# Multimodal Large Language Models for Hyperparameter Optimization

**Roshan Sridhar**
**Oracle America, Inc.**

**Longjiao Zhang**
**Oracle America, Inc.**

## Abstract

This study aims to automate machine learning tasks using LLM agents by examining the performance of Multimodal Large Language Models (MM-LLMs) in hyperparameter optimization (HPO). We believe MM-LLMs may perform better than traditional Large Language Models (LLMs) and other methods like random search and Bayesian optimization because they can process both visual and textual data. We used the HPOBench dataset for our experiments to compare MM-LLMs with LLMs and other established HPO methods. Our results indicate that MM-LLMs not only improve the efficiency of the HPO process but also offer deeper insights through their ability to interpret data, potentially setting a new benchmark in automated machine learning methods.

## 1 Introduction

Hyperparameter Optimization (HPO) is a widely used practice in traditional machine learning that involves selecting a set of parameters to control aspects like model complexity and regularization strategy. This is crucial for ensuring the model performs well on new datasets. Various methods, such as random search, grid search, and Bayesian optimization, have been developed to automate HPO tasks [1,10]. The main hypothesis of this research is that Multimodal Large Language Models (MM-LLMs) can greatly improve HPO tasks by processing both textual and visual data, potentially outperforming traditional LLMs and standard methods like random search and Bayesian optimization. We explore: 1) how single-modal LLMs perform in HPO tasks, 2) whether MM-LLMs can surpass single-modal LLMs in HPO by using their ability to analyze visual data [8], and 3) if MM-LLMs can combine data from graphs and text to create a unified analysis for HPO.

Previous studies have shown that under budget constraints, LLMs [3,9] can match or exceed the performance of conventional HPO methods [11]. MM-LLMs, which are capable of interpreting visual data such as graphs, might also surpass these traditional methods and single-modal LLMs under similar constraints.

Additionally, this research evaluates MM-LLMs' skills in processing and understanding both images and textual data within the HPO context. With AI advancements enabling models to handle increasingly complex data types, MM-LLMs could significantly impact HPO [3, 7, 9]. Our experiments aim to add a new baseline for MM-LLMs and encourage further exploration into their applications.

In our experimental setup, we started by providing the model name, hyperparameters, and search space, asking both LLMs and MM-LLMs to recommend initial hyperparameters. After training the model with these settings and evaluating the results, we asked for further recommendations based on past performance. The key difference in MM-LLMs' setup was including a graph of the validation losses over time alongside the textual input, enhancing the multimodal data analysis. We also tested MM-LLMs' ability to perform without textual loss data, relying solely on graphical information, to see if they could still produce comparable results to LLMs.

Our findings suggest that within the confines of a limited budget (10 iterations), both LLMs and MM-LLMs outperform traditional methods like random search and Bayesian optimization in optimizing a Support Vector Machine (SVM). MM-LLMs particularly enhance performance by utilizing visual data of the validation losses, showing an impressive ability to process and integrate multimodal inputs.

## 2 Related work

The concept of using large language models (LLMs) to automate hyperparameter optimization (HPO) has already shown promise in making processes more efficient and reducing both human effort and computational costs. Zhang et al. [11] demonstrated this by applying LLMs to datasets from HPOBench [6], where they effectively selected hyperparameters for machine learning models, often surpassing traditional methods like random search and Bayesian optimization [1, 10] under limited budget conditions. Interestingly, they also treated the model code as a hyperparameter itself, allowing LLMs to directly generate training code, which is a departure from traditional methods that only tune pre-set hyperparameters.

Hyperparameter Optimization Techniques: HPO is vital in machine learning as it involves adjusting parameters that are not directly learned during training. Traditional methods, such as grid search, random search, and Bayesian optimization, have their limitations. Grid search, though thorough, is too resource-intensive for large datasets or complex hyperparameter spaces. Random search offers more efficiency but can miss the best settings due to its lack of precision. Bayesian optimization is more effective because it uses a probabilistic model to predict optimal regions, making it suitable for vast hyperparameter spaces [1,10]. Bischl [2] provides a more detailed introduction to these techniques.

Use of Large Language Models in HPO: Recent advancements have led to the use of foundational models like GPT-3 [3] in HPO. These models can predict effective hyperparameters by analyzing extensive data patterns and insights from diverse sources, shifting HPO towards more autonomous machine learning workflows. This is particularly beneficial in scenarios with large search spaces, where traditional methods might struggle due to computational limits [11].

Divergence into Multi-Modal Large Language Models (MM-LLMs): Unlike conventional LLMs that primarily handle text, MM-LLMs process various data types including images and structured data, offering a more comprehensive approach to problem-solving. In HPO, MM-LLMs can use this wider range of data to provide better-informed hyperparameter suggestions, addressing a major limitation of current LLMs—dependence solely on text and code [7].

Comparative Analysis: This study expands on prior research by using MM-LLMs to fill research gaps identified in earlier studies, which focused mainly on LLMs limited to text or code inputs. By utilizing MM-LLMs, our methodology enhances the ability of models to interpret complex and varied datasets, which could lead to more accurate and robust hyperparameter recommendations.

Theoretical and Practical Contributions: Theoretically, this research broadens the scope of LLM applications in HPO by introducing MM-LLMs, potentially setting a new direction for future research. Practically, the insights gained could make HPO processes more nuanced and efficient, saving time and resources, and possibly improving machine learning performance across various fields.

This section underscores our study's innovative approach with MM-LLMs and its expected impact, highlighting how it advances HPO practices and contributes both theoretically and practically to the field.

## 3 Data

HPOBench offers a broad collection of benchmarks for hyperparameter optimization (HPO) featuring various tasks with well-defined hyperparameter spaces and evaluation criteria. These datasets are publicly accessible through the OpenML AutoML benchmarks [6]. To align our research with prior work by Zhang et al. [11], we used datasets and algorithms that overlap with their study. Due to the high costs associated with using GPT-4 and budget constraints, we selected the bottom four datasets out of the eight they used—namely phoneme, segment, credit-g, and kc1 from HPOBench. These datasets are larger and have more features, ranging from 5 to 22, providing a varied scope for hyperparameter tuning. They were chosen for their comprehensive feature sets and complexity, making them ideal for testing the effectiveness of Multimodal Large Language Models (MM-LLMs) in HPO tasks.

In terms of machine learning models to optimize, besides the SVM model, we also included the XGBoost model, which was not explored by Zhang et al. We conducted HPO on these four datasets for both SVM and XGBoost using a mix of Random Search, Bayesian optimization, and both traditional and multimodal LLMs (GPT-4 Turbo and GPT-4 Vision). For

SVM, we tuned two hyperparameters and four for XGBoost.

# 4 Model

This research investigates how the Multimodal Large Language Model (MM-LLM) GPT-4V (GPT-4 with Vision) is used for Hyperparameter Optimization (HPO). GPT-4V, developed by OpenAI, mixes the text-handling abilities of the language model GPT-4 with the skill to analyze visual data like images and diagrams.

GPT-4V Architecture: GPT-4V builds on the GPT-4 language model, which is a transformer-based neural network trained extensively on a large collection of text data through self-supervised learning. It follows the standard encoder-decoder setup where the encoder processes input and the decoder outputs it sequence by sequence. For handling images, GPT-4V includes a vision encoder that uses convolutional neural network (CNN) technology to pull out visual features from images. These are then merged with text data, allowing the model to work with both types of information simultaneously.

Multimodal Training: After initial training, GPT-4V was fine-tuned with text-image pairs to better understand the link between textual and visual data. This was enhanced by Reinforcement Learning from Human Feedback (RLHF), with human input helping to refine the model's responses. The training also included steps to boost safety and accuracy, like adding image data to text datasets and creating examples from risky prompts to train the model in handling sensitive content effectively.

Capabilities and Limitations: GPT-4V combines GPT-4's language abilities with new skills in visual understanding, making it ideal for tasks needing analysis of both text and images. However, like all AI models, GPT-4V has its limits. It may occasionally make errors, create misleading information, or reflect biases from its training data, which could lead to inappropriate or harmful outputs. OpenAI has tried to minimize these risks with several safety measures and checks.

In this study, we use GPT-4V's multimodal features to examine its effectiveness in HPO tasks, seeing if it can improve the process by using both text and visuals to make better decisions about model adjustments. We compare its performance with traditional HPO methods and other single-mode LLMs to understand the benefits and drawbacks of using visual data in HPO tasks.

# 5 Methods

The experimental approach, including descriptions of metrics, baseline models, etc. Details about hyperparameters, optimization choices, etc., are probably best given in appendices, unless they are central to the arguments.

## 5.1 Experimental Setup

The experimental section of this study delineates the methodologies employed to assess the efficacy of various hyperparameter optimization (HPO) techniques on machine learning models, specifically focusing on XGBoost and Support Vector Machine (SVM) models facilitated by HPOBench. We detail the setup and execution of experiments designed to compare the performance of different optimization algorithms, including both traditional approaches like Bayesian Optimization and innovative methods utilizing Large Language Models (LLMs) and vision-based optimizers. Each optimizer interacts with a systematically defined search space to propose hyperparameters, which are then evaluated using specific machine learning tasks. This section outlines the experimental configurations, describes the optimizers in detail, and explains how the results are collected, evaluated, and statistically analyzed to deduce the effectiveness and efficiency of each method. Through these experiments, we aim to identify which optimization strategies yield the best performance in terms of accuracy and computational efficiency, thereby providing insights into the strengths and limitations of each approach within a controlled and reproducible experimental environment.

## 5.2 Models and Frameworks

This study uses two primary models, XGBoost and Support Vector Machine (SVM), implemented through the HPOBench framework. These models were chosen due to their widespread use and distinct learning characteristics, providing a comprehensive basis for assessing hyperparameter optimization strategies.

## 5.3 Benchmarker Configuration

The Benchmarker class is crucial for interfacing with HPOBench and managing the experimental setup for different tasks. It supports dynamic selection of models (XGBoost and SVM) and the configuration of task-specific benchmarks. For each model, a specific fidelity setup is defined:

- XGBoost: Configured with 2000 estimators and a subsample rate of 1.
- SVM: Operates with a subsample rate of 1 to ensure consistency in data sampling across different runs.

The fidelity settings ensure that each model's evaluation is both comprehensive and computationally feasible, reflecting realistic scenarios where both training time and model performance are crucial.

## 5.4 Search Space Configuration

The search space for each model is dynamically generated based on the model type, ensuring that each parameter is tuned within an appropriate range. This setup includes bounds for key parameters such as gamma and C for SVM, and max_depth and learning_rate for XGBoost, which are crucial for model performance.

## 5.5 Optimization Algorithms

### 5.5.1 Base Optimizer Structure

The BaseOptimizer serves as an abstract base class, providing a common interface for different optimization strategies. It handles basic functionalities like maintaining a history of evaluations and generating validation loss plots, which are vital for analyzing the optimization process over iterations.

### 5.5.2 Specific Optimizers

- Random Search: Provides a baseline by sampling configurations uniformly at random within the predefined search space.
- Bayesian Optimization: Utilizes a Gaussian Process-based surrogate model to guide the search towards regions of the space expected to yield improvements in validation loss.
- LLM Optimizer: Leverages a pretrained LLM (GPT-4-turbo) to predict effective hyperparameter configurations based on the historical performance data formatted as JSON inputs.
- Vision Optimizer: An advanced version of the LLM optimizer GPT-4-vision that also considers visual data from the validation loss plots to further inform the decision-making process of the LLM, potentially capturing insights that are not evident from numeric data alone.

## 5.6 Experimental Protocol

### 5.6.1 Data Collection and Preprocessing

Each model's performance is evaluated using datasets from the HPOBench, a benchmark suite that provides a variety of datasets with predefined hyperparameter spaces. This setup ensures that the results are comparable across different studies and optimization techniques.

### 5.6.2 Evaluation Metrics

The primary metrics for evaluating the performance of each optimizer are:

- Accuracy: Measured on a hold-out validation set to ensure that the optimized hyperparameters generalize well beyond the training data.
- Computational Efficiency: Assessed based on the time and computational resources required to reach a certain level of accuracy, providing insights into the practicality of each optimization method in real-world scenarios.

### 5.6.3 Iterative Optimization Process

Each optimizer is run for a fixed number of iterations (e.g., 10 trials), with each iteration involving:

- Generating a new set of hyperparameters.
- Evaluating the model's performance using these hyperparameters.
- Updating the optimizer's state and logging the history based on the results.

This iterative process allows for continuous refinement of the search strategy based on accumulating knowledge about the hyperparameter space's landscape.

### 5.7 Implementation Details

All experiments are conducted using a controlled environment to ensure that the results are reproducible and not influenced by external factors. The code for setting up the experiments, running the optimizers, and analyzing the results is documented and made available for review to ensure transparency and facilitate further research in the field.

This comprehensive experimental setup and methodology ensure that the findings of this study are both valid and applicable to a wide range of real-world scenarios where hyperparameter optimization plays a critical role in machine learning model performance.

#### 5.7.1 Prompt Design for Multimodal Large Language Models in HPO

The design of the prompts for the MM-LLMs in our study was guided by the need to effectively communicate the requirements of the HPO tasks while leveraging the multimodal capabilities of the models. Given that MM-LLMs can process and integrate information from both textual and visual data sources, our prompts were structured to include both types of data to simulate real-world machine learning tasks where data comes in diverse formats.

Prompt Structure: The prompts for the MM-LLMs were designed to contain the following elements:

1. Textual Description: Each prompt began with a concise description of the HPO task. This included the name of the machine learning model to be optimized, a brief description of the dataset, and the specific hyperparameters that needed tuning. This textual input served to orient the model towards the nature of the task and the context of the optimization.

2. Visual Data: To leverage the multimodal capabilities of the MM-LLMs, we included visual representations of historical performance data. These visuals, typically graphs of validation loss versus hyperparameter values over previous trials, provided a rich source of context that is often more intuitively digestible for humans. Including these in the prompts aimed to test whether MM-LLMs could similarly derive actionable insights from visual data, enhancing their decision-making process.

3. Dynamic Data Integration: As the optimization process progressed, the prompts were dynamically updated with new data from the latest evaluations. This design intended to mimic the iterative nature of HPO, where each decision is based on the accumulation of prior knowledge. The dynamic aspect of the prompts ensured that the MM-LLMs were continually updated with the most recent data, enhancing their ability to learn and adapt throughout the optimization process.

4. Multimodal Synergy: The prompts were specifically designed to test the synergy between textual and visual data inputs. By analyzing how MM-LLMs integrate these two data types, we could explore whether the multimodal approach provided a measurable improvement over single-modality data inputs in terms of the accuracy and efficiency of hyperparameter recommendations.

To implement these prompts, we used a controlled setup where the MM-LLMs received a standardized input format across all experiments. This consistency was crucial for ensuring that differences in model performance were attributable to the models' capabilities and not variations in prompt design. Each prompt was constructed using a template-based approach, where the textual data was formatted in JSON for clarity and consistency, and the visual data was embedded directly within the computational environment used for the experiments.

The inclusion of image plots aims to exploit the MM-LLMs' capability to process and integrate visual data, hypothesizing that this ability could lead to more informed and accurate hyperparameter settings by visual pattern recognition alongside textual data analysis.

The effectiveness of this prompt design was evaluated based on the MM-LLMs' ability to recommend hyperparameter settings that minimized validation loss. Additionally, we assessed the models' capability to interpret and utilize the visual data, comparing their performance against traditional LLMs that received only textual data.

Creation of Image Plots: The creation of image plots is a structured process tailored to ensure consistency and relevance to the optimization task:

1. Data Collection: Data for the plots is collected from each iteration of the model training, capturing key metrics such as validation loss, and the corresponding hyperparameter settings.
2. Plot Design: The plots are designed to clearly illustrate the relationship between hyperparameter configurations and their performance outcomes. Commonly, line graphs or scatter plots are used, with hyperparameters on one axis and performance metrics on the other.
3. Dynamic Updating: As the optimization progresses, the plots are dynamically updated with new data from each iteration. This ensures that the MM-LLMs receive the most current data, reflecting the latest state of the model training.
4. Integration into Prompts: The generated plots are embedded into the prompts as images alongside the textual description. This integration is handled programmatically to ensure that the image aligns correctly with the associated text, maintaining the coherence of the data presented to the MM-LLMs.

Prompt Example:

```
  You are helping tune hyperparameters for a XGBoost model. Here is what is tried so far:

  Config: {'colsample_bytree': 0.55, 'eta': 0.5, 'max_depth': 25, 'reg_lambda': 512.0}, Validation_loss: 0.13381294964028778
  Config: {'colsample_bytree': 0.8, 'eta': 0.1, 'max_depth': 20, 'reg_lambda': 256.0}, Validation_loss: 0.1553956834532374
  Config: {'colsample_bytree': 0.65, 'eta': 0.05, 'max_depth': 15, 'reg_lambda': 128.0}, Validation_loss: 0.13956834532374096

  Please provide the next configuration strictly in JSON format within the search space:
  {
    "colsample_bytree": [0.1,1.0],
    "eta": [0.0009765625,1.0],
    "max_depth": [1,50],
    "reg_lambda": [0.0009765625,1024.0]
  }
  The graph of validation loss versus hyperparameter values over previous trials is attached for your reference.
Config:
```


Figure 1

Example MM-LLM Response:


```
  {'id':<id>, 'model': 'gpt-4-turbo-2024-04-09', 'choices': [{'index': 0, 'message': {'role': 'assistant', 'content': '```json\n{\n "colsample_bytree": 0.7,\n "eta": 0.05,\n "max_depth": 16,\n "reg_lambda": 512.0\n}\n```'}, 'logprobs': None, 'finish_reason': 'stop'}], 'usage': {'prompt_tokens': 1005, 'completion_tokens': 45, 'total_tokens': 1050}, 'system_fingerprint':<fp>}
```


Parsed response from GPT4 after post-processing:

```
  {
  "colsample_bytree": 0.7,
  "eta": 0.05,
  "max_depth": 16,
  "reg_lambda": 512.0
  }
```

# 6 Results

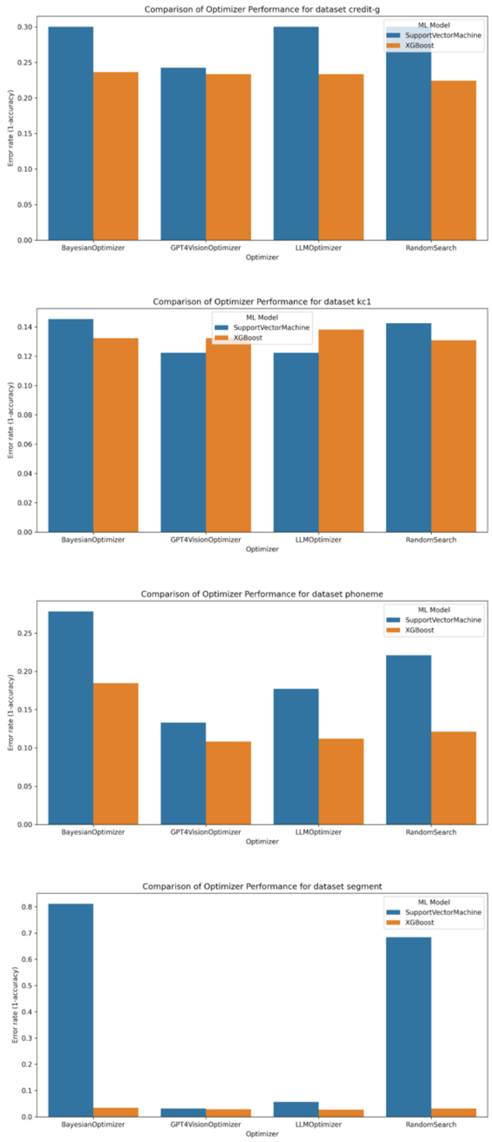


Figure 2

From the Error rate plots in Figure 2, we can see for model SVM, LLMs and MM-LLMs outperform Random search and Bayesian Optimization across all 4 datasets, especially for dataset segment, the difference is significant. In contrast, XGBoost as a robust model, after 10 iterations, the best set of hyperparameters selected by different optimizers do not vary dramatically.

From figure 2, we observe in the task of optimizing SVM, MM-LLMs outperform LLMs across all 4 datasets. Although the enhancement is not significant, it is consistent. Hence, the visualization provided to MM-LLMs helps to enhance HPO performance.

Furthermore, to assess the information comprehension through multimodal input, we conducted a different experiment for MM-LLMs: instead of providing validation loss log in both of text and image, we removed validation loss log from text input, only keeping the iteration number and the configuration history in text. The validation loss of each iteration is only presented through the plot. The aim of this setting is to see if MM-LLMs can understand the plot and connect the pieces of information from the text and the image together to form an overall picture. The results are rather promising, we ran this setup for all 4 datasets, the results are comparable with single modal LLMs fed with loss log in text format. For model SVM, it still outperforms traditional HPO approaches. Performance of one of the datasets is in Figure 3.

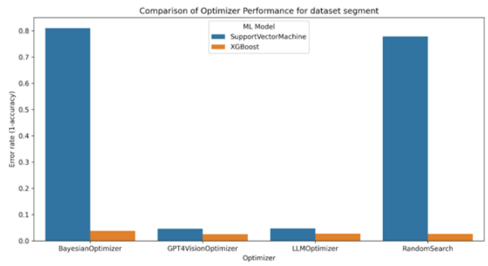


Figure 3

Refer to the code https://github.com/roshansridhar/mm-llm-hpo for all the specific configurations.

# 7 Conclusion

In conclusion, the use of large language models represents a groundbreaking shift in hyperparameter optimization practices. This study highlights the potential of multimodal LLMs to automate and enhance the efficiency and effectiveness of HPO process, marking a significant step toward more intelligent and autonomous machine learning systems. It also shows exceptional ability of MM-LLMs to intake data from multimodal input and comprehend them. By leveraging multimodal data, these models open new avenues for automating complex machine learning workflows, potentially reducing the need for human intervention, and allowing for more scalable solutions. Future advancements in this area will focus on expanding the types of data processed by MM-LLMs and exploring their applicability in other domains of machine learning and hold the promise of fully automated machine learning pipelines, where manual tuning becomes a task of the past.

## Known Project Limitations

While LLMs show promising results in automating HPO, they are not without limitations. We discuss the computational costs associated with deploying LLMs, potential biases stemming from the model's training data, and the risk of overfitting to validation data. Future research directions might include developing more computationally efficient LLM architectures or novel approaches to reduce operational costs. Moreover, enhancing model transparency and addressing data biases present further areas for research, aiming to make LLM-based HPO not only more effective but also more accessible to practitioners in the field.

## Authorship Statement

Longjiao Zhang: Focused on benchmarking, specifically in managing and setting up the two models, Support Vector Machine (SVM) and XGBoost, across four datasets from HPOBench. Longjiao ensured the experiments were robust by handling the model configurations and performance assessments.

Roshan Sridhar: Took the lead in developing and implementing all four optimization methods: Bayesian Optimization, Random Search, Large Language Models (LLMs), and Multimodal Large Language Models (MM-LLMs). Roshan's involvement was key in setting up the frameworks for the experiments and in analyzing the results from each optimizer.

Joint Efforts: Both authors worked closely together throughout the design, data collection, and analysis stages of the experiments. We teamed up to carry out the experiments, refine our methods based on initial findings, and write the paper together. This collaborative process involved blending the insights from our individual tasks into a unified study.

## References

[1] Bergstra, J., & Bengio, Y. (2012). *Random search for hyper-parameter optimization.* Journal of Machine Learning Research, 13(1), 281-305.

[2] Bernd Bischl, Martin Binder, Michel Lang, Tobias Pielok, Jakob Richter, Stefan Coors, Janek Thomas, Theresa Ullmann, Marc Becker, Anne-Laure Boulesteix, Difan Deng, and Marius Lindauer. 2021. *"Hyperparameter Optimization: Foundations, Algorithms, Best Practices and Open Challenges."* arXiv preprint arXiv:2107.05847.

[3] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... & Amodei, D. (2020). *Language models are few-shot learners.* arXiv preprint arXiv:2005.14165.

[4] Chen, T., & Guestrin, C. (2016). *Xgboost: A scalable tree boosting system.* In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794).

[5] Cortes, C., & Vapnik, V. (1995). *Support-vector networks.* Machine Learning, 20(3), 273-297.

[6] Eggensperger, K., Lindauer, M., Hoos, H. H., Hutter, F., & Leyton-Brown, K. (2019). *Efficient benchmarking of algorithm configurators via model-based surrogates.* Machine Learning, 108(9), 1567-1589.

[7] OpenAI, *"GPT-4V(ision) System Card"* https://openai.com/research/gpt-4v-system-card, 2023.

[8] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., ... & Sutskever, I. (2021). *Learning transferable visual models from natural language supervision.* arXiv preprint arXiv:2103.00020.

[9] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). *Language models are unsupervised multitask learners.* OpenAI blog, 1(8), 9.

[10] Snoek, J., Larochelle, H., & Adams, R. P. (2012). *Practical bayesian optimization of machine learning algorithms.* Advances in neural information processing systems, 25.

[11] Zhang, M., Nishkrit Desai, Juhan Bae, Jonathan Lorraine, and Jimmy Ba. 2023. *"Using Large Language Models for Hyperparameter Optimization."* In NeurIPS 2023 Foundation Models for Decision Making Workshop. URL https://openreview.net/forum?id=FUdZ6HEOre