$$

has finite negative moments of all orders locally in t and z, by the same finite-block argument as Lemma B.1, applied under the conditional Gaussian law. These bounds dominate the difference quotients for the smooth part. The boundary term is handled explicitly by the preceding one-dimensional integral representation in the variable z. Therefore F_{Ano} is differentiable at zero with the derivative computed above.

Finally, since $F_{\mathrm{Ano}}(0) = 0$ and $F_{\mathrm{Ano}}(t) = tF'_{\mathrm{Ano}}(0) + o(t)$, with $t = \mu/\sigma$, we obtain

$$S_\sigma^{\text{Ano-EMA}} = \frac{1}{\mu} F_{\mathrm{Ano}}(\mu/\sigma) = \frac{F'_{\mathrm{Ano}}(0)}{\sigma} + o(\sigma^{-1}).$$

Thus

$$C_{\text{Ano-EMA}} = F'_{\mathrm{Ano}}(0),$$

which proves the claimed expansion. □

Numerical values. Table 5 reports Monte Carlo estimates of the three constants for standard hyperparameter values (10^6 samples).

β_1	β_2	C_{RMS}	C_{Adam}	$C_{\text{Ano-EMA}}$	$C_{\text{Ano-EMA}}/C_{\mathrm{RMS}}$
0.90	0.900	0.948	0.988	2.654	2.80
0.92	0.900	0.948	0.994	2.918	3.08
0.95	0.900	0.948	1.005	3.892	4.11
0.90	0.990	0.994	0.995	2.714	2.73
0.92	0.990	0.994	0.995	2.969	2.99
0.95	0.990	0.994	0.996	3.941	3.97
0.90	0.999	1.000	1.000	2.726	2.73
0.92	0.999	1.000	1.000	2.983	2.98
0.95	0.999	1.000	1.000	3.964	3.96

Table 5: Monte Carlo estimates of signal-alignment constants (2M samples, 4 chains, burn-in 200k). $C_{\text{Ano-EMA}} > C_{\mathrm{RMS}}$ holds across all hyperparameter values; the ratio grows with β_1 and is approximately independent of β_2 for $\beta_2 \geq 0.9$.

B.2 Ano-EMA and Full Ano Comparison

Combining Lemma B.5 with Corollary B.6 gives, for the reported hyperparameters,

$$S_\sigma^{\text{RMSprop}} \lesssim S_\sigma^{\text{Ano-EMA}} \leq S_\sigma^{Ano}, \tag{29}$$

where the first inequality is supported by Monte Carlo evaluation (Table 5) and the second is proved below.

Corollary B.6 (Ano dominates Ano-EMA in expected signal alignment)**.** *Under Assumption 2.1, assume $\sigma > 0$, $\varepsilon > 0$, and $\beta_2 \in [\frac{1}{2}, 1)$. Suppose that $v_{0,i} = v_{0,i}^{\mathrm{EMA}} = 0$ for every coordinate i. Then, for every $k \geq 1$,*

$$S_\sigma^{Ano} \geq S_\sigma^{\text{Ano-EMA}},$$

where both signal-alignment quantities are computed with the bias-corrected denominators

$$\sqrt{\hat{v}_{k,i}} + \varepsilon, \qquad \sqrt{\hat{v}_{k,i}^{\mathrm{EMA}}} + \varepsilon,$$

with

$$\hat{v}_{k,i} := \frac{v_{k,i}}{1-\beta_2^k}, \qquad \hat{v}_{k,i}^{\mathrm{EMA}} := \frac{v_{k,i}^{\mathrm{EMA}}}{1-\beta_2^k}.$$

Proof. Fix a coordinate i and suppress the index i.

First, we record two deterministic properties of the accumulators. The Ano accumulator satisfies

$$v_k = \beta_2 v_{k-1} - (1-\beta_2)\,\mathrm{sign}(v_{k-1} - g_k^2) g_k^2,$$

with $v_0 = 0$. Hence, by induction, v_k is measurable with respect to the squared-gradient history $(g_k^2, g_{k-1}^2, \ldots)$. The same holds for the EMA accumulator, since

$$v_k^{\mathrm{EMA}} = \beta_2 v_{k-1}^{\mathrm{EMA}} + (1-\beta_2) g_k^2, \qquad v_0^{\mathrm{EMA}} = 0.$$

Moreover, by Lemma B.10,

$$0 \le v_k \le v_k^{\mathrm{EMA}} \qquad \text{a.s.}$$

Since $k \ge 1$, we have $1 - \beta_2^k > 0$, so

$$0 \le \hat{v}_k \le \hat{v}_k^{\mathrm{EMA}} \qquad \text{a.s.}$$

Define

$$R := \frac{1}{\sqrt{\hat{v}_k} + \varepsilon} - \frac{1}{\sqrt{\hat{v}_k^{\mathrm{EMA}}} + \varepsilon}.$$

Because $x \mapsto (\sqrt{x} + \varepsilon)^{-1}$ is nonincreasing on $[0, \infty)$ and $\varepsilon > 0$, the previous inequality gives

$$R \ge 0 \qquad \text{a.s.}$$

Also, since both accumulators are measurable with respect to the squared-gradient history, R is measurable with respect to the gradient magnitudes.

Now set

$$Y_j := \frac{g_{k-j}}{\sigma} = t + \xi_{k-j}, \qquad t := \frac{\mu}{\sigma} \neq 0.$$

Then

$$g_k = \sigma Y_0,$$

and

$$m_k = \sigma \sum_{j=0}^{\infty} a_j Y_j, \qquad a_j = (1-\beta_1)\beta_1^j \ge 0.$$

Let $\mathcal{M}$ denote the sigma-field generated by the magnitudes $(|Y_j|)_{j \ge 0}$. The variable R is $\mathcal{M}$-measurable.

We need to show

$$\frac{1}{\mu} \mathbb{E}\left[|g_k| \operatorname{sign}(m_k) R \right] \ge 0.$$

Since $\sigma > 0$ and $\operatorname{sign}(t) = \operatorname{sign}(\mu)$, it is enough to prove

$$\operatorname{sign}(t) \mathbb{E}\left[|Y_0| R \operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j \right) \right] \ge 0.$$

The factor $|Y_0|R$ is nonnegative and $\mathcal{M}$-measurable, so

$$\mathbb{E}\left[|Y_0| R \operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j \right) \right] = \mathbb{E}\left[|Y_0| R \mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j \right) \mid \mathcal{M} \right] \right].$$

Thus it is enough to prove

$$\operatorname{sign}(t) \mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j \right) \mid \mathcal{M} \right] \ge 0 \qquad \text{a.s.}$$

Write

$$Y_j = s_j r_j, \qquad r_j := |Y_j|, \qquad s_j \in \{-1, +1\}.$$

Conditional on r_j, since $Y_j \sim \mathcal{N}(t, 1)$,

$$p_j(r_j) := \mathbb{P}(s_j = +1 \mid r_j) = \frac{e^{t r_j}}{e^{t r_j} + e^{-t r_j}}.$$

Hence $p_j(r_j) \geq 1/2$ if $t > 0$, and $p_j(r_j) \leq 1/2$ if $t < 0$. Conditional on the magnitudes, the signs $(s_j)_{j\geq 0}$ are independent.

We first consider the case $t > 0$. Couple these biased signs with fair Rademacher signs as follows. Let $(U_j)_{j\geq 0}$ be independent uniform random variables on $[0, 1]$ and define

$$s_j^+ = +1 \quad \Longleftrightarrow \quad U_j \leq p_j(r_j),$$

and

$$\eta_j = +1 \quad \Longleftrightarrow \quad U_j \leq \frac{1}{2}.$$

Then $s_j^+ \geq \eta_j$ for every j, since $p_j(r_j) \geq 1/2$.

For fixed magnitudes $(r_j)_{j\geq 0}$, the series

$$\sum_{j=0}^{\infty} a_j r_j$$

is finite a.s., because

$$\mathbb{E}\left[\sum_{j=0}^{\infty} a_j |Y_j|\right] = \sum_{j=0}^{\infty} a_j \mathbb{E}[|Y_j|] = \mathbb{E}[|Y_0|] < \infty.$$

Therefore

$$\sum_{j=0}^{\infty} a_j r_j s_j$$

is absolutely convergent.

The map

$$s \mapsto \operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j s_j\right)$$

is coordinatewise nondecreasing, since each coefficient $a_j r_j$ is nonnegative and we use the convention $\operatorname{sign}(0) = 0$. Hence the coupling gives

$$\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j s_j^+\right) \geq \operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j \eta_j\right).$$

Taking conditional expectations given $\mathcal{M}$ yields

$$\mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j\right) \mid \mathcal{M}\right] \geq \mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j \eta_j\right) \mid \mathcal{M}\right].$$

The right-hand side is zero. Indeed, conditional on $\mathcal{M}$, the law of $(\eta_j)_{j\geq 0}$ is invariant under the global flip $\eta \mapsto -\eta$, while

$$\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j (-\eta_j)\right) = -\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j r_j \eta_j\right).$$

Thus, for $t > 0$,

$$\mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j\right) \mid \mathcal{M}\right] \geq 0.$$

If $t < 0$, apply the previous argument to the variables $-Y_j$, whose mean is $-t > 0$. Since

$$\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j (-Y_j)\right) = -\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j\right),$$

we obtain

$$\mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j\right) \mid \mathcal{M}\right] \leq 0.$$

Therefore, in both cases,

$$\operatorname{sign}(t)\mathbb{E}\left[\operatorname{sign}\left(\sum_{j=0}^{\infty} a_j Y_j\right) \mid \mathcal{M}\right] \geq 0.$$

Consequently,

$$\operatorname{sign}(\mu)\mathbb{E}\left[|g_k|\operatorname{sign}(m_k)\left(\frac{1}{\sqrt{\hat{v}_k}+\varepsilon} - \frac{1}{\sqrt{\hat{v}_k^{\mathrm{EMA}}}+\varepsilon}\right)\right] \geq 0.$$

Since $\mu \neq 0$, this is equivalent to

$$\frac{1}{\mu}\mathbb{E}\left[|g_k|\operatorname{sign}(m_k)\left(\frac{1}{\sqrt{\hat{v}_k}+\varepsilon} - \frac{1}{\sqrt{\hat{v}_k^{\mathrm{EMA}}}+\varepsilon}\right)\right] \geq 0.$$

Equivalently,

$$\frac{1}{\mu}\mathbb{E}\left[\frac{|g_k|\operatorname{sign}(m_k)}{\sqrt{\hat{v}_k}+\varepsilon}\right] \geq \frac{1}{\mu}\mathbb{E}\left[\frac{|g_k|\operatorname{sign}(m_k)}{\sqrt{\hat{v}_k^{\mathrm{EMA}}}+\varepsilon}\right].$$

The expectations are finite because

$$\left|\frac{|g_k|\operatorname{sign}(m_k)}{\sqrt{\hat{v}_k}+\varepsilon}\right| \leq \frac{|g_k|}{\varepsilon},$$

and $\mathbb{E}|g_k| < \infty$ under the Gaussian noise assumption. The same bound applies to the EMA denominator.

Applying the argument to each coordinate, summing over i, and dividing by d gives

$$S_\sigma^{Ano} \geq S_\sigma^{\text{Ano-EMA}}.$$

□

B.3 Raw Update Magnitude

While $S_\sigma \to 0$ for all optimizers, the raw per-coordinate update magnitude does not vanish. Define

$$U_\sigma^{\mathrm{alg}} := \frac{\|\Delta x_k^{\mathrm{alg}}\|_2}{\eta\sqrt{d}}. \tag{30}$$

Theorem B.7 (Raw magnitude plateaus)**.** *Under Assumption 2.1, in the stationary EMA regime,*

$$\lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{Adam}}] = \mathbb{E}\left[\left(\frac{1}{d}\sum_{i=1}^{d}\frac{Z_i^2}{Q_i}\right)^{1/2}\right], \tag{31}$$

$$\lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{RMSprop}}] = \mathbb{E}\left[\left(\frac{1}{d}\sum_{i=1}^{d}\frac{X_i^2}{Q_i}\right)^{1/2}\right]. \tag{32}$$

In particular, for large d,

$$\lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{Adam}}] \approx \left(\mathbb{E}\left[\frac{Z^2}{Q}\right]\right)^{1/2}, \tag{33}$$

$$\lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{RMSprop}}] \approx \left(\mathbb{E}\left[\frac{X^2}{Q}\right]\right)^{1/2}. \tag{34}$$

When β_2 is close to 1, Q concentrates near 1, giving the approximations

$$\lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{Adam}}] \approx \sqrt{\frac{1-\beta_1}{1+\beta_1}}, \qquad \lim_{\sigma\to\infty} \mathbb{E}[U_\sigma^{\mathrm{RMSprop}}] \approx 1. \tag{35}$$

Both limiting plateaus are strictly positive and independent of μ.

Proof. Using the stationary decompositions (16), the squared normalized update magnitude for Adam is

$$(U_\sigma^{\mathrm{Adam}})^2 = \frac{1}{d}\sum_{i=1}^{d} \frac{m_{k,i}^2}{v_{k,i}^{\mathrm{EMA}}} = \frac{1}{d}\sum_{i=1}^{d} \frac{(\mu/\sigma + Z_i)^2}{\mu^2/\sigma^2 + 2\mu A_i/\sigma + Q_i}.$$

For each coordinate, as $\sigma \to \infty$,

$$\frac{(\mu/\sigma + Z_i)^2}{\mu^2/\sigma^2 + 2\mu A_i/\sigma + Q_i} \longrightarrow \frac{Z_i^2}{Q_i} \qquad \text{a.s.}$$

Therefore

$$U_\sigma^{\mathrm{Adam}} \longrightarrow \left(\frac{1}{d}\sum_{i=1}^{d} \frac{Z_i^2}{Q_i}\right)^{1/2} \qquad \text{a.s.}$$

The convergence also holds in expectation. Indeed, by weighted Cauchy–Schwarz,

$$Z^2 = \left(\sum_{j=0}^{\infty} a_j \xi_{k-j}\right)^2 \le \left(\sum_{j=0}^{\infty} \frac{a_j^2}{c_j}\right)\left(\sum_{j=0}^{\infty} c_j \xi_{k-j}^2\right) = \left(\sum_{j=0}^{\infty} \frac{a_j^2}{c_j}\right) Q.$$

The constant is finite because $\beta_1^2 < \beta_2$:

$$\sum_{j=0}^{\infty} \frac{a_j^2}{c_j} = \frac{(1-\beta_1)^2}{1-\beta_2} \sum_{j=0}^{\infty} \left(\frac{\beta_1^2}{\beta_2}\right)^j < \infty.$$

Thus Z^2/Q is bounded by a deterministic constant, and dominated convergence gives (31).

For RMSprop,

$$(U_\sigma^{\mathrm{RMSprop}})^2 = \frac{1}{d}\sum_{i=1}^{d} \frac{g_{k,i}^2}{v_{k,i}^{\mathrm{EMA}}} = \frac{1}{d}\sum_{i=1}^{d} \frac{(\mu/\sigma + X_i)^2}{\mu^2/\sigma^2 + 2\mu A_i/\sigma + Q_i}.$$

Hence, coordinatewise,

$$\frac{(\mu/\sigma + X_i)^2}{\mu^2/\sigma^2 + 2\mu A_i/\sigma + Q_i} \longrightarrow \frac{X_i^2}{Q_i} \qquad \text{a.s.}$$

Moreover, since $Q_i \ge c_0 X_i^2$ with $c_0 = 1 - \beta_2 > 0$, we have

$$\frac{X_i^2}{Q_i} \le \frac{1}{c_0}.$$

The same domination argument gives (32).

For large d, the coordinate terms are i.i.d., so the law of large numbers gives

$$\frac{1}{d}\sum_{i=1}^{d} \frac{Z_i^2}{Q_i} \approx \mathbb{E}\left[\frac{Z^2}{Q}\right], \qquad \frac{1}{d}\sum_{i=1}^{d} \frac{X_i^2}{Q_i} \approx \mathbb{E}\left[\frac{X^2}{Q}\right].$$

Finally, when β_2 is close to 1, the weighted average

$$Q = (1-\beta_2)\sum_{j=0}^{\infty} \beta_2^j \xi_{k-j}^2$$

concentrates near its mean $\mathbb{E}Q = 1$. Therefore

$$\mathbb{E}\left[\frac{Z^2}{Q}\right] \approx \mathbb{E}[Z^2] = (1-\beta_1)^2 \sum_{j=0}^{\infty} \beta_1^{2j} = \frac{1-\beta_1}{1+\beta_1},$$

and

$$\mathbb{E}\left[\frac{X^2}{Q}\right] \approx \mathbb{E}[X^2] = 1.$$

Taking square roots gives (35). □

Corollary B.8 (Ano raw magnitude dominates RMSprop)**.** *Assume the conditions of Lemma B.10. In particular, suppose $v_{0,i} \le v_{0,i}^{\mathrm{EMA}}$ for all i. Then, for every $k \ge 1$ and every $\sigma \ge 0$,*

$$U_\sigma^{Ano} \ge U_\sigma^{\mathrm{RMSprop}}.$$

The RMSprop plateau in (35) *is therefore a lower reference level for Ano's raw magnitude.*

Proof. By Lemma B.10,

$$v_{k,i} \le v_{k,i}^{\mathrm{EMA}} \qquad \text{for every coordinate } i.$$

Therefore,

$$\frac{1}{\sqrt{v_{k,i}}} \ge \frac{1}{\sqrt{v_{k,i}^{\mathrm{EMA}}}}$$

whenever the denominators are positive. Thus

$$|\Delta x_{k,i}^{Ano}| = \eta \frac{|g_{k,i}|}{\sqrt{v_{k,i}}} \ge \eta \frac{|g_{k,i}|}{\sqrt{v_{k,i}^{\mathrm{EMA}}}} = |\Delta x_{k,i}^{\mathrm{RMSprop}}|.$$

Squaring and summing over coordinates gives

$$\|\Delta x_k^{Ano}\|_2 \ge \|\Delta x_k^{\mathrm{RMSprop}}\|_2.$$

Dividing by $\eta\sqrt{d}$ yields the claim. □

Remark B.9. Taken together, Theorem B.7 and Theorem 2.3 characterize the full picture under persistent noise: the optimizer keeps moving, since U_σ remains bounded away from zero, but its motion becomes signal-free, since $S_\sigma \to 0$. The two quantities measure different effects: U_σ measures whether the optimizer is active, whereas S_σ measures whether the update is aligned with the signal.

As an illustration, in the same spirit as Figure 1a, Figure 5 shows the high-noise plateau of the normalized raw update magnitude. Although the signal-aligned component vanishes as σ increases, the raw update magnitude remains bounded away from zero.

B.4 Accumulator Domination

We establish the upper bound $v_{k,i} \le v_{k,i}^{\mathrm{EMA}}$ used in Corollaries B.6 and B.8 and Appendix C.

Lemma B.10 (ACA accumulator upper bound)**.** *If $v_{0,i} \le v_{0,i}^{\mathrm{EMA}}$ for all i, then $v_{k,i} \le v_{k,i}^{\mathrm{EMA}}$ for all $k \ge 0$ and all $i \in [d]$. Moreover, $v_{k,i} \ge 0$ for all $k \ge 0$, provided $\beta_2 \ge 1/2$.*

Proof. We proceed by induction on k.

Base case. Holds by assumption at $k = 0$.

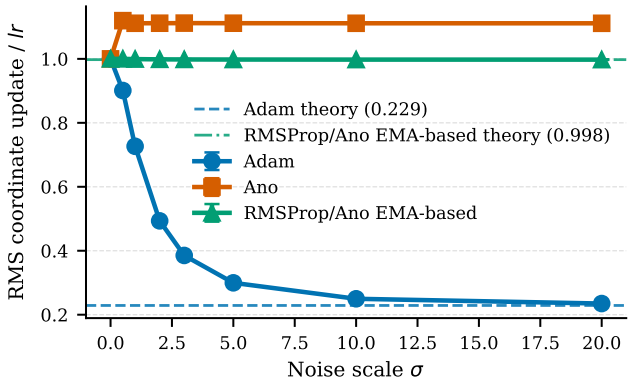


Figure 5: Normalized raw update magnitude $\|\Delta x_k\|_2/(\eta\sqrt{d})$ as a function of the noise scale σ. Adam converges to the lower plateau predicted by Theorem B.7, while RMSprop and the EMA-based Ano variant remain close to the unit plateau. Full Ano stays above the EMA-based baseline, consistent with Corollary B.8. This illustrates that, under persistent noise, optimizers may keep making large raw updates even when their signal-aligned component collapses.

Inductive step. Assume $v_{k-1,i} \le v_{k-1,i}^{\mathrm{EMA}}$.

Case $v_{k-1,i} \le g_{k,i}^2$: $\operatorname{sign}(v_{k-1,i} - g_{k,i}^2) = -1$, so the Ano update reduces to $v_{k,i} = \beta_2 v_{k-1,i} + (1-\beta_2)g_{k,i}^2 \le v_{k,i}^{\mathrm{EMA}}$ by the inductive hypothesis.

Case $v_{k-1,i} > g_{k,i}^2$: $\operatorname{sign}(v_{k-1,i} - g_{k,i}^2) = +1$, so $v_{k,i} = \beta_2 v_{k-1,i} - (1-\beta_2)g_{k,i}^2 \le \beta_2 v_{k-1,i} + (1-\beta_2)g_{k,i}^2 \le v_{k,i}^{\mathrm{EMA}}$.

In both cases $v_{k,i} \le v_{k,i}^{\mathrm{EMA}}$.

Non-negativity. In the first case, $v_{k,i}$ is a convex combination of non-negative terms. In the second case, $v_{k,i} = \beta_2 v_{k-1,i} - (1-\beta_2)g_{k,i}^2 \ge (2\beta_2 - 1)g_{k,i}^2 \ge 0$ since $\beta_2 \ge 1/2$.

□

B.5 Self-normalization of ACA

Lemma B.11 (Self-normalization of the uncapped current conditioner)**.** *Assume $\beta_2 > 1/2$. Consider the uncapped current ACA update*

$$v_{k,i} = \beta_2 v_{k-1,i} - (1-\beta_2)\operatorname{sign}(v_{k-1,i} - g_{k,i}^2)g_{k,i}^2.$$

Define

$$c_{\beta_2} := \min\{1-\beta_2,\, 2\beta_2 - 1\}.$$

Then $c_{\beta_2} > 0$, and for every coordinate $i \in [d]$,

$$v_{k,i} \ge c_{\beta_2} g_{k,i}^2.$$

Consequently, the normalized stochastic magnitude is uniformly bounded:

$$\frac{|g_{k,i}|}{\sqrt{v_{k,i}} + \varepsilon} \le \frac{1}{\sqrt{c_{\beta_2}}}.$$

In particular, the coordinate update satisfies

$$|x_{k+1,i} - x_{k,i}| = \eta_k \frac{|g_{k,i}|}{\sqrt{v_{k,i}} + \varepsilon} \le \frac{\eta_k}{\sqrt{c_{\beta_2}}},$$

and therefore

$$\|x_{k+1} - x_k\|^2 \le \frac{d\eta_k^2}{c_{\beta_2}}.$$

Proof. Fix $i \in [d]$. We distinguish two cases.

If $v_{k-1,i} \le g_{k,i}^2$, then

$$\operatorname{sign}(v_{k-1,i} - g_{k,i}^2) = -1,$$

and hence

$$v_{k,i} = \beta_2 v_{k-1,i} + (1-\beta_2) g_{k,i}^2 \ge (1-\beta_2) g_{k,i}^2.$$

If $v_{k-1,i} > g_{k,i}^2$, then

$$\operatorname{sign}(v_{k-1,i} - g_{k,i}^2) = 1,$$

and hence

$$v_{k,i} = \beta_2 v_{k-1,i} - (1-\beta_2) g_{k,i}^2.$$

Since $v_{k-1,i} > g_{k,i}^2$, we have

$$\beta_2 v_{k-1,i} > \beta_2 g_{k,i}^2.$$

Therefore,

$$v_{k,i} > (\beta_2 - (1-\beta_2))\, g_{k,i}^2 = (2\beta_2 - 1) g_{k,i}^2.$$

Combining the two cases gives

$$v_{k,i} \ge \min\{1-\beta_2,\, 2\beta_2 - 1\} g_{k,i}^2 = c_{\beta_2} g_{k,i}^2.$$

Since $\beta_2 > 1/2$, $c_{\beta_2} > 0$.

It follows that

$$\sqrt{v_{k,i}} \ge \sqrt{c_{\beta_2}}\, |g_{k,i}|.$$

Hence, if $g_{k,i} \neq 0$,

$$\frac{|g_{k,i}|}{\sqrt{v_{k,i}} + \varepsilon} \le \frac{|g_{k,i}|}{\sqrt{v_{k,i}}} \le \frac{1}{\sqrt{c_{\beta_2}}}.$$

If $g_{k,i} = 0$, the inequality is trivial.

Finally, from the update rule,

$$|x_{k+1,i} - x_{k,i}| = \eta_k \frac{|g_{k,i}|}{\sqrt{v_{k,i}} + \varepsilon},$$

because $|\operatorname{sign}(m_{k,i})| \le 1$. Therefore,

$$|x_{k+1,i} - x_{k,i}| \le \frac{\eta_k}{\sqrt{c_{\beta_2}}}.$$

Squaring and summing over coordinates yields

$$\|x_{k+1} - x_k\|^2 = \sum_{i=1}^{d} |x_{k+1,i} - x_{k,i}|^2 \le \frac{d\eta_k^2}{c_{\beta_2}}.$$

□

B.6 Post-Burst Recovery: Proofs of Lemma 2.5 and Proposition 3.2

All three derivations follow the same two-step pattern: solve the recursion in closed form under the sign stability condition $v_{k-1} > g_\infty^2$ (verified separately in each case); find the first integer k at which the closed-form solution crosses $(1+\delta)g_\infty^2$.

Case 1: Adam. The Adam update is $v_k = \beta_2 v_{k-1} + (1-\beta_2) g_\infty^2$. Define the residual $e_k := v_k - g_\infty^2$. Then $e_k = \beta_2\, e_{k-1}$, so by induction:

$$v_k^{\mathrm{Adam}} = g_\infty^2 + \beta_2^k (v_b - g_\infty^2). \tag{36}$$

The crossing condition $\beta_2^k (v_b - g_\infty^2) \le \delta g_\infty^2$ gives T_{Adam}^* as in (3).

Case 2: Ano. *Sign stability.* The sequence (37) is strictly decreasing (since $\beta_2 < 1$), so $v_{k-1} > (1+\delta)g_\infty^2 > g_\infty^2$ for all $k < T^*_{Ano}$.

Under $v_{k-1} > g_\infty^2$, the update reduces to $v_k = \beta_2 v_{k-1} - (1-\beta_2)g_\infty^2$. Define $e_k := v_k + g_\infty^2$: $e_k = \beta_2(v_{k-1} + g_\infty^2) = \beta_2\, e_{k-1}$, giving by induction:

$$v_k^{Ano} = \beta_2^k(v_b + g_\infty^2) - g_\infty^2. \tag{37}$$

Remark B.12. The fixed point $-g_\infty^2$ lies outside the admissible range; the recursion is only valid for $k < T^*_{Ano}$. After the burst episode ends, the sign flips and Ano converges monotonically to g_∞^2 from below.

The crossing condition $\beta_2^k(v_b + g_\infty^2) \le (2+\delta)g_\infty^2$ gives T^*_{Ano} as in (6).

Case 3: Yogi. *Sign stability.* By the same argument as Case 2, $v_{k-1} > g_\infty^2$ throughout, so $\operatorname{sign}(g_\infty^2 - v_{k-1}) = -1$.

The Yogi update $v_k = v_{k-1} - (1-\beta_2)g_\infty^2$ is arithmetic:

$$v_k^{\text{Yogi}} = v_b - k(1-\beta_2)g_\infty^2. \tag{38}$$

The crossing condition gives T^*_{Yogi} as in Lemma B.13.

Comparison $T^*_{Ano} \le T^*_{\text{Adam}}$. From (6) and (3), the inequality is equivalent to $(v_b + g_\infty^2)\delta \le (v_b - g_\infty^2)(2+\delta)$, i.e. $2v_b \ge (2+2\delta)g_\infty^2$, which holds by Assumption 2.4.

Lemma B.13 (Yogi recovery)**.** *Under Assumption 2.4,*

$$T^*_{\text{Yogi}} = \left\lceil \frac{v_b - (1+\delta)\, g_\infty^2}{(1-\beta_2)\, g_\infty^2} \right\rceil = \Theta\big(v_b/g_\infty^2\big) \quad \textit{as } v_b/g_\infty^2 \to \infty,$$

whereas $T^*_{\text{Adam}}, T^*_{Ano} = \mathcal{O}\big(\log(v_b/g_\infty^2)\big)$.

Remark B.14 (Yogi comparison). When $v_{k-1} > g_\infty^2$, Yogi's update reduces to a fixed arithmetic decrement with no geometric factor, which explains its linear dependence on the burst size. The ACA keeps Yogi's regime switch but retains Adam's geometric factor.

Remark B.15 (Size of the ACA advantage). From (3) and (6), up to rounding,

$$T^*_{\text{Adam}} - T^*_{Ano} = \frac{1}{\log(1/\beta_2)} \log\left(\frac{(v_b - g_\infty^2)(2+\delta)}{(v_b + g_\infty^2)\,\delta} \right),$$

which increases with v_b/g_∞^2 and saturates at $\log(1 + 2/\delta)/\log(1/\beta_2)$. The ratio $T^*_{\text{Adam}}/T^*_{Ano}$ therefore tends to 1 for very large bursts: the advantage of the ACA lies in the final approach to the post-burst scale, which the EMA completes only asymptotically.

C Stationarity Proof

This appendix gives the complete proof of Theorem 6.1. Section C.1 specifies the analyzed algorithm. Section C.3.1 introduces the key per-step quantities and derives the smooth descent inequality. Sections C.3.2–C.3.4 prove the auxiliary estimates. Section C.4 assembles the main result.

C.1 Analyzed Algorithm

We analyze the capped current-denominator variant. At each iteration, the stochastic gradient g_k is sampled, the momentum and second-moment accumulator are updated, and the parameter update uses the current accumulator v_k. The cap gives a uniform upper bound on the denominator, which is used to convert the adaptive descent estimate into Euclidean stationarity. For each coordinate $i \in [d]$, the updates are:

$$m_{k,i} = \beta_{1,k}\, m_{k-1,i} + (1-\beta_{1,k})\, g_{k,i}, \tag{39}$$

$$v_{k,i}^{\text{raw}} = \beta_2\, v_{k-1,i} - (1-\beta_2)\operatorname{sign}(v_{k-1,i} - g_{k,i}^2)\, g_{k,i}^2, \tag{40}$$

$$v_{k,i} = \min\big\{\max\{v_{k,i}^{\text{raw}},\, 0\},\, V_{\max}\big\}, \tag{41}$$

$$x_{k+1,i} = x_{k,i} - \frac{\eta_k}{\sqrt{v_{k,i}} + \varepsilon}\, |g_{k,i}|\, \operatorname{sign}(m_{k,i}), \tag{42}$$

with $m_{0,i} = v_{0,i} = 0$,

$$\beta_{1,k} = 1 - \frac{1}{\sqrt{k+1}}, \qquad \eta_k = \frac{\eta}{(k+2)^{3/4}}, \tag{43}$$

$\beta_2 \in [0.5, 1)$ fixed, $\varepsilon > 0$, and $V_{\max} > 0$. The projection in (41) ensures

$$0 \le v_{k,i} \le V_{\max} \qquad \forall k \ge 0,\ i \in [d].$$

Under $\beta_2 \ge 1/2$, the raw update already satisfies $v_{k,i}^{\mathrm{raw}} \ge 0$. Indeed, if $v_{k-1,i} \ge g_{k,i}^2$, then

$$v_{k,i}^{\mathrm{raw}} = \beta_2 v_{k-1,i} - (1-\beta_2) g_{k,i}^2 \ge (2\beta_2 - 1) g_{k,i}^2 \ge 0.$$

If $v_{k-1,i} < g_{k,i}^2$, then

$$v_{k,i}^{\mathrm{raw}} = \beta_2 v_{k-1,i} + (1-\beta_2) g_{k,i}^2 \ge 0.$$

Thus the lower projection in (41) is a harmless safeguard. We write

$$\mathbb{E}_{k-1}[\cdot] := \mathbb{E}[\cdot \mid \mathcal{F}_{k-1}].$$

C.2 Assumptions

We use the following assumptions throughout Appendix C.

Assumption C.1 (Smoothness)**.** $f : \mathbb{R}^d \to \mathbb{R}$ is differentiable and L-smooth:

$$\|\nabla f(x) - \nabla f(y)\| \le L\|x - y\| \qquad \forall x, y \in \mathbb{R}^d.$$

Assumption C.2 (Lower boundedness)**.** f is bounded below: there exists $f^\star > -\infty$ such that

$$f(x) \ge f^\star \qquad \forall x \in \mathbb{R}^d.$$

Assumption C.3 (Unbiased stochastic gradients)**.** At each iteration k, the stochastic gradient $g_k \in \mathbb{R}^d$ satisfies

$$\mathbb{E}[g_{k,i} \mid \mathcal{F}_{k-1}] = \nabla_i f(x_k) \qquad \forall i \in [d],$$

where $\mathcal{F}_{k-1}$ is the σ-algebra generated by all randomness up to step $k-1$.

Assumption C.4 (Bounded conditional variance)**.** There exists $\sigma > 0$ such that, for every $i \in [d]$ and $k \ge 1$,

$$\mathbb{E}\Big[(g_{k,i} - \nabla_i f(x_k))^2 \mid \mathcal{F}_{k-1}\Big] \le \sigma^2.$$

Remark C.5. Assumption C.4 is weaker than a uniformly bounded conditional second moment. Indeed, it allows $\mathbb{E}_{k-1}[g_{k,i}^2]$ to grow with $\nabla_i f(x_k)^2$. This creates a coupling between the momentum tracking error and the stationarity measure, which is handled through the cumulative tracking budget in Lemma C.11.

C.3 Auxiliary Lemmas

C.3.1 Auxiliary Quantities and Smooth Descent

Recall that $0 \le v_{k,i} \le V_{\max}$ for all $k \ge 0$ and $i \in [d]$ by (41). Define the two per-step scalars

$$A_k := \sum_{i=1}^{d} \frac{\nabla_i f(x_k)\, |g_{k,i}|\, \operatorname{sign}(m_{k,i})}{\sqrt{v_{k,i}} + \varepsilon}, \qquad B_k := \frac{L}{2} \sum_{i=1}^{d} \frac{g_{k,i}^2}{(\sqrt{v_{k,i}} + \varepsilon)^2}. \tag{44}$$

Lemma C.6 (Smooth descent)**.** *Under Assumption C.1, for all $k \ge 0$,*

$$f(x_{k+1}) \le f(x_k) - \eta_k A_k + \eta_k^2 B_k. \tag{45}$$

Proof. By L-smoothness,

$$f(x_{k+1}) \le f(x_k) + \langle \nabla f(x_k), x_{k+1} - x_k \rangle + \frac{L}{2}\|x_{k+1} - x_k\|^2.$$

From (42),

$$x_{k+1,i} - x_{k,i} = -\frac{\eta_k |g_{k,i}| \operatorname{sign}(m_{k,i})}{\sqrt{v_{k,i}} + \varepsilon}.$$

Therefore,

$$\langle \nabla f(x_k), x_{k+1} - x_k \rangle = -\eta_k A_k,$$

and

$$\frac{L}{2}\|x_{k+1} - x_k\|^2 = \eta_k^2 B_k.$$

This proves (45). □

C.3.2 COUPLED MOMENTUM TRACKING

Define

$$G_k := \mathbb{E}[\|\nabla f(x_k)\|^2],$$

and, for each coordinate,

$$e_{k,i} := m_{k,i} - \nabla_i f(x_k), \qquad W_{k,i} := \mathbb{E}[e_{k,i}^2], \qquad \mathcal{W}_k := \sum_{i=1}^{d} W_{k,i}.$$

Lemma C.7 (Coupled momentum tracking)**.** *Under Assumptions C.1, C.3, and C.4, with updates* (39)–(42) *and schedules* (43)*, for all* $k \geq 1$*,*

$$\mathcal{W}_k \leq \left(1 - \frac{1}{\sqrt{k+1}}\right)\mathcal{W}_{k-1} + \frac{C_1}{k+1} + \frac{C_2}{k+1}G_{k-1}, \tag{46}$$

where one may take

$$C_1 := d\sigma^2 + \frac{2d^2L^2\eta^2\sigma^2}{\varepsilon^2}, \qquad C_2 := \frac{2dL^2\eta^2}{\varepsilon^2}.$$

Proof. Fix $i \in [d]$. Define

$$\Delta_{k,i} := \nabla_i f(x_{k-1}) - \nabla_i f(x_k), \qquad \xi_{k,i} := g_{k,i} - \nabla_i f(x_k).$$

Subtracting $\nabla_i f(x_k)$ from (39) gives

$$e_{k,i} = \beta_{1,k}e_{k-1,i} + \beta_{1,k}\Delta_{k,i} + (1-\beta_{1,k})\xi_{k,i}. \tag{47}$$

The random variable $e_{k-1,i} + \Delta_{k,i}$ is $\mathcal{F}_{k-1}$-measurable, and

$$\mathbb{E}_{k-1}[\xi_{k,i}] = 0$$

by Assumption C.3. Therefore, after conditioning on $\mathcal{F}_{k-1}$, the cross-term with $\xi_{k,i}$ vanishes:

$$W_{k,i} = \beta_{1,k}^2\mathbb{E}[(e_{k-1,i} + \Delta_{k,i})^2] + (1-\beta_{1,k})^2\mathbb{E}[\xi_{k,i}^2].$$

By Assumption C.4,

$$\mathbb{E}[\xi_{k,i}^2] \leq \sigma^2.$$

Let $l := k+1$ and choose $\gamma_k = l^{-1/2}$. Young's inequality gives

$$(e_{k-1,i} + \Delta_{k,i})^2 \leq (1+\gamma_k)e_{k-1,i}^2 + (1+\gamma_k^{-1})\Delta_{k,i}^2.$$

Hence

$$W_{k,i} \leq \beta_{1,k}^2(1+\gamma_k)W_{k-1,i} + \beta_{1,k}^2(1+\gamma_k^{-1})\mathbb{E}[\Delta_{k,i}^2] + \frac{\sigma^2}{l}. \tag{48}$$

Using $\beta_{1,k} = 1 - l^{-1/2}$,

$$\beta_{1,k}^2(1+\gamma_k) = \left(1 - \frac{1}{\sqrt{l}}\right)^2\left(1 + \frac{1}{\sqrt{l}}\right) = 1 - \frac{1}{\sqrt{l}} - \frac{1}{l} + \frac{1}{l^{3/2}} \leq 1 - \frac{1}{\sqrt{l}}.$$

It remains to control $\mathbb{E}[\Delta_{k,i}^2]$. By L-smoothness,

$$\Delta_{k,i}^2 \leq L^2\|x_k - x_{k-1}\|^2.$$

From (42),

$$\|x_k - x_{k-1}\|^2 = \eta_{k-1}^2\sum_{j=1}^{d}\frac{g_{k-1,j}^2}{(\sqrt{v_{k-1,j}} + \varepsilon)^2} \leq \frac{\eta_{k-1}^2}{\varepsilon^2}\sum_{j=1}^{d} g_{k-1,j}^2.$$

Under Assumptions C.3 and C.4,

$$\mathbb{E}[g_{k-1,j}^2] \leq \mathbb{E}[\nabla_j f(x_{k-1})^2] + \sigma^2.$$

Therefore

$$\mathbb{E}[\|g_{k-1}\|^2] \leq G_{k-1} + d\sigma^2.$$

Thus

$$\mathbb{E}[\Delta_{k,i}^2] \le \frac{L^2\eta_{k-1}^2}{\varepsilon^2}\left(G_{k-1} + d\sigma^2\right).$$

Since

$$\eta_{k-1} = \frac{\eta}{(k+1)^{3/4}} = \frac{\eta}{l^{3/4}},$$

and $1 + \gamma_k^{-1} = 1 + \sqrt{l}$, we have

$$(1+\sqrt{l})\eta_{k-1}^2 = (1+\sqrt{l})\frac{\eta^2}{l^{3/2}} \le \frac{2\eta^2}{l}.$$

Substituting into (48) yields

$$W_{k,i} \le \left(1 - \frac{1}{\sqrt{l}}\right) W_{k-1,i} + \frac{\sigma^2}{l} + \frac{2L^2\eta^2}{\varepsilon^2}\frac{G_{k-1} + d\sigma^2}{l}.$$

Summing over $i \in [d]$ gives

$$\mathcal{W}_k \le \left(1 - \frac{1}{\sqrt{k+1}}\right)\mathcal{W}_{k-1} + \frac{d\sigma^2}{k+1} + \frac{2dL^2\eta^2}{\varepsilon^2}\frac{G_{k-1} + d\sigma^2}{k+1}.$$

This is (46). □

C.3.3 Lower Bound on the Descent Term

Lemma C.8 (Lower bound on the descent term)**.** *Let*

$$D := \sqrt{V_{\max}} + \varepsilon.$$

Under Assumptions C.3 and C.4, for every $k \ge 1$,

$$\mathbb{E}[A_k] \ge \frac{1}{2D}G_k - C_A\mathcal{W}_k - C_\sigma\sqrt{\mathcal{W}_k}, \tag{49}$$

where

$$C_A := \frac{2D}{\varepsilon^2}, \qquad C_\sigma := \frac{2\sigma\sqrt{d}}{\varepsilon}.$$

Proof. For a fixed coordinate i, write

$$h_{k,i} := \nabla_i f(x_k), \qquad D_{k,i} := \sqrt{v_{k,i}} + \varepsilon, \qquad e_{k,i} := m_{k,i} - h_{k,i}.$$

By the cap,

$$\varepsilon \le D_{k,i} \le D.$$

If $h_{k,i} = 0$, the i-th contribution to A_k is zero. Otherwise, define

$$s_{k,i} := \operatorname{sign}(h_{k,i}), \qquad \chi_{k,i} := \mathbf{1}\{\operatorname{sign}(m_{k,i}) \ne s_{k,i}\}.$$

Then

$$h_{k,i}|g_{k,i}|\operatorname{sign}(m_{k,i}) = |h_{k,i}|\,|g_{k,i}|(1 - 2\chi_{k,i}).$$

Therefore,

$$\mathbb{E}[A_k] = \sum_{i=1}^{d}\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|}{D_{k,i}}\right] - 2\sum_{i=1}^{d}\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|\chi_{k,i}}{D_{k,i}}\right]. \tag{50}$$

We first lower bound the positive term. Since $D_{k,i} \le D$,

$$\frac{|h_{k,i}|\,|g_{k,i}|}{D_{k,i}} \ge \frac{|h_{k,i}|\,|g_{k,i}|}{D}.$$

Since $h_{k,i}$ is $\mathcal{F}_{k-1}$-measurable, Jensen's inequality and Assumption C.3 give

$$\mathbb{E}_{k-1}[|g_{k,i}|] \ge |\mathbb{E}_{k-1}[g_{k,i}]| = |h_{k,i}|.$$

Thus

$$\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|}{D_{k,i}}\right] \geq \frac{1}{D}\mathbb{E}[h_{k,i}^2].$$

Summing over i gives

$$\sum_{i=1}^{d}\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|}{D_{k,i}}\right] \geq \frac{G_k}{D}. \tag{51}$$

We now upper bound the mismatch term. On the event $\{\chi_{k,i}=1\}$,

$$|h_{k,i}|\chi_{k,i} \leq |m_{k,i}-h_{k,i}| = |e_{k,i}|.$$

Since $D_{k,i} \geq \varepsilon$,

$$\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|\chi_{k,i}}{D_{k,i}}\right] \leq \frac{1}{\varepsilon}\mathbb{E}[|g_{k,i}|\,|e_{k,i}|].$$

Write

$$g_{k,i} = h_{k,i} + \xi_{k,i}, \qquad \xi_{k,i} := g_{k,i} - h_{k,i}.$$

Then

$$\mathbb{E}[|g_{k,i}|\,|e_{k,i}|] \leq \mathbb{E}[|h_{k,i}|\,|e_{k,i}|] + \mathbb{E}[|\xi_{k,i}|\,|e_{k,i}|].$$

For any $\rho > 0$, Young's inequality gives

$$\mathbb{E}[|h_{k,i}|\,|e_{k,i}|] \leq \rho\,\mathbb{E}[h_{k,i}^2] + \frac{1}{4\rho}W_{k,i}.$$

By Cauchy–Schwarz and Assumption C.4,

$$\mathbb{E}[|\xi_{k,i}|\,|e_{k,i}|] \leq \sqrt{\mathbb{E}[\xi_{k,i}^2]}\sqrt{\mathbb{E}[e_{k,i}^2]} \leq \sigma\sqrt{W_{k,i}}.$$

Summing over i, and using

$$\sum_{i=1}^{d}\sqrt{W_{k,i}} \leq \sqrt{d\mathcal{W}_k},$$

we get

$$\sum_{i=1}^{d}\mathbb{E}\left[\frac{|h_{k,i}|\,|g_{k,i}|\chi_{k,i}}{D_{k,i}}\right] \leq \frac{\rho}{\varepsilon}G_k + \frac{1}{4\rho\varepsilon}\mathcal{W}_k + \frac{\sigma\sqrt{d}}{\varepsilon}\sqrt{\mathcal{W}_k}.$$

Combining with (50) and (51),

$$\mathbb{E}[A_k] \geq \left(\frac{1}{D} - \frac{2\rho}{\varepsilon}\right)G_k - \frac{1}{2\rho\varepsilon}\mathcal{W}_k - \frac{2\sigma\sqrt{d}}{\varepsilon}\sqrt{\mathcal{W}_k}.$$

Choose

$$\rho := \frac{\varepsilon}{4D}.$$

Then

$$\frac{1}{D} - \frac{2\rho}{\varepsilon} = \frac{1}{2D},$$

and

$$\frac{1}{2\rho\varepsilon} = \frac{2D}{\varepsilon^2}.$$

This proves (49). □

Remark C.9 (Why the current denominator is harmless under the cap). With the current denominator, $v_{k,i}$ depends on the current stochastic gradient $g_{k,i}$, so the reciprocal $(\sqrt{v_{k,i}}+\varepsilon)^{-1}$ is not $\mathcal{F}_{k-1}$-measurable. In the capped analysis this does not create a circularity in the lower bound on A_k, because the cap gives the pointwise bound

$$\frac{1}{\sqrt{v_{k,i}}+\varepsilon} \geq \frac{1}{\sqrt{V_{\max}}+\varepsilon}.$$

Thus the positive part of the descent can be bounded below without pulling the denominator outside the conditional expectation. The mismatch and smoothness-remainder terms only use the lower bound $\sqrt{v_{k,i}}+\varepsilon \geq \varepsilon$.

C.3.4 Upper Bound on the Smoothness Remainder

Lemma C.10 (Upper bound on the smoothness remainder)**.** *Under Assumptions C.3 and C.4, for all* $k \geq 1$,

$$\mathbb{E}[B_k] \leq C_B^{(g)} G_k + C_B^{(\sigma)}, \tag{52}$$

where

$$C_B^{(g)} := \frac{L}{2\varepsilon^2}, \qquad C_B^{(\sigma)} := \frac{Ld\sigma^2}{2\varepsilon^2}.$$

Proof. Since $(\sqrt{v_{k,i}} + \varepsilon)^2 \geq \varepsilon^2$,

$$B_k \leq \frac{L}{2\varepsilon^2} \sum_{i=1}^{d} g_{k,i}^2.$$

By Assumptions C.3 and C.4,

$$\mathbb{E}_{k-1}[g_{k,i}^2] = \nabla_i f(x_k)^2 + \mathbb{E}_{k-1}\left[(g_{k,i} - \nabla_i f(x_k))^2\right] \leq \nabla_i f(x_k)^2 + \sigma^2.$$

Summing over i and taking total expectation gives

$$\mathbb{E}[B_k] \leq \frac{L}{2\varepsilon^2}\left(G_k + d\sigma^2\right).$$

This proves the claim. □

C.3.5 Cumulative Tracking Budget

Lemma C.11 (Cumulative tracking budget)**.** *Suppose that* $\mathcal{W}_k$ *satisfies the recursion*

$$\mathcal{W}_k \leq \left(1 - \frac{1}{\sqrt{k+1}}\right)\mathcal{W}_{k-1} + \frac{C_1}{k+1} + \frac{C_2}{k+1} G_{k-1}.$$

Then, for every $K \geq 2$,

$$\sum_{k=1}^{K-1} \frac{\mathcal{W}_k}{\sqrt{k+1}} \leq \mathcal{W}_0 + 2C_1(1 + \log K) + 2C_2 \sum_{k=1}^{K-1} \frac{G_{k-1}}{k+1}. \tag{53}$$

Consequently, there exists a constant $C > 0$*, independent of* K*, such that*

$$\sum_{k=1}^{K-1} \eta_k \mathcal{W}_k \leq C(1 + \log K) + C \sum_{k=1}^{K-1} \frac{G_{k-1}}{k+1}, \tag{54}$$

$$\sum_{k=1}^{K-1} \eta_k \sqrt{\mathcal{W}_k} \leq C(1 + \log K) + C \sum_{k=1}^{K-1} \frac{G_{k-1}}{k+1}. \tag{55}$$

Proof. Let

$$a_k := \frac{1}{\sqrt{k+1}}, \qquad b_k := \frac{C_1}{k+1} + \frac{C_2}{k+1} G_{k-1}.$$

The recursion reads

$$\mathcal{W}_k \leq (1 - a_k)\mathcal{W}_{k-1} + b_k.$$

Rearranging gives

$$a_k \mathcal{W}_{k-1} \leq \mathcal{W}_{k-1} - \mathcal{W}_k + b_k.$$

Moreover,

$$\mathcal{W}_k \leq \mathcal{W}_{k-1} + b_k.$$

Since $a_k \leq 1$,

$$a_k \mathcal{W}_k \leq a_k \mathcal{W}_{k-1} + a_k b_k \leq \mathcal{W}_{k-1} - \mathcal{W}_k + 2b_k.$$

Summing from $k = 1$ to $K - 1$ gives

$$\sum_{k=1}^{K-1} a_k \mathcal{W}_k \leq \mathcal{W}_0 + 2 \sum_{k=1}^{K-1} b_k.$$

Since $a_k = (k+1)^{-1/2}$, this proves

$$\sum_{k=1}^{K-1} \frac{\mathcal{W}_k}{\sqrt{k+1}} \le \mathcal{W}_0 + 2C_1 \sum_{k=1}^{K-1} \frac{1}{k+1} + 2C_2 \sum_{k=1}^{K-1} \frac{G_{k-1}}{k+1}.$$

Using $\sum_{k=1}^{K-1} \frac{1}{k+1} \le 1 + \log K$ gives (53). Next, since

$$\eta_k = \frac{\eta}{(k+2)^{3/4}} \le C(k+1)^{-1/2},$$

we have

$$\sum_{k=1}^{K-1} \eta_k \mathcal{W}_k \le C \sum_{k=1}^{K-1} \frac{\mathcal{W}_k}{\sqrt{k+1}},$$

which proves (54) from (53). For the square-root term, use

$$\sqrt{x} \le \frac{r}{2}x + \frac{1}{2r} \qquad \forall x \ge 0,\ r > 0.$$

With $r_k = (k+1)^{1/4}$,

$$\eta_k \sqrt{\mathcal{W}_k} \le \frac{\eta_k r_k}{2} \mathcal{W}_k + \frac{\eta_k}{2r_k}.$$

Now

$$\eta_k r_k \le C(k+1)^{-1/2}, \qquad \frac{\eta_k}{r_k} \le \frac{C}{k+1}.$$

Therefore

$$\sum_{k=1}^{K-1} \eta_k \sqrt{\mathcal{W}_k} \le C \sum_{k=1}^{K-1} \frac{\mathcal{W}_k}{\sqrt{k+1}} + C \sum_{k=1}^{K-1} \frac{1}{k+1}.$$

Applying (53) again proves (55). □

C.4 Proof of Theorem 6.1

Proof of Theorem 6.1. Let

$$D := \sqrt{V_{\max}} + \varepsilon.$$

Taking expectations in the smooth descent inequality (45) and applying Lemmas C.8 and C.10, we obtain

$$\mathbb{E}[f(x_{k+1})] \le \mathbb{E}[f(x_k)] - \eta_k \left(\frac{1}{2D} G_k - C_A \mathcal{W}_k - C_\sigma \sqrt{\mathcal{W}_k} \right) + \eta_k^2 \left(C_B^{(g)} G_k + C_B^{(\sigma)} \right).$$

Equivalently,

$$\mathbb{E}[f(x_{k+1})] \le \mathbb{E}[f(x_k)] - \eta_k \left(\frac{1}{2D} - \eta_k C_B^{(g)} \right) G_k + \eta_k C_A \mathcal{W}_k + \eta_k C_\sigma \sqrt{\mathcal{W}_k} + \eta_k^2 C_B^{(\sigma)}.$$

Since $\eta_k \to 0$, there exists $k_1 \ge 1$, independent of K, such that for all $k \ge k_1$,

$$\eta_k C_B^{(g)} \le \frac{1}{4D}.$$

Thus, for all $k \ge k_1$,

$$\mathbb{E}[f(x_{k+1})] \le \mathbb{E}[f(x_k)] - \frac{\eta_k}{4D} G_k + \eta_k C_A \mathcal{W}_k + \eta_k C_\sigma \sqrt{\mathcal{W}_k} + \eta_k^2 C_B^{(\sigma)}. \tag{56}$$

Summing (56) from $k = k_1$ to $K-1$, and using $f(x_K) \ge f^\star$, gives

$$\begin{aligned} \frac{1}{4D} \sum_{k=k_1}^{K-1} \eta_k G_k \le \mathbb{E}[f(x_{k_1})] - f^\star + C_A \sum_{k=k_1}^{K-1} \eta_k \mathcal{W}_k \\ + C_\sigma \sum_{k=k_1}^{K-1} \eta_k \sqrt{\mathcal{W}_k} + C_B^{(\sigma)} \sum_{k=k_1}^{K-1} \eta_k^2. \end{aligned} \tag{57}$$

By Lemma C.11,

$$\sum_{k=k_1}^{K-1} \eta_k \mathcal{W}_k + \sum_{k=k_1}^{K-1} \eta_k \sqrt{\mathcal{W}_k} \le C(1+\log K) + C \sum_{k=k_1}^{K-1} \frac{G_{k-1}}{k+1}.$$

Also,

$$\sum_{k=k_1}^{K-1} \eta_k^2 = \eta^2 \sum_{k=k_1}^{K-1} (k+2)^{-3/2} \le \eta^2 S_{3/2},$$

where

$$S_{3/2} := \sum_{k=0}^{\infty} (k+2)^{-3/2} < \infty.$$

Therefore,

$$\frac{1}{4D} \sum_{k=k_1}^{K-1} \eta_k G_k \le C(1+\log K) + C \sum_{k=k_1}^{K-1} \frac{G_{k-1}}{k+1}. \tag{58}$$

It remains to absorb the last term. The finite first term $G_{k_1-1}/(k_1+1)$ can be absorbed into the constant. For $k \ge k_1+1$, set $t = k-1$. Then

$$\frac{G_{k-1}}{k+1} = \frac{G_t}{t+2}.$$

Since

$$\eta_t = \frac{\eta}{(t+2)^{3/4}},$$

we have

$$\frac{1}{t+2} = \frac{1}{\eta}(t+2)^{-1/4}\eta_t.$$

Hence, by increasing k_1 if necessary, we may ensure that for all $t \ge k_1$,

$$C\frac{1}{t+2} \le \frac{1}{8D}\eta_t.$$

Therefore,

$$C \sum_{k=k_1}^{K-1} \frac{G_{k-1}}{k+1} \le C + \frac{1}{8D} \sum_{t=k_1}^{K-2} \eta_t G_t.$$

Substituting into (58) and absorbing the last term into the left-hand side yields

$$\sum_{k=k_1}^{K-1} \eta_k G_k \le C(1+\log K), \tag{59}$$

for a constant $C > 0$ independent of K. Finally, since η_k is decreasing, for every $k \le K-1$,

$$\eta_k \ge \eta_{K-1} = \frac{\eta}{(K+1)^{3/4}}.$$

Thus

$$\eta_{K-1} \sum_{k=k_1}^{K-1} G_k \le \sum_{k=k_1}^{K-1} \eta_k G_k \le C(1+\log K),$$

and therefore

$$\sum_{k=k_1}^{K-1} G_k \le \frac{C}{\eta}(K+1)^{3/4}(1+\log K).$$

Dividing by K gives

$$\frac{1}{K} \sum_{k=k_1}^{K-1} \mathbb{E}[\|\nabla f(x_k)\|^2] = O\left(\frac{1+\log K}{K^{1/4}}\right).$$

The finite prefix $0 \le k < k_1$ contributes only $O(K^{-1})$ to the average, since k_1 is independent of K and the corresponding terms are finite. Combining the prefix and tail bounds proves

$$\frac{1}{K}\sum_{k=0}^{K-1}\mathbb{E}[\|\nabla f(x_k)\|^2] = O\left(\frac{1+\log K}{K^{1/4}}\right) = \tilde{O}(K^{-1/4}).$$

□

Remark C.12 (On the scheduled hyperparameters). The schedule $\beta_{1,k} = 1 - 1/\sqrt{k+1}$ is used to make the momentum tracking error compatible with persistent stochastic noise. In the proof, the contraction scale $1 - \beta_{1,k} \asymp k^{-1/2}$ balances the gradient-drift term induced by the moving iterate with the stochastic noise term in Lemma C.7. The schedule $\eta_k \propto (k+2)^{-3/4}$ then balances the resulting tracking penalty with the accumulated descent.

Remark C.13 (On the convergence rate). The $\tilde{O}(K^{-1/4})$ rate is driven by the coupled momentum-tracking budget. The tracking recursion contains a contraction of order $(k+1)^{-1/2}$, while the learning rate is $\eta_k \asymp k^{-3/4}$. The square-root tracking term is handled by a time-dependent Young inequality, which produces a harmonic contribution and therefore the logarithmic factor.

C.5 Why Removing the Cap Is Hard

The proof of Theorem 6.1 analyzes the current-denominator variant, but still keeps the cap $V_{\max}$. The cap is not used to avoid the measurability issue created by the current denominator; it is used to convert self-normalized descent into Euclidean stationarity.

Indeed, even though $v_{k,i}$ depends on the current stochastic gradient $g_{k,i}$, the cap gives the pointwise bound

$$0 \le v_{k,i} \le V_{\max}, \qquad \frac{1}{\sqrt{v_{k,i}}+\varepsilon} \ge \frac{1}{\sqrt{V_{\max}}+\varepsilon}.$$

Therefore the positive part of the descent can be bounded directly:

$$\mathbb{E}\left[\frac{|\nabla_i f(x_k)|\,|g_{k,i}|}{\sqrt{v_{k,i}}+\varepsilon}\right] \ge \frac{1}{\sqrt{V_{\max}}+\varepsilon}\mathbb{E}\left[|\nabla_i f(x_k)|\,|g_{k,i}|\right].$$

Since $\nabla_i f(x_k)$ is $\mathcal{F}_{k-1}$-measurable and $\mathbb{E}_{k-1}[g_{k,i}] = \nabla_i f(x_k)$, Jensen's inequality gives

$$\mathbb{E}_{k-1}[|g_{k,i}|] \ge |\nabla_i f(x_k)|.$$

This is the step that recovers a Euclidean descent contribution proportional to

$$\mathbb{E}[\|\nabla f(x_k)\|^2].$$

Without the cap, the same argument naturally controls only the self-normalized quantity

$$\mathbb{E}\left[\sum_{i=1}^{d}\frac{\nabla_i f(x_k)^2}{\sqrt{v_{k,i}}+\varepsilon}\right],$$

rather than $\mathbb{E}[\|\nabla f(x_k)\|^2]$. Recovering Euclidean stationarity would require a mechanism that upper bounds $v_{k,i}$, either pointwise or in a sufficiently strong averaged sense. Bounded conditional variance alone does not provide such a bound, since

$$\mathbb{E}_{k-1}[g_{k,i}^2] \le \nabla_i f(x_k)^2 + \sigma^2.$$

Thus the accumulator may grow precisely when the gradient is large.

The uncapped current denominator does control the smoothness remainder. For $\beta_2 > 1/2$, the ACA update satisfies

$$v_{k,i} \ge c_{\beta_2} g_{k,i}^2, \qquad c_{\beta_2} := \min\{1-\beta_2, 2\beta_2 - 1\} > 0,$$

and hence

$$B_k = \frac{L}{2}\sum_i \frac{g_{k,i}^2}{(\sqrt{v_{k,i}}+\varepsilon)^2} \le \frac{Ld}{2c_{\beta_2}}.$$

The obstruction is therefore not stability of the smoothness term, but the opposite lower bound needed for descent: self-normalization does not imply

$$\frac{|\nabla_i f(x_k)|\,|g_{k,i}|}{\sqrt{v_{k,i}}+\varepsilon} \gtrsim \nabla_i f(x_k)^2.$$

This issue is not specific to the ACA accumulator. Consider the more favorable local EMA denominator

$$u_{k,i}^{\mathrm{loc}} := \beta_2 v_{k-1,i} + (1-\beta_2) g_{k,i}^2.$$

This is an upper envelope of the uncapped ACA update in the sense that

$$0 \le u_{k,i}^{\mathrm{loc}} - v_{k,i}^{\mathrm{cur}} \le 2(1-\beta_2) g_{k,i}^2.$$

Thus replacing $v_{k,i}^{\mathrm{cur}}$ by $u_{k,i}^{\mathrm{loc}}$ can only make the denominator larger, and is therefore conservative for the positive descent term. Nevertheless, the desired lower bound can already fail for this local EMA denominator under bounded conditional variance.

To see this, consider one coordinate, condition on $\mathcal{F}_{k-1}$, and let

$$h := \nabla_i f(x_k), \qquad b := 1-\beta_2, \qquad v_{k-1,i} = 0.$$

Then $u_{k,i}^{\mathrm{loc}} = b g_{k,i}^2$. Choose a nonnegative stochastic gradient satisfying

$$g = M \text{ with probability } p, \qquad g = 0 \text{ otherwise}, \qquad M = \frac{\sigma^2}{h}, \qquad p = \frac{h^2}{\sigma^2}.$$

Then

$$\mathbb{E}_{k-1}[g] = h, \qquad \mathbb{E}_{k-1}[g^2] = \sigma^2.$$

For small h, M is large and

$$\frac{M}{\sqrt{b}\,M+\varepsilon} \approx \frac{1}{\sqrt{b}}.$$

Hence

$$|h|\,\mathbb{E}_{k-1}\left[\frac{|g|}{\sqrt{b g^2+\varepsilon}}\right] \approx \frac{h^3}{\sigma^2\sqrt{b}}.$$

In contrast, the natural local-EMA adaptive stationarity scale is quadratic in h:

$$\frac{h^2}{\sqrt{b\,\mathbb{E}_{k-1}[g^2]}+\varepsilon} = \frac{h^2}{\sqrt{b}\,\sigma+\varepsilon}.$$

Thus bounded conditional variance allows the conditional mean to be carried by rare large stochastic gradients, while the current self-normalized denominator saturates exactly on those rare events. This prevents a uniform lower bound of the normalized descent by either a Euclidean or a standard adaptive stationarity measure.

Consequently, removing the cap while preserving the Euclidean stationarity guarantee would require a sharper mechanism that rules out this rare-spike degeneracy, such as stronger tail assumptions on the stochastic gradients, a growing/truncated cap, or another way of controlling the accumulator. We leave the fully uncapped analysis under bounded conditional variance for future work.

D Deep RL Experimental Details and Full Results

This appendix provides the full experimental details and additional results for the deep reinforcement learning experiments in Section 4. We first describe the evaluation protocol, then define the aggregate metrics used in Table 1, explain the default/tuned/best-configuration reporting protocol, and finally report full per-domain and learning-curve results.

D.1 Why Deep RL Concentrates Both Stressors

Section 2 isolates two optimizer-level stressors: persistent gradient noise and moving optimization targets. This appendix documents why deep RL is the main regime in which these two stressors appear together. The goal is not to claim that supervised learning is noise-free, but to explain why the combination studied in this paper is more pronounced in the RL settings we evaluate.

Persistent gradient noise. We measure gradient noise using the gradient noise scale (GNS) (McCandlish et al., 2018), which compares gradient variance to gradient signal. Figure 6 reports GNS throughout training across RL and supervised domains. The MuJoCo critic under SAC maintains GNS above 10^3 throughout training, whereas the GLUE aggregate remains near 10^1. The MuJoCo actor and Atari PPO lie between these regimes, but remain elevated over training. Thus, the persistent-noise mechanism of Theorem 2.3 is most relevant in the RL settings.

Moving targets. Deep RL also introduces non-stationarity in the optimization target. Bootstrapped value estimates make the target depend on changing network parameters (van Hasselt et al., 2018); replay buffers mix data from older policies with the current update distribution; and policy updates change the data distribution seen by subsequent gradients. These effects make small or poorly aligned updates harder to compensate for by simply taking more steps, because the target being tracked continues to move.

Contrast with supervised checks. The supervised benchmarks in Appendix E are used as out-of-regime sanity checks. They may still contain stochastic gradient noise, especially on small tasks, but the objective is fixed and the same combination of persistent high GNS and moving targets is less pronounced. This is consistent with the empirical pattern: Ano remains competitive on supervised checks, while the clearest gains appear in the deep RL evaluations.

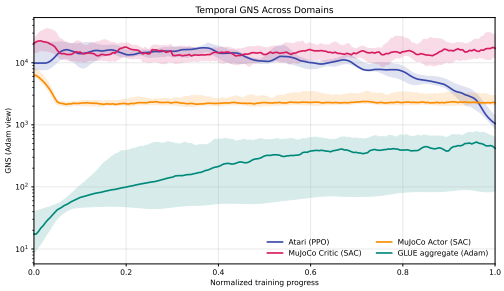


Figure 6: Gradient noise scale (GNS, Adam view) throughout training across RL and supervised domains. MuJoCo critic gradients under SAC maintain substantially higher GNS than the GLUE aggregate, while MuJoCo actor and Atari PPO gradients lie between these regimes.

D.2 Implementation and evaluation protocol

We evaluate optimizers in two deep RL settings: SAC on MuJoCo continuous-control tasks and PPO on Atari discrete-control tasks. The MuJoCo suite contains HalfCheetah, Hopper, Ant, Walker2d, and Humanoid. The Atari suite is the fixed Atari-5 subset from Aitchison et al. (2023): BattleZone, DoubleDunk, Name This Game, Phoenix, and Q*bert.

All reported main RL results use 25 independent seeds per environment and optimizer. Unless otherwise stated, aggregate statistics are computed across the 10-environment suite using stratified-bootstrap 95% confidence intervals following Agarwal et al. (2021).

MuJoCo/SAC. For continuous control, we use the CleanRL SAC implementation (Huang et al., 2022) on Gymnasium MuJoCo tasks (Todorov et al., 2012; Towers et al., 2025). Actor, critic, and entropy-temperature optimizers are replaced by the evaluated optimizer. All other SAC hyperparameters are held fixed across optimizers and are reported in Table 9.

Atari/PPO. For discrete control, we use the CleanRL PPO implementation (Huang et al., 2022) on the Atari Learning Environment (Bellemare et al., 2013). Environments are vectorized with EnvPool (Weng et al., 2022). We follow the standard Atari preprocessing protocol: observations are resized to 84×84, grayscaled, and stacked over 4 frames; we use action repeat 4, up to 30 no-op actions on reset, sticky actions with probability 0.25 (Machado et al., 2018), FireReset when required, and the full action set. Training uses clipped rewards in $[-1, 1]$, while evaluation reports unclipped returns. PPO hyperparameters are held fixed across optimizers and are reported in Table 10.

Existing assets and licenses. We use existing open-source research assets only for standard benchmarking and implementation infrastructure. The RL implementations build on CleanRL; MuJoCo experiments use Gymnasium and MuJoCo; Atari experiments use the Arcade Learning Environment and EnvPool; supervised checks use CIFAR-100, GLUE, and BERT-base. We cite the original papers or repositories for these assets in the relevant experimental sections. To the best of our knowledge, we respect their intended research use and license terms: Gymnasium is MIT licensed, MuJoCo source code is Apache-2.0 licensed, EnvPool is Apache-2.0 licensed, and BERT-base is Apache-2.0 licensed. GLUE is a benchmark composed of existing datasets, and we follow the benchmark distribution and refer to the original task licenses.

Optimizer hyperparameters are reported in Table 15.

D.3 Default, tuned, and best-configuration protocol

We report three configurations for each optimizer.

Default. The default configuration uses the optimizer hyperparameters commonly used in deep learning or recommended by the corresponding optimizer paper. This is the primary comparison reported in the main text: it tests whether the optimizer works as a drop-in replacement under a fixed RL training recipe.

Tuned. The tuned configuration is selected by the common tuning protocol described in Appendix G. The RL proxy task is HalfCheetah/SAC trained for 100k steps with 5 seeds per configuration. This proxy is used only to select optimizer hyperparameters; final tuned results are then evaluated on the full MuJoCo/SAC and Atari/PPO suites with 10 seeds, compared with 25 seeds for the default configurations.

Best Configuration. For transparency, we also report the better of the default and tuned configuration for each optimizer and RL algorithm family. These rows are descriptive only: they indicate whether the observed trends depend strongly on defaults, but they are not used for the primary comparison or for confirmatory statistical claims, because selecting the better of default and tuned configurations introduces an additional model-selection layer.

D.4 Aggregate metrics

We report aggregate metrics following Agarwal et al. (2021). Confidence intervals are computed by stratified bootstrap over seeds and environments.

Interquartile mean. For each optimizer and environment, we report the interquartile mean (IQM) of final evaluation returns across seeds. IQM averages the middle 50% of runs and is less sensitive to outlier seeds than the mean.

Probability of improvement. For each baseline B, $P(Ano > B)$ denotes the probability of improvement of Ano over B under the stratified-bootstrap distribution. Values above 0.5 favor Ano.

Normalized score. To aggregate across environments with different reward scales, we linearly normalize each environment score:

$$\tilde{s}_{e,o} = 100 \cdot \frac{s_{e,o} - s_e^{\min}}{s_e^{\max} - s_e^{\min}},$$

where $s_{e,o}$ is the IQM score of optimizer o on environment e. We use a fixed lower reference score as $s_e^{\min}$: 0 for all environments except DoubleDunk, where the score range is $[-24, 24]$ and we set $s_e^{\min} = -24$. We set $s_e^{\max}$ to the best optimizer score observed for that environment within the corresponding comparison. Thus, in pairwise comparisons, $s_e^{\max}$ is computed over Ano and the baseline B; in the full per-domain tables, it is computed over all reported optimizers.

The average normalized score is

$$\mathrm{NormAvg}(o) = \frac{1}{|\mathcal{E}|} \sum_{e \in \mathcal{E}} \tilde{s}_{e,o}.$$

Higher values are better.

D.5 Full pairwise aggregate results

Table 6 reports the full pairwise bootstrap statistics corresponding to Table 1.

Baseline	Ano norm. avg.	Baseline norm. avg.	Δ norm. avg.	$P(Ano > B)$
Adam	99.09 [96.82, 100.00]	92.83 [88.60, 96.60]	+6.27 [+1.09, +11.13]	0.661 [0.539, 0.766]
RMSprop	99.54 [98.30, 100.00]	92.49 [87.76, 96.60]	+7.05 [+2.37, +12.06]	0.664 [0.559, 0.761]
Adan	100.00 [99.92, 100.00]	77.68 [67.48, 87.29]	+22.32 [+12.68, +32.52]	0.854 [0.748, 0.939]
Grams	99.98 [99.88, 100.00]	77.00 [67.85, 85.80]	+22.98 [+14.12, +32.15]	0.858 [0.744, 0.947]
Lion	100.00 [100.00, 100.00]	63.77 [43.08, 81.80]	+36.23 [+18.20, +56.92]	0.879 [0.805, 0.944]

Table 6: Pairwise bootstrap statistics between Ano and each baseline across the 10-environment RL suite under default optimizer configurations. Normalized averages are computed from per-environment IQM final returns. Δ norm. avg. is the bootstrap difference in normalized aggregate performance, in percentage points, and $P(Ano > B)$ is the stratified-bootstrap probability of improvement over baseline B.

D.6 Per-domain and per-environment results

Tables 7 and 8 report the full per-environment results for MuJoCo/SAC and Atari/PPO. Each table reports Default, Tuned, and Best Configuration results. Entries are IQM final returns with stratified-bootstrap 95% confidence intervals; *Norm* denotes the environment-normalized score defined in Section D.4.

Interpreting per-environment intervals. The per-environment confidence intervals are reported for transparency and diagnostics, not as the primary statistical test. As emphasized by Agarwal et al. (2021), individual learning curves and per-task intervals can be noisy and their overlap should not be used as a reliable criterion for aggregate algorithm comparison. Our main statistical claims therefore use stratified-bootstrap aggregate metrics across environments (Table 1); the per-environment tables below show where the aggregate differences come from.

Optimizers	HalfCheetah IQM [95% CI]	Norm	Ant IQM [95% CI]	Norm	Humanoid IQM [95% CI]	Norm	Walker2d IQM [95% CI]	Norm	Hopper IQM [95% CI]	Norm	Avg. Norm
Baseline											
Ano (Ours)	11298.74 [10451.63, 11839.17]	100.00	5207.57 [4765.13, 5462.73]	100.00	5336.53 [5264.30, 5389.59]	99.44	5078.71 [4797.55, 5337.56]	100.00	3486.74 [3295.12, 3536.33]	100.00	99.89 [99.03, 100.00]
Adam	10591.14 [10327.38, 10815.72]	93.74	4269.73 [3769.06, 4590.96]	81.99	5272.08 [5106.05, 5375.13]	98.24	4501.98 [4349.40, 4666.47]	88.64	3235.55 [2998.13, 3416.76]	92.80	91.08 [85.21, 96.13]
RMSprop	10419.97 [9976.82, 10781.98]	92.22	4305.97 [3868.40, 4623.20]	82.69	5366.51 [5264.77, 5482.49]	100.00	4205.85 [4027.17, 4383.04]	82.81	2988.02 [2519.96, 3307.78]	85.70	88.68 [82.23, 95.42]
Adan	8441.00 [7657.69, 9164.69]	74.71	3530.72 [3166.30, 3857.53]	67.80	5182.68 [5087.58, 5286.95]	96.57	4100.22 [3948.32, 4223.78]	80.73	3198.81 [3032.39, 3272.83]	91.74	82.31 [72.85, 91.80]
Grams	7205.56 [6680.38, 7731.67]	63.77	3434.75 [3096.42, 3814.76]	65.96	5170.23 [5065.78, 5248.41]	96.34	3845.94 [3523.92, 4046.52]	75.73	1897.85 [1111.11, 2665.30]	54.43	71.25 [58.12, 85.02]
Lion	9416.07 [8908.85, 10091.65]	83.34	4528.79 [3891.53, 4810.16]	86.97	88.34 [69.64, 117.21]	1.65	4664.14 [4479.19, 4828.61]	91.84	1409.59 [967.70, 2188.46]	40.43	60.84 [27.84, 88.59]
Tuned											
Ano (Ours)	11657.91 [11198.48, 11989.90]	100.00	5498.58 [5152.23, 5653.36]	100.00	5170.59 [5021.30, 5329.30]	99.75	4783.82 [4552.57, 5012.60]	97.43	3141.80 [2530.17, 3454.75]	87.51	96.94 [89.42, 99.85]
Adan	10722.97 [10150.88, 11126.73]	91.98	5223.06 [4935.67, 5514.22]	94.99	4712.05 [3536.95, 5193.38]	90.91	4690.65 [4193.36, 5216.23]	95.53	3516.42 [3362.27, 3569.17]	97.95	94.27 [86.96, 97.46]
Grams	10346.50 [9234.60, 11038.89]	88.75	4622.18 [4211.05, 4888.54]	84.06	5155.50 [4695.07, 5325.64]	99.46	4657.98 [4172.68, 5102.90]	94.87	2977.98 [2148.64, 3317.07]	82.95	90.02 [81.15, 95.23]
Adam	8242.31 [4150.98, 9360.17]	70.70	5059.02 [4537.04, 5345.72]	92.01	5183.44 [4812.53, 5399.10]	100.00	4417.66 [3634.36, 4873.76]	89.97	2897.16 [2074.91, 3479.38]	80.70	86.68 [71.24, 94.86]
Lion	10375.24 [9119.44, 11175.12]	89.00	4841.42 [3769.15, 5278.45]	88.05	1572.68 [655.80, 3405.50]	30.34	4909.96 [4648.01, 5207.54]	100.00	3590.17 [3528.14, 3656.16]	100.00	81.48 [54.88, 97.76]
RMSprop	9954.55 [7628.52, 10682.09]	85.39	3553.76 [2693.63, 4538.33]	64.63	68.03 [58.16, 130.04]	1.31	4613.98 [3790.87, 4971.87]	93.97	2061.05 [1143.80, 2656.20]	57.41	60.54 [28.73, 84.39]
Best Configuration											
Ano (Ours) [Baseline]	11298.74 [10451.63, 11839.17]	100.00	5207.57 [4765.13, 5462.73]	99.70	5336.53 [5264.30, 5389.59]	99.44	5078.71 [4797.55, 5337.56]	100.00	3486.74 [3295.12, 3536.33]	97.12	99.25 [93.22, 99.80]
Adan [Tuned]	10722.97 [10150.88, 11126.73]	94.90	5223.06 [4935.67, 5514.22]	100.00	4712.05 [3536.95, 5193.38]	87.80	4690.65 [4193.36, 5216.23]	92.36	3516.42 [3362.27, 3569.17]	97.95	94.60 [86.11, 98.30]
Adam [Baseline]	10591.14 [10327.38, 10815.72]	93.74	4269.73 [3769.06, 4590.96]	81.75	5272.08 [5106.05, 5375.13]	98.24	4501.98 [4349.40, 4666.47]	88.64	3235.55 [2998.13, 3416.76]	90.12	90.50 [80.03, 95.50]
Grams [Tuned]	10346.50 [9234.60, 11038.89]	91.57	4622.18 [4211.05, 4888.54]	88.50	5155.50 [4695.07, 5325.64]	96.07	4657.98 [4172.68, 5102.90]	91.72	2977.98 [2148.64, 3317.07]	82.95	90.16 [81.52, 94.14]
RMSprop [Baseline]	10419.97 [9976.82, 10781.98]	92.22	4305.97 [3868.40, 4623.20]	82.44	5366.51 [5264.77, 5482.49]	100.00	4205.85 [4027.17, 4383.04]	82.81	2988.02 [2519.96, 3307.78]	83.23	88.14 [75.60, 94.49]
Lion [Tuned]	10375.24 [9119.44, 11175.12]	91.83	4841.42 [3769.15, 5278.45]	92.69	1572.68 [655.80, 3405.50]	29.31	4909.96 [4648.01, 5207.54]	96.68	3590.17 [3528.14, 3656.16]	100.00	82.10 [54.23, 97.62]

Table 7: MuJoCo/SAC results. Entries are per-environment IQM final returns with stratified-bootstrap 95% confidence intervals and normalized scores. Avg. Norm is a descriptive domain-level normalized average; the primary aggregate comparison is reported in Table 1.

D.7 Full learning curves

Figures 7 and 8 report the full learning curves for all 10 RL environments. The main text shows a representative subset in Figure 3. Shaded regions denote stratified-bootstrap 95% confidence intervals across 25 seeds.

Optimizers	BattleZone IQM [95% CI]	Norm	DoubleDunk IQM [95% CI]	Norm	Name This Game IQM [95% CI]	Norm	Phoenix IQM [95% CI]	Norm	Q*bert IQM [95% CI]	Norm	Avg. Norm
Baseline											
Ano (Ours)	8626.15 [8104.62, 9403.08]	100.00	−0.93 [−1.01, −0.83]	99.80	805.18 [734.32, 849.06]	100.00	3070.26 [2940.72, 3256.89]	90.92	4747.77 [4601.38, 5048.46]	100.00	98.14 [93.90, 100.00]
RMSprop	7904.62 [7666.15, 8183.08]	91.64	−0.96 [−1.09, −0.81]	99.65	762.32 [713.23, 790.14]	94.68	3199.57 [3077.43, 3322.52]	94.75	4523.77 [4379.69, 4654.73]	95.28	95.20 [90.91, 98.16]
Adam	8276.92 [7633.85, 9289.23]	95.95	−1.05 [−1.15, −0.93]	99.28	692.66 [647.82, 744.55]	86.03	3376.88 [3236.97, 3557.54]	100.00	4313.50 [4256.38, 4397.12]	90.85	94.42 [88.57, 99.23]
Grams	7646.15 [7301.54, 8152.31]	88.64	−0.88 [−0.99, −0.74]	100.00	634.42 [616.32, 657.06]	78.79	2207.60 [2098.52, 2308.97]	65.37	3507.69 [2948.46, 3740.77]	73.88	81.34 [70.28, 91.98]
Adan	6584.62 [6193.85, 7015.38]	76.33	−0.93 [−1.05, −0.82]	99.78	629.11 [615.62, 642.25]	78.13	2048.45 [1991.69, 2129.45]	60.66	2065.42 [1674.65, 2580.39]	43.50	71.68 [55.17, 87.69]
Lion	1380.00 [1301.54, 1481.54]	16.00	−1.88 [−2.51, −1.48]	95.66	536.12 [468.48, 603.11]	66.58	2343.77 [2179.20, 2542.22]	69.41	3737.00 [3361.65, 3979.77]	78.71	65.27 [38.59, 85.28]
Tuned											
Ano (Ours)	8695.38 [7524.62, 9706.15]	100.00	−0.94 [−1.08, −0.80]	99.65	737.29 [666.25, 809.02]	98.71	2898.37 [2711.72, 3081.29]	100.00	4865.42 [4590.12, 5494.96]	97.08	99.09 [95.84, 100.00]
Grams	7710.00 [7503.33, 8643.33]	88.67	−1.35 [−2.39, −1.03]	97.88	705.03 [634.13, 800.00]	94.39	2016.40 [1909.17, 2268.57]	69.57	5011.75 [4485.42, 5749.67]	100.00	90.10 [76.72, 95.26]
Adan	4956.67 [2543.33, 7986.67]	57.00	−0.91 [−1.05, −0.63]	99.77	746.90 [684.00, 789.93]	100.00	2653.53 [2153.47, 3019.97]	91.55	4508.08 [4274.17, 4934.75]	89.95	87.66 [66.71, 96.58]
Adam	6523.33 [5566.67, 7460.08]	75.02	−0.97 [−1.19, −0.76]	99.54	534.27 [463.00, 569.17]	71.53	409.03 [382.13, 527.33]	14.11	4483.67 [4235.58, 5498.17]	89.46	69.93 [38.88, 88.63]
Lion	1290.00 [1153.33, 1543.33]	14.84	−2.42 [−3.67, −1.21]	93.27	554.60 [463.10, 613.00]	74.25	2218.57 [1883.30, 2624.93]	76.55	3722.92 [2750.08, 4049.42]	74.28	66.64 [37.56, 83.88]
RMSprop	0.00 [0.00, 333.33]	0.00	−0.86 [−1.03, −0.79]	100.00	63.33 [0.00, 198.33]	8.48	15.93 [14.07, 17.27]	0.55	70.83 [0.00, 145.83]	1.41	22.09 [0.38, 60.66]
Best Configuration											
Ano (Ours) [Baseline]	8626.15 [8104.62, 9403.08]	100.00	−0.93 [−1.01, −0.83]	99.94	805.18 [734.32, 849.06]	100.00	3070.26 [2940.72, 3256.89]	90.92	4747.77 [4601.38, 5048.46]	94.73	97.12 [90.70, 99.87]
RMSprop [Baseline]	7904.62 [7666.15, 8183.08]	91.64	−0.96 [−1.09, −0.81]	99.80	762.32 [713.23, 790.14]	94.68	3199.57 [3077.43, 3322.52]	94.75	4523.77 [4379.69, 4654.73]	90.26	94.22 [86.38, 97.21]
Adam [Baseline]	8276.92 [7633.85, 9289.23]	95.95	−1.05 [−1.15, −0.93]	99.42	692.66 [647.82, 744.55]	86.03	3376.88 [3236.97, 3557.54]	100.00	4313.50 [4256.38, 4397.12]	86.07	93.49 [83.93, 98.55]
Grams [Tuned]	7710.00 [7503.33, 8643.33]	89.38	−1.35 [−2.39, −1.03]	98.11	705.03 [634.13, 800.00]	87.56	2016.40 [1909.17, 2268.57]	59.71	5011.75 [4485.42, 5749.67]	100.00	86.95 [71.98, 97.58]
Adan [Tuned]	4956.67 [2543.33, 7986.67]	57.46	−0.91 [−1.05, −0.63]	100.00	746.90 [684.00, 789.93]	92.76	2653.53 [2153.47, 3019.97]	78.58	4508.08 [4274.17, 4934.75]	89.95	83.75 [67.16, 94.83]
Lion [Baseline]	1380.00 [1301.54, 1481.54]	16.00	−1.88 [−2.51, −1.48]	95.80	536.12 [468.48, 603.11]	66.58	2343.77 [2179.20, 2542.22]	69.41	3737.00 [3361.65, 3979.77]	74.56	64.47 [38.04, 84.94]

Table 8: Atari/PPO results. Entries are per-environment IQM final returns with stratified-bootstrap 95% confidence intervals and normalized scores. Avg. Norm is a descriptive domain-level normalized average; the primary aggregate comparison is reported in Table 1.

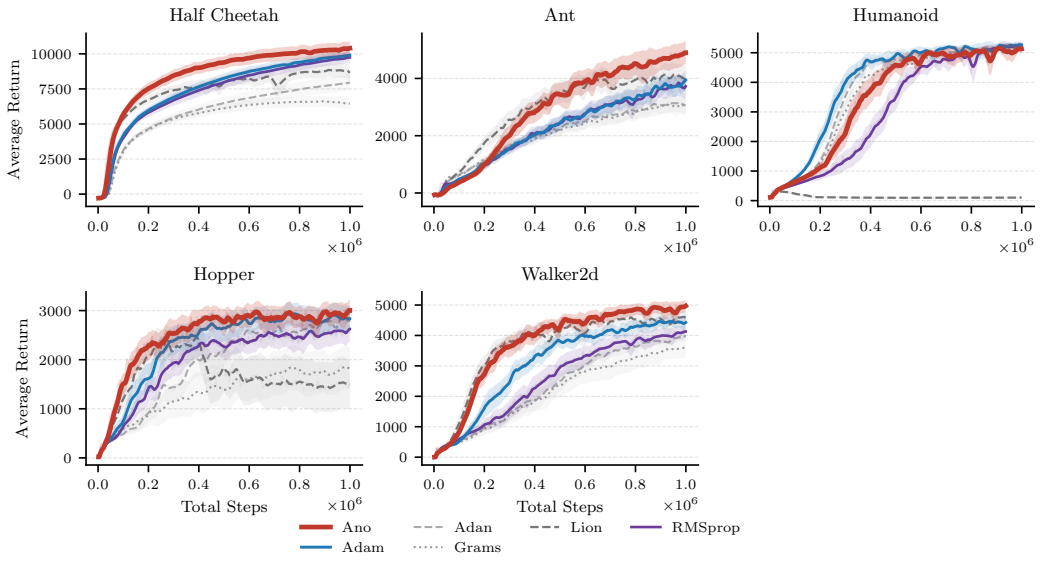


Figure 7: Full MuJoCo/SAC learning curves. Shaded regions show 95% stratified-bootstrap confidence intervals across 25 seeds.

D.8 RL Algorithms Hyperparameters

E Supervised Sanity Checks

The supervised experiments are included as out-of-regime sanity checks. Ano is designed for noisy tracking regimes such as deep RL, where both gradient statistics and optimization targets change during training. Supervised learning is comparatively more stationary: the objective is fixed, and standard learning-rate schedules already provide late-stage damping. We therefore use GLUE and CIFAR-100 only to check that Ano remains well-behaved outside its target regime, not to argue that it is a new default optimizer for supervised learning.

NLP: GLUE. We fine-tune BERT-base (Devlin et al., 2019) on the GLUE benchmark, excluding WNLI (Wang et al., 2019). We use a standard fine-tuning recipe with sequence length 128, batch size 32, weight decay 0.01, mixed precision, and a linear learning-rate schedule with 10% warmup. We train for 3 epochs on the larger tasks and 5 epochs on CoLA, MRPC, RTE, and STS-B. Table 11 reports mean performance with 95% confidence intervals over 5 seeds for both default and tuned optimizer hyperparameters.

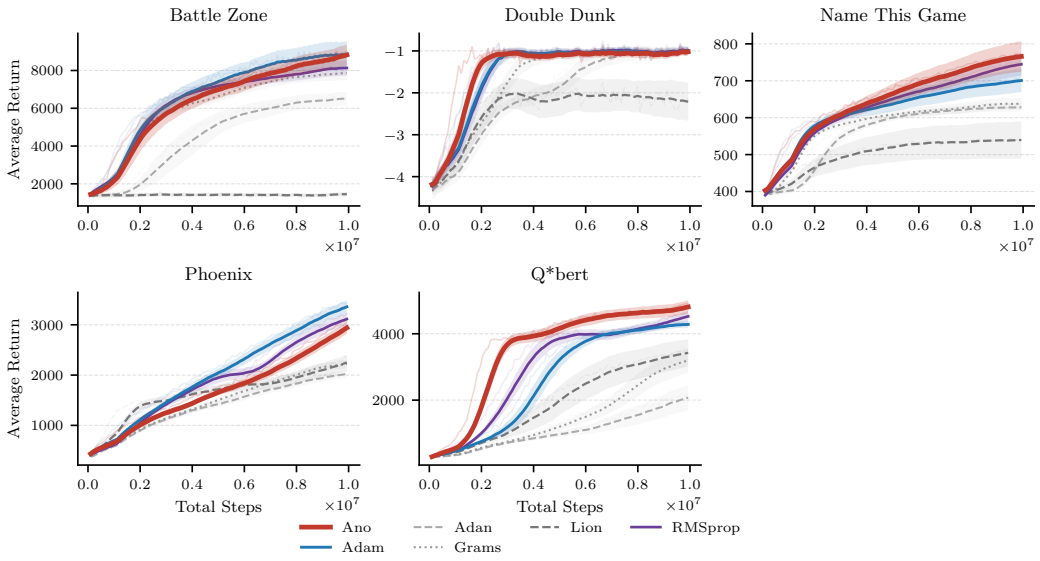


Figure 8: Full Atari/PPO learning curves. Shaded regions show 95% stratified-bootstrap confidence intervals across 25 seeds.

Hyperparameter	Value
Total training steps	1,000,000
Discount γ	0.99
Soft update rate τ	0.005
Replay buffer size	10^6
Batch size	256
Learning starts	5,000 steps
Actor LR / Critic LR	see Table 15
Policy update frequency	2
Target network update freq.	1
Entropy coeff. α (init)	0.2
Entropy autotune	✓
Max grad. norm (actor)	0.5
Logging interval	2048 steps

Table 9: SAC hyperparameters used in our MuJoCo experiments (values taken from the code).

Across GLUE, Ano remains close to Adam and other adaptive baselines under both default and tuned hyperparameters. Its average score is slightly higher than Adam in the default setting (82.92 vs 82.64) and comparable in the tuned setting (82.89 vs 82.50). The differences are small relative to the paper's RL effects, which is the expected pattern: the mechanisms targeted by Ano are not meant to dominate in stationary supervised fine-tuning.

Computer vision: CIFAR-100. We train a ResNet-34 (He et al., 2016) on CIFAR-100 (Alex, 2009) with standard data augmentation: random crop with padding, horizontal flip, and Cutout (Zagoruyko & Komodakis, 2016; Wightman et al., 2021).We use the same model, training schedule, and augmentation pipeline across optimizers, changing only the optimizer and its hyperparameters. Table 12 reports test accuracy and final training loss for default and tuned configurations.

On CIFAR-100, Ano is again competitive with the adaptive baselines. It achieves the highest accuracy in both the default and tuned settings in this experiment, but the result should be interpreted as a sanity check: the benchmark is stationary, uses a decaying learning-rate schedule, and does not isolate the noisy tracking mechanisms studied in the main paper.

Takeaway. The supervised results support a narrow conclusion: Ano does not show pathological behavior on the stationary supervised checks considered here. They should not be read as evidence

Hyperparameter	Value
Total timesteps	10,000,000
Number of envs (N_{env})	64
Steps per rollout (N_{steps})	64
Batch size ($N_{\text{env}} \times N_{\text{steps}}$)	4096
Minibatches	4
Minibatch size	1024
Update epochs	4
Discount γ	0.99
GAE λ	0.95
Learning rate	see Table 15
LR annealing	linear (enabled)
Advantage normalization	✓
Policy clip coef.	0.10
Value clip	✓
Entropy coef.	0.01
Value loss coef.	0.5
Max grad-norm	0.5
Target KL	none

Table 10: PPO hyperparameters used in our Atari experiments (defaults from code).

Optimizer	CoLA	MNLI	MRPC	QNLI	QQP	RTE	SST-2	STS-B	Average
Default									
Adam	$\mathbf{59.40 \pm 1.67}$	$\mathbf{84.62 \pm 0.10}$	88.06 ± 0.82	$\mathbf{91.60 \pm 0.15}$	89.64 ± 0.10	66.67 ± 1.59	92.73 ± 0.46	88.44 ± 0.27	82.64
Adan	55.65 ± 0.53	84.17 ± 0.07	84.40 ± 0.83	91.10 ± 0.14	88.85 ± 0.04	61.49 ± 1.30	92.02 ± 0.20	87.26 ± 0.63	80.62
Lion	57.76 ± 1.76	83.76 ± 0.23	87.13 ± 0.81	90.63 ± 0.68	89.46 ± 0.05	62.89 ± 1.17	91.67 ± 0.52	88.00 ± 0.25	81.41
Grams	56.15 ± 0.92	83.89 ± 0.11	84.92 ± 0.60	91.10 ± 0.04	88.48 ± 0.08	63.36 ± 1.60	92.34 ± 0.19	87.57 ± 0.32	80.98
Ano	58.36 ± 1.15	84.33 ± 0.17	$\mathbf{88.96 \pm 0.50}$	91.25 ± 0.46	$\mathbf{89.71 \pm 0.11}$	$\mathbf{69.25 \pm 2.94}$	$\mathbf{92.80 \pm 0.41}$	88.70 ± 0.12	**82.92**
Tuned									
Adam	57.66 ± 2.39	84.18 ± 0.16	88.09 ± 0.79	91.12 ± 0.17	89.55 ± 0.08	68.47 ± 2.91	92.18 ± 0.08	88.76 ± 0.36	82.50
Adan	57.71 ± 0.92	$\mathbf{84.84 \pm 0.10}$	88.14 ± 0.40	91.71 ± 0.21	$\mathbf{89.78 \pm 0.07}$	65.40 ± 1.66	92.68 ± 0.26	88.57 ± 0.44	82.35
Lion	56.30 ± 0.55	82.38 ± 0.06	86.83 ± 2.91	90.36 ± 0.42	88.60 ± 0.13	63.75 ± 5.50	91.47 ± 0.24	88.58 ± 0.40	81.03
Grams	58.18 ± 1.12	84.64 ± 0.11	$\mathbf{89.05 \pm 0.36}$	$\mathbf{91.79 \pm 0.17}$	89.66 ± 0.06	67.22 ± 2.55	$\mathbf{92.98 \pm 0.31}$	88.53 ± 0.26	82.76
Ano	$\mathbf{58.51 \pm 0.75}$	84.39 ± 0.12	88.53 ± 1.14	91.30 ± 0.48	89.73 ± 0.07	$\mathbf{69.25 \pm 3.01}$	92.66 ± 0.14	88.74 ± 0.11	**82.89**

Table 11: GLUE sanity check. Entries are mean task scores with 95% confidence intervals over 5 seeds. WNLI is excluded. The average is the unweighted macro-average over the reported tasks. These results are intended to verify stable supervised behavior, not to establish state-of-the-art supervised optimization performance.

that Ano universally improves supervised training. The intended regime remains deep RL and other settings where persistent gradient noise and drifting objectives interact.

F Additional Diagnostic Experiments

F.1 Capped Ano vs. Ano

To assess whether the cap used in the analysis materially changes the optimizer in practice, we compare Ano, Capped Ano($V_{\max} = 10^6$), and Capped Ano($V_{\max} = 10^4$) on HalfCheetah-v5 over 10 seeds. We report learning curves and cap-activity diagnostics for both cap values.

Figure 9 shows that all three variants exhibit nearly identical learning dynamics throughout training, with overlapping stratified-bootstrap 95% confidence intervals and no visible performance separation — including Capped Ano($V_{\max} = 10^4$), which, as shown below, activates far more frequently than its looser counterpart.

Figures 10a and 10b report cap-activity diagnostics for both cap values. Capped Ano($V_{\max} = 10^6$) is nearly always inactive: at no point do more than 3 seeds activate it simultaneously, with a peak update rate below 2.64%. Capped Ano($V_{\max} = 10^4$) is substantially more active: up to 9–10 seeds activate the cap, with a maximum update rate reaching approximately 50%. Despite this difference in cap activity, the learning curves remain indistinguishable across all three variants (Figure 9), indicating that the cap has negligible effect in this HalfCheetah diagnostic even when frequently triggered.

These results suggest that, in this diagnostic, Capped Ano is a reasonable empirical proxy for the practical optimizer.

Optimizer	Test Accuracy	Training Loss
Default		
Adam	69.57 ± 0.22	0.037
Adan	69.87 ± 0.09	0.049
Lion	66.25 ± 0.61	0.064
Grams	68.33 ± 0.40	0.045
Ano	$\mathbf{70.31 \pm 0.50}$	**0.015**
Tuned		
Adam	69.61 ± 0.23	0.042
Adan	69.09 ± 0.16	0.049
Lion	68.77 ± 0.26	0.048
Grams	68.11 ± 0.27	0.048
Ano	$\mathbf{69.89 \pm 0.42}$	**0.022**

Table 12: CIFAR-100 sanity check with ResNet-34. Entries are test accuracy with 95% confidence intervals and final training loss. The goal is to verify that Ano remains well-behaved on a stationary supervised vision benchmark, not to claim a new supervised-learning optimizer.

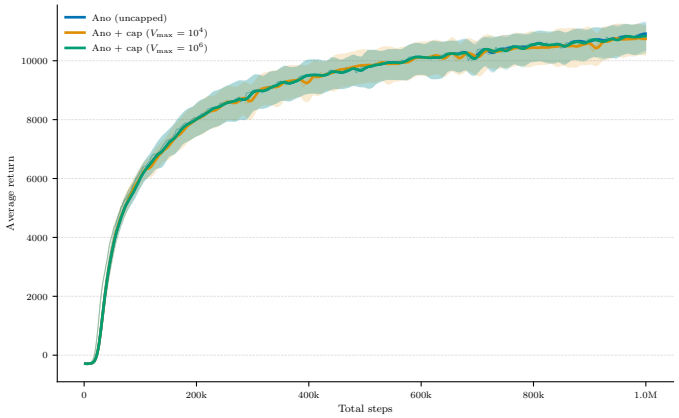


Figure 9: Learning curves of Ano, Capped Ano($V_{\max} = 10^4$), and Capped Ano($V_{\max} = 10^6$) on HalfCheetah-v5 over 10 seeds. Shaded regions denote stratified-bootstrap 95% confidence intervals. All three variants exhibit nearly indistinguishable performance throughout training.

F.2 Rosenbrock Analysis

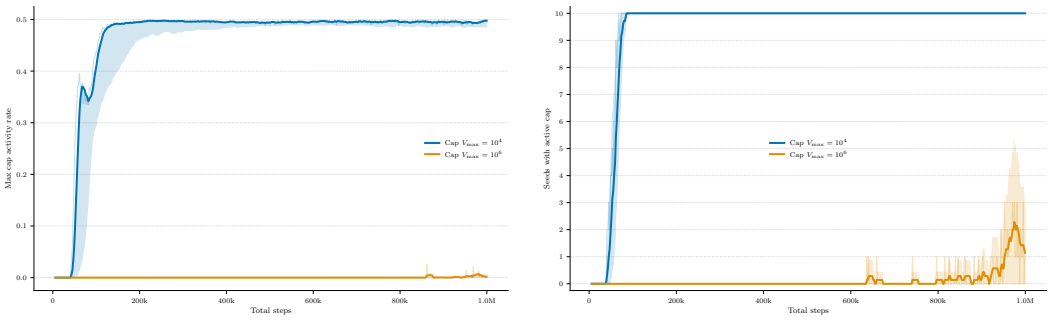
To further characterize the optimization behavior of Ano in a controlled low-dimensional setting, we compare it against Adam on the two-dimensional Rosenbrock function,

$$f(x, y) = (1 - x)^2 + 100(y - x^2)^2,$$

whose global minimizer is $(1, 1)$. While this objective is stationary and therefore does not reflect the full non-stationary nature of RL optimization, it provides a useful diagnostic for studying optimizer trajectories in a narrow curved valley and for assessing potential oscillatory behavior near convergence.

We run both optimizers from multiple initializations using the same iteration budget and compare their trajectories in parameter space, their distance to the optimum, and their final optimization error. Figure 11 shows that Ano generally follows the Rosenbrock valley effectively and reaches the vicinity of the optimum from all tested initializations. Across most initializations, Ano also achieves a lower final loss and a smaller final distance to $(1, 1)$ than Adam, as summarized in Table 13. In particular, Ano obtains better final loss and final distance on $5/6$ initializations.

An important observation is that for the initialization $(0, 0)$, Figure 12 shows that Ano exhibits visible late-stage oscillations in the neighborhood of the minimizer, whereas Adam converges more smoothly in the final phase. Thus, the oscillatory behavior can indeed occur around the minimum. However, this does not prevent Ano from reaching the basin of attraction of the optimum; rather, it mainly affects the final asymptotic refinement under a fixed learning rate.



(a) Cap-activity diagnostics for Capped Ano on HalfCheetah-v5. Despite markedly different activation levels across cap values, learning curves remain indistinguishable (Figure 9).

(b) Maximum cap-active update rate across seeds at each training step, for $V_{\max} \in \{10^4, 10^6\}$.

Figure 10: Number of seeds (out of 10) for which the cap is active at each training step, for $V_{\max} \in \{10^4, 10^6\}$.

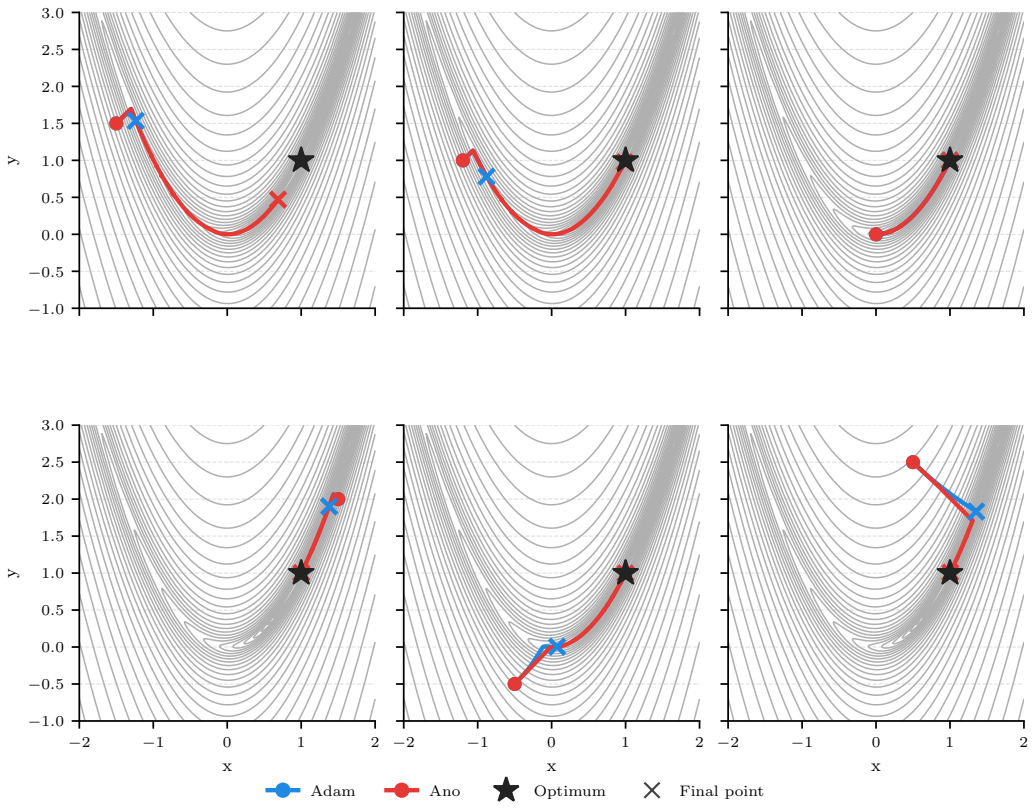


Figure 11: Optimization trajectories of Adam and Ano on the 2D Rosenbrock function across six initializations. Each panel shows the contour lines of the objective, the optimizer trajectory in parameter space, and the final point after the same number of iterations. Ano generally follows the curved valley effectively and reaches a lower final loss and a smaller final distance to the optimum on most initializations.

This behavior is consistent with the design of Ano. Ano is intended to react more rapidly to changing gradient statistics, which is particularly beneficial in non-stationary settings such as RL, where the objective, targets, and data distribution evolve throughout training. This increased responsiveness

Initialization	Adam final loss	Ano final loss	Adam final dist	Ano final dist	Comment
$(-1.5,\ 1.5)$	5.005e+00	**9.802e−02**	2.299e+00	**6.141e−01**	Ano better
$(-1.2,\ 1.0)$	3.537e+00	**1.498e−04**	1.893e+00	**2.725e−02**	Ano better
$(0.0,\ 0.0)$	**8.575e−08**	1.090e−03	**6.548e−04**	3.844e−03	Adam smoother near optimum
$(1.5,\ 2.0)$	1.435e−01	**1.555e−04**	9.779e−01	**1.973e−03**	Ano better
$(-0.5,\ -0.5)$	8.578e−01	**9.508e−04**	1.360e+00	**6.284e−03**	Ano better
$(0.5,\ 2.5)$	1.249e−01	**3.211e−03**	9.042e−01	**7.543e−03**	Ano better
Median	5.006e−01	**1.020e−03**	1.169e+00	**6.914e−03**	Ano better on $5/6$ cases

Table 13: Final loss and distance to the optimum on the Rosenbrock function across six initializations. Ano is better on five of the six starts; Adam is smoother only for the $(0, 0)$ initialization.

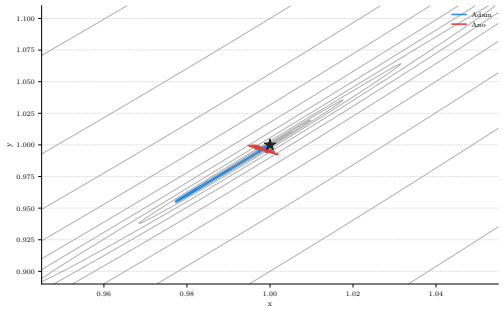


Figure 12: Late-stage zoom near the minimizer for initialization $(0, 0)$. Ano reaches the neighborhood of the optimum but exhibits residual oscillations around the minimizer, whereas Adam follows a smoother final trajectory. This indicates that the oscillatory behavior can occur near convergence, although it does not prevent Ano from entering the basin of attraction of the optimum.

can improve tracking ability, but on a stationary objective such as Rosenbrock it may provide less damping close to convergence and therefore lead to residual oscillations near the minimizer.

Importantly, this effect can be mitigated with a decaying learning rate. Figure 13 shows that adding a cosine decay preserves the rapid early progress of Ano while significantly reducing the residual oscillations in the final stage, leading to a smaller final distance to the optimum. This is also consistent with our supervised experiments, where decaying learning-rate schedules are already used. Overall, the Rosenbrock experiment illustrates a trade-off between responsiveness and asymptotic smoothness: Ano can make faster progress and better track the valley under a fixed budget, while a decreasing step size improves late-stage stability on stationary objectives.

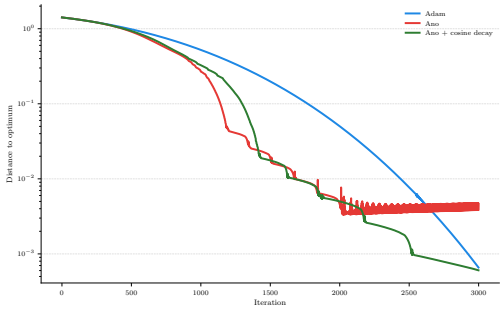


Figure 13: Distance to the optimum over the full training trajectory for initialization $(0, 0)$. Ano makes rapid early progress toward the optimum, but under a constant learning rate it exhibits residual late-stage oscillations. Adding a cosine learning-rate decay preserves the fast initial approach while improving final stability and reducing the distance to the optimum.

F.3 Switching Quadratic Analysis

We assess the behavior of four optimizers (Lion, Grams, Adam, and Ano) in a controlled, non-stationary synthetic setting by evaluating their tracking performance under drifting objectives. This is a controlled mechanism diagnostic rather than a tuned performance comparison; identical hyperparameters are used to isolate tracking behavior under target switches. We consider a switching quadratic sequence $f_t(x) = \frac{1}{2}\|x - \mu_t\|^2$, where the target μ_t is abruptly reassigned every 1000 iterations. This generates a piecewise-stationary landscape that isolates the effect of non-stationarity from model- or data-dependent factors. All optimizers are run with identical hyperparameters (learning rate, momentum coefficients, etc.), and results are averaged over three random seeds. We report the resulting tracking error $\|x_t - \mu_t\|$ across time.

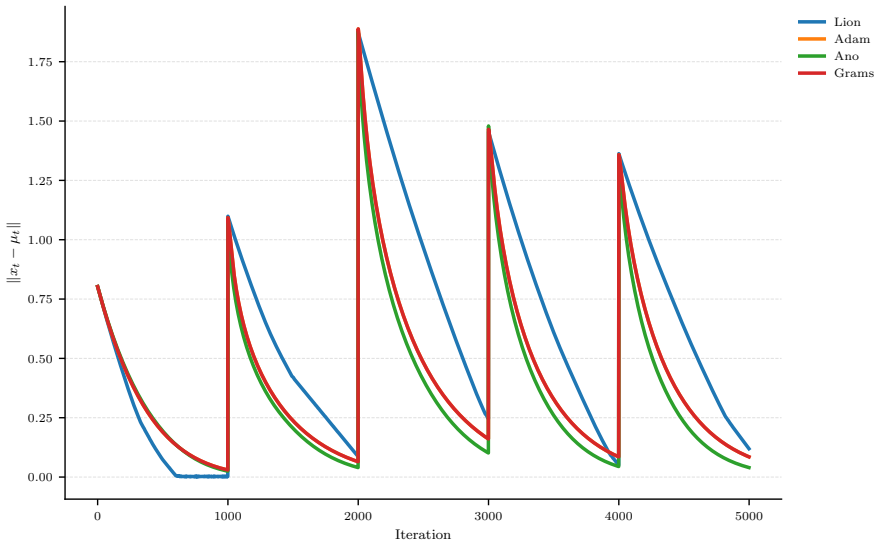


Figure 14: Tracking error $\|x_t - \mu_t\|$ in the switching quadratic setting (mean over 3 seeds). Vertical jumps correspond to resets of the target μ_t.

Figure 14 shows that all methods exhibit the characteristic sawtooth pattern induced by the periodic switches. Grams and Adam produce nearly identical trajectories, reflecting similar sensitivity to abrupt shifts despite their different update formulations. Lion consistently recovers more slowly after each jump, leading to higher transient and steady-state errors. Ano adapts the fastest overall, rapidly reducing the post-switch deviation and achieving the lowest tracking error throughout the experiment.

G Hyperparameter Selection Protocol

We use lightweight proxy tasks to select tuned optimizer configurations before running the final experiments. The purpose of this protocol is to avoid comparing Ano against poorly tuned baselines while keeping the tuning budget comparable across optimizers. Proxy-task results are used only for hyperparameter selection; all reported benchmark results are computed on the final tasks described in Sections 4 and E.

The main RL aggregate comparison in Table 1 uses default optimizer configurations. In Appendix D, we additionally report tuned configurations and best-of-default-or-tuned configurations for each optimizer. Thus, the main result tests the default setting, while the appendix checks that the conclusion is not an artifact of unfavorable baseline defaults.

Across optimizers, we search learning rates and momentum parameters using 5 independent seeds per configuration. The full tuning campaign contains **2115** independent training runs. Optimizers with three tunable parameters use approximately 35 GPU-hours each; optimizers with two tunable parameters, such as RMSprop, use approximately 12 GPU-hours. For Adan, which has three

momentum coefficients, we vary (β_1, β_3) and fix the Nesterov coefficient to its default value $\beta_2 = 0.92$, preserving a comparable search budget.

Optimizers are tuned over the parameters they expose. Adam, Grams, and Ano use the default momentum grid in Table 14. Lion uses the same β_1 values but a Lion-specific β_2 grid. For Adan, we follow the search ranges recommended for the optimizer: we vary (β_1, β_3) and keep the Nesterov coefficient fixed at $\beta_2 = 0.92$. RMSprop uses only the learning-rate and second-moment decay grids.

For each optimizer and domain, we select one tuned configuration by averaging proxy-task validation performance across the 5 tuning seeds. The resulting default, tuned, and best-of-default-or-tuned hyperparameters are reported in Table 15.

Domain proxy	Learning rates	Default momentum grid	Seeds/config.
CIFAR-10 / ResNet-18	$\{3\times10^{-5}, 3\times10^{-4}, 3\times10^{-3}\}$	$\beta_1 \in \{0.9, 0.92, 0.95\}$; $\beta_2 \in \{0.95, 0.99, 0.999\}$	5
MRPC / BERT-base	$\{7\times10^{-6}, 2\times10^{-5}, 7\times10^{-5}\}$	$\beta_1 \in \{0.9, 0.92, 0.95\}$; $\beta_2 \in \{0.95, 0.99, 0.999\}$	5
HalfCheetah / SAC, 100k	$\{3\times10^{-5}, 3\times10^{-4}, 3\times10^{-3}\}$	$\beta_1 \in \{0.9, 0.92, 0.95\}$; $\beta_2 \in \{0.95, 0.99, 0.999\}$	5

Table 14: Proxy-task hyperparameter grids. The learning-rate grid is shared within each domain. The default momentum grid applies to Adam, RMSprop (where applicable), Grams, and Ano. Lion uses $\beta_1 \in \{0.9, 0.92, 0.95\}$ and $\beta_2 \in \{0.9, 0.99, 0.999\}$. Adan uses the learning-rate grid and its recommended search ranges, $\beta_1 \in \{0.95, 0.97, 0.98\}$ and $\beta_3 \in \{0.9, 0.94, 0.96\}$, while fixing $\beta_2 = 0.92$. RMSprop uses only the learning-rate and β_2 grids.

Computer vision proxy. For the vision proxy, we use CIFAR-10 (Alex, 2009) with a ResNet-18 backbone (He et al., 2016). We draw a balanced subset of 10,000 training images and 2,000 validation images, and use the same augmentation family as in the final CIFAR experiments: random crop, horizontal flip, and Cutout (Wightman et al., 2021). Each configuration is trained for 20 epochs under 5 seeds and selected by validation accuracy.

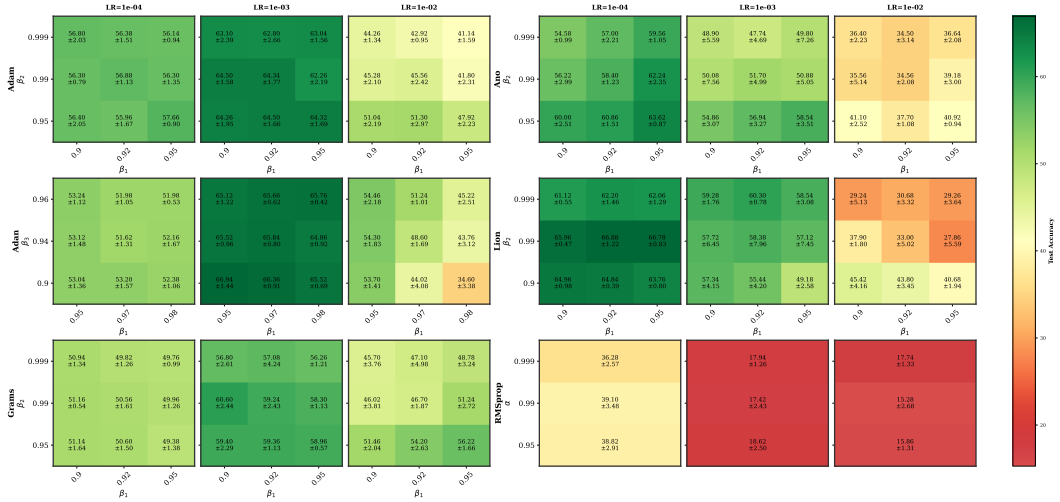


Figure 15: Grid search on the CIFAR-10 proxy with ResNet-18. The proxy is used only for hyperparameter selection; final CIFAR-100 results are reported in Table 12.

NLP proxy. For the NLP proxy, we fine-tune BERT-base (Devlin et al., 2019) on MRPC from GLUE (Wang et al., 2019). MRPC is small enough to make the tuning campaign computationally feasible while retaining the optimizer sensitivity typical of low-resource fine-tuning. A preliminary sweep showed that learning rates outside $[10^{-5}, 7\times10^{-5}]$ performed poorly, so the final grid focuses on this range. Configurations are selected by validation performance averaged over 5 seeds.

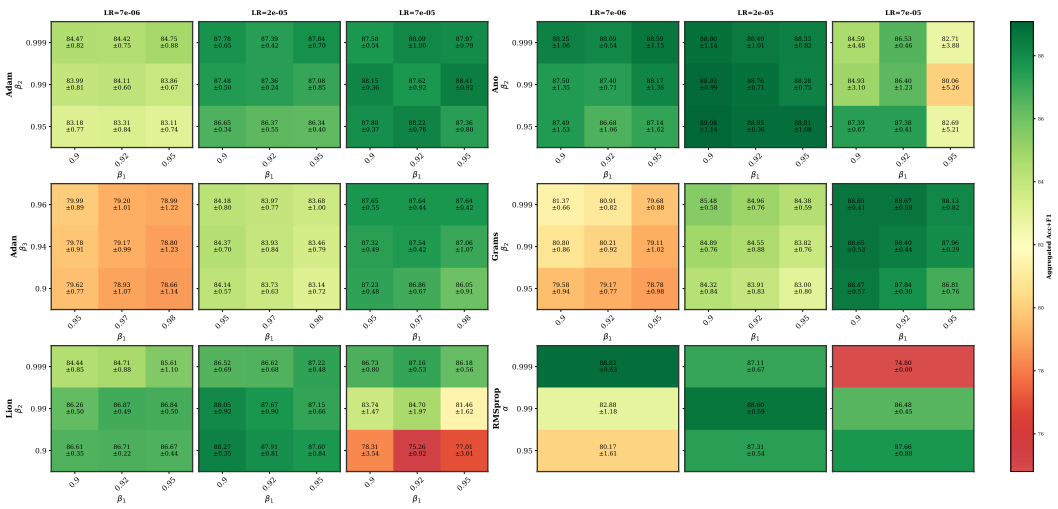


Figure 16: Grid search on the MRPC proxy with BERT-base. The proxy is used only for hyperparameter selection; final GLUE results are reported in Table 11.

Deep RL proxy. For the RL proxy, we train SAC (Haarnoja et al., 2018) on HalfCheetah-v5 from Gymnasium MuJoCo (Todorov et al., 2012; Towers et al., 2025) for 100k environment steps. This short-horizon proxy is used only for hyperparameter selection and may favor more aggressive learning rates than the final 1M-step runs. For this reason, Appendix D reports default, tuned, and best-of-default-or-tuned results separately.

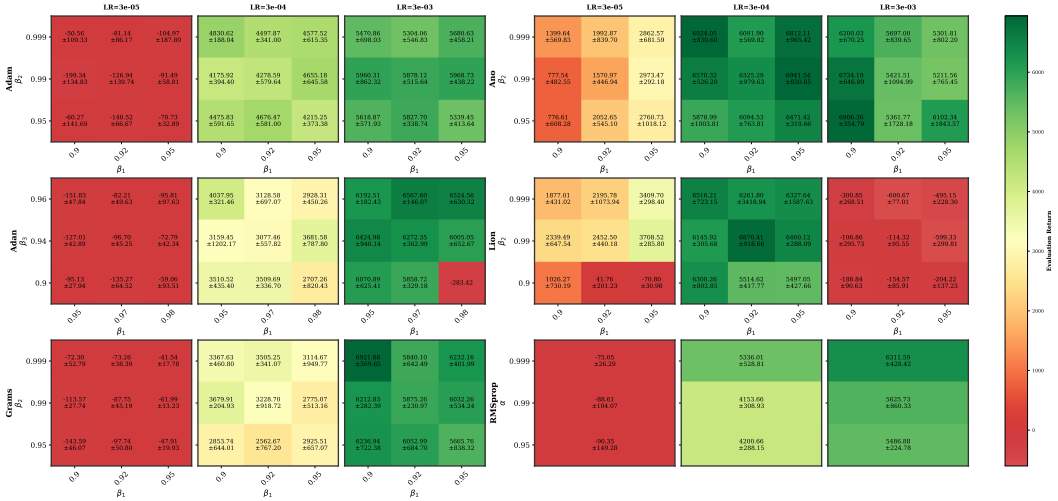


Figure 17: Grid search on the HalfCheetah/SAC 100k-step proxy. The proxy is used only for hyperparameter selection; final RL results are reported in Tables 7 and 8.

H Optimizer Hyperparameter Settings

Table 15: Optimizer hyperparameters used across diagnostics, supervised experiments, and deep RL experiments. "Default" denotes the standard configuration used before tuning. "Tuned" denotes the configuration selected by the proxy-task tuning protocol in Appendix G. "Best" denotes the configuration used for best-of-default-and-tuned appendix results. A dash indicates that the optimizer does not use the corresponding hyperparameter.

Setting	**Model**	**Optimizer**	β_1	β_2	β_3	lr	λ
Computer vision: CIFAR-100 / ResNet-34							
Default	ResNet-34	AdamW	0.9	0.999	–	1e-3	1e-2
Default	ResNet-34	Adan	0.98	0.92	0.99	1e-3	1e-2
Default	ResNet-34	Ano	0.92	0.99	–	1e-3	1e-2
Default	ResNet-34	Lion	0.9	0.99	–	1e-3	1e-2
Default	ResNet-34	Grams	0.9	0.999	–	1e-3	1e-2
Tuned	ResNet-34	AdamW	0.9	0.99	–	1e-3	1e-2
Tuned	ResNet-34	Adan	0.95	0.92	0.9	1e-3	1e-2
Tuned	ResNet-34	Ano	0.95	0.95	–	1e-4	1e-2
Tuned	ResNet-34	Lion	0.92	0.99	–	1e-4	1e-2
Tuned	ResNet-34	Grams	0.9	0.99	–	1e-3	1e-2
Natural language processing: GLUE / BERT-base							
Default	BERT-base	AdamW	0.9	0.999	–	2e-5	1e-2
Default	BERT-base	Adan	0.98	0.92	0.99	2e-5	1e-2
Default	BERT-base	Ano	0.92	0.99	–	2e-5	1e-2
Default	BERT-base	Lion	0.9	0.99	–	2e-5	1e-2
Default	BERT-base	Grams	0.9	0.999	–	2e-5	1e-2
Tuned	BERT-base	AdamW	0.95	0.99	–	7e-5	1e-2
Tuned	BERT-base	Adan	0.95	0.92	0.96	7e-5	1e-2
Tuned	BERT-base	Ano	0.9	0.95	–	2e-5	1e-2
Tuned	BERT-base	Lion	0.9	0.9	–	7e-6	1e-2
Tuned	BERT-base	Grams	0.92	0.999	–	7e-5	1e-2
Deep reinforcement learning: MuJoCo/SAC and Atari/PPO							
Default	SAC/PPO	Adam	0.9	0.999	–	3e-4	–
Default	SAC/PPO	Adan	0.98	0.92	0.99	3e-4	–
Default	SAC/PPO	Ano	0.92	0.99	–	3e-4	–
Default	SAC/PPO	Lion	0.9	0.99	–	3e-4	–
Default	SAC/PPO	Grams	0.9	0.999	–	3e-4	–
Default	SAC/PPO	RMSprop	–	0.99	–	3e-4	–
Tuned	SAC/PPO	Adam	0.9	0.99	–	3e-3	–
Tuned	SAC/PPO	Adan	0.97	0.92	0.96	3e-3	–
Tuned	SAC/PPO	Ano	0.95	0.99	–	3e-4	–
Tuned	SAC/PPO	Lion	0.92	0.99	–	3e-4	–
Tuned	SAC/PPO	Grams	0.9	0.999	–	3e-3	–
Tuned	SAC/PPO	RMSprop	–	0.999	–	3e-3	–
Best configuration used in best-of-default-and-tuned RL appendix results							
Best SAC	SAC	Adam	0.9	0.999	–	3e-4	–
Best SAC	SAC	Adan	0.97	0.92	0.96	3e-3	–
Best SAC	SAC	Ano	0.92	0.99	–	3e-4	–
Best SAC	SAC	Lion	0.92	0.99	–	3e-4	–
Best SAC	SAC	Grams	0.9	0.999	–	3e-3	–
Best SAC	SAC	RMSprop	–	0.99	–	3e-4	–
Best PPO	PPO	Adam	0.9	0.999	–	3e-4	–
Best PPO	PPO	Adan	0.97	0.92	0.96	3e-3	–
Best PPO	PPO	Ano	0.95	0.99	–	3e-4	–
Best PPO	PPO	Lion	0.92	0.99	–	3e-4	–
Best PPO	PPO	Grams	0.9	0.999	–	3e-3	–
Best PPO	PPO	RMSprop	–	0.99	–	3e-4	–