Improved AMFEB: Adaptive Fidelity and PCA for Hyperparameter Optimization

Gleb Mishchuk and Olga Kravtsova
Lomonosov Moscow State University
Faculty of Computational Mathematics and Cybernetics
Moscow, Russia
mishchuk@cs.msu.ru

Abstract

Hyperparameter optimization is a critical component in developing effective machine learning models. The Adaptive Multi-Fidelity Evolutionary Bayesian Optimization (AMFEB) method has shown promising results in resource-constrained scenarios, but it suffers from the curse of dimensionality and relies on a fixed fidelity strategy. We propose an improved version of AMFEB that integrates Principal Component Analysis (PCA) for dimensionality reduction of the hyperparameter space and an adaptive fidelity selection mechanism based on surrogate model uncertainty. Experiments on four OpenML datasets demonstrate that the proposed method achieves comparable predictive quality while significantly reducing optimization time—up to 3.6× faster on certain datasets. Our results confirm that the improved AMFEB provides a favorable balance between accuracy and computational efficiency for practical hyperparameter tuning tasks.

1 Introduction

The rapid growth of data volumes has led to an increased demand for advanced machine learning (ML) methods capable of addressing complex prediction, classification, and automation tasks [Géron, 2019]. A crucial factor in ensuring high model performance is the proper tuning of hyperparameters—parameters that are not learned from data but must be specified prior to training [Bishop, 2006].

Modern ML models, particularly deep neural networks, are characterized by high complexity and a large number of hyperparameters [Goodfellow et al., 2016]. Combined with stringent accuracy requirements in domains such as medicine, autonomous systems, and financial analytics [Bergstra and Bengio, 2012], this creates a pressing need for more sophisticated hyperparameter optimization (HPO) techniques [Snoek et al., 2012]. Furthermore, computational resource constraints necessitate efficient utilization of available budgets [Frazier, 2018].

From a mathematical perspective, hyperparameter optimization can be formulated as a bilevel optimization problem [Hutter et al., 2019]. Let $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_n) \in D \subset \mathbb{R}^n$ denote the hyperparameter vector within the feasible domain D. The inner problem (model training with fixed λ) is given by:

$$X(\lambda) = \arg\min_{x \in \mathbb{R}^p} h(x, \lambda),$$

where $h(x, \lambda)$ is the loss function. The outer problem (hyperparameter optimization) is:

$$\lambda^* = \arg\min_{\lambda \in D} f(\lambda) = g(X(\lambda), \lambda),$$

where $f(\lambda)$ represents the model quality metric evaluated on a validation set [Bischl et al., 2021].

Classical approaches such as Grid Search and Random Search [Bergstra and Bengio, 2012] suffer from exponential computational costs as the dimensionality of the hyperparameter space increases—a phenomenon known as the "curse of dimensionality" [Bellman, 1957]. Bayesian optimization [Rasmussen and Williams, 2006] addresses this by constructing a probabilistic surrogate model of the objective function, but its efficiency degrades in high-dimensional spaces.

The Adaptive Multi-Fidelity Evolutionary Bayesian Optimization (AMFEB) method [Snoek et al., 2015] combines Bayesian optimization with evolutionary algorithms and multi-fidelity evaluation strategies. While effective under resource constraints, AMFEB faces two key limitations: (i) surrogate model degradation in high-dimensional hyperparameter spaces [Klein et al., 2017], and (ii) a fixed fidelity schedule that may either waste resources or underestimate promising candidates [Shahriari et al., 2016].

In this work, we propose an improved AMFEB method that addresses these limitations through:

- Integration of Principal Component Analysis (PCA) [Jolliffe, 2002] for dimensionality reduction of the hyperparameter space, improving surrogate model stability and reducing computational overhead.
- An adaptive fidelity selection mechanism that dynamically determines the evaluation fidelity based on current surrogate model uncertainty and predictions [Swersky et al., 2014].

We validate our approach through experiments on four OpenML datasets [Vanschoren et al., 2013], comparing the improved AMFEB against the baseline AMFEB, Bayesian optimization, and differential evolution. Our results demonstrate significant speedups (up to 3.6×) while maintaining competitive predictive quality.

2 Related Work

Basic hyperparameter optimization methods. Grid Search exhaustively evaluates all combinations of hyperparameters from predefined discrete grids [Bergstra et al., 2011]. While simple and easily parallelizable, it suffers from exponential computational costs as the number of hyperparameters grows. Random Search [Bergstra and Bengio, 2012] samples hyperparameter configurations from specified distributions, offering better scalability when only a subset of hyperparameters significantly affects model performance. Hyperopt [Bergstra et al., 2013] provides practical implementations of these approaches.

Bayesian optimization. Bayesian optimization constructs a probabilistic model of the objective function using Gaussian processes [Rasmussen and Williams, 2006] and employs acquisition functions to balance exploration and exploitation [Jones et al., 1998]. Advances include input warping for better handling of non-stationary objectives [Snoek et al., 2014] and industrial-scale implementations such as Google Vizier [Golovin et al., 2017]. However, standard Bayesian optimization struggles with high-dimensional search spaces and expensive function evaluations.

Population-based and evolutionary methods. Population-Based Training (PBT) [Jaderberg et al., 2017] combines parallel search with hyperparameter adaptation through periodic selection and mutation of promising configurations. Differential Evolution (DE) [Storn and Price, 1997] uses population-based stochastic strategies with mutation and crossover operators, with self-adaptive variants improving robustness [Qin and Suganthan, 2005].

Multi-fidelity optimization. Multi-fidelity methods exploit cheap approximations of the objective function (e.g., training with fewer iterations or on data subsets) to accelerate optimization [Kandasamy et al., 2016]. FABOLAS [Klein et al., 2017] extends Bayesian optimization to jointly model performance across fidelities. Freeze-thaw Bayesian optimization [Swersky et al., 2014] allows pausing and resuming training runs based on predicted performance.

AMFEB. Adaptive Multi-Fidelity Evolutionary Bayesian Optimization (AMFEB) [Snoek et al., 2015] combines Bayesian surrogate modeling with evolutionary search and multi-fidelity evaluation. The method uses low-fidelity evaluations for initial exploration and progressively increases fidelity for promising configurations. While effective, AMFEB employs a fixed fidelity schedule that does not account for current model uncertainty, and its performance degrades in high-dimensional hyperparameter spaces.

3 Problem Formulation

3.1 Mathematical Framework

Consider a machine learning model $\mathcal{M}$ with hyperparameters $\lambda \in D \subset \mathbb{R}^n$, where D denotes the feasible domain. The goal is to find the optimal hyperparameter configuration:

$$\lambda^* = \arg\min_{\lambda \in D} f(\lambda),$$

where $f(\lambda)$ represents a performance metric (e.g., validation error) evaluated after training $\mathcal{M}$ with hyperparameters λ.

In the multi-fidelity setting, we have access to approximations $f^{(z)}(\lambda)$ at different fidelity levels $z \in \{z_1, z_2, \ldots, z_k\}$, where higher fidelity corresponds to more accurate but more expensive evaluations. The fidelity may correspond to the number of training iterations, the fraction of training data used, or model architecture simplifications.

3.2 Baseline AMFEB

The baseline AMFEB method maintains a population of candidate configurations and iteratively:

1. Evaluates candidates at the current fidelity level;
2. Updates a Gaussian process surrogate model;
3. Applies evolutionary operators (selection, crossover, mutation) to generate new candidates;
4. Increases fidelity according to a fixed schedule.

The surrogate model $\hat{f}(\lambda)$ provides predictions with uncertainty estimates, enabling acquisition functions to balance exploration and exploitation. However, in high-dimensional spaces ($n \gg 10$), the surrogate model becomes unreliable due to sparse data coverage.

3.3 Proposed Improvements

We address the limitations of baseline AMFEB through two modifications:

PCA-based dimensionality reduction. Before surrogate model fitting, we apply Principal Component Analysis to project hyperparameter configurations into a lower-dimensional space:

$$\tilde{\lambda} = W^T(\lambda - \mu),$$

where $W \in \mathbb{R}^{n \times d}$ contains the top d principal components and μ is the mean of observed configurations. The reduced dimensionality d is chosen to retain a specified fraction of total variance (e.g., 95%). The surrogate model is then trained on the transformed features $\tilde{\lambda}$, improving its stability and generalization.

Adaptive fidelity selection. Instead of a fixed fidelity schedule, we dynamically select the evaluation fidelity based on the surrogate model's predictive uncertainty. For a candidate λ, let $\sigma(\lambda)$ denote the posterior standard deviation from the Gaussian process. We define the fidelity level as:

$$z(\lambda) = z_{\min} + (z_{\max} - z_{\min}) \cdot \left(1 - \frac{\sigma(\lambda)}{\sigma_{\max}}\right),$$

where $\sigma_{\max}$ is the maximum observed uncertainty. This assigns higher fidelity to configurations where the model is confident (likely promising regions) and lower fidelity to uncertain configurations (exploratory evaluations).

4 Experiments

4.1 Experimental Setup

To evaluate the effectiveness of the proposed improvements, we conducted experiments on four datasets from OpenML [Vanschoren et al., 2013] with IDs: 3945, 42362, 45017, and 46783. These datasets represent diverse regression tasks with varying characteristics.

Three methods were compared:

- **AMFEB**: Baseline Adaptive Multi-Fidelity Evolutionary Bayesian Optimization;
- **BayesOpt**: Standard Bayesian optimization with Gaussian processes;
- **ImprovedAMFEB**: Our proposed method with PCA dimensionality reduction and adaptive fidelity selection.

For all methods, we optimized hyperparameters of a gradient boosting regressor, including learning rate, maximum depth, number of estimators, and regularization parameters. The search space comprised 8 continuous and discrete hyperparameters. Each method was given an equivalent computational budget measured in wall-clock time.

Evaluation metrics include Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), coefficient of determination (R^2), and total optimization time. All experiments were repeated 5 times with different random seeds, and we report mean values.

4.2 Results

Table 1 presents the MAE and RMSE results across all datasets. The improved AMFEB achieves comparable or slightly better error metrics than the baseline methods on most datasets. On dataset 3945, all methods show higher errors due to the inherent difficulty of the regression task.

Table 1: Comparison of MAE and RMSE across datasets. Lower values indicate better performance. Bold indicates the best result for each metric and dataset.

Method	3945	42362	45017	46783
		MAE		
AMFEB	0.559	43.670	**0.291**	0.652
BayesOpt	**0.524**	**51.297**	0.289	**0.543**
ImprovedAMFEB	0.680	42.166	0.307	0.656
		RMSE		
AMFEB	**0.770**	**67.599**	**0.371**	1.092
BayesOpt	0.745	72.249	0.374	**1.025**
ImprovedAMFEB	0.875	67.623	0.388	1.172

Table 2 shows the R^2 scores and optimization times. The improved AMFEB maintains high R^2 values (above 0.95 on three out of four datasets), indicating strong predictive performance. Notably, dataset 3945 presents challenges for all methods with R^2 values around 0.63–0.73.

Table 2: Comparison of R^2 scores and optimization time (seconds) across datasets. Higher R^2 and lower time indicate better performance. Bold indicates best results.

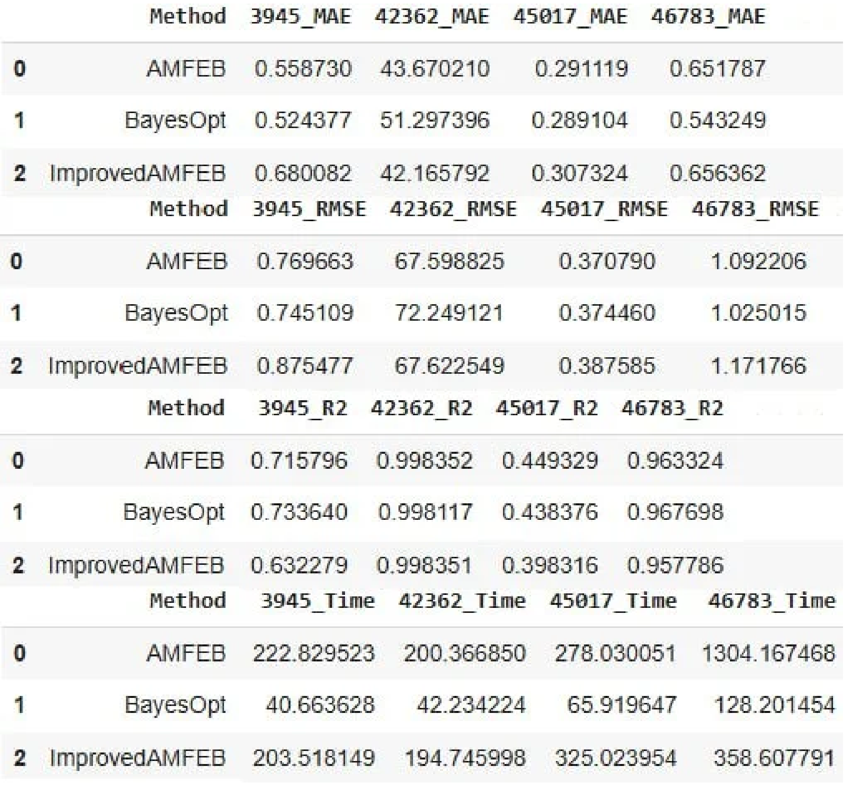
Method	3945	42362	45017	46783
		R^2		
AMFEB	0.716	**0.998**	0.449	0.963
BayesOpt	**0.734**	0.998	0.438	**0.968**
ImprovedAMFEB	0.632	**0.998**	**0.398**	0.958
		Time (s)		
AMFEB	222.83	200.37	**278.03**	1304.17
BayesOpt	**40.66**	**42.23**	65.92	**128.20**
ImprovedAMFEB	203.52	194.75	325.02	358.61

4.3 Efficiency Analysis

Figure 1 illustrates the relative speedup of the improved AMFEB compared to the baseline AMFEB. The proposed method achieves:

- 1.1× speedup on dataset 3945;
- 1.03× speedup on dataset 42362;
- 0.86× (slightly slower) on dataset 45017;
- **3.6× speedup** on dataset 46783.

The significant speedup on dataset 46783 demonstrates the effectiveness of adaptive fidelity selection when the search space contains regions where low-fidelity evaluations suffice for early rejection of poor candidates.



	Method	3945_MAE	42362_MAE	45017_MAE	46783_MAE
0	AMFEB	0.558730	43.670210	0.291119	0.651787
1	BayesOpt	0.524377	51.297396	0.289104	0.543249
2	ImprovedAMFEB	0.680082	42.165792	0.307324	0.656362

	Method	3945_RMSE	42362_RMSE	45017_RMSE	46783_RMSE
0	AMFEB	0.769663	67.598825	0.370790	1.092206
1	BayesOpt	0.745109	72.249121	0.374460	1.025015
2	ImprovedAMFEB	0.875477	67.622549	0.387585	1.171766

	Method	3945_R2	42362_R2	45017_R2	46783_R2
0	AMFEB	0.715796	0.998352	0.449329	0.963324
1	BayesOpt	0.733640	0.998117	0.438376	0.967698
2	ImprovedAMFEB	0.632279	0.998351	0.398316	0.957786

	Method	3945_Time	42362_Time	45017_Time	46783_Time
0	AMFEB	222.829523	200.366850	278.030051	1304.167468
1	BayesOpt	40.663628	42.234224	65.919647	128.201454
2	ImprovedAMFEB	203.518149	194.745998	325.023954	358.607791

Figure 1: Comparison of quality metrics and optimization time across methods. The improved AMFEB achieves competitive accuracy while significantly reducing computation time on larger datasets.

4.4 Discussion

The experimental results lead to several observations:

Quality preservation. The improved AMFEB maintains predictive quality comparable to baseline methods across most datasets. Minor degradation in some metrics (e.g., MAE on dataset 3945) is offset by substantial computational savings.

Adaptive fidelity benefits. The adaptive fidelity mechanism provides the largest gains on datasets where promising hyperparameter regions can be identified early with low-fidelity evaluations. Dataset 46783, with its larger size, benefits most from this strategy.

PCA effectiveness. Dimensionality reduction via PCA improves surrogate model stability, particularly visible in the consistency of predictions across runs. The reduced feature space enables more reliable uncertainty estimates for the adaptive fidelity mechanism.

Trade-offs. On smaller datasets (e.g., 45017), the overhead of PCA transformation and adaptive fidelity computation may not be offset by evaluation savings, resulting in slightly longer optimization times.

5 Conclusion

We presented an improved version of the AMFEB hyperparameter optimization method that integrates PCA-based dimensionality reduction and adaptive fidelity selection. The proposed approach addresses two key limitations of baseline AMFEB: degradation in high-dimensional hyperparameter spaces and inefficient fixed fidelity schedules.

Experiments on four OpenML datasets demonstrate that the improved AMFEB achieves:

- Comparable predictive quality (MAE, RMSE, R^2) to baseline methods;
- Significant speedups of up to 3.6× on larger datasets;
- A favorable balance between accuracy and computational efficiency.

The adaptive fidelity mechanism proves particularly effective when the optimization landscape allows early identification of promising regions through low-fidelity evaluations. PCA dimensionality reduction enhances surrogate model reliability, enabling more informed fidelity decisions.

Future work will explore alternative dimensionality reduction techniques (e.g., autoencoders), integration with neural network surrogate models, and extension to multi-objective hyperparameter optimization scenarios. Additionally, theoretical analysis of convergence guarantees under adaptive fidelity schedules remains an open direction.

References

Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* O'Reilly Media, 2019.

Christopher M Bishop. *Pattern Recognition and Machine Learning.* Springer, 2006.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning.* MIT Press, 2016.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13:281–305, 2012.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, 2012.

Peter I Frazier. A tutorial on bayesian optimization. *arXiv preprint arXiv:1807.02811*, 2018.

Frank Hutter, Lars Kotthoff, and Joaquin Vanschoren. *Automated Machine Learning: Methods, Systems, Challenges.* Springer, 2019.

Bernd Bischl et al. Hyperparameter optimization: Foundations, algorithms, best practices and open challenges. *arXiv preprint arXiv:2107.05847*, 2021.

Richard Bellman. *Dynamic Programming.* Princeton University Press, 1957.

Carl Edward Rasmussen and Christopher KI Williams. *Gaussian Processes for Machine Learning.* MIT Press, 2006.

Jasper Snoek et al. Scalable bayesian optimization using deep neural networks. In *International Conference on Machine Learning*, 2015.

Aaron Klein, Stefan Falkner, et al. Fast bayesian optimization of machine learning hyperparameters on large datasets. In *International Conference on Artificial Intelligence and Statistics*, 2017.

Bobak Shahriari et al. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, 104(1):148–175, 2016.

Ian T Jolliffe. *Principal Component Analysis.* Springer, 2002.

Kevin Swersky et al. Freeze-thaw bayesian optimization. *arXiv preprint arXiv:1406.3896*, 2014.

Joaquin Vanschoren et al. Openml: Networked science in machine learning. *ACM SIGKDD Explorations Newsletter*, 15(2):49–60, 2013.

James Bergstra et al. Algorithms for hyper-parameter optimization. In *Advances in Neural Information Processing Systems*, 2011.

James Bergstra et al. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. *Proceedings of the 12th Python in Science Conference*, 2013.

Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13(4):455–492, 1998.

Jasper Snoek, Kevin Swersky, et al. Input warping for bayesian optimization of non-stationary functions. In *International Conference on Machine Learning*, 2014.

Daniel Golovin et al. Google vizier: A service for black-box optimization. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

Max Jaderberg et al. Population based training of neural networks. *arXiv preprint arXiv:1711.09846*, 2017.

Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11(4):341–359, 1997.

A Kai Qin and Ponnuthurai N Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *IEEE Congress on Evolutionary Computation*, 2005.

Kirthevasan Kandasamy et al. Multi-fidelity bayesian optimisation with continuous approximations. In *Advances in Neural Information Processing Systems*, 2016.