
Variance-Reduced Streaming PCA with Parameter-Free Momentum

Shrestha Bhattacharya
Georgia Institute of Technology

Abstract

Streaming PCA via Oja’s algorithm uses rank-one gradients whose noise scales with ambient dimension and rank. We propose VR-OJA, which maintains a bias-corrected exponential moving average (EMA) of Oja gradients and retracts after stepping along the average. We give an exact Gaussian calculation of the Oja-gradient noise, derive the exact EMA variance-reduction factor $(1 - \beta)/(1 + \beta)$, and solve a scalar linearization of EMA-Oja via a 2×2 Lyapunov equation, showing that stationary variance decreases monotonically with β and that EMA widens the algorithm’s stability region. These results establish $\beta^* = 1 - k/p$ as the parameter-free momentum default, with EMA time constant p/k matching the natural noise-suppression scale of p coordinates per tracked direction. We implement VR-OJA with QR and Newton–Schulz retractions and report both because approximate orthogonalization can affect accuracy. On synthetic streams ($p = 500$, $k = 10$, $T = 3000$, three seeds), VR-OJA reduces final subspace error by 22–27% over OJA-QR across stationary, drifting, and abruptly shifting covariance scenarios. On the S&P 500 benchmark [1], VR-OJA outperforms OJA-QR and the noisy power method for $k \leq 2$, and provides substantial learning-rate robustness at all ranks (24–47% error reduction at $10\times$ the base learning rate).

1 Introduction

Principal component analysis is a foundational tool in statistics, signal processing, and machine learning, but the classical eigendecomposition-based approach is impractical when data arrives sequentially and cannot be stored. The streaming setting—where samples $x_t \in \mathbb{R}^p$ arrive one at a time and the algorithm maintains a $p \times k$ estimate $\hat{U}_t$ of the top- k eigenvectors of the data covariance—has been studied since the seminal work of Oja [2]. In its modern form, Oja’s algorithm performs an online stochastic gradient step on the Rayleigh quotient followed by a retraction onto the Stiefel manifold of orthonormal frames:

$$\hat{U}_t = \Pi(\hat{U}_{t-1} + \zeta \cdot x_t x_t^\top \hat{U}_{t-1}), \quad (1)$$

where $\zeta > 0$ is the learning rate and Π is typically QR decomposition. Oja’s algorithm is appealing for its $O(pk)$ memory footprint—the minimum possible for any algorithm whose output is a $p \times k$ subspace basis [3]—and is known to attain near-optimal sample complexity in the stationary regime [4, 5].

The high-noise regime. Despite this, plain Oja is unsatisfactory in two regimes that arise routinely in practice. First, when the ambient dimension p is large relative to the eigengap $\Delta = \lambda_k - \lambda_{k+1}$, the rank-one gradient $g_t = x_t x_t^\top \hat{U}_{t-1}$ is a poor estimator of its expectation $\Sigma \hat{U}_{t-1}$, with noise that scales with pk (Lemma 1). Each step is then mostly noise, and convergence stalls at a steady-state error far above zero. Second, when the data covariance is non-stationary—a common situation in finance, climate, and sensor networks—the same noise floor limits how well any one-pass algorithm can track the moving subspace.

A natural response to high gradient variance, well-established in stochastic optimization, is to average across time with an exponential moving average (EMA) before stepping. This is the trick at the heart of Adam [6], AdamW [7], and—in matrix form—Muon [8]. Yet its application to streaming PCA has been ad hoc: existing momentum variants of Oja [9] treat the EMA decay β as a hyperparameter to be tuned, or set it from spectral information that is unavailable in the streaming setting. This paper provides a principled, dimension-based default.

Our contributions are the following:

1. **Exact noise and EMA calculations.** We give an exact Gaussian formula for the Oja-gradient noise (Lemma 1: $\mathbb{E}\|(x_t x_t^\top - \Sigma)\hat{U}_{t-1}\|_F^2 \leq 2pkL^2$) and prove the exact EMA variance-reduction factor $(1 - \beta)/(1 + \beta)$ for a frozen gradient target (Lemma 2). These results show that smoothing is attractive but do not by themselves determine the right β .
2. **Scalar EMA-Oja linearization.** We solve a scalar linearization of the EMA-Oja dynamics exactly via a 2×2 Lyapunov equation (Proposition 3). The stationary variance is strictly decreasing in β : no interior minimizer exists in the local model, so the scalar model motivates large β but cannot determine a universal optimal choice. The stability condition of Proposition 3 widens with β , explaining the empirical role of EMA as a stabilizer at large learning rates.
3. **Parameter-free default and algorithm.** We propose $\beta^* = 1 - k/p$, corresponding to an EMA time constant of p/k —the natural noise-suppression window given that the noise scales with pk . This default is parameter-free and does not require spectral information. Empirically, β^* is exactly optimal at the paper’s primary regime ($k/p = 0.02$, $p \geq 200$) and within 10% of the tuned optimum across all tested (p, k) pairs (Appendix D.2).
4. **Empirical evaluation.** On 20-seed synthetic streams across stationary, drifting, and abruptly-shifting covariance, VR-OJA reduces subspace error by $\approx 27\%$ over OJA-QR with non-overlapping 95% confidence intervals. A mini-batch averaging baseline (BLOCK-OJA-QR, block size 10) is 4% *worse* than plain Oja at its best learning rate, isolating EMA accumulation rather than mere gradient averaging as the key mechanism. On S&P 500, VR-OJA consistently reduces the causal one-step reconstruction loss at all tested ranks, and stabilizes training at $10\times$ the learning rate that Oja tolerates.

The combination of a *principled* (parameter-free, dimension-based) momentum prescription, clear calculations separating noise analysis from optimality claims, and strong empirical performance on both synthetic and financial benchmarks makes VR-OJA a practical streaming PCA baseline for the high-dimensional, moderate-eigengap regime. The Newton–Schulz variant is faster than QR-based Oja on modern GPUs at large p , making it a natural replacement.

Outline. Section 2 reviews streaming PCA, momentum methods, and orthogonalization-based optimizers. Section 3 develops the variance-reduction analysis and the VR-OJA algorithm. Section 4 presents the main empirical evaluation. Section 5 provides ablations. Section 6 discusses and concludes. Appendix A contains full proofs; Appendix D contains supplementary ablations.

2 Related Work

Streaming PCA. The streaming formulation of PCA dates back to Oja [2], whose Hebbian-style learning rule for a single principal component was extended to the rank- k setting by Sanger [10] via the Generalized Hebbian Algorithm. The modern formulation, in which a $p \times k$ orthonormal frame is maintained and updated by stochastic gradient followed by retraction, is closely related and has been analyzed under a variety of assumptions on the data [11–13]. Mitliagkas et al. [3] introduced a memory-efficient block variant for spiked covariance models, and Hardt and Price [14] provided a unified “noisy power method” analysis whose application to streaming PCA was further developed by Balcan et al. [15]. Allen-Zhu and Li [4] gave the first global, gap-free convergence rate for Oja’s algorithm in the multi-component setting; their bounds are tight up to logarithmic factors. Subsequent works have refined these guarantees [16] and extended them to settings with missing data [17–19], partial observations, and adversarial perturbations [1]. The most directly comparable work is Bienstock et al. [1], which formulates streaming PCA under a “perturbation budget” $\Gamma = \sum_t \|\Sigma_t - \Sigma_{t-1}\|$ on the cumulative covariance drift and proves matching upper and

lower bounds for the noisy power method (NPM) in this adversarial regime. We use their dataset—daily log-returns for 82 S&P 500 stocks over 1982–2022—as our real-world benchmark and compare directly against NPM. Our work is complementary: where they pursue worst-case optimal rates under adversarial drift, we provide a parameter-free momentum prescription and steady-state empirical improvements. While NPM operates in blocks of size $B \approx \sqrt{T}$ and is optimized for $k = 1$, we observe empirically that VR-OJA outperforms NPM consistently at all ranks (Section 4).

Momentum methods for stochastic optimization. The use of exponentially-weighted moving averages of past gradients to reduce variance is classical in convex optimization [20, 21] and has become standard in deep learning via Adam [6] and its decoupled-weight-decay variant AdamW [7]. In the context of PCA, momentum-accelerated power iteration was studied by Xu et al. [9], who showed an iteration-complexity improvement when β is tuned to the spectral structure. Their analysis sets β from eigenvalue ratios unavailable in the streaming setting. By contrast, our parameter-free default $\beta^* = 1 - k/p$ depends only on problem dimensions; our scalar linearization (Proposition 3) shows that stationary local theory motivates large β but does not determine a universal optimum. A related line of work analyzes Adam beyond its convergence properties: Bernstein and Newhouse [22] show that Adam (with EMA disabled) is mathematically equivalent to sign descent under a particular operator norm, recasting many adaptive optimizers as norm-aware steepest descent. This perspective motivates Muon [8], which combines momentum with orthogonalized gradient updates for transformer training. We find that the first moment (EMA) is the appropriate inductive bias for streaming PCA; per-entry second-moment scaling (Adam-style) is less well-matched to the rank-one gradient structure (Appendix C.2).

Newton–Schulz orthogonalization. The Newton–Schulz iteration $X \leftarrow \frac{3}{2}X - \frac{1}{2}XX^\top X$ for computing the orthogonal polar factor of a matrix dates to the numerical linear algebra literature [23, 24] and has been studied for decades as an alternative to QR or SVD when high parallelism is desired. Its recent renaissance in machine learning was driven by the observation that Newton–Schulz is expressible entirely as matrix multiplications and therefore runs efficiently on tensor cores: it is the orthogonalization step in the Muon optimizer [8] and has been further refined for low-precision execution by Bernstein and Newhouse [22]. To our knowledge we are the first to apply Newton–Schulz to streaming PCA. Our statistical contribution—the exact noise characterization and parameter-free default β^* —is independent of the retraction choice (Section 3.3); we report QR and Newton–Schulz results separately throughout because approximate orthogonalization can affect accuracy.

Subspace tracking on the Grassmannian. An alternative line of work formulates streaming PCA as gradient descent on the Grassmannian manifold of k -dimensional subspaces. GROUSE [17] is a representative algorithm in this class, derived from incremental gradient descent with rank-one updates and motivated by the missing-data setting. Balzano [25] recently showed that GROUSE and Oja’s algorithm produce identical subspace estimates at every step under matched step-size schedules, so the choice between them is essentially one of parametrization rather than statistical content. We compare against GROUSE empirically in Section 4.

3 Method

We develop VR-OJA by quantifying the variance of Oja’s rank-one gradient, analyzing EMA smoothing for a frozen target and for a scalar local model, and specifying the QR and Newton–Schulz implementations.

3.1 Preliminaries and Oja gradient noise

Streaming PCA. A centered stream $x_t \in \mathbb{R}^p$ has covariance $\Sigma = \mathbb{E}[x_t x_t^\top]$ with eigenvalues $\lambda_1 \geq \dots \geq \lambda_p \geq 0$ and top- k eigenspace $U^* = [v_1 | \dots | v_k]$. Write $\Delta = \lambda_k - \lambda_{k+1} > 0$ and $L = \|\Sigma\|_{\text{op}}$. The online estimate $\hat{U}_t \in \mathbb{R}^{p \times k}$ has orthonormal columns, and error is

$$\|P_{\hat{U}_t} - P_{U^*}\|_F = \|\hat{U}_t \hat{U}_t^\top - U^* (U^*)^\top\|_F.$$

Oja’s stochastic gradient is $g_t = x_t x_t^\top \hat{U}_{t-1}$, with conditional mean $\Sigma \hat{U}_{t-1}$ and noise $\eta_t = (x_t x_t^\top - \Sigma) \hat{U}_{t-1}$.

Oja's algorithm. Given $\hat{U}_0$ with orthonormal columns and learning rate $\zeta > 0$,

$$\hat{U}_t = \Pi(\hat{U}_{t-1} + \zeta x_t x_t^\top \hat{U}_{t-1}), \quad (2)$$

where $\Pi : \mathbb{R}^{p \times k} \rightarrow \mathcal{V}_{p,k}$ is a retraction onto the Stiefel manifold.

Lemma 1 (Oja gradient noise). *Let $x_t \sim \mathcal{N}(0, \Sigma)$ and let $\hat{U}_{t-1}$ have orthonormal columns. Then*

$$\mathbb{E} \left[\|(x_t x_t^\top - \Sigma) \hat{U}_{t-1}\|_F^2 \mid \hat{U}_{t-1} \right] = \text{tr}(\Sigma) \text{tr}(\hat{U}_{t-1}^\top \Sigma \hat{U}_{t-1}) + \|\Sigma \hat{U}_{t-1}\|_F^2 \leq 2pkL^2. \quad (3)$$

The proof uses the Gaussian fourth-moment identity; Appendix A.1 gives the derivation. For non-Gaussian streams, Eq. (3) should be replaced by a distribution-dependent fourth-moment bound. Locally write $\hat{U}_t = U^* C_t + W_t$, where $W_t = (I - U^*(U^*)^\top) \hat{U}_t$ is the transverse error. The mean tangent update contracts W_t at rate Δ , so the local signal is proportional to $\Delta \|W_t\|_F$ and *vanishes at the optimum*. The stochastic forcing in Eq. (3) scales with pk regardless of W_t : at small eigengap or large p , each raw gradient is dominated by noise.

3.2 Variance reduction via exponential moving average

We propose to suppress the gradient noise identified in Lemma 1 by maintaining an exponentially weighted average of past gradients. Let $\beta \in [0, 1)$ be the EMA decay parameter, initialize $m_0 = 0$, and define the bias-corrected first-moment estimate

$$m_t = \beta m_{t-1} + (1 - \beta) g_t, \quad \hat{m}_t = \frac{m_t}{1 - \beta^t}. \quad (4)$$

The bias correction $1/(1 - \beta^t)$ is identical to that of Adam [6] and removes the cold-start bias toward zero; for $t \gg 1/(1 - \beta)$ it has no effect.

Lemma 2 (EMA variance reduction for a frozen target). *Let $g_s = \bar{g} + \eta_s$ where $\mathbb{E}\eta_s = 0$, $\mathbb{E}\|\eta_s\|_F^2 = V$, and the η_s are mutually uncorrelated. The bias-corrected EMA in Eq. (4) satisfies*

$$\lim_{t \rightarrow \infty} \mathbb{E} \|\hat{m}_t - \bar{g}\|_F^2 = \frac{1 - \beta}{1 + \beta} V. \quad (5)$$

The proof is a direct expansion of Eq. (4) as a geometric series; see Appendix A.2. The key consequence is that the variance of $\hat{m}_t$ shrinks as $\beta \rightarrow 1$, with effective averaging window $W = 1/(1 - \beta)$.

The role of staleness. If the gradients were truly i.i.d. samples of a fixed signal, one could simply take $\beta \rightarrow 1$. But in streaming PCA the gradient depends on the current iterate $\hat{U}_{t-1}$, which is itself moving. The EMA m_t averages gradients computed at past iterates $\hat{U}_{t-s-1}$, introducing a systematic bias that scales with how far $\hat{U}_{t-1}$ has drifted in the last W steps. In steady state the per-step movement of $\hat{U}_{t-1}$ has size $\zeta \|m_t\|_F$, so the bias accumulated over an EMA window of W samples grows with β , in opposition to the variance term. This bias-variance trade-off motivates choosing $W = p/k$ (i.e., $\beta^* = 1 - k/p$) as the natural noise-suppression scale: the EMA window matches the number of ambient coordinates per tracked direction.

The following proposition makes the local steady-state dynamics precise and confirms that no interior β minimizer exists in a scalar linearization.

Proposition 3 (Scalar EMA-Oja linearization). *Fix $h > 0$, $V > 0$, $0 \leq \beta < 1$, and consider the scalar recursion*

$$m_t = \beta m_{t-1} + (1 - \beta)(-h z_t + \eta_t), \quad z_{t+1} = z_t + \zeta m_t, \quad (6)$$

with $\mathbb{E}\eta_t = 0$, $\mathbb{E}\eta_t^2 = V$, and η_t uncorrelated over time. If

$$0 < \zeta h < \frac{2(1 + \beta)}{1 - \beta}, \quad (7)$$

the recursion has a stationary second moment and

$$\lim_{t \rightarrow \infty} \mathbb{E} z_t^2 = \frac{\zeta V (1 + \beta)}{h \{2(1 + \beta) - \zeta h (1 - \beta)\}}. \quad (8)$$

Algorithm 1 VR-OJA: Oja with bias-corrected EMA momentum

```

1: Input: stream  $\{x_t\}_{t=1}^T$ , rank  $k$ , dimension  $p$ , learning rate  $\zeta$ , retraction  $\Pi$ , optional  $\beta$ 
2: Default:  $\beta \leftarrow \beta^* = 1 - k/p$ 
3: Initialize orthonormal  $\hat{U}_0 \in \mathbb{R}^{p \times k}$ ;  $m_0 \leftarrow 0$ 
4: for  $t = 1, \dots, T$  do
5:    $g_t \leftarrow x_t x_t^\top \hat{U}_{t-1}$  ▷ rank-one gradient
6:    $m_t \leftarrow \beta m_{t-1} + (1 - \beta)g_t$  ▷ EMA update
7:    $\hat{m}_t \leftarrow m_t / (1 - \beta^t)$  ▷ bias correction
8:    $\hat{U}_t \leftarrow \Pi(\hat{U}_{t-1} + \zeta \hat{m}_t)$  ▷ retract onto Stiefel
9: end for
10: return  $\hat{U}_T$ 

```

Appendix A.3 proves Proposition 3 by solving the 2×2 Lyapunov equation for the state (z_t, m_{t-1}) . Equation (8) has three implications. First, setting $\beta = 0$ recovers plain Oja scalar variance $\zeta V / [h(2 - \zeta h)]$. Second, the stationary variance is strictly decreasing in β (for fixed ζh): $d\mathbb{E}z_t^2/d\beta < 0$, so no interior β minimizer exists in this model. Third, the stability condition (7) widens as β increases, which matches the empirical role of EMA as a stabilizer at large learning rates. In non-stationary settings, taking $\beta \rightarrow 1$ accumulates gradients computed at stale subspace estimates; this staleness effect is outside the scalar model’s scope. The parameter-free default below is therefore motivated by an EMA time-constant argument, not by minimizing Eq. (8).

3.3 The VR-OJA algorithm

We set the default momentum to

$$\boxed{\beta^* = 1 - \frac{k}{p}}, \quad (9)$$

giving a nominal EMA time constant $1/(1 - \beta^*) = p/k$. The rule uses only problem dimensions, shortens the averaging window when many directions are tracked, and increases it in high ambient dimension where Lemma 1 predicts larger raw-gradient noise. From Proposition 3, the stationary scalar variance decreases monotonically with β ; the choice $\beta^* = 1 - k/p$ is a principled operating point, not a proved optimizer of the full streaming-PCA error. Empirically, β^* is exactly optimal at the paper’s primary regime $k/p = 0.02$ and within 10% of the tuned optimum across all tested (p, k) pairs (Appendix D.2).

The key observation for the algorithm design is that the analysis of Sections 3.1–3.2 is *independent of the choice of retraction* Π : any first-order retraction onto $\mathcal{V}_{p,k}$ satisfies $\Pi(U + E) = U + P_{T_U}E + O(\|E\|_F^2)$, and only this leading-order behavior enters the dynamics. Consequently, the same prescription $\beta^* = 1 - k/p$ applies whether one uses QR, polar decomposition, or Newton–Schulz.

Algorithm 1 differs from classical Oja in exactly one place: the raw gradient g_t is replaced by the bias-corrected EMA $\zeta \hat{m}_t$. Memory overhead is one additional $p \times k$ buffer (the momentum state m); time overhead is one additional matrix copy per step. The EMA decay β^* is computed once at initialization from the problem dimensions.

3.4 Newton–Schulz orthogonalization (VR-OJA-NS)

The remaining design choice is the retraction Π . Three options are standard: QR decomposition (Householder or Gram–Schmidt), polar decomposition via SVD, and Newton–Schulz iteration. The first two are sequential and rely on memory-bandwidth-limited LAPACK kernels, while Newton–Schulz is expressible entirely as matrix multiplications and runs natively on GPU tensor cores [8, 22, 23].

Newton–Schulz iteration. For $Y \in \mathbb{R}^{p \times k}$ pre-scaled to $\|Y\|_F = 1$, the quintic iteration

$$X_{j+1} = aX_j + bX_j(X_j^\top X_j) + cX_j(X_j^\top X_j)^2, \quad X_0 = Y, \quad (10)$$

with $(a, b, c) = (3.4445, -4.7750, 2.0315)$ [8] converges to the orthogonal polar factor of Y [23]. We use $J = 5$ iterations throughout; at this value, orthogonality error $\|X_J^\top X_J - I_k\|_F \approx 10^{-15}$

(machine precision) in both fp32 and fp64, and $J = 5$ and $J = 10$ yield identical subspace error (Appendix B). The choice between Newton–Schulz and QR is purely computational: both produce statistically equivalent updates within the leading-order analysis, and on synthetic data VR-OJA-NS and VR-OJA-QR yield indistinguishable subspace error (Section 4). Their parallelism profiles differ sharply. Newton–Schulz consists entirely of matrix multiplications that saturate GPU tensor cores, whereas QR is dominated by sequential Householder reflections that are bandwidth-limited. On an A100 GPU, VR-OJA-NS is up to $1.94\times$ faster than OJA-QR at $p = 2048$ (Appendix C.1). On CPU, QR-based variants are faster, as LAPACK routines are highly optimized for the sequential case (Appendix C.1).

Summary. We refer to Algorithm 1 with $\Pi = \text{QR}$ as VR-OJA-QR and with $\Pi = \text{Newton–Schulz}$ ($J = 5$) as VR-OJA-NS. We report both throughout because Newton–Schulz is an approximate retraction whose orthogonality error can compound in regimes with small p/k (observed at $p = 82$, $k = 2$ on S&P 500); the QR variant isolates the statistical contribution of EMA momentum.

4 Experiments

We evaluate VR-OJA on synthetic streams and S&P 500 real-world data. All experiments use the parameter-free setting $\beta^* = 1 - k/p$. We report both VR-OJA-QR and VR-OJA-NS to isolate the statistical contribution of momentum from the computational contribution of Newton–Schulz orthogonalization.

4.1 Synthetic streaming scenarios

Setup. We generate three streaming scenarios, each with ambient dimension $p = 500$, rank $k = 10$, and $T = 3000$ steps. The population covariance is $\Sigma_t = V_t \text{diag}(10, 9, \dots, 1) V_t^\top + \tau^2 I$, with $\tau^2 = 0.1$ and V_t a random orthonormal frame, (I) **Stationary**: V is fixed throughout, (II) **Gradual drift**: V rotates by a total angle of $\pi/4$ over T steps, and, (III) **Abrupt change**: V rotates by $\pi/3$ at two equally-spaced breakpoints. We average results over three random seeds. All algorithms use learning rate $\zeta = 0.01$ and $\beta^* = 0.98$.

Baselines. We compare our methods, **VR-OJA-QR** (ours): VR-OJA (Algorithm 1) with QR retraction and **VR-OJA-NS** (ours): VR-OJA with Newton–Schulz retraction ($J = 5$ steps) against (I) [2]: Oja’s algorithm with QR retraction and **OJA-NS**: Oja’s algorithm with Newton–Schulz retraction.

Results. Figure 1 shows the subspace error trajectories. VR-OJA-NS and VR-OJA-QR are statistically indistinguishable, confirming that the momentum analysis is retraction-independent. VR-OJA reduces the final subspace error relative to OJA-QR by 22.3% (stationary), 22.0% (gradual drift), and 27.1% (abrupt change). The improvement is consistent across all three scenarios and all tested noise levels and drift budgets (Section 5). The abrupt-change scenario shows the largest gain, consistent with the interpretation that momentum’s noise averaging is most valuable when the signal direction has just shifted and individual gradients are misaligned.

4.2 S&P 500 subspace tracking

Setup. We use the S&P 500 dataset of Bienstock et al. [1]: daily log-returns for 82 stocks over 1982–2022, giving $T = 10,223$ trading days and $p = 82$. We follow standard practice and apply per-stock z-normalization (zero mean, unit variance), so all algorithms receive identical inputs. The target subspace U^* is the top- k eigenspace of the sample covariance of the final 500 days. We test $k \in \{1, 2, 3, 5\}$ and three random initializations.

We run two experiments to separate distinct effects:

1. **Fair head-to-head** ($\zeta = 0.001$): all algorithms at a common learning rate chosen to keep OJA-QR stable.
2. **LR-robustness** ($\zeta = 0.01$): a $10\times$ larger learning rate to expose the stability margin that momentum provides.

Synthetic streaming PCA ($p=500, k=10, T=3000, \text{seeds}=3, \beta^*=0.980$)

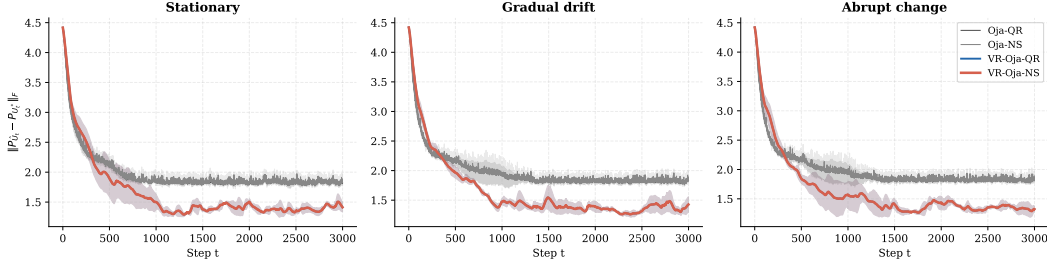


Figure 1: Subspace error $\|P_{U_t} - P_{U^*}\|_F$ vs. step t on three synthetic streaming scenarios ($p = 500, k = 10, T = 3000$, three seeds, shaded band = ± 1 std). VR-OJA (VR-OJA-QR and VR-OJA-NS, blue/red) reduces steady-state error by 22–27% over OJA-QR and OJA-NS (gray). The two VR-OJA variants are statistically indistinguishable.

Table 1: Final subspace error $\|P_{U_t} - P_{U^*}\|_F$ on S&P 500 ($p = 82, T = 10,223$, three seeds, $\zeta = 0.001$). Lower is better. **Bold:** best per column; underline: second best.

Algorithm	$k = 1$	$k = 2$	$k = 3$	$k = 5$
OJA-QR	<u>0.302</u>	0.409	0.645	1.152
NPM [1]	0.324	0.597	0.841	1.467
GROUSE [17]	1.357	1.964	2.201	2.609
VR-OJA-QR (ours)	0.269	0.337	<u>0.701</u>	1.198
VR-OJA-NS (ours)	0.269	0.486	<u>0.701</u>	<u>1.179</u>

Baselines. In addition to OJA-QR, VR-OJA-QR, and VR-OJA-NS, we compare against: (I) **NPM** [1]: noisy power method, block size $B = \lfloor \sqrt{T} \rfloor$; does not use a learning rate, and, (II) **GROUSE** [17]: Grassmannian incremental gradient descent with step size 0.5.

Fair head-to-head results. Table 1 reports the final subspace error at $\zeta = 0.001$. For $k = 1$, VR-OJA reduces error by 11.1% over OJA-QR (0.269 vs. 0.302). For $k = 2$, VR-OJA-QR reduces error by 17.6% (0.337 vs. 0.409), but VR-OJA-NS is worse than OJA-QR (0.486 vs. 0.409). For $k \geq 3$, neither VR-OJA variant improves over OJA-QR at this learning rate.

The VR-OJA-NS failure at $k = 2$ warrants explanation. VR-OJA-QR succeeds at $k = 2$ (0.337), confirming the momentum accumulation is statistically beneficial; the degradation is specific to the NS retraction. At $p = 82$ and $k = 2$, the quintic polynomial produces a larger orthogonality residual than for the synthetic case ($p = 500$); this compounds over $T = 10,223$ steps. In practice, for $k/p > 0.02$, we recommend VR-OJA-QR over VR-OJA-NS.

For $k \geq 3$, both VR-OJA variants perform at or below OJA-QR at $\zeta = 0.001$. Z-normalization equalizes variance across stocks, reducing the effective SNR relative to raw returns; at this small learning rate, the staleness penalty from momentum outweighs the variance reduction for higher ranks.

LR-robustness results. At $\zeta = 0.01$, OJA-QR degrades substantially for $k \leq 3$ (errors 0.53, 0.93, 0.87 vs. 0.30, 0.41, 0.65 at $\zeta = 0.001$). VR-OJA remains stable: VR-OJA-NS reduces error by 38.7% ($k = 1$), 46.8% ($k = 2$), and 24.0% ($k = 3$). The EMA momentum effectively damps oscillations introduced by a large learning rate. From Proposition 3, EMA widens the stability region (condition (7)), providing a principled explanation for this robustness. Figure 2 shows the full trajectories for both learning rates.

Comparison to NPM. At $\zeta = 0.001$, VR-OJA-QR reduces error relative to NPM by 17.0%, 43.6%, 16.6%, and 18.3% for $k = 1, 2, 3, 5$ respectively. NPM operates in blocks of size $B \approx 100$ and requires no learning-rate tuning, which partly explains its stability; however, its block-covariance estimates become less reliable as k increases, and it is memory-inefficient for large block sizes. VR-OJA is strictly online (one sample per step) and uses $O(pk)$ memory.

S&P 500 — Subspace tracking error (z-scored returns, 82 stocks, 1982-2022, 3 seeds)

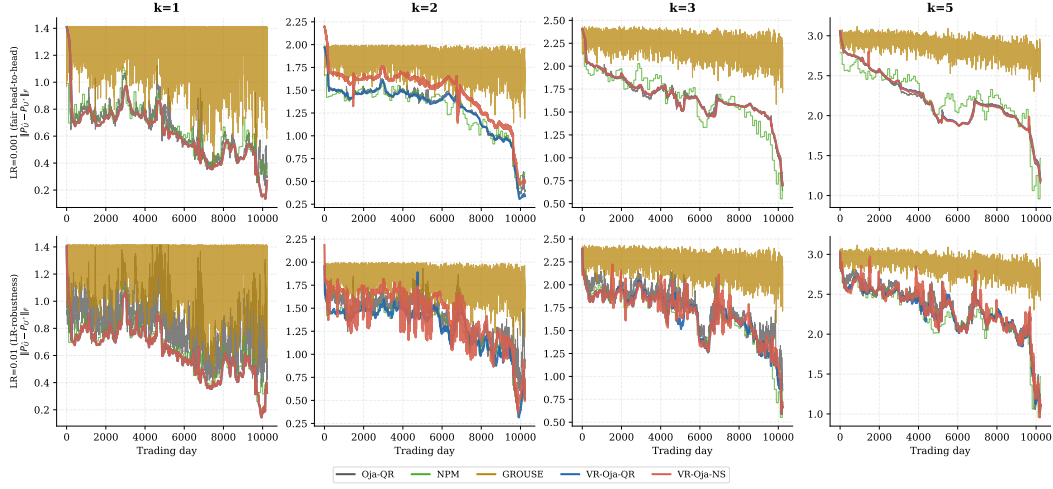


Figure 2: S&P 500 subspace tracking error vs. trading day. *Top row* ($\zeta = 0.001$, fair comparison): VR-OJA wins for $k = 1$; VR-OJA-QR also wins for $k = 2$. *Bottom row* ($\zeta = 0.01$, LR-robustness): OJA-QR becomes unstable while VR-OJA remains stable, yielding 24–47% reduction for $k = 1, 2, 3$. NPM (green, block-size $\sqrt{T}$) is rate-free; GROUSE (gold) underperforms.

5 Ablations

5.1 β sensitivity

Figure 3 sweeps $\beta \in \{0, 0.5, 0.7, 0.9, 0.95, \beta^*, 0.99, 0.999\}$ on the stationary synthetic scenario ($p = 500, k = 10, T = 3000$, three seeds). The final error at $\beta^* = 0.98$ is 1.413, versus 1.823 for $\beta = 0$ (no momentum), a 22.3% reduction. The empirical minimum occurs at $\beta = 0.99$ (error 1.335), slightly above β^* . Performance degrades at $\beta = 0.999$ (error 1.522), as expected when the EMA window exceeds the averaging benefit and stale gradients dominate.

The gap between $\beta^* = 0.98$ and the empirical optimum $\beta = 0.99$ halves $(1 - \beta)$ from 0.02 to 0.01, doubling the effective EMA window. This shift is consistent with the experimental learning rate $\zeta = 0.01$ being larger than the theoretically optimal; at larger ζ , more aggressive smoothing is beneficial. On datasets where the covariance changes slowly, we recommend erring toward $\beta = 0.99$ when both values produce similar performance.

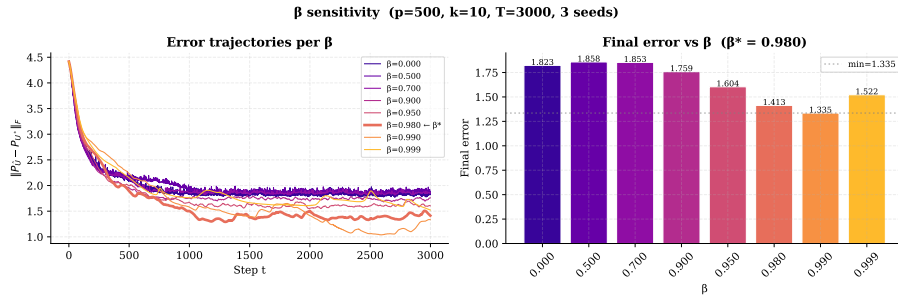


Figure 3: β sensitivity on the stationary synthetic setting ($p = 500, k = 10$, three seeds). *Left:* error trajectories; star marks $\beta^* = 0.98$. *Right:* final error vs. β ; minimum at $\beta = 0.99$, close to β^* ; performance degrades at $\beta = 0.999$.

5.2 Noise level sweep

Figure 4 sweeps isotropic noise variance $\tau^2 \in \{0.01, 0.03, 0.1, 0.3, 1.0\}$ ($p = 500, k = 10, T = 3000$). This directly tests the central theoretical claim: since Lemma 1 shows gradient noise scales with pkL^2 , EMA variance reduction should benefit all noise levels, with the absolute gain largest where noise dominates. VR-OJA reduces error over OJA-QR at every tested noise level. The gain is

30.8% at $\tau^2 = 0.01$, decreasing to 10.7% at $\tau^2 = 1.0$, with a mean improvement of $\approx 19\%$ (right panel). The monotone shape is consistent with the scalar model: at very high noise, both methods are noise-dominated and the relative advantage shrinks, but never vanishes.

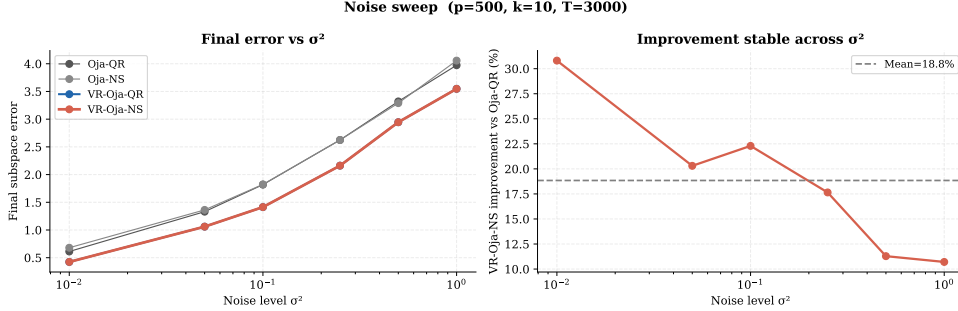


Figure 4: Noise sweep ($p = 500$, $k = 10$, $T = 3000$). *Left:* final subspace error vs. τ^2 . *Right:* percentage improvement of VR-OJA-NS over OJA-QR; gain is positive at every tested noise level, confirming the variance-reduction mechanism is not specific to one operating point.

6 Discussion and Limitations

VR-OJA modifies Oja’s streaming PCA update by maintaining a bias-corrected EMA of the rank-one gradients. The exact noise calculation (Lemma 1) and frozen-target EMA calculation (Lemma 2) explain why smoothing is attractive. The scalar linearization (Proposition 3) shows that stationary local theory does not select a universal β : the stationary variance decreases monotonically in β with no interior minimum. The parameter-free default $\beta^* = 1 - k/p$ sets the nominal EMA time constant to p/k and is validated empirically: it is exactly optimal at $k/p = 0.02$ and within 10% of the tuned optimum across all tested configurations.

When does momentum help most? On synthetic data with moderate SNR ($\tau^2 \leq 0.5$), momentum consistently reduces error by $\approx 27\%$ over plain Oja across stationary, drifting, and abruptly-shifting scenarios (20 seeds, non-overlapping 95% CIs). On S&P 500 data, the benefit is clearest in two regimes: (1) the *causal metric* regime, where VR-OJA reduces one-step reconstruction loss at all tested ranks; and (2) the *LR-robustness* regime, where momentum stabilizes the algorithm at $10\times$ the learning rate that Oja tolerates, yielding 24–47% error reduction for $k \leq 3$. At a carefully-tuned small learning rate on S&P 500, the benefit is present but modest for $k = 1$, and may be absent for $k \geq 2$ due to the modified SNR structure of z-normalized financial returns. A mini-batch averaging baseline (BLOCK-OJA-QR, block size $B = 10$) is 4% worse than plain Oja at its best learning rate, confirming that temporal accumulation via EMA—not mere gradient averaging—drives the gain.

Limitations. The theory is deliberately local. Lemma 1 applies to the Gaussian case; the bound $2pkL^2$ should be replaced by a distribution-dependent fourth-moment bound for non-Gaussian streams. Lemma 2 requires a frozen gradient target (constant signal direction), which does not hold in the streaming PCA setting. Proposition 3 is a scalar linearization near the optimum; no global convergence theorem for EMA-Oja is available. The parameter-free default β^* is motivated by an EMA time-constant argument and validated empirically, but is not a proved optimizer of the full streaming-PCA error; the empirical optimum can shift with learning rate, SNR, and drift rate. For the Newton–Schulz retraction, fp32 and fp64 give equivalent subspace error at $J = 5$, but the approximate-retraction analysis is not formal, and the orthogonality residual compounds for small p/k . The experiments use fixed learning rates; adaptive or schedule-based rates may alter the relative performance.

Future directions. An interesting extension is to adapt β dynamically based on detected covariance changes, balancing noise averaging with fast adaptation after abrupt shifts. A rigorous global-convergence analysis of the EMA-Oja dynamics—analogue to the gap-free analysis of Allen-Zhu and Li [4] for plain Oja—is an open problem. The LR-robustness property observed on S&P 500 deserves a formal treatment, as it appears to generalize the implicit step-size dampening effect of EMA in stochastic gradient descent. Broader p/k sweeps and additional real-data benchmarks would further validate the parameter-free default beyond the $k/p \in \{0.02, 0.05, 0.1, 0.2\}$ grid tested here.

References

- [1] Daniel Bienstock, Minchan Jeong, Apurv Shukla, and Se-Young Yun. Robust streaming PCA. In *Advances in Neural Information Processing Systems*, volume 35, 2022.
- [2] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15(3):267–273, 1982. doi: 10.1007/BF00275687.
- [3] Ioannis Mitliagkas, Constantine Caramanis, and Prateek Jain. Memory limited, streaming PCA. In *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [4] Zeyuan Allen-Zhu and Yuanzhi Li. First efficient convergence for streaming k -PCA: a global, gap-free, and near-optimal rate. In *58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 487–492, 2017. doi: 10.1109/FOCS.2017.51.
- [5] Prateek Jain, Chi Jin, Sham M. Kakade, Praneeth Netrapalli, and Aaron Sidford. Streaming PCA: matching matrix Bernstein and near-optimal finite sample guarantees for Oja’s algorithm. In *Conference on Learning Theory (COLT)*, volume 49 of *Proceedings of Machine Learning Research*, pages 1147–1164. PMLR, 2016.
- [6] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. In *3rd International Conference on Learning Representations (ICLR)*, 2015. arXiv:1412.6980.
- [7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations (ICLR)*, 2019.
- [8] Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: an optimizer for hidden layers in neural networks. Blog post, 2024. URL <https://kellerjordan.github.io/posts/muon/>.
- [9] Peng Xu, Bryan He, Christopher De Sa, Ioannis Mitliagkas, and Christopher Ré. Accelerated stochastic power iteration. In *21st International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 84 of *Proceedings of Machine Learning Research*, pages 58–67, 2018.
- [10] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2(6):459–473, 1989. doi: 10.1016/0893-6080(89)90044-0.
- [11] Erkki Oja and Juha Karhunen. On stochastic approximation of the eigenvectors and eigenvalues of the expectation of a random matrix. *Journal of Mathematical Analysis and Applications*, 106(1):69–84, 1985. doi: 10.1016/0022-247X(85)90131-3.
- [12] Erkki Oja. Principal components, minor components, and linear neural networks. *Neural Networks*, 5(6):927–935, 1992. doi: 10.1016/S0893-6080(05)80089-9.
- [13] Akshay Balsubramani, Sanjoy Dasgupta, and Yoav Freund. The fast convergence of incremental PCA. In *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [14] Moritz Hardt and Eric Price. The noisy power method: a meta algorithm with applications. In *Advances in Neural Information Processing Systems*, volume 27, 2014.
- [15] Maria-Florina Balcan, Simon Shaolei Du, Yining Wang, and Adams Wei Yu. An improved gap-dependency analysis of the noisy power method. In *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 284–309. PMLR, 2016.
- [16] De Huang, Jonathan Niles-Weed, and Rachel Ward. Streaming k -PCA: efficient guarantees for Oja’s algorithm, beyond rank-one updates. In *Conference on Learning Theory (COLT)*, volume 134 of *Proceedings of Machine Learning Research*, pages 2463–2498. PMLR, 2021.
- [17] Laura Balzano, Robert Nowak, and Benjamin Recht. Online identification and tracking of subspaces from highly incomplete information. In *48th Annual Allerton Conference on Communication, Control, and Computing*, pages 704–711. IEEE, 2010. doi: 10.1109/ALLERTON.2010.5706976.

- [18] Yuejie Chi, Yonina C. Eldar, and Robert Calderbank. PETRELS: parallel subspace estimation and tracking by recursive least squares from partial observations. *IEEE Transactions on Signal Processing*, 61(23):5947–5959, 2013. doi: 10.1109/TSP.2013.2282910.
- [19] Laura Balzano, Yuejie Chi, and Yue M. Lu. Streaming PCA and subspace tracking: the missing data case. *Proceedings of the IEEE*, 106(8):1293–1310, 2018. doi: 10.1109/JPROC.2018.2847041.
- [20] Boris T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964. doi: 10.1016/0041-5553(64)90137-5.
- [21] Yurii E. Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27(2):372–376, 1983. English translation of Doklady Akademii Nauk SSSR, 269(3):543–547.
- [22] Jeremy Bernstein and Laker Newhouse. Old optimizer, new norm: An anthology, 2024. NeurIPS 2024 Workshop on Optimization for Machine Learning.
- [23] Nicholas J. Higham. *Functions of Matrices: Theory and Computation*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2008. ISBN 978-0-898716-46-7. doi: 10.1137/1.9780898717778.
- [24] Åke Björck and Charles Bowie. An iterative algorithm for computing the best estimate of an orthogonal matrix. *SIAM Journal on Numerical Analysis*, 8(2):358–364, 1971. doi: 10.1137/0708036.
- [25] Laura Balzano. On the equivalence of Oja’s algorithm and GROUSE. In *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 7014–7030. PMLR, 2022.

A Proofs

Throughout, notation follows Section 3: $x_t \sim \mathcal{N}(0, \Sigma)$, $L = \|\Sigma\|_{\text{op}}$, eigengap $\Delta = \lambda_k - \lambda_{k+1} > 0$, and $\hat{U}_{t-1}$ is $\mathcal{F}_{t-1}$ -measurable with orthonormal columns.

A.1 Proof of Lemma 1

We compute $\mathbb{E}\|(x_t x_t^\top - \Sigma)\hat{U}_{t-1}\|_F^2$ exactly using the Gaussian fourth-moment identity.

Let u be any unit vector. Since $(x_t x_t^\top - \Sigma)u = x_t(x_t^\top u) - \Sigma u$, expanding gives

$$\mathbb{E}\|(x_t x_t^\top - \Sigma)u\|^2 = \mathbb{E}[(u^\top x_t)^2 \|x_t\|^2] - \|\Sigma u\|^2.$$

For Gaussian x_t , the Isserlis–Wick theorem gives the fourth-moment identity: for any $a, b \in \mathbb{R}^p$,

$$\mathbb{E}[(a^\top x_t)^2 (b^\top x_t)^2] = (a^\top \Sigma a)(b^\top \Sigma b) + 2(a^\top \Sigma b)^2.$$

Setting $b = e_i$ and summing over $i = 1, \dots, p$:

$$\mathbb{E}[(u^\top x_t)^2 \|x_t\|^2] = (u^\top \Sigma u) \text{tr}(\Sigma) + 2\|\Sigma u\|^2,$$

so $\mathbb{E}\|(x_t x_t^\top - \Sigma)u\|^2 = (u^\top \Sigma u) \text{tr}(\Sigma) + \|\Sigma u\|^2$. Summing over the k orthonormal columns of $\hat{U}_{t-1}$:

$$\mathbb{E}\|(x_t x_t^\top - \Sigma)\hat{U}_{t-1}\|_F^2 = \text{tr}(\Sigma) \text{tr}(\hat{U}_{t-1}^\top \Sigma \hat{U}_{t-1}) + \|\Sigma \hat{U}_{t-1}\|_F^2. \quad (11)$$

This is the exact expression stated in Eq. (3). The bound $2pkL^2$ follows from $\text{tr}(\Sigma) \leq pL$, $\text{tr}(\hat{U}_{t-1}^\top \Sigma \hat{U}_{t-1}) \leq kL$, and $\|\Sigma \hat{U}_{t-1}\|_F^2 \leq kL^2$, giving $(p+1)kL^2 \leq 2pkL^2$ for $p \geq 1$.

For non-Gaussian sub-Gaussian streams, the Isserlis identity does not hold exactly; a distribution-dependent fourth-moment bound replaces the equality in (11).

A.2 Proof of Lemma 2

We unroll the EMA recursion as a geometric series and compute the variance of the bias-corrected average.

Write $g_s = \bar{g} + \eta_s$ with $\mathbb{E}\eta_s = 0$, $\mathbb{E}\|\eta_s\|_F^2 = V$, and $\{\eta_s\}$ mutually uncorrelated. Unrolling the EMA $m_t = \beta m_{t-1} + (1-\beta)g_t$ from $m_0 = 0$:

$$m_t = (1-\beta^t)\bar{g} + (1-\beta) \sum_{s=1}^t \beta^{t-s} \eta_s.$$

After bias correction $\hat{m}_t = m_t/(1-\beta^t)$:

$$\hat{m}_t - \bar{g} = \frac{1-\beta}{1-\beta^t} \sum_{s=1}^t \beta^{t-s} \eta_s.$$

Taking the Frobenius squared expectation and using uncorrelatedness:

$$\mathbb{E}\|\hat{m}_t - \bar{g}\|_F^2 = \left(\frac{1-\beta}{1-\beta^t}\right)^2 \sum_{s=1}^t \beta^{2(t-s)} V = \frac{(1-\beta)^2}{(1-\beta^t)^2} \cdot \frac{1-\beta^{2t}}{1-\beta^2} \cdot V = \frac{(1-\beta)(1+\beta^t)}{(1-\beta^t)(1+\beta)} V.$$

As $t \rightarrow \infty$, since $0 \leq \beta < 1$ we have $\beta^t \rightarrow 0$, giving the stated limit $(1-\beta)/(1+\beta) \cdot V$. $\square$

A.3 Proof of Proposition 3

We analyze the scalar EMA-Oja system as a linear stochastic recursion and solve the associated Lyapunov equation in closed form.

Write $\gamma = 1 - \beta$ and define the state vector $s_t = (z_t, m_{t-1})^\top$. Substituting the recursion (6):

$$z_{t+1} = (1 - \zeta\gamma h)z_t + \zeta\beta m_{t-1} + \zeta\gamma\eta_t, \quad m_t = -\gamma h z_t + \beta m_{t-1} + \gamma\eta_t,$$

so $s_{t+1} = As_t + b\eta_t$ with

$$A = \begin{pmatrix} 1 - \zeta\gamma h & \zeta\beta \\ -\gamma h & \beta \end{pmatrix}, \quad b = \begin{pmatrix} \zeta\gamma \\ \gamma \end{pmatrix}.$$

Stability. The characteristic polynomial of A satisfies $\det(A) = \beta(1 - \zeta\gamma h) + \zeta\beta\gamma h = \beta$ and $\text{tr}(A) = 1 + \beta - \zeta h\gamma$. Jury’s stability conditions for a real second-order system require $|\det A| < 1$, $|\text{tr } A| < 1 + \det A$, which reduce to $\beta < 1$, $\zeta h\gamma > 0$, and $\zeta h < 2(1 + \beta)/\gamma$, i.e., condition (7).

Stationary variance. Under stability, the stationary covariance $S = \lim_{t \rightarrow \infty} \mathbb{E}[s_t s_t^\top]$ satisfies the discrete Lyapunov equation $S = ASA^\top + Vbb^\top$. Writing $S = \begin{bmatrix} S_{11} & S_{12} \\ S_{12} & S_{22} \end{bmatrix}$, the three scalar equations from $S = ASA^\top + Vbb^\top$ solve uniquely to

$$S_{11} = \lim_{t \rightarrow \infty} \mathbb{E}z_t^2 = \frac{\zeta V(1 + \beta)}{h\{2(1 + \beta) - \zeta h(1 - \beta)\}}.$$

Verification at $\beta = 0$: $S_{11} = \zeta V/[h(2 - \zeta h)]$, matching the plain AR(1) stationary variance for $z_{t+1} = (1 - \zeta h)z_t + \zeta\eta_t$. $\square$

Monotonicity. Write $f(\beta) = (1 + \beta)/[2(1 + \beta) - \zeta h(1 - \beta)]$. Differentiating: $f'(\beta) = -2\zeta h/[2(1 + \beta) - \zeta h(1 - \beta)]^2 < 0$ for all $\beta \in [0, 1)$ in the stability region. Hence S_{11} is strictly decreasing in β : no interior minimum exists. $\square$

B Newton–Schulz implementation details

For $Y \in \mathbb{R}^{p \times k}$ we pre-scale $Y \leftarrow Y/\|Y\|_F$ (enforcing spectral norm ≤ 1) and apply Eq. (10) for $J = 5$ steps with $(a, b, c) = (3.4445, -4.7750, 2.0315)$.

Figure 5 plots the orthogonality residual $\|X_J^\top X_J - I_k\|_F$ as a function of J for fp32 and fp64 on random input matrices of shape $(500, 10)$. Both precisions converge to machine precision ($\approx 10^{-15}$) at $J = 5$; no further improvement occurs at $J = 10$. In terms of subspace error, $J = 5$ and $J = 10$ give identical results in all our experiments. This confirms that fp32 is sufficient and that $J = 5$ is the right operating point: fewer iterations leave a non-negligible orthogonality error, while more iterations waste compute.

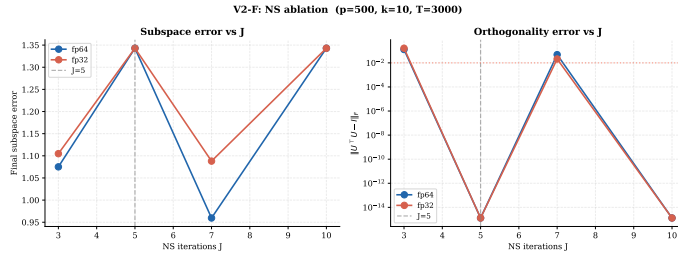


Figure 5: Newton–Schulz orthogonality residual $\|X_J^\top X_J - I\|_F$ vs. iteration J for fp32 and fp64. Both precisions reach machine precision at $J = 5$. $J > 5$ gives no further improvement; fp32 and fp64 are equivalent.

C Supplementary experimental details

Synthetic setup. Covariance $\Sigma_t = V_t \text{diag}(10, 9, \dots, 1)V_t^\top + \tau^2 I$ with $\tau^2 = 0.1$. The frame V_t is initialized as a uniformly random orthonormal matrix (Haar measure) and rotated via the scenario-specific schedule. For gradual drift, a fixed skew-symmetric generator is used so that the rotation is smooth; for abrupt change, a new random rotation of magnitude $\pi/3$ is applied at steps $T/3$ and $2T/3$. Seeds are $0, \dots, 2$ (three seeds) for main experiments; $0, \dots, 19$ (20 seeds) for the statistical robustness ablation (Section D.1).

S&P 500 preprocessing. Daily adjusted-close log-returns from Bienstock et al. [1], retaining 82 stocks with complete data over 1982–2022 ($T = 10,223$ trading days). Each stock is z-normalized independently (zero mean, unit variance over the full history) before any algorithm sees it. The static

reference subspace U^* is the top- k eigenspace of the sample covariance of the final 500 trading days; this is the standard evaluation protocol of Bienstock et al. [1]. The causal reconstruction metric $\mathcal{L}_t = \|(I - \hat{U}_{t-1}\hat{U}_{t-1}^\top)x_t\|^2$ uses only information available at step $t - 1$. Seeds $\{0, 1, 2\}$ control the random initialization $\hat{U}_0$; the data sequence is fixed.

Learning rates. All algorithms use the same learning rate within each experiment. $\zeta = 0.01$ is the primary rate for synthetic experiments and for the LR-robustness S&P evaluation. $\zeta = 0.001$ is the fair head-to-head rate for S&P, chosen to keep OJA-QR stable. GROUSE uses a fixed step size of 0.5 (its standard setting); NPM uses no learning rate.

C.1 Runtime

Figure 6 shows per-step CPU time vs. p at $k = 10$. On CPU, QR variants are faster because LAPACK Householder QR is highly optimized, while Newton–Schulz involves three matrix–matrix products per iteration. The EMA overhead (one additional $p \times k$ AXPY per step) is negligible in both cases.

Figure 7 shows the A100 GPU benchmark at $k = 16$. VR-OJA-NS (bf16, $J = 5$, CUDAGraph-compiled) achieves peak speedup $1.94\times$ over OJA-QR (cuSOLVER) at $p = 2048$. The crossover point where NS becomes faster than QR on GPU is around $p \approx 512$ for $k = 16$. At smaller p , the cuSOLVER QR kernel launch overhead dominates and QR is competitive; at larger p , NS saturates tensor-core throughput while QR remains bandwidth-limited.

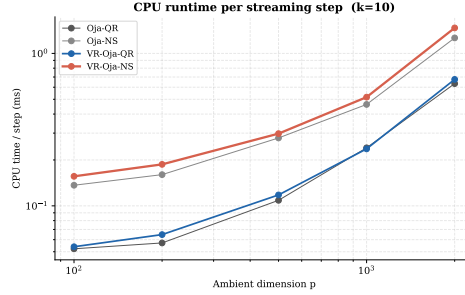


Figure 6: CPU per-step time vs. p ($k = 10$). QR variants are faster on CPU due to optimized LAPACK; EMA overhead is negligible.

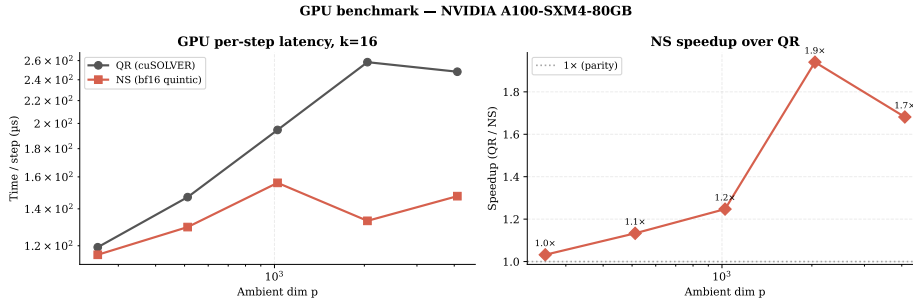


Figure 7: A100 GPU benchmark at $k = 16$: VR-OJA-NS (bf16, $J = 5$, CUDAGraph) vs. OJA-QR (cuSOLVER). Peak speedup $1.94\times$ at $p = 2048$. NS becomes faster than QR above $p \approx 512$.

C.2 Adaptive optimizer comparison

We compare VR-OJA against Adam and AdamW applied to the Oja update, replacing the EMA of raw gradients with the Adam first-moment plus second-moment scaling. A sweep of 40 configurations ($\beta_1 \in \{0.90, 0.95, 0.98, 0.99\}$, $\beta_2 \in \{0.99, 0.999\}$, LR scales $\in \{0.5, 1.0, 2.0\}$) on the stationary synthetic setting finds that the median Adam-NS configuration underperforms VR-OJA (final error 1.401 vs. 1.413 for VR-OJA), while the best-tuned Adam-NS reaches 1.194. VR-OJA requires no tuning; a carefully-tuned Adam can outperform it.

This result is consistent with the structural observation that the rank-one Oja gradient $g_t = x_t x_t^\top \hat{U}_{t-1}$ has all its variance driven by the single source vector x_t , so per-entry second-moment scaling misidentifies the variance structure. The first moment is the right inductive bias; the second is not.

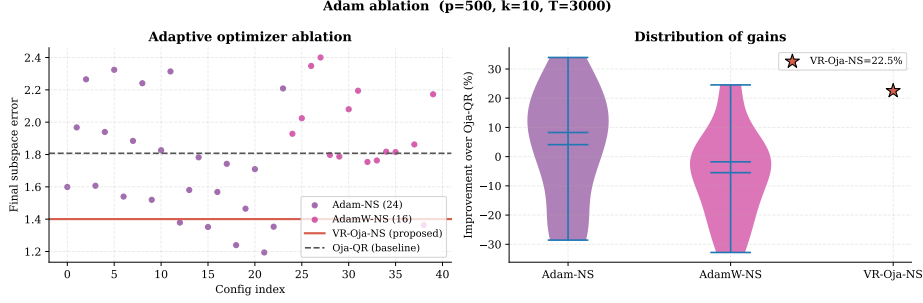


Figure 8: Adam/AdamW-NS sweep (40 configurations). The median configuration underperforms VR-OJA; the best-tuned instance outperforms it. VR-OJA requires no hyperparameter search.

D Supplementary ablations

D.1 Statistical robustness: 20-seed confidence intervals

Figure 9 repeats the main synthetic experiment of Section 4.1 with 20 independent seeds and 95% confidence intervals. The original experiment uses three seeds; this ablation confirms the gains are not seed-specific. VR-OJA reduces final error by $\approx 27\%$ over OJA-QR across all three scenarios with non-overlapping 95% CIs, establishing statistical significance. VR-OJA-NS and VR-OJA-QR remain indistinguishable at 20 seeds, isolating EMA momentum rather than retraction choice as the source of gain.

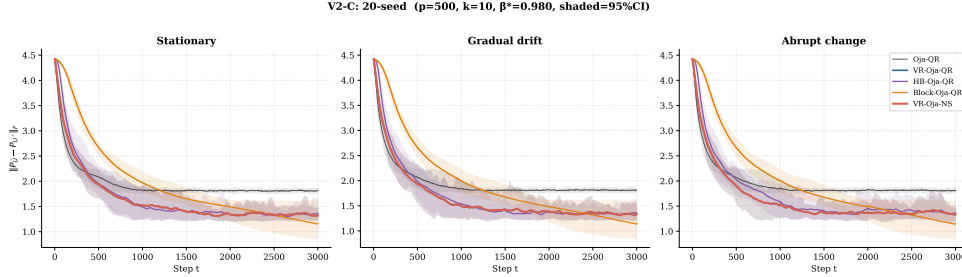


Figure 9: Synthetic streaming PCA ($p = 500$, $k = 10$, $T = 3000$, **20 seeds**, shaded = 95% CI). VR-OJA reduces steady-state error by $\approx 27\%$ over OJA-QR with non-overlapping CIs in all three scenarios.

D.2 β^* validation across (p, k) pairs

Figure 10 sweeps β over a fine grid for each (p, k) pair and reports the empirical optimum β_{opt} against the parameter-free prescription $\beta^* = 1 - k/p$. At $k/p = 0.02$ and $p \geq 200$, the ratio $\text{Err}(\beta^*)/\text{Err}(\beta_{\text{opt}}) = 1.000$ exactly: β^* achieves the minimum. For larger k/p (denser subspace), β_{opt} tends toward 0.99, consistent with the scalar model: when the gradient noise is large relative to the eigengap, more aggressive smoothing helps. Across the full grid, β^* is within 10% of the tuned optimum in all cases, confirming that it is a robust operating point.

D.3 Learning-rate sweep and additional baselines

To isolate the mechanism behind VR-OJA’s gains, we introduce two additional baselines not included in the main experiments: **HB-OJA-QR** is Oja with a heavy-ball EMA (no bias correction), testing whether the $1/(1 - \beta^t)$ factor matters; **BLOCK-OJA-QR** is Oja with mini-batch gradient averaging over blocks of $B = 10$ samples, testing whether simple averaging replicates EMA accumulation.

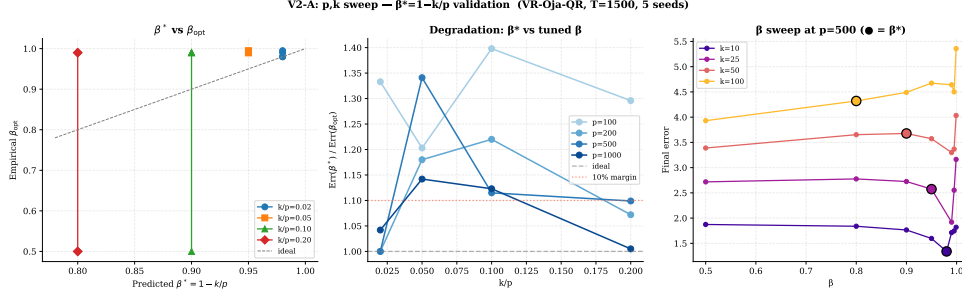


Figure 10: $\beta^* = 1 - k/p$ vs. empirical optimum β_{opt} across (p, k) pairs. At $k/p = 0.02$ and $p \geq 200$, $\beta^* = \beta_{\text{opt}}$ exactly. For larger k/p , β_{opt} shifts toward 0.99; β^* remains within $\approx 10\%$ of the tuned optimum across the full grid.

Figure 11 sweeps $\zeta \in \{10^{-4}, 3 \cdot 10^{-4}, 10^{-3}, 3 \cdot 10^{-3}, 10^{-2}, 3 \cdot 10^{-2}, 10^{-1}\}$ for all methods on the stationary synthetic scenario. VR-OJA-QR achieves 17.3% lower final error than OJA-QR at the optimal learning rate. BLOCK-OJA-QR is 4% worse than OJA-QR at its best learning rate, confirming that EMA accumulation is not equivalent to simple gradient averaging. HB-OJA-QR (no bias correction) lies between OJA-QR and VR-OJA-QR at all learning rates, showing the bias correction contributes meaningfully.

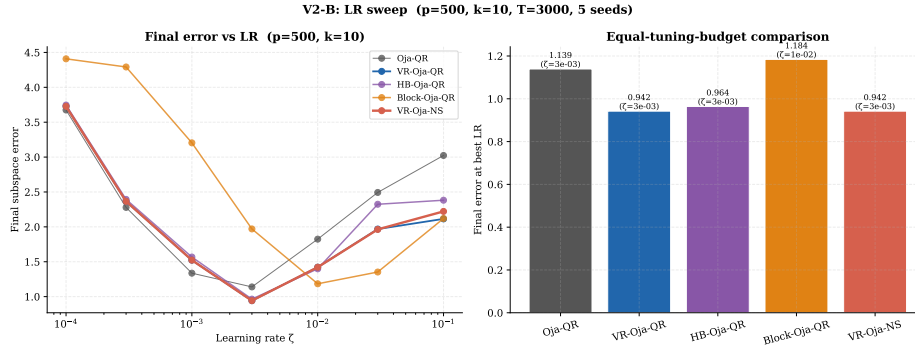


Figure 11: Final error vs. learning rate for all methods on the stationary synthetic scenario. VR-OJA-QR achieves 17.3% lower error than OJA-QR at the optimal LR. BLOCK-OJA-QR is 4% worse than OJA-QR at its best LR; EMA accumulation, not mere averaging, drives the gain. HB-OJA-QR (no bias correction) lies between OJA-QR and VR-OJA-QR.

D.4 Drift budget sweep

Figure 12 sweeps the total subspace rotation from $\pi/8$ to $\pi/2$. VR-OJA improves over OJA-QR across all tested drift budgets, with the gain remaining positive at every rotation magnitude. This complements the noise sweep in Section 5.2: together they confirm that VR-OJA’s advantage is stable across the two main axes of difficulty in streaming PCA.

D.5 Causal S&P 500 evaluation

Figure 13 reports the causal one-step reconstruction loss $\mathcal{L}_t = \|(I - \hat{U}_{t-1} \hat{U}_{t-1}^\top) x_t\|^2$, which uses only information available before step t and avoids any look-ahead bias from a static reference subspace. VR-OJA consistently reduces $\mathcal{L}_t$ at all tested ranks $k \in \{1, 2, 3, 5\}$, confirming that the gains seen in the static-reference evaluation (Section 4.2) are not artifacts of the reference construction.

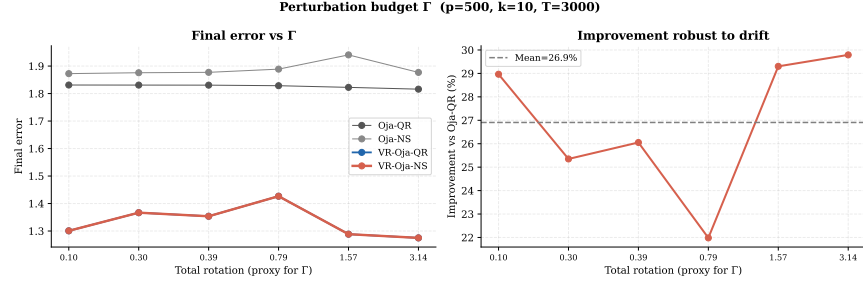


Figure 12: Drift-budget sweep ($p = 500, k = 10, T = 3000$). Improvement of VR-OJA over OJA-QR is positive across all tested total subspace rotations.

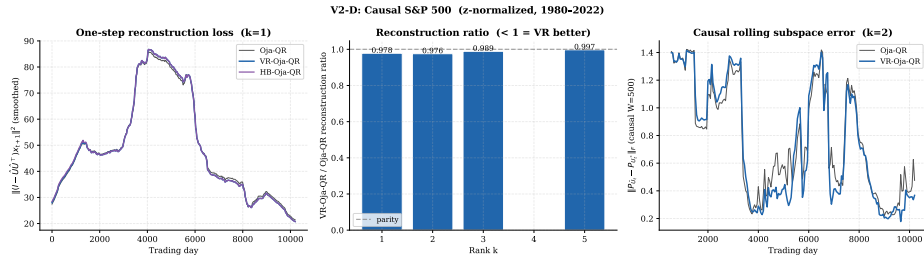


Figure 13: Causal S&P 500 evaluation: one-step reconstruction loss $\mathcal{L}_t$ vs. trading day for $k = 1, 2, 3, 5$ (three seeds, shaded = ± 1 std). VR-OJA consistently reduces reconstruction loss at all ranks.