UNIVERSIDAD PONTIFICIA COMILLAS
Higher Technical School of Engineering ICAI

MATHEMATICAL ENGINEERING AND ARTIFICIAL INTELLIGENCE

Final Degree Project

Default Configuration in Bayesian Optimization Priors

Author
Nicolás Villagrán Prieto - 202100607

Director
Eduardo César Garrido Merchán

Co-director
Simón Rodríguez Santana

Madrid, June 2025

I hereby declare, under my own responsibility, that the Project presented with the title

Default Configuration in Bayesian Optimization Priors

at the High Technical School of Engineering - ICAI of Universidad Pontificia Comillas in the academic year **2024/25** is my own work, original and unpublished, and has not been previously submitted for any other purpose.

The Project is not a copy of someone else's work, neither totally nor partially, and any information taken from other documents has been properly referenced.

Signed:  Date: 17/06/2025

Project submission authorized
Eduardo César Garrido Merchán

Signed:  Date: 17/06/2025

Agradecimientos

Quisiera comenzar expresando mi más sincero y profundo agradecimiento a mis padres, Marta Prieto y Bosco Villagrán, que no representan sino la quintaesencia de la abnegación y el sacrificio. Todo cuanto han hecho en su vida adulta ha tenido como propósito brindarnos –a mis hermanos y a mí– la pujanza y plenitud vital de la que gozamos, sin que aún hayamos hecho mérito suficiente.

Si hoy puedo cerrar con éxito mi etapa universitaria, es en gran parte gracias a la inmensa fortuna de haber nacido en un entorno tan privilegiado. Soy plenamente consciente de que nada de ello depende de mí, y de que millones de personas, con mayor talento y esfuerzo, no han contado con semejante suerte. Por ello, siento una responsabilidad moral ineludible, no solo hacia quienes me han acompañado, sino también hacia el mundo. Aspiro a seguir formándome y creciendo con el propósito de disponer de más herramientas, más capacidad de influencia y de acción. Hoy expreso mi gratitud con palabras, pero mi propósito vital es hacerlo mediante obras: devolver con creces todo lo recibido y contribuir a que otros puedan desarrollarse en un entorno tan enriquecedor como el mío.

En este empeño ocupa un lugar diferencial ICAI, mi segunda casa —y en muchas ocasiones, la primera— durante los últimos cuatro años. A esta institución le debo haber sido la primera que me obligó a pensar de manera pausada, profunda y rigurosa, y que me retó constantemente a superarme. Uno espera adquirir conocimientos técnicos en una universidad de tan merecido prestigio, pero descubre que aún más valiosas son las virtudes humanas que en ella se cultivan: la disciplina, la humildad intelectual, la resiliencia y el compromiso con los demás. Entré siendo un niño; hoy me reconozco como un hombre. Gracias de corazón, ICAI, por exigirme, por humanizarme y por enseñarme a pensar.

La universidad la forman sus profesores. Gracias a Manuel Villanueva Pesqueira por ser un faro en mis primeros años, y con frases como “mejor poco pero bien, que mucho y mal” marcar para siempre mi forma de estudiar. David Alfaya, Jaime Boal y Simón Rodríguez me enseñaron que la pasión por el trabajo bien hecho no solo es posible, sino profundamente inspiradora. En su forma de enseñar, en su entrega diaria, vi reflejado un amor sincero por lo que hacen. Con ellos aprendí que trabajar con entusiasmo, rigor y compromiso no es una excepción, sino un ideal alcanzable. Gracias a su ejemplo, nació en mí un interés por la ciencia, pero sobre todo, una forma distinta de entender el trabajo: la vocación es necesaria. Agradezco también de forma especial a Eduardo César Garrido, mi director de Trabajo de Fin de Grado, por su guía cercana y paciente durante estos últimos meses y por haberme introducido al apasionante universo bayesiano.

Superar los momentos difíciles no hubiera sido posible sin la figura de mi abuelo Francisco Prieto. Convivió con la enfermedad con una entereza admirable y supo hacer de su fragilidad una expresión de amor profundo hacia su familia. Su frase, “vamos pa’lante”, resuena en mí como una herencia moral que me impulsa a seguir, a no rendirme jamás. Este trabajo también es suyo.

A mis amigos de Sevilla, gracias por estar siempre presentes, incluso en la distancia. De manera especial, doy las gracias a Álvaro Morales y Felipe Loring, por ser un apoyo incondicional durante estos últimos cuatro años. A los amigos que la vida me regaló en Madrid y en San Diego, gracias por convertiros en familia y por hacer que estos fueran

mucho más que lugares de paso. En particular le doy las gracias a Mario Kroll, fiel amigo y compañero de innumerables proyectos y retos, gracias por tu paciencia y por nunca tirar la toalla en aquellas largas jornadas en que todo parecía resistirse.

Deseo dedicar también unas palabras a mis compañeros de la primera promoción del grado en Ingeniería Matemática e Inteligencia Artificial (IMAT), con quienes he compartido alegrías, frustraciones y logros. Ser parte de un proyecto pionero nos unió de forma especial, y siempre llevaré conmigo el orgullo de haber construido juntos sus primeros pasos. Asimismo, expreso mi más sincero agradecimiento a David Contreras, jefe de estudios del grado, cuya dedicación incansable, cercanía y esfuerzo constante han sido determinantes para que esta primera promoción alcanzara sus metas.

Concluyo esta etapa con la certeza de que, más allá de cualquier aspiración profesional o intelectual, es la dimensión humana de la experiencia la que otorga verdadero sentido al recorrido. Decía Aristóteles que el ser humano es, por naturaleza, un animal social; y ninguna forma de inteligencia, por sofisticada que sea, podrá jamás desligar la grandeza de nuestras obras del entramado de relaciones que las hace posibles. Lo esencial en toda hazaña humana no reside en el individuo aislado, sino en el vínculo que lo une a los otros. En mi caso, espero poder volver a agradecer en el futuro, con la misma energía con la que lo hago hoy, las relaciones que he fraguado en el camino. Esto no ha hecho más que empezar.

Resumen

La Optimización Bayesiana (BO) se ha convertido en un estándar para el ajuste de hiperparámetros de los modelos de Aprendizaje Automático debido a su eficiencia y eficacia para optimizar funciones caja negra costosas de evaluar. Si bien el enfoque habitual comienza con distribuciones a priori no informadas o con un muestreo uniforme sobre el espacio de hiperparámetros como hipótesis de la localización del óptimo, este estudio investiga si los hiperparámetros por defecto incluidos en las librerías populares de ML pueden actuar como distribuciones a priori informativas. Estos valores por defecto suelen considerarse puntos diseñados por expertos con un desempeño razonable, pero su papel en el proceso de optimización no ha sido examinado con rigor.

A través de una extensa batería de experimentos en diversos modelos de aprendizaje automático, librerías de Optimización Bayesiana y conjuntos de datos, respaldada por análisis cuantitativos rigurosos y pruebas estadísticas, nuestros resultados demuestran que los hiperparámetros por defecto no ofrecen una guía significativa hacia las regiones óptimas del espacio de búsqueda. No proporcionan ventajas sustanciales ni en el rendimiento final ni en la velocidad de convergencia frente al muestreo aleatorio de una distribución uniforme.

Concluimos que los expertos de aprendizaje automático no deberían confiar en los valores por defecto de las librerías cuando buscan un rendimiento óptimo. En su lugar, la Optimización Bayesiana ofrece un enfoque fundamentado y guiado que equilibra eficazmente la exploración y la explotación del espacio de hiperparámetros. Nuestros hallazgos evidencian el riesgo de asumir que los valores por defecto son suficientes y refuerzan la necesidad de estrategias de optimización robustas y conscientes de la incertidumbre. Los valores por defecto no son informativos; son convenientes, y la conveniencia no es un sustituto fiable del rendimiento. **Palabras clave:** Optimización Bayesiana, Ajuste de Hiperparámetros, Aprendizaje Automático, Distribuciones a priori informativas, Hiperparámetros por defecto.

1. Introducción

Este proyecto investiga el papel de los hiperparámetros por defecto en el contexto de la Optimización Bayesiana. El objetivo es evaluar si estos valores, comúnmente utilizados en la práctica, pueden aprovecharse para mejorar la convergencia y el rendimiento final al actuar como distribuciones a priori informativas dentro del proceso de BO. Para una introducción al tema, consúltese el Apéndice A.1.

2. Definición del Proyecto

Se ha formulado un proceso de Optimización Bayesiana que integra los valores por defecto de hiperparámetros provenientes de bibliotecas populares de ML como la media de distribuciones gaussianas truncadas sobre el espacio de hiperparámetros. Se comparó este enfoque con la inicialización aleatoria estándar en un conjunto diverso de modelos, librerías de optimización y datasets. Además, se estudió el efecto de utilizar un único punto de inicialización por defecto frente a un único punto muestreado de una distribución uniforme, en un número reducido de iteraciones del método.

Definimos los experimentos con muestreo como `Sample`, y la inicialización determinista

con hiperparámetros por defecto como `Default`.

3. Descripción del Sistema

El marco experimental emplea tres bibliotecas de Optimización Bayesiana (BoTorch, Optuna, Skopt) y tres tipos de modelos (Random Forest, MLP, SVM) sobre cinco datasets de aprendizaje supervisado para garantizar que la hipótesis se pruebe sobre una muestra representativa del amplio y diverso panorama del Aprendizaje Automático y la Optimización Bayesiana. Para asegurar robustez, los experimentos variaron el número de puntos de inicialización y el valor de la desviación estándar de la distribución normal truncada, analizando su efecto sobre la eficiencia de la optimización.

Para evaluar estadísticamente nuestra hipótesis, se utilizó una prueba binomial unilateral. Esta prueba determina si las estrategias de inicialización basadas en muestreo superan significativamente a la inicialización aleatoria más allá de lo que cabría esperar por azar bajo la hipótesis nula (H_0): los métodos de muestreo no ofrecen una mejora estadísticamente significativa respecto a la inicialización aleatoria. Se define una victoria como una instancia en la que el método basado en muestreo supera a la inicialización aleatoria por encima de un umbral específico del dominio, ya sea en términos de convergencia más rápida hacia una configuración óptima o de un mejor rendimiento predictivo final. Los valores p resultantes cuantifican si las mejoras observadas constituyen evidencia estadísticamente significativa para rechazar la hipótesis nula o si se deben probablemente al azar.

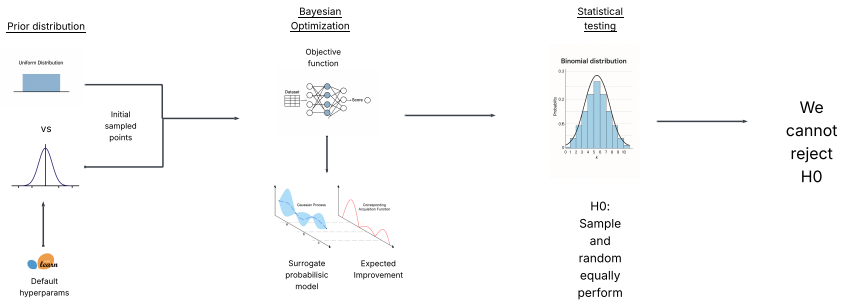

Figure 1: Resumen general de la metodología y descripción del pipeline.

4. Resultados

- Las configuraciones por defecto no ofrecieron mejoras estadísticamente significativas en la convergencia ni en el rendimiento final.
- En múltiples comparaciones entre estrategias, las pruebas binomiales arrojaron valores p muy por encima de 0.05, lo que indica que no hay ventajas consistentes.

Comparación	S/D	R	Empates	p-valor
S vs. R Convergencia	18	15	15	0.364
S vs. R Métrica	10	12	26	0.738
D vs. R Convergencia	19	12	17	0.141
D vs. R Métrica	11	17	20	0.908

Table 1: Resumen del número de victorias, empates y valores p de la prueba binomial unilateral para cada comparación. S se refiere a la estrategia Sample, R a Random, y D a Default. El método Sample emplea múltiples puntos de inicialización extraídos de una distribución gaussiana truncada sobre el espacio de hiperparámetros, mientras que el método Default se basa en una configuración predefinida única.

5. Conclusiones

Nuestros resultados muestran que, si bien los valores por defecto de hiperparámetros pueden resultar tentadores por su simplicidad y rendimiento inicial, no proporcionan una guía robusta para la optimización. En esencia, no pueden modelarse como distribuciones a priori informativas válidas para el ajuste de hiperparámetros mediante Optimización Bayesiana. De hecho, se recomienda favorecer estrategias de optimización conscientes de la incertidumbre como las que propone la Optimización Bayesiana estándar, en lugar de utilizar los valores por defecto de las bibliotecas de ML cuando el rendimiento sea un objetivo.

6. Referencias

Aunque todas las referencias se encuentran listadas en la sección correspondiente, para facilitar la comprensión temática, se proporciona la Tabla 12, que organiza las referencias por sección.

Abstract

Bayesian Optimization (BO) has become a standard for hyperparameter tuning in Machine Learning due to its efficiency and performance in optimizing expensive-to-evaluate black-box functions. While the standard approach typically begins with uninformed priors or uniform sampling over the hyperparameter space, this study investigates whether the default hyperparameters included in popular ML libraries can serve as informative priors. These defaults are often viewed as expert-designed points that offer reasonable performance, yet their role in optimization remains unexamined.

Through extensive benchmarking across diverse models and datasets, supported by rigorous quantitative analysis and statistical testing, our results demonstrate that default hyperparameters do not provide meaningful guidance toward optimal regions in the search space. They offer no significant advantage in terms of final performance or convergence speed when compared to the baseline of uniform random sampling.

We conclude that practitioners should avoid relying on defaults when seeking optimal performance. Instead, Bayesian Optimization provides a principled approach that balances exploration and exploitation effectively. Our findings highlight the risk of assuming defaults are enough and reinforce the need for robust, uncertainty-aware optimization

strategies. Defaults are not informative; they are convenient, and convenience is not a reliable substitute for performance.

Keywords: Bayesian Optimization, Hyperparameter Tuning, Machine Learning, Informative Priors, Default Hyperparameters.

1. Introduction

The project investigates the role of default hyperparameters in Bayesian Optimization. The aim was to assess whether these defaults, commonly used in practice, could be leveraged to improve convergence and final performance by acting as informative priors of the Bayesian Optimization process. To have an introduction to the topic consult the Appendix A.1

2. Project Definition

We formulated a Bayesian Optimization pipeline that integrates default hyperparameter values from popular ML libraries as the mean of truncated Gaussian priors over the hyperparameter space. We compared this approach against standard random initialization across a diverse benchmark of Bayesian Optimization libraries, models and datasets. In addition to this, We compared the effect of using a single default initialization point versus a single point drawn from a uniform distribution in a restricted number of iterations of the method. We define the sampling-based initialization experiments as `Sample`, and the single-point, deterministic initialization using default hyperparameters as `Default`.

3. Description of the System

The framework uses three Bayesian Optimization libraries (BoTorch, Optuna, Skopt) and three types of models (Random Forest, MLP, SVM) on five supervised-learning benchmark datasets to ensure the hypothesis is tested on a representative sample of the vast and diverse landscape of Machine Learning and Bayesian Optimization. To ensure robustness, experiments varied the number of initialization points and the value of the standard deviation of the truncated gaussian distribution, analyzing their effect on optimization efficiency.

To statistically evaluate our formulated hypothesis, a one-sided binomial test was employed. This test specifically determines whether sampling-based initialization strategies significantly outperform random initialization beyond what would be expected by chance under the null hypothesis (H_0): sampling methods do not yield a statistically significant improvement compared to random initialization. A win is defined as an instance in which the sampling-based method surpasses random initialization by a domain-specific threshold, either in terms of faster convergence to an optimal configuration or by achieving greater final predictive performance. The resulting p-values quantify whether observed improvements represent statistically significant evidence to reject the null hypothesis or are likely attributable to random variation.

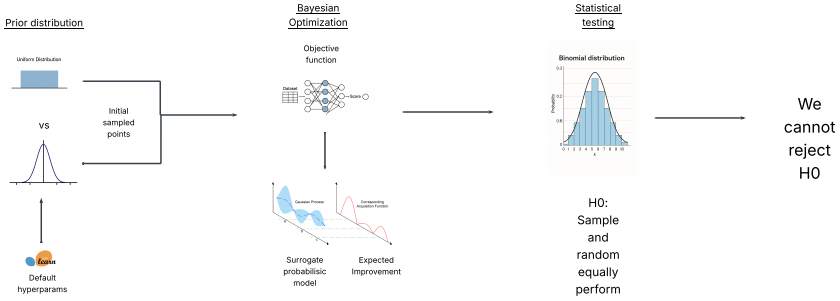

Figure 2: General pipeline overview and summary of methodology.

4. Results

- Default configurations did not offer statistically significant improvements in convergence or final performance.
- In multiple comparisons between strategies, binomial tests yielded p-values far above 0.05, indicating no consistent advantage.

Comparison	S/D	R	Ties	p-value
S vs. R Convergence	18	15	15	0.364
S vs. R Metric	10	12	26	0.738
D vs. R Convergence	19	12	17	0.141
D vs. R Metric	11	17	20	0.908

Table 2: Summary of win counts, ties, and one-sided binomial p-values for each comparison. S refers to the Sample strategy, R to Random, and D to Default. The Sample method uses multiple initialization points drawn from a truncated Gaussian distribution over the hyperparameter space, whereas the Default method relies on a single predefined configuration.

5. Conclusions

Our findings show that while default hyperparameter values are tempting due to their simplicity and initial performance, they do not provide robust guidance for optimization. In essence, they cannot be modeled as valid informative priors for Hyperparameter tuning using Bayesian Optimization. In fact, it is recommended to favor uncertainty-aware initialization strategies like those in standard Bayesian Optimization, instead of using default hyperparameters from ML libraries when performance is a requirement.

6. References

While all references are listed in the reference section, to facilitate a better understanding of the topic by section, we provide Table 12, which organizes the references accordingly.

Contents

1 Introduction

Bayesian optimization is the state-of-the-art technique for optimizing expensive-to-evaluate black-box functions, i.e., functions whose analytical expression is unknown. Hyperparameter tuning in machine or deep learning algorithms is a quintessential example of this family of optimization problems, as no gradients are available and the training process is usually expensive in terms of time and computation. The current standard approach is to use a non-informative prior given by an unconditioned Gaussian process model or sampling a small set of initial points from a uniform distribution over the hyperparameter space. The assumption behind this practice is that there is no prior knowledge about the location of the optimum, requiring an initial exploration of the complete space. From a Bayesian perspective, we argue that the default parameters provided by machine learning libraries may represent valid prior information for the optimum set of hyperparameter values. To test this hypothesis, we will a priori condition the Gaussian Process using these default point estimates, modeling them with prior distributions.

1.1 Context and Motivation

The increasing complexity of machine learning models has heightened the need for effective and efficient hyperparameter optimization, as ML algorithm hyperparameters play a critical role in the performance of the resulting models. Bayesian Optimization (BO) has become the leading framework for tackling this challenge due to its ability to manage exploration and exploitation efficiently and obtain promising results with a small number of evaluations, in contrast to other methods such as random search or evolutionary algorithms[1].

The standard approach of the process usually starts with a non-informative prior or uniform sampling, which may overlook valuable domain-specific knowledge. In practice, ML libraries like `scikit-learn`, `XGBoost`, and `LightGBM` include well-chosen default hyperparameter values that are broadly effective across many different datasets. These defaults may encode implicit expert knowledge, yet they have not been studied as valid informative priors.

By conditioning BO on the default hyperparameters of ML libraries, this study explores whether such defaults can act as such, thereby improving convergence rates and performance. This approach could offer a low-effort enhancement to standard BO pipelines, contributing to more computing-efficient hyperparameter tuning strategies.

The rest of the document is organized as follows: Section 2 reviews the state of the art in Bayesian Optimization, with a particular focus on prior modeling and initialization strategies. Section 3 defines the scope of the research by presenting the objectives, hypotheses, assumptions, and constraints that underpin the study. Section 4 presents the methodology, including the integration of default hyperparameters from widely-used ML libraries into existing Bayesian Optimization frameworks, as well as the overall evaluation design. Section 5 describes the experimental setup, including dataset selection, hyperparameter configuration, and the formulation of prior distributions. Section 6 discusses the results in both qualitative and quantitative terms, and includes a sensitivity analysis regarding the role of prior distributions. Section 7 summarizes the main conclusions and reflects on the broader implications of the findings. Lastly, Section 8 outlines directions for future research, while the Appendix provides complementary theoretical background and additional technical details.

2 State of the Art

Bayesian Optimization (BO) has been the focus of deep research aimed at improving its performance and convergence rates, particularly in the context of machine-learning hyperparameter tuning. Various efforts have targeted different components of the BO pipeline—acquisition functions, surrogate models, evaluation strategies, and the use of informative priors.

For acquisition functions, improvement-based rules such as LogEI provide strong empirical gains [2], while information-theoretic criteria rooted in entropy, Predictive Entropy Search, and Max-value Entropy Search, explicitly reduce uncertainty about the optimum and often require fewer evaluations to locate high-value regions [3], [4]. Trust-region methods like TuRBO scale effectively to high-dimensional problems by coupling global exploration with rigorous local exploitation [5]. In parallel hardware settings, batch BO schemes such as Local Penalisation issue multiple queries at once with near-linear speed-ups in wall-clock time [6]. Multi-fidelity BO further cuts cost by mixing cheap approximations of the objective with expensive high-fidelity evaluations [7], and tuning the surrogate kernel or acquisition-function hyperparameters themselves can bring additional efficiency gains [8].

Surrogate modelling has likewise diversified. Classical stationary Gaussian-process kernels remain popular, but deep-kernel learning combines GPs with neural feature extractors to capture non-stationary structure without sacrificing calibrated uncertainty, yielding improved performance on complex vision and language benchmarks [9]. When dimensionality is extreme, latent-space techniques such as REMBO embed the search in a random low-dimensional sub-space, enabling BO to operate on hundreds or thousands of variables while retaining convergence guarantees if the intrinsic dimension is low [10].

Beyond internal algorithmic refinements, many studies demonstrate the value of incorporating prior knowledge. In robotics, user-specified priors on likely poses or controller parameters shorten the learning curve for new tasks [11].

In machine-learning hyperparameter optimisation, informative-prior work is dominated by transfer learning. Gaussian-process models augmented with task-similarity metrics or ensemble priors leverage previous optimisation traces [12], [13], and meta-models can predict promising initial configurations that dramatically shorten subsequent BO runs [14].

Yet the hyperparameter defaults from mainstream ML libraries—chosen via extensive empirical testing—have not been systematically evaluated as priors. Because these defaults encode pragmatical expert knowledge and do not incur in any additional cost, conditioning BO on them may offer a simple, broadly applicable improvement to existing tuning pipelines.

3 Research Scope

This chapter clearly defines the main objectives of the research, formulates the hypotheses, presents the underlying assumptions, and outlines the restrictions that affect the study.

3.1 Objectives

The primary objective is to rigorously investigate whether the default hyperparameters provided by standard Machine Learning libraries are effective informative priors for Bayesian Optimization (BO). In particular, the following objectives will be addressed:

- Integrate default hyperparameters from common ML libraries into ex-

isting Bayesian Optimization frameworks as informative priors.

- Define and employ robust and rigurous quantitative and qualitative evaluation standards to assess the soundness and effectiveness of using these defaults as priors.
- Empirically validate the effectiveness of default hyperparameters as priors across diverse Machine Learning algorithms, Bayesian Optimization frameworks and benchmark datasets.
- Analyze whether the incorporation of these defaults leads to significant improvements in convergence speed and/or final model performance compared to the random baseline.

3.2 Hypotheses

The core hypothesis that sets a basis for this research can be formulated as the following question:

Are default hyperparameters provided by standard Machine Learning libraries valid informative priors for Bayesian Optimization?

Formally, we can define the two different hypotheses for statistical testing as follows:

Null Hypothesis (H_0):
Default configuration initialization methods do not perform statistically significantly better than random initialization. Formally:

$$H_0 : p_{\text{sampling}} \leq 0.5$$

Alternative Hypothesis (H_1):
Default configuration initialization methods perform statistically significantly better than random initialization. Formally:

$$H_1 : p_{\text{sampling}} > 0.5$$

p_{sampling} represents the proportion of times the sampling-based initialization method yields superior performance compared to random initialization. The criteria for superior performance will be formalized in subsequent sections, where we define the specific metrics and thresholds used for comparison.

By conducting this one-sided binomial test, we evaluate whether the probability of sampling-based initialization outperforming random initialization is significantly greater than chance (i.e., greater than 50%).

3.3 Assumptions

Several assumptions support this research:

- **Generality of Results:** By evaluating multiple machine learning models (Random Forest, Support Vector Machines, Multilayer Perceptron) and multiple Bayesian Optimization libraries (BoTorch, Optuna, Skopt), we assume that findings will generalize across typical supervised learning tasks.
- **Dataset Representativeness:** We assume that the chosen benchmark datasets represent a broad range of realistic scenarios in Machine Learning practice. This includes diversity in feature dimensionality, class balance, noise levels, and model sensitivity to hyperparameters.
- **Smoothness of objetctive function:** Bayesian Optimization relies on the assumption that the performance surface is continuous and sufficiently smooth such that local information can guide global search effectively. It is assumed that the objective function is suitable for the use of Bayesian Optimization.
- **Enough Evaluation Budget:** It is assumed that the fixed number of optimization evaluations per run is enough to yield meaningful and com-

parable performance across methods, even if full convergence is not always reached.

- **Feasibility of Hyperparameter Domains:** The defined hyperparameter search spaces are assumed to be wide enough to include quasi-optimal configurations but narrow enough to ensure feasible optimization within the evaluation budget.
- **Model Training Stability:** It is assumed that for each configuration, the corresponding model training is stable and converges reliably, without pathological or numerical failures.
- **Gaussian Prior Assumption:** It is assumed that the underlying distribution of effective hyperparameter configurations can be reasonably approximated by a Gaussian distribution.
- **Truncated Distribution Robustness:** The chosen variances are assumed sufficient to prevent biased sampling toward extreme or unrepresentative configurations.

3.4 Constraints and Restrictions

The study acknowledges several practical and computational constraints that shaped the experimental design. Each optimization run was limited in both runtime and number of model evaluations to ensure feasibility within the available resources. To further reduce computational load, all datasets were subsampled to a maximum of 1,000 instances, which was particularly important given the need for repeated evaluations across high-dimensional spaces. Moreover, the requirement to parallelize thousands of experiments and ensure statistically meaningful repetition made the inclusion of deep learning benchmarks infeasible. As a result, the study focuses exclusively on classical supervised learning models. The hardware environment consisted of a Linux-based machine with 60 GB of RAM and an NVIDIA GPU with 12 GB of memory. While adequate for classical machine learning models, these resources were insufficient for training or fine-tuning deep learning models at the needed scale.

4 Methodology

Before meticulously describing the methodology, we introduce the main steps of our approach. We propose initializing the optimization procedure sampling points from truncated gaussian distributions, centered around these default values. This scheme differs from the usual practice of uniform random sampling. We evaluate our hypothesis consistently across different machine learning models, various bayesian optimization libraries, and multiple benchmark datasets. This includes analyzing both the convergence speed towards optimal hyperparameters and the final predictive performance, employing statistical and visualization methods to ensure robustness and comprehensive understanding of the results.

4.1 Technical Description

Bayesian Optimization (BO) is a probabilistic and sample-efficient method for global optimization of black-box functions. Formally, let $f\colon X \to \mathbb{R}$ be an objective function defined over a bounded domain $X \subset \mathbb{R}^d$ where evaluations of f are expensive and non-differentiable. BO seeks to find

$$x^* = \arg\max_{x \in X} f(x)$$

using as few evaluations as possible. It achieves this by building a surrogate probabilistic model, in our case a Gaussian Process, to approximate f and uses an

acquisition function $\alpha(x)$ to decide where to sample next. For more details on the topic consult the Appendix A.1 or check the resources [15]–[17]

A common strategy initializes BO with points sampled from a uniform distribution over the parameter space X. In contrast, our approach proposes sampling initial points from a truncated normal distribution over the hyperparameter space and centered at the model's default hyperparameter values, reflecting a Bayesian prior belief that the optimal points are likely to be near these. In particular, for a parameter x_i with bounds $[a_i, b_i]$ the initial samples are drawn from

$$x_i \sim \text{TruncNormal}\Big(\mu_i, \sigma_i^2; a_i, b_i\Big),$$

where μ_i is the default hyperparameter value and $\sigma_i = \lambda(b_i - a_i)$ is the standard deviation proportional to the range of the parameter space, with $\lambda \in (0, 1)$ controlling the confidence in the prior belief, allowing for tunable expressiveness of the aforementioned.

To decide which point to sample next, we employ the Expected Improvement acquisition function

$$\alpha_{\text{EI}}(x) = \mathbb{E}\Big[\max\Big(f(x) - f(x^+),\, 0\Big)\Big],$$

where $f(x^+)$ denotes the best value observed so far. EI balances exploration and exploitation effectively and has become one of the most used acquisition functions in Bayesian Optimization [18]

The objective function in our case is defined as the cross-validated performance of a machine learning model trained with hyperparameters x. This methodology ensures the model's performance estimate is not biased by any train/test split.

$$f(x) = \frac{1}{k} \sum_{j=1}^{k} M_j(x),$$

where $M_j(x)$ is the performance metric for the supervised learning task. We have chosen to use accuracy for classification tasks and root mean squared error for regression.

Following the philosophy of cross-validation, to ensure that the observed performance is not due to randomness or an isolated artifact of a particular sampling configuration, we systematically evaluate our hypothesis across a range of reasonably small standard deviation values for the truncated distribution. We vary the standard deviation, keeping it in a sufficiently narrow range that reflects confidence in the informative prior means. We also use different numbers of initialization points for the same reason.

4.2 Libraries and Models Choice

To explore the space of Bayesian optimization methods and ensure generality of the results we use the three following optimization backends:

- **Scikit-Optimize (Skopt):** which is a GP-based library with a simple but effective use.
- **BoTorch:** which is a GP-based library with advanced gradient-powered optimization and a more complex but powerful use.
- **Optuna:** A TPE-based method that uses a tree-structure as a surrogate model.

These three libraries cover the whole actual landscape of Bayesian Optimization, including both kernel-based probabilistic surrogates and tree-based nonparametric ones.

Similarly, to provide a representative landscape of the most popular supervised learning models [19], We evaluate our hypothesis across three different model families:

- **Random Forests:** Powerful ensemble models that aggregate decision

trees providing strong performance and robustness to noise.

- **Support Vector Machines:** Are effective in high dimensional spaces and can model very complex decision boundaries via infinite-dimensional kernel functions.
- **Multilayer Perceptron:** Are feed-forward networks theoretically capable of approximating any arbitrary continuous function.

We do not include linear regression or logistic regression in our study, as these models expose a very limited number of hyperparameters. Although they consistently rank among the most widely used models in the machine learning community [19], their simplicity makes them less suitable for evaluating the impact of different initialization strategies in hyperparameter optimization.

By using the Scikit-learn library and evaluating Random Forests, Support Vector Machines, and Multilayer Perceptrons, We ensure coverage of a wide range of inductive biases in the learning process related to distinct model arquitectures— namely ensemble-based decision trees, margin-based kernel methods and deep nonlinear function approximators.

We selected five datasets for benchmarking, two for regression and three for classification.

4.3 Metrics and Evaluation

Our hypothesis would be validated if the default configuration demonstrates at least one of the following results:

- Faster convergence towards a reasonably high-performance solution.
- Achievement of a better solution than the baseline.

To expose the metrics and evaluation strategies used to check the validity of the results, we need to introduce the following notation:

Symbol	Description
D	Set of datasets, i
M	Set of models, j
S	Initialization strategies (`default`, `random`)
R	Number of repetitions per experiment
T	Number of recorded iterations
$r_{i,j}^{(s)}(t)$	Running-best metric at iteration t
$t_{i,j,s}^{*(r)}$	Iteration when best metric is first reached
$f_{i,j,s}^{*(r)}$	Best metric value achieved

Table 3: Notation used throughout the Metrics and Evaluation section.

Because the number of convergence traces grows combinatorially, it is impossible to examine every curve by eye. Instead, we propose a quantitative way of measuring the performance of the informed prior method against the uninformed baseline. For each combination of optimization backend, model, dataset, and task we record two things: the convergence index, defined as the iteration at which the best performance is reached, and the best metric value itself—accuracy or RMSE. For consistency, RMSE is negated so that higher values always indicate better performance, aligning it with the behavior of accuracy. The results are aggregated by averaging these metrics across runs for both initialization strategies (different number of initialization points and values standard deviations). The relative improvements of the default-centered method over the random baseline are then compared with a pre-set threshold to decide whether the advantage is practically significant. We can summarize this with the following equations:

Mean convergence and metric

$$\mu_{i,j}^{\text{conv}}(s) = \overline{t_{i,j,s}^{*(r)}}, \qquad \mu_{i,j}^{\text{metric}}(s) = \overline{f_{i,j,s}^{*(r)}}. \tag{1}$$

Relative improvement (Default vs. Random)

$$\Delta_{i,j}^{\text{conv}} = \frac{\mu_{i,j}^{\text{conv}}(\text{random}) - \mu_{i,j}^{\text{conv}}(\text{default})}{\mu_{i,j}^{\text{conv}}(\text{random})} \tag{2}$$

$$\Delta_{i,j}^{\text{metric}} = \frac{\mu_{i,j}^{\text{metric}}(\text{default}) - \mu_{i,j}^{\text{metric}}(\text{random})}{\mu_{i,j}^{\text{metric}}(\text{random})} \tag{3}$$

Threshold-based indicators

$$\mathbf{1}_{\text{conv}}^{(i,j)} = \begin{cases} 1 & \Delta_{i,j}^{\text{conv}} > \tau_{\text{conv}}, \\ 0 & \text{otherwise}, \end{cases} \tag{4}$$

$$\mathbf{1}_{\text{metric}}^{(i,j)} = \begin{cases} 1 & \Delta_{i,j}^{\text{metric}} > \tau_{\text{metric}}, \\ 0 & \text{otherwise}. \end{cases} \tag{5}$$

Total wins

$$N_{\text{conv}} = \sum_{(i,j)\in D\times M} \mathbf{1}_{\text{conv}}^{(i,j)} \tag{6}$$

$$N_{\text{metric}} = \sum_{(i,j)\in D\times M} \mathbf{1}_{\text{metric}}^{(i,j)} \tag{7}$$

Next, we will count how many times the default configuration method outperforms the random method, and vice versa. We will then perform a one-sided binomial test to determine whether we can reject the null hypothesis that the probability of winning of the sampling method is less than or equal random. For simplicity, ties will be excluded from the analysis.

One could argue that the analysis of a paired-difference test would be more adequate to our subject of study. However, paired-difference tests address the question "Is the average improvement nonzero?" whereas our methodological requirement is to ask, "In how many cases does the improvement exceed this domain-specific threshold?" Because adherence to the expert threshold is vital to our evaluation, we cannot rely on paired-difference tests.

To support a more qualitative, visual investigation of the convergence traces, we select a single dataset on which hyperparameter configuration has the greatest impact observed. We propose a metric called the relative spread to guide this selection, which is defined as follows:

Spread metric

$$\text{Spread}(D_i) = \frac{1}{|M|\,|S|} \sum_{(j,s)\in M\times S} \frac{\max_t r_{i,j}^{(s)}(t) - \min_t r_{i,j}^{(s)}(t)}{\min_t r_{i,j}^{(s)}(t)}. \tag{8}$$

We consider clarifying to use the selected dataset to examine the effect of increasing the standard deviation of the truncated normal distribution used for initial sampling. This transitions the prior from a strongly informed one (centered tightly around the default configuration) to a looser, less informative prior that incorporates more noise and uncertainty. Our hypothesis is that as the standard deviation increases, the convergence behavior should worsen, i.e. the optimization takes longer to reach its optimum performance (slower convergence), or it fails to find configurations as effective as it would have (lower final performance). By systematically varying the standard deviation across a reasonable range and plotting the resulting convergence curves, we aim to assess whether this expected degradation is in practice real.

We propose a general diagram to clarify the methodology (see Figure 3).

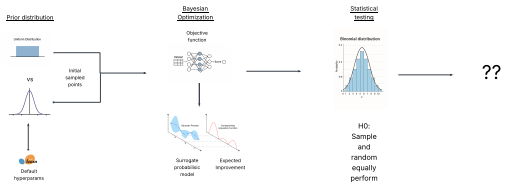

Figure 3: General overview of the proposed methodology.

5 Experiments

This chapter details the actual implementation of the methodology aforementioned. It presents the choice of standard deviation values, datasets, default hyperparameter values and ranges, thresholds for metrics, and other configuration settings.

5.1 Dataset Selection

To ensure both diversity and consistency in experimental conditions, we select five datasets—three for classification tasks and two for regression tasks. These datasets are subsampled to a maximum of 1,000 instances (via a fixed random seed) to control training time and ensure reproducibility. The chosen datasets cover both simpler and more challenging settings:

Name	Description
AmazonAcc	Binary access prediction from human resources
Letter	Multiclass character recognition from images
Higgs	Particle physics dataset (large and noisy)

Table 4: Classification datasets (subsampled to 1,000 instances).

Name	Description
BikeSharing	Bike rental demand over time
Airlines10M	Airline delay prediction (very large-scale)

Table 5: Regression datasets (subsampled to 1,000 instances).

5.2 Hyperparameter Selection

The default values for each hyperparameter were extracted directly from the official Scikit-learn implementation of each model [20]. Bounds for each hyperparameter's search space follow standard practices in the hyperparameter optimization literature (e.g., [21]) and common benchmarking studies. All hyperparameter defaults and bounds are listed in Tables 9, 10, and 11 in Appendix A.2.

5.3 Standard Deviation and Thresholds

To span a range of prior confidence levels while remaining "reasonably concentrated," we choose five evenly spaced λ values in $[0.05, 0.30]$. Then we have that

$$\begin{aligned} \sigma &= \lambda\,(b-a), \quad \lambda = 0.05, \\ \Pr\big(|X-\mu| \le \sigma\big) &= 0.68, \\ \frac{2\,\sigma}{b-a} &= 0.10. \end{aligned} \tag{9}$$

Eq. (9) shows that for $\lambda = 0.05$, 68% of the normal mass lies within $\pm\sigma$, which corresponds to 10% of the total range $[a, b]$. Conversely, $\lambda = 0.30$ marks our upper noise-tolerance limit. This structured design avoids spurious chance effects and enables aggregation.

We define the metric threshold as the minimum relative spread observed across all datasets (see Eq. (8)), here 0.3%, so only improvements exceeding the smallest

meaningful variation are counted as significant. For convergence speed, we fix a 10% relative-improvement threshold.

5.4 Configuration

All experiments use 3-fold cross-validation ($k = 3$) to estimate the performance of each hyperparameter configuration reliably.

We evaluate two distinct initialization strategies against a random baseline:

1. **Default vs. Random:** In this setting, Bayesian Optimization is initialized with a single deterministic point, namely the model's default hyperparameter configuration. A total of 15 iterations are run.
2. **Sample vs. Random:** In this setting, the optimization process is initialized with multiple points (three, four, or five) sampled from a truncated Gaussian distribution centered at the default values. This represents a probabilistic version of the default prior, where nearby points are explored. A total of 30 iterations are performed. This setup evaluates whether an uncertainty-aware initialization improves over uninformed random baselines.

6 Results

6.1 Qualitative Results

Before starting to give qualitative, graphical results of the convergence results, for the sake of size of this study, we will select the datasets that more information provide based on the spread distance (Eq. (8)). The following table provides the results 6:

Dataset	Spread Distance
Airlines10M	0.019
AmazonAcc	0.003
BikeSharing	0.18
Higgs	0.017
Letter	0.12

Table 6: Spread distance for each dataset.

We will not use the dataset AmazonAcc for the following qualitative visualizations because it provides little information.

The distributional summaries in the boxplots(4), 5 reveal no systematic or practically meaningful difference between the competing initialisation strategies. After normalising performance within each dataset, the inter-quartile ranges of default and random (15-evaluation budget) 4 and of sample and random (30-evaluation budget) 5 overlap almost completely. In the absence of a consistent shift in central tendency or dispersion, a visual inspection provides no evidence that default (or sample) confers a persistent advantage over a purely random baseline.

The convergence trajectories lead to the same conclusion 6. With a 30-evaluation budget, the sample curves do begin at a higher score than their random opponent. However, the gap contracts quickly, and both methods approach a common plateau within the same evaluation horizon. When the budget is restricted to 15 evaluations, the two methods stabilise at indistinguishable performance levels. Hence, whatever exploratory head-start sample (or default) may enjoy at iteration zero is transient: random search closes the gap rapidly, and all strategies converge to essentially the same asymptotic outcome within an equivalent number of evaluations.

These observations remain qualitative. In subsequent chapters the visual impressions will be subjected to statistical verification.

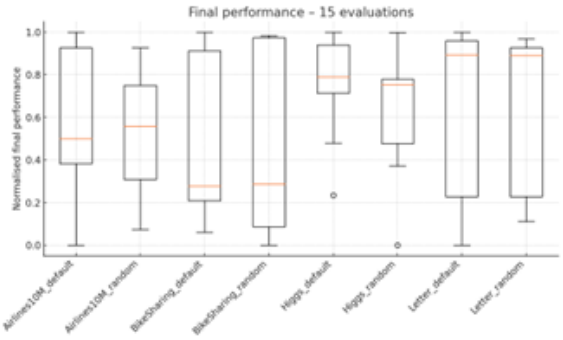

Figure 4: Final performance after 15 evaluations (normalized).

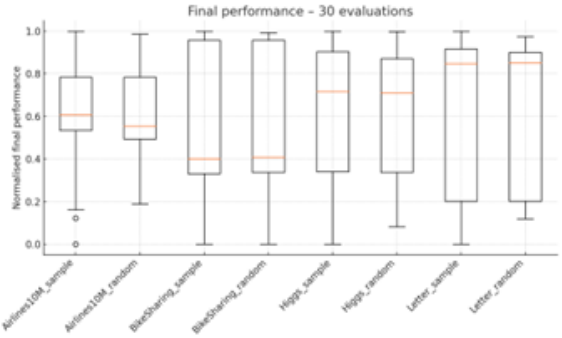

Figure 5: Final performance after 30 evaluations (normalized).

6.2 Quantitative Results

Since our methodology has already been detailed, we focus here on interpreting the counts of wins and the associated binomial p-values.

Examining the convergence counts under our 10% threshold, we found that out of 48 combinations, sample beat random in 18 cases while random beat sample in 15, leaving 15 ties. The resulting binomial p-value was 0.364—far above 0.05—indicating no statistically significant advantage for sample over random in convergence speed. On the metric side for 30 evaluations, sample outperformed random in 10 cases, while random outperformed sample in 12, with 26 ties; here the p-value rose to 0.738, also nonsignificant. Moving to the 15-evaluation comparison, default converged at least 10% faster than random in 19 of 48 combinations, whereas random did so in 12

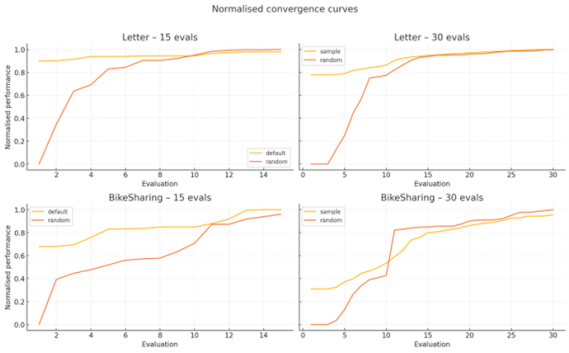

Figure 6: Convergence trajectories (normalized running-best vs. iteration).

cases, leaving 17 ties. The corresponding p-value was 0.141, again failing to reject the null hypothesis. Finally, when looking at the final metric for 15 evaluations, default secured at least a 0.3% improvement over random in 11 cases versus 17 for random, with 20 ties, yielding a p-value of 0.908—clearly nonsignificant. You can check the results in the table below (see Table 7).

Comparison	S/D	R	Ties	p-value
S vs. R Convergence	18	15	15	0.364
S vs. R Metric	10	12	26	0.738
D vs. R Convergence	19	12	17	0.141
D vs. R Metric	11	17	20	0.908

Table 7: Summary of win counts, ties, and one-sided binomial p-values for each comparison. S refers to the Sample strategy, R to Random, and D to Default. The Sample method uses multiple initialization points drawn from a truncated Gaussian distribution over the hyperparameter space, whereas the Default method relies on a single predefined configuration.

Taken together, these results show that neither the sample-based approach with 30 evaluations nor the default deterministic approach with 15 evaluations produces a statistically significant win over

random initialization in terms of convergence. Hence, we cannot reject the null hypothesis that Default configurations are not a valid informative prior for localizing the global optimum. Quantitatively, the data does not support any strong claim that starting from default settings leads to faster or better convergence than sampling points at random.

6.3 Sensibility analysis of Standard Deviation.

In this section, we will study the effect of varying the standard deviation of the truncated gaussian distribution on the performance of the method proposed on the dataset letter, which has been identified as particularly sensible to different hyperparameters configurations by the sparse distance table 6 and provides very clarifying insights. In particular, we explore the consequences in the maximum value reached, the average value of the running best, and in the convergence index. We explore the sample method as we consider it to be the more insightful. All metrics are averaged across the different models and normalized using min–max scaling to allow for consistent visual comparison.

As we can observe in the figure 7, the final performance of the method remains largely unchanged across different values of the standard deviation (sdv). The scatterplot shows a flat distribution of values, indicating that the optimizer consistently reaches high-performing solutions regardless of noise levels. This observation is quantitatively supported by the correlation table 8, which reveals a near-zero Pearson correlation and a non-significant p-value.

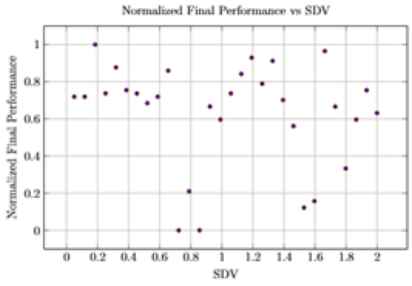

Figure 7: Normalized final performance across SDV values.

Metric	**Pearson** r	p**-value**
final perf	-0.182	0.335
mean perf	-0.626	2.185×10^{-4}
conv. index	0.048	0.801
early mean	-0.641	1.368×10^{-4}
late mean	-0.185	0.328

Table 8: Pearson correlation coefficients between standard deviation (SDV) and various normalized metrics.

When we look at the plot 8 we observe a clear downward trend as sdv increases. This indicates that although the optimizer reaches similarly good results, the journey toward that solution becomes less efficient in noisier settings. The optimizer spends more time evaluating poor or misleading configurations, which lowers the overall average performance throughout the run. This is statistically confirmed by the correlation table, where the negative Pearson coefficient and the highly significant p-value reflect a robust relationship between noise and degradation in mean performance. This degradation, however, is not caused by a longer time to convergence or by a worse final performance—as the next analysis reveals—but by poorer initial evaluations.

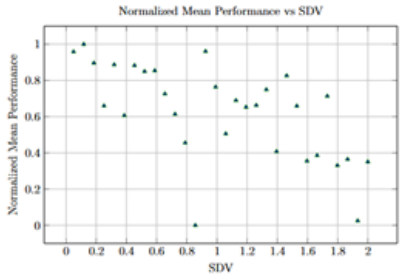

Figure 8: Normalized mean running-best performance across SDV values.

When we examine the normalized convergence index plot 9, we find no clear correlation with sdv. The convergence index remains approximately constant across noise levels. This suggests that, despite noisy and poor evaluations, the optimizer often finds its final solution at a similar stage in the run. This observation may initially appear inconsistent with the decline seen in mean performance, but the key insight lies in understanding what kind of information is available to the optimizer.

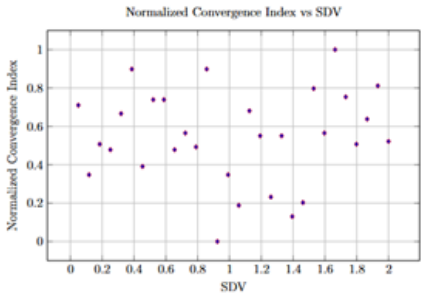

Figure 9: Normalized convergence index versus SDV values.

The last two metrics (8, 9) seem to contradict each other, but this discrepancy is explained by the behavior of default configurations. Defaults are designed using expert knowledge accumulated across many prior tasks. As a result, they are deliberately chosen to lie in high-performing regions of the search space. This makes them excellent single-shot guesses: a default setting is much more likely to land in a "safe" zone of decent performance than a purely random sample, which may fall in poor-performing regions. This is precisely why the early part of the optimization process often looks better in default-initialized settings. However, this early advantage can be misleading. As shown in the next plot 10, the early mean decreases with higher sdv, while the late mean remains stable. This indicates that noise primarily damages early evaluations, particularly those around the default without affecting the optimizer's eventual convergence.

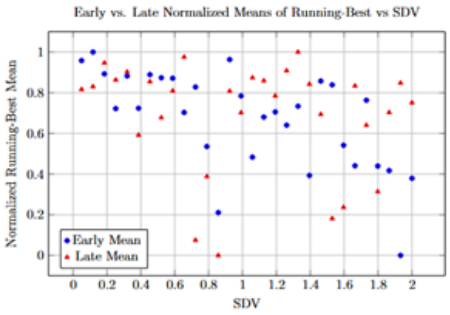

Figure 10: Early vs. Late mean performance across SDV values.

We therefore conclude that the degradation in mean performance is primarily due to early evaluations near default configurations. These defaults, while better than random samples at first glance, are not informative for optimization. They offer no insight into how performance varies across broader space, nor do they help identify the location in which the optimum lies. For an explanation based on how Gaussian Process regression work check Appendix A.1.

7 Conclusion

The main objective of this project was to evaluate whether default hyperparameter configurations from popular machine learning libraries could serve as informative priors for hyperparameter optimization. Our initial aim was to explore strategies to incorporate these defaults into the

optimization process—either through probabilistic prior sampling or deterministic initializations—in order to accelerate convergence and improve performance in a sample-efficient manner.

However, the results led us to a more powerful conclusion: the use of default configurations is misleading. While they occasionally produced decent initial evaluations—thanks to being designed by experts for general-purpose use—they offered no consistent or reliable guidance about the location of optimal configurations. Our experimental analysis showed that methods initialized at or around the defaults converged no faster, and yielded no better final performance, than purely random baselines. We validated this via qualitative analysis and in rigorous quantitative and stadistical tests across multiple models and datasets.

The conclusion is therefore clear and significant: default hyperparameters, though convenient, should not be relied upon when high performance is a requirement. They may encourage premature convergence or instill a false sense of security, ultimately hindering optimization rather than helping it. Bayesian optimization and other principled search strategies remain essential when model tuning is critical, as they provide a structured exploration of the hyperparameter space based on actual performance feedback—not assumptions rooted in arbitrary defaults.

This insight has direct implications for practitioners and researchers alike. In automated machine learning systems blind trust in default configurations may compromise the validity or competitiveness of results. Instead, the community should embrace optimization as an integral part of model development, even in seemingly routine setups.

7.1 Objectives Achieved

All core objectives outlined at the beginning of this study have been covered. Specifically, through the following contributions:

- **Evaluate the use of defaults as priors:** Accomplished through theoretical formalization and probabilistic modeling of truncated normal distributions centered around default values.
- **Design a systematic evaluation methodology:** Implemented a robust framework using cross-validation, convergence thresholds, and metric-based indicators to ensure reliable comparison across initialization strategies
- **Quantitatively assess the influence of default configurations:** Conducted comprehensive experiments across multiple models (Random Forests, SVMs, MLPs) and datasets (classification and regression), using statistical significance tests (binomial tests, correlation analysis).
- **Visual and statistical comparison of optimization dynamics:** Delivered via convergence plots, normalized performance metrics, and spread-based dataset selection.
- **Explore the effect of prior strength via standard deviation sensitivity:** A dedicated analysis on the LETTER dataset with varying standard deviation values confirmed that broader priors reduce early performance but do not impact final convergence quality.

7.2 On Deep Learning Experiments

As part of this project, we also conducted exploratory experiments in the domain of deep learning, focusing specifically on *fine-tuning and full training* of visual models (e.g., CNNs and vision transformers). The intent was to determine whether similar conclusions about defaults being misleading held in deep neural network arquitectures.

However, we decided not to include these results in the final manuscript for two reasons:

- **Redundancy**: The insights from the deep learning experiments closely mirrored those found in the machine learning setups. Including them would have added volume without providing fundamentally new understanding.
- **Sample Size Limitations**: Due to the high computational cost of training vision models, our experimental runs were limited in number. Relying on such small-sample results could have undermined the statistical confidence of our broader conclusions.

8 Future Work

This study opens several promising research directions:

- **Learning priors from experience**: Rather than relying on static defaults, future work could explore meta-learning frameworks that learn priors from similar tasks or domains.
- **Extending to other learning paradigms**: It would be valuable to test whether the same findings hold in areas like reinforcement learning or generative modeling, where hyperparameters play a critical role.
- **Multi-fidelity BO**: Do research on multi-fidelity Bayesian optimization methods that can leverage low-cost evaluations (e.g., early stopping) to guide hyperparameter search more efficiently.

References

[1] R. Turner, D. Eriksson, M. McCourt, *et al.*, "Bayesian optimization is superior to random search for machine learning hyperparameter tuning: Analysis of the black-box optimization challenge 2020," *arXiv preprint arXiv:2104.10201*, 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2104.10201.

[2] S. Ament, S. Daulton, D. Eriksson, M. Balandat, and E. Bakshy, "Unexpected improvements to expected improvement for bayesian optimization," *arXiv preprint arXiv:2310.20708*, 2025, NeurIPS 2023 Spotlight. [Online]. Available: https://doi.org/10.48550/arXiv.2310.20708.

[3] J. M. Hernández-Lobato, M. W. Hoffman, and Z. Ghahramani, "Predictive entropy search for efficient global optimization of black-box functions," *arXiv preprint arXiv:1406.2541*, 2014. [Online]. Available: https://doi.org/10.48550/arXiv.1406.2541.

[4] Z. Wang and S. Jegelka, "Max-value entropy search for efficient bayesian optimization," *arXiv preprint arXiv:1703.01968*, 2018, Proceedings of the 34th International Conference on Machine Learning, PMLR 70, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1703.01968.

[5] D. Eriksson, M. Pearce, J. R. Gardner, R. Turner, and M. Poloczek, "Scalable global optimization via local bayesian optimization," in *Advances in Neural Information Processing Systems*, NeurIPS 2019 Spotlight, vol. 32, 2019, pp. 5497–5508. [Online]. Available: https://doi.org/10.48550/arXiv.1910.01739.

[6] J. Gonzalez, Z. Dai, P. Hennig, and N. Lawrence, "Batch bayesian optimization via local penalization," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, A. Gretton and C. C. Robert, Eds., ser. Proceedings of Machine Learning Research, vol. 51, Cadiz, Spain: PMLR, 2016, pp. 648–657. [Online]. Available: https://proceedings.mlr.press/v51/gonzalez16a.html.

[7] J. Wu, S. Toscano-Palmerin, P. I. Frazier, and A. G. Wilson, "Practical multi-fidelity bayesian optimization for hyperparameter tuning," in *Proceedings of The 35th Uncertainty in Artificial Intelligence Conference*, R. P. Adams and V. Gogate, Eds., ser. Proceedings of Machine Learning Research, vol. 115, PMLR, 2020, pp. 788–798. [Online]. Available: https://proceedings.mlr.press/v115/wu20a.html.

[8] M. Lindauer, M. Feurer, K. Eggensperger, A. Biedenkapp, and F. Hutter, "Towards assessing the impact of bayesian optimization's own hyperparameters," *arXiv preprint arXiv:1908.06674*, 2019, Accepted at DSO Workshop, IJCAI 2019. [Online]. Available: https://doi.org/10.48550/arXiv.1908.06674.

[9] A. G. Wilson, Z. Hu, R. Salakhutdinov, and E. P. Xing, "Deep kernel learning," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, A. Gretton and C. C. Robert, Eds., ser. Proceedings of Machine Learning Research, vol. 51, Cadiz, Spain: PMLR, 2016, pp. 370–378. [Online]. Available: https://proceedings.mlr.press/v51/wilson16.html.

[10] Z. Wang, F. Hutter, M. Zoghi, D. Matheson, and N. de Freitas, "Bayesian optimization in a billion dimensions via random embeddings," in *Journal of Artificial Intelligence Research*, vol. 55, 2016, pp. 361–387. [Online]. Available: https://doi.org/10.48550/arXiv.1301.1942.

[11] M. Mayr, C. Hvarfner, K. Chatzilygeroudis, L. Nardi, and V. Krueger, "Learning skill-based industrial robot tasks with user priors," in *2022 IEEE International Conference on Automation Science and Engineering (CASE)*, Accepted, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2208.01605.

[12] P. Tighineanu, K. Skubch, P. Baireuther, A. Reiss, F. Berkenkamp, and J. Vinogradska, "Transfer learning with gaussian processes for bayesian optimization," *arXiv preprint arXiv:2111.11223*, 2022. [Online]. Available: https://doi.org/10.48550/arXiv.2111.11223.

[13] T. Bai, Y. Li, Y. Shen, X. Zhang, W. Zhang, and B. Cui, "Transfer learning for bayesian optimization: A survey," *arXiv preprint arXiv:2302.05927*, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2302.05927.

[14] S. K. Jungtaek Kim, *Learning to Transfer Initializations for Bayesian Hyperparameter Optimization*. Bayes Opt Book, 2017. [Online]. Available: https://bayesopt.github.io/papers/2017/8.pdf.

[15] E. C. Garrido Merchán, "Advanced methods for bayesian optimization in complex scenarios," Unpublished doctoral dissertation. Fecha de lectura: 26-07-2021, PhD thesis, Universidad Autónoma de Madrid, Escuela Politécnica Superior, Departamento de Ingeniería Informática, 2021. [Online]. Available: http://hdl.handle.net/10486/699441.

[16] P. I. Frazier, "A tutorial on bayesian optimization," *arXiv preprint arXiv:1807.02811*, 2018. [Online]. Available: https://doi.org/10.48550/arXiv.1807.02811.

[17] R. Garnett, *Bayesian Optimization*. Cambridge University Press, 2023.

[18] J. Snoek, H. Larochelle, and R. P. Adams, *Practical bayesian optimization of machine learning algorithms*, 2012. arXiv: 1206.2944 [stat.ML]. [Online]. Available: https://arxiv.org/abs/1206.2944.

[19] P. Mooney, *2022 kaggle machine learning and data science survey*, https://kaggle.com/competitions/kaggle-survey-2022, Kaggle, 2022.

[20] F. Pedregosa, G. Varoquaux, A. Gramfort, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[21] K. Eggensperger, F. Hutter, H. H. Hoos, and K. Leyton-Brown, "Efficient benchmarking of hyperparameter optimizers via surrogates," in *Proceedings of the AAAI Workshop on Bayesian Optimization*, 2013.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.

[23] B. Bischl, M. Binder, M. Lang, *et al.*, "Hyperparameter optimization: Foundations, algorithms, best practices and open challenges," *arXiv preprint arXiv:2107.05847*, 2021.

[24] V. Nguyen, "Recent advances in bayesian optimization," *arXiv preprint arXiv:2003.01870*, 2020.

[25] E. H. Lee, V. Perrone, C. Archambeau, and M. Seeger, "Cost-aware bayesian optimization," *arXiv preprint arXiv:2003.10870*, 2020.

A Appendix

A.1 Bayesian Optimization Introduction

A.1.1 Gaussian Processes

A **Gaussian Process** (GP) is a collection of random variables, any finite number of which have a joint Gaussian distribution. Formally, a GP is defined as:

$$f(x) \sim \mathcal{GP}(m(x), k(x, x'))$$

where:

- $m(x) = \mathbb{E}[f(x)]$ is the mean function,
- $k(x, x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$ is the covariance (kernel) function.

For a finite set of inputs $X = \{x_1, \ldots, x_n\}$, the output vector $\mathbf{f} = [f(x_1), \ldots, f(x_n)]^\top$ follows a multivariate normal distribution:

$$\mathbf{f} \sim \mathcal{N}(\mathbf{m}, K)$$

with $\mathbf{m} = [m(x_1), \ldots, m(x_n)]^\top$ and $K_{ij} = k(x_i, x_j)$. For more information on the topic read [22]

A.1.2 Kernels and Uncertainty Quantification

The kernel function $k(x, x')$ quantifies the similarity between input points. A commonly used kernel is the **Squared Exponential (SE)** kernel:

$$k(x, x') = \sigma_f^2 \exp\left(-\frac{\|x - x'\|^2}{2\ell^2}\right)$$

This kernel gives high similarity values for nearby points and rapidly decays for distant ones. This property is crucial for uncertainty quantification:

- If a point x' is close to a previously observed point x, then $k(x, x') \approx \sigma_f^2$ and the predictive variance $\sigma^2(x')$ is low.
- If a point is far from all known data, the kernel returns low values, leading to higher uncertainty in predictions.

The kernel thus controls how information from training data generalizes to nearby points, shaping the model's confidence across the input space.

A.1.3 Bayesian Optimization with Gaussian Processes

Bayesian Optimization (BO) is a global optimization strategy suited for expensive-to-evaluate functions. It relies on a Gaussian Process to model the objective function and an acquisition function to guide the next evaluation.

Steps:

1. Place a GP prior over the unknown function $f(x)$.
2. Given data $\mathcal{D}_t = \{(x_i, y_i)\}_{i=1}^t$, compute the posterior:

$$f(x) \mid \mathcal{D}_t \sim \mathcal{GP}(\mu_t(x), \sigma_t^2(x))$$

   where:

$$\mu_t(x) = k(x, X)^\top [K + \sigma_n^2 I]^{-1} \mathbf{y}, \quad \sigma_t^2(x) = k(x, x) - k(x, X)^\top [K + \sigma_n^2 I]^{-1} k(x, X)$$

3. Select the next query point using an acquisition function, e.g.:

$$x_{t+1} = \arg\max_x \alpha_t(x)$$

A.1.4 Informative Priors vs Misleading Defaults

If the default hyperparameters were indeed close to the optimum, our proposed strategy could significantly improve convergence speed and solution quality by guiding exploration to a promising region. In such cases, the prior is *informative* and acts as a valuable head start.

However, when the default is not close to the optimum, it introduces serious risks:

- **Posterior Bias:** The posterior mean remains biased towards the prior, potentially misguiding the search.
- **Suppressed Exploration:** The prior affects the variance estimate, discouraging exploration in low-prior regions—even if they contain the optimum.
- **Misleading Assumptions:** Defaults reflect safe or average-case configurations, not performance peaks. The GP implicitly favors functions near the shape implied by the prior and kernel. A misaligned prior constrains the search to unpromising subspaces.

Thus, using defaults as priors can lead to *conformist* behavior: the model quickly finds decent but suboptimal solutions and fails to explore regions that might hold better performance. Our empirical results support this, showing no statistical advantage of default-based initializations over random ones.

A.2 Hyperparameter Tables

A.2.1 Random Forests

Hyperparameter	Default (C, R)	Bounds	Description
Number estimators	500, 500	[100, 2000]	Number of trees grown in the forest.
Max. features	$\sqrt{\#\text{feat}}$, $\frac{\#\text{feat}}{3}$	[1, #feat]	Maximum features considered when selecting a split.
Max. samples	all, all	[50%data, all]	Fraction or count of samples drawn for each tree.
Min. samples leaf	1, 5	[1, 20] (C), [2, 50] (R)	Minimum samples that a leaf node must contain.

Table 9: Random-Forest hyperparameter defaults (values are *Classifier*, *Regressor*), bounds, and concise sentence descriptions.

A.2.2 Multilayer Perceptrons

Hyperparameter	Default	Bounds	Description
n_layers	1	[1, 5]	Number of hidden layers
hidden_units	100	[50, 500]	Units per hidden layer
alpha	0.001	$[10^{-5}, 10^{-1}]$	L_2 regularization strength
lr	0.01	$[10^{-4}, 10^{-1}]$	Initial learning rate

Table 10: Multilayer Perceptron hyperparameter defaults and bounds.

A.2.3 Support Vector Machines

Hyperparameter	Default	Bounds	Applies To	Description
C	1.0	[0.1, 10.0]	SVC, SVR	Regularization strength
gamma	0.01	$[10^{-4}, 1.0]$	SVC, SVR	RBF/polynomial kernel coefficient
degree	3	[2, 5]	SVC, SVR	Degree of polynomial kernel
coef0	0.0	[0.0, 1.0]	SVC only	Offset in polynomial kernel
tol	10^{-3}	$[10^{-4}, 10^{-1}]$	SVC only	Stopping criterion tolerance
epsilon	0.1	$[10^{-4}, 1.0]$	SVR only	ε-insensitive tube

Table 11: Support Vector Machine hyperparameter defaults and bounds.

A.3 Categorized references.

Category	**References**
Foundations, Surveys & GP Theory	[15]–[17], [22]–[24]
Acquisition Functions & Core BO Algorithms	[2]–[4], [6], [8]
Scalability, Efficiency & High-Dimensionality	[5]–[7], [10], [25]
Transfer Learning & Priors for BO	[11]–[14]
Benchmarking, Best Practices & Tools	[1], [20]

Table 12: Categorization of Bayesian-optimization literature with categories and cited references.