High-Efficiency Geometric Adaptation (HEGA)

A Geometry Aware First Order Optimizer with Strong Nonconvex Performance

Omar Graia

Acton Boxborough Regional High School
Acton, Massachusetts, United States of America

Under the guidance of
Dr. Nabil Mesbah
Geneva, Switzerland

August 24, 2025

High-Efficiency Geometric Adaptation (HEGA): A Geometry Aware First Order Optimizer with Strong Nonconvex Performance

Omar Graia

Abstract

We present HEGA, a strictly first-order optimizer that augments an AMSGrad stabilized diagonal preconditioner with two lightweight geometry signals computed along the iterate gradient path. A clipped secant based curvature estimate provides a scalar path scale, while an exponentially smoothed gradient alignment score gates a convex mixture between the scalar and diagonal branches and modulates the step via α^{C_t}. All operations are vectorized, giving $\mathcal{O}(d)$ time and memory per step. We prove uniform bounds on the effective preconditioner, obtain $\mathcal{O}(\sqrt{T})$ regret in online convex optimization, and establish local linear convergence for smooth objectives with extensions under the Polyak–Łojasiewicz condition (deterministic and stochastic). Across 20 different test functions on dimensions ranging from 5 to $10,000$, HEGA achieves the best overall score compared with SGD, AdaGrad, RMSprop, Adam, AMSGrad, and NAdam.

Keywords: optimization; gradient descent; first order methods; adaptive preconditioning; curvature; Polyak–Łojasiewicz condition; online convex optimization.

Commitments on Academic Honesty and Integrity

We hereby declare that we

1. are fully committed to the principle of honesty, integrity and fair play throughout the competition.
2. actually perform the research work ourselves and thus truly understand the content of the work.
3. observe the common standard of academic integrity adopted by most journals and degree theses.
4. have declared all the assistance and contribution we have received from any personnel, agency, institution, etc. for the research work.
5. undertake to avoid getting in touch with assessment panel members in a way that may lead to direct or indirect conflict of interest.
6. undertake to avoid any interaction with assessment panel members that would undermine the neutrality of the panel member and fairness of the assessment process.
7. observe the safety regulations of the laboratory(ies) where we conduct the experiment(s), if applicable.
8. observe all rules and regulations of the competition.
9. agree that the decision of YHSA is final in all matters related to the competition.

We understand and agree that failure to honour the above commitments may lead to disqualification from the competition and/or removal of reward, if applicable; that any unethical deeds, if found, will be disclosed to the school principal of team member(s) and relevant parties if deemed necessary; and that the decision of YHSA is final and no appeal will be accepted.

(Signatures of full team below)

X Omar Graia **X Dr. Nabil Mesbah**

Date: August 24, 2025

Contents

1 Introduction

Modern machine learning and scientific computing rely heavily on first order methods. Simple methods such as stochastic gradient descent with momentum are fast and memory light, however they can stall on harsh landscapes. Fully adaptive methods that scale coordinates by running gradient variance estimates are robust to coordinate imbalance, but they may react sluggishly when local curvature changes sharply. Richer preconditioners based on matrix approximations can improve conditioning, although their overhead is often prohibitive in high dimensions. This gap between robustness and efficiency motivates an optimizer that remains strictly first order, keeps the computational budget at $\mathcal{O}(d)$, and still extracts meaningful geometric cues from the optimization path.

We present High-Efficiency Geometric Adaptation (HEGA), a geometry aware first order method that augments stabilized coordinate wise adaptivity with two inexpensive path level statistics. The first is a clipped secant curvature scale built from $s_t = x_t - x_{t-1}$ and $y_t = g_t - g_{t-1}$,

$$H_t \;=\; \frac{\langle s_t,\, y_t\rangle}{\|s_t\|^2 + \delta},$$

with a small damping $\delta > 0$. The second is a running alignment coefficient C_t that measures cosine similarity between successive gradients and serves as a reliability signal for the current search direction. These signals are combined with an AMSGrad stabilized variance accumulator to form an effective preconditioner that interpolates, at each step, between a diagonal L_2 type metric and a scalar path curvature metric; the interpolation weight depends smoothly on $|C_t|$. In addition, a correlation driven modulation multiplies the step size by α^{C_t}, strengthening steps when recent gradients agree and attenuating them when they disagree. All operations are vectorized, with only two dot products per iteration, preserving $\mathcal{O}(d)$ time and memory.

On the theory side, we show that HEGA inherits the stability of variance based adaptivity while benefiting from curvature sensing. The AMSGrad anchor and the damped, exponentially smoothed geometry signals yield uniform lower/upper bounds $P_{\min} \le P_{\mathrm{eff},t,i} \le P_{\max}$ on the effective preconditioner, which give a clean descent inequality. In online convex optimization we obtain $\mathcal{O}(\sqrt{T})$ regret without any mixing assumption by staying entirely in the AMSGrad anchor metric, introducing a ghost projection step, and handling the alignment modulated step via monotone envelopes $\underline{s}_t \le s_t \le \overline{s}_t$ in the weighted telescoping argument. For stationary objectives we give explicit small step conditions under which the EMA tracks the gradient and HEGA enjoys local linear convergence under strong convexity. Under the Polyak–Łojasiewicz condition we obtain linear decay deterministically and linear convergence in expectation to a variance controlled neighborhood in the stochastic case. Formal statements appear in Section 5, with complete proofs in the appendix.

Empirically, we evaluate HEGA on twenty classical smooth test functions with dimensions ranging from 5 to 1000. The protocol uses float64 precision, JIT warm ups outside timing windows, function specific domains for initialization, and a simple dimension rule that scales a single $d = 5$ tuned base step size: square root scaling for SGD like methods and quarter power scaling for adaptive methods, including HEGA. We report normalized time and accuracy scores, Dolan Moré performance profiles, distributional views pooled across tasks, scaling curves with respect to dimension, and pairwise dominance rates. Across this suite, HEGA attains the best aggregate score against SGD, AdaGrad, RMSprop, Adam, AMSGrad, and NAdam, with the clearest margins in high dimensions and low τ regions of the performance profiles.

In summary, we propose a strictly first order optimizer that blends a stabilized elementwise variance preconditioner with a scalar path curvature preconditioner, gated by gradient alignment and coupled with correlation driven step size modulation, all in $\mathcal{O}(d)$ time and memory. We also provide

theoretical guarantees, boundedness of the effective preconditioner, $\mathcal{O}(\sqrt{T})$ regret in the online convex setting, local linear convergence for smooth objectives, and PL results in deterministic and stochastic regimes. We design reproducible benchmarks over twenty functions and eight dimensions that show HEGA's strength.

2 Related Work

Current research on first order optimization varies a lot in how directions are chosen and also on how step magnitudes are scaled across coordinates. Momentum methods are the most widely used baselines, though. There is a lot of work developing coordinate wise adaptivity and its stabilized variations. Another line of work incorporates curvature via path statistics or quasi Newton ideas, and deep learning has motivated structured preconditioners that are able to capture geometry at the layer or block level. We continue the trend of evaluation practices that compared solvers fairly with normalized, aggregate views.

The base of modern optimization is stochastic gradient descent [19]. Early efforts to speed up this baseline introduced momentum, which dampens oscillations and accelerates progress by incorporating an exponential moving average of past updates [16, 17]. Nesterov's accelerated gradient achieves faster convex rates by strategically shifting the point of gradient evaluation [15]. Despite their success, these methods share the limitation of applying the same learning rate to every parameter. This strategy struggles badly in steep, narrow valleys landscapes where curvature is directionally varying or coordinates differ widely in scale.

A major class of optimizers scale updates by adapting to the magnitude of gradients, often on a per-coordinate basis. AdaGrad achieves this by accumulating squared gradients and dividing by their square root [6]. RMSprop refines this by using an exponentially decaying average of squared gradients instead of a cumulative sum [21]. Adam builds upon these ideas by incorporating both first and second moments of the gradients, along with bias correction for improved early stage performance [8]. Subsequent research has introduced numerous variants to enhance stability and generalization. For example, AMSGrad stabilizes Adam with a non-decreasing second moment buffer [18], NAdam integrates Nesterov acceleration [5], AdamW decouples weight decay [11], and others like Yogi [23], AdaBelief [25], RAdam [10], memory-efficient AdaFactor [20], and sign-based methods [3]. While these adaptive techniques are computationally efficient with them requiring only $\mathcal{O}(d)$ memory, relying solely on variance estimates can make them slow to react to abrupt changes in curvature or periods of directional unreliability.

A substantial literature analyzes the stability of coordinate wise preconditioners, especially beyond convexity. A common theme is to bound the inverse square root of the second moment accumulator, either via nondecreasing buffers or explicit lower bounds, yielding descent type guarantees under smoothness [4, 18]. HEGA follows this line by retaining a max buffer on the variance estimate and by damping auxiliary curvature statistics, which together deliver bounded effective scaling in the analysis. Our proof technique is closest in spirit to mirror descent telescoping arguments, but it is carried out entirely in the (monotone) AMSGrad anchor metric with a ghost projection step, which avoids any mixing assumption even when the alignment-modulated step is nonmonotone.

Curvature can be estimated from the optimization path without forming matrices. The Barzilai Borwein step uses the secant pair to fit a scalar curvature model [2]. Diagonal quasi Newton updates propagate coordinate wise second order information at low cost [14]. Lookahead methods introduce slow/fast paths to stabilize exploration [24]. In deep models, natural gradient and Kronecker factored approximations adapt to local metrics [1, 13], and Shampoo forms factored second moment matrices

to build richer preconditioners with tractable overhead [7]. HEGA borrows the scalar secant idea and blends it with variance based adaptivity. A smooth gate driven by gradient alignment controls this blend so the update remains vectorized and strictly first order.

Directional agreement and correlation have long been used to detect stalling, schedule learning rates, or modulate momentum [9, 12, 22]. In HEGA, a cosine similarity alignment coefficient is used as a reliability signal since it gates the interpolation between a diagonal variance preconditioner and a scalar path curvature preconditioner, and it modulates the global step multiplicatively. Together, these elements boost progress on smooth, coherent parts of the optimization landscape, but slows down when oscillations arise due to sharp curvature or noise.

While diagonal scaling is simple, more structured preconditioners can capture increased geometric information at a controlled computational cost. Methods like KFAC for instance employ Kronecker factorization to condense second-order information into layer wise factors [13]. Shampoo takes a similar approach by tracking matrix roots of per layer statistics [7]. These techniques are highly effective for large neural networks where model structure is known and exploitable. However, they can become cumbersome in black box optimization scenarios or when strict $\mathcal{O}(d)$ memory limits and single pass updates are important. HEGA is specifically designed for this latter use case with its reliance on only dot products and elementwise operations makes it indifferent to model structure.

In summary, HEGA uses both coordinate wise adaptivity and path informed curvature sensing. It still retains the stabilized adaptive engine of AMSGrad but adjusts it with a damped secant scalar and uses an alignment signal to gate the mixture and to modulate the step size. The result is an optimizer that retains the $\mathcal{O}(d)$ efficiency but offers significantly greater responsiveness to local geometric features. Our empirical evaluation specifically targets deterministic, smooth functions that challenge conditioning and multimodality. Crucially, HEGA is not designed to take the place of matrix based preconditioners where structure is readily available. Instead, it offers a more robust, geometry aware alternative optimized for strictly first order settings.

3 Preliminaries and Notation

This section sets notation, describes the two settings that we analyze, and defines the quantities used by the optimizer. The first setting is stationary optimization of a single differentiable function $f : \mathbb{R}^d \to \mathbb{R}$. The second setting is online convex optimization with a sequence $\{f_t\}_{t=1}^T$ over a closed convex domain $\mathcal{X} \subset \mathbb{R}^d$. In both settings the method generates iterates $\{\mathbf{x}_t\} \subset \mathcal{X}$ and uses Euclidean projection $\mathrm{Proj}_{\mathcal{X}}(\cdot)$ when constraints are present.

Vectors are bold lower case and matrices are bold upper case. The inner product is $\langle \mathbf{a}, \mathbf{b} \rangle$ and the norms are $\|\cdot\|_2$ and $\|\cdot\|_\infty$. The Hadamard product is $\odot$. For a vector $\mathbf{w}$, the diagonal matrix with $\mathbf{w}$ on its diagonal is $\mathrm{diag}(\mathbf{w})$. For stochastic analysis, $\{\mathcal{F}_t\}$ is the natural filtration generated by past iterates and gradients, and $\mathbb{E}_t[\cdot] = \mathbb{E}[\cdot \mid \mathcal{F}_{t-1}]$.

For any diagonal positive definite matrix A, the weighted norm is $\|x\|_A^2 := x^\top A x$. The weighted projection onto $\mathcal{X}$ is

$$\mathrm{Proj}_{\mathcal{X}}^{(A)}(z) := \arg\min_{x \in \mathcal{X}} \tfrac{1}{2}\|x - z\|_A^2,$$

which agrees with the Euclidean projection when $A = I$. For diagonal symmetric matrices A and B we write $A \preceq B$ when $x^\top A x \le x^\top B x$ holds for all $x \in \mathbb{R}^d$.

3.1 Problem Setting and Assumptions

We work with standard assumptions that are referenced by label where needed.

Assumption 1 (Convex Domain)**.** In the online setting each f_t is convex on a nonempty, closed, convex set $\mathcal{X} \subset \mathbb{R}^d$. The set has finite diameter $D > 0$ in the Euclidean norm, so $\|\mathbf{x} - \mathbf{y}\|_2 \le D$ for all $\mathbf{x}, \mathbf{y} \in \mathcal{X}$.

Assumption 2 (Stochastic Gradients)**.** At time t, the method receives a stochastic gradient $\tilde{\mathbf{g}}_t$ at $\mathbf{x}_{t-1}$ with $\mathbb{E}_t[\tilde{\mathbf{g}}_t] = \nabla f_t(\mathbf{x}_{t-1})$. Gradients are uniformly bounded coordinate-wise almost surely: there exists $G_\infty > 0$ with $\|\tilde{\mathbf{g}}_t\|_\infty \le G_\infty$. It follows that $\|\tilde{\mathbf{g}}_t\|_2 \le \sqrt{d}\, G_\infty$.

We reserve $\nabla f_t(\mathbf{x}_{t-1})$ for the true gradient and use $\tilde{\mathbf{g}}_t$ for the stochastic gradient used by the algorithm. All EMAs $(\mathbf{m}_t, \mathbf{v}_t, \hat{\mathbf{v}}_t)$ below are formed from $\tilde{\mathbf{g}}_t$.

Assumption 3 (Lipschitz Smoothness)**.** When specified, an objective f is L smooth. This means $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \le L\|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{x}, \mathbf{y}$.

Assumption 4 (Strong Convexity)**.** For local rate analysis we assume that f is μ strongly convex on a neighborhood of a minimizer $\mathbf{x}^*$. That is, $f(\mathbf{y}) \ge f(\mathbf{x}) + \langle \nabla f(\mathbf{x}), \mathbf{y} - \mathbf{x}\rangle + \frac{\mu}{2}\|\mathbf{y} - \mathbf{x}\|_2^2$ for some $\mu > 0$.

Assumption 5 (Polyak–Łojasiewicz condition)**.** For nonconvex analysis we adopt the PL condition on a sublevel set $\mathcal{S}_C = \{x : f(x) \le C\}$. There exists $\mu_{\mathrm{PL}} > 0$ such that $\frac{1}{2}\|\nabla f(\mathbf{x})\|_2^2 \ge \mu_{\mathrm{PL}}\big(f(\mathbf{x}) - f^*\big)$ for all $\mathbf{x} \in \mathcal{S}_C$.

Assumption 6 (Inter round Gradient Drift)**.** In the online setting the sequence $\{f_t\}$ may change between rounds. There is $\Delta \ge 0$ with $\|\nabla f_t(\mathbf{x}) - \nabla f_{t-1}(\mathbf{x})\|_2 \le \Delta$ for all $\mathbf{x} \in \mathcal{X}$. For stationary problems $\Delta = 0$.

In the online setting performance is measured by regret

$$R(T) = \sum_{t=1}^{T} f_t(\mathbf{x}_{t-1}) - \inf_{\mathbf{x}^* \in \mathcal{X}} \sum_{t=1}^{T} f_t(\mathbf{x}^*).$$

3.2 Algorithmic Quantities

The method maintains exponential moving averages. The first and second moments use decays $\beta_1, \beta_2 \in [0, 1)$ and are built from the stochastic gradients $\tilde{\mathbf{g}}_t$,

$$\mathbf{m}_t = \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\tilde{\mathbf{g}}_t, \qquad \mathbf{v}_t = \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)(\tilde{\mathbf{g}}_t \odot \tilde{\mathbf{g}}_t),$$

and the AMSGrad buffer $\hat{\mathbf{v}}_t = \max(\hat{\mathbf{v}}_{t-1}, \mathbf{v}_t)$ is taken coordinate-wise. The bias-corrected momentum is $\hat{\mathbf{m}}_t = \mathbf{m}_t/(1 - \beta_1^t)$.

Path alignment and curvature are tracked with decay $\gamma \in [0, 1)$. The raw alignment c_t and the curvature estimate are

$$c_t = \frac{\langle \tilde{\mathbf{g}}_t, \tilde{\mathbf{g}}_{t-1}\rangle}{\|\tilde{\mathbf{g}}_t\|_2 \, \|\tilde{\mathbf{g}}_{t-1}\|_2 + \epsilon_c}, \qquad H_{\mathrm{est},t} = \frac{\langle \mathbf{y}_t, \mathbf{s}_t\rangle}{\|\mathbf{s}_t\|_2^2 + \epsilon_q},$$

where $\mathbf{s}_t = \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$ and $\mathbf{y}_t = \tilde{\mathbf{g}}_t - \tilde{\mathbf{g}}_{t-1}$. The smoothed states are

$$C_t = \gamma C_{t-1} + (1 - \gamma)c_t, \qquad V_{\mathrm{path},t} = \gamma V_{\mathrm{path},t-1} + (1 - \gamma)\max\{H_{\mathrm{est},t}, \delta_0\},$$

with $\delta_0 > 0$ a curvature floor and $\epsilon_c, \epsilon_q > 0$ stabilizers.

These signals combine into the effective diagonal preconditioner $\mathbf{P}_{\text{eff},t}$. It interpolates between the diagonal AMSGrad branch and a scalar path-based branch,

$$\mathbf{P}_{\text{eff},t} = \lambda_t \left(\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1}\right)^{-1} + (1-\lambda_t)\left(\sqrt{V_{\text{path},t}} + \epsilon_p\right)^{-1}\mathbf{1}, \tag{1}$$

where $\lambda_t = 1 - |C_t|^{\lambda_p}$ with $\lambda_p > 0$ and $\epsilon_p > 0$ is a stabilizer. The base learning rate η_t is modulated by the alignment score through $M_t = \alpha^{C_t}$ with $\alpha \geq 1$. The update is

$$\mathbf{x}_t = \text{Proj}_{\mathcal{X}}\left(\mathbf{x}_{t-1} - \eta_t\, M_t\, (\mathbf{P}_{\text{eff},t} \odot \hat{\mathbf{m}}_t)\right). \tag{2}$$

For regret we use $\eta_t = \eta/\sqrt{t}$. For local analysis we use a constant $\eta_t \equiv \eta$.

Unless stated otherwise the initialization is $\mathbf{x}_{-1} = \mathbf{x}_0 \in \mathcal{X}$, $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{v}_0 = \hat{\mathbf{v}}_0 = \mathbf{0}$, $\tilde{\mathbf{g}}_0 = \mathbf{0}$, $C_0 = 0$, and $V_{\text{path},0} = \delta_0$. This yields $C_1 = 0$ and $V_{\text{path},1} = \delta_0$.

For analysis we set $\mathbf{D}_t := \text{diag}(\mathbf{P}_{\text{eff},t})$ and $\mathbf{A}_t := \text{diag}(1/\mathbf{P}_{\text{eff},t}) = \mathbf{D}_t^{-1}$. We also use the AMSGrad anchor metric $\bar{\mathbf{A}}_t := \text{diag}(\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1})$, which is coordinate wise nondecreasing in t. When a weighted projection is used we write $\text{Proj}_{\mathcal{X}}^{(\bar{\mathbf{A}}_t)}$.

4 The HEGA Optimizer

This section gives the full formulation of the HEGA optimizer and explains its main components. As set out in Section 3, HEGA augments an AMSGrad style update engine with two geometry signals, the path alignment C_t and the path curvature $V_{\text{path},t}$. These signals adapt the preconditioner and the step size and every iteration costs $O(d)$.

The core mechanism is the alignment gated preconditioner in (1). It blends a coordinate wise AMSGrad preconditioner with a scalar term computed from path curvature. The mixing weight $\lambda_t = 1 - |C_t|^{\lambda_p}$ directs the method toward the scalar path preconditioner when consecutive gradients are well aligned ($|C_t| \to 1$), which signals a coherent search direction. When alignment weakens, the update falls back to the diagonal preconditioner, which helps when there is noise.

The same alignment score also scales the step size through $M_t = \alpha^{C_t}$ in (2). Agreement among gradients leads to larger steps, while disagreement shrinks them in local geometry. The AMSGrad buffer $\hat{\mathbf{v}}_t$ keeps the diagonal preconditioner stable, and the curvature floor δ_0 plus the stabilizer ϵ_q prevent the path based term from becoming ill defined. These pieces make the optimizer sensitive to trajectory geometry while preserving the stability of modern adaptive methods.

For experiments, the base step η at dimension d is obtained from tuned values at $d = 5$ using a fixed scaling rule. For SGD and AdaGrad we set $\eta(d) = \eta_5\sqrt{5/d}$. For methods in the Adam family, including HEGA, we use a quarter power rule $\eta(d) = \eta_5(5/d)^{1/4}$. This policy gives a fair comparison across scales.

The following pseudocode implements HEGA.

Algorithm 1 High-Efficiency Geometric Adaptation (HEGA)

Inputs: base schedule η_t, modulator base $\alpha \geq 1$, EMA decays $\beta_1, \beta_2, \gamma \in [0, 1)$, interpolation power $\lambda_p > 0$, stabilizers $\epsilon_c, \epsilon_q, \epsilon_p > 0$, curvature floor $\delta_0 > 0$, domain $\mathcal{X} \subset \mathbb{R}^d$.

Init: $\mathbf{x}_{-1} = \mathbf{x}_0 \in \mathcal{X}$, $\mathbf{g}_0 = \mathbf{0}$, $\mathbf{m}_0 = \mathbf{0}$, $\mathbf{v}_0 = \hat{\mathbf{v}}_0 = \mathbf{0}$, $C_0 = 0$, $V_{\text{path},0} = \delta_0$.

for $t = 1, 2, \ldots$ **do**

$\tilde{\mathbf{g}}_t \leftarrow \nabla f_t(\mathbf{x}_{t-1})$

$\mathbf{m}_t \leftarrow \beta_1 \mathbf{m}_{t-1} + (1 - \beta_1)\tilde{\mathbf{g}}_t$; $\quad \hat{\mathbf{m}}_t \leftarrow \mathbf{m}_t / (1 - \beta_1^t)$

$\mathbf{v}_t \leftarrow \beta_2 \mathbf{v}_{t-1} + (1 - \beta_2)(\tilde{\mathbf{g}}_t \odot \tilde{\mathbf{g}}_t)$; $\quad \hat{\mathbf{v}}_t \leftarrow \max(\hat{\mathbf{v}}_{t-1}, \mathbf{v}_t)$

$c_t \leftarrow \frac{\langle \tilde{\mathbf{g}}_t, \mathbf{g}_{t-1} \rangle}{\|\tilde{\mathbf{g}}_t\|_2 \|\mathbf{g}_{t-1}\|_2 + \epsilon_c}$; $\quad C_t \leftarrow \gamma C_{t-1} + (1 - \gamma) c_t$

$\mathbf{s}_t \leftarrow \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$; $\quad \mathbf{y}_t \leftarrow \tilde{\mathbf{g}}_t - \mathbf{g}_{t-1}$

$H_{\text{est},t} \leftarrow \frac{\langle \mathbf{y}_t, \mathbf{s}_t \rangle}{\|\mathbf{s}_t\|_2^2 + \epsilon_q}$; $\quad V_{\text{path},t} \leftarrow \gamma V_{\text{path},t-1} + (1 - \gamma) \max\{H_{\text{est},t}, \delta_0\}$

$\lambda_t \leftarrow 1 - |C_t|^{\lambda_p}$

$\mathbf{P}_{\text{L2},t} \leftarrow \left(\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1}\right)^{-1}$; $\quad P_{\text{path},t} \leftarrow \left(\sqrt{V_{\text{path},t}} + \epsilon_p\right)^{-1}$

$\mathbf{P}_{\text{eff},t} \leftarrow \lambda_t \mathbf{P}_{\text{L2},t} + (1 - \lambda_t) P_{\text{path},t} \mathbf{1}$

$s_t \leftarrow \eta_t \cdot \alpha^{C_t}$

$\mathbf{u}_t \leftarrow s_t \left(\mathbf{P}_{\text{eff},t} \odot \hat{\mathbf{m}}_t\right)$

$\mathbf{x}_t \leftarrow \text{Proj}_{\mathcal{X}}(\mathbf{x}_{t-1} - \mathbf{u}_t)$

5 Theoretical Results

This section states the guarantees for HEGA. We first bound the geometry signals and the effective preconditioner. We then give an $O(\sqrt{T})$ regret bound in online convex optimization without any mixing assumption by working entirely in the AMSGrad anchor metric through a ghost iterate. Finally, we give local linear convergence results for stationary objectives under strong convexity and under the Polyak–Łojasiewicz condition. All proofs are deferred to the appendix.

5.1 Stability and Boundedness of Geometric Components

The adaptive pieces of HEGA remain controlled along the run. The statements below collect the uniform bounds that are used later.

Proposition 7 (Bounds on geometric modulators)**.** *For all $t \geq 1$, the smoothed alignment satisfies $C_t \in [-1, 1]$. With $\alpha \geq 1$ and $\lambda_p > 0$ it follows that $M_t = \alpha^{C_t} \in [\alpha^{-1}, \alpha]$ and $\lambda_t = 1 - |C_t|^{\lambda_p} \in [0, 1]$.*

Proposition 8 (Boundedness of path curvature)**.** *Under Assumptions 3 and 6, the path curvature estimate satisfies $V_{\text{path},t} \in [\delta_0, V_{\max}]$ for all t, where $V_{\max} = \max\{\delta_0, L + \Delta/(2\sqrt{\epsilon_q})\}$. In the stationary case with $\Delta = 0$ this simplifies to $V_{\max} = \max\{\delta_0, L\}$.*

These component wise bounds imply that the effective preconditioner used in the update step (2) stays uniformly between fixed positive constants.

Proposition 9 (Uniform bounds for the effective preconditioner)**.** *Under Assumption 2 with $\|\tilde{\mathbf{g}}_t\|_\infty \leq G_\infty$, there exist constants $P_{\min}$ and $P_{\max}$ such that for all t, i,*

$$0 < P_{\min} \leq P_{\text{eff},t,i} \leq P_{\max} < \infty,$$

where $P_{\min} = \min\left\{(G_\infty + \epsilon_p)^{-1}, (\sqrt{V_{\max}} + \epsilon_p)^{-1}\right\}$ and $P_{\max} = \max\left\{\epsilon_p^{-1}, (\sqrt{\delta_0} + \epsilon_p)^{-1}\right\}$.

We compare the effective preconditioner to the AMSGrad branch since this sandwich is used throughout the analysis. The anchor metric $\bar{\mathbf{A}}_t = \mathrm{diag}(\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1})$ is coordinate wise nondecreasing in t because of the max buffer.

Lemma 10 (Comparability of preconditioners)**.** *Let* $\mathbf{P}_{\mathrm{L2},t} = (\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1})^{-1}$ *and let* $\mathbf{P}_{\mathrm{eff},t}$ *be given by* (1)*. With* $V_{\max}$ *from Proposition 8 and* $\delta_0 > 0$*, for all* t, i*,*

$$c_{\min} P_{\mathrm{L2},t,i} \;\le\; P_{\mathrm{eff},t,i} \;\le\; c_{\max} P_{\mathrm{L2},t,i},$$

where $c_{\min} = \dfrac{\epsilon_p}{\sqrt{V_{\max}} + \epsilon_p}$ *and* $c_{\max} = \dfrac{G_\infty + \epsilon_p}{\sqrt{\delta_0} + \epsilon_p}$*. Equivalently, in matrix order,*

$$c_{\min} \bar{\mathbf{A}}_t^{-1} \;\preceq\; \mathbf{D}_t \;\preceq\; c_{\max} \bar{\mathbf{A}}_t^{-1}.$$

In particular, $c_{\min} \in (0, 1]$ *and* $c_{\max} \ge 1$*.*

5.2 Regret Bound for Online Convex Optimization

We use $\eta_t = \eta/\sqrt{t}$ and set $s_t := \eta_t \, \alpha^{C_t}$. Since s_t need not be monotone in t, we work with the deterministic envelopes

$$\underline{s}_t := \frac{\eta}{\alpha \sqrt{t}} \;\le\; s_t \;\le\; \overline{s}_t := \frac{\eta \, \alpha}{\sqrt{t}},$$

and carry out the proof entirely in the AMSGrad anchor metric $\bar{\mathbf{A}}_t$ with a ghost projection step.

Lemma 11 (Mirror one step bound in the anchor metric)**.** *Let* $\tilde{\mathbf{x}}_t = \mathrm{Proj}_{\mathcal{X}}^{(\bar{\mathbf{A}}_t)}\left(\mathbf{x}_{t-1} - s_t \, \bar{\mathbf{A}}_t^{-1} \hat{\mathbf{m}}_t\right)$*. Then*

$$\left\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^* \right\rangle \;\le\; \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2 - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2}{2 s_t} \;+\; \frac{s_t}{2} \|\hat{\mathbf{m}}_t\|_{\bar{\mathbf{A}}_t^{-1}}^2.$$

Since $s_t = \eta \, \alpha^{C_t}/\sqrt{t}$ need not be monotone in t, we use the bounds

$$\underline{s}_t := \frac{\eta}{\alpha \sqrt{t}} \;\le\; s_t \;\le\; \overline{s}_t := \frac{\eta \, \alpha}{\sqrt{t}}.$$

Applying Lemma 21 and then replacing $1/s_t$ by $1/\underline{s}_t$ and s_t by $\overline{s}_t$ yields

$$\left\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^* \right\rangle \le \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2 - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2}{2 \, \underline{s}_t} + \frac{\overline{s}_t}{2} \|\hat{\mathbf{m}}_t\|_{\bar{\mathbf{A}}_t^{-1}}^2.$$

We then telescope with the nondecreasing weights $a_t := 1/\underline{s}_t = (\alpha/\eta)\sqrt{t}$, and bound all quadratic terms using $\overline{s}_t \le (\eta\alpha)/\sqrt{t}$.

We also write $\nabla f_t(\mathbf{x}_{t-1}) = \tilde{\mathbf{g}}_t - \boldsymbol{\xi}_t$ with $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_{t-1}] = 0$, noting that $\hat{\mathbf{m}}_t$ is built from $\tilde{\mathbf{g}}_t$.

The lemma provides the pairing with $\hat{\mathbf{m}}_t$ and a telescoping term entirely in $\bar{\mathbf{A}}_t$. The quadratic term can be compared to the update quadratic through Lemma 10. The next theorem states the regret result.

Theorem 12 (Regret bound for HEGA)**.** *Let Assumptions 1, 2, and 6 hold. With the update* (2) *and* $\eta_t = \eta/\sqrt{t}$*, the regret satisfies*

$$R(T) = O(\sqrt{T}).$$

A standard extension shows that if each f_t is μ strongly convex then with $\eta_t = \eta/t$ the regret is $O(\log T)$.

5.3 Convergence for Stationary Objectives

We consider a fixed objective f and a constant step size $\eta_t \equiv \eta$. The analysis explains how the method locally behaves like a scalar preconditioned gradient step when gradients align.

Lemma 13 (Momentum tracking for the EMA)**.** *Assume f is L smooth. For $\hat{\mathbf{m}}_t = \frac{1}{1-\beta_1^t}\sum_{k=1}^{t}(1-\beta_1)\beta_1^{t-k}\nabla f(\mathbf{x}_{k-1})$ one has*

$$\|\hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_{t-1})\|_2 \;\le\; \frac{\beta_1}{1-\beta_1} L \sum_{j=1}^{\infty} \beta_1^{j-1} \|\mathbf{x}_{t-j} - \mathbf{x}_{t-j-1}\|_2.$$

In particular, if the preconditioner is uniformly bounded and $\eta > 0$ is chosen so that the step lengths are bounded by a small constant, then there is $\rho \in (0,1)$ with $\|\hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_{t-1})\|_2 \le \rho \|\nabla f(\mathbf{x}_{t-1})\|_2$ for all large t.

Assumption 14 (Local regularity at the solution)**.** Either $\mathbf{x}^* \in \operatorname{int}(\mathcal{X})$ so projection is inactive in a neighborhood of $\mathbf{x}^*$, or $\mathbf{x}^* \in \partial\mathcal{X}$ and the tangent-cone condition holds so the projection is nonexpansive along the iterates in that neighborhood.

Theorem 15 (Local linear rate)**.** *Under Assumption 14, assume f is L smooth and μ strongly convex. In a directional convergence regime near a minimizer $\mathbf{x}^*$ where $|C_t| \to 1$, there exists $\eta > 0$ such that for all sufficiently large t,*

$$\|\mathbf{x}_t - \mathbf{x}^*\|_2 \;\le\; \rho \|\mathbf{x}_{t-1} - \mathbf{x}^*\|_2$$

holds with some $\rho \in (0,1)$.

It suffices to choose $\eta \in \big(0, \bar{\eta}\big)$ with

$$\bar{\eta} \;\le\; \frac{1}{\alpha} \cdot \min\left\{ \frac{P_{\min}\mu}{4P_{\max}^2 L^2},\; \frac{1}{8C_\delta P_{\max}} \right\},$$

where $C_\delta = \frac{\beta_1}{1-\beta_1}L$ is the constant from Lemma 13. This ensures (i) $2sP_{\min}\mu - 2s^2P_{\max}^2L^2 \ge sP_{\min}\mu/2$ and (ii) the EMA tracking factor absorbed by Young's inequality.

Theorem 16 (Linear convergence under the PL condition)**.** *Assume f is L smooth and satisfies Assumption 5 on a sublevel set that contains the iterates. There exists a constant step $\eta > 0$, depending on L, α, and the bounds $P_{\min}, P_{\max}$, such that*

$$f(\mathbf{x}_t) - f^* \;\le\; (1-\kappa)\big(f(\mathbf{x}_{t-1}) - f^*\big)$$

with $\kappa = \eta\,\mu_{\mathrm{PL}}\,\alpha^{-1}P_{\min} \in (0,1)$.

Any constant $\eta \in \big(0, \frac{P_{\min}}{4\alpha L P_{\max}^2}\big)$ yields a contraction factor $\kappa = \eta\,\mu_{\mathrm{PL}}\,\alpha^{-1}P_{\min} \in (0,1)$.

Corollary 17 (Stochastic PL convergence)**.** *Under the conditions of Theorem 16 with stochastic gradients as in Assumption 2, the method converges linearly in expectation to a neighborhood of the minimum,*

$$\mathbb{E}\big[f(\mathbf{x}_t) - f^*\big] \;\le\; (1-\kappa)\,\mathbb{E}\big[f(\mathbf{x}_{t-1}) - f^*\big] \;+\; O(\eta^2\sigma^2),$$

where σ^2 is the gradient noise variance. With a diminishing step $\eta_t = O(1/t)$ one recovers $\mathbb{E}[f(\mathbf{x}_t) - f^*] = O(1/t)$.

These results show that HEGA is stable and convergent. The method matches the usual worst case rates and gives faster self tuning behavior when the geometry along the path is informative.

6 Implementation Details

This section records the experimental setup to ensure the results are reproducible. All experiments were implemented in Python using the JAX library, which provides automatic differentiation for gradients and just in time (JIT) compilation for performance. To maintain high numerical precision, all computations were performed using 64 bit floating point arithmetic. All benchmark runs are deterministic. We thus evaluate exact gradients and set $\tilde{\mathbf{g}}_t = \nabla f(\mathbf{x}_{t-1})$ in the implementation, consistent with the theoretical notation.

The benchmark workload consists of twenty classical smooth test functions, selected to represent a diverse set of optimization challenges. The evaluation was conducted in two stages. The main benchmark covered dimensions $d \in \{5, 10, 25, 50, 100, 250, 500, 1000\}$, with each optimizer performing 20 independent runs per problem. A high dimensional stress test was performed on dimensions $d \in \{2000, 5000, 10000\}$, with 5 independent runs per problem. For each run, the starting point was sampled uniformly from the function's conventional domain using a fixed random seed for reproducibility.

We compare HEGA against SGD, AdaGrad, RMSprop, Adam, AMSGrad, and NAdam. To ensure a fair comparison, a consistent hyperparameter tuning policy was adopted. All optimizers' hyperparameters were first tuned on the $d = 5$ problems. These base learning rates were then scaled to higher dimensions using a fixed heuristic rule. For SGD and AdaGrad, we apply a square-root scaling rule which is $\eta(d) = \eta(5)\sqrt{5/d}$. For all Adam-family methods, including RMSprop and HEGA, we use a more conservative quarter power rule which is $\eta(d) = \eta(5)(5/d)^{1/4}$. Other parameters, such as momentum decay rates, were kept constant across all dimensions.

Each optimization run proceeds for a maximum of 10,000 iterations or until the Euclidean norm of the parameter update step falls below a tolerance of 10^{-12}. To prevent the one time cost of JIT compilation from biasing timing measurements, a warm up phase was executed for each optimizer on each dimension before the timed benchmark runs began. If a run produced non-finite values (‘NaN‘ or ‘inf‘), it was marked as a failure.

For each problem instance (a specific function and dimension), the results from all independent runs were aggregated. The wall-clock time was averaged arithmetically. To handle solution quality values that can span several orders of magnitude, the distance to the global minimum, $|f(\mathbf{x}) - f^*|$, was aggregated using the geometric mean. These aggregated, per problem statistics form the basis for the normalized scores and performance analyses presented in Section 7.

7 Experiments

7.1 Evaluation Metrics and Protocol

To ensure a fair comparison, we define a standardized evaluation protocol. For each problem, defined by a specific function and dimension, every optimizer is run from multiple random starting points sampled uniformly from the function's standard domain. Performance is measured along two primary axes. The first is time efficiency, which captures the computational overhead of the algorithm by measuring the wall clock time required to reach the final solution. The second is solution quality, which measures the accuracy of the result as the absolute difference between the function value at the final iterate and the known global minimum, $|f(\mathbf{x}_{\text{final}}) - f^*|$.

To aggregate performance across the diverse set of problems, we compute normalized scores. For a given problem, we first identify the best performing optimizer for each metric (e.g., the minimum time achieved by any optimizer). The normalized score for every other optimizer is then calculated as the ratio of the best performance to its own performance. For a metric m where lower is better,

the normalized score for optimizer s on problem p is:

$$\text{Score}_{p,s}(m) = \frac{\min_{s'} m_{p,s'}}{m_{p,s}}.$$

This transformation places the best performing optimizer at a score of 1.0, while others receive a score between 0 and 1. An overall score is then computed for each problem as the arithmetic mean of the normalized time and quality scores, providing a single, balanced measure of performance.

7.2 Main Benchmark Results (Dimensions 5 – 1000)

The main benchmark demonstrates HEGA's strong and consistent performance in standard settings. Table 1 presents the average normalized scores aggregated across all twenty functions and eight dimensions up to $d = 1000$. HEGA achieves the highest overall score of 0.790, leading all other competitors. HEGA also received the best score in both time efficiency and solution quality.

Table 1: Average normalized performance scores on the main benchmark ($d \in [5, 1000]$). All values are between 0 and 1, with higher values indicating better performance.

Optimizer	Time	Accuracy	Overall
HEGA	0.832	0.748	0.790
AdaGrad	0.541	0.477	0.509
NAdam	0.376	0.625	0.500
AdamAMSGrad	0.401	0.485	0.443
Adam	0.328	0.505	0.417
RMSprop	0.376	0.438	0.407
SGD	0.423	0.379	0.401

The magnitude of this lead is quantified in Table 2. HEGA shows a substantial overall performance improvement against every baseline, ranging from 55.1% over AdaGrad to 96.9% over SGD. The advantage over other adaptive methods like Adam (89.5%) and RMSprop (94.1%) is particularly noteworthy.

Table 2: HEGA percentage improvement over competing optimizers on the main benchmark.

Competitor	Time (%)	Accuracy (%)	Overall (%)
AdaGrad	53.7	56.8	55.1
Adam	153.5	48.0	89.5
AdamAMSGrad	107.4	54.2	78.2
NAdam	121.3	19.8	57.9
RMSprop	121.3	70.7	94.1
SGD	96.5	97.3	96.9

To understand the robustness of this performance, we use Dolan-Moré performance profiles (Figure 1, left). A solver's curve represents the fraction of problems it solved within a factor τ of the best solver. HEGA's profile dominates all others, positioned highest and furthest to the left. This indicates it is the most efficient optimizer (steepest initial rise) and also the most robust, eventually solving the largest fraction of problems to near optimality.

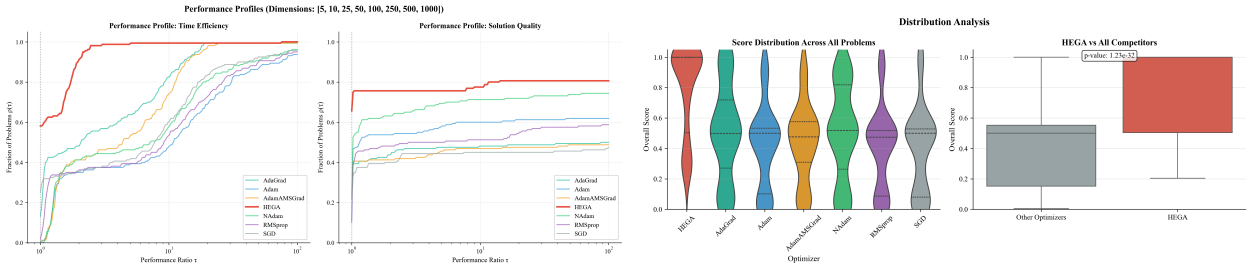

Figure 1: Performance profiles (left) and score distributions (right) on the main benchmark suite (dimensions 5–1000). The profiles show HEGA solves problems faster and more reliably. The distributions show HEGA achieves a statistically significant higher median score with lower variance than the pool of competitors.

The distributional analysis in Figure 1 (right) supports this finding. The violin plot shows that HEGA's distribution of overall scores is concentrated at a higher level than its competitors. The box plot comparison of HEGA against all other optimizers combined reveals a substantially higher median score and a tighter interquartile range, signifying greater consistency. A Mann-Whitney U test confirms this visual result, yielding a p-value of 1.23×10^{-32} and indicating that HEGA's performance is statistically better.

Figure 2 provides a detailed view of performance, showing the normalized overall score for each optimizer on every test function, averaged across the main dimensions. HEGA consistently appears as one of the top performers (green cells) across a wide variety of function types, from the separable Sphere function to the highly non-convex and non-separable Rosenbrock problem. This shows the versatility of its geometric adaptation mechanisms.

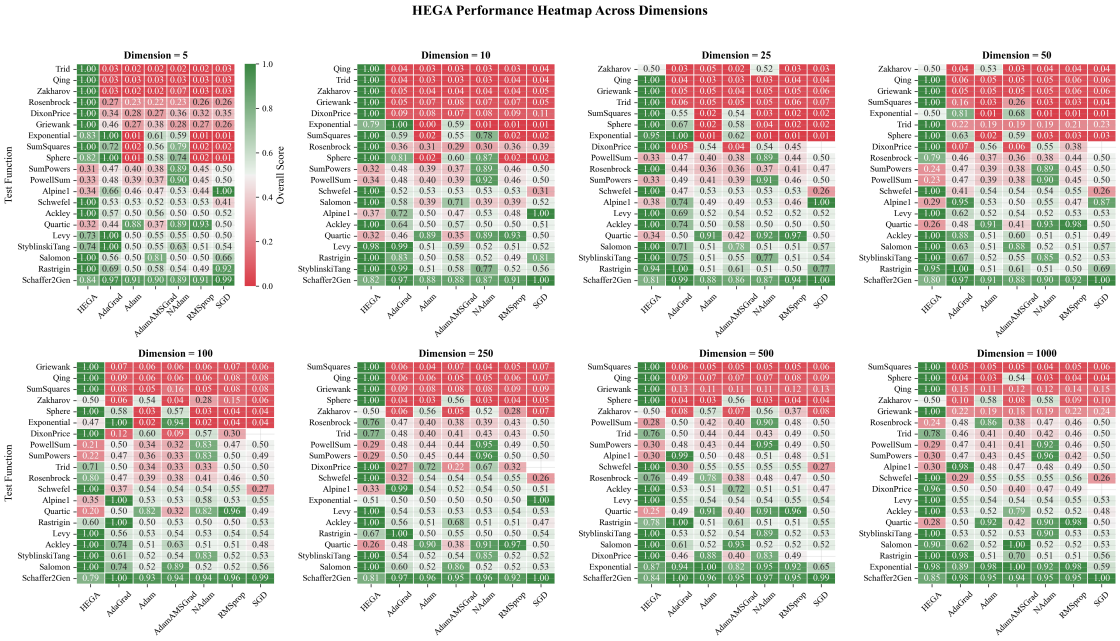

Figure 2: Heatmap of normalized overall scores by function, averaged across main dimensions (5–1000). Green indicates better performance (closer to 1.0). HEGA shows consistently strong performance across the diverse benchmark suite.

7.3 High-Dimensional Stress Test (Dimensions 2000 – 10000)

An important test for any optimizer is its ability to scale to high dimensional problems, where poor conditioning and large search spaces can often degrade performance. We evaluated all methods on dimensions up to $10,000$.

Figure 3 plots the average normalized score as a function of problem dimension, combining data from both the main and stress tests. Most of the tested optimizers show consistent scores as dimension increases including HEGA with HEGA's performance curve remaining significantly higher than the rest. Its ability to use path level geometric information appears to keep providing a distinct advantage in navigating even higher dimensional landscapes.

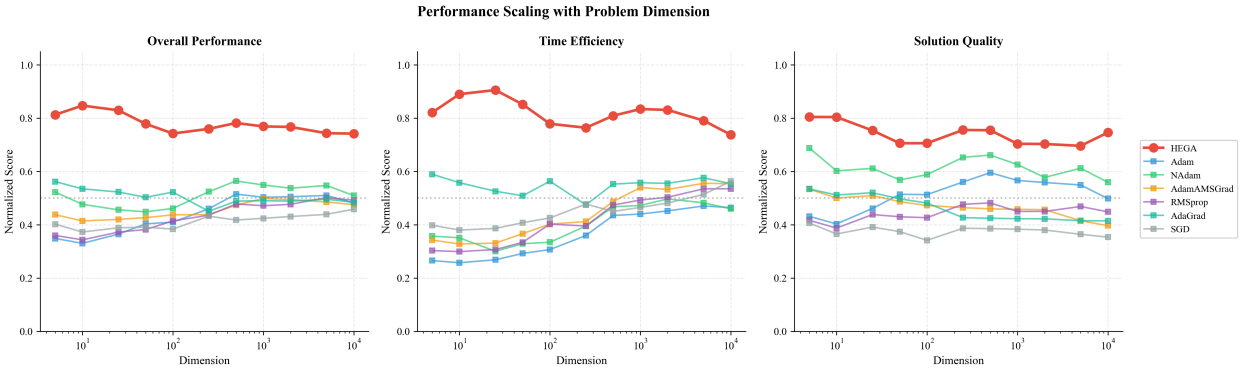

Figure 3: Performance scaling with problem dimension, from $d = 5$ to $d = 10000$. Each point is the average normalized score across all test functions at that dimension. HEGA maintains a consistent high level of performance.

This advantage even in higher dimensions is quantified in Table 3. In the high dimensions ($d \geq 2000$), HEGA's overall performance improvement is still substantial, ranging from 41.3% over NAdam, which was the second best performer in the stress tests, to 69.6% over SGD.

Table 3: HEGA percentage improvement over competing optimizers in the high-dimensional stress test ($d \in [2000, 10000]$).

Competitor	Time (%)	Accuracy (%)	Overall (%)
AdaGrad	40.0	71.4	53.4
Adam	70.2	33.6	50.5
AdamAMSGrad	43.7	69.2	54.8
NAdam	64.0	22.7	41.3
RMSprop	50.0	56.9	53.2
SGD	51.4	95.4	69.6

In summary, the empirical results consistently show that HEGA provides a significant performance benefit over standard first order methods. Its strength is not confined to a specific problem type but is demonstrated across a wide range of functions and is still pronounced even as the problem dimension grows.

8 Discussion and Future Work

HEGA combines two low cost geometric signals with a stable adaptive core, and this combination consistently improves first order performance across a wide range of smooth test functions and dimensions. The aggregate results indicate that HEGA attains the best overall score on twenty classical functions across 5 to 10000 dimensions. The interpolation between a diagonal AMSGrad style preconditioner and a scalar path curvature preconditioner helps align the effective step with the local geometry when the trajectory is coherent, while the alignment controlled learning rate keeps steps conservative when gradients oscillate. This division of labor appears to be a practical way to trade speed for stability without complicated schedules.

Theoretical results support these observations. The regret analysis shows that the mixed metric machinery preserves the $O(\sqrt{T})$ guarantee by staying in the AMSGrad anchor metric and handling the alignment modulated step via deterministic envelopes. The local analysis explains why, in a directional convergence regime near a minimizer, the method behaves essentially like a well tuned scalar preconditioned gradient method and achieves a linear rate. Under the PL condition the same mechanism yields linear decay of function values with a clean variance controlled neighborhood in the stochastic case. The continuous time limit clarifies stability through an explicit small gain condition and highlights how the alignment and curvature states respond to the dynamics of $\mathbf{x}_t$.

There are important limitations. The benchmark is deterministic and smooth, which isolates the geometry but does not explore robustness to gradient noise, data heterogeneity, or nonsmooth structure. The hyperparameter policy is intentionally simple, using a 5D base and analytic scaling with dimension, which favors reproducibility over per problem tuning. Alignment and curvature rely on exponential averages and stabilizers. These design choices can bias estimates when signals are very small or very noisy, and they can slow adaptation when the path alternates between straight segments and sharp turns. Although the failure rate is low, the few failures concentrate on functions with narrow ravines and rapidly changing curvature where per coordinate adaptation can help. Theoretical constants are conservative, and the metric mixing assumption likely admits weaker forms.

Several directions follow naturally. A stochastic evaluation with mini batches, heavy tailed noise models, and adaptive clipping would test the robustness of the alignment and curvature signals. Systematic ablations that independently disable alignment modulation, curvature interpolation, and the AMSGrad anchor would quantify the marginal contribution of each component. Sensitivity studies for α, γ, λ_p, and stabilizers across functions and dimensions would help refine defaults and may suggest automatic rules that replace fixed choices. Backtracking or line search variants deserve attention, since inexpensive acceptance tests can reduce rare failures without erasing the benefits of a large effective step in coherent situations. Extensions that capture more curvature at fixed cost are promising, such as block diagonal preconditioners per layer or module, or directional secant refinements along the momentum direction that adjust the scalar step using Barzilai–Borwein style ratios. These additions preserve $O(d)$ complexity while potentially unlocking superlinear progress within small blocks.

On the theory side, it would be valuable to remove or weaken the metric mixing assumption in the regret proof, sharpen constants in the PL regime using variance dependent bounds, and develop high probability results that track the coupled evolution of alignment, curvature, and momentum. A diffusion limit with state dependent noise could clarify the stationary distribution induced by constant steps and suggest principled noise robust stabilizers. Local results that guarantee quadratic contraction in more than one direction would likely require block structure or richer secant information; formalizing this connection is an open problem.

Finally, broader applications should be explored. Large scale stochastic optimization, deep

neural network training, and reinforcement learning present situations where curvature varies across scales and where stable acceleration is valuable. The evidence suggests that using geometry to guide changes makes the optimizer more efficient and robust. With broader experiments, careful ablations, and refined analysis, HEGA has the potential to become a reliable default in high dimensional settings that require speed, stability, and simplicity.

Acknowledgments

I would like to sincerely thank Dr. Nabil Mesbah for his guidance and input throughout this research.

References

[1] S.-i. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

[2] J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *SIAM Journal on Scientific and Statistical Computing*, 8(1):141–148, 1988.

[3] J. Bernstein, Y.-X. Wang, K. Azizzadenesheli, and A. Anandkumar. signsgd with majority vote is communication efficient and fault tolerant. In *International Conference on Learning Representations*, 2019.

[4] X. Chen, S. Xu, C. Caramanis, and S. Mannor. On the convergence of adam-type algorithms. arXiv preprint, 2018.

[5] T. Dozat. Incorporating nesterov momentum into adam. In *ICLR Workshop Track*, 2016.

[6] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159, 2011.

[7] V. Gupta, T. Koren, and Y. Singer. Shampoo: Preconditioned stochastic tensor optimization. In *International Conference on Machine Learning*, 2018.

[8] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[9] X. Li and colleagues. On the stability of adaptive optimization methods. arXiv preprint, 2019.

[10] L. Liu, H. Jiang, P. He, W. Chen, X. Liu, J. Gao, and J. Han. On the variance of the adaptive learning rate and beyond. arXiv preprint, 2019.

[11] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

[12] M. Mahsereci and P. Hennig. A probabilistic line search for stochastic optimization. *Journal of Machine Learning Research*, 18(119):1–59, 2017.

[13] J. Martens and R. Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning*, pages 2408–2417, 2015.

[14] A. Mokhtari and A. Ribeiro. Diagonal quasi-newton methods for machine learning. arXiv preprint, 2014.

[15] Y. Nesterov. A method for solving the convex programming problem with convergence rate $o(1/k^2)$. *Soviet Mathematics Doklady*, 27:372–376, 1983.

[16] B. T. Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 4(5):1–17, 1964.

[17] N. Qian. On the momentum term in gradient descent learning algorithms. *Neural Networks*, 12(1):145–151, 1999.

[18] S. J. Reddi, S. Kale, and S. Kumar. On the convergence of adam and beyond. In *International Conference on Learning Representations*, 2018. Introduces AMSGrad.

[19] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22(3):400–407, 1951.

[20] N. Shazeer and M. Stern. Adafactor: Adaptive learning rates with sublinear memory. In *International Conference on Machine Learning*, 2018.

[21] T. Tieleman and G. Hinton. Rmsprop: Divide the gradient by a running average of its recent magnitude. Lecture notes for Neural Networks for Machine Learning, 2012.

[22] Y. Wu and colleagues. Sgdr+: Enhanced warm restarts for stochastic optimization. arXiv preprint, 2018.

[23] M. Zaheer, S. J. Reddi, D. Sachan, S. Kale, and S. Kumar. Adaptive methods for nonconvex optimization. arXiv preprint, 2018.

[24] M. R. Zhang, J. Lucas, J. Ba, and G. E. Hinton. Lookahead optimizer: k steps forward, 1 step back. In *Advances in Neural Information Processing Systems*, 2019.

[25] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan. Adabelief optimizer: Adapting stepsizes by the belief in observed gradients. In *Advances in Neural Information Processing Systems*, 2020.

A Algorithmic Details and Hyperparameters

This section records the specific implementation of HEGA used in our study, along with its variants and the hyperparameters that control its behavior. The goal is to ensure full reproducibility. The definitions remain consistent with the notation introduced earlier, particularly the momentum and second moment EMAs, the alignment statistic C_t, the path curvature surrogate $V_{\text{path},t}$, and the effective diagonal preconditioner $\mathbf{P}_{\text{eff},t}$.

Algorithmic Variants

The primary algorithm evaluated in the main experiments, which we refer to simply as HEGA, uses the AMSGrad style monotone accumulator $\hat{\mathbf{v}}_t$ for the coordinate wise preconditioner. This is the version analyzed in our regret and PL convergence results. For diagnostic purposes and to understand the contribution of individual components, we also considered several variants. A key variant replaces the AMSGrad accumulator with the standard Adam second moment $\mathbf{v}_t$, which allows us to isolate the impact of the monotonicity condition on stability. Another more specialized variant HEGA-N, uses a directional Newton-style scalar branch, $P_{\text{path},t} = (V_{\text{path},t} + \epsilon_{\text{path}})^{-1}$, to explore faster local convergence when gradient alignment is high.

Hyperparameters and Practical Tuning

The hyperparameter settings used for HEGA and all baselines in the experimental suite are detailed in Table 4. We maintained a single configuration across all functions and dimensions, with only the base learning rate adjusted via a scaling rule. This policy was chosen to ensure a fair and reproducible comparison.

Table 4: Hyperparameter settings for HEGA used in the main experimental suite.

Hyperparameter	Description	Default Value	Typical Range
β_1	First moment (momentum) decay	0.9	0.8 to 0.95
β_2	Second moment (variance) decay	0.999	0.99 to 0.9995
γ	Geometry EMA decay for C_t and $V_{\text{path},t}$	0.9	0.9 to 0.99
α	LR modulation base in α^{C_t}	2.0	1.0 to 4.0
λ_p	Interpolation exponent in $1 - \lvert C_t \rvert^{\lambda_p}$	2.0	1.0 to 4.0
$\epsilon_c,\ \epsilon_q,\ \epsilon_p$	Denominator stabilizers	10^{-8}	–
δ_0	Curvature estimate floor	10^{-12}	–

Our practical tuning protocol was designed to be minimal. We first selected a base learning rate $\eta(5)$ at dimension $d = 5$ from a small logarithmic grid, holding all other hyperparameters at their default values from Table 4. For any task specific adjustments, we found it effective to first tune α in small increments before revisiting the base learning rate. The algorithm was initialized with zero for all moments and accumulators. No weight decay or gradient clipping was used for the reported benchmarks.

B Proofs of Theoretical Results

This appendix gives complete proofs for all results in Section 5. Throughout, $\|\cdot\|_2$ is the Euclidean norm, $\langle\cdot,\cdot\rangle$ is the Euclidean inner product, and for any diagonal positive definite matrix $\mathbf{H}$ we write $\|x\|_{\mathbf{H}}^2 := \langle \mathbf{H}x, x\rangle$. We recall

$$\mathbf{D}_t := \operatorname{diag}(\mathbf{P}_{\text{eff},t}) \quad \text{and} \quad \mathbf{A}_t := \operatorname{diag}(1/\mathbf{P}_{\text{eff},t}) = \mathbf{D}_t^{-1},$$

and the anchor metric $\bar{\mathbf{A}}_t := \operatorname{diag}(\sqrt{\hat{\mathbf{v}}_t} + \epsilon_p \mathbf{1})$, which is coordinatewise nondecreasing in t because $\hat{\mathbf{v}}_t$ is a max buffer.

A. Proofs for stability and boundedness

Proof of Proposition 7. By Cauchy–Schwarz, $|\langle \tilde{\mathbf{g}}_t, \tilde{\mathbf{g}}_{t-1}\rangle| \le \|\tilde{\mathbf{g}}_t\|_2 \|\tilde{\mathbf{g}}_{t-1}\|_2$. Since $\epsilon_c > 0$, the raw alignment $c_t = \frac{\langle \tilde{\mathbf{g}}_t, \tilde{\mathbf{g}}_{t-1}\rangle}{\|\tilde{\mathbf{g}}_t\|_2 \|\tilde{\mathbf{g}}_{t-1}\|_2 + \epsilon_c}$ satisfies $|c_t| \le 1$. The initialization gives $C_0 = 0$. The smoothed alignment $C_t = \gamma C_{t-1} + (1-\gamma)c_t$ is a convex combination of terms in $[-1, 1]$, hence $C_t \in [-1, 1]$ for all t. Monotonicity of $z \mapsto \alpha^z$ gives $M_t = \alpha^{C_t} \in [\alpha^{-1}, \alpha]$. Since $|C_t| \in [0, 1]$ and $\lambda_p > 0$, one has $\lambda_t = 1 - |C_t|^{\lambda_p} \in [0, 1]$. □

Lemma 18. *For $\epsilon_q > 0$ and $a \ge 0$, the function $\phi(a) = \frac{a}{a^2+\epsilon_q}$ attains its maximum $1/(2\sqrt{\epsilon_q})$ at $a = \sqrt{\epsilon_q}$.*

Proof. Compute $\phi'(a) = \frac{\epsilon_q - a^2}{(a^2+\epsilon_q)^2}$. The unique critical point on $[0, \infty)$ is $a = \sqrt{\epsilon_q}$. The sign change of ϕ' shows a maximum and $\phi(\sqrt{\epsilon_q}) = 1/(2\sqrt{\epsilon_q})$. □

Proof of Proposition 8. The lower bound holds by inspection since $V_{\text{path},0} = \delta_0$ and the update $V_{\text{path},t} = \gamma V_{\text{path},t-1} + (1-\gamma)\max\{H_{\text{est},t}, \delta_0\}$ is a convex combination of values at least δ_0. For the

upper bound write $\mathbf{s}_t = \mathbf{x}_{t-1} - \mathbf{x}_{t-2}$ and

$$\mathbf{y}_t = \nabla f_t(\mathbf{x}_{t-1}) - \nabla f_{t-1}(\mathbf{x}_{t-2}) = \underbrace{\nabla f_t(\mathbf{x}_{t-1}) - \nabla f_t(\mathbf{x}_{t-2})}_{\mathrm{I}} + \underbrace{\nabla f_t(\mathbf{x}_{t-2}) - \nabla f_{t-1}(\mathbf{x}_{t-2})}_{\mathrm{II}}.$$

Assumption 3 gives $\langle \mathrm{I}, \mathbf{s}_t \rangle \le L\|\mathbf{s}_t\|_2^2$. Assumption 6 gives $\|\mathrm{II}\|_2 \le \Delta$, hence $\langle \mathrm{II}, \mathbf{s}_t \rangle \le \Delta\|\mathbf{s}_t\|_2$. Therefore

$$H_{\mathrm{est},t} = \frac{\langle \mathbf{y}_t, \mathbf{s}_t \rangle}{\|\mathbf{s}_t\|_2^2 + \epsilon_q} \le L + \Delta \frac{\|\mathbf{s}_t\|_2}{\|\mathbf{s}_t\|_2^2 + \epsilon_q} \le L + \frac{\Delta}{2\sqrt{\epsilon_q}},$$

using Lemma 18. The clipped exponential average then satisfies $V_{\mathrm{path},t} \le \max\{\delta_0,\, L + \Delta/(2\sqrt{\epsilon_q})\} =: V_{\max}$ for all t. When $\Delta = 0$, $V_{\max} = \max\{\delta_0, L\}$. □

Proof of Proposition 9. Assumption 2 gives $|\tilde{g}_{t,i}| \le G_\infty$ almost surely, so $v_{t,i} \le G_\infty^2$ by induction and $\hat{v}_{t,i} \le G_\infty^2$ by the max rule. Hence

$$(G_\infty + \epsilon_p)^{-1} \le P_{\mathrm{L2},t,i} = \frac{1}{\sqrt{\hat{v}_{t,i}} + \epsilon_p} \le \epsilon_p^{-1}.$$

Proposition 8 gives $\delta_0 \le V_{\mathrm{path},t} \le V_{\max}$, hence

$$(\sqrt{V_{\max}} + \epsilon_p)^{-1} \le P_{\mathrm{path},t} = \frac{1}{\sqrt{V_{\mathrm{path},t}} + \epsilon_p} \le (\sqrt{\delta_0} + \epsilon_p)^{-1}.$$

The effective term is a convex combination $P_{\mathrm{eff},t,i} = \lambda_t P_{\mathrm{L2},t,i} + (1-\lambda_t) P_{\mathrm{path},t}$. Taking the minimum of the lower endpoints and the maximum of the upper endpoints over the two ranges yields the claimed constants $P_{\min}$ and $P_{\max}$. □

Proof of Lemma 10. Fix t and i, write $A := P_{\mathrm{L2},t,i}$ and $B := P_{\mathrm{path},t}$. Then $P_{\mathrm{eff},t,i} = \lambda_t A + (1-\lambda_t)B$. For the lower bound,

$$\inf_{\lambda\in[0,1]} \frac{\lambda A + (1-\lambda)B}{A} = \min\left\{1, \frac{B}{A}\right\} \ge \frac{B_{\min}}{A_{\max}} = \frac{(\sqrt{V_{\max}} + \epsilon_p)^{-1}}{\epsilon_p^{-1}} = \frac{\epsilon_p}{\sqrt{V_{\max}} + \epsilon_p} = c_{\min}.$$

For the upper bound,

$$\sup_{\lambda\in[0,1]} \frac{\lambda A + (1-\lambda)B}{A} = \max\left\{1, \frac{B}{A}\right\} \le \frac{B_{\max}}{A_{\min}} = \frac{(\sqrt{\delta_0} + \epsilon_p)^{-1}}{(G_\infty + \epsilon_p)^{-1}} = \frac{G_\infty + \epsilon_p}{\sqrt{\delta_0} + \epsilon_p} = c_{\max}.$$

Therefore $c_{\min} A \le P_{\mathrm{eff},t,i} \le c_{\max} A$ and, equivalently, $c_{\min}\, \bar{\mathbf{A}}_t^{-1} \preceq \mathbf{D}_t \preceq c_{\max}\, \bar{\mathbf{A}}_t^{-1}$. □

B. A weighted telescoping inequality for varying anchor metrics

We state the weighted telescoping tool used in the regret proof. It keeps the anchor metric and the time-varying step together.

Lemma 19 (Weighted telescoping with increasing diagonal anchors)**.** *Let $\{\bar{\mathbf{A}}_t\}_{t\ge1}$ be diagonal positive definite matrices with $\bar{\mathbf{A}}_{t+1} \succeq \bar{\mathbf{A}}_t$ for all t. Let $a_t > 0$ be nondecreasing. For any sequence $\{\mathbf{z}_t\}_{t\ge0} \subset \mathcal{X}$ and any $\{\tilde{\mathbf{z}}_t\}_{t\ge1} \subset \mathcal{X}$,*

$$\sum_{t=1}^{T} a_t\big(\|\mathbf{z}_{t-1} - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2 - \|\tilde{\mathbf{z}}_t - \mathbf{x}^*\|_{\bar{\mathbf{A}}_t}^2\big) \;\le\; a_T\|\mathbf{z}_T - \mathbf{x}^*\|_{\bar{\mathbf{A}}_T}^2 + a_1\|\mathbf{z}_0 - \mathbf{x}^*\|_{\bar{\mathbf{A}}_1}^2.$$

Proof. Write $V_t(u) := \|u - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}$. Then

$$a_t\big(V_t(\mathbf{z}_{t-1}) - V_t(\tilde{\mathbf{z}}_t)\big) = \big(a_t V_t(\mathbf{z}_{t-1}) - a_{t+1}V_{t+1}(\mathbf{z}_t)\big) + \big(a_{t+1}V_{t+1}(\mathbf{z}_t) - a_t V_t(\tilde{\mathbf{z}}_t)\big).$$

Summing over $t = 1, \dots, T$ gives

$$\sum_{t=1}^{T} a_t\big(V_t(\mathbf{z}_{t-1}) - V_t(\tilde{\mathbf{z}}_t)\big) = \underbrace{a_1 V_1(\mathbf{z}_0) - a_T V_T(\mathbf{z}_T)}_{\mathrm{I}} + \sum_{t=1}^{T-1} \underbrace{\big(a_{t+1}V_{t+1}(\mathbf{z}_t) - a_t V_t(\tilde{\mathbf{z}}_t)\big)}_{\mathrm{II}_t}.$$

For II_t, monotonicity of $\bar{\mathbf{A}}_{t+1}$ and of a_t gives $a_{t+1}V_{t+1}(\mathbf{z}_t) \geq a_t V_t(\mathbf{z}_t)$. Since $\tilde{\mathbf{z}}_t \in \mathcal{X}$ is arbitrary here, $V_t(\mathbf{z}_t) - V_t(\tilde{\mathbf{z}}_t) \geq -V_t(\tilde{\mathbf{z}}_t)$. Hence $\mathrm{II}_t \geq -a_t V_t(\tilde{\mathbf{z}}_t)$. Dropping these nonpositive terms yields

$$\sum_{t=1}^{T} a_t\big(V_t(\mathbf{z}_{t-1}) - V_t(\tilde{\mathbf{z}}_t)\big) \leq a_1 V_1(\mathbf{z}_0) + a_T V_T(\mathbf{z}_T),$$

which proves the claim. □

C. A weighted AMSGrad sum

The regret proof multiplies the AMSGrad branch by $s_t = \eta\,\alpha^{C_t}/\sqrt{t}$. The useful inequality is therefore weighted by $1/\sqrt{t}$.

Lemma 20 (Weighted AMSGrad bound)**.** *For each coordinate i,*

$$\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}} + \epsilon_p} \;\leq\; \frac{2\,G_\infty^2}{(1-\beta_1)^2\,\epsilon_p}\sqrt{T}.$$

Consequently, $\sum_{t=1}^{T} \frac{1}{\sqrt{t}} \sum_{i=1}^{d} \frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}}+\epsilon_p} \leq \frac{2d\,G_\infty^2}{(1-\beta_1)^2\,\epsilon_p}\sqrt{T}$.

Proof. By Assumption 2, $|\tilde{g}_{t,i}| \leq G_\infty$ almost surely. The bias-corrected momentum is a convex combination of past stochastic gradients, so $|\hat{m}_{t,i}| \leq (1-\beta_1)^{-1} G_\infty$. Also $\sqrt{\hat{v}_{t,i}} + \epsilon_p \geq \epsilon_p$. Hence

$$\frac{1}{\sqrt{t}} \frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}} + \epsilon_p} \leq \frac{1}{\sqrt{t}} \frac{G_\infty^2}{(1-\beta_1)^2\,\epsilon_p}.$$

Summing over t and using $\sum_{t=1}^{T} t^{-1/2} \leq 2\sqrt{T}$ proves the claim. □

D. Proof of the regret bound

We prove Theorem 12 without any mixing assumption. The argument uses the ghost iterate in the anchor metric, the *monotone envelopes*

$$\underline{s}_t := \frac{\eta}{\alpha\sqrt{t}} \;\leq\; s_t := \eta\,\alpha^{C_t}/\sqrt{t} \;\leq\; \overline{s}_t := \frac{\eta\,\alpha}{\sqrt{t}},$$

and the weighted telescoping Lemma 19 with the nondecreasing weights $a_t := 1/\underline{s}_t = (\alpha/\eta)\sqrt{t}$.

Lemma 21 (Mirror one step bound in the anchor metric)**.** *Let* $\tilde{\mathbf{x}}_t = \mathrm{Proj}_{\mathcal{X}}^{(\bar{\mathbf{A}}_t)}\left(\mathbf{x}_{t-1} - s_t\,\bar{\mathbf{A}}_t^{-1}\hat{\mathbf{m}}_t\right)$ *with* $s_t = \eta_t \alpha^{C_t}$*. Then*

$$\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^* \rangle \leq \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}}{2s_t} + \frac{s_t}{2}\,\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}}.$$

Proof. Let $\psi_t(x) = \frac{1}{2}\|x\|^2_{\bar{\mathbf{A}}_t}$. The optimality condition for $\tilde{\mathbf{x}}_t$ says that for all $u \in \mathcal{X}$, $\langle \bar{\mathbf{A}}_t(\tilde{\mathbf{x}}_t - \mathbf{y}_t), u - \tilde{\mathbf{x}}_t\rangle \geq 0$ with $\mathbf{y}_t = \mathbf{x}_{t-1} - s_t\,\bar{\mathbf{A}}_t^{-1}\hat{\mathbf{m}}_t$. Choosing $u = \mathbf{x}^*$ gives $\langle \hat{\mathbf{m}}_t, \tilde{\mathbf{x}}_t - \mathbf{x}^*\rangle \leq \frac{1}{s_t}\langle \bar{\mathbf{A}}_t(\mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t), \tilde{\mathbf{x}}_t - \mathbf{x}^*\rangle$. The three-point identity for the quadratic Bregman divergence yields

$$2\langle \bar{\mathbf{A}}_t(\mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t), \tilde{\mathbf{x}}_t - \mathbf{x}^*\rangle = \|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t\|^2_{\bar{\mathbf{A}}_t}.$$

Hence

$$\langle \hat{\mathbf{m}}_t, \tilde{\mathbf{x}}_t - \mathbf{x}^*\rangle \leq \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}}{2s_t} - \frac{\|\mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t\|^2_{\bar{\mathbf{A}}_t}}{2s_t}.$$

Adding $\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t\rangle$ to both sides and applying Cauchy–Schwarz in the $\bar{\mathbf{A}}_t$ inner product gives $\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t\rangle \leq \frac{s_t}{2}\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}} + \frac{\|\mathbf{x}_{t-1} - \tilde{\mathbf{x}}_t\|^2_{\bar{\mathbf{A}}_t}}{2s_t}$. This cancels the last negative term and proves the claim. □

Proof of Theorem 12. By convexity, $R(T) = \sum_{t=1}^T f_t(\mathbf{x}_{t-1}) - f_t(\mathbf{x}^*) \leq \sum_{t=1}^T \langle \nabla f_t(\mathbf{x}_{t-1}), \mathbf{x}_{t-1} - \mathbf{x}^*\rangle$. We work directly with $\tilde{\mathbf{g}}_t$ since $\mathbb{E}[\tilde{\mathbf{g}}_t \mid \mathcal{F}_{t-1}] = \nabla f_t(\mathbf{x}_{t-1})$. Taking total expectation and conditioning on $\mathcal{F}_{t-1}$,

$$\mathbb{E}R(T) \;\leq\; \sum_{t=1}^T \mathbb{E}\big[\langle \tilde{\mathbf{g}}_t, \mathbf{x}_{t-1} - \mathbf{x}^*\rangle\big] = \sum_{t=1}^T \mathbb{E}\big[\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^*\rangle\big] + \sum_{t=1}^T \mathbb{E}\big[\langle \tilde{\mathbf{g}}_t - \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^*\rangle\big] =: \mathbb{E}S_1 + \mathbb{E}S_2.$$

Step 1: bound S_1 via the ghost inequality and the envelopes. Apply Lemma 21 and then replace $1/s_t$ by $1/\underline{s}_t$ and s_t by $\overline{s}_t$:

$$\langle \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^*\rangle \leq \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}}{2\,\underline{s}_t} + \frac{\overline{s}_t}{2}\,\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}}.$$

Summing and using Lemma 19 with $a_t = 1/\underline{s}_t$ yields

$$\sum_{t=1}^T \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t} - \|\tilde{\mathbf{x}}_t - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}}{2\,\underline{s}_t} \;\leq\; \frac{a_T}{2}\,\|\mathbf{x}_T - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_T} + \frac{a_1}{2}\,\|\mathbf{x}_0 - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_1}.$$

Since $\|\mathbf{z}\|^2_{\bar{\mathbf{A}}_t} \leq (G_\infty + \epsilon_p)\|\mathbf{z}\|_2^2 \leq (G_\infty + \epsilon_p)D^2$ and $a_T = (\alpha/\eta)\sqrt{T}$, this part is $O(\sqrt{T})$. For the quadratic term, comparability gives $\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}} \leq c_{\max}\langle \mathbf{D}_t\hat{\mathbf{m}}_t, \hat{\mathbf{m}}_t\rangle$. Split $P_{\mathrm{eff},t,i} = \lambda_t P_{\mathrm{L2},t,i} + (1 - \lambda_t)P_{\mathrm{path},t}$ and use $\overline{s}_t \leq (\eta\alpha)/\sqrt{t}$:

$$\sum_{t=1}^T \frac{\overline{s}_t}{2}\,\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}} \;\leq\; \frac{c_{\max}}{2}\Big[\underbrace{\eta\alpha \sum_{t=1}^T \frac{1}{\sqrt{t}} \sum_{i=1}^d \frac{\hat{m}_{t,i}^2}{\sqrt{\hat{v}_{t,i}} + \epsilon_p}}_{=:Q_1} + \underbrace{\sum_{t=1}^T \frac{\eta\alpha}{\sqrt{t}}\,P_{\mathrm{path},t}\,\|\hat{\mathbf{m}}_t\|_2^2}_{=:Q_2}\Big].$$

By Lemma 20, $Q_1 = O(\sqrt{T})$. For Q_2, $P_{\mathrm{path},t} \leq (\sqrt{\delta_0} + \epsilon_p)^{-1}$ and $\|\hat{\mathbf{m}}_t\|_2 \leq \sqrt{d}\,G_\infty/(1 - \beta_1)$, thus $Q_2 = O(\sqrt{T})$. Altogether $\mathbb{E}S_1 = O(\sqrt{T})$.

Step 2: bound S_2 (difference term). By Cauchy–Schwarz in the $\bar{\mathbf{A}}_t$ inner product and Young,

$$\langle \tilde{\mathbf{g}}_t - \hat{\mathbf{m}}_t, \mathbf{x}_{t-1} - \mathbf{x}^*\rangle \leq \frac{\|\mathbf{x}_{t-1} - \mathbf{x}^*\|^2_{\bar{\mathbf{A}}_t}}{2\,\underline{s}_t} + \frac{\overline{s}_t}{2}\,\|\tilde{\mathbf{g}}_t - \hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}}.$$

The first term is $O(\sqrt{T})$ by the same envelope-telescoping bound as in Step 1. For the second term, $\|\tilde{\mathbf{g}}_t - \hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}} \leq 2\|\tilde{\mathbf{g}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}} + 2\|\hat{\mathbf{m}}_t\|^2_{\bar{\mathbf{A}}_t^{-1}}$, and $\bar{\mathbf{A}}_t^{-1} \preceq c_{\max}\mathbf{D}_t$. The two pieces are then controlled

exactly as Q_1 and Q_2 above (with $\tilde{\mathbf{g}}_t$ in place of $\hat{\mathbf{m}}_t$), giving $O(\sqrt{T})$. Taking expectations preserves all bounds since the right-hand sides are deterministic upper envelopes. Therefore $\mathbb{E}S_2 = O(\sqrt{T})$.

Combining Steps 1–2 proves $\mathbb{E}R(T) = O(\sqrt{T})$. In the deterministic case $\tilde{\mathbf{g}}_t = \nabla f_t(\mathbf{x}_{t-1})$, the same argument gives the pointwise bound $R(T) = O(\sqrt{T})$. □

E. Proof of local linear convergence

We prove Theorem 15. The argument is deterministic with constant step $s_t = \eta\alpha^{C_t}$, uses Assumption 14 to ensure the projection is nonexpansive in a neighborhood of $\mathbf{x}^*$, and combines the preconditioner bounds with a quantitative tracking of the EMA.

Lemma 22 (EMA tracking)**.** *Assume f is L smooth. Then*

$$\|\hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_{t-1})\|_2 \;\le\; \frac{\beta_1}{1-\beta_1} L \sum_{j=1}^{\infty} \beta_1^{j-1} \|\mathbf{x}_{t-j} - \mathbf{x}_{t-j-1}\|_2.$$

Proof. Write $\hat{\mathbf{m}}_t = \frac{1}{1-\beta_1^t}\sum_{k=1}^{t}(1-\beta_1)\beta_1^{t-k}\nabla f(\mathbf{x}_{k-1})$. Add and subtract $\nabla f(\mathbf{x}_{t-1})$ inside the sum, use the triangle inequality, and apply L-smoothness,

$$\|\hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_{t-1})\|_2 \le \frac{1}{1-\beta_1^t}\sum_{k=1}^{t}(1-\beta_1)\beta_1^{t-k} L \|\mathbf{x}_{k-1} - \mathbf{x}_{t-1}\|_2.$$

Expand the telescoping difference $\|\mathbf{x}_{k-1} - \mathbf{x}_{t-1}\|_2 \le \sum_{r=k}^{t-1}\|\mathbf{x}_r - \mathbf{x}_{r-1}\|_2$. Swap sums and use $\sum_{k=1}^{r}(1-\beta_1)\beta_1^{t-k} \le \beta_1^{t-r}/(1-\beta_1)$. Let $j = t - r$ to obtain the claimed bound. □

Proof of Theorem 15. Assume f is L smooth and μ strongly convex. Let $\mathbf{e}_t = \mathbf{x}_t - \mathbf{x}^*$. In the neighborhood specified by Assumption 14, the projection is inactive (or nonexpansive along the iterates), so $\mathbf{e}_t = \mathbf{e}_{t-1} - s_t\,\mathbf{D}_t\hat{\mathbf{m}}_t$. Therefore

$$\|\mathbf{e}_t\|_2^2 = \|\mathbf{e}_{t-1}\|_2^2 - 2s_t\langle \mathbf{D}_t\hat{\mathbf{m}}_t, \mathbf{e}_{t-1}\rangle + s_t^2\|\mathbf{D}_t\hat{\mathbf{m}}_t\|_2^2.$$

Decompose $\hat{\mathbf{m}}_t = \nabla f(\mathbf{x}_{t-1}) + \boldsymbol{\delta}_t$ with $\boldsymbol{\delta}_t := \hat{\mathbf{m}}_t - \nabla f(\mathbf{x}_{t-1})$. Using $P_{\min} \le P_{\mathrm{eff},t,i} \le P_{\max}$, strong convexity, and Cauchy–Schwarz,

$$\langle \mathbf{D}_t\hat{\mathbf{m}}_t, \mathbf{e}_{t-1}\rangle \ge P_{\min}\langle \nabla f(\mathbf{x}_{t-1}), \mathbf{e}_{t-1}\rangle - P_{\max}\|\boldsymbol{\delta}_t\|_2\|\mathbf{e}_{t-1}\|_2 \ge P_{\min}\mu\|\mathbf{e}_{t-1}\|_2^2 - P_{\max}\|\boldsymbol{\delta}_t\|_2\|\mathbf{e}_{t-1}\|_2.$$

Also $\|\mathbf{D}_t\hat{\mathbf{m}}_t\|_2^2 \le 2P_{\max}^2\big(\|\nabla f(\mathbf{x}_{t-1})\|_2^2 + \|\boldsymbol{\delta}_t\|_2^2\big) \le 2P_{\max}^2(L^2\|\mathbf{e}_{t-1}\|_2^2 + \|\boldsymbol{\delta}_t\|_2^2)$. Hence

$$\|\mathbf{e}_t\|_2^2 \le \Big(1 - 2s_tP_{\min}\mu + 2s_t^2P_{\max}^2L^2\Big)\|\mathbf{e}_{t-1}\|_2^2 + 2s_tP_{\max}\|\boldsymbol{\delta}_t\|_2\|\mathbf{e}_{t-1}\|_2 + 2s_t^2P_{\max}^2\|\boldsymbol{\delta}_t\|_2^2.$$

By Lemma 22, $\|\boldsymbol{\delta}_t\|_2 \le C_\delta\sum_{j\ge1}\beta_1^{j-1}\|\mathbf{x}_{t-j}-\mathbf{x}_{t-j-1}\|_2$ with $C_\delta = \frac{\beta_1}{1-\beta_1}L$. Since $\mathbf{x}_k-\mathbf{x}_{k-1} = s_k\,\mathbf{D}_k\hat{\mathbf{m}}_k$, one has $\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2 \le s_kP_{\max}\|\hat{\mathbf{m}}_k\|_2$. Near $\mathbf{x}^*$, $\|\hat{\mathbf{m}}_k\|_2 \le c_g\|\nabla f(\mathbf{x}_{k-1})\|_2 \le c_gL\|\mathbf{e}_{k-1}\|_2$ for a constant c_g that depends only on β_1. With $s_k \equiv s \in [\eta\alpha^{-1}, \eta\alpha]$ and $\sum_{j\ge1}\beta_1^{j-1} = 1/(1-\beta_1)$, one gets $\|\boldsymbol{\delta}_t\|_2 \le \bar{c}\,\|\mathbf{e}_{t-1}\|_2$ for a constant $\bar{c}$ that can be made as small as desired by taking η small. Substituting and absorbing the mixed term by $ab \le \frac{1}{2}\varepsilon a^2 + \frac{1}{2\varepsilon}b^2$ yields

$$\|\mathbf{e}_t\|_2^2 \le \Big(1 - 2sP_{\min}\mu + 2s^2P_{\max}^2L^2 + \tilde{c}\,s\Big)\|\mathbf{e}_{t-1}\|_2^2,$$

with $\tilde{c}$ that can be made arbitrarily small by decreasing η. Choose $\eta > 0$ so that $2sP_{\min}\mu - 2s^2P_{\max}^2L^2 - \tilde{c}\,s \ge \kappa_0 > 0$. This gives $\|\mathbf{e}_t\|_2^2 \le (1-\kappa)\|\mathbf{e}_{t-1}\|_2^2$ for some $\kappa \in (0,1)$, hence $\|\mathbf{e}_t\|_2 \le \rho\|\mathbf{e}_{t-1}\|_2$ with $\rho = \sqrt{1-\kappa} \in (0,1)$. If the projection activates, Euclidean projection is nonexpansive and the bound still holds. □

F. Proofs under the PL condition

We now prove Theorem 16 and Corollary 17. The argument uses the descent lemma, the preconditioner bounds, and the EMA tracking bound.

Proof of Theorem 16. Let $s_t = \eta\alpha^{C_t} \in [\eta\alpha^{-1}, \eta\alpha]$ with $\eta > 0$ constant. Using L-smoothness of f and the update,

$$f(\mathbf{x}_t) \le f(\mathbf{x}_{t-1}) - \langle \nabla f(\mathbf{x}_{t-1}), s_t\, \mathbf{D}_t \hat{\mathbf{m}}_t \rangle + \frac{L s_t^2}{2} \|\mathbf{D}_t \hat{\mathbf{m}}_t\|_2^2.$$

Write $\hat{\mathbf{m}}_t = \nabla f(\mathbf{x}_{t-1}) + \boldsymbol{\delta}_t$. With $P_{\min} \le P_{\mathrm{eff},t,i} \le P_{\max}$,

$$\langle \nabla f, \mathbf{D}_t \hat{\mathbf{m}}_t \rangle \ge P_{\min} \|\nabla f\|_2^2 - P_{\max} \|\nabla f\|_2 \|\boldsymbol{\delta}_t\|_2, \qquad \|\mathbf{D}_t \hat{\mathbf{m}}_t\|_2^2 \le P_{\max}^2 (\|\nabla f\|_2 + \|\boldsymbol{\delta}_t\|_2)^2,$$

where all norms are at $\mathbf{x}_{t-1}$. By Lemma 22 and the small-step choice, $\|\boldsymbol{\delta}_t\|_2 \le \rho' \|\nabla f(\mathbf{x}_{t-1})\|_2$ with $\rho' \in (0, 1/2)$. Substituting and shrinking η if necessary yields

$$f(\mathbf{x}_t) \le f(\mathbf{x}_{t-1}) - \frac{s_t}{2} P_{\min} \|\nabla f(\mathbf{x}_{t-1})\|_2^2.$$

The PL condition gives $\frac{1}{2}\|\nabla f(\mathbf{x})\|_2^2 \ge \mu_{\mathrm{PL}} \big(f(\mathbf{x}) - f^*\big)$ on the relevant sublevel set. Therefore

$$f(\mathbf{x}_t) - f^* \le \big(1 - s_t P_{\min} \mu_{\mathrm{PL}}\big)\big(f(\mathbf{x}_{t-1}) - f^*\big) \le \big(1 - \eta\, \mu_{\mathrm{PL}}\, \alpha^{-1} P_{\min}\big)\big(f(\mathbf{x}_{t-1}) - f^*\big),$$

which proves linear decay with contraction $\kappa = \eta\, \mu_{\mathrm{PL}}\, \alpha^{-1} P_{\min} \in (0, 1)$. □

Proof of Corollary 17. Let $\tilde{\mathbf{g}}_t = \nabla f(\mathbf{x}_{t-1}) + \boldsymbol{\xi}_t$ with $\mathbb{E}[\boldsymbol{\xi}_t \mid \mathcal{F}_{t-1}] = 0$ and $\mathbb{E}[\|\boldsymbol{\xi}_t\|_2^2 \mid \mathcal{F}_{t-1}] \le \sigma^2$. Apply the descent lemma conditionally,

$$\mathbb{E}_{t-1}\big[f(\mathbf{x}_t)\big] \le f(\mathbf{x}_{t-1}) - s_t\, \mathbb{E}_{t-1}\big[\langle \nabla f(\mathbf{x}_{t-1}), \mathbf{D}_t \hat{\mathbf{m}}_t \rangle\big] + \frac{L s_t^2}{2} \mathbb{E}_{t-1}\big[\|\mathbf{D}_t \hat{\mathbf{m}}_t\|_2^2\big].$$

Cross terms with $\boldsymbol{\xi}_t$ vanish by zero mean. Using the same bounds as in the deterministic case and $P_{\max}^2\, \mathbb{E}_{t-1}[\|\hat{\mathbf{m}}_t\|_2^2] \le c_4 \|\nabla f(\mathbf{x}_{t-1})\|_2^2 + c_5 \sigma^2$ gives

$$\mathbb{E}_{t-1}\big[f(\mathbf{x}_t) - f^*\big] \le \big(1 - \eta\, \mu_{\mathrm{PL}}\, \alpha^{-1} P_{\min}\big)\big(f(\mathbf{x}_{t-1}) - f^*\big) + c_6 \eta^2 \sigma^2,$$

for constants c_4, c_5, c_6 depending only on $L, P_{\min}, P_{\max}, \alpha$. Taking total expectation yields linear convergence to a noise-dominated neighborhood. With $s_t = \eta_0/t$, the Robbins–Siegmund lemma gives $\mathbb{E}[f(\mathbf{x}_t) - f^*] = O(1/t)$. □

C Benchmark Definitions and Domains

Table 5 summarizes the twenty objectives used in our study. For each function we record qualitative properties, the canonical domain used for all dimensions, and the global minimizer/value. Exact algebraic forms appear in Table 6. All methods use Euclidean projection $\Pi_{\mathcal{X}}$ onto the stated domain.

Table 5: Set of twenty benchmarks used in our code. Mod.=modality; Sep.=separability; Cvx.=convexity; Cond.=typical conditioning. Canonical domains follow standard practice or our code defaults.

Name	Mod.	Sep.	Cvx.	Cond.	Canonical domain $\mathcal{X}$	Global minimizer $x^\star$ and value $f^\star$
Sphere	Uni	Yes	Yes	Low	$[-5.12,\, 5.12]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Rosenbrock	Uni	No	No	High	$[-2.048,\, 2.048]^d$	$x^\star = \mathbf{1},\ f^\star = 0$
Rastrigin	Multi (high)	Yes	No	Mod.	$[-5.12,\, 5.12]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Ackley	Multi	Yes	No	Mod.	$[-32.768,\, 32.768]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Griewank	Multi	Partial	No	Mod.	$[-600,\, 600]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Schwefel 2.26	Multi	Yes	No	Mod.	$[-500,\, 500]^d$	$x^\star = 420.968746\ldots\,\mathbf{1},\ f^\star = 0$
Lévy (N.13)	Multi	Yes	No	Mod.	$[-10,\, 10]^d$	$x^\star = \mathbf{1},\ f^\star = 0$
Styblinski–Tang	Multi	Yes	No	Mod.	$[-5,\, 5]^d$	$x^\star \approx -2.903534\,\mathbf{1},\ f^\star \approx -39.166\,d$
Zakharov	Uni	No	Yes	Mod.	$[-5,\, 10]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Dixon–Price	Multi	No	No	Mod./High	$[-10,\, 10]^d$	$x_1^\star = 1,\ \ x_i^\star = 2^{-\frac{2^i-2}{2^i}}\ (i \geq 2);\ \ f^\star = 0$
Powell Sum†	Uni	Yes	Yes	Mod.	$[-1,\, 1]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Trid	Uni	No	No	Mod.	$[-d^2,\, d^2]^d$	$x_i^\star = i(d+1-i);\ f^\star = -\frac{d(d+4)(d-1)}{6}$
Sum of Different Powers†	Uni	Yes	Yes	Mod.	$[-1,\, 1]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Qing	Multi	Yes	No	Mod.	$[-500,\, 500]^d$	$x_i^\star = \pm\sqrt{i};\ f^\star = 0$
Salomon	Multi	No	No	Mod.	$[-100,\, 100]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Alpine 1	Multi	Yes	No	Mod.	$[-10,\, 10]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Exponential	Uni	No	No	Low	$[-1,\, 1]^d$	$x^\star = \mathbf{0},\ f^\star = -1$
Schaffer N.2 (gen.)	Multi	No	No	Mod.	$[-100,\, 100]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
Quartic (weighted)	Uni	Yes	Yes	Mod.	$[-1.28,\, 1.28]^d$	$x^\star = \mathbf{0},\ f^\star = 0$
SumSquares (weighted)	Uni	Yes	Yes	Mod./High	$[-10,\, 10]^d$	$x^\star = \mathbf{0},\ f^\star = 0$

Table 6: Closed form definitions used in our code. Let $\mathbf{x} = (x_1, \ldots, x_d)^\top$ and $r = \sqrt{\sum_{i=1}^{d} x_i^2}$.

Name	**Definition** $f(\mathbf{x})$
Sphere	$\sum_{i=1}^{d} x_i^2$
Rosenbrock	$\sum_{i=1}^{d-1} \Big(100\,(x_{i+1} - x_i^2)^2 + (1 - x_i)^2\Big)$
Rastrigin	$10d + \sum_{i=1}^{d} \big(x_i^2 - 10\cos(2\pi x_i)\big)$
Ackley	$-20\exp\Big(-0.2\sqrt{\frac{1}{d}\sum_{i=1}^{d} x_i^2}\Big) \;-\; \exp\Big(\frac{1}{d}\sum_{i=1}^{d}\cos(2\pi x_i)\Big) \;+\; 20 + e$
Griewank	$1 + \frac{1}{4000}\sum_{i=1}^{d} x_i^2 \;-\; \prod_{i=1}^{d} \cos\Big(\frac{x_i}{\sqrt{i}}\Big)$
Schwefel 2.26	$418.9829\,d \;-\; \sum_{i=1}^{d} x_i \,\sin\big(\sqrt{\lvert x_i\rvert}\big)$
Lévy (N.13)	*Let* $w_i = 1 + \frac{x_i - 1}{4}$. *Then* $\sin^2(\pi w_1) \;+\; \sum_{i=1}^{d-1} (w_i - 1)^2\big[1 + 10\sin^2(\pi w_i + 1)\big] \;+\; (w_d - 1)^2\big[1 + \sin^2(2\pi w_d)\big]$
Styblinski–Tang	$\frac{1}{2}\sum_{i=1}^{d} \big(x_i^4 - 16x_i^2 + 5x_i\big)$
Zakharov	$\sum_{i=1}^{d} x_i^2 \;+\; \Big(\sum_{i=1}^{d} \frac{i}{2} x_i\Big)^2 \;+\; \Big(\sum_{i=1}^{d} \frac{i}{2} x_i\Big)^4$
Dixon–Price	$(x_1 - 1)^2 \;+\; \sum_{i=2}^{d} i\,\big(2x_i^2 - x_{i-1}\big)^2$
Powell Sum†	$\sum_{i=1}^{d} \lvert x_i\rvert^{\,i+1}$
Trid	$\sum_{i=1}^{d} (x_i - 1)^2 \;-\; \sum_{i=2}^{d} x_i x_{i-1}$
Sum of Different Powers†	$\sum_{i=1}^{d} \lvert x_i\rvert^{\,i+1}$
Qing	$\sum_{i=1}^{d} \big(x_i^2 - i\big)^2$
Salomon	$1 - \cos(2\pi r) \;+\; 0.1\,r$
Alpine 1	$\sum_{i=1}^{d} \big\lvert\, x_i \sin x_i + 0.1\, x_i \,\big\rvert$
Exponential	$-\exp\Big(-\frac{1}{2}\sum_{i=1}^{d} x_i^2\Big)$
Schaffer N.2 (gen.)	$\sum_{i=1}^{d-1} \left\{ 0.5 + \frac{\sin^2(x_i^2 - x_{i+1}^2) - 0.5}{\big(1 + 0.001(x_i^2 + x_{i+1}^2)\big)^2} \right\}$
Quartic (weighted)	$\sum_{i=1}^{d} i\,x_i^4$
SumSquares (weighted)	$\sum_{i=1}^{d} i\,x_i^2$