https://doi.org/10.58225/sw.2026.1-53-61

MODELING CONVERGENCE RATE ESTIMATES OF THE HEAVY BALL ALGORITHM VIA A RANDOM FOREST WITH SELF-ATTENTION

Daniil Miroshnichenko[1*], Majid Abbasov[2]
[1*,2]*St. Petersburg State University, Saint Petersburg, Russia*
miroshnichienko.daniil@gmail.com, m.abbasov@spbu.ru

Abstract. This paper addresses the problem of estimating convergence rates of iterative optimization algorithms, which is essential for analyzing and accelerating the training of modern machine learning models. Classical approaches to convergence rate estimation for momentum-based methods typically rely on analytical bounds derived under restrictive assumptions or on extensive empirical experimentation, both of which are computationally expensive and poorly scalable. As an alternative, we propose a data-driven framework for modeling convergence rate estimates using supervised machine learning. The proposed approach is demonstrated for the Heavy Ball optimization method applied to quadratic objective functions with heterogeneous curvature. A synthetic dataset is constructed by simulating Heavy Ball trajectories under randomly sampled objective parameters, optimization hyperparameters, and normalized initial conditions. The input features capture key characteristics of the optimization process, while the target variable represents a logarithmic estimate of the objective function decay along the trajectory, serving as a proxy for the convergence rate. To model nonlinear dependencies between optimization parameters and convergence behavior, we employ a Random Forest model augmented with a self-attention mechanism. This architecture enables adaptive weighting of feature interactions and improves predictive accuracy without requiring explicit analytical modeling of the underlying dynamics. Numerical experiments show that the proposed model achieves a coefficient of determination exceeding 0.8 when predicting convergence-related quantities, indicating strong agreement with empirically observed behavior. The results suggest that machine learning-based surrogate models can substantially simplify convergence rate estimation and that the proposed methodology is applicable to other optimization algorithms and objective classes.

Keywords: Convergence rate estimation, heavy ball method, momentum-based optimization, random Forest with self-attention, synthetic trajectory modeling, optimization dynamics

Cite this article as: *D. Miroshnichenko and M. Abbasov, "MODELING CONVERGENCE RATE ESTIMATES OF THE HEAVY BALL ALGORITHM VIA A RANDOM FOREST WITH SELF-ATTENTION", SW AzUAC, vol. 1, 2026, doi:10.58225/sw.2026.1-53-61*
Article history: *Received: 16 April 2026 / Revised: 13 June 2026 / Accepted: 23 June 2026*

1. Introduction. The analysis of convergence rates constitutes a fundamental problem in numerical optimization and plays a central role in the training of modern machine learning models. In large-scale learning settings, optimization algorithms are often executed for a very large number of iterations, and their convergence behavior directly determines computational cost, resource consumption, and overall training efficiency. Reliable estimates of convergence rates are therefore essential for selecting suitable optimization algorithms, tuning hyperparameters such as step sizes and momentum coefficients, and predicting the time required to reach a prescribed accuracy level. These considerations are especially important in neural network training, where optimization dynamics interact with high-dimensional parameter spaces and complex loss landscapes.

Traditionally, convergence rate estimates are obtained through either analytical derivations or empirical evaluation. Analytical approaches rely on structural assumptions on the objective function, such as Lipschitz continuity of the gradient and strong convexity, and provide worst-case bounds on the decay of the optimization error [1],[2]. While such results offer valuable theoretical insight, they

are often conservative and may not accurately reflect the behavior observed in practice. Moreover, deriving tight bounds for momentum-based methods remains challenging even for relatively simple objective classes. Empirical approaches typically involve repeated runs of the optimization algorithm followed by fitting convergence models to observed objective values or gradient norms, for instance via linear regression in logarithmic scale. Although more flexible, such procedures are computationally expensive, sensitive to noise, and difficult to automate or transfer across different problem settings.

These limitations motivate alternative approaches that avoid explicit analytical modeling of optimization dynamics. In this work, we adopt a data-driven perspective and treat the optimization algorithm as a black-box dynamical system whose convergence behavior can be learned from observed trajectories. Instead of deriving convergence rates from first principles, we propose to approximate them using supervised machine learning models trained on synthetically generated optimization data. This viewpoint is closely related to recent research on learning-based analysis and design of optimization algorithms, where machine learning techniques are used to predict performance characteristics or guide algorithmic decisions [3,4].

As a representative and technically challenging case study, we focus on the Heavy Ball method originally introduced by Polyak [1], which augments standard gradient descent with an inertial term in order to accelerate convergence. Unlike first-order methods without momentum, the Heavy Ball algorithm induces a second-order discrete-time dynamical system whose behavior is governed by the interaction between the step size, the momentum coefficient, and the curvature of the objective function. Even in the case of quadratic objectives, the resulting dynamics can exhibit oscillatory or underdamped regimes, as well as sensitivity to small perturbations in hyperparameter values. The convergence properties of the method are tightly linked to the spectral distribution of the Hessian matrix, leading to different convergence modes along distinct eigendirections and complicating the derivation of uniform convergence guarantees. Consequently, obtaining accurate convergence rate estimates for the Heavy Ball method requires careful analysis that goes beyond classical worst-case bounds and often remains analytically intractable in practice [5,6]. These characteristics make the Heavy Ball method a demanding yet informative testbed for data-driven convergence rate modeling, and a natural candidate for assessing the potential of machine learning-based surrogate models in the analysis of optimization dynamics.

The application of machine learning techniques to the analysis and enhancement of optimization algorithms has attracted increasing interest in recent years. A notable paradigm within this space is learning to optimize (L_2O), where machine learning models are trained to either design or improve optimization procedures based on empirical performance data. The comprehensive survey by Chen et al. describes this emerging paradigm and benchmarks multiple L_2O approaches, highlighting their ability to automate and adapt optimization strategies while also discussing limitations related to generalization beyond training distributions [7]. Unlike traditional theory-driven designs of optimization methods, which yield performance guarantees for specific classes of problems, data-driven optimization seeks to exploit patterns in empirical behavior to generate more efficient procedures.

In addition to L_2O, supervised machine learning models have been employed for performance prediction and algorithm selection. Trajanov et al. investigated explainable landscape analysis techniques for automated algorithm performance prediction, where the goal is to learn models that relate problem features to algorithm performance outcomes, thereby guiding the selection of the most suitable optimizer for new instances [8]. This line of research is conceptually aligned with the present work's objective of modeling convergence-related quantities, as both problems seek to approximate complex relationships between optimization settings and observed outcomes.

A broader context for surrogate modeling is found in engineering optimization, where surrogate-assisted optimization frameworks deploy machine learning regressors as approximations of expensive simulation models to accelerate design processes. In global antenna optimization, for

example, surrogate models such as kriging and variable-resolution approaches are used within iterative search schemes to reduce computation cost while preserving solution quality [9]. Reviews in this domain emphasize the trade-offs between surrogate fidelity and computational efficiency and underscore the importance of integrating surrogate predictions with optimization loops.

While surrogate modeling has a rich tradition in engineering contexts, general surrogate-assisted strategies have also been explored in automated history matching processes in reservoir engineering, where high-fidelity simulations are expensive and surrogate models reduce the costly runs required for optimization. These works review common surrogate model families including Gaussian processes, radial basis functions, and neural networks, illustrating the broad utility of data-driven approximations across different scientific domains.

Within the machine learning literature itself, ensemble methods such as random forests and boosting are widely studied for their predictive performance and generalization properties [10, 11]. However, classical ensemble models do not inherently capture structured relationships across samples or emphasize similarities driven by specific performance targets. This motivates the development of attention-augmented ensembles, such as the Self-Attention Forest framework utilized in this paper, which enriches tree-based models with data-dependent weighting mechanisms to better model complex regression targets. By grounding the proposed approach within this body of literature, the present work positions itself at the intersection of optimization analysis, performance prediction, and surrogate modeling, contributing a new perspective that combines these strands for learning convergence rate estimates.

1 Problem Statement and dataset. We consider the problem of estimating the convergence behavior of iterative optimization algorithms from observed optimization trajectories. Let $f:R^d \to R$ be a smooth objective function, and $\{x_k\}_{k\geq 0}$ denote the sequence of iterates generated by the Heavy Ball method with step size $\alpha > 0$ and momentum coefficient $\beta \in [0,1)$. Classical convergence analysis aims to characterize the rate at which $f(x_k)$ approaches its minimum value $f^\star$, typically by establishing bounds of the form $f(x_k)-f^\star \leq C\rho^k$ for some contraction factor $\rho \in (0,1)$. In this work, we adopt an empirical perspective and define a convergence rate estimate as a local, trajectory- dependent quantity that reflects the observed decay of the objective function along the optimization path rather than a global worst-case bound.

More specifically, given an initial point x_0 and the corresponding trajectory $\{x_k\}$, we define the target variable as a logarithmic estimate of the normalized objective value (1), computed as

$$\log_{10}\left(\frac{f(x_k)-f^\star}{f(x_0)-f^\star}\right) \tag{1}$$

at a given iteration index k. This quantity captures the effective contraction behavior of the algorithm and serves as a proxy for the convergence rate at different stages of the optimization process. The modeling task is then formulated as a supervised regression problem, where the goal is to predict this convergence-related quantity from a set of parameters characterizing the optimization dynamics.

The dataset used in this study is synthetically generated and constitutes a central component of the proposed methodology. Objective functions are chosen from a family of quadratic functions with heterogeneous curvature, parameterized by coefficients that control the conditioning of the problem. For each objective instance, Heavy Ball trajectories are simulated under randomly sampled optimization hyperparameters and normalized initial conditions. Each data sample corresponds to a randomly selected point along a stable optimization trajectory and is represented by a feature vector encoding the objective parameters, the Heavy Ball hyperparameters, the normalized initial state, and the relative iteration index. The associated target value reflects the observed logarithmic decay of the objective function at that point. This construction yields a diverse dataset that spans a wide range of dynamical regimes, including fast convergence, oscillatory behavior, and near-critical parameter settings. By explicitly modeling convergence-related quantities at the level of individual trajectory

samples, the dataset enables learning fine-grained relationships between optimization parameters and convergence behavior that are difficult to capture through analytical analysis alone.

The core object of study in this work is a supervised dataset that links a compact description of a Heavy Ball (HB) run to an empirical, trajectory-level proxy of convergence. We generate a synthetic regression dataset by repeatedly simulating HB dynamics on a two-dimensional quadratic objective of the form $f(x) = \frac{1}{2}(ax_1^2 + bx_2^2)$, where the curvature parameters $a$and $b$define the Hessian spectrum and, consequently, the conditioning of the problem. To ensure broad coverage of regimes, the pair $(a, b)$is drawn from four complementary sampling rules: highly ill-conditioned instances with small $a$and large b, isotropic instances with $a = b$, low-curvature instances with both coefficients near zero, and a wide-range regime where both coefficients vary up to large values. With 5000samples generated by the rule, the resulting dataset contains $N = 20000$labeled examples.

Each example corresponds to a single randomly chosen state along a stable HB trajectory and is represented by a seven-dimensional feature vector $(a, b, \alpha, \beta, x_{0,1}^{\text{norm}}, x_{0,2}^{\text{norm}}, k_{\text{norm}})$. The HB hyperparameters are sampled as $\beta \sim \mathcal{U}[0,0.999)$and $\alpha \sim \mathcal{U}(0, \alpha_{\max})$, where $\alpha_{\max} = \min\{0.95 \cdot 2(1+\beta)/L, 10\}$and $L = \max(a, b)$is the smoothness constant of the quadratic objective. This coupling between α, β, and $L$is a deliberate stability-oriented design choice: it concentrates sampling in parameter regions that are likely to produce meaningful trajectories while still allowing near-critical behavior. The initial point is generated by first sampling a radius $r \sim \mathcal{U}[10^{-3}, 20]$and then drawing $x_0$uniformly from the square $[-r, r]^2$; we store the normalized initialization $x_0^{\text{norm}} = x_0/r$, which lies in $[-1,1]^2$by construction and removes a trivial scaling degree of freedom. Trajectories are simulated for at most 300iterations, and early termination occurs when the gradient norm falls below 10^{-8}. To prevent numerical pathologies from contaminating the dataset, a run is rejected if the objective becomes non-finite or if $f(x_k)$exceeds $10^{12}\, f(x_0)$; when this happens, new (α, β)values are resampled up to 50attempts before drawing a new (a, b)pair. This rejection-resampling mechanism ensures that the final dataset consists of stable trajectories while still retaining challenging regimes close to instability.

The regression target is defined to reflect the observed objective decay along the trajectory in a scale-stable manner. Let $f_0 = \max\{f(x_0), 10^{-12}\}$denote a safeguarded initial objective value, and let $f_k = f(x_k)$be the objective at iteration k. We store $y = \log_{10}(f_k/f_0 + 10^{-30})$, which is typically non-positive and becomes more negative as the objective decreases. This target can be interpreted as a local, trajectory-dependent estimate of convergence progress at a particular stage of the run. In parallel, we record the normalized iteration index $k_{\text{norm}} = k/300$, which enables the learning model to represent time-varying behavior, including transient oscillations and phase transitions between fast initial decrease and slower asymptotic contraction. Because the quadratic Hessian is diagonal with eigenvalues $\{a, b\}$, the conditioning can be summarized by ratios such as b/a(when $a > 0$) or more robustly by $(\max(a, b) + \varepsilon)/(\min(a, b) + \varepsilon)$with a small ε, and our sampling rules explicitly produce both well-conditioned and ill-conditioned regimes. This diversity is important because momentum methods can exhibit qualitatively different convergence modes across eigendirections, and a dataset dominated by a single conditioning regime would lead to overly optimistic conclusions about generalization.

A key design decision is that each labeled example is obtained by sampling a random point along the trajectory rather than using only the final iterate or fitting a single global rate per run. This construction is intentional: Heavy Ball dynamics are not uniformly contracting at every iteration, and the observed decay may include oscillatory and nonmonotone segments even when the method eventually converges. By sampling along the trajectory, the dataset captures local regimes that are otherwise averaged out, allowing the model to learn how (α, β), curvature, and initialization jointly affect the instantaneous progress of optimization at different stages. Practically, this also increases the informativeness of the dataset because it exposes the learner to a wider range of convergence behaviors within a fixed simulation budget, enabling finer-grained prediction of convergence-related

quantities beyond what a single per-run summary statistic can provide. The appearance of the dataset is presented in table 1.

Table 1. Table captions should be placed above the tables

	a	b	alpha	beta	x0_1_norm	x0_2_norm	iter_norm	target_log10
0	3,75	385,21	0,01	0,73	-0,69	-0,88	0,29	-9,92
1	3,34	142,86	0,00	0,65	0,88	-1,00	0,92	-4,65
2	6,17	283,50	0,00	0,01	-0,20	-0,91	0,62	-3,13
3	3,66	236,82	0,00	0,78	0,18	-0,91	0,45	-10,63
4	1,71	119,52	0,03	0,95	-0,39	-0,80	0,30	-3,14
,,,	…	…	…	…	…	…	…	…
19995	458,77	311,31	0,00	0,17	-0,55	-0,58	0,08	-15,66
19996	295,70	116,68	0,00	0,93	0,09	0,06	0,18	-2,44
19997	334,71	120,41	0,00	0,36	0,15	0,31	0,05	-6,06
19998	89,13	181,54	0,00	0,59	0,53	0,27	0,06	-2,30
19999	161,28	113,63	0,00	0,89	-0,57	-0,41	0,34	-5,40

2. Model Training&Numerical Experiments. To model the relationship between optimization parameters and convergence behavior, we employ a tree-based ensemble method augmented with a self-attention mechanism. Random Forest–type models are particularly well suited for this task due to their ability to approximate highly nonlinear dependencies, robustness to feature scaling, and stability when trained on heterogeneous synthetic data. Unlike parametric models, ensemble trees do not impose restrictive assumptions on the functional form of the mapping between inputs and outputs, which is essential when modeling complex optimization dynamics driven by interacting hyperparameters and trajectory-dependent effects.

However, standard Random Forests treat all samples independently and lack an explicit mechanism for adaptively weighing feature interactions across the dataset. To address this limitation, we adopt the Self-Attention Forest framework proposed in [12], which extends classical tree ensembles by integrating an attention mechanism into the aggregation of individual estimators. In this framework, attention weights are computed based on similar relationships defined in the target space, allowing the model to dynamically reweigh tree contributions depending on the structure of the regression problem. In the context of convergence modeling, this mechanism enables the model to emphasize trajectory samples exhibiting similar objective decay patterns while reducing the influence of outliers arising from transient oscillations or near-critical parameter regimes. As a result, the architecture retains the interpretability and robustness of tree ensembles while incorporating a data-dependent aggregation strategy inspired by attention-based models.

The model is trained in a supervised regression setting using the synthetically generated dataset described in the previous section. Extra Trees are employed as base learners to increase variance reduction and improve generalization across diverse optimization regimes. Model hyperparameters, including the number of estimators, tree depth, and attention-related coefficients, are selected to balance expressive power and stability. Training is performed using a fixed random seed to ensure reproducibility, and all experiments are conducted without problem-specific tuning for individual objective functions.

To ensure that the reported results reflect genuine generalization rather than memorization of trajectory patterns, the dataset was randomly partitioned into training and validation subsets using an 80/20 split with a fixed random seed. The partitioning was performed after full dataset construction and shuffling, so that samples originating from different objective parameter regimes and trajectory segments were evenly distributed across both subsets. The model was trained exclusively on the training portion, while all quantitative and qualitative evaluations were conducted on the held-out

validation set. No resampling or reweighting strategies were applied, and the validation data were not used for hyperparameter tuning. This protocol ensures that the predictive performance reflects the model's ability to infer convergence behavior for previously unseen combinations of curvature parameters, hyperparameters, and trajectory stages.

Model performance is evaluated using the coefficient of determination R^2, which measures the agreement between predicted and empirically observed convergence-related quantities. In addition to global regression accuracy, qualitative agreement between predicted and true optimization trajectories is assessed by reconstructing objective decay curves from model predictions and comparing them to actual Heavy Ball trajectories. Numerical experiments are conducted on several benchmark objective functions, including the Rastrigin, Booth, and Himmelblau functions, which exhibit distinct landscape properties and convergence behaviors. Across all tested functions, the proposed model accurately captures the overall decay trend of the objective value, closely matching the empirical convergence profiles despite the presence of oscillations and nonmonotonic behavior in the true trajectories.

A complementary view of predictive performance is provided by the parity plot shown in Figure 4, where predicted values of the logarithmic decay measure are plotted against their empirical counterparts on the validation set. The concentration of points along the diagonal indicates a strong correspondence between predicted and observed convergence behavior across the full dynamic range of the target variable. Importantly, dispersion remains controlled even in the regime of large negative values, which correspond to late-stage convergence where objective values approach numerical precision limits. The absence of systematic bias in the central region and the preservation of monotonic structure across scales confirm that the model captures the dominant contraction patterns of the Heavy Ball dynamics rather than overfitting isolated trajectory segments.

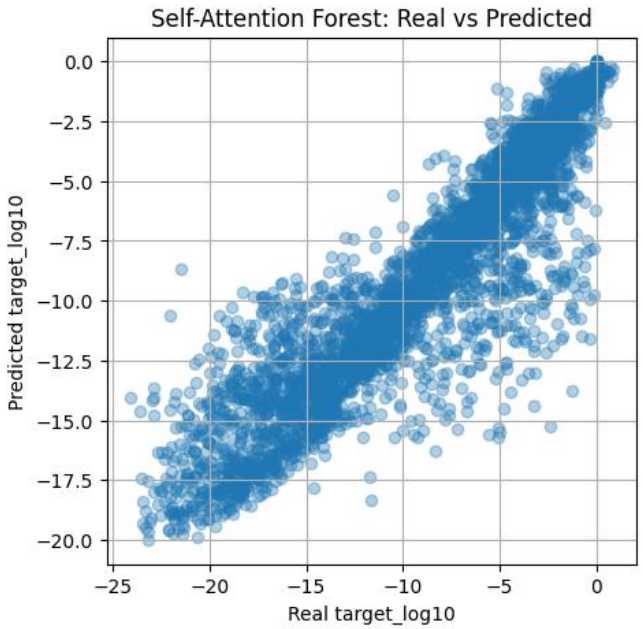

Fig. 1. Parity plot comparing the empirical logarithmic convergence measure with the corresponding predictions of the self-attention–augmented Random Forest on the validation set

The high R^2 values observed in these experiments indicate that the model successfully learns the underlying relationship between optimization parameters and convergence dynamics. These results demonstrate that self-attention–augmented tree ensembles provide an effective and flexible tool for modeling convergence rate estimates and can generalize across objective functions with substantially different structural properties. The results are presented on the figures 2-4.

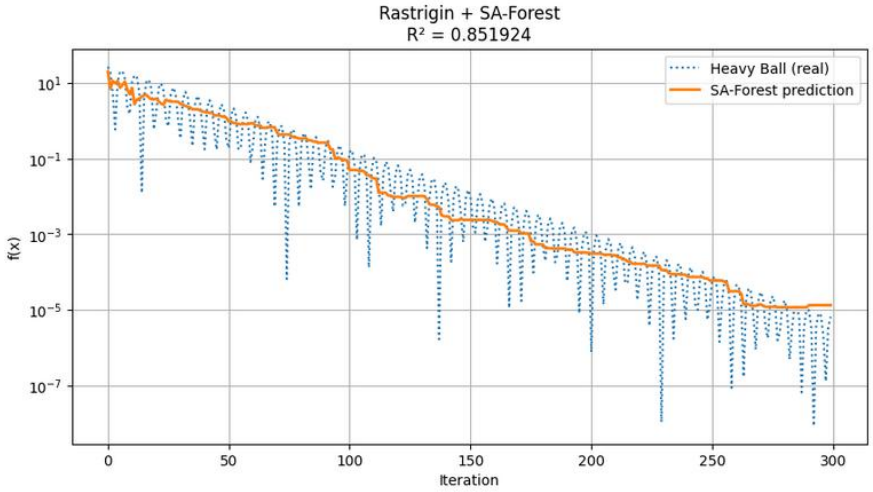

Fig. 2. Comparison between the true Heavy-Ball convergence trajectory and the trajectory predicted by the self-attention–augmented Random Forest for the Rastrigin function

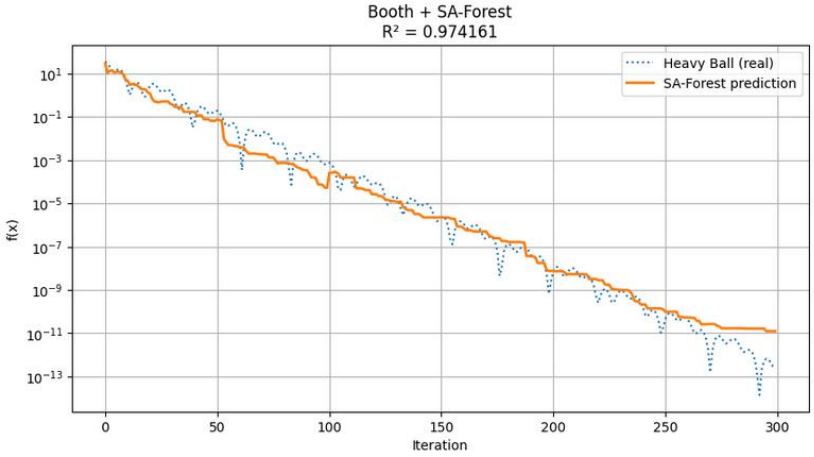

Fig. 3. Comparison of the true Heavy Ball convergence trajectory and the convergence profile predicted by the self-attention–augmented Random Forest for the Booth function

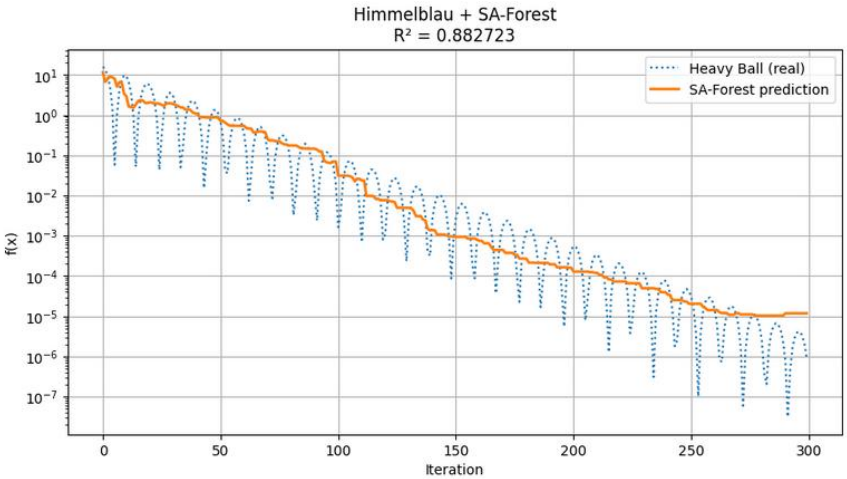

Fig. 4. Comparison of the true Heavy Ball convergence trajectory and the convergence profile predicted by the self-attention–augmented Random Forest for the Himmelblau function

3. Conclusion. This paper proposed a data-driven approach to modeling convergence rate estimates of iterative optimization methods, motivated by the fact that classical convergence analysis is often either overly conservative or impractical when applied to modern training pipelines. Instead of relying on closed-form bounds derived under restrictive assumptions, we treated the optimization algorithm as a dynamical system and learned a surrogate mapping from a compact description of the optimization setting to convergence-related quantities measured along trajectories. The Heavy Ball method was used as a technically demanding case study, since its inertial term induces oscillatory and regime-dependent behavior, and its performance is strongly influenced by the interaction between the step size, the momentum coefficient, and curvature properties of the objective.

A central contribution of the work is the construction of a synthetic dataset that explicitly targets convergence modeling. By simulating trajectories under diverse objective parameters, hyperparameters, and initial conditions, and by sampling states along stable runs, the dataset captures a broad spectrum of dynamical regimes, including fast contraction, transient oscillations, and near-critical configurations. This design allows the learning problem to focus on predicting empirically observed decay behavior rather than reproducing worst-case theoretical rates. Using this dataset, we trained a self-attention–augmented tree ensemble and demonstrated that the model can accurately predict convergence-related quantities and reconstruct objective value decay profiles across multiple benchmark functions. The obtained results indicate that the surrogate model captures the dominant convergence trends even when the underlying trajectory is nonmonotone, which is a typical feature of momentum-based optimization.

Beyond empirical accuracy, the proposed framework has methodological implications. First, it provides a practical alternative to repeated trial-and-error experimentation when selecting hyperparameters, since the surrogate can be queried to anticipate the likely convergence behavior for a given configuration. Second, it creates a pathway toward automated convergence analysis, where learned models can be used to compare optimizers, detect unstable regimes, and prioritize hyperparameter regions that yield favorable convergence profiles. Third, although the present study focuses on Heavy Ball dynamics, the same dataset-driven design can be extended to other optimization schemes. Momentum-accelerated methods such as Nesterov's acceleration exhibit similar second-order behavior and sensitivity to curvature, making them natural candidates for the same modeling pipeline. Adaptive methods such as Adam and RMSProp introduce additional state variables through moment estimates and coordinate-wise scaling; however, they can still be viewed within the same black-box dynamical perspective by expanding the feature representation to include algorithmic state summaries or statistics of gradient history. In this sense, the proposed methodology is not tied to a particular optimizer, but rather to the general principle of learning convergence behavior from trajectory data.

Finally, several directions remain for future work. Extending the approach from synthetic objectives to real training losses would require addressing stochastic gradients, minibatch noise, and nonstationarity, as well as defining convergence proxies that remain stable under stochastic dynamics. Another important direction is improving interpretability: while attention-enhanced tree ensembles offer some transparency, a deeper analysis of feature sensitivities could yield actionable insights into how curvature, momentum, and initialization jointly shape convergence regimes. Overall, the results of this study support the view that machine learning surrogates can substantially reduce the cost of convergence rate estimation, provide flexible tools for optimizer analysis, and serve as building blocks for automated systems that predict and improve optimization behavior in large-scale machine learning.

Author contributions D.M. (Daniil Miroshnichenko) conceptualized the study, developed the methodology, implemented the random forest with self-attention model, conducted the experiments, analyzed the results, prepared the figures, and wrote the original draft of the manuscript.M.A. (Majid Abbasov) supervised the research, contributed to the theoretical analysis and interpretation of the results, reviewed and edited the manuscript, validated the methodology, and approved the final version of the manuscript.All authors have read and agreed to the published version of the manuscript.
Funding. The authors received no financial support from any funding agency, commercial organization, or not-for-profit institution for the research, authorship, and/or publication of this article.
Data Availability. No datasets were generated or analyzed during the current study.
Declarations
Conflict of Interest. The authors declare that they have no competing interests or conflicts of interest.
Ethical Approval. This article does not contain any studies involving human participants or animals performed by any of the authors.
Informed Consent. Not applicable.

References

[1] Polyak, B. T. (1964). Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics, 4*(5), 1–17. https://doi.org/10.1016/0041-5553(64)90137-5
[2] Nesterov, Y. (2004). *Introductory lectures on convex optimization: A basic course*. Springer. https://doi.org/10.1007/978-1-4419-8853-9
[3] Bottou, L., Curtis, F. E., & Nocedal, J. (2018). Optimization methods for large-scale machine learning. *SIAM Review*, *60*(2), 223–311. https://doi.org/10.1137/16M1080173
[4] Andrychowicz, M., Denil, M., Gómez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., & de Freitas, N. (2016). Learning to learn by gradient descent by gradient descent. In *Advances in Neural Information Processing Systems* (Vol. 29).
[5] Lessard, L., Recht, B., & Packard, A. (2016). Analysis and design of optimization algorithms via integral quadratic constraints. *SIAM Journal on Optimization, 26*(1), 57–95. https://doi.org/10.1137/15M1009597
[6] Su, W., Boyd, S., & Candès, E. J. (2016). A differential equation for modeling Nesterov's accelerated gradient method. *Journal of Machine Learning Research, 17*(153), 1–43.
[7] Chen, T., Chen, X., Chen, W., & Wang, Z. (2022). Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research, 23*, 1–59.
[8] Trajanov, R., Dimeski, S., Popovski, M., Korošec, P., & Eftimov, T. (2022). *Explainable landscape analysis in automated algorithm performance prediction*. arXiv:2203.11828.
[9] Pietrenko-Dąbrowska, A., et al. (2024). Variable resolution machine learning for global antenna optimization using sensitivity-analysis-based dimensionality reduction. *Scientific Reports, 14*, 28796. https://doi.org/10.1038/s41598-024-77367-w
[10] Zhao, Y., et al. (2024). A review on optimization algorithms and surrogate models. *Geoenergy Science and Engineering, 234*, 212554. https://doi.org/10.1016/j.geoen.2023.212554
[11] Rane, N. L., Choudhary, S., & Rane, J. (2024). Techniques and optimization algorithms in machine learning: A review. In *Machine Learning and Artificial Intelligence* (pp. 27–48). Deep Science Publishing. https://doi.org/10.70593/978-81-981271-4-3_2
[12] Utkin, L. V., & Konstantinov, A. V. (2022). *Self-Attention Forests*. arXiv:2207.04293.