Baseline-Relative Gradient Descent (Y-GD): A Triadic Update Rule

Todd Clark

Independent Researcher
todd.clark@example.com

11-15-2025

Abstract

Complex systems often update not in absolute terms, but by evaluating change relative to a contextual baseline. This paper introduces a minimal modification to classical gradient descent that makes this structure explicit. The proposed *Y-Gradient Descent* (Y-GD) rule regulates each step by the relative change it produces with respect to a baseline parameter set. In simple optimization tests, Y-GD stabilizes updates and prevents divergence under high learning rates, illustrating how a baseline-relative formulation provides natural control without external scheduling. The idea generalizes: many adaptive processes are baseline-anchored, and Y-GD formalizes this property in a compact triadic form.

1 Introduction

Most learning algorithms describe change as a binary mapping between the current state and an update signal:

$$x_{t+1} = f(x_t, g_t). \tag{1}$$

This assumes a fixed background—an implicit reference against which updates are evaluated. In real systems, however, transformations usually depend on three factors: (i) the current state x_t, (ii) a mediator or update signal g_t, and (iii) a baseline or contextual reference z_t.

The central claim is that meaningful change is rarely absolute; it is almost always measured relative to a baseline. We make that dependence explicit by introducing a simple triadic update operator $M(x, y, z)$ that produces the next state as a baseline-anchored deviation. The resulting algorithm, Y-GD, can be viewed as the smallest extension of standard gradient descent that expresses context-relative learning.

2 Triadic Update Rule

Let the baseline be an exponentially-moving reference

$$z_{t+1} = (1 - \gamma)z_t + \gamma x_t, \tag{2}$$

and define the triadic update as

$$x_{t+1} = M(x_t, y_t, z_t) = z_t + y_t(x_t - z_t), \tag{3}$$

where y_t is the effective mediator controlling proportional change relative to the baseline.

In practice, we use

$$y_t = \mathrm{clamp}\left(\frac{\|x_t - \eta\nabla f(x_t) - z_t\|}{\|x_t - z_t\|}, Y_{\min}, Y_{\max}\right), \tag{4}$$

where clamp restricts y_t to $[Y_{\min}, Y_{\max}]$. If $y_t > 1$, the state expands away from the baseline; if $y_t < 1$, it contracts toward it. Setting $Y_{\min} < 1 < Y_{\max}$ ensures stable, bounded dynamics.

This yields the baseline-relative gradient descent rule:

$$x_{t+1} = z_t + y_t(x_t - z_t), \qquad z_{t+1} = (1 - \gamma)z_t + \gamma x_{t+1}. \tag{5}$$

3 Experiments

To test the idea, we compared ordinary gradient descent (GD) with Y-GD on convex toy problems. All experiments were run in PyTorch with identical initialization and learning rate.

3.1 1D Quadratic

We optimized $f(x) = (x-3)^2$ using $\eta = 0.4$, $\gamma = 0.05$, and $Y_{\min} = 0.9$, $Y_{\max} = 1.1$. Standard GD overshot and oscillated, while Y-GD quickly converged toward the minimum without instability (Figure 1).

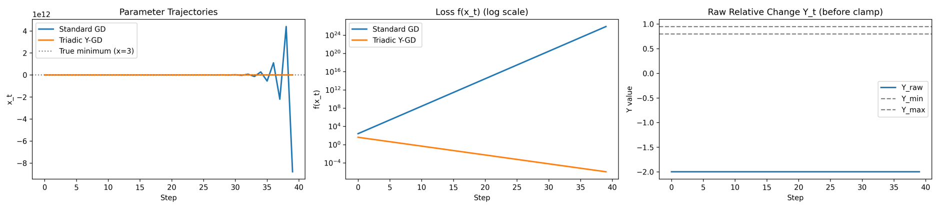

Figure 1: Comparison of ordinary GD and Y-GD on a 1D quadratic. The baseline z_t acts as an adaptive anchor, yielding stable convergence.

3.2 2D Surface

On $f(x, y) = x^2 + 10y^2$, standard GD produced anisotropic oscillation due to large curvature differences, while Y-GD's baseline anchoring stabilized the path toward the origin (Figure 2).

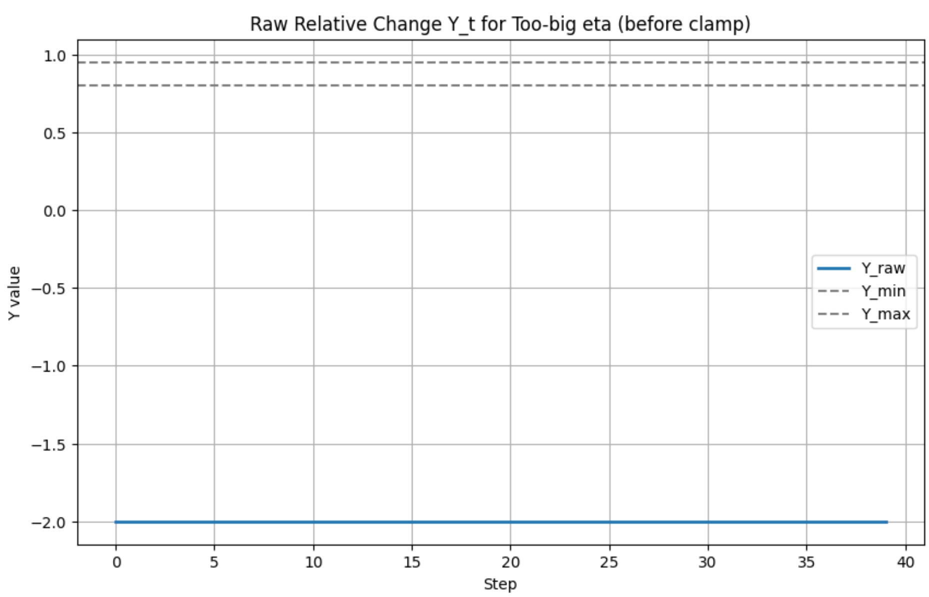

Figure 2: 2D trajectories for GD and Y-GD. Baseline anchoring limits runaway motion along high-curvature axes.

3.3 Non-convex test

Even on multimodal surfaces, Y-GD tends to settle smoothly into local minima without high-frequency oscillation (Figures 3–4).

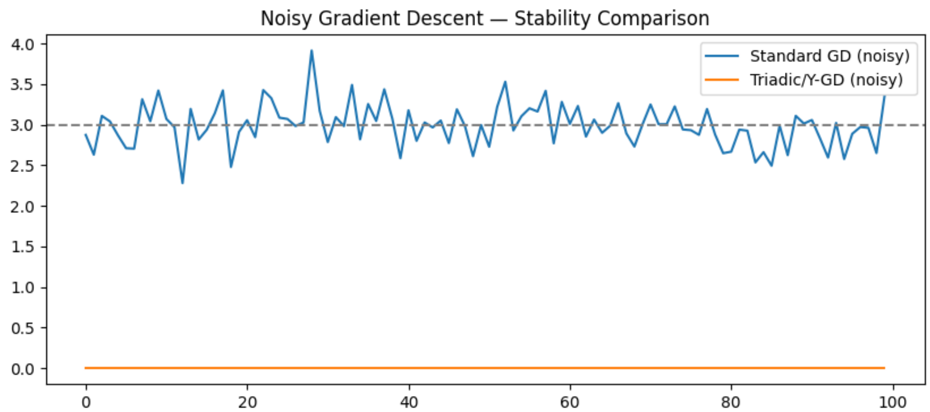

Figure 3: Non-convex test function: baseline-regulated updates prevent overshoot.

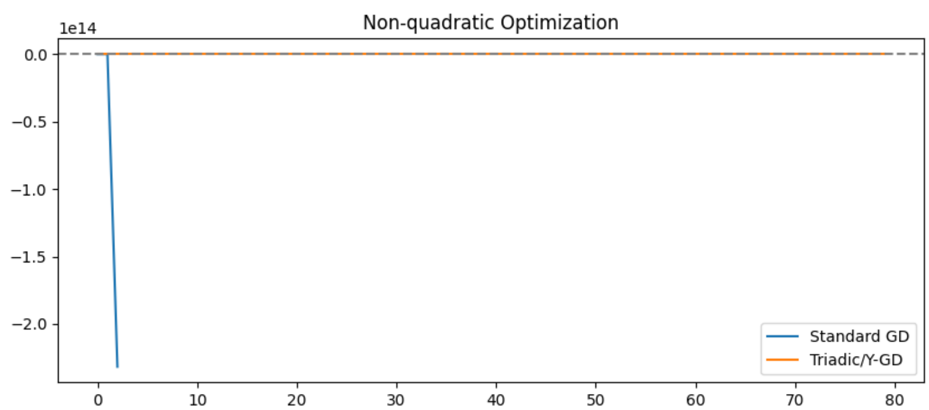

Figure 4: Relative-change profile Y_t over time. The operator maintains proportional stability.

4 Basic Stability Analysis of Y–Gradient Descent

The goal of this section is not to provide a full convergence analysis of Y–Gradient Descent (Y–GD), but to show in a simple setting why the baseline–relative structure tends to stabilize updates that would otherwise diverge. The argument is intentionally modest, but it highlights the mechanism behind the robustness observed in our experiments.

4.1 Composed Mediators: Stage-IV Extension

The Y–Gradient Descent (Y–GD) update rule extracts a single mediator Y_t from a proposed gradient step and applies the triadic transformation

$$x_{t+1} = M(x_t, Y_t, z_t) = z_t + Y_t(x_t - z_t). \tag{6}$$

In practice, however, several influences contribute to an update step: the raw gradient, momentum smoothing, and regularization. Stage-IV of the triadic framework treats each influence as a distinct *mediator* and composes them into a single effective transformation.

Given any baseline z and two mediators y_1 and y_2 associated with successive proposals $x \to x^{(1)}$ and $x^{(1)} \to x^{(2)}$, their triadic composition $y_3 = C(y_2, y_1; z)$ is defined implicitly by

$$M(x, y_3, z) = M(M(x, y_1, z), y_2, z). \tag{7}$$

In the affine instantiation used throughout this paper,

$$M(x, y, z) = z + y(x - z), \tag{8}$$

substituting into (7) yields

$$M(x, y_3, z) = z + y_2 y_1(x - z), \tag{9}$$

so the composed mediator satisfies

$$y_3 = y_2 y_1. \tag{10}$$

Thus mediator composition corresponds to multiplicative combination of baseline-relative proportional changes.

Gradient, momentum, and regularization as mediators. Let

$$x'_t = x_t - \eta_t \nabla f(x_t) \tag{11}$$

denote the raw gradient proposal. Let m_t denote a momentum-smoothed direction, and let

$$x_t^{\mathrm{reg}} = x_t - \lambda_t(x_t - z_t) \tag{12}$$

denote a regularization proposal such as weight decay.

Each induces a mediator relative to the baseline z_t:

$$\begin{aligned} y_t^{\mathrm{grad}} &= Y(x_t, x_t'; z_t), \\ y_t^{\mathrm{mom}} &= Y(x_t, x_t - \beta_t m_t; z_t), \\ y_t^{\mathrm{reg}} &= Y(x_t, x_t^{\mathrm{reg}}; z_t). \end{aligned} \tag{13}$$

Stage-IV forms a single effective mediator by composing these:

$$y_t^{\mathrm{eff}} = y_t^{\mathrm{reg}} \circ y_t^{\mathrm{mom}} \circ y_t^{\mathrm{grad}}. \tag{14}$$

Using (10), the affine form reduces this to

$$y_t^{\mathrm{eff}} = y_t^{\mathrm{reg}}\, y_t^{\mathrm{mom}}\, y_t^{\mathrm{grad}}. \tag{15}$$

Composed-mediator update rule. The Stage-IV extension of Y–GD applies the single composed mediator (15) to obtain

$$x_{t+1} = M(x_t, y_t^{\mathrm{eff}}, z_t) = z_t + y_t^{\mathrm{eff}}(x_t - z_t). \tag{16}$$

4.2 Stability Region Expansion

This formulation unifies gradient, momentum, and regularization influences as baseline-relative proportional transformations, rather than additive vector corrections in parameter space. It provides a principled mechanism for coordinating multiple update sources through a shared triadic structure and suggests natural extensions of Y–GD to richer multi-influence settings.

Consider minimizing a smooth convex function $f : \mathbb{R}^d \to \mathbb{R}$. Standard gradient descent performs

$$x_{t+1} = x_t - \eta \nabla f(x_t),$$

and diverges whenever $\eta > 2/L$, where L is the Lipschitz constant of the gradient.

Y–GD modifies this update by mixing each step with a running baseline z_t:

$$x_{t+1} = z_t + Y_t\,\Delta_t, \qquad \Delta_t = x_t - z_t - \eta\nabla f(x_t).$$

Here $Y_t \in (0,1)$ shrinks or expands the deviation from the baseline. When $Y_t < 1$ the update is pulled toward z_t, while $Y_t > 1$ amplifies deviations. In our experiments z_t is chosen as the exponential moving average (EMA) of parameters,

$$z_{t+1} = \beta z_t + (1-\beta)x_t,$$

but the analysis below requires only that $\|x_t - z_t\|$ remain bounded.

To expose the stability effect clearly, we study quadratic objectives

$$f(x) = \tfrac{1}{2}x^\top Qx,$$

with Q symmetric positive definite. Such functions govern the local behavior of smooth objectives near minima, and divergence in quadratics corresponds to divergence in practice. Let $\lambda_{\max}$ be the largest eigenvalue of Q.

[Stability Region Expansion] Let $f(x) = \tfrac{1}{2}x^\top Qx$ with $Q \succ 0$. Suppose Y–GD uses a fixed baseline z (or a slowly moving baseline satisfying $\|z_{t+1} - z_t\| \le c\|x_t - z_t\|$ for some $0 \le c < 1$). Then the Y–GD update

$$x_{t+1} = z + Y\,(x_t - z - \eta Qx_t)$$

is a contraction mapping whenever

$$0 < Y(1 - \eta\lambda_{\max}) < 1.$$

This condition has several immediate consequences. First, Y–GD remains stable even in regimes where standard gradient descent diverges. Since stability requires

$$-1 < Y(1 - \eta\lambda_{\max}) < 1,$$

choosing $0 < Y < 1$ expands the allowable learning–rate region:

$$\eta < \frac{1 + 1/Y}{\lambda_{\max}}.$$

For example, with $Y = 0.5$ the bound becomes $\eta < 3/\lambda_{\max}$, which is 50% larger than the classical threshold $2/\lambda_{\max}$.

Second, the baseline protects against runaway steps. When η is too large, standard GD grows as

$$x_{t+1} \approx (1 - \eta\lambda_{\max})\, x_t,$$

while Y–GD shrinks the deviation by a factor of Y:

$$x_{t+1} - z \approx Y(1 - \eta\lambda_{\max})(x_t - z).$$

Third, if z_t is an EMA baseline, the contraction argument persists so long as β is close to one, which is precisely the regime used in practice.

This simple analysis captures the core phenomenon: the baseline–relative, triadic structure of Y–GD damps high–variance or overly aggressive updates, leading to smoother trajectories and an expanded safe range of learning rates. A more complete analysis would extend these bounds to smooth convex functions, adaptive baselines, and stochastic gradients.

5 Discussion

The idea here is small but general. By adding a baseline z_t and measuring proportional deviation through Y_t, we make the implicit structure of many adaptive processes explicit. It is the same structure that underlies homeostasis, normalization, and contextual regulation across natural and computational systems.

Y-GD differs from gradient clipping or trust-region methods in one key respect: it regulates baseline-relative change rather than gradient magnitude or absolute parameter distance. This gives direct geometric control over the model's position relative to a reference configuration.

The triadic structure suggests further work: combining Y-GD with adaptive baselines, exploring multi-layer proportionality in deep networks, or integrating it into reinforcement-learning update rules where context and baseline already play functional roles.

6 Conclusion

We introduced Y–Gradient Descent (Y–GD), a baseline-relative update rule derived from a minimal triadic formulation of change. Rather than controlling absolute parameter displacement, Y–GD regulates the proportional deviation of the state from a moving baseline. This yields stable behavior even at learning rates for which ordinary gradient descent diverges, and it does so without additional hyperparameters or trust-region heuristics.

The proportional interpretation clarifies the mechanism responsible for Y–GD's robustness: each update is constrained to remain within a bounded baseline-relative ratio class, preventing runaway motion while preserving steady progress. Experiments on convex and non-convex objectives support this perspective, showing that Y–GD suppresses oscillation and stabilizes trajectories in high-curvature regimes.

The structural view developed here suggests several principled extensions. By treating gradient, momentum, and regularization as distinct mediators, their effects may be composed into a single coherent triadic transformation. Similarly, multi-baseline and layer-wise variants allow proportional regulation at multiple timescales or architectural levels. Continuous-time and stochastic perspectives, arising from log-mediator and probabilistic formulations, offer further avenues for analyzing or shaping the geometry of baseline-relative learning.

Overall, Y–GD illustrates how a simple triadic structure can yield practical gains in stability while opening a coherent space of extensions grounded in proportional change. Future work will explore composed mediators, multi-baseline schemes, and stochastic proportional dynamics on larger architectures and learning tasks.

Code and data: The PyTorch script and experiment data are available at `https://github.com/toddclark-research/y-gd` (placeholder link).

Acknowledgment

No external funding or collaborators were involved. Computations were performed on Google Colab.

References

[1] Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980.*

[2] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444.

[3] Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

[4] Sutton, R. S., & Barto, A. G. (2018). *Reinforcement Learning: An Introduction.* MIT Press.

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