DISTANCE-AWARE ADAPTIVE SCALING FOR STOCHASTIC OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The ideal learning rate for each model parameter depends on both the magnitude of its stochastic gradients and the distance from its initial value to an optimal value. Given comparable gradient histories, a parameter that starts farther from its optimal value should take larger steps. However, standard adaptive methods such as AdaGrad and Adam adapt to stochastic gradient magnitudes without explicitly accounting for these *coordinatewise distances*. To address this limitation, we introduce **distance-aware adaptive scaling**, which uses the largest observed parameter displacement from a reference point as a surrogate for unavailable distance information. Coordinates with larger displacement receive larger relative updates, while those with small displacement retain approximately the original scaling. We integrate this mechanism into AdaGrad and AdamW, yielding DA-AdaGrad and DA-AdamW, both of which recover their base optimizers when distance-aware scaling is disabled. For DA-AdaGrad, we derive an AdaGrad-style regret bound and identify conditions under which distance-aware scaling improves the regret-bound expression on a fixed trajectory. Experiments on synthetic and real-world datasets demonstrate the effectiveness of the proposed approach.

1 INTRODUCTION

Adaptive optimization methods have achieved broad success in machine learning by adjusting coordinatewise learning rates to stochastic gradient history. AdaGrad scales each coordinate's update by the inverse square root of its accumulated squared gradients, making it particularly effective for sparse, high-dimensional problems (Duchi et al., 2011). RMSProp and Adam use exponential moving averages to adapt to recent gradient statistics (Kingma & Ba, 2015; Tieleman & Hinton, 2012), while AdamW incorporates decoupled weight decay (Loshchilov & Hutter, 2019). By accommodating differences in stochastic gradient magnitudes across parameters, these methods support the training of models ranging from sparse predictors to large neural networks.

Despite this success, adapting to stochastic gradient magnitudes alone does not fully capture the coordinatewise scaling suggested by optimization theory. A standard regret-bound expression for a fixed diagonal scaling balances two quantities: accumulated squared gradients and the distance from initialization to an optimal solution. Minimizing this expression yields a coordinatewise learning-rate scale proportional to the initial distance and inversely proportional to the square root of accumulated squared gradients. Thus, given comparable gradient histories, a parameter that starts farther from its optimal value should take larger steps. Standard adaptive methods such as AdaGrad and Adam adapt to gradient magnitudes without explicitly incorporating these coordinatewise distances. Their relative learning rates can therefore be poorly matched to the different distances parameters need to cover. Although these distances are unavailable during training, treating them implicitly as a common scale leaves potentially useful information unexplored. Several recent methods have incorporated distance information into step-size adaptation (Defazio & Mishchenko, 2023; Ivgi et al., 2023; Mishchenko & Defazio, 2024). However, their use of aggregate distance information does not explicitly capture differences in how far *individual coordinates must travel*. This motivates using **coordinatewise displacement** to adjust their relative adaptive scales.

We introduce *distance-aware adaptive scaling*, which uses the largest observed parameter displacement from a reference point as a surrogate for unavailable distance information. The intuition is that, when iterates move toward a solution, coordinates that must cover larger distances can exhibit

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

larger displacements. This motivates supplementing gradient-based adaptation with a coordinatewise displacement factor: coordinates with larger displacement receive larger relative updates, while those with small displacement retain approximately the original scaling. Specifically, we divide the adaptive second-moment statistic by a factor $1 + \gamma(\delta_{t,i}^{\max})^2$, where $\delta_{t,i}^{\max}$ is the largest displacement observed so far and $\gamma \geq 0$ controls the strength of the adjustment. We instantiate this principle in DA-AdaGrad using displacement from initialization and in DA-AdamW using an exponential moving-average reference point. Both recover their base optimizers exactly when $\gamma = 0$. For DA-AdaGrad, we derive an AdaGrad-style regret bound. The analysis suggests benefits when larger observed displacement aligns with greater coordinatewise distance to a solution. Experiments on synthetic problems and real-world learning tasks support the effectiveness of this approach.

We summarize our contributions below.

- We introduce a coordinatewise adjustment that complements gradient history with observed parameter displacement, motivated by the dependence of regret bounds on both gradients and distances. We propose DA-AdaGrad that uses running max displacement from initialization, and establish an AdaGrad-style regret bound that recovers the corresponding AdaGrad expression at $\gamma = 0$. We also analyze when distance-aware scaling improves the regret bound on a fixed trajectory.
- We also introduce this mechanism to other adaptive methods including AdamW and AMSGrad, yielding DA-AdamW and DA-AMSGrad. We use an exponential moving-average anchor with the same decay parameter as its second-moment estimate for the displacement. Both retain the coordinatewise maximum displacement observed so far and recover their base optimizers when the adjustment is disabled.
- We conduct controlled stochastic squared-loss experiments to examine coordinatewise distance heterogeneity and magnitude at fixed curvature, and curvature at fixed target distances. On six nonconvex LIBSVM classification problems, DA-AdaGrad gives the lowest mean final objective among the compared methods; we also compare AdamW and DA-AdamW on image classification and machine translation.

2 Related Work

Several methods use distance information to replace manual selection of a global learning-rate scale. DoG selects a scalar step size from the largest Euclidean displacement from initialization and accumulated squared gradient norms; its layerwise variant applies the same rule separately within each parameter group (Ivgi et al., 2023). AdaGrad++ and Adam++ combine a shared factor based on the largest root-mean-square displacement across coordinates with coordinatewise gradient normalization (Tao et al., 2026). D-Adaptation and Prodigy instead estimate a common initial distance to a solution from gradients and parameter movement (Defazio & Mishchenko, 2023; Mishchenko & Defazio, 2024). Their adaptive variants allow different effective learning rates across coordinates, but the distance estimate remains shared. Our approach uses distance information for a different purpose: adjusting the relative scaling of individual coordinates while retaining a tunable global learning rate.

These methods also differ in how distance enters the gradient history. DoWG uses squared running distance both in the step-size numerator and to weight accumulated squared gradient norms (Khaled et al., 2023). In Prodigy's Adam variant, the evolving distance estimate weights gradients before they enter the first- and second-moment recursions (Mishchenko & Defazio, 2024). Thus, Prodigy's replacement of the global learning-rate scale is not simply a scalar schedule applied to an otherwise unchanged Adam update. DA-AdaGrad and DA-AdamW retain their base optimizers' gradient accumulation rules and apply displacement scaling when forming the update.

Hot DoG provides a closer coordinatewise comparison. Developed for optimizing Bayesian coreset weights, it combines Adam-like gradient moments with a separate displacement statistic for each coordinate (Chen et al., 2025). Its distance recursion is smoothed and nondecreasing relative to initialization, and its update includes an additional inverse-square-root dependence on the number of optimization steps. Our construction instead introduces a correction that recovers the base optimizer when $\gamma = 0$. DA-AdaGrad measures displacement from initialization, whereas DA-AdamW measures it from an EMA anchor. Both retain the largest coordinatewise displacement observed so far.

AdaDelta also uses coordinatewise parameter movement, but scales gradients by the ratio of moving root-mean-square estimates of previous updates and gradients (Zeiler, 2012). Since each update is

squared before averaging, opposite-signed steps both contribute positively to its movement statistic. Our displacement statistics combine signed updates before taking their magnitudes, so opposing movement can cancel within each displacement measurement. We then retain the coordinatewise maximum of these measurements; subsequent cancellation does not reduce the stored maximum.

COCOB adapts coordinates through separate coin-betting procedures (Orabona & Tommasi, 2017). Its updates use a reward statistic formed from products of gradients and displacement, and its convergence bound depends on individual coordinate distances. This reward reflects the alignment of displacement with descent directions, rather than displacement magnitude alone. Our construction instead uses squared displacement to adjust an existing adaptive denominator, without replacing the update by a coin-betting rule.

3 PRELIMINARIES

For differentiable losses $f_1, \ldots, f_T$, let $w_t \in \mathbb{R}^d$ denote the parameter vector at iteration t, and $g_{t,i}$ the ith component of the gradient of f_t at w_t. With $A_{t,i} = \sum_{\tau=1}^{t} g_{\tau,i}^2$, the standard AdaGrad update is

$$w_{t+1,i} = w_{t,i} - \eta \frac{g_{t,i}}{\sqrt{\varepsilon + A_{t,i}}}. \tag{1}$$

A standard regret bound motivates AdaGrad's coordinatewise gradient-history adaptation. Assume the losses $f_1, \ldots, f_T$ are convex. For a comparator $w^\star$, define $d_i := |w_{1,i} - w_i^\star|$. A fixed diagonal scaling with positive entries s_i, applied through $w_{t+1,i} = w_{t,i} - \eta g_{t,i}/s_i$, gives the standard upper bound (Duchi et al., 2011)

$$\sum_{t=1}^{T} \big(f_t(w_t) - f_t(w^\star)\big) \le \frac{1}{2\eta} \sum_{i=1}^{d} d_i^2 s_i + \frac{\eta}{2} \sum_{i=1}^{d} \frac{A_{T,i}}{s_i}. \tag{2}$$

Here $\eta > 0$ is the global learning rate, and η/s_i is the learning rate in coordinate i. Appendix A derives the bound. AdaGrad (Duchi et al., 2011) can be obtained by constraining the sum of s_i to control the distance term and replacing the full-horizon accumulator $A_{T,i}$ with the observed accumulator $A_{t,i}$. This gives the relative scaling $s_{t,i} \propto \sqrt{A_{t,i}}$, with $\varepsilon > 0$ added as a numerical offset. Coordinates with larger accumulators receive smaller effective learning rates, while those with smaller accumulators retain larger ones. This scaling accommodates differences in gradient magnitude and update frequency, making AdaGrad well suited to sparse, high-dimensional problems. Appendix B.1 gives the derivation.

4 DISTANCE-AWARE ADAPTIVE SCALING

In this section, we first present the proposed Distance-Aware AdaGrad variant and then discuss how to extend to other adaptive methods.

4.1 INCORPORATING COORDINATEWISE DISTANCE

Accumulated squared gradients need not reflect the distances parameters must cover. Coordinates with comparable squared-gradient histories receive similar learning rates under AdaGrad, even when their updates differ in directional persistence. The tradeoff in (2) makes the role of distance explicit: reducing the $\frac{A_{T,i}}{s_i}$ term increases the corresponding $d_i^2 s_i$ term. Minimizing this expression with access to both $A_{T,i}$ and d_i, and then replacing the full-horizon accumulator by its running estimate, gives the coordinate scaling $s_{t,i} \propto \sqrt{\frac{A_{t,i}}{d_i^2}}$.

AdaGrad uses $A_{t,i} = \sum_{\tau=1}^{t} g_{\tau,i}^2$ as a surrogate for $A_{T,i} = \sum_{\tau=1}^{T} g_{\tau,i}^2$. Analogously, defining displacement at time t as $\delta_{t,i} = |w_{t,i} - w_{1,i}|$, we use the observed running maximum displacement at time t, $\delta_{t,i}^{\max} = \max_{1 \le \tau \le t} \delta_{\tau,i}$, as a surrogate for d_i.

Algorithm 1 Distance-Aware AdaGrad

Input: Initial point w_1, horizon T, learning rate $\eta > 0$, displacement multiplier $\gamma \geq 0$, numerical offset $\varepsilon > 0$

$A_0 \leftarrow 0$, $\delta_0^{\max} \leftarrow 0$
for $t = 1, \ldots, T$ **do**
  Compute the gradient g_t at w_t
  $A_t \leftarrow A_{t-1} + g_t^2$
  $\delta_{t,i}^{\max} \leftarrow \max\{\delta_{t-1,i}^{\max}, |w_{t,i} - w_{1,i}|\}$
  $b_t \leftarrow 1 + \gamma(\delta_t^{\max})^2$ ◇ element-wise square
  $w_{t+1} \leftarrow w_t - \eta g_t / \sqrt{\varepsilon + A_t / b_t}$ ◇ element-wise division
end for
return w_{T+1}

Substituting this displacement for the unavailable d_i gives

$$s_{t,i} \propto \sqrt{\frac{A_{t,i}}{(\delta_{t,i}^{\max})^2}}.$$

However, directly using this scaling can suppress the effective learning rate before substantial displacement has accumulated. To encourage effective movements at the beginning stage, we add a regularizing term in s_i and solve

$$s_{t,i} = \operatorname*{argmin}_{\sigma>0} \Big\{ \max_{1\leq\tau\leq t} (w_{\tau,i} - w_{1,i})^2 \sigma + \sigma^{-1} \sum_{\tau=1}^{t} g_{\tau,i}^2 + \lambda\sigma \Big\}$$

The first-order optimality condition gives

$$s_{t,i} \propto \sqrt{\frac{A_{t,i}}{\lambda + (\delta_{t,i}^{\max})^2}}, \qquad \lambda > 0.$$

To make more explicit the roles of λ and η when tuning, observe that

$$\eta \left(\frac{A_{t,i}}{\lambda + (\delta_{t,i}^{\max})^2} \right)^{-1/2} = \eta\sqrt{\lambda} \left(\frac{A_{t,i}}{1 + \lambda^{-1}(\delta_{t,i}^{\max})^2} \right)^{-1/2}.$$

We may therefore absorb $\sqrt{\lambda}$ into the learning rate and define $\gamma = \lambda^{-1}$. Adding the numerical offset $\varepsilon > 0$ gives the final update

$$w_{t+1,i} = w_{t,i} - \eta \left(\varepsilon + \frac{A_{t,i}}{1 + \gamma(\delta_{t,i}^{\max})^2} \right)^{-1/2} g_{t,i}, \qquad \gamma \geq 0.$$

The detailed steps of DA-AdaGrad are presented in Algorithm 2. It is notable that setting $\gamma = 0$ recovers AdaGrad exactly. When displacement stops growing, $1 + \gamma(\delta_{t,i}^{\max})^2$ becomes constant; further growth of $A_{t,i}$ then restores the usual $A_{t,i}^{-1/2}$ step decay, up to a constant factor.

For a fixed gradient history and learning rate, larger displacement increases the effective learning rate relative to AdaGrad. This motivates considering $\eta_\gamma \leq \eta_0$ when tuning, where η_0 denotes the optimal AdaGrad learning rate; this is a tuning heuristic rather than a guaranteed relation between the optimal rates. Appendix B gives the complete derivation.

4.2 Extensions to Adam and AMSGrad

The same distance correction can be applied to adaptive methods that use exponential moving averages. Adam (Kingma & Ba, 2015) replaces AdaGrad's cumulative sum of squared gradients by an exponential moving average and also smooths the gradient in the numerator. Its second-moment estimate and bias correction are

$$v_{t,i} = (1-\beta_2)\sum_{s=1}^{t}\beta_2^{t-s}g_{s,i}^2, \qquad \widehat{v}_{t,i} = \frac{v_{t,i}}{1-\beta_2^t}.$$

AMSGrad (Reddi et al., 2018) retains a coordinatewise running maximum of v_t, preserving the influence of earlier large squared gradients as the moving average decreases.

We likewise replace the signed displacement sum $w_t - w_1 = \sum_{s=1}^{t-1}(w_{s+1} - w_s)$ by an exponential moving average with the same decay β_2:

$$z_t = (1-\beta_2)\sum_{s=1}^{t-1}\beta_2^{t-1-s}(w_{s+1}-w_s) = (1-\beta_2)(w_t - u_{t-1}),$$
$$u_t = \beta_2 u_{t-1} + (1-\beta_2)w_t, \qquad u_0 = w_1.$$

The factor $1-\beta_2$ in the displacement EMA can be absorbed into its coefficient: setting $\gamma = \widetilde{\gamma}(1-\beta_2)^2$ gives $1 + \widetilde{\gamma} z_{t,i}^2 = 1 + \gamma\delta_{t,i}^2$, where $\delta_{t,i} = |w_{t,i} - u_{t-1,i}|$. Thus both accumulators use exponential averaging. The shared β_2 discounts past movement and squared gradients at the same rate, giving the two estimates a common time scale.

We also retain the largest observed displacement from the moving anchor:

$$\delta_{t,i}^{\max} = \max\{\delta_{t-1,i}^{\max}, \delta_{t,i}\}, \qquad \delta_0^{\max} = 0.$$

As in DA-AdaGrad, the stored maximum prevents the displacement correction from shrinking when recent movement decreases under a decaying learning-rate schedule.

Writing V_t for the base optimizer's second-moment estimate, the adaptive denominator becomes

$$\sqrt{V_t/(1+\gamma(\delta_t^{\max})^2)} + \varepsilon.$$

For Adam, $V_t = \widehat{v}_t$; for AMSGrad, we use $V_t = v_t^{\max}/(1-\beta_2^t)$, where $v_t^{\max} = \max\{v_{t-1}^{\max}, v_t\}$ and $v_0^{\max} = 0$. Both methods retain Adam's bias-corrected first moment in the numerator and can incorporate decoupled weight decay as in AdamW (Loshchilov & Hutter, 2019). Just as with AdaGrad, setting $\gamma = 0$ recovers the respective base optimizer exactly. Appendix B gives the complete updates, and Appendix G proves the identity above. The analysis below concerns DA-AdaGrad; the moving-anchor extensions are evaluated empirically.

5 Theoretical Results

We present a regret bound of DA-AdaGrad in Algorithm 1 so that it can recover AdaGrad's regret bound when $\gamma = 0$. Let $f_1, \ldots, f_T$ be convex on an axis-aligned box, let $w^\star$ be a comparator in that box, and project each coordinate of the update back onto the box. We take $\varepsilon = 0$, with zero update when the accumulator is zero; Appendix C gives the result for $\varepsilon > 0$. Define

$$\begin{aligned} a_{t,i} &:= \sqrt{A_{t,i}}, & \Delta a_{t,i} &:= a_{t,i} - a_{t-1,i}, \\ D_i &:= \max_{1\le t\le T+1} |w_{t,i} - w_i^\star|, & \mathcal{I} &:= \{(t,i) : g_{t,i} \neq 0\}, \end{aligned} \tag{3}$$

With $a_i := a_{T,i}$, standard diagonal AdaGrad satisfies (Duchi et al., 2011)

$$R_T := \sum_{t=1}^{T}\big(f_t(w_t) - f_t(w^\star)\big) \le \frac{1}{2\eta}\sum_i D_i^2 a_i + \eta\sum_i a_i. \tag{4}$$

Minimizing the right-hand side over η gives

$$U_0^\star = \sqrt{2\left(\sum_i D_i^2 a_i\right)\left(\sum_i a_i\right)}.$$

Define the distance and gradient coefficients

$$P_\gamma := \sum_{(t,i)\in\mathcal{I}} \frac{D_i^2 \Delta a_{t,i}}{\sqrt{1+\gamma(\delta_{t,i}^{\max})^2}}, \qquad Q_\gamma := \sum_{(t,i)\in\mathcal{I}} \frac{g_{t,i}^2}{a_{t,i}} \sqrt{1+\gamma(\delta_{t,i}^{\max})^2}. \tag{5}$$

Theorem 5.1 (Regret bound for DA-AdaGrad)**.** *For every $\eta > 0$, the projected DA-AdaGrad update satisfies*

$$R_T := \sum_{t=1}^{T} \big(f_t(w_t) - f_t(w^\star)\big) \le \frac{P_\gamma}{2\eta} + \frac{\eta Q_\gamma}{2}. \tag{6}$$

Appendix D provides the complete proof.

Minimizing the two-term expression over η gives $B_\gamma^\star = \sqrt{P_\gamma Q_\gamma}$. At $\gamma = 0$, $P_0 = \sum_i D_i^2 a_i$ and $Q_0 \le 2\sum_i a_i$, so (6) implies the AdaGrad bound (4), with $B_0^\star \le U_0^\star$.

For $\gamma > 0$, the correction gives $P_\gamma \le P_0$ and $Q_\gamma \ge Q_0$. Improving the bound therefore requires the decrease in P_γ to outweigh the increase in Q_γ.

When does distance-aware scaling improve the bound? Our analysis in Appendix C.2 identifies when distance-aware scaling can tighten the AdaGrad bound. For a fixed trajectory in the limit $\varepsilon \to 0$, let $B_\gamma^\star$ denote the bound expression minimized over the global learning rate. Let M_P and M_Q be weighted averages of the squared running displacement, with weights proportional to the contributions to distance-dependent and squared-gradient terms, respectively. We show that

$$\frac{B_\gamma^\star}{B_0^\star} = 1 - \frac{\gamma}{4}(M_P - M_Q) + O(\gamma^2).$$

Thus, $M_P > M_Q$ guarantees a strictly tighter bound expression for sufficiently small positive γ.

This condition suggests a connection to the dispersion of coordinatewise distances. When we have $D_i \approx d_i$, comparable gradient contributions across coordinates, and temporally averaged squared displacements scaling approximately as $c\,d_i^2$ for a common $c > 0$, then (see Appendix C.3):

$$M_P - M_Q \approx c\,\mathbb{E}_i[d_i^2]\ \mathrm{CV}(d_i^2)^2,$$

where $\mathrm{CV}(d_i^2) := \mathrm{SD}(d_i^2)/\mathbb{E}[d_i^2]$. Under these approximations, greater distance dispersion strengthens the first-order improvement at fixed c and total squared distance. Section 6.1 examines this prediction by varying $\mathrm{CV}(d_i^2)$ while holding curvature and $\sum_i d_i^2$ fixed. A second sweep varies $\sum_i d_i^2$ while holding both curvature and $\mathrm{CV}(d_i^2)$ fixed. Appendix F.3 separately varies curvature while holding all initial coordinate distances fixed.

6 Experiments

We evaluate distance-aware scaling on controlled squared-loss problems, nonconvex classification objectives, image classification, and machine translation. Appendix F gives the full experimental protocols.

6.1 Controlled Squared-Loss Experiments

We compare AdaGrad with DA-AdaGrad in three synthetic squared-loss experiments, tuning the learning rate separately for each method and setting. We present the two distance sweeps here and the curvature sweep in Appendix F.3. We measure progress by the objective gap divided by its initial value. Appendix F gives the full protocol.

The first experiment uses the unpenalized squared loss

$$F(w) = \frac{1}{2N}\sum_{j=1}^{N} \|w - Y_j\|_2^2.$$

We set $Y_j = w^\star + \xi_j$ with $\dim(w^*) = 128$, using the same noise vectors with zero empirical mean in every condition. Thus $w^\star$ is the minimizer and $\nabla^2 F = I_{128}$, so the strong-convexity parameter

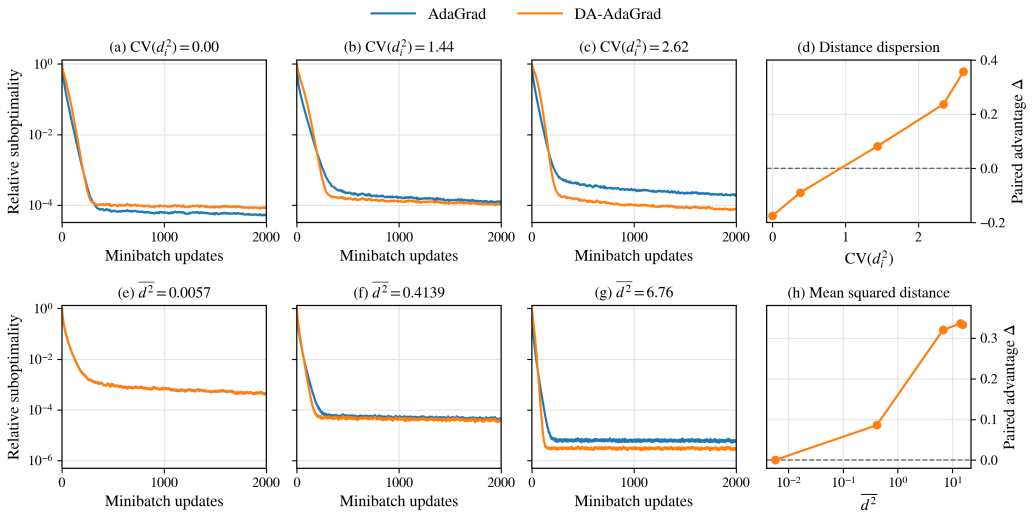


Figure 1: **Controlled squared-loss experiments.** Top: increasing distance dispersion at fixed curvature and mean squared distance. Bottom: increasing mean squared distance $\overline{d^2} = m^{-1}\sum_i d_i^2$ at fixed curvature and $\mathrm{CV}(d_i^2)$. Trajectories show median relative suboptimality over 20 runs at three representative settings; bands and bars show interquartile ranges. Right: median advantage Δ across all five settings, where Δ is the time-averaged $\log_{10}$ ratio of AdaGrad's relative suboptimality to DA-AdaGrad's; positive values favor DA-AdaGrad.

remains 1 regardless of the target. From initialization $w_1 = 0$, we vary $d_i = |w_i^\star|$ by assigning an increasing fraction r of the total squared distance to a fixed set G of 16 coordinates:

$$d_i^2 = \begin{cases} 8r, & i \in G, \\ 8(1-r)/7, & i \notin G. \end{cases}$$

The mean squared distance is therefore $(16 \cdot 8r + 112 \cdot 8(1-r)/7)/128 = 1$ in every condition, while $\mathrm{CV}(d_i^2) = (8r-1)/\sqrt{7}$ increases with $r \geq 1/8$. Increasing r moves the same 16 target coordinates farther from initialization and the remaining coordinates closer, preserving the total squared distance and initial objective gap. We also reuse the coordinate signs and minibatch streams. DA-AdaGrad's advantage increases with this dispersion (Figure 1a–d); Appendix F.1 gives the full construction and results. These results show that the larger $\mathrm{CV}(d_i^2)$ corresponds to larger improvement of DA-AdaGrad.

The second experiment scales the minimizer while holding curvature fixed,

$$F_a(w) = \frac{1}{2n}\|Xw - (Xau + e)\|_2^2, \qquad X^\top e = 0, \qquad \frac{1}{m}\sum_i u_i^2 = 1.$$

Here $w^\star = au$, so the mean squared distance is a^2, while $\mathrm{CV}(d_i^2)$ and the Hessian $H = X^\top X/n$ remain fixed. Figure 1e–h shows that DA-AdaGrad's advantage grows from approximately zero at the smallest distance to about 0.33 at the largest distances. This supports the distance-magnitude hypothesis without changing curvature. Appendix F.2 gives the construction and controls; Appendix F.3 varies curvature at fixed distances.

6.2 Nonconvex Classification on LIBSVM Datasets

We evaluate linear classifiers on Covtype, epsilon, and RCV1 from the LIBSVM dataset collection (LIBSVM Data, n.d.). For labels $y_j \in \{-1, 1\}$, we minimize

$$F(w) = \frac{1}{n}\sum_{j=1}^{n} \ell(y_j x_j^\top w) + \rho\|w\|_1, \qquad \text{where} \quad \ell_{\mathrm{sigmoid}}(z) = \frac{1}{1+e^z},$$
$$\ell_{\mathrm{ramp}}(z) = \min\{2, \max\{0, 1-z\}\}.$$

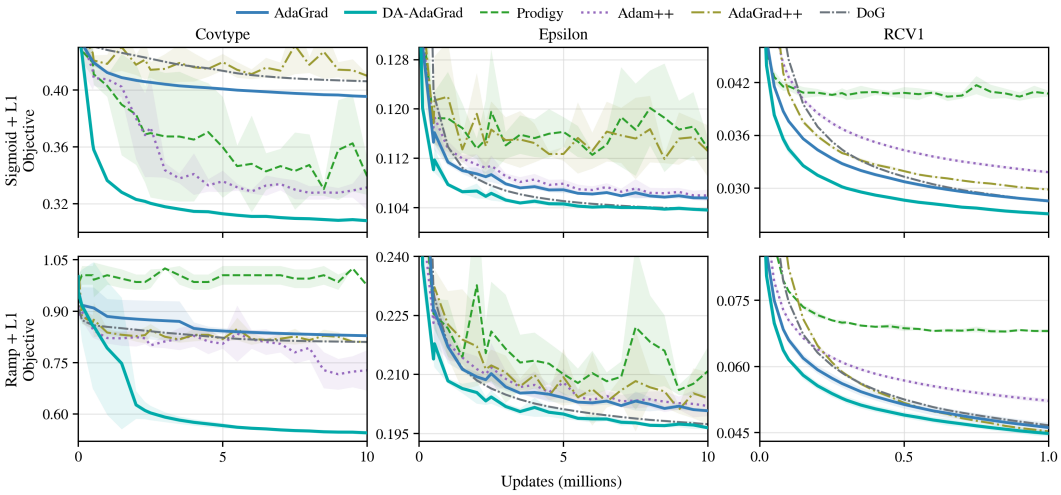


Figure 2: **Nonconvex classification with L_1 regularization.** Sigmoid (top) and ramp (bottom) losses on Covtype, epsilon, and RCV1 (left to right). Lines and bands show mean training objective and one standard deviation over five runs. Table 8 in Appendix F.4 gives the final results.

These losses combine saturation with an L_1 penalty that can favor sparse solutions (Huang et al., 2014). From zero initialization, a sparse nonzero solution has heterogeneous coordinate distances, connecting these problems to the distance-dispersion motivation of our analysis. The penalty adds no quadratic curvature. These experiments assess whether the adjustment remains useful beyond the convex setting covered by our theory. Appendix F.4 gives the experimental protocol and full numerical results. The results are presented in Figure 2, where we compare DA-AdaGrad with 5 baselines including AdaGrad, DoG, Prodigy, Adam++, AdaGrad++. The results demonstrate superior performance of DA-AdaGrad.

6.3 Image Classification

We train ResNet-18 (He et al., 2016) on CIFAR-10 and CIFAR-100 (Krizhevsky, 2009) for 50 epochs, and ViT-S/16 (Dosovitskiy et al., 2021; Touvron et al., 2021) on ImageNet-100 (Deng et al., 2009) for 300 epochs. For this deep learning experiment, we compare DA-AdamW with AdamW and for minimizing multiclass cross-entropy loss. Table 1 reports the best accuracy attained during each run. Figure 3 shows the ImageNet-100 validation trajectories from epochs 100–300. Appendix F.5 gives the settings. DA-AdamW attains higher mean best accuracy than AdamW on all three datasets.

Table 1: **Best image-classification accuracy (%).** Mean $\pm$ sample standard deviation of the highest accuracy attained in each run. Training lasts 50 epochs for CIFAR-10 and CIFAR-100 and 300 epochs for ImageNet-100, with 25, ten, and three seeds per method, respectively. CIFAR columns use test accuracy; ImageNet-100 uses validation accuracy.

Method	CIFAR-10 ResNet-18	CIFAR-100 ResNet-18	ImageNet-100 ViT-S/16
AdamW	92.95 ± 0.17	73.58 ± 0.34	82.45 ± 0.36
DA-AdamW	$\mathbf{93.26 \pm 0.12}$	$\mathbf{74.02 \pm 0.38}$	$\mathbf{82.86 \pm 0.26}$

6.4 Machine Translation

We train an encoder–decoder Transformer (Vaswani et al., 2017) on IWSLT14 German–English (Cettolo et al., 2014) for 20 epochs over six paired seeds, minimizing token-level cross-entropy with label smoothing. Figure 4 compares test BLEU (Papineni et al., 2002) and negative log-likelihood (NLL), evaluated at the checkpoint with the lowest validation NLL in each run. Appendix F.6 gives the settings. It again demonstrates the improvements of DA-AdamW over AdamW.

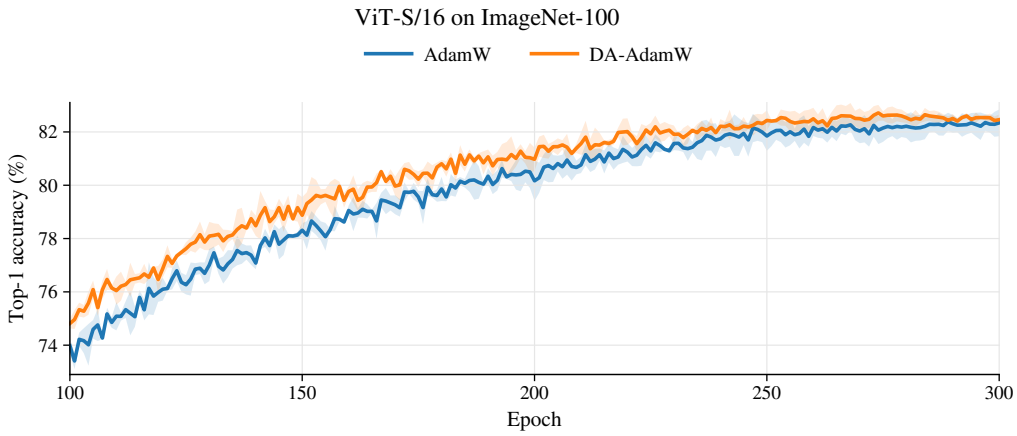


Figure 3: **ViT-S/16 on ImageNet-100.** Mean validation accuracy from epochs 100–300 over three paired seeds. Shaded bands show one sample standard deviation.

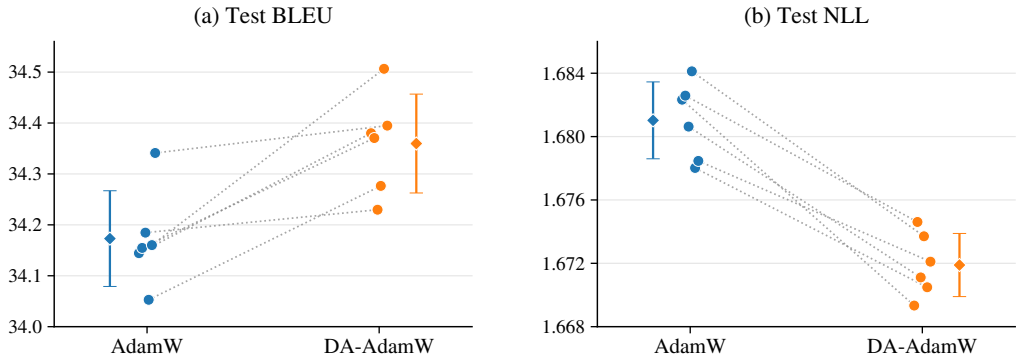


Figure 4: **IWSLT14 German–English translation.** Test BLEU (higher is better) and NLL (lower is better) after 20 training epochs, evaluated at the checkpoint selected by validation NLL. Dotted lines connect matching seeds; DA-AdamW improves both metrics in all six pairs. Diamonds and bars show the mean and one sample standard deviation.

7 Conclusion

DA-AdaGrad uses observed displacement to adjust relative coordinatewise step sizes and recovers AdaGrad exactly at $\gamma = 0$. Our analysis identifies when this adjustment lowers the regret-bound expression on a fixed path, even after optimizing the global learning rate. The controlled experiments show a larger advantage when a larger share of a fixed total squared distance is assigned to the same subset of coordinates, and when the overall distance to the minimizer increases at fixed curvature. A separate curvature sweep preserves all initial coordinate distances and finds no monotone relation between strong convexity and the advantage.

The applications extend this comparison to nonconvex objectives and neural-network training. On LIBSVM classification problems with sigmoid and ramp losses and L_1 regularization, DA-AdaGrad attains the lowest mean final objective in all six comparisons. DA-AdamW also attains higher mean best accuracy in the image-classification experiments and higher test BLEU with lower NLL on IWSLT14 translation.

An important next step is to establish conditions under which DA-AdamW and DA-AMSGrad improve on their respective baselines. In particular, understanding how the anchor's memory should relate to gradient averaging and the learning-rate schedule could clarify when recent movement provides the most useful information for setting coordinatewise step sizes.

Reproducibility Statement

Section 4 and Appendix B give the optimizer updates and pseudocode. Appendices A, C, D, and E give the theoretical assumptions and proofs, and Appendix G proves the EMA-anchor identity. Appendix F describes the experimental objectives, data construction and splits, optimizer settings, evaluation protocols, and hyperparameter-selection procedures. Anonymized implementations of the proposed optimizers and code for all reported experiments are included in the supplementary material.

AI Use Statement

Generative AI tools were used to assist with manuscript restructuring, copyediting, literature searching, citation checking, and LaTeX formatting. The AI tools also assisted with the writing of experimental code as well as the refinement of mathematical arguments. The authors reviewed all AI-assisted text, math, and code and take responsibility for the final content.

Ethics Statement

This work develops and evaluates general-purpose optimization methods. We do not identify ethical concerns specific to the methods or experiments presented here. Their broader societal impacts depend on the applications in which they are used.

References

Peter Auer, Nicolò Cesa-Bianchi, and Claudio Gentile. Adaptive and self-confident on-line learning algorithms. *Journal of Computer and System Sciences*, 64(1):48–75, February 2002. ISSN 0022-0000. doi: 10.1006/jcss.2001.1795. URL http://dx.doi.org/10.1006/jcss.2001.1795.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, May 2003. ISSN 0167-6377. doi: 10.1016/S0167-6377(02)00231-6.

L.M. Bregman. The relaxation method of finding the common point of convex sets and its application to the solution of problems in convex programming. *USSR Computational Mathematics and Mathematical Physics*, 7(3):200–217, 1967. ISSN 0041-5553. doi: 10.1016/0041-5553(67)90040-7. URL http://dx.doi.org/10.1016/0041-5553(67)90040-7.

N. Cesa-Bianchi, A. Conconi, and C. Gentile. On the generalization ability of on-line learning algorithms. *IEEE Transactions on Information Theory*, 50(9):2050–2057, September 2004. ISSN 0018-9448. doi: 10.1109/tit.2004.833339. URL http://dx.doi.org/10.1109/TIT.2004.833339.

Mauro Cettolo, Jan Niehues, Sebastian Stüker, Luisa Bentivogli, and Marcello Federico. Report on the 11th IWSLT evaluation campaign. In Marcello Federico, Sebastian Stüker, and François Yvon (eds.), *Proceedings of the 11th International Workshop on Spoken Language Translation: Evaluation Campaign*, pp. 2–17, Lake Tahoe, California, December 4-5 2014. URL https://aclanthology.org/2014.iwslt-evaluation.1/.

Naitong Chen, Jonathan H. Huggins, and Trevor Campbell. Tuning-free coreset markov chain monte carlo via hot dog. In Silvia Chiappa and Sara Magliacane (eds.), *Proceedings of the Forty-first Conference on Uncertainty in Artificial Intelligence*, volume 286 of *Proceedings of Machine Learning Research*, pp. 647–672. PMLR, 21–25 Jul 2025. URL https://proceedings.mlr.press/v286/chen25d.html.

Aaron Defazio and Konstantin Mishchenko. Learning-rate-free learning by d-adaptation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7449–7479. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/defazio23a.html.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. IEEE, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(61):2121–2159, 2011. URL http://jmlr.org/papers/v12/duchi11a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Xiaolin Huang, Lei Shi, and Johan A.K. Suykens. Ramp loss linear programming support vector machine. *Journal of Machine Learning Research*, 15(64):2185–2211, 2014. URL http://jmlr.org/papers/v15/huang14a.html.

Maor Ivgi, Oliver Hinder, and Yair Carmon. DoG is SGD's best friend: A parameter-free dynamic step size schedule. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 14465–14499. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/ivgi23a.html.

Ahmed Khaled, Konstantin Mishchenko, and Chi Jin. Dowg unleashed: An efficient universal parameter-free gradient descent method. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 6748–6769. Curran Associates, Inc., 2023. doi: 10.52202/075280-0296. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/15ce36d35622f126f38e90167de1a350-Paper-Conference.pdf.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

LIBSVM Data. LIBSVM Data: Classification (Binary Class). National Taiwan University dataset repository, n.d. URL https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html. Accessed September 25, 2026.

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

Konstantin Mishchenko and Aaron Defazio. Prodigy: An expeditiously adaptive parameter-free learner. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35779–35804. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/mishchenko24a.html.

A. S. Nemirovsky and D. B. Yudin. *Problem Complexity and Method Efficiency in Optimization*. Wiley-Interscience Series in Discrete Mathematics. John Wiley & Sons, Chichester, 1983. ISBN 0-471-10345-4. URL https://www2.isye.gatech.edu/~nemirovs/Nemirovskii_Yudin_1983.pdf. Translated by E. R. Dawson.

Francesco Orabona and Tatiana Tommasi. Training deep networks without learning rates through coin betting. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/7c82fab8c8f89124e2ce92984e04fb40-Paper.pdf.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318, Philadelphia, Pennsylvania, USA, July 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Matt Post. A call for clarity in reporting BLEU scores. In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Lucia Specia, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Third Conference on Machine Translation: Research Papers*, pp. 186–191, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/W18-6319. URL https://aclanthology.org/W18-6319/.

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

Yuanzhe Tao, Yifeng Liu, Huizhuo Yuan, Xun Zhou, Yuan Cao, and Quanquan Gu. Towards simple and provable parameter-free adaptive gradient methods, 2026. URL https://arxiv.org/abs/2412.19444v2.

Tijmen Tieleman and Geoffrey Hinton. RMSProp: Divide the gradient by a running average of its recent magnitude. Unpublished method described in Geoffrey Hinton's *Neural Networks for Machine Learning*, Coursera, Lecture 6e, 2012. URL https://www.cs.toronto.edu/~hinton/coursera/lecture6/lec6.pdf.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Tianbao Yang. Compositional Optimization for Advanced Machine Learning, 2026. URL https://opt4ml.org/assets/compositional-optimization-for-adv-ML.pdf. Online book manuscript, January 2026 version.

Matthew D. Zeiler. Adadelta: An adaptive learning rate method, 2012. URL https://arxiv.org/abs/1212.5701.

Martin Zinkevich. Online Convex Programming and Generalized Infinitesimal Gradient Ascent. In Tom Fawcett and Nina Mishra (eds.), *Proceedings of the Twentieth International Conference on Machine Learning*, pp. 928–936, Menlo Park, California, 2003. AAAI Press. ISBN 1-57735-189-4. URL https://cdn.aaai.org/ICML/2003/ICML03-120.pdf.

Appendix Contents

1 Introduction **1**

2 Related Work **2**

3 Preliminaries **3**

4 Distance-Aware Adaptive Scaling **3**

4.1 Incorporating coordinatewise distance 3

4.2 Extensions to Adam and AMSGrad 5

5 Theoretical Results **5**

6 Experiments **6**

6.1 Controlled Squared-Loss Experiments 6

6.2 Nonconvex Classification on LIBSVM Datasets 7

6.3 Image Classification 8

6.4 Machine Translation 8

7 Conclusion **9**

A Fixed-Metric Preliminaries **15**

A.1 A Fixed Diagonal Metric 15

B Derivation and Optimizer Updates **17**

B.1 The AdaGrad Scaling 17

B.2 Retaining the Displacement Term 17

B.3 Reparameterization and Behavior 18

B.4 Adam and AMSGrad Updates 19

B.5 Optimizer Algorithms 20

C Formal Theory Statements and Fixed-Path Analysis **21**

C.1 A Regret Bound for DA-AdaGrad 21

C.2 When Does $\gamma > 0$ Improve the Bound? 22

C.3 Connection to the dispersion of coordinatewise distances 23

C.4 Interpreting the Weights 26

D Proof of the DA-AdaGrad Regret Bound **28**

D.1 Control of the Positive Variation 28

D.2 Proof of the Regret Theorem 28

D.3 AdaGrad Recovery and the Fixed-Sequence Tradeoff 29

E Proofs for the Direct Comparison with AdaGrad **31**

E.1 Minimization over the Scalar Learning Rate 31

E.2 Exact Normalized Identity 31

E.3 Derivative at the AdaGrad Endpoint 32

F Experimental Details **32**

F.1 Distance Heterogeneity at Fixed Curvature and Total Distance 33

F.2 Distance Magnitude at Fixed Curvature 36

F.3 Curvature at Fixed Target Distances 38

F.4 Nonconvex Objectives on LIBSVM Datasets 39

F.5 Image Classification 41

F.6 Machine Translation 42

G Equivalence of the EMA Anchor and a Leaky Update Sum **43**

A Fixed-Metric Preliminaries

A.1 A Fixed Diagonal Metric

Let $\mathcal{W} \subseteq \mathbb{R}^d$ be a nonempty closed convex set, let $w_1, w^\star \in \mathcal{W}$, and let $f_1, \dots, f_T : \mathcal{W} \to \mathbb{R}$ be convex. Let $g_t \in \partial f_t(w_t)$ be a subgradient of f_t at w_t. For a fixed diagonal matrix

$$H := \operatorname{diag}(s_1, \dots, s_d), \qquad s_i > 0, \tag{7}$$

define

$$\begin{aligned} \|x\|_H &:= \sqrt{x^\top H x}, \\ \|g\|_{H,*} &:= \sup_{\|x\|_H \le 1} \langle g, x \rangle = \sqrt{g^\top H^{-1} g} = \|g\|_{H^{-1}}. \end{aligned} \tag{8}$$

Let

$$\psi_H(x) := \frac{1}{2}\|x\|_H^2. \tag{9}$$

For this quadratic function, the Bregman divergence (Bregman, 1967) is

$$D_{\psi_H}(x, y) := \psi_H(x) - \psi_H(y) - \langle \nabla \psi_H(y), x - y \rangle = \frac{1}{2}\|x - y\|_H^2. \tag{10}$$

Projection onto $\mathcal{W}$ in this norm is

$$\Pi_{\mathcal{W}}^H(z) := \arg\min_{w \in \mathcal{W}} \|w - z\|_H^2. \tag{11}$$

Consider the update

$$w_{t+1} = \Pi_{\mathcal{W}}^H\left(w_t - \eta H^{-1} g_t\right). \tag{12}$$

The potential ψ_H is 1-strongly convex with respect to $\|\cdot\|_H$; no strong convexity of the losses f_t is assumed. The standard fixed-metric mirror-descent argument (Beck & Teboulle, 2003; Nemirovsky & Yudin, 1983) gives the following bound (Duchi et al., 2011, Proposition 3):

$$R_T := \sum_{t=1}^{T}\big(f_t(w_t) - f_t(w^\star)\big) \le \frac{\|w_1 - w^\star\|_H^2}{2\eta} + \frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_{H,*}^2. \tag{13}$$

We use the regret form because Section 5 studies the same quantity. For example, if $f_t(w) = f(w; \zeta_t)$ for independent samples ζ_t, $\mathbb{E}[g_t \mid w_t] \in \partial F(w_t)$ for $F(w) := \mathbb{E}[f(w; \zeta)]$, and the required expectations are finite, then convexity and division by T give the corresponding expected bound for the averaged iterate $T^{-1}\sum_{t=1}^{T} w_t$ (Cesa-Bianchi et al., 2004).

For $t = 1, \dots, T$, define

$$A_{t,i} := \sum_{s=1}^{t} g_{s,i}^2, \qquad d_i := |w_{1,i} - w_i^\star|, \tag{14}$$

where d_i is the distance between w_1 and $w^\star$ in coordinate i. The right-hand side of (13) can then be written as a sum over coordinates:

$$\frac{1}{2\eta}\sum_{i=1}^{d} d_i^2 s_i + \frac{\eta}{2}\sum_{i=1}^{d}\frac{A_{T,i}}{s_i}. \tag{15}$$

Lemma A.1 (Dual norm of a positive diagonal metric)**.** *For every $H \succ 0$ and $g \in \mathbb{R}^d$,*

$$\sup_{\|x\|_H \le 1} \langle g, x \rangle = \sqrt{g^\top H^{-1} g}.$$

Proof. Let $y := H^{1/2}x$. Then $\|x\|_H = \|y\|_2$ and

$$\langle g, x \rangle = \langle H^{-1/2}g, y \rangle \le \|H^{-1/2}g\|_2 \|y\|_2 \le \sqrt{g^\top H^{-1} g}.$$

If $g \neq 0$, equality is attained by $y = H^{-1/2}g / \|H^{-1/2}g\|_2$ and $x = H^{-1/2}y$; if $g = 0$, both sides are zero. □

Proposition A.2 (Fixed-metric regret bound)**.** *The update in* (12) *satisfies*

$$\sum_{t=1}^{T}\big(f_t(w_t)-f_t(w^\star)\big)\le\frac{\|w_1-w^\star\|_H^2}{2\eta}+\frac{\eta}{2}\sum_{t=1}^{T}\|g_t\|_{H^{-1}}^2.$$

Proof. This is the metric version of the projected-gradient regret argument of Zinkevich (2003, Theorem 1). Let $z_t := w_t - \eta H^{-1} g_t$ and $w_{t+1} := \Pi_{\mathcal{W}}^H(z_t)$. For $w \in \mathcal{W}$, the optimality condition for the metric projection is

$$\langle H(z_t - w_{t+1}), w - w_{t+1}\rangle \le 0.$$

Taking $w = w^\star$ and expanding $\|z_t - w^\star\|_H^2$ give

$$\|w_{t+1} - w^\star\|_H^2 \le \|z_t - w^\star\|_H^2.$$

Consequently,

$$\|w_{t+1} - w^\star\|_H^2 \le \|w_t - w^\star\|_H^2 - 2\eta\langle g_t, w_t - w^\star\rangle + \eta^2\|g_t\|_{H^{-1}}^2.$$

Rearranging gives

$$\langle g_t, w_t - w^\star\rangle \le \frac{\|w_t - w^\star\|_H^2 - \|w_{t+1} - w^\star\|_H^2}{2\eta} + \frac{\eta}{2}\|g_t\|_{H^{-1}}^2.$$

By convexity, $f_t(w_t) - f_t(w^\star) \le \langle g_t, w_t - w^\star\rangle$. Summing over t telescopes the first term and proves the result. □

B Derivation and Optimizer Updates

We use the coordinatewise regret bound from Appendix A to recover the usual AdaGrad scaling and motivate the displacement substitution.

B.1 The AdaGrad Scaling

The usual AdaGrad derivation controls the first term in (15) by restricting the trace of H (Duchi et al., 2011, Eq. (12)); see also Yang (2026, Section 3.5). If, for some $C > 0$,

$$\sum_{i=1}^{d} s_i \le C, \tag{16}$$

then

$$\sum_{i=1}^{d} d_i^2 s_i \le \|w_1 - w^\star\|_\infty^2 \sum_{i=1}^{d} s_i \le C\|w_1 - w^\star\|_\infty^2. \tag{17}$$

With this bound, choosing the s_i amounts to solving

$$\min_{s_i>0} \left\{ \sum_{i=1}^{d} \frac{A_{T,i}}{s_i} : \sum_{i=1}^{d} s_i \le C \right\}. \tag{18}$$

Coordinates with zero accumulated squared gradient can be omitted from the scaling problem, so without loss of generality we restrict to coordinates with $A_{T,i} > 0$. The trace constraint is then active, and the Lagrangian first-order conditions give:

$$-\frac{A_{T,i}}{s_i^2} + \nu = 0, \qquad \sum_{i=1}^{d} s_i = C, \tag{19}$$

and hence

$$s_{T,i}^{\mathrm{tr}} = \frac{C\sqrt{A_{T,i}}}{\sum_{j=1}^{d} \sqrt{A_{T,j}}} \propto \sqrt{A_{T,i}}. \tag{20}$$

Applying the same calculation at time t, using only $g_1, \dots, g_t$, gives $s_{t,i}^{\mathrm{tr}} \propto \sqrt{A_{t,i}}$. AdaGrad uses this form for the next update, with the common factor absorbed into η. Adding $\varepsilon > 0$ for numerical stability, the unconstrained AdaGrad update is

$$\begin{aligned} A_{0,i} &:= 0, \\ A_{t,i} &= A_{t-1,i} + g_{t,i}^2, \\ w_{t+1,i} &= w_{t,i} - \eta \frac{g_{t,i}}{\sqrt{\varepsilon + A_{t,i}}}. \end{aligned} \tag{21}$$

The trace constraint is used only to derive the relative sizes of the s_i; AdaGrad itself does not keep $\sum_i s_i$ fixed.

B.2 Retaining the Displacement Term

The fixed-H bound depends on both the accumulated squared gradients and the unknown distances d_i. At time t, we do not know these distances, but we do know the largest displacement of each coordinate from its initial value:

$$\delta_{t,i}^{\max} := \max_{1\le s\le t} |w_{s,i} - w_{1,i}|. \tag{22}$$

This is not an estimate of the remaining distance to $w^\star$; it records the largest excursion of coordinate i from its starting value.

To see what update this substitution suggests, replace d_i in (15) by $\delta_{t,i}^{\max}$ and minimize

$$\Psi_t(s) := \frac{1}{2\eta} \sum_{i=1}^{d} (\delta_{t,i}^{\max})^2 s_i + \frac{\eta}{2} \sum_{i=1}^{d} \frac{A_{t,i}}{s_i}. \tag{23}$$

This substitution is used only to derive the update. Since $\delta_{t,i}^{\max}$ is not the distance to $w^\star$, (23) is not itself a regret bound.

For each coordinate,

$$\frac{\partial \Psi_t}{\partial s_i} = \frac{(\delta_{t,i}^{\max})^2}{2\eta} - \frac{\eta A_{t,i}}{2s_i^2}. \tag{24}$$

Setting the derivative equal to zero yields

$$s_i^2 = \frac{\eta^2 A_{t,i}}{(\delta_{t,i}^{\max})^2}, \tag{25}$$

so, after absorbing the common factor into the scalar learning rate,

$$s_{t,i} \propto \sqrt{\frac{A_{t,i}}{(\delta_{t,i}^{\max})^2}}. \tag{26}$$

This expression is undefined at initialization, where $\delta_{t,i}^{\max} = 0$. Adding a trace penalty gives

$$\Psi_{t,\lambda}(s) := \frac{1}{2\eta} \sum_{i=1}^{d} \big((\delta_{t,i}^{\max})^2 + \lambda\big) s_i + \frac{\eta}{2} \sum_{i=1}^{d} \frac{A_{t,i}}{s_i}, \qquad \lambda > 0. \tag{27}$$

The first-order condition becomes

$$\frac{(\delta_{t,i}^{\max})^2 + \lambda}{2\eta} - \frac{\eta A_{t,i}}{2s_i^2} = 0, \tag{28}$$

which gives

$$s_{t,i} \propto \sqrt{\frac{A_{t,i}}{(\delta_{t,i}^{\max})^2 + \lambda}}. \tag{29}$$

When $(\delta_{t,i}^{\max})^2$ is small relative to λ, this scaling is approximately proportional to $\sqrt{A_{t,i}}$. For a fixed gradient history, larger $\delta_{t,i}^{\max}$ gives smaller $s_{t,i}$ and therefore a larger update in coordinate i.

B.3 Reparameterization and Behavior

Only the ratios between the $s_{t,i}$ matter here; any factor common to all coordinates can be absorbed into η. The update from (29) can therefore be written as

$$w_{t+1,i} = w_{t,i} - \eta g_{t,i} \sqrt{\frac{(\delta_{t,i}^{\max})^2 + \lambda}{A_{t,i}}} = w_{t,i} - \eta\sqrt{\lambda}\, g_{t,i} \sqrt{\frac{1 + \lambda^{-1}(\delta_{t,i}^{\max})^2}{A_{t,i}}}. \tag{30}$$

Let

$$\gamma := \lambda^{-1}, \tag{31}$$

and absorb the common factor $\sqrt{\lambda}$ into the learning rate. From this point onward, η denotes this rescaled learning rate, and η and γ are treated as independent hyperparameters. Adding $\varepsilon > 0$ again for numerical stability gives the DA-AdaGrad rule

$$\begin{aligned} h_{t,i}(\gamma) &:= \sqrt{\varepsilon + \frac{A_{t,i}}{1 + \gamma(\delta_{t,i}^{\max})^2}}, \\ w_{t+1,i} &= w_{t,i} - \eta \frac{g_{t,i}}{h_{t,i}(\gamma)}. \end{aligned} \tag{32}$$

At $\gamma = 0$, (32) reduces exactly to (21). Larger γ gives displacement more influence on the step size. The scaling factor $1 + \gamma(\delta_{t,i}^{\max})^2$ departs substantially from its baseline value once $|\delta_{t,i}^{\max}|$ is comparable to or larger than $\gamma^{-1/2}$.

Two cases illustrate this behavior. If $A_{t,i}$ and $(\delta_{t,i}^{\max})^2$ increase together, then

$$\frac{A_{t,i}}{1 + \gamma(\delta_{t,i}^{\max})^2} \tag{33}$$

Table 2: Distance-aware optimizers. The displacement maximum is coordinatewise; DA-AdamW additionally applies decoupled weight decay.

Method	Numerator	Second-moment estimate	Displacement statistic
DA-AdaGrad	g_t	$A_t = A_{t-1} + g_t \odot g_t$	$\delta_t^{\max} = \max_{1\le s\le t} \lvert w_s - w_1\rvert$
DA-Adam / DA-AdamW	$\widehat{m}_t$	$\widehat{v}_t$	$\delta_t^{\max} = \max_{1\le s\le t} \lvert w_s - u_{s-1}\rvert$
DA-AMSGrad	$\widehat{m}_t$	$v_t^{\max}/(1-\beta_2^t)$	$\delta_t^{\max} = \max_{1\le s\le t} \lvert w_s - u_{s-1}\rvert$

can grow much more slowly than $A_{t,i}$ alone. The method therefore need not reduce the effective learning rate as quickly when $A_{t,i}$ is large if the coordinate has also moved far from w_1. If $A_{t,i}$ continues to grow after $\delta_{t,i}^{\max}$ has stopped changing, then $1 + \gamma(\delta_{t,i}^{\max})^2$ is constant and $h_{t,i}$ again grows like $\sqrt{A_{t,i}}$. The effective learning rate then decreases at the same rate as AdaGrad's, up to a constant factor.

B.4 Adam and AMSGrad Updates

The same factor $1 + \gamma(\delta_t^{\max})^2$ can be combined with different first moments, second moments, and reference points. The symbols $\odot$ and $\oslash$ denote elementwise multiplication and division; absolute values, powers, square roots, and maxima are also coordinatewise. Table 2 summarizes the methods. Hats denote the usual bias-corrected moment estimates.

For DA-AdaGrad, the displacement maximum makes $b_{t,i}$ nondecreasing. The regret bound and fixed-path comparison in Section 5 apply to this method. The moving-anchor methods below are evaluated empirically and are not covered by these guarantees.

The displacement from w_1 is the sum of all previous updates:

$$w_t - w_1 = \sum_{s=1}^{t-1} (w_{s+1} - w_s). \tag{34}$$

Adam uses an exponential moving average instead of the full sum of squared gradients (Kingma & Ba, 2015). For the displacement term, we likewise replace the full sum in (34) by a leaky sum, using the same decay β_2 as the second moment. Initialize

$$m_0 = 0, \qquad v_0 = 0, \qquad u_0 = w_1, \qquad \delta_0^{\max} = 0. \tag{35}$$

At step t, compute the usual Adam moments

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, & \widehat{m}_t &= \frac{m_t}{1-\beta_1^t}, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t \odot g_t, & \widehat{v}_t &= \frac{v_t}{1-\beta_2^t}, \end{aligned} \tag{36}$$

then measure displacement from the previous anchor, retain its maximum, and update the anchor in that order:

$$\delta_t := |w_t - u_{t-1}|, \qquad \delta_t^{\max} := \max\{\delta_{t-1}^{\max}, \delta_t\}, \qquad u_t := \beta_2 u_{t-1} + (1-\beta_2) w_t. \tag{37}$$

The signed displacement from the anchor satisfies

$$w_{t+1} - u_t = (w_{t+1} - w_t) + \beta_2 (w_t - u_{t-1}), \tag{38}$$

so $w_t - u_{t-1}$ is a leaky sum of recent updates, just as (34) is the full sum. Appendix G gives the corresponding finite-sum identity.

Although u_t is an exponential moving average, $w_t - u_{t-1}$ is an unnormalized leaky sum: no factor $1-\beta_2$ multiplies the new update in (38). Consequently, changing β_2 changes both the memory length and the scale of the displacement statistic, so values of γ are not directly comparable across different β_2.

Using the AdamW convention with the numerical offset outside the square root (Loshchilov & Hutter, 2019), define

$$b_t := \mathbf{1} + \gamma \delta_t^{\max} \odot \delta_t^{\max}, \qquad h_t := \sqrt{\widehat{v}_t \oslash b_t} + \varepsilon \mathbf{1}. \tag{39}$$

The DA-AdamW update is

$$w_{t+1} = (1 - \eta_t\sigma)w_t - \eta_t \widehat{m}_t \oslash h_t, \tag{40}$$

where $\sigma \geq 0$ is the decoupled weight-decay coefficient. Setting $\sigma = 0$ gives DA-Adam. At $\gamma = 0$, (39)–(40) reduce exactly to Adam or AdamW with ε in the same position. Weight decay is still applied directly to w_t; γ changes only the adaptive update.

For DA-AMSGrad, also initialize $v_0^{\max} = 0$, update $v_t^{\max} = \max\{v_{t-1}^{\max}, v_t\}$, and replace $\widehat{v}_t$ by $v_t^{\max}/(1 - \beta_2^t)$ in the denominator. The gradient and displacement maxima are taken separately. At $\gamma = 0$, this recovers AMSGrad with the same bias correction and weight-decay convention.

B.5 Optimizer Algorithms

Algorithm 2 gives DA-AdaGrad using the coordinatewise conventions above. Relative to their base optimizers, DA-AdaGrad and DA-AdamW each store two additional parameter-sized vectors: the reference point and the displacement maximum. They add $O(d)$ elementwise work per step. Algorithm 3 gives DA-AdamW; setting $\sigma = 0$ gives DA-Adam. At $\gamma = 0$, the same pseudocode recovers the respective base optimizer.

Algorithm 2 Distance-Aware AdaGrad

Input: Initial point w_1, horizon T, learning rate $\eta > 0$, displacement gain $\gamma \geq 0$, numerical offset $\varepsilon > 0$

$A_0 \leftarrow 0$, $\delta_0^{\max} \leftarrow 0$
for $t = 1, \ldots, T$ **do**
  Compute $g_t \in \partial f_t(w_t)$
  $A_t \leftarrow A_{t-1} + g_t \odot g_t$
  $\delta_t^{\max} \leftarrow \max\{\delta_{t-1}^{\max}, |w_t - w_1|\}$
  $b_t \leftarrow \mathbf{1} + \gamma\, \delta_t^{\max} \odot \delta_t^{\max}$
  $h_t \leftarrow \sqrt{\varepsilon\mathbf{1} + A_t \oslash b_t}$
  $w_{t+1} \leftarrow w_t - \eta\, g_t \oslash h_t$
end for
return w_{T+1}

Algorithm 3 Distance-Aware Adam and AdamW

Input: Initial point w_1, horizon T, learning rates $\eta_t > 0$, moment decays $\beta_1, \beta_2 \in [0, 1)$, displacement gain $\gamma \geq 0$, numerical offset $\varepsilon > 0$, weight decay $\sigma \geq 0$

$m_0 \leftarrow 0$, $v_0 \leftarrow 0$, $u_0 \leftarrow w_1$, $\delta_0^{\max} \leftarrow 0$
for $t = 1, \ldots, T$ **do**
  Compute the gradient g_t at w_t
  $m_t \leftarrow \beta_1 m_{t-1} + (1 - \beta_1)g_t$
  $v_t \leftarrow \beta_2 v_{t-1} + (1 - \beta_2)g_t \odot g_t$
  $\widehat{m}_t \leftarrow m_t/(1 - \beta_1^t)$, $\widehat{v}_t \leftarrow v_t/(1 - \beta_2^t)$
  $\delta_t \leftarrow |w_t - u_{t-1}|$
  $\delta_t^{\max} \leftarrow \max\{\delta_{t-1}^{\max}, \delta_t\}$
  $u_t \leftarrow \beta_2 u_{t-1} + (1 - \beta_2)w_t$
  $b_t \leftarrow \mathbf{1} + \gamma\, \delta_t^{\max} \odot \delta_t^{\max}$
  $h_t \leftarrow \sqrt{\widehat{v}_t \oslash b_t} + \varepsilon\mathbf{1}$
  $w_{t+1} \leftarrow (1 - \eta_t\sigma)w_t - \eta_t\, \widehat{m}_t \oslash h_t$
end for
return w_{T+1}

C Formal Theory Statements and Fixed-Path Analysis

We analyze DA-AdaGrad. The factor $b_{t,i}$ is nondecreasing, which permits an AdaGrad-style regret bound even though the coordinatewise denominator itself need not be monotone.

C.1 A Regret Bound for DA-AdaGrad

Let

$$\mathcal{W} = \prod_{i=1}^{d} [\ell_i, u_i] \subseteq \mathbb{R}^d$$

be an axis-aligned closed box. Let $f_1, \dots, f_T : \mathcal{W} \to \mathbb{R}$ be convex, fix $w_1, w^\star \in \mathcal{W}$, and let $g_t \in \partial f_t(w_t)$. For $t \geq 1$, define

$$\begin{aligned} A_{0,i} &:= a_{0,i} := 0, \qquad A_{t,i} := \sum_{s=1}^{t} g_{s,i}^2, \\ a_{t,i} &:= \sqrt{A_{t,i}}, \qquad \Delta a_{t,i} := a_{t,i} - a_{t-1,i}, \end{aligned} \tag{41}$$

and

$$\delta_{t,i} := |w_{t,i} - w_{1,i}|, \qquad \delta_{t,i}^{\max} := \max_{1 \leq s \leq t} \delta_{s,i}, \qquad q_{t,i} := \sqrt{1 + \gamma(\delta_{t,i}^{\max})^2}. \tag{42}$$

$$D_i := \max_{1 \leq t \leq T+1} |w_{t,i} - w_i^\star|. \tag{43}$$

For $\gamma \geq 0$ and $\varepsilon > 0$, set

$$h_{t,i} := \sqrt{\varepsilon + \frac{A_{t,i}}{1 + \gamma(\delta_{t,i}^{\max})^2}} = \sqrt{\varepsilon + \frac{A_{t,i}}{q_{t,i}^2}}, \tag{44}$$

and use the projected coordinatewise update

$$w_{t+1,i} = \Pi_{[\ell_i, u_i]} \left(w_{t,i} - \eta \frac{g_{t,i}}{h_{t,i}} \right). \tag{45}$$

At $\gamma = 0$, this is projected diagonal AdaGrad.

The regret bound depends on

$$U_i^{(\gamma)} := h_{T,i} + \sum_{t=1}^{T-1} h_{t,i} \left(1 - \frac{q_{t,i}}{q_{t+1,i}} \right), \tag{46}$$

$$S_i^{(\gamma)} := \sum_{t=1}^{T} \frac{g_{t,i}^2}{h_{t,i}}. \tag{47}$$

Because $q_{t,i}$ is nondecreasing, every term added to $h_{T,i}$ in (46) is nonnegative. This quantity accounts for the growth of the coordinatewise denominator: in particular,

$$h_{1,i} + \sum_{t=2}^{T} (h_{t,i} - h_{t-1,i})_+ \leq U_i^{(\gamma)}. \tag{48}$$

The second quantity, $S_i^{(\gamma)}$, is the usual squared-gradient term from the diagonal AdaGrad analysis.

Theorem C.1 (Regret bound)**.** *For every $\eta > 0$, the iterates in* (45) *satisfy*

$$\begin{aligned} R_T &:= \sum_{t=1}^{T} \big(f_t(w_t) - f_t(w^\star) \big) \\ &\leq \frac{1}{2\eta} \sum_{i=1}^{d} D_i^2 U_i^{(\gamma)} + \frac{\eta}{2} \sum_{i=1}^{d} S_i^{(\gamma)}. \end{aligned} \tag{49}$$

The proof is given in Appendix D.

Corollary C.2 (Recovery of AdaGrad)**.** *At $\gamma = 0$,*

$$U_i^{(0)} = \sqrt{\varepsilon + A_{T,i}}, \qquad S_i^{(0)} = \sum_{t=1}^{T} \frac{g_{t,i}^2}{\sqrt{\varepsilon + A_{t,i}}}, \tag{50}$$

so Theorem C.1 recovers the unrelaxed bound obtained from the standard diagonal AdaGrad analysis (Duchi et al., 2011).

The proof and the identity used below are given in Appendix D.

The scaling factor changes the two terms in opposite directions. When $\varepsilon = 0$ and the same iterates and gradients are used, all sums containing $g_{t,i}^2/a_{t,i}$ are understood to run only over pairs with $g_{t,i} \neq 0$; equivalently, a zero-gradient summand is defined to be zero. This avoids the otherwise undefined ratio $0/0$ before a coordinate receives its first nonzero gradient.

$$\begin{aligned} U_i^{(\gamma)} &= \sum_{t=1}^{T} \frac{\Delta a_{t,i}}{q_{t,i}} \leq U_i^{(0)}, \\ S_i^{(\gamma)} &= \sum_{t=1}^{T} q_{t,i} \frac{g_{t,i}^2}{a_{t,i}} \geq S_i^{(0)}. \end{aligned} \tag{51}$$

Thus the factor $1 + \gamma(\delta_{t,i}^{\max})^2$ decreases the coefficient of $1/\eta$ and increases the coefficient of η. The next result minimizes their sum over η and identifies when $\gamma > 0$ reduces the resulting bound.

C.2 When Does $\gamma > 0$ Improve the Bound?

For this comparison, take the limit as ε approaches zero in the coefficient formulas and fix a completed sequence while varying $\gamma \geq 0$. Changing γ then changes the coefficients but not the iterates or gradients. Let

$$\begin{aligned} \mathcal{T} &:= (w_1, \ldots, w_{T+1}; g_1, \ldots, g_T), \\ \mathcal{I} &:= \{(t,i) : g_{t,i} \neq 0\}. \end{aligned}$$

The resulting coefficients are:

$$P_\gamma(\mathcal{T}) := \sum_{(t,i)\in\mathcal{I}} D_i^2 \Delta a_{t,i} (1 + \gamma(\delta_{t,i}^{\max})^2)^{-1/2}, \tag{52}$$

$$Q_\gamma(\mathcal{T}) := \sum_{(t,i)\in\mathcal{I}} \frac{g_{t,i}^2}{a_{t,i}} (1 + \gamma(\delta_{t,i}^{\max})^2)^{1/2}. \tag{53}$$

For this fixed $\mathcal{T}$, minimizing the two-term expression in (49) over η gives

$$B_\gamma^\star(\mathcal{T}) := \inf_{\eta>0} \left\{ \frac{P_\gamma(\mathcal{T})}{2\eta} + \frac{\eta Q_\gamma(\mathcal{T})}{2} \right\} = \sqrt{P_\gamma(\mathcal{T}) Q_\gamma(\mathcal{T})}. \tag{54}$$

Assume $P_0(\mathcal{T}) > 0$ and $Q_0(\mathcal{T}) > 0$, and define

$$\pi_{t,i}^P := \frac{D_i^2 \Delta a_{t,i}}{P_0(\mathcal{T})}, \qquad \pi_{t,i}^Q := \frac{g_{t,i}^2/a_{t,i}}{Q_0(\mathcal{T})}, \qquad (t,i) \in \mathcal{I}. \tag{55}$$

Both sets of weights sum to one. The weight $\pi_{t,i}^P$ is the fraction of P_0 contributed by (t,i), while $\pi_{t,i}^Q$ is the corresponding fraction of Q_0. Thus M_P and M_Q below are weighted averages of $(\delta_{t,i}^{\max})^2$.

Define

$$\begin{aligned} M_P &:= \sum_{(t,i)\in\mathcal{I}} \pi_{t,i}^P (\delta_{t,i}^{\max})^2, \\ M_Q &:= \sum_{(t,i)\in\mathcal{I}} \pi_{t,i}^Q (\delta_{t,i}^{\max})^2, \\ \Delta_{\max} &:= M_P - M_Q, \end{aligned} \tag{56}$$

Proposition C.3. *For every $\gamma \geq 0$,*

$$\frac{B_\gamma^\star(\mathcal{T})^2}{B_0^\star(\mathcal{T})^2} = \left[\sum_{(t,i)\in\mathcal{I}} \pi_{t,i}^P (1+\gamma(\delta_{t,i}^{\max})^2)^{-1/2}\right] \times \left[\sum_{(t,i)\in\mathcal{I}} \pi_{t,i}^Q (1+\gamma(\delta_{t,i}^{\max})^2)^{1/2}\right]. \tag{57}$$

Moreover,

$$\frac{d}{d\gamma}\log\frac{B_\gamma^\star(\mathcal{T})}{B_0^\star(\mathcal{T})}\Big|_{\gamma=0} = -\frac{\Delta_{\max}}{4}. \tag{58}$$

If $\Delta_{\max} > 0$, then every sufficiently small positive γ satisfies

$$B_\gamma^\star(\mathcal{T}) < B_0^\star(\mathcal{T}). \tag{59}$$

If $\Delta_{\max} < 0$, the reverse inequality holds for every sufficiently small positive γ; if $\Delta_{\max} = 0$, the first-order comparison is inconclusive.

The proof is given in Appendix E.

C.3 Connection to the dispersion of coordinatewise distances

We explain the approximation

$$M_P - M_Q \approx c\,\mathbb{E}_i[d_i^2]\ \mathrm{CV}(d_i^2)^2$$

and quantify the error incurred by its underlying assumptions. Throughout, we consider the coefficient comparison on a fixed trajectory in the limit $\varepsilon \to 0$. Define the initial coordinatewise distances and their squares by

$$d_i := |w_{1,i} - w_i^\star|, \qquad x_i := d_i^2, \qquad m := \mathbb{E}_i[x_i] = \frac{1}{d}\sum_{i=1}^d x_i > 0.$$

Here $\mathbb{E}_i$, Var_i, SD_i, and Cov_i use the uniform distribution over coordinates. In particular,

$$\mathrm{CV}(d_i^2)^2 = \frac{\mathrm{Var}_i(x_i)}{m^2}.$$

Let $r_{t,i} := (\delta_{t,i}^{\max})^2$ and $r_{\max} := \max_{t,i} r_{t,i}$. Retain the definitions $a_{t,i} = \sqrt{\sum_{s=1}^t g_{s,i}^2}$, $\Delta a_{t,i} = a_{t,i} - a_{t-1,i}$, and $D_i = \max_{1\leq t\leq T+1} |w_{t,i} - w_i^\star|$. Assuming $P_0, Q_0 > 0$, write the weights from the preceding analysis as

$$\pi_{t,i}^P = \frac{D_i^2 \Delta a_{t,i}}{P_0}, \qquad \pi_{t,i}^Q = \frac{g_{t,i}^2/a_{t,i}}{Q_0}, \qquad P_0 = \sum_{t,i} D_i^2 \Delta a_{t,i}, \quad Q_0 = \sum_{t,i} \frac{g_{t,i}^2}{a_{t,i}}.$$

Zero-gradient summands are defined to be zero. With this convention, both weight arrays are probability distributions on all time-coordinate pairs, and

$$\Delta_{\max} := M_P - M_Q = \sum_{t,i} (\pi_{t,i}^P - \pi_{t,i}^Q) r_{t,i}. \tag{60}$$

An exact covariance identity. For every pair with $g_{t,i} \neq 0$, define

$$\theta_{t,i} := \frac{a_{t,i}}{a_{t,i} + a_{t-1,i}}, \qquad z_{t,i} := D_i^2 \theta_{t,i}, \qquad \rho_{t,i} := \pi_{t,i}^Q.$$

Their values outside the support of ρ are immaterial. Since

$$\Delta a_{t,i} = \frac{g_{t,i}^2}{a_{t,i} + a_{t-1,i}} = \theta_{t,i}\frac{g_{t,i}^2}{a_{t,i}},$$

we have

$$\pi_{t,i}^P = \frac{\rho_{t,i} z_{t,i}}{\mathbb{E}_\rho[z]}.$$

Consequently,

$$\Delta_{\max} = \frac{\mathrm{Cov}_\rho(z, r)}{\mathbb{E}_\rho[z]}. \tag{61}$$

This identity is exact. When each new squared gradient is small relative to its accumulated history, $\theta_{t,i}$ is close to $1/2$. The principal association is then between the path radius D_i^2 and the squared running displacement $r_{t,i}$, under gradient-dependent weights. Additional assumptions are needed to express this association using the unweighted dispersion of the initial distances.

A quantitative approximation. Let $\mu = (\mu_1, \dots, \mu_T)$ be a probability distribution over iterations, and define reference weights

$$\bar{\pi}^P_{t,i} := \mu_t \frac{x_i}{\sum_j x_j}, \qquad \bar{\pi}^Q_{t,i} := \frac{\mu_t}{d}. \tag{62}$$

These weights represent a common temporal distribution, with coordinates weighted by initial squared distance in the first term and uniformly in the second. Define their deviations from the actual weights by

$$\tau_P := \frac{1}{2} \sum_{t,i} |\pi^P_{t,i} - \bar{\pi}^P_{t,i}|, \qquad \tau_Q := \frac{1}{2} \sum_{t,i} |\pi^Q_{t,i} - \bar{\pi}^Q_{t,i}|.$$

For a constant $c > 0$, write the temporally averaged displacement as

$$\bar{r}_i := \sum_{t=1}^{T} \mu_t r_{t,i} = c x_i + e_i. \tag{63}$$

The residual e_i measures departure from proportionality between averaged squared displacement and initial squared distance.

Proposition C.4 (Quantitative relation to distance dispersion)**.** *Under the definitions above, for every choice of μ and $c > 0$,*

$$\left|\Delta_{\max} - c\, m\ \mathrm{CV}(d_i^2)^2\right| \le \mathrm{CV}(d_i^2)\, \mathrm{SD}_i(e_i) + r_{\max}(\tau_P + \tau_Q). \tag{64}$$

Proof. First replace the actual weights in (60) with the reference weights. The resulting quantity is

$$\begin{aligned} \bar{\Delta} &:= \sum_{t,i} (\bar{\pi}^P_{t,i} - \bar{\pi}^Q_{t,i}) r_{t,i} \\ &= \frac{\mathbb{E}_i[x_i \bar{r}_i]}{m} - \mathbb{E}_i[\bar{r}_i] = \frac{\mathrm{Cov}_i(x_i, \bar{r}_i)}{m}. \end{aligned} \tag{65}$$

Substituting (63) gives

$$\begin{aligned} \bar{\Delta} &= c\, \frac{\mathrm{Var}_i(x_i)}{m} + \frac{\mathrm{Cov}_i(x_i, e_i)}{m} \\ &= c\, m\ \mathrm{CV}(d_i^2)^2 + \frac{\mathrm{Cov}_i(x_i, e_i)}{m}. \end{aligned} \tag{66}$$

By Cauchy–Schwarz,

$$\frac{|\,\mathrm{Cov}_i(x_i, e_i)|}{m} \le \frac{\mathrm{SD}_i(x_i)}{m}\, \mathrm{SD}_i(e_i) = \mathrm{CV}(d_i^2)\, \mathrm{SD}_i(e_i).$$

For any two probability distributions p, q and any array r with $0 \le r \le r_{\max}$,

$$|\mathbb{E}_p[r] - \mathbb{E}_q[r]| \le r_{\max}\, \mathrm{TV}(p, q), \qquad \mathrm{TV}(p, q) := \frac{1}{2} \sum |p - q|.$$

Applying this inequality to the two weight replacements yields $|\Delta_{\max} - \bar{\Delta}| \le r_{\max}(\tau_P + \tau_Q)$. Combining these estimates proves (64). □

In particular, the claimed CV formula is exact for the reference weights when $\bar{r}_i = c x_i$. More generally, it is accurate when the right-hand side of (64) is small relative to $c\, m\ \mathrm{CV}(d_i^2)^2$.

When do the reference weights approximate the actual weights? Consider first a trajectory satisfying $D_i = d_i$ and $g^2_{t,i} = v_t$ for every coordinate i. The radius equality holds, for example, when each coordinate stays between its initial point and the comparator. Let

$$a_t := \sqrt{\sum_{s=1}^{t} v_s}, \qquad \mu^P_t := \frac{a_t - a_{t-1}}{a_T}, \qquad \mu^Q_t := \frac{v_t / a_t}{\sum_{s=1}^{T} v_s / a_s}.$$

Then the actual weights factor exactly as

$$\pi^P_{t,i} = \mu^P_t \frac{d_i^2}{\sum_j d_j^2}, \qquad \pi^Q_{t,i} = \frac{\mu^Q_t}{d}.$$

Choosing $\mu = \mu^Q$ gives

$$\tau_Q = 0, \qquad \tau_P = \mathrm{TV}(\mu^P, \mu^Q). \tag{67}$$

Thus, a common temporal distribution is a good approximation when these two normalized time-weight distributions are close. Departures from equal coordinate gradient histories or from $D_i = d_i$ are included in τ_P and τ_Q in the general proposition.

Constant-magnitude gradients: a finite-horizon estimate. For $v_t = \sigma^2 > 0$, the preceding distributions reduce to

$$\mu^P_t = \frac{\sqrt{t} - \sqrt{t-1}}{\sqrt{T}}, \qquad \mu^Q_t = \frac{t^{-1/2}}{H_T}, \qquad H_T := \sum_{s=1}^{T} s^{-1/2}.$$

They satisfy

$$\mathrm{TV}(\mu^P, \mu^Q) \le \frac{2\sqrt{T} - H_T}{2\sqrt{T}} \le \frac{1}{\sqrt{T}}. \tag{68}$$

To prove this, let $\xi_t := 2(\sqrt{t} - \sqrt{t-1}) - t^{-1/2} \ge 0$ and $C_T := \sum_t \xi_t = 2\sqrt{T} - H_T$. The integral bound $H_T \ge 2(\sqrt{T} - 1)$ gives $C_T \le 2$. Moreover, with $\zeta_t := \xi_t / C_T$,

$$\mu^P = \frac{H_T}{2\sqrt{T}} \mu^Q + \frac{C_T}{2\sqrt{T}} \zeta.$$

The total variation distance is therefore at most $C_T/(2\sqrt{T})$, proving (68). Consequently, when $D_i = d_i$ and the squared gradients have a common constant magnitude, Proposition C.4 gives

$$\left| \Delta_{\max} - c\, m\, \mathrm{CV}(d_i^2)^2 \right| \le \mathrm{CV}(d_i^2)\, \mathrm{SD}_i(e_i) + \frac{r_{\max}}{\sqrt{T}}, \tag{69}$$

where e_i is defined using $\mu = \mu^Q$. This estimate controls the approximation of the displacement averages, not just the convergence of their weight distributions. In particular, the temporal error vanishes when $r_{\max}$ remains bounded as T grows.

Interpreting displacement alignment. If coordinates have approximately the same fractional displacement profile, so that $\delta^{\max}_{t,i} \approx \alpha_t d_i$, then

$$\bar{r}_i \approx \left(\sum_{t=1}^{T} \mu_t \alpha_t^2 \right) d_i^2, \qquad c := \sum_{t=1}^{T} \mu_t \alpha_t^2.$$

Another favorable situation occurs when the running displacements have largely stabilized near the initial coordinatewise distances during iterations carrying most of the temporal weight; then $c \approx 1$. These are alignment conditions on the observed trajectory. Distance heterogeneity alone does not imply them.

Implications for the balanced regret-bound expression. Define

$$\mathcal{E}_{\mathrm{CV}} := \mathrm{CV}(d_i^2)\, \mathrm{SD}_i(e_i) + r_{\max}(\tau_P + \tau_Q).$$

The proposition and the preceding first-order expansion imply

$$\frac{B^\star_\gamma}{B^\star_0} = 1 - \frac{\gamma c m}{4} \mathrm{CV}(d_i^2)^2 + O\big(\gamma \mathcal{E}_{\mathrm{CV}} + \gamma^2\big), \qquad \gamma \to 0. \tag{70}$$

The remainder in γ is for the fixed trajectory; its constant need not be uniform over different distance configurations. In particular, the sufficient condition

$$c\, m\, \mathrm{CV}(d_i^2)^2 > \mathcal{E}_{\mathrm{CV}} \tag{71}$$

ensures $\Delta_{\max} > 0$ and hence $B^\star_\gamma < B^\star_0$ for every sufficiently small $\gamma > 0$. At fixed dimension and $\sum_i d_i^2$, the mean m is fixed, so greater distance dispersion increases the leading improvement term, provided c and the approximation errors remain comparable. At fixed CV, the leading term instead grows with the mean squared distance.

Nonuniform coordinate gradient weights. If the coordinate gradient contributions are substantially nonuniform, the natural dispersion measure is weighted. Specifically, suppose the common temporal approximation instead has the form

$$\pi^Q_{t,i} \approx \mu_t \nu_i, \qquad \pi^P_{t,i} \approx \mu_t \frac{\nu_i d_i^2}{\sum_j \nu_j d_j^2}, \qquad \sum_i \nu_i = 1.$$

The same calculation gives

$$\Delta_{\max} \approx \frac{\mathrm{Cov}_\nu(d_i^2, \bar{r}_i)}{\mathbb{E}_\nu[d_i^2]} \approx c\, \mathbb{E}_\nu[d_i^2]\ \mathrm{CV}_\nu(d_i^2)^2.$$

The unweighted relation corresponds to approximately uniform ν_i. Holding curvature fixed does not by itself ensure comparable gradient weights, since the gradients can still depend on the initial distances. Accordingly, the CV sweeps test the predicted association under these alignment and weighting conditions; they do not establish a universal monotonic relation between unweighted CV and optimization performance. All comparisons here concern bound expressions on the same fixed trajectory, rather than regret comparisons between separate algorithm runs.

C.4 Interpreting the Weights

The comparison simplifies once each new squared gradient is small relative to its accumulated history:

$$\Delta a_{t,i} = \frac{g_{t,i}^2}{a_{t,i} + a_{t-1,i}} \approx \frac{1}{2} \frac{g_{t,i}^2}{a_{t,i}}$$

For bounded, persistent gradients whose squared sum grows linearly, the normalized weights formed from $\Delta a_{t,i}$ and $g_{t,i}^2/a_{t,i}$ therefore become asymptotically equivalent. Constant-gradient paths, including affine regions and the absolute-value loss before a crossing, are simple examples; over a fixed traversal, the equivalence holds as $\eta \to 0$.

The following separable example illustrates the weighting condition. Its two temporal weight distributions agree asymptotically, so the main distinction is how they weight coordinate distances. Consider

$$f(w) = \sum_{i=1}^{d} |w_i - w_i^\star|$$

and a sequence of T updates on which no coordinate equals its minimizer. Then

$$g_{t,i}^2 = 1, \qquad a_{t,i} = \sqrt{t},$$
$$\Delta a_{t,i} = \sqrt{t} - \sqrt{t-1}, \quad \frac{g_{t,i}^2}{a_{t,i}} = t^{-1/2}.$$

Normalize the two sets of time weights by

$$\mu_T^P(t) := \frac{\sqrt{t} - \sqrt{t-1}}{\sqrt{T}}, \qquad \mu_T^Q(t) := \frac{t^{-1/2}}{\sum_{s=1}^T s^{-1/2}}.$$

For fixed $u \in (0, 1]$, let $k = \lfloor uT \rfloor$. As $T \to \infty$,

$$\sum_{t=1}^{k} \mu_T^P(t) = \sqrt{\frac{k}{T}} \longrightarrow \sqrt{u},$$
$$\sum_{t=1}^{k} \mu_T^Q(t) = \frac{\sum_{t=1}^k t^{-1/2}}{\sum_{t=1}^T t^{-1/2}} \longrightarrow \sqrt{u}.$$

Both cumulative sums therefore converge to the cumulative distribution function of the same distribution μ on $[0, 1]$, given by $F(u) = \sqrt{u}$. Their coordinate weights differ:

$$\pi_{t,i}^P = \frac{D_i^2}{\sum_j D_j^2} \mu_T^P(t), \qquad \pi_{t,i}^Q = \frac{1}{d} \mu_T^Q(t).$$

Using $q_{t,i}$ from equation (42), equation (57) becomes

$$\frac{B_\gamma^\star(\mathcal{T})^2}{B_0^\star(\mathcal{T})^2} = \left[\sum_{i=1}^{d} \frac{D_i^2}{\sum_j D_j^2} \mathbb{E}_{\mu_T^P}[q_{t,i}^{-1}]\right] \times \left[\frac{1}{d}\sum_{i=1}^{d} \mathbb{E}_{\mu_T^Q}[q_{t,i}]\right].$$

The ratio $\mu_T^P(t)/\mu_T^Q(t)$ is nonincreasing in t. Since $q_{t,i}^{-1}$ is also nonincreasing, we have

$$\mathbb{E}_{\mu_T^P}[q_{t,i}^{-1}] \geq \mathbb{E}_{\mu_T^Q}[q_{t,i}^{-1}].$$

Cauchy–Schwarz, applied jointly over uniformly weighted coordinates and iterations weighted by μ_T^Q, therefore gives

$$\left(\frac{1}{d}\sum_{i=1}^{d} \mathbb{E}_{\mu_T^P}[q_{t,i}^{-1}]\right)\left(\frac{1}{d}\sum_{i=1}^{d} \mathbb{E}_{\mu_T^Q}[q_{t,i}]\right) \geq 1.$$

Thus, for the bound ratio to be less than one, weighting coordinates by D_i^2 must reduce the first factor relative to uniform coordinate weighting. Equivalently, D_i^2 must be negatively correlated across coordinates with $\mathbb{E}_{\mu_T^P}[q_{t,i}^{-1}]$. This condition is necessary, but not sufficient. Section 6 explores this condition by varying the relative dispersion in squared initial coordinate distances, d_i^2.

D Proof of the DA-AdaGrad Regret Bound

We use the notation of Section C.1. For the proofs in this appendix, define

$$z_{t,i} := q_{t,i} h_{t,i} = \sqrt{A_{t,i} + \varepsilon q_{t,i}^2}, \qquad z_{0,i} := \sqrt{\varepsilon}. \tag{72}$$

Because both $A_{t,i}$ and $q_{t,i}$ are nondecreasing, $z_{t,i}$ is nondecreasing as well. Also, $\delta_{1,i}^{\max} = 0$, so $q_{1,i} = 1$.

D.1 Control of the Positive Variation

Lemma D.1 (Equivalent form of $U_i^{(\gamma)}$)**.** *For every coordinate i,*

$$U_i^{(\gamma)} = \sqrt{\varepsilon} + \sum_{t=1}^{T} \frac{z_{t,i} - z_{t-1,i}}{q_{t,i}}. \tag{73}$$

Proof. Using $h_{t,i} = z_{t,i}/q_{t,i}$ in the definition of $U_i^{(\gamma)}$ gives

$$\begin{aligned} U_i^{(\gamma)} &= \frac{z_{T,i}}{q_{T,i}} + \sum_{t=1}^{T-1} z_{t,i} \left(\frac{1}{q_{t,i}} - \frac{1}{q_{t+1,i}} \right) \\ &= \frac{z_{1,i}}{q_{1,i}} + \sum_{t=2}^{T} \frac{z_{t,i} - z_{t-1,i}}{q_{t,i}}. \end{aligned}$$

Since $q_{1,i} = 1$ and $z_{0,i} = \sqrt{\varepsilon}$,

$$\frac{z_{1,i}}{q_{1,i}} = \sqrt{\varepsilon} + \frac{z_{1,i} - z_{0,i}}{q_{1,i}}.$$

Substitution proves (73). □

Lemma D.2 (Positive variation of the denominator)**.** *For every coordinate i,*

$$h_{1,i} + \sum_{t=2}^{T} \big(h_{t,i} - h_{t-1,i}\big)_+ \le U_i^{(\gamma)}, \qquad x_+ := \max\{x, 0\}. \tag{74}$$

Proof. The first term satisfies

$$h_{1,i} = z_{1,i} = \sqrt{\varepsilon} + \frac{z_{1,i} - z_{0,i}}{q_{1,i}}.$$

For $t \ge 2$, monotonicity of $q_{t,i}$ implies

$$h_{t,i} - h_{t-1,i} = \frac{z_{t,i}}{q_{t,i}} - \frac{z_{t-1,i}}{q_{t-1,i}} \le \frac{z_{t,i} - z_{t-1,i}}{q_{t,i}}.$$

The right-hand side is nonnegative because $z_{t,i}$ is nondecreasing. Hence the same inequality holds after taking the positive part. Summing these inequalities and applying Lemma D.1 proves the result. □

D.2 Proof of the Regret Theorem

The proof uses the time-varying metric decomposition in Duchi et al. (2011, Proposition 3 and the proof of Theorem 5), with Lemma D.2 controlling increases in the DA-AdaGrad denominator.

Proof of Theorem C.1. Let

$$e_{t,i} := w_{t,i} - w_i^\star.$$

Since $w_i^\star \in [\ell_i, u_i]$ and Euclidean projection onto an interval is nonexpansive,

$$e_{t+1,i}^2 \le \left(e_{t,i} - \eta \frac{g_{t,i}}{h_{t,i}} \right)^2 = e_{t,i}^2 - 2\eta \frac{g_{t,i} e_{t,i}}{h_{t,i}} + \eta^2 \frac{g_{t,i}^2}{h_{t,i}^2}.$$

Multiplying by $h_{t,i}/(2\eta)$ and rearranging gives

$$g_{t,i}e_{t,i} \le \frac{h_{t,i}}{2\eta}\left(e_{t,i}^2 - e_{t+1,i}^2\right) + \frac{\eta}{2}\frac{g_{t,i}^2}{h_{t,i}}. \tag{75}$$

For each coordinate, summation by parts yields

$$\begin{aligned}\sum_{t=1}^{T} h_{t,i}\left(e_{t,i}^2 - e_{t+1,i}^2\right) &= h_{1,i}e_{1,i}^2 + \sum_{t=2}^{T}\big(h_{t,i} - h_{t-1,i}\big)e_{t,i}^2 - h_{T,i}e_{T+1,i}^2 \\ &\le D_i^2\left[h_{1,i} + \sum_{t=2}^{T}\big(h_{t,i} - h_{t-1,i}\big)_+\right] \\ &\le D_i^2 U_i^{(\gamma)}, \end{aligned} \tag{76}$$

where the last inequality is Lemma D.2.

Convexity gives

$$f_t(w_t) - f_t(w^\star) \le \langle g_t, w_t - w^\star\rangle = \sum_{i=1}^{d} g_{t,i}e_{t,i}.$$

Summing (75) over t and i, applying (76), and using

$$S_i^{(\gamma)} = \sum_{t=1}^{T}\frac{g_{t,i}^2}{h_{t,i}}$$

gives

$$R_T \le \frac{1}{2\eta}\sum_{i=1}^{d} D_i^2 U_i^{(\gamma)} + \frac{\eta}{2}\sum_{i=1}^{d} S_i^{(\gamma)}.$$

This is (49). □

D.3 AdaGrad Recovery and the Fixed-Sequence Tradeoff

Proof of Corollary C.2. At $\gamma = 0$, $q_{t,i} = 1$ for every t and i. Therefore the correction term in (46) vanishes and

$$U_i^{(0)} = h_{T,i} = \sqrt{\varepsilon + A_{T,i}}.$$

Likewise,

$$S_i^{(0)} = \sum_{t=1}^{T}\frac{g_{t,i}^2}{\sqrt{\varepsilon + A_{t,i}}}.$$

Substitution into Theorem C.1 gives the stated unrelaxed diagonal AdaGrad bound.

Now take $\varepsilon = 0$ and evaluate the expressions for $\gamma > 0$ and $\gamma = 0$ on the same sequence. For each pair with $g_{t,i} \neq 0$, $a_{t,i} > 0$ and

$$h_{t,i} = \frac{a_{t,i}}{q_{t,i}}.$$

Terms with $g_{t,i} = 0$ contribute zero. Lemma D.1 therefore gives

$$U_i^{(\gamma)} = \sum_{t=1}^{T}\frac{\Delta a_{t,i}}{q_{t,i}} \le \sum_{t=1}^{T}\Delta a_{t,i} = a_{T,i} = U_i^{(0)},$$

because $q_{t,i} \ge 1$. Similarly,

$$S_i^{(\gamma)} = \sum_{t=1}^{T} q_{t,i}\frac{g_{t,i}^2}{a_{t,i}} \ge \sum_{t=1}^{T}\frac{g_{t,i}^2}{a_{t,i}} = S_i^{(0)}.$$

This proves (51). □

The next estimate is the shifted summation inequality of Auer et al. (2002, Lemma 3.5); its earlier conference version is credited in Duchi et al. (2011, Lemma 4).

Lemma D.3 (Standard AdaGrad accumulator estimate)**.** *Let $c_1, \dots, c_T \geq 0$ and $C_t := \sum_{s=1}^{t} c_s$. For every $\varepsilon > 0$,*

$$\sum_{t=1}^{T} \frac{c_t}{\sqrt{\varepsilon + C_t}} \leq 2\left(\sqrt{\varepsilon + C_T} - \sqrt{\varepsilon}\right). \tag{77}$$

Proof. The function $x \mapsto x^{-1/2}$ is decreasing. Hence, for every t,

$$\frac{c_t}{\sqrt{\varepsilon + C_t}} = \frac{(\varepsilon + C_t) - (\varepsilon + C_{t-1})}{\sqrt{\varepsilon + C_t}} \leq \int_{\varepsilon + C_{t-1}}^{\varepsilon + C_t} \frac{dx}{\sqrt{x}} = 2\left(\sqrt{\varepsilon + C_t} - \sqrt{\varepsilon + C_{t-1}}\right).$$

Summing over t telescopes and proves (77). □

E Proofs for the Direct Comparison with AdaGrad

This appendix proves Proposition C.3 and its local improvement condition. Throughout this section, $\varepsilon = 0$ and the completed sequence $\mathcal{T}$ is fixed as in Section C.2.

E.1 Minimization over the Scalar Learning Rate

For $P, Q > 0$ and $\eta > 0$, the arithmetic–geometric mean inequality gives

$$\frac{P}{2\eta} + \frac{\eta Q}{2} \geq \sqrt{PQ}.$$

Equality holds when $P/(2\eta) = \eta Q/2$, or equivalently when $\eta = \sqrt{P/Q}$. Taking $P = P_\gamma(\mathcal{T})$ and $Q = Q_\gamma(\mathcal{T})$ proves (54).

E.2 Exact Normalized Identity

Lemma E.1 (Exact normalization)**.** *The arrays $\{\pi^P_{t,i}\}_{(t,i)\in\mathcal{I}}$ and $\{\pi^Q_{t,i}\}_{(t,i)\in\mathcal{I}}$ are probability weights. Moreover, for every $\gamma \geq 0$,*

$$\frac{P_\gamma(\mathcal{T})}{P_0(\mathcal{T})} = \sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{-1/2}, \tag{78}$$

$$\frac{Q_\gamma(\mathcal{T})}{Q_0(\mathcal{T})} = \sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{1/2}. \tag{79}$$

Consequently, (57) *holds.*

Proof. If $g_{t,i} = 0$, then $A_{t,i} = A_{t-1,i}$ and therefore $\Delta a_{t,i} = 0$. Thus

$$\begin{aligned}
\sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i} &= \frac{1}{P_0(\mathcal{T})} \sum_{(t,i)\in\mathcal{I}} D_i^2 \Delta a_{t,i} \\
&= \frac{1}{P_0(\mathcal{T})} \sum_{i=1}^{d} D_i^2 \sum_{t=1}^{T} (a_{t,i} - a_{t-1,i}) \\
&= \frac{1}{P_0(\mathcal{T})} \sum_{i=1}^{d} D_i^2 a_{T,i} \\
&= 1.
\end{aligned}$$

Nonnegativity is immediate. Likewise,

$$\sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i} = \frac{1}{Q_0(\mathcal{T})} \sum_{(t,i)\in\mathcal{I}} \frac{g_{t,i}^2}{a_{t,i}} = 1.$$

Hence both arrays are probability weights.

Using the definitions of P_γ and π^P,

$$\begin{aligned}
P_\gamma(\mathcal{T}) &= \sum_{(t,i)\in\mathcal{I}} D_i^2 \Delta a_{t,i} (1+\gamma(\delta^{\max}_{t,i})^2)^{-1/2} \\
&= P_0(\mathcal{T}) \sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{-1/2},
\end{aligned}$$

which proves (78). Similarly,

$$Q_\gamma(\mathcal{T}) = \sum_{(t,i)\in\mathcal{I}} \frac{g_{t,i}^2}{a_{t,i}} (1+\gamma(\delta^{\max}_{t,i})^2)^{1/2}$$

$$= Q_0(\mathcal{T}) \sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{1/2},$$

which proves (79). Finally,

$$\frac{B^\star_\gamma(\mathcal{T})^2}{B^\star_0(\mathcal{T})^2} = \frac{P_\gamma(\mathcal{T})Q_\gamma(\mathcal{T})}{P_0(\mathcal{T})Q_0(\mathcal{T})} = \frac{P_\gamma(\mathcal{T})}{P_0(\mathcal{T})}\frac{Q_\gamma(\mathcal{T})}{Q_0(\mathcal{T})},$$

and substituting (78)–(79) gives (57). Because both balanced expressions are positive, $B^\star_\gamma(\mathcal{T}) < B^\star_0(\mathcal{T})$ if and only if the product on the right-hand side of (57) is smaller than one. □

E.3 Derivative at the AdaGrad Endpoint

Define

$$F_-(\gamma) := \sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{-1/2}, \tag{80}$$

$$F_+(\gamma) := \sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i}(1+\gamma(\delta^{\max}_{t,i})^2)^{1/2}, \tag{81}$$

and

$$\Phi(\gamma) := \log\frac{B^\star_\gamma(\mathcal{T})}{B^\star_0(\mathcal{T})} = \frac{1}{2}\log F_-(\gamma) + \frac{1}{2}\log F_+(\gamma). \tag{82}$$

The sums are finite and every $1+\gamma(\delta^{\max}_{t,i})^2$ is positive for $\gamma \geq 0$, so termwise differentiation is valid. We have

$$F'_-(\gamma) = -\frac{1}{2}\sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i}(\delta^{\max}_{t,i})^2(1+\gamma(\delta^{\max}_{t,i})^2)^{-3/2},$$

$$F'_+(\gamma) = \frac{1}{2}\sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i}(\delta^{\max}_{t,i})^2(1+\gamma(\delta^{\max}_{t,i})^2)^{-1/2}.$$

Because the two arrays are probability weights, $F_-(0) = F_+(0) = 1$. Therefore

$$\begin{aligned}\Phi'(0) &= \frac{1}{2}\frac{F'_-(0)}{F_-(0)} + \frac{1}{2}\frac{F'_+(0)}{F_+(0)} \\ &= -\frac{1}{4}\sum_{(t,i)\in\mathcal{I}} \pi^P_{t,i}(\delta^{\max}_{t,i})^2 + \frac{1}{4}\sum_{(t,i)\in\mathcal{I}} \pi^Q_{t,i}(\delta^{\max}_{t,i})^2 \\ &= -\frac{1}{4}(M_P - M_Q) \\ &= -\frac{\Delta_{\max}}{4}.\end{aligned}$$

This proves (58). Since $\Phi(0) = 0$ and Φ is differentiable at zero,

$$\Phi(\gamma) = -\frac{\Delta_{\max}}{4}\gamma + o(\gamma).$$

Thus $\Delta_{\max} > 0$ gives $B^\star_\gamma(\mathcal{T}) < B^\star_0(\mathcal{T})$ for all sufficiently small positive γ; the inequality reverses when $\Delta_{\max} < 0$.

F Experimental Details

Common protocol. The three synthetic experiments start from $w_1 = 0$ and compare AdaGrad with DA-AdaGrad. For coordinate i,

$$A_{t,i} = \sum_{s=1}^{t} g^2_{s,i}, \qquad \delta_{t,i} = |w_{t,i} - w_{1,i}|, \qquad \delta^{\max}_{t,i} = \max_{1\leq s\leq t}\delta_{s,i},$$

and the update is

$$w_{t+1,i} = w_{t,i} - \eta \frac{g_{t,i}}{\sqrt{10^{-12} + A_{t,i}/(1+\gamma(\delta_{t,i}^{\max})^2)}}.$$

Setting $\gamma = 0$ gives AdaGrad exactly. Neither method is given the optimum or the coordinatewise target distances; DA-AdaGrad uses only the displacement produced by its own iterates.

For an objective F with minimizer $w^\star$, we measure convergence by relative suboptimality

$$E_t := \frac{F(w_t) - F(w^\star)}{F(w_1) - F(w^\star)}.$$

Let S denote the trapezoidal integral of $\log_{10} E_t$ with respect to the normalized completed-update index $(t-1)/T$, from w_1 through w_{T+1}. Lower values indicate smaller relative errors over the update budget. For each method and condition, we select the learning rate with the smallest mean S over the tuning minibatch sequences. Both methods use the same learning-rate grid and minibatch sequences. We tune their rates separately because γ changes the update denominator. The selected rates are then held fixed during evaluation on 20 separate minibatch sequences, each shared by both methods. The reported advantage is

$$\Delta = S_{\text{AdaGrad}} - S_{\text{DA-AdaGrad}},$$

so $\Delta > 0$ favors DA-AdaGrad. Trajectory curves in Figures 5, 6, and 7 show the median E_t across evaluation sequences, with interquartile bands; summary panels show the median paired advantage. For numerical stability, the logarithm uses E_t clipped below at 10^{-12}. A tuning candidate is rejected if a recorded gap is nonfinite or exceeds 10^8 in the distance-heterogeneity experiment or 10^3 in the other two experiments.

F.1 Distance Heterogeneity at Fixed Curvature and Total Distance

The first experiment varies the dispersion of the coordinatewise target distances while holding curvature, total squared distance, initial suboptimality, and gradient noise fixed.

Loss and stochastic gradients. Let $m = 128$ and $N = 4096$. First draw $\widetilde{\xi}_j \sim \mathcal{N}(0, 0.8^2 I_m)$ and center the realized noise coordinatewise,

$$\xi_j = \widetilde{\xi}_j - \frac{1}{N}\sum_{k=1}^{N} \widetilde{\xi}_k.$$

For a target $w^\star$, define observations $Y_j = w^\star + \xi_j$ and minimize the empirical squared loss

$$F(w) = \frac{1}{2N}\sum_{j=1}^{N} \|w - Y_j\|_2^2. \tag{83}$$

Exact centering gives $N^{-1}\sum_j Y_j = w^\star$, and hence

$$F(w) - F(w^\star) = \frac{1}{2}\|w - w^\star\|_2^2, \qquad \nabla^2 F(w) = I_m.$$

Thus the empirical minimizer and Hessian are known exactly. A minibatch B_t contains 64 indices sampled with replacement, and its stochastic gradient is

$$g_t = w_t - w^\star - \frac{1}{|B_t|}\sum_{j \in B_t} \xi_j.$$

For this loss,

$$E_t = \frac{\|w_t - w^\star\|_2^2}{\|w^\star\|_2^2}.$$

Controlled distance construction. Write $d_i = |w_i^\star|$. A fixed set G of 16 coordinates is designated as the far group, with $\rho := |G|/m = 1/8$. The set G and the coordinate signs are sampled once and reused in every condition. For

$$r \in \{0.125, 0.25, 0.60, 0.90, 0.99\},$$

the far group is assigned fraction r of the total squared target distance by setting

$$d_i^2 = \begin{cases} r/\rho, & i \in G, \\ (1-r)/(1-\rho), & i \notin G. \end{cases}$$

Fixed signs are then applied to obtain $w^\star$. This construction gives

$$\frac{1}{m}\sum_{i=1}^{m} d_i^2 = 1 \quad \text{and} \quad F(0) - F(w^\star) = \frac{m}{2} = 64$$

in every condition. More explicitly, the mean squared distance is

$$\rho\frac{r}{\rho} + (1-\rho)\frac{1-r}{1-\rho} = 1.$$

Using the population variance across coordinates, the squared coefficient of variation is

$$\mathrm{CV}^2(d_i^2) = \rho\left(\frac{r}{\rho} - 1\right)^2 + (1-\rho)\left(\frac{1-r}{1-\rho} - 1\right)^2 = \frac{(r-\rho)^2}{\rho(1-\rho)}.$$

For $\rho = 1/8$ and $r \geq \rho$, this gives $\mathrm{CV}(d_i^2) = (8r-1)/\sqrt{7}$, which increases strictly with r. The five conditions have values 0, 0.378, 1.436, 2.343, and 2.616. The case $r = \rho$ has $d_i = 1$ for every coordinate and is the uniform-distance control. Increasing r concentrates a fixed total squared distance in the same 16 coordinates.

The same noise matrix and minibatch indices are used for both methods and all five values of r. Therefore the sweep changes only the distribution of target distance across coordinates, not the Hessian, initial suboptimality, noise realization, or minibatch order. For each condition, both methods select η from 31 geometrically spaced values in $[0.002, 1]$ using three shared 2000-update minibatch sequences with seeds 101–103. The data seed is 12 and the 20 evaluation seeds are 201–220. DA-AdaGrad uses $\gamma = 10$ in every condition. The selected rates are evaluated for 2000 updates on 20 separate minibatch sequences shared by both methods, with E_t recorded every five updates. Fixing γ prevents the trend across conditions from being produced by condition-specific retuning of γ.

The controlled quantities d_i are initial distances to the optimum, distinct from the trajectory-dependent radii $D_i = \max_t |w_{t,i} - w_i^\star|$ in the regret bound. Across the evaluation runs, the largest observed ratio D_i/d_i is 3.79. The full trajectories and paired advantage are shown in Figure 5, and Table 3 lists the selected learning rates.

Results and interpretation. DA-AdaGrad's median paired advantage increases with distance concentration, reaching $\Delta = 0.357$ at the greatest concentration. Since the Hessian, total squared distance, initial suboptimality, and stochastic-gradient noise are held fixed, this trend reflects the redistribution of distance across coordinates. It is consistent with the distance-dispersion motivation: coordinates that must travel farther can benefit from a stronger displacement adjustment even when curvature is identical in every direction. The global learning rate is selected separately in each condition, while $\gamma = 10$ remains fixed.

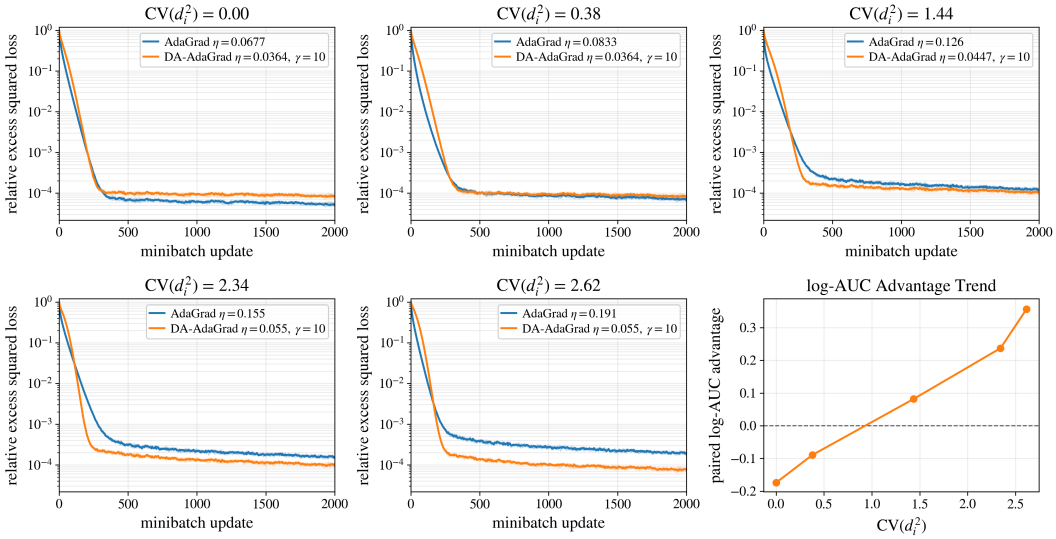


Figure 5: **Concentrating coordinate distance.** Five conditions redistribute the same total squared target distance; larger $\mathrm{CV}(d_i^2)$ means greater dispersion. Curves show median relative suboptimality over 20 paired streams, with interquartile bands. The last panel shows the median paired advantage Δ. Learning rates are selected separately; $\gamma = 10$ is fixed.

Table 3: Selected learning rates for the distance-heterogeneity sweep. Each method is tuned separately in every condition; DA-AdaGrad uses $\gamma = 10$ throughout. Values are rounded to six significant digits.

r	$\mathrm{CV}(d_i^2)$	AdaGrad η	DA-AdaGrad η
0.125	0	0.0676778	0.0363538
0.25	0.377964	0.0832553	0.0363538
0.6	1.43626	0.125992	0.0447214
0.9	2.34338	0.154992	0.0550149
0.99	2.61551	0.190667	0.0550149

F.2 Distance Magnitude at Fixed Curvature

The second experiment varies the overall distance to the optimum while holding the Hessian and the relative dispersion of squared coordinate distances fixed.

Data and target distances. Let $m = 50$ and $n = 4000$. Draw w^{true} from a standard normal distribution and rescale it so that $m^{-1}\sum_i |w_i^{\mathrm{true}}| = 3$. Then draw

$$x_j \sim \mathcal{N}(0, I_m/m), \qquad y_j = x_j^\top w^{\mathrm{true}} + 0.05\,\varepsilon_j, \qquad \varepsilon_j \sim \mathcal{N}(0,1),$$

using data seed 7. Define

$$H = \frac{X^\top X}{n}, \qquad b = \frac{X^\top y}{n}, \qquad z = H^{-1}b, \qquad e = y - Xz.$$

The design has full column rank and $X^\top e = 0$. Normalize the signed target profile as $u = z/\sqrt{m^{-1}\|z\|_2^2}$, so $m^{-1}\sum_i u_i^2 = 1$. To retain the distance levels of the earlier ridge construction, set

$$a_\lambda^2 = \frac{1}{m}\left\|(H+\lambda I)^{-1}b\right\|_2^2, \qquad \lambda \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 0\}.$$

Here λ only defines the five distance levels; no ridge penalty is used in this experiment. For $v_\lambda = a_\lambda u$, construct new targets $y_\lambda = Xv_\lambda + e$ and minimize

$$F_\lambda(w) = \frac{1}{2n}\|Xw - y_\lambda\|_2^2. \tag{84}$$

Since $X^\top e = 0$, the minimizer is $w_\lambda^\star = v_\lambda$, the Hessian is H, and

$$\frac{1}{m}\sum_i d_i^2 = a_\lambda^2, \qquad \mathrm{CV}(d_i^2) = \mathrm{CV}(u_i^2).$$

Thus all coordinate distances change by the same factor. Across the five conditions, the mean squared distance ranges from 0.00570757 to 15.5352, while $\mathrm{CV}(d_i^2) = 1.44893$, strong convexity $\mu = 0.0161664$, smoothness $L = 0.0245542$, and the condition number $L/\mu = 1.51884$ remain fixed.

Gradients, controls, and tuning. For a minibatch B_t of 256 indices sampled with replacement, write $H_{B_t} = X_{B_t}^\top X_{B_t}/|B_t|$. The stochastic gradient is

$$g_t = H_{B_t}(w_t - v_\lambda) - \frac{X_{B_t}^\top e_{B_t}}{|B_t|}.$$

We reuse X, e, and every minibatch stream across methods and conditions. In particular, gradient noise at the optimum is unchanged. Away from the optimum, feature-sampling noise depends on $w_t - v_\lambda$; scaling the target also changes the initial objective gap. We therefore measure relative suboptimality as

$$E_t = \frac{(w_t - v_\lambda)^\top H(w_t - v_\lambda)}{v_\lambda^\top H v_\lambda}.$$

The known optimum is used to construct the targets and measure error, not to set either optimizer's updates.

Each method selects its learning rate from 41 geometrically spaced values in $[0.003, 5]$ using two shared 1000-update minibatch streams, with seeds 1101 and 1102. DA-AdaGrad uses $\gamma = 1$ throughout. The selected rates are evaluated for 2000 updates on 20 separate streams with seeds 2101–2120, shared by both methods and all conditions. We record E_t after every update. Figure 6 shows all five conditions, and Table 4 lists the selected rates.

Results and interpretation. The median paired advantage rises from $\Delta = 0.000267$ at the smallest distance to 0.320 at mean squared distance 6.75939, and remains near 0.33 at the two largest distances. The advantage is positive in 19 of 20 pairs at the smallest distance and all 20 pairs at each remaining setting. This trend supports the distance-magnitude motivation with the Hessian and relative distance dispersion held fixed. It does not require a change in strong convexity.

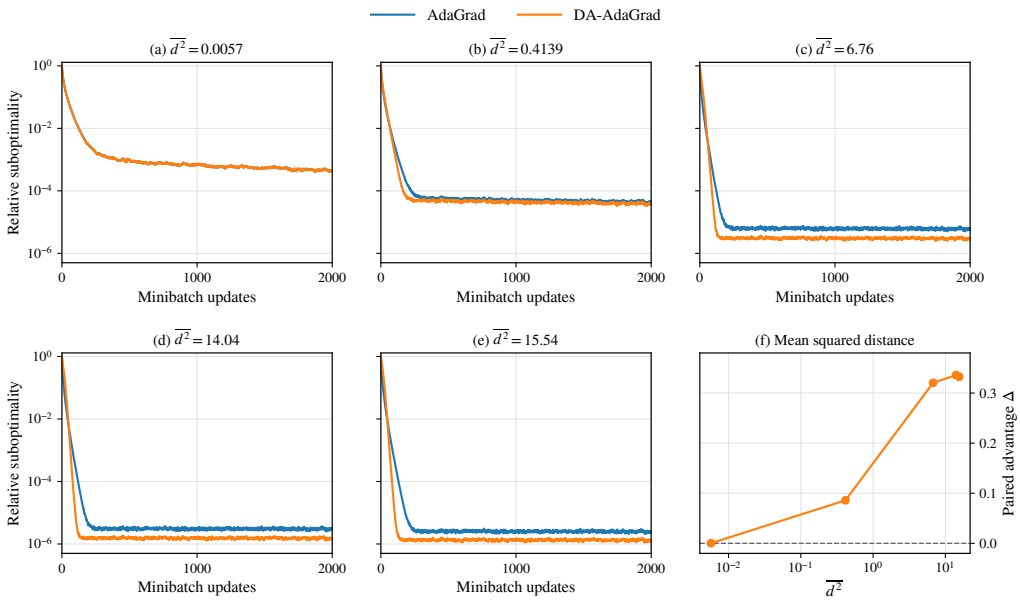


Figure 6: **Increasing distance at fixed curvature.** Five target scales preserve the Hessian and $\mathrm{CV}(d_i^2)$. Curves show median relative suboptimality over 20 paired streams, with interquartile bands. The last panel shows the median paired advantage Δ. Learning rates are selected separately; $\gamma = 1$ is fixed.

Table 4: Selected learning rates for the distance-magnitude sweep. The parameter λ defines the target scale only; all five objectives are unpenalized. Each method is tuned separately, with $\gamma = 1$ fixed for DA-AdaGrad. Values are rounded to six significant digits.

λ	$m^{-1}\sum_i d_i^2$	AdaGrad η	DA-AdaGrad η
1	0.00570757	0.0109887	0.0109887
0.1	0.413877	0.122474	0.101742
0.01	6.75939	0.650072	0.257176
0.001	14.0426	0.942007	0.309583
0	15.5352	0.942007	0.309583

F.3 Curvature at Fixed Target Distances

The third experiment changes curvature while fixing every coordinate's initial distance to the optimum. Using the same X and residual e as above, fix $v = z$ and consider

$$F_\lambda(w) = \frac{1}{2n}\|Xw - (Xv + e)\|_2^2 + \frac{\lambda}{2}\|w - v\|_2^2, \qquad \lambda \in \{1, 10^{-1}, 10^{-2}, 10^{-3}, 0\}. \tag{85}$$

The penalty is centered at v, so $w^\star = v$ in every condition. Consequently, the mean squared distance remains 15.5352 and $\mathrm{CV}(d_i^2)$ remains 1.44893. The Hessian and the strong-convexity and smoothness parameters are

$$\nabla^2 F_\lambda = H + \lambda I, \qquad \mu_\lambda = \sigma_{\min}(H) + \lambda, \qquad L_\lambda = \sigma_{\max}(H) + \lambda.$$

Thus the sweep changes strong convexity, smoothness, and conditioning, while preserving the minimizer. It does not vary strong convexity alone.

The stochastic gradient is

$$g_t = (H_{B_t} + \lambda I)(w_t - v) - \frac{X_{B_t}^\top e_{B_t}}{|B_t|},$$

and relative suboptimality is

$$E_t = \frac{(w_t - v)^\top (H + \lambda I)(w_t - v)}{v^\top (H + \lambda I) v}.$$

Gradient noise at the optimum is fixed, but the initial gradient and objective gap change with λ. The optimizer receives only the minibatch gradient; it is not given v as distance information. We use the same learning-rate grid, tuning and evaluation seeds, update budgets, batch size, and fixed $\gamma = 1$ as in the distance-magnitude experiment. Learning rates are selected separately for each method and penalty. At $\lambda = 0$, this experiment agrees with the largest-distance condition above.

Results and interpretation. DA-AdaGrad has a positive paired advantage in all 20 evaluation pairs at each penalty. Its median advantage is not monotone in curvature: it is 0.239 at $\lambda = 1$, reaches 0.678 at $\lambda = 0.1$, and decreases to 0.333 at $\lambda = 0$. These results do not support a simple claim that weaker strong convexity necessarily produces a larger advantage. Figure 7 shows the full trajectories and paired advantage, and Table 5 gives the selected rates.

Table 5: Selected learning rates for the curvature sweep. Each method is tuned separately at every penalty, with $\gamma = 1$ fixed for DA-AdaGrad. Values are rounded to six significant digits.

λ	μ_λ	AdaGrad η	DA-AdaGrad η
1	1.01617	2.86636	0.448610
0.1	0.116166	1.64321	0.309583
0.01	0.0261664	1.13397	0.309583
0.001	0.0171664	0.942007	0.309583
0	0.0161664	0.942007	0.309583

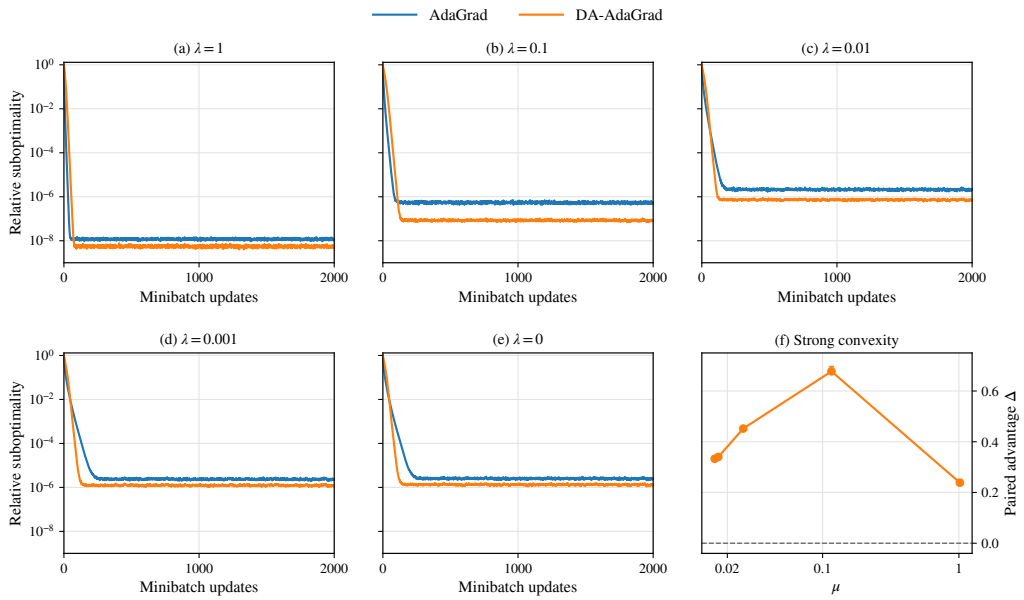


Figure 7: **Varying curvature at fixed distances.** The centered penalty preserves the optimum and all coordinate distances while changing $\mu_\lambda = \sigma_{\min}(H) + \lambda$. Curves show median relative suboptimality over 20 paired streams, with interquartile bands. The last panel shows the median paired advantage Δ. Learning rates are selected separately; $\gamma = 1$ is fixed.

F.4 Nonconvex Objectives on LIBSVM Datasets

Data and objectives. We use the binary covtype, epsilon, and RCV1 datasets from the LIBSVM collection (LIBSVM Data, n.d.). Each model is a linear predictor $x^\top w$ without an intercept, initialized at $w_1 = 0$. Covtype examples are normalized to unit Euclidean norm. Epsilon uses the supplied normalized features, and RCV1 retains the supplied representation. The official RCV1 training and test files are pooled for this finite-data optimization comparison. Table 6 gives the problem sizes and update budgets.

For labels $y_j \in \{-1, 1\}$, define the signed margin $z_j = y_j x_j^\top w$. The two objectives are

$$F(w) = \frac{1}{n}\sum_{j=1}^{n} \ell(z_j) + \rho\|w\|_1, \qquad \rho = \frac{0.1}{n}, \tag{86}$$

with

$$\ell_{\text{sigmoid}}(z) = \frac{1}{1+e^z}, \qquad \ell_{\text{ramp}}(z) = \min\{2, \max\{0, 1-z\}\}.$$

The sigmoid loss is the bounded margin loss, rather than logistic cross-entropy. Both objectives are nonconvex. Each update samples one example uniformly with replacement and uses its loss gradient together with the full penalty subgradient $\rho\,\text{sign}(w)$, with $\text{sign}(0) = 0$. The ramp derivative is taken to be zero at its two breakpoints. Computation uses float64 arithmetic. There is no proximal thresholding, additional weight decay, gradient clipping, or external learning-rate schedule.

Table 6: LIBSVM problem sizes and optimization budgets. Each dataset is evaluated with both losses in (86).

Dataset	Examples n	Features	Updates
covtype	581,012	54	10,000,000
epsilon	400,000	2,000	10,000,000
RCV1	697,641	47,236	1,000,000

Optimizers. AdaGrad and DA-AdaGrad use $\varepsilon = 10^{-8}$ inside the square root, so their denominators are respectively $\sqrt{\varepsilon + A_{t,i}}$ and $\sqrt{\varepsilon + A_{t,i}/[1 + \gamma(\delta_{t,i}^{\max})^2]}$. The comparison includes AdaGrad++ and Adam++ Case 2 (Tao et al., 2026), DoG (Ivgi et al., 2023), and Prodigy (Mishchenko & Defazio, 2024). The ++ methods use a global RMS displacement radius initialized at 10^{-6}; Adam++ uses moment decays $(0.9, 0.999)$, a running maximum second moment, and no moment bias correction. Its denominator retains the Case 2 factor $\sqrt{t v_{t,i}^{\max}}$. AdaGrad++ uses $\varepsilon = 10^{-10}$ outside the square root, and Adam++ uses 10^{-8} outside it.

DoG uses a global Euclidean displacement radius initialized at 10^{-6}, a cumulative squared-gradient norm with offset 10^{-8}, and scalar multiplier one. Its reported objectives and accuracies use the power-eight polynomial average, updated after step t with weight $9/(t+8)$; this average does not enter the training updates. The other methods report their final iterates. Prodigy uses its dense Adam form, moment decays $(0.9, 0.999)$, $\beta_3 = \sqrt{0.999}$, initial distance estimate 10^{-6}, and $\varepsilon = 10^{-8}$. Moment bias correction, the warmup safeguard, and distance-estimation subsampling are disabled. Its learning-rate multiplier remains one while its distance coefficient is tuned.

Selection and evaluation. Initial screening uses $100{,}000$ updates with seed zero and evaluates the final objective on a fixed subset of 8192 examples. The two best screened candidates are evaluated at the full budget on seeds zero and one; we select the configuration with the lower mean final objective, including the penalty. Previously completed pairs of full-budget AdaGrad tuning runs remain eligible during learning-rate refinement. Final evaluation uses seeds 100–104, with the same sampled example sequence shared across methods for each seed. DoG uses the fixed configuration above without screening. Neither accuracy on the evaluation seeds nor area under the objective curve is used for hyperparameter selection.

AdaGrad is tuned independently, with geometric learning-rate proposals and local midpoint refinement until the available neighbors of the best pilot rate differ from it by at most $2^{1/4}$. The DA-AdaGrad search uses the selected AdaGrad rate divided by $1, \sqrt{2}, 2, 2\sqrt{2}, 4$, and the ordered gamma candidates $0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30$. At the largest learning rate, gamma increases until the first failure to improve the best matched-budget pilot score. A lower learning rate starts from the last successful gamma, continues through non-improvements until its first improvement or the end of the chain, and stops after the first subsequent non-improvement. All five learning rates are visited, and the best two finite DA candidates are eligible for full-budget tuning.

AdaGrad++ and Adam++ independently screen scalar multipliers $0.5, 1, 2$; Prodigy starts with distance coefficients $0.25, 0.5, 1$. Their searches extend a winning endpoint by factors of two, with at most four added values, then evaluate one geometric midpoint on each available side of the leading candidate. Table 7 lists the selected values. After tuning, we evaluate objective values and accuracy on the full dataset. Accuracy is therefore measured in-sample; it does not assess performance on held-out data. Table 8 reports means and standard errors over the five evaluation runs.

Table 7: Selected LIBSVM settings. The ++ columns are scalar multipliers; the Prodigy column is its distance coefficient. Prodigy and DoG retain learning-rate multiplier one. Values are rounded to six significant digits.

Dataset	Loss	AdaGrad η	DA η	DA γ	AdaGrad++	Adam++	Prodigy
covtype	Sigmoid	5.96104	2.98052	0.01	2	0.707107	0.015625
covtype	Ramp	0.404071	0.285721	1	0.5	0.707107	0.5
epsilon	Sigmoid	3.44161	1.21679	0.1	1.41421	0.707107	0.0220971
epsilon	Ramp	1.73205	0.433013	1	1	0.707107	0.015625
RCV1	Sigmoid	1.35373	0.676865	0.3	2.82843	2.82843	0.125
RCV1	Ramp	0.706519	0.353259	1	2	2.82843	0.015625

Table 8: **Final LIBSVM results.** Mean ± standard error over five runs. Objectives include the L_1 penalty, and accuracy is in-sample. Bold entries mark the lowest mean objective or highest mean accuracy within each dataset and loss.

Dataset	Method	Sigmoid + L_1		Ramp + L_1	
		Objective	Accuracy	Objective	Accuracy
Covtype	AdaGrad	0.395425 ± 0.000083	62.345 ± 0.041%	0.828384 ± 0.001475	59.482 ± 0.092%
	DA-AdaGrad	**0.308170 ± 0.000256**	**74.014 ± 0.049%**	**0.544927 ± 0.000725**	**75.701 ± 0.088%**
	Prodigy	0.339398 ± 0.005775	66.578 ± 0.600%	0.975210 ± 0.000005	51.240 ± 0.000%
	Adam++	0.331509 ± 0.006997	67.257 ± 0.727%	0.727445 ± 0.026252	64.394 ± 1.309%
	AdaGrad++	0.409979 ± 0.000593	59.440 ± 0.060%	0.810356 ± 0.001440	59.823 ± 0.072%
	DoG	0.406123 ± 0.000081	59.613 ± 0.010%	0.810247 ± 0.000082	59.649 ± 0.008%
epsilon	AdaGrad	0.105565 ± 0.000151	90.350 ± 0.023%	0.200780 ± 0.000189	90.344 ± 0.012%
	DA-AdaGrad	**0.103610 ± 0.000107**	**90.445 ± 0.017%**	**0.196465 ± 0.000168**	**90.456 ± 0.010%**
	Prodigy	0.113618 ± 0.000757	89.807 ± 0.077%	0.210844 ± 0.002736	89.726 ± 0.137%
	Adam++	0.105978 ± 0.000407	90.330 ± 0.041%	0.201990 ± 0.001152	90.310 ± 0.042%
	AdaGrad++	0.113180 ± 0.001861	89.798 ± 0.193%	0.203957 ± 0.002958	90.055 ± 0.155%
	DoG	0.103715 ± 0.000014	90.419 ± 0.006%	0.197373 ± 0.000045	90.435 ± 0.005%
RCV1	AdaGrad	0.028562 ± 0.000030	98.020 ± 0.007%	0.046135 ± 0.000506	98.039 ± 0.026%
	DA-AdaGrad	**0.027096 ± 0.000046**	97.962 ± 0.005%	**0.044794 ± 0.000299**	98.040 ± 0.015%
	Prodigy	0.040740 ± 0.000146	96.375 ± 0.013%	0.068050 ± 0.000200	96.898 ± 0.013%
	Adam++	0.031825 ± 0.000075	97.729 ± 0.012%	0.052208 ± 0.000214	97.816 ± 0.010%
	AdaGrad++	0.029860 ± 0.000064	**98.034 ± 0.006%**	0.045282 ± 0.000109	**98.154 ± 0.008%**
	DoG	0.028506 ± 0.000030	97.855 ± 0.005%	0.046655 ± 0.000110	97.960 ± 0.006%

F.5 Image Classification

CIFAR experiments. We train a CIFAR-stem ResNet-18 for 50 epochs, with batch size 128, moment decays $(0.9, 0.999)$, $\epsilon = 10^{-8}$, zero weight decay, and a cosine learning-rate schedule with a floor at 5% of the initial rate and no warmup. Both datasets use random crops and horizontal flips; CIFAR-100 additionally uses Cutout of side length 8 and Mixup with parameter 0.2. Final fits use all 50,000 training images. Table 1 reports the mean and sample standard deviation of each run's maximum accuracy on the 10,000-image official test set, over 25 seeds for CIFAR-10 and ten for CIFAR-100. The DA-AdamW runs use the running maximum of displacement in the denominator, as defined in the main text.

CIFAR-10 selection. We use a fixed 45,000/5,000 training/validation split. Initial screening uses one seed and the first 20 epochs of the 50-epoch schedule, ranking epoch-20 validation accuracy. AdamW is screened at $2.5 \times 10^{-4} 2^{k/4}$, $k = 0, \dots, 16$; its selected rate $\eta_0 = 10^{-3}/\sqrt{2}$ is the baseline reported in the table. DA-AdamW is independently screened at $\eta_0 2^{-j/2}$, $j = 0, \dots, 4$, with $\gamma \in \{30, 100, 300, 1000, 3000\}$. A later confirmation trains the top three AdamW candidates afresh for 50 epochs on three seeds. Its winner, $\eta_p = 10^{-3} 2^{3/4}$, supplies the two rates $\{\eta_p, \eta_p/\sqrt{2}\}$ for an additional DA-AdamW screen with the same gamma grid. The top three new DA candidates and two earlier candidates are then trained afresh for 50 epochs on the same three seeds. Mean epoch-50 validation accuracy selects $\eta = 10^{-3}/\sqrt{2}$ and $\gamma = 3000$. The reported AdamW result retains the original screening winner; compatible existing DA final runs are reused. Both reported methods use final seeds 1103–1127.

CIFAR-100 selection. AdamW is screened at $10^{-3} 2^{k/4}$, $k = 0, \dots, 4$, on two seeds through epoch 20 of the 50-epoch schedule. The top three configurations continue to epoch 50, or reuse completed runs from the same trajectories, selecting $\eta_0 = 10^{-3}$ at the lower bound of the grid. DA-AdamW is screened at $\eta_0\{2^{-1/4}, 2^{-1/2}\}$ with $\gamma \in \{1000, 3000\}$ and uses the same top-three promotion. Both stages rank the mean of each seed's maximum accuracy within the allotted epochs. The 50-epoch winner is $\eta = 10^{-3}/\sqrt{2}$, $\gamma = 3000$. The official test set is used for both hyperparameter selection and reporting. The ten final seeds are 1103–1112, including the two selection seeds; compatible earlier runs are reused.

ImageNet-100 experiment. ViT-S/16 is trained from scratch for 300 epochs with all 21,704,164 parameters updated by the chosen optimizer. AdamW uses learning rate 2.5×10^{-4}; DA-AdamW uses $2.5 \times 10^{-4}/\sqrt{2}$ and constant $\gamma = 10000$. Both use moment decays $(0.9, 0.999)$, $\epsilon = 10^{-8}$, and weight decay 0.05, excluding biases and one-dimensional parameters. DA-AdamW uses anchor decay 0.999.

Both methods use a global batch size of 256 on two A100 GPUs, bf16 autocasting, gradient clipping at norm 1, and a 300-epoch cosine schedule with five warmup epochs. The DeiT augmentation recipe (Touvron et al., 2021) includes RandAugment, Mixup/CutMix, random erasing, label smoothing 0.1, and stochastic depth 0.1.

Hyperparameters are selected using two seeds and 20-epoch pilots under a 100-epoch cosine schedule. The AdamW search expands from five rates between 10^{-5} and 10^{-4} to ten rates between 10^{-5} and 10^{-3}. DA-AdamW tests the selected AdamW rate and that rate divided by $\sqrt{2}$, with $\gamma \in \{3000, 10000\}$. Each method selects the highest mean per-seed best validation accuracy over the 20 pilot epochs. Final runs restart from initialization on paired seeds 0–2 using the 300-epoch schedule; seeds 0 and 1 also appeared in tuning. Table 1 reports the maximum validation accuracy within each final run. Figure 3 shows mean accuracy with one sample standard deviation, without smoothing, for epochs 100–300 inclusive.

F.6 Machine Translation

We use the prepared IWSLT14 German–English data with 160,239 training pairs, 7,283 validation pairs, and 6,750 test pairs. The Transformer has six encoder and six decoder layers, model width 512, four attention heads, feedforward width 1024, dropout 0.3, and label smoothing 0.1. Training uses an effective batch of 4096 padded tokens, a 20-epoch cosine schedule, and 4000 warmup updates.

AdamW learning rate is tuned over $5 \times 10^{-4}\{\sqrt{2}, 1, \frac{\sqrt{2}}{2}\}$, with expansion by a factor of $\sqrt{2}$ if the best run falls at the boundary, using five-epoch pilots on seeds 0 and 1, following the same 20-epoch schedule as the final runs. DA-AdamW tested at the selected AdamW rate and that rate divided by $\sqrt{2}$, each with $\gamma \in \{1.7 \times 10^4, 9 \times 10^4\}$. Unlike in previous experiments, these two values are calibrated once for target learning rate gains of $1.25\times$ and $2\times$ at the 75th-percentile of the selected AdamW pilot's running-max EMA displacement; γ then remains constant. Each method selects the lowest mean validation NLL at epoch five; configurations with a failed pilot are excluded. Both select learning rate 5×10^{-4}, and DA-AdamW selects $\gamma \approx 1.7 \times 10^5$. Both use moment decays $(0.9, 0.98)$, $\epsilon = 10^{-8}$, and weight decay 10^{-4}. The selected pilots on seeds 0 and 1 are continued to epoch 20, and seeds 2–5 are trained from initialization.

For each run, we evaluate the checkpoint with the lowest validation NLL, which occurred at epoch 19 or 20. NLL is the unsmoothed negative log-likelihood per nonpadding target token. Translations are generated with beam size 5 and length penalty 1. After removing BPE, we compute BLEU on lowercased, tokenized test text using SacreBLEU (Post, 2018) with tokenization disabled. Figure 4 shows the individual test results and their means with one sample standard deviation.

G Equivalence of the EMA Anchor and a Leaky Update Sum

Proposition G.1. *Let* $u_0 = w_1$ *and*

$$u_t = \beta u_{t-1} + (1-\beta) w_t, \qquad 0 \le \beta < 1.$$

Define $d_t := w_t - u_{t-1}$. *Then* $d_1 = 0$ *and*

$$d_{t+1} = (w_{t+1} - w_t) + \beta d_t. \tag{87}$$

Consequently,

$$d_t = \sum_{s=1}^{t-1} \beta^{t-1-s} (w_{s+1} - w_s), \qquad t \ge 2. \tag{88}$$

Proof. Because $u_0 = w_1$, $d_1 = w_1 - u_0 = 0$. Moreover,

$$\begin{aligned} d_{t+1} &= w_{t+1} - u_t = w_{t+1} - \beta u_{t-1} - (1-\beta) w_t \\ &= (w_{t+1} - w_t) + \beta (w_t - u_{t-1}) = (w_{t+1} - w_t) + \beta d_t, \end{aligned}$$

which proves (87). Iterating this recurrence from $d_1 = 0$ gives (88). Equivalently, the formula follows by induction: it is immediate for $t = 2$, and substitution of the induction hypothesis into (87) advances it from t to $t + 1$. □