A NOVEL MULTI-SEQUENCE OPTIMIZATION FRAMEWORK FOR NONLINEAR REGRESSION WITH GRADIENT BOOSTING MACHINES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Emirhan İlhan
September 2025

A NOVEL MULTI-SEQUENCE OPTIMIZATION FRAMEWORK FOR NONLINEAR REGRESSION WITH GRADIENT BOOSTING MACHINES

By Emirhan İlhan

September 2025

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Süleyman Serdar Kozat(Advisor)

Sinan Gezici

Ramazan Gökberk Cinbiş

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

A NOVEL MULTI-SEQUENCE OPTIMIZATION FRAMEWORK FOR NONLINEAR REGRESSION WITH GRADIENT BOOSTING MACHINES

Emirhan İlhan
M.S. in Electrical and Electronics Engineering
Advisor: Süleyman Serdar Kozat
September 2025

Gradient Boosting Machines (GBMs) consistently achieve state-of-the-art performance on a wide range of machine learning applications, particularly for problems tabular data. However, their underlying optimization mechanism largely relies on a greedy form of functional gradient descent. This classical approach, while effective, can be path-dependent under noise and prone to locally optimal updates that yield globally suboptimal models. To address these fundamental limitations, we propose a multi-sequence framework that integrates principles from modern optimization directly into the gradient boosting process.Unlike prior work that attempts direct adaptations of vector-based optimizers such as Nesterov's accelerated gradient, our method employs a decoupled architecture inspired by stochastic primal averaging (SPA). This architecture provides a stable foundation upon which we build a series of adaptive target updaters that translate the mechanics of modern optimizers such as Adam into principled learning signals for weak learners. We provide a theoretical analysis, including a linear convergence proof for our base model under standard smoothness and strong convexity assumptions. Our experimental results demonstrate significant improvements in stability and accuracy over LightGBM and other baselines, proving our framework's ability to offer a new level of control over the optimization trajectory while preserving the simplicity and modularity that make GBMs practical, leading to more robust and powerful gradient boosting models.

Keywords: Machine learning, regression, gradient boosting machines (GBM), optimization, functional gradient descent, momentum, stochastic primal averaging (SPA), ADAM optimizer, convergence.

ÖZET

GRADYAN ARTIRMALI MAKİNELERLE DOĞRUSAL OLMAYAN REGRESYON İÇİN ÇOKLU SEKANSLI YENİ BİR OPTİMİZASYON ÇERÇEVESİ

Emirhan İlhan
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Süleyman Serdar Kozat
Eylül 2025

Gradyan Artırmalı Makineler (GBM'ler), özellikle tablo verisine dayalı problemlerde, geniş bir makine öğrenimi uygulama yelpazesinde sürekli olarak ileri düzey performans sergiler. Ancak alttaki optimizasyon mekanizması büyük ölçüde açgözlü bir fonksiyonel gradyan inişine dayanır. Bu klasik yaklaşım etkili olsa da, gürültü altında yola bağımlı olabilir ve küresel olarak optimal olmayan modellere yol açan yerel olarak optimal güncellemelere eğilimlidir. Bu temel sınırlamaları gidermek için, modern optimizasyon ilkelerini doğrudan gradyan artırma sürecine entegre eden çoklu sekanslı bir çerçeve öneriyoruz. Nesterov'un ivmelendirilmiş gradyanı gibi vektör tabanlı optimize edicilerin doğrudan uyarlamalarını denemek yerine, yöntemimiz stokastik primal ortalama (SPA) esinli ayrıştırılmış bir mimari kullanır. Bu mimari, Adam gibi modern optimize edicilerin mekaniklerini zayıf öğrenenler için ilkesel öğrenme sinyallerine çeviren bir dizi uyarlanabilir hedef güncelleyicisinin inşa edildiği sağlam bir temel sunar. Standart düzgünlük ve kuvvetli dışbükeylik varsayımları altında temel modelimiz için doğrusal yakınsama kanıtını da içeren kuramsal bir analiz sağlıyoruz. Deneysel sonuçlarımız, LightGBM ve diğer karşılaştırma yöntemlerine kıyasla kararlılık ve doğrulukta önemli iyileşmeler göstererek, çerçevemizin optimizasyon yörüngesi üzerinde yeni bir kontrol düzeyi sunarken GBM'leri pratik kılan sadelik ve modülerliği koruduğunu ve bunun daha sağlam ve güçlü gradyan artırma modellerine yol açtığını kanıtlamaktadır.

Anahtar sözcükler: Makine öğrenimi, regresyon, gradyan artırmalı makineler (GBM), optimizasyon, fonksiyonel gradyan inişi, momentum, stokastik primal ortalama (SPA), Adam optimizasyon algoritması, yakınsama.

Acknowledgement

I extend my profound gratitude to my advisor, Prof. Süleyman Serdar Kozat, for his unwavering support, dedicated mentorship, and insightful guidance throughout my master's studies. His expertise, careful feedback, and patience strengthened both the theoretical foundations and the practical components of this work. It has been an honor to learn under his supervision.

I also thank Prof. Sinan Gezici and Assoc. Prof. Ramazan Gökberk Cinbiş for serving on my thesis examination committee and for their time and consideration.

I am sincerely grateful to my research team, colleagues, and friends—Ahmet Berker Koç, Bilal Onur Eskili, Furkan Burak Mutlu, İsmail Balaban, Kaan Berk Kabadayı, Mehmet Efe Lorasdağı, Mehmet Yiğit Turalı, and Ramazan Burak Güler—for their collaboration, stimulating discussions, and constant encouragement.

Last but not least, I owe boundless gratitude to my family for their unwavering love and support. I am deeply thankful to my parents for their enduring belief in me and the many sacrifices they have made. I thank my brothers—Fatih, for his guidance and constant encouragement, and Ahmet Tunahan, to whom I wish every success. Their faith and presence have been my continuous source of motivation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Gradient Boosting Machines (GBMs) are a class of ensemble learning methods that have proven highly effective for prediction tasks, particularly with structured or tabular data. GBM implementations such as XGBoost, LightGBM, and CatBoost are widely employed in industrial systems and consistently rank among the top performers in machine learning competitions [1–3]. The success of these models is rooted in the principle of functional gradient descent, where an ensemble of weak learners, typically decision trees, is built sequentially by fitting each new learner to the negative gradient of the loss [4–6]. Although this stage-wise approach is simple and powerful, the underlying optimization mechanism behind gradient boosting remains limited.

The standard GBM optimization procedure, while convergent and effective in practice, exhibits several challenging properties. First, the classical procedure is inherently greedy. At each iteration, the new weak learner is chosen solely to minimize the current residuals without regard to the long-term optimization trajectory. This path-dependent nature means that early, high-variance updates can set the model on a suboptimal path, drifting the ensemble away from better solutions [7, 8]. Second, directly fitting raw residuals makes boosting sensitive to noise, leading to high-variance weak learners whose errors can be propagated and amplified through the additive process [7]. Moreover, early missteps force later

trees to expend capacity on correcting accumulated errors rather than uncovering new structure behind the learning signal.

More broadly, classical GBMs suffer from a lack of sophisticated optimization control. Although common safeguards such as shrinkage, subsampling, and early stopping provide regularization and a degree of stabilization to the ensemble, they do not adapt to local gradient statistics nor proactively guide optimization [7, 8]. In contrast, modern optimization for deep learning enjoys the leverage of momentum, adaptivity, and averaging to improve stability and efficiency [9–11].

Existing strategies in the literature to address these issues consist of different approaches. Traditional, conservative techniques such as small learning rates, subsampling, and early stopping can slow or halt training when instability appears but do not actively alter the greedy dynamic [8]. Conversely, there have been direct attempts to incorporate more sophisticated optimizers from the deep learning literature (e.g., Nesterov momentum) into boosting [12, 13]. Such optimizers, however, are originally designed to adjust a fixed-dimensional parameter vector with smooth updates. Gradient boosting, on the other hand, operates by adding entire nonlinear functions at each step, and naive momentum-like buffers can accumulate brittle, high-variance function updates; moreover, acceleration methods are known to lose their advantages in the presence of inexact gradient oracles, which naturally arises in tree-based residual fitting [14]. Thus, direct adaptations of these optimizers struggle with the functional nature of GBMs, leading to inconsistent improvements. Hybrid formulations such as differentiable boosting and neural–boost combinations bring adaptivity at the cost of additional complexity, often sacrificing the simplicity and modularity that make GBMs attractive [1–3]. Consequently, despite decades of engineering progress, the core optimization mechanism of boosting has remained largely unchanged and lacks a principled way to incorporate momentum, adaptivity, and noise robustness in function space.

To address these limitations, we propose a novel multi-sequence optimization framework that, unlike prior approaches, synthesizes ideas from modern optimization in a manner native to the functional space of boosting. Our approach

is inspired by averaging techniques in stochastic optimization—stochastic primal averaging and schedule-free perspectives in particular—which are known to improve stability and convergence [11, 15–17]. Our key idea is to maintain multiple interacting sequences of models that decouple exploration from stability.

Specifically, we propose to utilize an exploratory sequence that accumulates the aggressive, high-variance updates from new weak learners, while a separate stable sequence averages these updates over time. This structure provides momentum-like effects without an explicit velocity buffer [15]. Upon this stable architecture, we introduce adaptive target updaters that reinterpret the mechanics of modern optimizers such as Adam to produce variance- and momentum-aware learning signals for weak learners [10]. Instead of modifying the ensemble directly, these updaters refine the learning signal itself, guiding weak learners with momentum- and variance-aware targets. The result is a framework that (i) retains the exploratory power of greedy tree learners, (ii) inherits the stability of principled averaging and momentum, and (iii) gains the adaptivity of modern optimizers, all while avoiding an unnatural analogy to vector-space optimization.

Our contributions are as follows:

1. We introduce a multi-sequence optimization framework for GBMs, inspired by Primal Averaging and Schedule-Free Learning, that stabilizes training by decoupling greedy exploration from stable predictions, and provides a foundation for principled extensions [11, 16].

2. We provide a theoretical analysis of the framework, including a proof of linear convergence for the base model under standard smoothness and strong convexity assumptions.

3. We design adaptive target updaters that translate the principles of Adam and AdaBelief into refined learning signals for trees, achieving noise robustness and adaptivity [10, 18].

4. We adapt schedule-free momentum to boosting, enabling acceleration through interpolated residuals without auxiliary buffers [16, 17].

5. We demonstrate the practical strengths of our methods through ablations and experiments on real-life datasets, showing improved stability, robustness, and predictive accuracy.

Chapter 2

A Novel Multi-Sequence Optimization Framework for Nonlinear Regression with Gradient Boosting Machines

This chapter details the proposed multi-sequence optimization framework. As outlined in the introduction, the core of our approach is to decouple model exploration from stabilization. We achieve this by maintaining two interacting model sequences: a stable sequence that provides a reliable, averaged foundation for gradient evaluation, and a parallel exploratory sequence that accumulates more aggressive, high-variance updates. Building upon this dual-sequence architecture, we then introduce adaptive target updaters. These updaters translate the mechanics of modern vector-based optimizers, namely Adam, into principled learning signals for the weak learners [10, 18]. This process embeds momentum and variance awareness into the functional optimization path itself, rather than applying them post-hoc to the ensemble.

2.1 Background

In this section, we review the conceptual foundations of gradient boosting and optimization, situating our framework within the broader landscape of statistical learning and algorithmic design. We proceed from classical methods to modern optimization techniques.

2.1.1 Gradient Boosting Machines

Gradient Boosting Machines (GBMs) are a class of supervised learning algorithms that construct accurate predictors by iteratively combining a series of simpler models (typically regression trees) in a stage-wise manner [4, 7]. The underlying principle treats model building as optimization in a function space: at each iteration, a new model is trained to correct the errors made by the current ensemble via functional gradient descent [4–6]. This viewpoint extends boosting beyond its original classification setting (e.g., AdaBoost) to arbitrary differentiable losses.

The objective is to learn a predictive function $F(\mathbf{x})$ that minimizes the expected loss over the data distribution:

$$F^* = \arg\min_F \ \mathbb{E}_{\mathbf{x},y}\big[L\big(y, F(\mathbf{x})\big)\big].$$

In practice, this expectation is approximated by the empirical risk on a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$. The model $F(\mathbf{x})$ is constructed as an additive ensemble

$$F_T(\mathbf{x}) = \sum_{t=0}^{T} f_t(\mathbf{x}),$$

where $f_0(\mathbf{x})$ is a constant initializer and, for $t \geq 1$, $f_t(\mathbf{x}) = \nu\, h_t(\mathbf{x})$ for a weak learner $h_t \in \mathcal{H}$ and learning rate $\nu \in (0, 1]$ (shrinkage) [4, 8]. At iteration t, the negative gradient of the empirical risk with respect to the current predictions yields the pseudo-residuals

$$r_{it} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_{t-1}}, \quad i = 1, \ldots, N,$$

and the new weak learner is trained to approximate these residuals, typically by least-squares fitting [4].

Algorithm 1 Gradient Tree Boosting [4]

Input: Training data $\{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$, differentiable loss $L(y, F)$, iterations T, learning rate ν.

Initialize:

$$F_0(\mathbf{x}) = \arg\min_{c} \sum_{i=1}^{N} L(y_i, c).$$

for $t = 1$ to T **do**

  Compute pseudo-residuals:

$$r_{it} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_{t-1}}, \quad i = 1, \ldots, N.$$

  Fit a regression tree $h_t(\mathbf{x}) \in \mathcal{H}$ to $\{(\mathbf{x}_i, r_{it})\}_{i=1}^{N}$ (least-squares surrogate).

  Update:

$$F_t(\mathbf{x}) = F_{t-1}(\mathbf{x}) + \nu\, h_t(\mathbf{x}).$$

end for

Output: $F_T(\mathbf{x})$.

This formulation highlights the dual role of weak learners: they approximate residuals flexibly while their limited complexity, combined with shrinkage and early stopping, regularizes the ensemble and supports generalization [7, 8].

The widespread adoption of GBMs has been driven by highly engineered implementations that address scalability, speed, and regularization while retaining the stage-wise functional gradient descent core. XGBoost [1] popularized second-order tree fitting via a per-tree Taylor expansion and added explicit regularization on leaf weights and tree complexity, alongside systems optimizations (e.g., out-of-core training). LightGBM [2] introduced histogram-based gradient binning and leaf-wise growth to accelerate training, with additional regularization to mitigate overfitting risks. CatBoost [3] emphasized stability through ordered boosting to reduce target leakage and employed symmetric trees for computational regularity. Despite these advances, the core optimization remains a greedy, stage-wise procedure in function space: each new learner is added to correct the current residuals without an explicit mechanism for long-horizon trajectory control [4, 6].

2.1.1.1 Advanced Optimization Tools for Deep Learning

The modern success of deep learning has been closely linked to advances in optimization. While the goal remains empirical risk minimization, the choice of optimizer critically affects convergence speed, stability, and robustness. We review methods most relevant to our framework: stochastic gradient descent (SGD), momentum, adaptive methods (Adam), and averaging-based perspectives (stochastic primal averaging and schedule-free learning).

Stochastic Gradient Descent (SGD). Given an empirical risk $\mathcal{R}(\theta) = \frac{1}{N}\sum_{i=1}^{N} L\big(y_i, f(\mathbf{x}_i;\theta)\big)$, SGD updates parameters using a mini-batch estimate of the gradient:

$$g_t = \frac{1}{|\mathcal{B}_t|}\sum_{(\mathbf{x},y)\in\mathcal{B}_t} \nabla_\theta L\big(y, f(\mathbf{x};\theta_t)\big), \qquad \theta_{t+1} = \theta_t - \eta_t\, g_t.$$

SGD is simple and scalable but is sensitive to step size choice and can converge slowly on ill-conditioned problems [19].

Momentum methods. Momentum accumulates a moving average of past gradients to smooth the update trajectory. The heavy-ball method [20] maintains a velocity $v_{t+1} = \beta v_t + g_t$ and updates $\theta_{t+1} = \theta_t - \eta v_{t+1}$. Nesterov's accelerated gradient evaluates the gradient at a lookahead point and achieves optimal rates for smooth convex problems [12]; momentum has also been shown empirically effective in deep learning [9]. Despite improvements in practice, momentum remains sensitive to learning-rate schedules.

Adam and adaptive methods. Adaptive optimizers compute per-parameter stepsizes using moment estimates of the gradient. Adam combines first- and second-moment estimates to rescale updates and often improves robustness to step size choice [10]. Related variants refine the second-moment estimate for noise handling [18]. While these methods operate in parameter space, directly porting them to boosting requires care due to inexact functional gradients.

Averaging, SPA connections, and analysis. Iterate averaging is a classical stabilization device [11]. Connections between averaging and acceleration

clarify why momentum-like behavior can emerge from averaging schemes and facilitate analysis [15]. Stochastic primal averaging (SPA) formalizes this view by coupling a "primal" sequence with an averaged sequence via simple interpolation rules, offering momentum-like effects with analysis-friendly structure [21]. This connection was rigorously established by [22], who demonstrated that classical momentum is mathematically equivalent to the SPA form through a direct reparameterization of the update rules. This equivalence is analytically powerful, as the SPA structure is more amenable to Lyapunov analysis, which in turn provides deeper insights into momentum's behavior. The analysis reveals that momentum's benefits extend beyond simple acceleration to a form of noise regularization, particularly during the early, high-gradient stages of optimization. These ideas motivate adopting averaging-first designs when mapping optimization tools to function space.

Schedule-free learning perspectives. A persistent challenge is reliance on hand-crafted learning-rate schedules. Schedule-free viewpoints in online/convex settings unify averaging, momentum, and schedules via interpolation, reducing dependence on total training steps and mitigating schedule mis-specification [16, 17, 23]. Conceptually, such interpolation can emulate linear-decay behavior without explicit schedules, a property attractive for stable, noise-aware optimization rules.

2.1.2 Related Work

A growing literature seeks to move beyond the greedy, last-iterate updates of classical Gradient Boosting Machines (GBMs) [4] by importing ideas from first-order optimization: momentum-based acceleration, iterate averaging, and schedule design. The central tension is that while accelerated first-order methods achieve $O(1/T^2)$ rates in convex settings [12, 13, 19], boosting operates with *inexact* descent directions delivered by weak learners, so naive acceleration can amplify approximation errors [14]. Below we review momentum-based accelerated boosting, alternative optimization-minded GBM formulations for regression, and position

our approach accordingly.

Momentum-based acceleration in boosting. Early work on acceleration directly translated Nesterov's two-sequence strategy into function space. [24] proposed *Accelerated Proximal Boosting*, introducing a momentum-like extrapolation step $F_{t+1} = (1-\alpha_t)F_t + \alpha_t G_t$ and then applying a proximal/gradient-like correction $G_{t+1} = F_{t+1} - \nu h_{t+1}$, where h_{t+1} is fitted to pseudo-residuals at the current model. The construction mirrors the "lookahead" of Nesterov's method [12, 13] and is motivated by the functional-gradient perspective on boosting [4, 5]. However, as [14] formalized in the context of inexact oracles, acceleration can lose its superiority when descent directions are approximate. Empirically, [24] reported speed-ups but also noted instability and a lack of unconditional guarantees, foreshadowing the need to control error accumulation from weak learners.

[25] advanced this line with *Accelerated Gradient Boosting* (AGB). AGB maintains two function sequences (F_t) and (G_t) in the spirit of Nesterov:

$$h_t \in \arg\min_{h\in\mathcal{H}} \langle \nabla\mathcal{R}(F_t), h\rangle, \quad F_{t+1} = G_t - \nu h_t, \quad G_{t+1} = F_{t+1} + \beta_t(F_{t+1} - F_t),$$

where $\mathcal{R}$ is the empirical risk. AGB offered strong empirical evidence: faster training-loss reduction, many fewer trees for a fixed accuracy, and reduced sensitivity to shrinkage ν relative to standard GBM. Those findings dovetail with broader practice on the importance of momentum for optimization efficiency [9]. At the same time, [25] explicitly cautioned that inexact gradients (weak learners, sampling noise) can undermine momentum's benefits, echoing negative results from inexact-oracle analyses [14].

The most theoretically complete treatment is *Accelerated Gradient Boosting Machine* (AGBM) by [26]. Their key innovation is a *corrected pseudo-residual* that disentangles momentum extrapolation from weak-learner approximation error, thereby avoiding error accumulation in the momentum buffer. With this correction, AGBM establishes the first $O(1/T^2)$ training-loss rate for boosting under convex losses. Empirically, AGBM reduces the number of boosting rounds

required to reach a target loss, placing momentum-boosting on firm ground relative to the earlier heuristic attempts [24, 25]. The broader conceptual lesson—aligned with the two-sequence templates of Nesterov and FISTA [12, 13]—is that acceleration in function space can work if the algorithm separates extrapolation effects from approximation error.

Other optimization-minded GBM formulations (regression). Beyond explicit momentum, several lines of work reinterpret or modify boosting through optimization lenses while retaining regression losses. [27] introduced *Randomized Gradient Boosting*, which randomizes coordinate/feature choices for weak learners to reduce the bias of greedy directions and stabilize progress, enabling explicit finite-time guarantees of order $O(1/T)$ under convexity. [28] cast boosting as a functional Frank–Wolfe (conditional gradient) procedure, where each weak learner solves the linearized subproblem $h_t \in \arg\min_{h \in \mathcal{H}} \langle \nabla \mathcal{R}(F_t), h \rangle$ and the model update is a convex combination $F_{t+1} = (1 - \gamma_t) F_t + \gamma_t h_t$, bringing Frank–Wolfe convergence insights to the boosting setting.

A complementary thread incorporates proximal/regularization mechanisms. Proximal boosting variants [24] and regularized objective designs (e.g., the explicit second-order regularization and sparsity-aware mechanisms popularized by XGBoost [1] and the efficiency-oriented leaf-wise growth and histogram binning of LightGBM [2]) can be interpreted as engineering controls on optimization trajectories rather than fundamental redesigns of the update logic. In a related spirit, CatBoost's *ordered boosting* [3] uses permutation-based target statistics to prevent leakage, which stabilizes residual estimates even in regression; while its focus is categorical feature handling, the mechanism also acts as a stability prior on the learning signal.

Averaging, momentum, and schedules: lessons from stochastic optimization. The optimization community has developed a unified view of momentum, iterate averaging, and scheduling. Heavy-ball momentum [20] and Nesterov acceleration [12] are linked to averaging schemes through both classical Polyak–Ruppert results [11] and modern analyses that connect averaging and acceleration [15]. The *stochastic primal averaging* (SPA) perspective clarifies that momentum can be written as averaging of two coupled sequences, simplifying analysis [21, 29]. Critically, when step-size schedules are hand-designed, momentum can exhibit pathologies unless parameters are coupled and varied gradually [29]. Schedule-free learning [30] eliminates the reliance on a pre-specified training horizon T (the "tyranny of T") and yields worst-case optimality guarantees for interpolated momentum in convex problems, while empirically permitting larger learning rates without instability. These insights are directly relevant in boosting, where greedy residuals constitute a noisy and inexact oracle and learning dynamics are sensitive to schedules and shrinkage.

Positioning of our contribution. Relative to the momentum-boosting line [24–26], our approach takes a different route to robust optimization in function space. Instead of accelerating *ensemble weights* via extrapolation and correcting residuals post hoc, we *reform the learning signal itself* through *adaptive target updaters* embedded in a *multi-sequence scaffold*. Concretely, we adopt SPA's averaging form [29] to obtain momentum directly in function space and then normalize pseudo-residuals by second-moment statistics (RMS scaling) or deviation-sensitive estimates (AdaBelief-style), analogous to Adam/AdaBelief in deep learning [10, 18]. Finally, drawing on schedule-free analysis [30], we replace hand-tuned averaging schedules with a principled interpolation rule. This *targets-first* philosophy—stabilizing and adapting the residuals before training weak learners—yields a holistic redesign aimed at stability, adaptivity, and robustness, not merely acceleration. It complements AGBM's theoretically sound corrected residuals by offering an orthogonal mechanism: rather than compensating for extrapolation error, we engineer the learning signal to be momentum-consistent and variance-aware by construction.

2.2 Methods

In this section, we detail the development of a novel optimization architecture for Gradient Boosting Machines. Departing from the classical single-sequence formulation of functional gradient descent [4, 5], we progressively construct our framework. We begin by formalizing the regression problem and the limitations of the standard paradigm and then introduce, in a step-by-step manner, principled mechanisms for systematically enhancing stability, introducing adaptivity, and incorporating momentum. The result is a more robust and controllable optimization framework for gradient boosting.

2.2.1 Problem Statement and Classical Gradient Boosting

We consider the standard supervised learning problem for regression. Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a training dataset, where each input $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ is a d-dimensional feature vector and $y_i \in \mathbb{R}$ is the corresponding target value. Our objective is to learn a prediction function $F : \mathcal{X} \to \mathbb{R}$ that minimizes the empirical risk over the training data with respect to a loss function $L : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$:

$$\mathcal{L}(F) \;=\; \frac{1}{N} \sum_{i=1}^{N} L(y_i, F(\mathbf{x}_i)). \tag{2.1}$$

Throughout this paper, we assume L is a differentiable loss function, with the squared loss $L(y, \hat{y}) = \frac{1}{2}(y - \hat{y})^2$ serving as a canonical example.

As detailed in Section 2.1, classical Gradient Boosting Machines (GBMs) solve this minimization problem by constructing the prediction function F as an additive ensemble of weak learners $h_t(\mathbf{x})$ from a function class $\mathcal{H}$:

$$F_T(\mathbf{x}) = F_0(\mathbf{x}) + \sum_{t=1}^{T} \nu_t h_t(\mathbf{x}), \tag{2.2}$$

where $F_0(\mathbf{x}) = \arg\min_c \sum_{i=1}^N L(y_i, c)$ is an initial constant model and $\nu_t > 0$ is a learning rate (shrinkage; typically $\nu_t \equiv \nu$). At each stage t, a new weak learner is

fit to the negative gradient of the loss function (the pseudo-residuals) and added to the ensemble according to the update rule [4]:

$$F_t(\mathbf{x}) \;=\; F_{t-1}(\mathbf{x}) + \nu_t h_t(\mathbf{x}). \tag{2.3}$$

Despite its strong empirical performance, the greedy, stage-wise nature of this process is its principal weakness. At each step, the algorithm adds the weak learner that best minimizes the current loss without considering the long-term optimization trajectory. Consequently, a sequence of locally optimal updates can yield inefficient trajectories or poor practical convergence [6, 28]. This path-dependent nature has further consequences: since the residuals directly reflect data fluctuations, the algorithm can aggressively fit to noise, leading to instability [7]. Furthermore, high-variance updates early in training can force later learners to expend their capacity on correcting accumulated errors rather than capturing new data structures.

Mitigation strategies such as shrinkage (small ν_t), early stopping, and subsampling have been introduced to alleviate these issues. While effective, these measures are reactive: they control divergence or overfitting after it appears but do not fundamentally alter the greedy mechanics of functional gradient descent [7, 8]. From an optimization perspective, classical GBM can be viewed as a first-order method with limited tools for trajectory control and efficient convergence [6, 19]. This gap motivates our framework: to design a principled multi-sequence approach that imports the benefits of modern optimization—averaging and momentum-inspired stabilization—while preserving the exploratory power of greedy tree learners [11, 16, 23].

2.2.2 Momentum and Averaging for Gradient Boosting

The greedy, stage-wise nature of the vanilla gradient boosting algorithm, as detailed in Section 2.2.1, lacks any mechanism for incorporating the history of the optimization path to guide future steps. In numerical optimization—particularly in deep learning—momentum was introduced to address exactly this limitation: it

accelerates convergence and stabilizes noisy updates [9,12,20]. For gradient boosting, where weak learners approximate gradients with varying fidelity, both properties are critical. In this section, we motivate momentum as a noise-regularizing mechanism, review its equivalence to stochastic primal averaging (SPA), and argue that SPA provides a more natural perspective for adapting momentum to the boosting framework. We conclude by proposing our SPA-GBM algorithm.

In Euclidean parameter space, momentum accelerates gradient descent by accumulating an exponentially weighted moving average of past gradients. This allows the optimizer to build speed in consistent directions and dampen oscillations. For stochastic gradient descent (SGD) with stochastic gradient g_t at iteration t, the heavy-ball updates are

$$v_{t+1} = \beta v_t + g_t, \tag{2.4}$$

$$\theta_{t+1} = \theta_t - \eta v_{t+1}, \tag{2.5}$$

where $\theta_t \in \mathbb{R}^d$ are the parameters, $\eta > 0$ is the learning rate, and $\beta \in [0, 1)$ is the momentum coefficient [20]. The term v_t aggregates historical gradients, encouraging updates that follow smoothed directions rather than oscillating with instantaneous noise [9].

Beyond acceleration, momentum acts as a noise regularizer: averaging successive gradient estimates suppresses high-variance fluctuations, yielding more stable trajectories [11, 15]. This is particularly relevant for gradient boosting, where pseudo-residuals (negative gradients of the empirical loss with respect to current predictions) are noisy approximations of functional gradients [4]. Naive greedy updates risk amplifying this noise: a single overfitted weak learner can perturb the optimization path and propagate errors forward.

However, directly transplanting the momentum rule in (2.5) to function space is ill-suited. A literal translation would maintain a momentum buffer that is itself a function and would add a new weak learner to this buffer at each step. With inexact descent directions (as provided by weak learners), such accumulation can compound approximation errors and lead to instability [14]; empirical accelerated-boosting variants that mirror Nesterov-style extrapolation have also

been observed to be fragile without corrections [25, 26].

A more suitable perspective is provided by stochastic primal averaging (SPA). SPA maintains two sequences of iterates—an auxiliary "primal" sequence and an averaged sequence—updated via

$$s_{t+1} = s_t - \eta g_t, \tag{2.6}$$

$$x_{t+1} = (1 - \gamma_{t+1})\, x_t + \gamma_{t+1}\, s_{t+1}, \tag{2.7}$$

where $\eta > 0$ is a stepsize and $\gamma_{t+1} \in (0, 1]$ is an averaging parameter. Classical iterate averaging [11] and links between averaging and acceleration [15] motivate SPA; moreover, SPA can be viewed as an alternative parameterization of momentum with analysis-friendly structure [21, 29]. This dual view recasts momentum not as an ad hoc velocity term but as principled averaging.

This equivalence is especially important for boosting. Whereas velocity-based updates (2.4) are awkward in function space, iterate averaging (2.7) naturally extends to ensembles. The separation of roles, an aggressive, exploratory sequence that receives direct updates and a stable, averaged sequence, decouples high-variance weak-learner updates from the model state used for prediction and subsequent gradient computation. This avoids directly accumulating functional errors in a single momentum buffer and offers a more stable translation of momentum's benefits.

We therefore propose the Stochastic Primal Averaging Gradient Boosting (SPA-GBM) algorithm. We adapt SPA to the functional setting by maintaining two distinct model sequences—an exploratory sequence G_t (analogous to s_t) and a stable sequence F_t (analogous to x_t). Here, η plays the role of a functional stepsize for incorporating a new weak learner, while $\{\gamma_t\}$ controls averaging.

Unlike vanilla GBM, which directly updates a single model, SPA-GBM separates the learning signal into two steps: a tentative update to the exploratory sequence (G_{t+1}) and a controlled interpolation to update the stable sequence (F_{t+1}). Computing gradients from the stable sequence breaks the feedback loop in which noise from an inexact weak learner immediately contaminates the next

Algorithm 2 Stochastic Primal Averaging Gradient Boosting (SPA-GBM)

Require: Dataset $\mathcal{D}$, loss $L(y, \hat{y})$, stepsize η, averaging schedule $\{\gamma_t\}$, iterations T.

1: Initialize $F_0(\mathbf{x}) = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$ and set $G_0(\mathbf{x}) = F_0(\mathbf{x})$.
2: **for** $t = 0, \ldots, T-1$ **do**
3: Compute pseudo-residuals at the stable model F_t [4]:

$$r_{it} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_t}, \quad i = 1, \ldots, N.$$

4: Fit weak learner $h_{t+1}(\mathbf{x})$ to $\{(\mathbf{x}_i, r_{it})\}_{i=1}^{N}$.
5: Update the exploratory sequence:

$$G_{t+1}(\mathbf{x}) = G_t(\mathbf{x}) + \eta\, h_{t+1}(\mathbf{x}).$$

6: Average to obtain the next stable iterate:

$$F_{t+1}(\mathbf{x}) = (1 - \gamma_{t+1})\, F_t(\mathbf{x}) + \gamma_{t+1}\, G_{t+1}(\mathbf{x}).$$

7: **end for**
8: **Output:** Final model $F_T(\mathbf{x})$.

learning signal, while the exploratory sequence absorbs variance that is smoothly averaged into the predictor over time [11,15]. This provides an implicit, analysis-friendly form of momentum suited to the functional setting.

2.2.3 Adaptive Target Updates

While Stochastic Primal Averaging (SPA) introduces momentum into gradient boosting in a function-space-compatible way, the learning signal itself, the pseudo-residual vector $\mathbf{r}_t$, is still treated as a raw, unfiltered target. This does not address a key challenge in boosting: the heterogeneous scale and variance of pseudo-residuals across samples and iterations [7]. In deep learning, adaptive optimization methods such as Adam adjust update magnitudes according to the history of gradients and often improve stability [10]. We reinterpret these ideas for boosting by introducing *adaptive pseudo-residuals*.

Vanilla GBM and SPA-GBM both use raw pseudo-residuals r_{it} (negative gradients of the empirical loss with respect to current predictions) to train weak learners [4]. However, these residuals can be highly variable. A single misfit tree can cause subsequent residuals to fluctuate, leading to unstable optimization. In parameter optimization, stochastic gradients often differ in scale across coordinates and time, which motivates adaptive step sizes. Analogously, in boosting, the learning signal may require normalization to prevent instability.

To motivate our construction, recall the Adam optimizer, which augments momentum with per-coordinate scaling. Given stochastic gradients g_t, Adam maintains exponential moving averages of the first and second moments:

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t, \tag{2.8}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2, \tag{2.9}$$

with bias corrections $\hat{m}_t = m_t/(1-\beta_1^t)$ and $\hat{v}_t = v_t/(1-\beta_2^t)$. The parameter update scales by these estimates [10]:

$$\theta_{t+1} = \theta_t - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}. \tag{2.10}$$

SPA-GBM provides a natural foundation to import Adam's variance-aware scaling into boosting. Since SPA-GBM already imparts momentum-like effects via averaging, we introduce *adaptivity* by rescaling the learning signal before it reaches the weak learner. Concretely, we maintain an exponential moving average of squared residuals for each training point i:

$$v_{it} \;=\; \beta_2\, v_{i,t-1} \;+\; (1-\beta_2)\, r_{it}^2, \qquad \beta_2 \in [0,1), \tag{2.11}$$

and apply the standard bias correction $\hat{v}_{it} = v_{it}/(1-\beta_2^t)$. We then normalize the raw residuals to form adaptive targets $\tilde{r}_{it}$:

$$\tilde{r}_{it} = \frac{r_{it}}{\sqrt{\hat{v}_{it}} + \epsilon}. \tag{2.12}$$

This per-sample, second-moment-based normalization tempers large or noisy residuals and yields variance-aware learning signals (cf. [10]; see also [18]). The

Algorithm 3 Adam-SPA Gradient Boosting (Adam-SPA-GBM)

Require: Dataset $\mathcal{D}$, loss L, stepsize η, decay β_2, averaging schedule $\{\gamma_t\}$, iterations T, $\epsilon > 0$.

Initialize $F_0(\mathbf{x}) = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$; set $G_0(\mathbf{x}) \leftarrow F_0(\mathbf{x})$ and $v_{i,0} \leftarrow 0$ for $i = 1, \ldots, N$.

for $t = 0, \ldots, T-1$ **do**

  Compute pseudo-residuals at the stable model F_t [4]:

$$r_{it} = -\left[\frac{\partial L(y_i, F(\mathbf{x}_i))}{\partial F(\mathbf{x}_i)}\right]_{F=F_t}, \quad i = 1, \ldots, N.$$

  Update second-moment estimate for each sample i via (2.11); set $\hat{v}_{it} \leftarrow v_{it}/(1-\beta_2^{t+1})$.

  Form adaptive pseudo-residuals for each sample i via (2.12).

  Fit weak learner $h_{t+1}(\mathbf{x})$ to $\{(\mathbf{x}_i, \tilde{r}_{it})\}_{i=1}^{N}$ (least-squares surrogate).

  Update exploratory sequence:

$$G_{t+1}(\mathbf{x}) = G_t(\mathbf{x}) + \eta\, h_{t+1}(\mathbf{x}).$$

  Average to obtain the next stable iterate:

$$F_{t+1}(\mathbf{x}) = (1-\gamma_{t+1})\, F_t(\mathbf{x}) + \gamma_{t+1}\, G_{t+1}(\mathbf{x}).$$

end for

Output: Final model $F_T(\mathbf{x})$.

momentum elements of Adam is already incorporated into SPA-GBM thanks to the primal sequence.

By reinterpreting Adam's variance-aware scaling at the level of pseudo-residuals, Adam-SPA-GBM regularizes the learning signal before weak learners are trained. Combined with SPA averaging, this yields a training process that is more robust to noisy samples and less sensitive to a single global step size [7, 10]. (Here, η controls the functional step for incorporating h_{t+1}, while $\{\gamma_t\}$ governs averaging; their roles are distinct but complementary.)

2.2.4 Schedule-Free Learning for Gradient Boosting

The final ingredient in our framework is the removal of explicit averaging schedules. While SPA provides momentum via iterate averaging, it still requires a hand-tuned averaging sequence $\{\gamma_t\}$. In practice, selecting $\{\gamma_t\}$ is often heuristic. Schedule-free learning replaces such schedules with analytically chosen interpolation rules that retain convergence guarantees while avoiding manual tuning [16, 17, 23, 30].

Following [30], we introduce an explicit *gradient-evaluation* sequence Y_t in addition to the exploratory and averaged sequences. Mapping the schedule-free notation (z_t, x_t, y_t) to boosting, we use (G_t, F_t, Y_t):

$$Y_t \;=\; (1-\beta)\, G_t \;+\; \beta\, F_t, \qquad \beta \in [0, 1). \tag{2.13}$$

Gradients (pseudo-residuals) are evaluated at Y_t, not at the last averaged model. This mirrors the schedule-free momentum interpolation $y_t = (1-\beta)z_t + \beta x_t$ [30].

Schedule-free methods employ a constant *base* step size with a short linear warmup [30, Alg. 1]. For boosting, we set

$$\eta_t \;=\; \eta \cdot \min\Big(1,\; \frac{t+1}{T_{\text{warmup}}}\Big), \tag{2.14}$$

where $\eta > 0$ is the base step and T_{warmup} is the number of warmup iterations. (For the Adam-style variant below, η_t additionally incorporates the usual second-moment bias correction, exactly as in [30, Alg. 1, line 8].)

Crucially, the averaging coefficient should reflect any transient variation in η_t during warmup. Following [30, Eq. (23)], we use the *squared-step* weighting

$$c_{t+1} \;=\; \frac{\eta_t^2}{\sum_{i=0}^{t} \eta_i^2}, \tag{2.15}$$

which yields $c_{t+1} = 1/(t+1)$ once η_t becomes constant after warmup (up to the usual off-by-one index shift discussed in [30]). This choice matches the theory linking learning-rate schedules and iterate averaging [16, 17, 23].

With Y_t for gradient evaluation, the exploratory sequence updated additively, and the averaged sequence updated by (2.15), we obtain:

Algorithm 4 Schedule-Free GBM (three-sequence)

Require: Dataset $\mathcal{D}$, loss L, base step η, warmup T_{warmup}, momentum/interpolation $\beta \in [0, 1)$, iterations T.

Initialize $F_0(\mathbf{x}) = \arg\min_c \sum_{i=1}^{N} L(y_i, c)$; set $G_0(\mathbf{x}) \leftarrow F_0(\mathbf{x})$.
for $t = 0, \ldots, T-1$ **do**
  Interpolate (gradient-location): $Y_t(\mathbf{x}) \leftarrow (1-\beta)G_t(\mathbf{x}) + \beta F_t(\mathbf{x})$. [30]
  Compute pseudo-residuals at Y_t: $r_{it} = -\big[\partial L(y_i, F(\mathbf{x}_i))/\partial F(\mathbf{x}_i)\big]_{F=Y_t}$ [4].
  Fit weak learner h_{t+1} to $\{(\mathbf{x}_i, r_{it})\}_{i=1}^{N}$.
  Set step: $\eta_t \leftarrow \eta \cdot \min\big(1, \frac{t+1}{T_{\text{warmup}}}\big)$.
  Exploratory update: $G_{t+1}(\mathbf{x}) \leftarrow G_t(\mathbf{x}) + \eta_t\, h_{t+1}(\mathbf{x})$.
  Averaging weight: $c_{t+1} \leftarrow \eta_t^2 \Big/ \sum_{i=0}^{t} \eta_i^2$.
  Averaged update: $F_{t+1}(\mathbf{x}) \leftarrow (1-c_{t+1})F_t(\mathbf{x}) + c_{t+1}G_{t+1}(\mathbf{x})$.
end for
Output: $F_T(\mathbf{x})$.

Compared to the SPA-GBM in Section 2.2.2, the key changes are: (i) residuals are computed at the interpolated point Y_t (not F_t), and (ii) the averaging coefficient c_{t+1} is the squared-step ratio (2.15), which reduces to $1/(t+1)$ only after warmup. Both changes follow [30].

Adam–Schedule-Free GBM. To incorporate variance-aware scaling (Section 2.2.3) while remaining faithful to the schedule-free design, we adopt the same three-sequence structure and warmup/bias-correction used in [30, Alg. 1]:

Notes and defaults. Empirically, [30] report that a constant β works well across tasks (e.g., $\beta = 0.9$), with larger values sometimes beneficial. The linear

Algorithm 5 Adam–Schedule-Free GBM (three-sequence)

Require: Dataset $\mathcal{D}$, loss L, base step η, warmup T_{warmup}, $\beta \in [0,1)$, $\beta_2 \in [0,1)$, iterations T, $\epsilon > 0$.

Initialize $F_0(\mathbf{x})$, set $G_0(\mathbf{x}) \leftarrow F_0(\mathbf{x})$, and $v_{i,0} \leftarrow 0$ for all i.
for $t = 0, \ldots, T-1$ **do**
  Interpolate (gradient-location): $Y_t(\mathbf{x}) \leftarrow (1-\beta)G_t(\mathbf{x}) + \beta F_t(\mathbf{x})$.
  Compute pseudo-residuals at Y_t: r_{it} [4].
  Second-moment update: $v_{it} \leftarrow \beta_2 v_{i,t-1} + (1-\beta_2) r_{it}^2$.
  Bias-corrected step: $\eta_t \leftarrow \eta \sqrt{1-\beta_2^{t+1}} \cdot \min\bigl(1, \frac{t+1}{T_{\text{warmup}}}\bigr)$. [30, Alg. 1]
  Adaptive residuals: $\tilde{r}_{it} \leftarrow \dfrac{r_{it}}{\sqrt{v_{it}} + \epsilon}$.
  Fit h_{t+1} to $\{(\mathbf{x}_i, \tilde{r}_{it})\}_{i=1}^N$.
  Exploratory update: $G_{t+1}(\mathbf{x}) \leftarrow G_t(\mathbf{x}) + \eta_t\, h_{t+1}(\mathbf{x})$.
  Averaging weight: $c_{t+1} \leftarrow \eta_t^2 \Big/ \sum_{i=0}^t \eta_i^2$. [30, Eq. (23)]
  Averaged update: $F_{t+1}(\mathbf{x}) \leftarrow (1-c_{t+1})F_t(\mathbf{x}) + c_{t+1} G_{t+1}(\mathbf{x})$.
end for
Output: $F_T(\mathbf{x})$.

warmup factor in (2.14) is short (a small fraction of total iterations), after which c_{t+1} decays as $1/(t+1)$ due to the constant step. Our use of Y_t for residual evaluation exactly mirrors the schedule-free gradient evaluation at y_t and is necessary for fidelity to the original method [30, Alg. 1, line 5–6].

2.2.5 Convergence Analysis for SPA-GBM

In this section, we establish linear convergence of the base SPA-GBM algorithm via a two-sequence Lyapunov analysis. The proof proceeds in four steps. First, under exact residual least-squares and a minimal cosine angle (MCA) assumption on the weak-learner class, the learner selected at x_m satisfies perfect alignment with the residual and a norm window $\Theta\|\nabla L(x_m)\| \le \|b_m\| \le \|\nabla L(x_m)\|$. Second, σ-smoothness of L controls the two-point mismatch between the exploratory point s_m (where the update is taken) and the stable point x_m (where residuals are computed), introducing a term proportional to $\|s_m - x_m\| = \chi\|\Delta_m\|$. Third, a movement identity couples successive averaged increments and weak-learner norms, yielding a kinetic bound on $\|x_{m+1} - x_m\|^2$. Finally, we define a three-term

Lyapunov potential mixing $L(s_{m+1})$, $L(x_m)$, and the kinetic term; with carefully chosen weights, cross-terms cancel and we obtain a one-step decrease. Strong convexity then converts gradient norms into function-value gaps, and together with the MCA window this yields a geometric contraction of the Lyapunov potential and hence of $L(x_m) - L^*$.

2.2.5.1 Setup and Notation

We analyze the algorithm in the empirical function space $\mathcal{F} = \mathrm{span}(\mathcal{B})$ endowed with the inner product $\langle f, g \rangle = \frac{1}{N} \sum_{i=1}^{N} f(\mathbf{x}_i) g(\mathbf{x}_i)$. The proof follows the SPA two-sequence template: (i) a *movement/kinetic* lemma controlling the averaged-step increment; (ii) a *descent* lemma bounding the loss decrease; (iii) a three-term Lyapunov potential Λ with weights chosen to cancel cross-terms and yield a *Lyapunov step inequality*; and (iv) a strong-convexity argument converting gradient norms into function-value gaps to obtain a per-iteration contraction. Weak-learner imperfection is quantified by a *minimal cosine angle* (MCA) $\Theta > 0$, and we work under an *exact residual least-squares* fit assumption.

Let $\mathcal{F} = \mathrm{span}(\mathcal{B})$ be the empirical function space generated by the class of weak learners $\mathcal{B}$, with the inner product and norm defined above. We analyze the two SPA sequences (cf. Algorithm 2), writing

$$s_{m+1} = s_m + \eta\, b_m, \qquad x_{m+1} = (1 - c)\, x_m + c\, s_{m+1},$$

where s_m and x_m correspond to the exploratory and stable sequences, respectively (mapping the notation used earlier: $s_m \equiv G_m$ and $x_m \equiv F_m$). Here $b_m \in \mathcal{B}$ is the weak learner selected at iteration m, and $c \in (0, 1)$ and $\eta > 0$ are constants. We define the pseudo-residual $r_m \coloneqq -\nabla L(x_m) \in \mathcal{F}$ and the stable-sequence increment $\Delta_m \coloneqq x_m - x_{m-1}$ with $\Delta_0 \coloneqq 0$. The loss functional $L : \mathcal{F} \to \mathbb{R}$ is the empirical risk, consistent with $\mathcal{L}(F)$ in Eq. 2.1. All inner products and norms below are those of $\mathcal{F}$.

Assumption 1 (Smoothness)**.** *L is σ-smooth on $\mathcal{F}$: $\|\nabla L(f) - \nabla L(g)\| \le \sigma \|f - g\|$ for all $f, g \in \mathcal{F}$.*

Assumption 2 (Strong convexity)**.** *L is μ-strongly convex on $\mathcal{F}$: $L(g) \geq L(f) + \langle \nabla L(f), g - f \rangle + \frac{\mu}{2}\|g - f\|^2$.*

Assumption 3 (Scalable weak learners and MCA)**.** *$\mathcal{B}$ is scalable/symmetric: if $h \in \mathcal{B}$ and $\alpha \in \mathbb{R}$ then $\alpha h \in \mathcal{B}$. There exists $\Theta \in (0, 1]$ s.t. for any $r \neq 0$ there is $h \in \mathcal{B}$ with $\langle r, h \rangle \geq \Theta \|r\| \, \|h\|$.*

Assumption 4 (Exact residual least-squares)**.** *At round m, with $r_m = -\nabla L(x_m)$, the algorithm returns $b_m \in \arg\min_{\alpha \in \mathbb{R},\, h \in \mathcal{B}} \|r_m - \alpha h\|^2$.*

Assumption A4 idealizes tree fitting but is standard in boosting theory: it isolates optimization effects from representational constraints. Any practical inexactness can be modeled by a small additive tolerance in the LS objective; we treat the exact-LS regime here for clarity.

2.2.5.2 Key Lemmas

The following lemmas and corollaries are presented as the tools used in the Lyapunov analysis: a movement/kinetic identity and consequences, an alignment lemma exploiting exact least-squares (LS) under the MCA assumption, and a smoothness-based corollary tailored to the Lyapunov kinetic weight.

Lemma 1 (Movement / Kinetic recursion)**.** *Fix $c \in (0, 1)$ and $\eta > 0$, and define the SPA sequences in $\mathcal{F}$ by*

$$s_{m+1} = s_m + \eta \, b_m, \qquad x_{m+1} = (1 - c) \, x_m + c \, s_{m+1}.$$

Let $\Delta_m \coloneqq x_m - x_{m-1}$ and $\chi \coloneqq \frac{1-c}{c}$. Then, for every $m \geq 1$,

$$x_{m+1} - x_m = c\big(\chi \, \Delta_m + \eta \, b_m\big), \tag{2.16}$$

$$\|x_{m+1} - x_m\|^2 = c^2\Big(\chi^2 \|\Delta_m\|^2 + \eta^2 \|b_m\|^2 + 2\chi\eta \, \langle \Delta_m, b_m \rangle\Big). \tag{2.17}$$

Proof. From $x_{m+1} - x_m = c(s_{m+1} - x_m)$ and $s_{m+1} = s_m + \eta b_m$, note $s_m - x_m = (s_m - x_{m-1}) - (x_m - x_{m-1}) = \frac{1}{c}\Delta_m - \Delta_m = \chi \Delta_m$, so $s_{m+1} - x_m = \chi \Delta_m + \eta b_m$, yielding (2.16). Squaring yields (2.17). □

Corollary 1 (Kinetic inequality)**.** *For any $\theta > 0$ and every $m \geq 1$, applying Young's inequality $2\langle a, b\rangle \leq \theta\|a\|^2 + \frac{1}{\theta}\|b\|^2$ with $a = \chi\,\Delta_m$ and $b = \eta\,b_m$ in* (2.17) *gives*

$$\frac{1}{c^2}\|x_{m+1} - x_m\|^2 \;\leq\; (1+\theta)\,\chi^2\|\Delta_m\|^2 \;+\; \Big(1 + \frac{1}{\theta}\Big)\,\eta^2\|b_m\|^2. \tag{2.18}$$

Corollary 2 (Kinetic bound with Lyapunov weight)**.** *Specializing* (2.18) *to $\theta = 1$ and multiplying by $C = \frac{\sigma}{2\eta^2 c^2}$ yields*

$$C\,\|x_{m+1} - x_m\|^2 \;\leq\; \frac{\sigma}{\eta^2}\,\chi^2\,\|\Delta_m\|^2 \;+\; \sigma\,\|b_m\|^2. \tag{2.19}$$

Lemma 2 (Exact-LS alignment under MCA)**.** *Fix m and let $r_m = -\nabla L(x_m) \neq 0$. Under Assumptions A3–A4, the exact LS solution b_m satisfies*

$$\Theta\,\|r_m\| \;\leq\; \|b_m\| \;\leq\; \|r_m\|, \qquad \langle r_m, b_m\rangle \;=\; \|b_m\|^2, \qquad \|r_m - b_m\|^2 \;=\; \|r_m\|^2 - \|b_m\|^2 \;\leq\; (1-\Theta^2)\,\|r$$

(2.20)

Moreover, by scalability of $\mathcal{B}$, the minimizer belongs to $\mathcal{B}$: there exist $h^* \in \mathcal{B}$ and $\alpha^* \in \mathbb{R}$ such that $b_m = \alpha^* h^* \in \mathcal{B}$.

Proof. For any $h \in \mathcal{B}$ and $\alpha \in \mathbb{R}$, $\|r_m - \alpha h\|^2 = \|r_m\|^2 - 2\alpha\langle r_m, h\rangle + \alpha^2\|h\|^2$ is minimized at $\alpha^* = \langle r_m, h\rangle/\|h\|^2$, with value $\|r_m\|^2 - \langle r_m, h\rangle^2/\|h\|^2$. Thus exact LS over the scalable class maximizes $\langle r_m, h\rangle^2/\|h\|^2$, and returns $b_m = \alpha^* h$ with $\langle r_m, b_m\rangle = \|b_m\|^2$ and $\|r_m - b_m\|^2 = \|r_m\|^2 - \|b_m\|^2$. Cauchy–Schwarz gives $\|b_m\| \leq \|r_m\|$. By MCA, there exists h with $\langle r_m, h\rangle/\|h\| \geq \Theta\|r_m\|$; since LS maximizes this quantity, $\|b_m\| \geq \Theta\|r_m\|$. Scalability implies $b_m = \alpha^* h^* \in \mathcal{B}$. □

Angle/weak-learning constants of the form used in Lemma 2 are standard in boosting convergence analyses; see, e.g., [31, 32]. The representational constraint of $\mathcal{B}$ is captured by $\Theta \in (0, 1]$.

Corollary 3 (MCA window in gradient norm)**.** *With $r_m = -\nabla L(x_m)$, Lemma 2 implies*

$$\|b_m\| \;\geq\; \Theta\,\|\nabla L(x_m)\|, \qquad \textit{hence} \qquad \boxed{\|b_m\|^2 \;\geq\; \Theta^2\,\|\nabla L(x_m)\|^2.}$$

Lemma 3 (Descent at the primal point)**.** *Let L be σ-smooth (Assumption A1) and let $\{(x_m, s_m)\}$ be generated with constants $c \in (0,1)$ and $\eta > 0$. For any $m \geq 1$ and any $\tau > 0$,*

$$\boxed{L(s_{m+1}) \;\leq\; L(s_m) \;-\; \eta\Big(1 - \tfrac{\sigma\eta}{2} - \tfrac{\sigma\eta}{2\tau}\Big)\,\|b_m\|^2 \;+\; \frac{\sigma\tau}{2}\,\chi^2\,\|\Delta_m\|^2} \tag{2.21}$$

where $\Delta_m \coloneqq x_m - x_{m-1}$ and $\chi \coloneqq \frac{1-c}{c}$.

Proof. By σ-smoothness and $s_{m+1} = s_m + \eta b_m$,

$$L(s_{m+1}) \leq L(s_m) + \eta\langle \nabla L(s_m), b_m\rangle + \tfrac{\sigma}{2}\eta^2\|b_m\|^2.$$

Decompose the inner product: $\langle \nabla L(s_m), b_m\rangle = \langle \nabla L(x_m), b_m\rangle + \langle \nabla L(s_m) - \nabla L(x_m), b_m\rangle$. Exact LS (Assumption 4) gives $\langle \nabla L(x_m), b_m\rangle = -\|b_m\|^2$ (Lemma 2 with $r_m = -\nabla L(x_m)$). Smoothness and $s_m - x_m = \chi\,\Delta_m$ yield $\langle \nabla L(s_m) - \nabla L(x_m), b_m\rangle \leq \sigma\,\chi\,\|\Delta_m\|\,\|b_m\|$. Apply Young with parameter $\tau > 0$ to the cross-term and collect terms to obtain (2.21). □

Remark 1 (Connection to gradient norm)**.** *Using the MCA norm window from Lemma 2, $\|b_m\| \geq \Theta\,\|\nabla L(x_m)\|$, we obtain*

$$L(s_{m+1}) \;\leq\; L(s_m) \;-\; \eta\Big(1 - \tfrac{\sigma\eta}{2} - \tfrac{\sigma\eta}{2\tau}\Big)\,\Theta^2\,\|\nabla L(x_m)\|^2 \;+\; \frac{\sigma\tau}{2}\,\chi^2\,\|\Delta_m\|^2. \tag{2.22}$$

This is the form used in the proof of Theorem 1.

In the main proof, we use Lemma 3 with $\tau = 1$ and multiply the entire inequality by the Lyapunov weight $A = 1/\eta^2$; note that under exact LS there is *no* additive error term.

Lemma 4 (Gradient-gap inequality under strong convexity)**.** *If L is μ-strongly convex on $\mathcal{F}$ (Assumption 2), then for every $z \in \mathcal{F}$,*

$$\boxed{\|\nabla L(z)\|^2 \;\geq\; 2\mu\,\big(L(z) - L^*\big), \qquad L^* \coloneqq \min_{f\in\mathcal{F}} L(f).} \tag{2.23}$$

In particular, $\|\nabla L(x_m)\|^2 \geq 2\mu\big(L(x_m) - L^*\big)$ and $\|\nabla L(s_m)\|^2 \geq 2\mu\big(L(s_m) - L^*\big)$.

Proof. By μ-strong convexity, for any $f, g \in \mathcal{F}$,

$$L(g) \;\geq\; L(f) + \langle \nabla L(f),\, g - f \rangle + \frac{\mu}{2}\,\|g - f\|^2.$$

Let $f = z$ and $g = f^* \in \arg\min_{f \in \mathcal{F}} L(f)$ (so $L(f^*) = L^*$). Rearranging gives

$$L(z) - L^* \;\leq\; \langle \nabla L(z),\, z - f^* \rangle - \frac{\mu}{2}\,\|z - f^*\|^2.$$

The right-hand side is the maximum, over $d \in \mathcal{F}$, of $\langle \nabla L(z), d\rangle - \frac{\mu}{2}\|d\|^2$, attained at $d = \nabla L(z)/\mu$ with value $\frac{1}{2\mu}\|\nabla L(z)\|^2$. Hence $L(z) - L^* \leq \frac{1}{2\mu}\|\nabla L(z)\|^2$, which is equivalent to (2.23). □

Apply Lemma 4 at $z = x_m$ to the Lyapunov step inequality (2.25), together with the MCA window $\|b_m\| \geq \Theta\,\|\nabla L(x_m)\|$. This converts the $\|b_m\|^2$ term in (2.25) into a multiple of $L(x_m) - L^*$, yielding the one-step contraction used in Theorem 1.

2.2.5.3 Lyapunov Analysis

We introduce a Lyapunov potential that couples the step objective, the averaged objective, and a kinetic term. The weights are chosen so that, when combined with the movement identity and the descent inequality, cross-terms cancel and the remaining kinetic remainder carries a non-positive coefficient in our parameter regime.

The Lyapunov potential combines three ingredients: (i) the *step* objective $L(s_{m+1})$, where the update is applied; (ii) the *averaged* objective $L(x_m)$, where gradients are computed; and (iii) a *kinetic* term $\|x_{m+1} - x_m\|^2$ that absorbs two-point mismatches and mixed terms. The weights are selected so that the descent lemma (Lemma 3) and the kinetic recursion (Lemma 1) cancel mixed products and leave a single $\|\Delta_m\|^2$ remainder (the "kinetic bracket") with a non-positive coefficient inside our (η, c) safe region.

Definition 1 (Lyapunov potential)**.** *For $m \geq 0$, define*

$$\boxed{\Lambda_{m+1} \;:=\; A\left(L(s_{m+1}) - L^*\right) \;+\; B\left(L(x_m) - L^*\right) \;+\; C\,\|x_{m+1} - x_m\|^2,} \tag{2.24}$$

with the weights

$$A \coloneqq \frac{1}{\eta^2}, \qquad B \coloneqq \frac{\sigma\,\chi}{\eta} \quad \big(\chi \coloneqq \tfrac{1-c}{c}\big), \qquad C \coloneqq \frac{\sigma}{2\,\eta^2\,c^2}.$$

Rationale for the weights. $A = \eta^{-2}$ normalizes the descent bound in Lemma 3; $C = \frac{\sigma}{2\eta^2 c^2}$ matches the kinetic inequality in Corollary 2 so that mixed terms cancel when combining Lemma 1 with smoothness; $B = \frac{\sigma\chi}{\eta}$ ensures $\Lambda_{m+1} \geq B\,(L(x_m) - L^*)$, which will dominate the suboptimality at the stable sequence and is crucial in the final contraction step.

Remark 2 (Properties of the potential)**.** *(i) Non-negativity: Since $L(f) \geq L^*$ and the norm term is non-negative, $\Lambda_{m+1} \geq 0$, and in particular $\Lambda_{m+1} \geq B\,(L(x_m) - L^*)$.*

(ii) Balance: $A = \eta^{-2}$ aligns with the scale of Lemma 3, and $C = \frac{\sigma}{2\eta^2 c^2}$ is the kinetic coefficient induced by Lemma 1 with Young's parameter $\theta = 1$.

(iii) Safe region: The constraints $2\sigma\eta < 1$ and $c \leq 1 - 1/\sqrt{3}$ yield a clean descent inequality for the potential. Practical choices such as $c \in [0.05, 0.3]$ lie strictly inside this region.

Lemma 5 (Lyapunov step inequality)**.** *Let A, B, C be as in Definition 1. With Young parameters $\theta = \tau = 1$, for all $m \geq 1$,*

$$\boxed{\Big(\frac{1}{\eta} - 2\sigma\Big)\,\|b_m\|^2 \;\leq\; \Lambda_m - \Lambda_{m+1} \;-\; \underbrace{\frac{\sigma}{2\eta^2 c^2}\Big(1 - 3(1-c)^2\Big)}_{\textsf{bracket}}\,\|\Delta_m\|^2\,.} \tag{2.25}$$

Proof. From Definition 1,

$$\Lambda_m - \Lambda_{m+1} = A\big(L(s_m) - L(s_{m+1})\big) + B\big(L(x_{m-1}) - L(x_m)\big) + C\big(\|\Delta_m\|^2 - \|\Delta_{m+1}\|^2\big). \tag{2.26}$$

(i) Descent term. Lemma 3 with $\tau = 1$, multiplied by $A = \eta^{-2}$, gives

$$A\big(L(s_m) - L(s_{m+1})\big) \;\geq\; \frac{1 - \sigma\eta}{\eta}\,\|b_m\|^2 \;-\; \frac{\sigma}{2\eta^2}\,\chi^2\,\|\Delta_m\|^2. \tag{2.27}$$

(ii) Kinetic term. By Corollary 1 with $\theta = 1$ and $C = \frac{\sigma}{2\eta^2 c^2}$,

$$C\,\|\Delta_{m+1}\|^2 \;\le\; \frac{\sigma}{\eta^2}\,\chi^2\,\|\Delta_m\|^2 \;+\; \sigma\,\|b_m\|^2,$$

so

$$-\,C\,\|\Delta_{m+1}\|^2 \;\ge\; -\,\frac{\sigma}{\eta^2}\,\chi^2\,\|\Delta_m\|^2 \;-\; \sigma\,\|b_m\|^2. \tag{2.28}$$

(iii) Assembly. Substitute (2.27) and (2.28) into (2.26):

$$\begin{aligned}\Lambda_m - \Lambda_{m+1} &\ge \Big(\tfrac{1-\sigma\eta}{\eta} - \sigma\Big)\,\|b_m\|^2 \;+\; B\big(L(x_{m-1}) - L(x_m)\big) \\ &\quad +\; \Big(C - \tfrac{\sigma}{2\eta^2}\chi^2 - \tfrac{\sigma}{\eta^2}\chi^2\Big)\,\|\Delta_m\|^2 \\ &= \Big(\tfrac{1}{\eta} - 2\sigma\Big)\,\|b_m\|^2 \;+\; B\big(L(x_{m-1}) - L(x_m)\big) \;+\; \underbrace{\Big(\frac{\sigma}{2\eta^2 c^2} - \frac{3\sigma}{2\eta^2}\chi^2\Big)}_{\textsf{bracket}}\,\|\Delta_m\|^2.\end{aligned}$$

Using $\chi^2 = \frac{(1-c)^2}{c^2}$ and $C = \frac{\sigma}{2\eta^2 c^2}$ yields $\textsf{bracket} = \frac{\sigma}{2\eta^2 c^2}\big(1 - 3(1-c)^2\big)$. Moving the bracket term to the right-hand side gives (2.25). □

We now enforce $c \le 1 - 1/\sqrt{3}$ so the bracket in (**??**) is nonpositive and can be dropped, and require $2\sigma\eta < 1$ so the coefficient $(1/\eta - 2\sigma)$ of $\|b_m\|^2$ is positive.

Corollary 4 (Negative-bracket regime)**.** *If $c \le 1 - 1/\sqrt{3}$, then* $\textsf{bracket} = \frac{\sigma}{2\eta^2 c^2}\big(1 - 3(1-c)^2\big) \le 0$*, and Lemma 5 simplifies to*

$$\boxed{\Big(\frac{1}{\eta} - 2\sigma\Big)\,\|b_m\|^2 \;\le\; \Lambda_m - \Lambda_{m+1}.} \tag{2.29}$$

2.2.5.4 Main Convergence Result

Theorem 1 (Linear convergence of SPA-GBM (exact LS))**.** *Assume **A1**–**A4**. Let the SPA-GBM parameters satisfy*

$$2\sigma\,\eta \;<\; 1 \qquad \textit{and} \qquad c \;\le\; 1 - \frac{1}{\sqrt{3}}\,.$$

Define the Lyapunov potential Λ_{m+1} as in Definition 1, and set

$$\alpha \;:=\; 2\mu\,\Theta^2\Big(\tfrac{1}{\eta} - 2\sigma\Big), \qquad \rho \;:=\; \frac{\alpha}{\alpha + B} \in (0,1), \quad \textit{where } B = \tfrac{\sigma\chi}{\eta},\ \chi = \tfrac{1-c}{c}.$$

Then, for all $m \geq 1$,

$$\boxed{\Lambda_{m+1} \;\leq\; (1-\rho)\,\Lambda_m} \tag{2.30}$$

and consequently $\Lambda_m \leq (1-\rho)^{m-1}\Lambda_1$. In particular,

$$L(x_m) - L^* \;\leq\; \frac{1}{B}\,\Lambda_{m+1} \;\leq\; \frac{(1-\rho)^m}{B}\,\Lambda_1,$$

i.e., the stable sequence x_m converges linearly to the minimizer in $\mathcal{F}$.

Proof. By Corollary 4, the bracket term in Lemma 5 is nonpositive under $c \leq 1 - 1/\sqrt{3}$, hence

$$\left(\tfrac{1}{\eta} - 2\sigma\right) \|b_m\|^2 \;\leq\; \Lambda_m - \Lambda_{m+1}.$$

Using the MCA window $\|b_m\| \geq \Theta\,\|\nabla L(x_m)\|$ (Corollary 3) yields

$$\left(\tfrac{1}{\eta} - 2\sigma\right) \Theta^2\,\|\nabla L(x_m)\|^2 \;\leq\; \Lambda_m - \Lambda_{m+1}.$$

Applying the gradient–gap inequality (Lemma 4), $\|\nabla L(x_m)\|^2 \geq 2\mu\,[L(x_m) - L^*]$, we obtain

$$\alpha\,[L(x_m) - L^*] \;\leq\; \Lambda_m - \Lambda_{m+1}, \qquad \alpha = 2\mu\,\Theta^2\left(\tfrac{1}{\eta} - 2\sigma\right) > 0$$

(the positivity follows from $2\sigma\eta < 1$). By Definition 1, $\Lambda_{m+1} \geq B\,[L(x_m) - L^*]$, hence

$$(1 + \alpha/B)\,\Lambda_{m+1} \;\leq\; \Lambda_m.$$

Letting $\rho = \alpha/(\alpha + B)$ gives (2.30). Unrolling the recursion and using $\Lambda_{m+1} \geq B\,[L(x_m) - L^*]$ yields the stated geometric decay of $L(x_m) - L^*$. □

Remark 3 (Specific parameter choices)**.** *With $\eta = \dfrac{1}{2\sigma}\dfrac{c(2-c)}{1-c}$, the condition $2\sigma\eta < 1$ is equivalent to $c^2 - 3c + 1 > 0$, which on $(0,1)$ gives $c \leq \frac{3-\sqrt{5}}{2} \approx 0.38197$. Together with $c \leq 1 - 1/\sqrt{3} \approx 0.42265$, the admissible range is*

$$c \in \left(0, \frac{3-\sqrt{5}}{2}\right].$$

In this setting, the contraction factor is

$$\rho \;=\; \frac{2\,\mu\,\Theta^2\,c\,(c^2 - 3c + 1)}{2\,\mu\,\Theta^2\,c\,(c^2 - 3c + 1) + \sigma\,(1-c)^2}.$$

The SPA averaging scheme, within its (η, c) safe region, preserves a non-positive kinetic bracket *without* correction. This completes the two-sequence Lyapunov analysis for SPA-GBM under exact residual least-squares and the MCA assumption.

2.3 Experiments

We evaluate the proposed multi-sequence boosting variants on seven real-world tabular regression suites and a controlled noise-injection study. We first detail the experimental protocol (datasets, preprocessing, models, and tuning), then examine convergence under injected label noise, and finally analyze predictive performance on the real datasets.

2.3.1 Experimental Setup

Datasets and feature structure. We target diverse tabular regimes spanning sample size, feature mix, and noise structure.

Abalone A marine-biology regression task predicting abalone age (via `Rings`) from sex and shell measurements. combines one categorical feature (`Sex` M,F,I) with seven continuous measurements (morphology, weights). Target is `Rings` (age proxy). We one-hot encode `Sex` and standardize numeric features. The distribution is moderately skewed with outliers (e.g., `height`).

Ames Housing A real-estate valuation regression task predicting home sale price from property, condition, and neighborhood attributes. is a mixed-type benchmark ($\tilde{7}$0–80 attributes) with nominal/ordinal categories, meaningful 'NA" encodings (e.g., absence), and heterogeneous scales. We impute numeric features by median and one-hot encode categoricals, collapsing singleton levels into an 'other" bucket to avoid overly sparse design matrices. Targets remain on the natural scale.

CASP A protein-structure regression task predicting tertiary-structure quality (RMSD) from physicochemical descriptors. is fully numeric (9 predictors), no missingness. We standardize features; the target is RMSD.

Communities and Crime A social-science/criminology regression task predicting violent-crime rates from community socio-economic and demographic

indicators. is high-dimensional with scaled numeric features capturing socio-economic/demographic statistics. We drop identifiers, standardize *after* split to avoid leakage, and impute any remaining missing values. The violent-crime target is heavy-tailed and sensitive to outliers.

Superconductivity A materials-science regression task predicting superconductors' critical temperature T_c from composition-derived descriptors. consists of engineered, fully numeric descriptors derived from composition; features are moderately high-dimensional and collinear. We standardize features; the target (critical temperature) has a wide dynamic range.

YearPredictionMSD A music-information-retrieval regression task predicting a song's release year from audio features. is large-scale (90 continuous features) and roughly half a million examples in the original corpus.

For each dataset we use five independent 60/20/20 train/validation/test splits (fixed seeds shared across methods). All preprocessing steps (imputation, encoding, standardization) are fit on the training fold only and then applied to validation and test to prevent leakage. We report mean ± standard error over seeds for of root mean squared error (RMSE), mean absolute error (MAE), and R^2 score on the test dataset.

We compare strong tree boosting baselines (LightGBM, XGBoost), momentum/acceleration baselines (VAGM [25], AGBM [26]), and four implementations of our optimization principles using identical trees: SPA-GBM (Sec. 2.2.2) maintains exploratory G_t and stable F_t sequences and computes pseudo-residuals at $y_t \equiv F_t$; Adam-SPA-GBM (Sec. 2.2.3) augments SPA-GBM with per-sample second-moment scaling of residuals; Schedule-Free GBM (Sec. 2.2.4) replaces hand-tuned averaging with $\gamma_t = \frac{1}{t+1}$ while preserving $y_t \equiv F_t$; Adam–SF-GBM combines schedule-free averaging with variance-aware residual scaling.

All methods use the same folds, early-stopping policy, and overlapping search spaces for fairness. We perform random search with multiple trials per model–dataset pair and select the configuration with the best validation MSE.

The tree budget is up to $T_{\max} = 5000$ learners with patience 200 for early stopping, tracking the MSE metric. Learning rates are drawn from $\{0.01, 0.03, 0.1, 0.3\}$ with standard depth/leaf regularization ranges. For SPA-GBM and ADAM-SPA-GBM we search $c \in \{0.05, 0.1, 0.2, 0.3\}$ within the safe region established in Theorem. 1 and tie the step-size multiplier to the tree shrinkage to maintain comparability. Schedule-free variants fix $\gamma_t = \frac{1}{t+1}$; to stabilize very early iterations we apply a short base-shrinkage warm-up for $T_{\text{warm}} \in \{0, 100, 250\}$ steps (uniformly across schedule-free methods). For adaptive residuals we search $\beta_2 \in \{0.9, 0.95, 0.99\}$ with $\epsilon = 10^{-8}$. All runs use a multi-core CPU workstation; thread counts and library versions are fixed across methods.

2.3.2 Convergence Under Noise: Controlled Study

To probe stability under stochastic targets, we inject Gaussian label noise at a single level by defining $\tilde{y} = y + 0.2\,\mathrm{std}(y) \cdot \varepsilon$ with $\varepsilon \sim \mathcal{N}(0, 1)$. We compare training-loss trajectories for LightGBM, SPA-GBM, and Adam–SPA-GBM. The synthetic data are generated to stress "greedy myopia": base features $X \in \mathbb{R}^{10{,}000 \times 50}$ are drawn from a correlated Gaussian with covariance $\Sigma = (A^\top A)/d + \rho I$ ($\rho = 0.2$), component-wise medians define thresholds, and the target combines weak marginal threshold effects $\sum_j \alpha_j \mathbf{1}\{x_j > t_j\}$ with a stronger, harder-to-discover interaction $3\,\mathbf{1}\{x_0 > t_0 \wedge x_1 < t_1\}$; additive noise with $\sigma = 0.2$ is applied, and 20 distractor features are appended. This construction yields informative but individually weak marginals alongside a dominant interaction, encouraging stage-wise learners to follow short-term residual reductions rather than uncovering the true signal.

Consistent with Sec. 2.2.5, primal averaging damps high-frequency oscillations by preventing any single misfit weak learner from dominating the stable model; computing gradients at the evaluation sequence $y_t \equiv F_t$ further decouples exploratory variance from the next residual. Adam–SPA-GBM adds variance-aware residual scaling (§2.2.3), which provides additional smoothing in this heterogeneous-residual regime. Empirically, the schedule-free variants trace

smoother, more monotone training curves than LightGBM under identical early-stopping and shrinkage settings, aligning with the theoretical role of averaging and the gradient-evaluation sequence in moderating noisy updates.

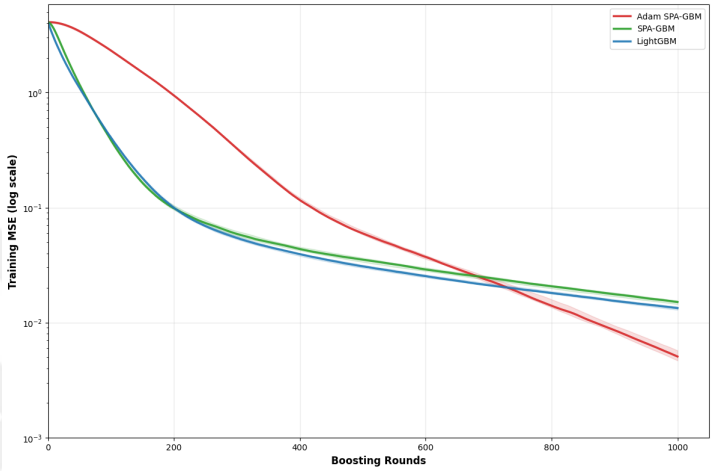


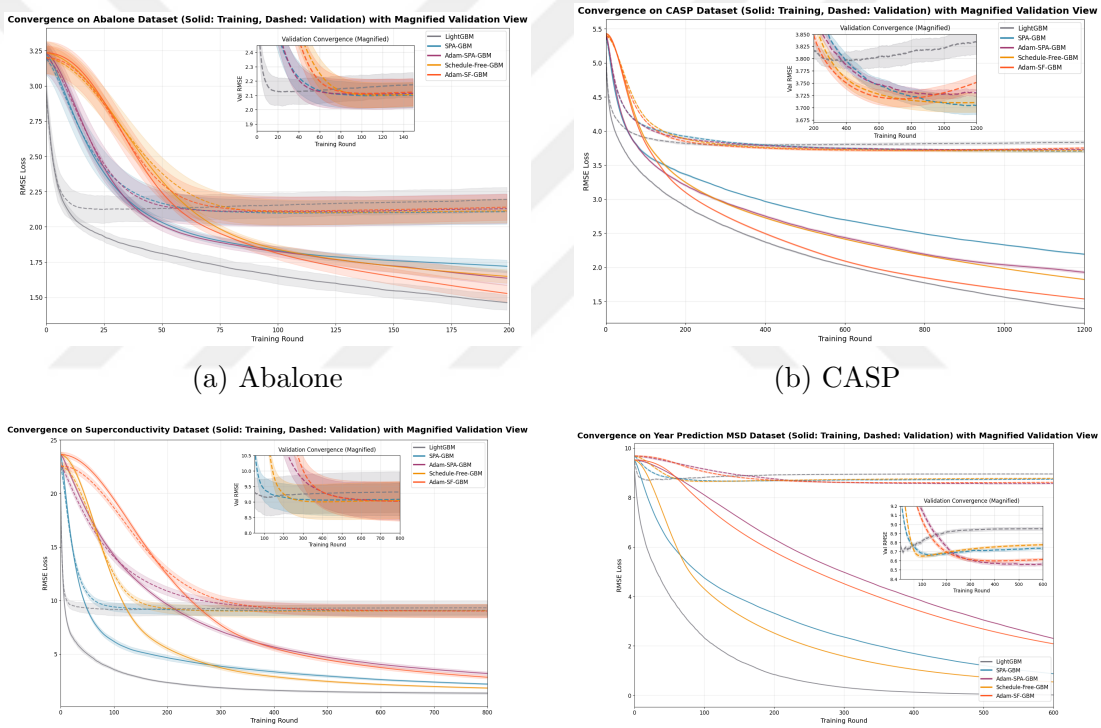
Figure 2.1: Training loss vs. iteration for synthetic dataset. As demonstrated in our theoretical analysis, SPA-GBM's linear convergence rate matches that of LightGBM

We also visualize train/validation dynamics on the selected datasets as an example for five methods: LightGBM, SPA-GBM, Adam-SPA-GBM, Schedule-Free GBM, and Adam–Schedule-Free GBM. SPA-based methods typically show reduced validation volatility and earlier plateaus, aligning with iterate-averaging effects and the use of the gradient-evaluation sequence $y_t \equiv F_t$.

2.3.3 Real-World Results and Discussion

We report RMSE, MAE, and R^2 across benchmarks for all eight methods. Tables/figures below contain the measured means $\pm$ standard errors over five seeds; best and second-best per dataset are highlighted in **bold** and underline, respectively.

Across datasets (Table 2.1-2.6), the observed gains of our variants can be traced to three interacting design choices that affect how learning signals are produced



(a) Abalone (b) CASP

(c) Superconductivity (d) Yearly Prediction MSD

Figure 2.2: Training (solid) and validation (dashed) loss vs. iteration on representative datasets. Averaging and schedule-free interpolation yield smoother validation curves.

Method	RMSE	MAE	R^2
LightGBM	0.130 ± 0.023	0.089 ± 0.008	0.906 ± 0.011
XGBoost	0.140 ± 0.018	0.090 ± 0.010	0.906 ± 0.010
VAGM	0.148 ± 0.014	0.099 ± 0.010	0.879 ± 0.013
AGBM	0.137 ± 0.019	0.089 ± 0.007	0.896 ± 0.009
SPA-GBM	0.126 ± 0.011	0.086 ± 0.008	0.911 ± 0.010
Adam-SPA-GBM	0.128 ± 0.008	0.086 ± 0.007	0.909 ± 0.008
Schedule-Free-GBM	0.126 ± 0.012	**0.083 ± 0.011**	0.912 ± 0.007
Adam-SF-GBM	**0.125 ± 0.007**	0.085 ± 0.009	**0.913 ± 0.011**

Table 2.1: Comparison of RMSE, MAE and R^2 scores on Ames dataset.

Method	RMSE	MAE	R^2
LightGBM	2.253 ± 0.128	1.555 ± 0.041	0.523 ± 0.010
XGBoost	2.263 ± 0.141	1.590 ± 0.037	0.509 ± 0.007
VAGM	2.294 ± 0.119	1.586 ± 0.046	0.509 ± 0.011
AGBM	2.244 ± 0.105	1.553 ± 0.045	0.529 ± 0.010
SPA-GBM	**2.151 ± 0.116**	1.519 ± 0.036	**0.549 ± 0.016**
Adam-SPA-GBM	2.160 ± 0.127	**1.503 ± 0.046**	0.546 ± 0.016
Schedule-Free-GBM	2.156 ± 0.117	1.520 ± 0.039	0.547 ± 0.014
Adam-SF-GBM	2.159 ± 0.126	1.505 ± 0.045	0.546 ± 0.016

Table 2.2: Comparison of RMSE, MAE and R^2 scores on Abalone dataset.

and consumed over time. First, computing pseudo-residuals at a *stable* gradient-evaluation sequence ($y_t \equiv F_t$) decouples noisy exploratory updates from the model state that generates the next learning signal; this limits feedback loops in which a single misfit tree perturbs subsequent residuals. Second, replacing a hand-tuned averaging schedule with the parameter-free interpolation $\gamma_t = 1/(t+1)$ avoids dataset-specific schedule mis-specification and systematically smooths validation trajectories. Third, scaling residuals by a per-sample second-moment estimate attenuates the impact of heterogeneous or outlier-driven targets without requiring a global reduction of the shrinkage. These mechanisms are complementary: averaging at $y_t \equiv F_t$ reduces the variance of the gradient signal itself; schedule-free interpolation removes a sensitive tuning knob; and variance-aware scaling normalizes difficult examples rather than shrinking all updates. Overall, schedule-free variants are typically at the forefront across tasks, while SPA variants also show consistent improvements over strong GBM baselines.

The dataset-level trends align with these hypotheses. On *Ames*, where mixed categorical/ordinal structure and heteroscedastic targets are common, both

Method	RMSE	MAE	R^2
LightGBM	3.886 ± 0.041	2.847 ± 0.025	0.597 ± 0.011
XGBoost	3.867 ± 0.044	2.819 ± 0.026	0.610 ± 0.010
VAGM	4.062 ± 0.054	3.081 ± 0.035	0.547 ± 0.013
AGBM	3.944 ± 0.038	2.943 ± 0.031	0.608 ± 0.012
SPA-GBM	**3.718 ± 0.046**	2.707 ± 0.031	**0.650 ± 0.012**
Adam-SPA-GBM	3.768 ± 0.047	**2.632 ± 0.038**	0.634 ± 0.012
Schedule-Free-GBM	3.732 ± 0.040	2.712 ± 0.025	0.645 ± 0.010
Adam-SF-GBM	3.765 ± 0.040	2.652 ± 0.031	0.635 ± 0.010

Table 2.3: Comparison of RMSE, MAE and R^2 scores on CASP dataset.

Method	RMSE	MAE	R^2
LightGBM	0.151 ± 0.001	0.099 ± 0.001	0.617 ± 0.009
XGBoost	0.148 ± 0.002	0.100 ± 0.001	**0.621 ± 0.007**
VAGM	0.148 ± 0.001	0.098 ± 0.001	0.620 ± 0.007
AGBM	0.149 ± 0.002	0.105 ± 0.002	0.608 ± 0.011
SPA-GBM	0.147 ± 0.001	**0.097 ± 0.001**	0.617 ± 0.010
Adam-SPA-GBM	0.147 ± 0.001	0.099 ± 0.001	0.615 ± 0.007
Schedule-Free-GBM	**0.146 ± 0.001**	**0.097 ± 0.001**	**0.621 ± 0.005**
Adam-SF-GBM	0.148 ± 0.001	0.099 ± 0.002	**0.621 ± 0.008**

Table 2.4: Comparison of RMSE, MAE and R^2 scores on Communities&Crime dataset.

schedule-free and SPA variants reduce MAE and RMSE relative to strong tree baselines, consistent with iterate averaging at $y_t \equiv F_t$ regularizing early, high-variance steps without sacrificing asymptotic fit. *Communities & Crime* exhibits heavy tails in the response; here our methods typically match or edge out baselines with lower variability across seeds, although the margins are small and several methods are statistically tied, so conclusions should be treated cautiously. In *Abalone*, improvements are smaller but consistent; measurement noise and a coarse one-hot for `Sex` appear to benefit from schedule-free smoothing and gentle normalization of residual magnitudes. On *CASP* (low-dimensional, well-scaled features), schedule-free and SPA-GBM variants both attain stronger RMSE while remaining competitive in MAE, suggesting that stabilized descent rather than aggressive shrinkage explains the gains. For *Superconductivity*, moderate collinearity and a wide target range favor smoothing; schedule-free variants tend to be best or tied, in line with their reduced sensitivity to averaging schedules. Finally, on *YearPredictionMSD*, which is the largest in scale, ADAM-SPA-GBM and ADAM-SF-GBM show especially impressive improvements.

Method	RMSE	MAE	R^2
LightGBM	9.843 ± 0.413	5.805 ± 0.122	0.918 ± 0.008
XGBoost	9.851 ± 0.380	5.935 ± 0.105	0.913 ± 0.003
VAGM	10.219 ± 0.513	6.267 ± 0.120	0.882 ± 0.003
AGBM	10.069 ± 0.441	6.001 ± 0.152	0.898 ± 0.005
SPA-GBM	9.791 ± 0.453	5.800 ± 0.113	0.920 ± 0.006
Adam-SPA-GBM	**9.511 ± 0.209**	**5.568 ± 0.054**	**0.931 ± 0.004**
Schedule-Free-GBM	9.719 ± 0.504	5.777 ± 0.146	0.923 ± 0.007
Adam-SF-GBM	9.568 ± 0.199	5.635 ± 0.059	0.929 ± 0.004

Table 2.5: Comparison of RMSE, MAE and R^2 scores on Superconductivity dataset.

Method	RMSE	MAE	R^2
LightGBM	8.831 ± 0.035	6.229 ± 0.016	0.346 ± 0.002
XGBoost	8.811 ± 0.027	6.249 ± 0.019	0.364 ± 0.001
VAGM	9.028 ± 0.032	6.398 ± 0.014	0.325 ± 0.001
AGBM	9.116 ± 0.020	6.497 ± 0.008	0.311 ± 0.002
SPA-GBM	8.596 ± 0.026	6.020 ± 0.012	0.474 ± 0.001
Adam-SPA-GBM	**8.482 ± 0.029**	**5.820 ± 0.016**	**0.535 ± 0.001**
Schedule-Free-GBM	8.580 ± 0.027	6.017 ± 0.013	0.482 ± 0.001
Adam-SF-GBM	8.509 ± 0.028	5.910 ± 0.015	0.520 ± 0.001

Table 2.6: Comparison of RMSE, MAE and R^2 scores on Year Prediction MSD dataset.

Training/validation curves (Fig. 2.2) supports our stability claims: under the same early-stopping and search budgets, SPA-based methods generally plateau earlier and exhibit less validation volatility. This mirrors the theory that averaging in function space, especially when residuals are computed at $y_t \equiv F_t$, attenuates high-variance steps, yielding a smoother optimization path without damping useful signal.

Sensitivity analyses were qualitatively consistent across tasks. Smaller averaging coefficients c increase stability but can slow progress on very clean datasets; values in $[0.1, 0.3]$ worked well across the board. For adaptive residuals, $\beta_2 \in [0.95, 0.99]$ trades responsiveness for smoothness; larger β_2 is preferred on noisier labels. A brief shrinkage warm-up improves early-phase stability for schedule-free variants in high-noise settings without materially changing final metrics.

We note two threats to validity. First, we report five seeds per dataset may shift with more seeds or full-corpus training. Second, preprocessing choices (e.g.,

rare-level bucketing for categoricals, imputation strategy) can influence absolute errors. We mitigate these via consistent, leakage-free pipelines and shared folds across methods, and we will release code, exact folds, and configuration files. Each table entry averages five seeds with identical preprocessing, early stopping, and library versions; thread counts are fixed to reduce nondeterminism. For small deltas, particularly on *Communities & Crime*, where results are very close—paired seed-wise tests are recommended to assess robustness.

Chapter 3

Conclusion

We reframed gradient boosting as a multi-sequence optimization process that separates exploration from stability. The method maintains an exploratory sequence that aggregates aggressive weak-learner updates and a stable sequence that both predicts and supplies the next pseudo-residuals. This simple decoupling curbs error feedback from noisy updates while preserving the strengths of greedy tree fitting. On top of this structure, we used iterate averaging in function space and a schedule-free interpolation rule to remove hand-tuned averaging schedules. We also introduced a lightweight per-sample second-moment normalization of pseudo-residuals, which tempers heteroscedastic or outlier-driven updates without shrinking all steps globally. Theoretically, a two-sequence Lyapunov analysis establishes linear convergence for the base model under standard smoothness and strong convexity assumptions together with a minimal cosine angle condition on the weak learners. Empirically, across diverse tabular regression tasks, the proposed variants match or improve accuracy over strong baselines while reducing validation volatility and tuning burden. Limitations include the reliance on exact residual fits and convex losses in the analysis, as well as the per-sample memory needed for primal sequence or variance tracking. Future work will extend the template to inexact fitting, nonconvex objectives, and alternative interpolation rules that further reduce tuning while remaining faithful to the additive, function-space nature of boosting.

Bibliography

[1] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, ACM, 2016.

[2] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems 30* (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), pp. 3146–3154, Curran Associates, Inc., 2017.

[3] L. Prokhorenkova, D. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: Unbiased boosting with categorical features," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 6638–6648, Curran Associates, Inc., 2018.

[4] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[5] L. Mason, J. Baxter, P. L. Bartlett, and M. Frean, "Functional Gradient Techniques for Combining Hypotheses," in *Advances in Large Margin Classifiers* (A. J. Smola, P. L. Bartlett, B. Schölkopf, and D. Schuurmans, eds.), pp. 221–246, Cambridge, MA, USA: MIT Press, 2000.

[6] M. Telgarsky, "A Primal–Dual View of Boosting," *Journal of Machine Learning Research*, vol. 13, no. 18, pp. 561–606, 2012.

[7] P. Bühlmann and B. Yu, "Boosting With the L2 Loss: Regression and Classification," *Journal of the American Statistical Association*, vol. 98, no. 462, pp. 324–339, 2003.

[8] T. Zhang and B. Yu, "Boosting with Early Stopping: Convergence and Consistency," *The Annals of Statistics*, vol. 33, no. 4, pp. 1538–1572, 2005.

[9] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning*, vol. 28 of *ICML'13*, pp. 1139–1147, PMLR, 2013.

[10] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014. Published at ICLR 2015.

[11] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM Journal on Control and Optimization*, vol. 30, no. 4, pp. 838–855, 1992.

[12] Y. Nesterov, "A method for solving a convex programming problem with convergence rate $O(1/k^2)$," *Soviet Mathematics Doklady*, *vol.* 27, *no.* 2, *pp.* 372−−376, 1983.

[13] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[14] O. Devolder, F. Glineur, and Y. Nesterov, "First-order methods of smooth convex optimization with inexact oracle," *Mathematical Programming*, vol. 145, no. 1–2, pp. 37–77, 2014.

[15] N. Flammarion and F. Bach, "From Averaging to Acceleration, There is Only a Step-size," *arXiv preprint arXiv:1504.06297*, 2015.

[16] A. Cutkosky, "Momentum/Averaging Generalizations in Online Convex Optimization," *arXiv preprint arXiv:1905.09264*, 2019.

[17] M. Zamani and F. Glineur, "Last-iterate convergence of cyclic and stochastic gradient descent with and without momentum," *arXiv preprint arXiv:2302.04640*, 2023.

[18] J. Zhuang, T. Tang, Y. Ding, S. Tatikonda, N. Dvornek, X. Papademetris, and J. S. Duncan, "AdaBelief Optimizer: Adapting Stepsizes by the Belief in Observed Gradients," in *Advances in Neural Information Processing Systems 33*, pp. 18795–18806, Curran Associates, Inc., 2020.

[19] S. Bubeck, *Convex Optimization: Algorithms and Complexity*, vol. 8. Now Publishers, 2015.

[20] B. T. Polyak, "Some methods of speeding up the convergence of iteration methods," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, no. 5, pp. 1–17, 1964.

[21] W. Tao, Y. Gao, and Y. Li, "Primal Averaging: A New Gradient Evaluation Step for First-Order Methods," *arXiv preprint arXiv:1806.01416*, 2018.

[22] A. Defazio, "Momentum is a Primal Averaging Method," *arXiv preprint arXiv:2002.05739*, 2020.

[23] A. Kavis, A. Mokhtari, and V. Cevher, "Primal Averaging and Accelerated Learning in Online Convex Optimization," in *Advances in Neural Information Processing Systems 32*, pp. 7382–7392, 2019.

[24] A. Fouillen, T. Pethick, N. Durrande, C. Brouard, K. Szafnicki, S. Gey, and F. d'Alché Buc, "Accelerated Proximal Boosting," *arXiv preprint arXiv:1806.02324*, 2018.

[25] G. Biau, B. Cadre, and L. Rouvière, "Accelerated Gradient Boosting," *arXiv preprint arXiv:1906.06950*, 2019.

[26] Y. Lu and R. Mazumder, "Accelerated Gradient Boosting," in *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics*, vol. 108 of *AISTATS 2020*, pp. 1560–1570, PMLR, 2020.

[27] Y. Lu and R. Mazumder, "Randomized Gradient Boosting Machine," *arXiv preprint arXiv:1810.09995*, 2018.

[28] J. Wang, T. Zhang, and P. Zhao, "Frank-Wolfe Boosting for General Loss Functions," *arXiv preprint arXiv:1506.01217*, 2015.

[29] A. Defazio, "Momentum is a Primal Averaging Method," *arXiv preprint arXiv:2002.05739*, 2020. This key points to the same paper as defazio2020.

[30] A. Defazio and R. M. Gower, "The Road Less Scheduled," *arXiv preprint arXiv:2106.12130*, 2021.

[31] C. Wang, Y. Wang, W. E, and R. E. Schapire, "Functional Frank–Wolfe Boosting for General Loss Functions," *Journal of Machine Learning Research*, vol. 18, no. 21, pp. 1–47, 2017. Published version of arXiv:1510.02558.

[32] H. Lu and R. Mazumder, "Randomized Gradient Boosting Machine," *SIAM Journal on Optimization*, vol. 30, no. 4, pp. 2555–2587, 2020.