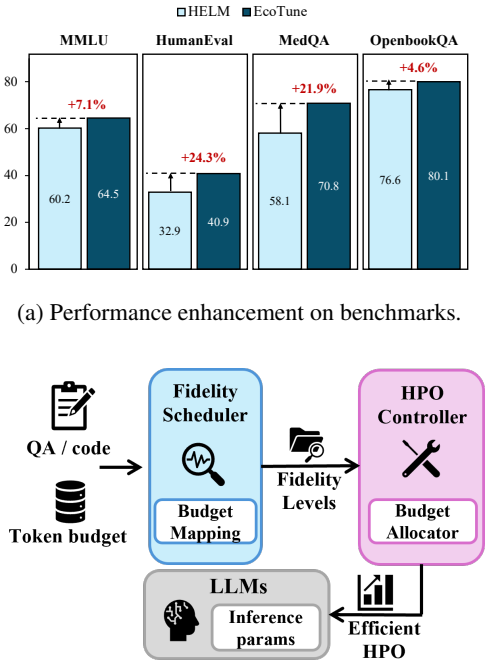
EcoTune: Token-Efficient Multi-Fidelity Hyperparameter Optimization for Large Language Model Inference

Yuebin Xu[1], **Zhiyi Chen**[1], **Zeyi Wen**[1,2*]
[1]HKUST (Guangzhou), [2]HKUST
{yxu349, zchen986}@hkust-gz.edu.cn, wenzeyi@ust.hk

Abstract

Tuning inference hyperparameters, such as temperature and maximum output tokens, on downstream tasks can enhance inference performance. However, directly applying hyperparameter optimization to these hyperparameters is token-expensive. Multi-fidelity optimization improves HPO efficiency with low-fidelity evaluations, but its static scheduling strategies ignore token consumption, leading to high costs. To address these limitations, we propose a token-efficient multi-fidelity optimization method, which enhances inference performance and minimizes token usage. Our method is empowered by (i) a token-based fidelity definition with explicit token cost modeling on configurations; (ii) a novel Token-Aware Expected Improvement acquisition function that selects configurations based on performance gain per token; and (iii) a dynamic fidelity scheduling mechanism that adapts to real-time budget status. We evaluate our method on LLaMA-2 and LLaMA-3 series across *MMLU*, *Humaneval*, *MedQA*, and *OpenBookQA*. Our method improves over the HELM leaderboard by 7.1%, 24.3%, 21.9%, and 4.6%, respectively. Compared to existing multi-fidelity HPO baselines, our method reduces token consumption by over 80% while maintaining or surpassing performance, demonstrating the state-of-the-art token efficiency for inference-time optimization.



(a) Performance enhancement on benchmarks.

(b) System overview.

Figure 1: (a) Performance comparison across *MMLU*, *Humaneval*, *MedQA*, and *OpenbookQA*; (b) System overview of EcoTune.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities across a broad spectrum of natural language processing tasks, such as question answering and code generation (Minaee et al., 2024). Notable examples of such models include the GPT (Achiam et al., 2023; Hurst et al., 2024), LLaMA (Touvron et al., 2023; Grattafiori et al., 2024), and DeepSeek series (Liu et al., 2024; Guo et al., 2025). These advances have been driven by scaling up model size, diverse training corpora, and powerful generalization capabilities (Brown et al., 2020; Chowdhery et al., 2022; Touvron et al., 2023). However, enhancing the performance of LLMs through training or fine-tuning is expensive, requiring massive computational resources, large-scale labeled datasets, and extensive engineering efforts (Bommasani et al., 2021; Zhao et al., 2023).

As a result, growing attention has shifted toward improving model behavior during inference, where the choice of decoding hyperparameters plays a critical role (Saad-Falcon et al., 2025). Optimizing these parameters, such as temperature (Du et al., 2025) or max_tokens (Wang et al., 2023a), offers a lightweight yet effective way to boost performance without retraining. These parameters control the

*Corresponding author

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 7735–7745
November 4-9, 2025

randomness, diversity, and length of the generated text, affecting the balance between creativity and coherence in responses. For instance, a higher temperature value can lead to more diverse outputs, while a lower value tends to produce more deterministic results. Vendors like OpenAI (Floridi and Chiriatti, 2020) and Meta (Grattafiori et al., 2024) provide default configurations for their LLM APIs. For example, OpenAI recommends setting the temperature to 0.2 for code generation and 0.7 for creative writing (Achiam et al., 2023). However, these configurations can not generalize across tasks or domains. Empirical studies have shown that task-specific tuning of inference hyperparameters can lead to significant performance gains without retraining (Zhang et al., 2024; Wang et al., 2023a).

Although tuning inference hyperparameters can improve LLM performance, directly applying hyperparameter optimization is often expensive due to the high token cost of each trial. For example, evaluating 50 GPT-4 configurations on full MMLU (Hendrycks et al., 2020) can cost hundreds of dollars under current API pricing. Multi-fidelity optimization (Peherstorfer et al., 2018) provides a principled framework that improves the efficiency of hyperparameter search by performing evaluations at multiple fidelity levels, such as reduced data or shorter runs. However, existing approaches such as Successive Halving (Karnin et al., 2013) and Hyperband (Li et al., 2018) rely on fixed resource schedules that are not aligned with actual token consumption, making them difficult to calibrate and often inefficient for LLM inference.

In this paper, we address the above limitations by proposing EcoTune, a token-efficient multi-fidelity optimization method for inference-time hyperparameter tuning. EcoTune is designed to improve task performance while strictly minimizing token consumption under budget constraints. The core idea is to treat token usage as the primary cost metric, which in turn guides both configuration selection and fidelity scheduling throughout the process. Our main contributions are summarized as follows:

- We introduce EcoTune (Figure 1b), a novel token-aware multi-fidelity optimization framework that simultaneously improves inference performance and reduces token consumption.
- We formalize a token-based fidelity model and develop two key components: Token-Aware Expected Improvement, an acquisition function that maximizes expected performance gain per token, and Dynamic Fidelity Scheduling, an adaptive mechanism that allocates fidelity levels based on current usage and remaining budget.
- We validate EcoTune on *LLaMA-2* and *LLaMA-3* across *MMLU*, *Humaneval*, *MedQA*, and *OpenBookQA*, showing performance improvements of 7.1%, 24.3%, 21.9%, and 4.6% over HELM-recommended configurations (Figure 1a), while reducing token consumption by over 80% compared to existing multi-fidelity HPO baselines.

2 Related Work

Enhancing LLMs at Inference Time. A wide range of approaches have been proposed to improve LLM performance during inference. Input-level strategies include prompt engineering (White et al., 2023) and retrieval-augmented generation (Zhao et al., 2024), while output-level methods such as self-consistency (Wang et al., 2023b) and model soups (Wortsman et al., 2022). More recently, research has gradually shifted toward decoding-level adaptations (Shi et al., 2024), particularly the careful tuning of inference hyperparameters such as temperature (Ackley et al., 1985; Renze, 2024; Zhang et al., 2024), top-p (Holtzman et al., 2019), and max_tokens (Wang et al., 2023a), which directly affect the quality and efficiency of generated outputs. However, traditional hyperparameter optimization methods, including Bayesian optimization and evolutionary strategies (Loshchilov and Hutter, 2016), often incur prohibitively high token costs due to repeated full-scale evaluations, highlighting the urgent need for more cost-efficient approaches tailored to LLM inference.

Inference Hyperparameters in LLMs. The behavior of LLMs during inference is highly sensitive to decoding hyperparameters such as temperature, top-p, and max_tokens (Arora et al., 2024). These parameters govern the randomness, diversity, and length of generated outputs, thereby shaping task-specific performance. Recent studies have shown that careful tuning of such hyperparameters can substantially improve results compared to the default configurations (Zhang et al., 2024; Wang et al., 2023a). For example, problem-solving benchmarks are particularly sensitive to temperature adjustments (Renze, 2024), while joint optimization strategies like Multi-Sample have been introduced to handle challenging tasks such as MATH (Du et al., 2025). Beyond standard parameters, new controls such as *min-p* have also been proposed to

more precisely regulate generation quality (Nguyen et al., 2024). Collectively, these findings underscore the central role of inference hyperparameters in optimizing LLM performance.

Multi-Fidelity Optimization. Multi-fidelity optimization has become a powerful paradigm for resource-efficient search for computationally expensive tasks in automated machine learning (Peherstorfer et al., 2018). By exploiting hierarchical approximations of the objective function, these methods accelerate optimization while retaining theoretical convergence guarantees (Forrester et al., 2007). Their effectiveness is exemplified by early-stopping strategies such as successive halving (Karnin et al., 2013) and Hyperband (Li et al., 2018), which allocate resources adaptively by quickly discarding underperforming candidates. More recent advances integrate multi-fidelity principles into Bayesian optimization (Wu et al., 2020), achieving state-of-the-art results in diverse AutoML tasks. Building on this foundation, we extend adaptive fidelity allocation to LLM inference, where fidelity is dynamically scheduled according to token consumption.

3 Methodology

In this section, we propose EcoTune, a token-constrained multi-fidelity hyperparameter optimization method tailored for LLM inference. Our method treats token consumption as a primary resource constraint and integrates fidelity control with budget-aware optimization to explore the hyperparameter space efficiently.

3.1 Problem Formulation

Given a hyperparameter search space Θ, our objective is to find the optimal configuration $\theta^* \in \Theta$ that maximizes a performance metric $f(\theta)$ (e.g., accuracy), subject to a total token budget B. The constrained optimization problem is defined as:

$$\max_{\theta \in \Theta} f(\theta), \quad \text{s.t.} \sum_{i=1}^{N} \mathcal{T}(\theta_i, r_i) \leq B \tag{1}$$

Here, $\mathcal{T}(\theta, r)$ denotes the token consumption of configuration θ evaluated at fidelity level r, which reflects the inference cost.

Multi-Fidelity Resource Mapping. To enable cost-efficient evaluation, we define a continuous fidelity space $r \in [r_{\min}, r_{\max}]$, where r controls evaluation granularity (e.g., the number of instances or decoding length). The token cost associated with evaluating configuration θ at fidelity r is modeled as:

$$\mathcal{T}(\theta, r) = \alpha(\theta) \cdot r + \beta(\theta) \tag{2}$$

In this formulation, $\alpha(\theta)$ captures the per-unit token cost (e.g., output length per instance), while $\beta(\theta)$ accounts for fixed overheads (e.g., prompt or input encoding). This linear model provides a tractable approximation for balancing performance and resource consumption across fidelity levels.

3.2 Framework Overview

As illustrated in Figure 1, our method operates in an iterative cycle that integrates configuration selection, fidelity control, and budget management. Each round begins with a token-aware acquisition function (Section 3.3) that balances expected performance improvement against token cost. Selected configurations are then evaluated at dynamically assigned fidelity levels (Section 3.4), enabling cost-efficient exploration under different granularities. Their outcomes update the surrogate model, and the remaining budget is adaptively reallocated (Section 3.5) to emphasize promising candidates while maintaining exploration of uncertain regions. This closed-loop process enforces strict token constraints, maximizes information gain per token, and underlies the efficiency and convergence properties discussed in Section 3.6. The overall procedure is summarized in Algorithm 1.

3.3 Token-Efficient Fidelity Allocation

Our allocation strategy is designed to select fidelity levels r that maximize the performance gain per token consumed. To this end, we estimate the marginal utility of increasing fidelity under a fixed token budget B. Given the initial reduction factor η and the average input/output token lengths, measured using a sliding window over the dataset, we allocate trials across multiple fidelity levels accordingly at the start of the optimization process.

Token-Aware Expected Improvement. We extend the classical Expected Improvement (EI) acquisition function (Močkus, 1975) to explicitly account for token consumption at different fidelity levels. For a configuration–fidelity pair (θ, r), the token-aware acquisition function is defined as

$$\alpha_{\text{EIpT}}(\theta, r) = \frac{\mathbb{E}\left[\left(f(\theta, r) - f^\star\right)^+\right]}{\mathbb{E}[\mathcal{T}(\theta, r)]}, \tag{3}$$

where $f^\star$ denotes the incumbent best performance observed at the maximum fidelity $r_{\max}$, and $\mathcal{T}(\theta, r)$ is the expected token cost (Eq. 2). Configurations are first ranked by their maximal score across fidelity levels, and the top-ranked configuration θ_t is selected for evaluation. This formulation naturally balances exploration and exploitation: when token costs are very small, the denominator $\mathbb{E}[\mathcal{T}(\theta, r)]$ amplifies the value of even modest performance gains, encouraging the evaluation of inexpensive configurations and thereby promoting exploration. Conversely, configurations with high token costs are only favored when they promise substantial improvements, ensuring budget efficiency.

3.4 Dynamic Fidelity Assignment

For the chosen configuration θ_t, the fidelity level r_t is determined using a budget-aware scheduling principle:

$$r_t = \begin{cases} r_{\max}, & \text{if } \alpha_{\text{EIpT}}(\theta_t, r_{\max}) > \tau, \\ \arg\max_{r \in \mathcal{R}} \alpha_{\text{EIpT}}(\theta_t, r), & \text{otherwise.} \end{cases} \tag{4}$$

Here, τ is a promotion threshold that triggers early evaluation at maximum fidelity when the EI-per-token at $r_{\max}$ is sufficiently high. Otherwise, the fidelity that maximizes cost-efficiency is selected. This formulation ensures that both configuration choice and fidelity scheduling are consistently guided by the same acquisition criterion.

3.5 Dynamic Budget Reallocation

At each iteration, the selected configuration θ_t is evaluated at its assigned fidelity level r_t, and the corresponding token cost is recorded. To monitor budget usage, we maintain cumulative statistics:

$$B_{\text{used}} = \sum_{i=1}^{t} \mathcal{T}(\theta_i, r_i), \quad B_{\text{remain}} = B - B_{\text{used}}. \tag{5}$$

Here, B_{used} denotes the total number of tokens consumed up to iteration t, while B_{remain} represents the remaining budget available for subsequent trials.

Surrogate Model Update. The multi-fidelity surrogate model (e.g., a Gaussian Process) is incrementally and consistently updated with the newly collected observation data $\mathcal{D}_t$:

$$\mathcal{D}_{t+1} = \mathcal{D}_t \cup \{(\theta_t, r_t, f(\theta_t, r_t), \mathcal{T}(\theta_t, r_t))\} \tag{6}$$

where (θ_t, r_t) denotes the evaluated configuration–fidelity pair and $f(\theta_t, r_t)$ the observed performance. The surrogate is defined as:

$$f(\theta, r) \sim \mathcal{GP}\big(\mu(\theta, r),\, k\big((\theta, r), (\theta', r')\big)\big)\,, \tag{7}$$

with mean function $\mu(\cdot)$ and kernel $k(\cdot, \cdot)$ modeling correlations across both hyperparameter and fidelity dimensions. Model hyperparameters are learned by maximizing the marginal likelihood.

Budget Reallocation. To maximize the overall utility of the remaining token budget, resources are adaptively redistributed between exploitation and exploration. Configurations showing strong performance indicators are promoted to higher fidelities. In particular, two key planning parameters are updated to reflect the current budget status:

$$\hat{T} \leftarrow \left\lfloor \frac{B_{\text{remain}}}{\mathbb{E}[\mathcal{T}]} \right\rfloor \tag{8}$$

$$\eta_{t+1} = \eta_t \cdot \exp\left(-\lambda \cdot \frac{B_{\text{remain}}}{B}\right) \tag{9}$$

Here, $\hat{T}$ estimates the number of additional trials that can be executed given the expected per-trial token cost, and η_{t+1} adjusts the fidelity reduction factor to balance exploration and exploitation as the budget shrinks. The optimization terminates once the token budget is exhausted ($B_{\text{remain}} \leq 0$), the performance metric converges, or the maximum number of trials is reached.

3.6 Efficiency and Convergence Analysis

Given a total token budget B and the expected token cost per evaluation trial $\mathbb{E}[\mathcal{T}(\theta, r)]$ (Eq. 2), the number of trials that can be executed is approximately bounded by $T \approx B/\mathbb{E}[\mathcal{T}(\theta, r)]$. If the average computation time per token is c, the overall optimization runtime scales linearly with the total number of tokens consumed, i.e., Runtime $= \mathcal{O}(B \cdot c)$. This indicates that both computational cost and wall-clock runtime are primarily determined by the token budget. For convergence, let $f^*(B)$ denote the best performance achieved under budget B. Empirical evidence suggests that performance typically follows a diminishing-return trajectory as the budget increases, which can be approximated by $f^*(B) \approx f_{\text{opt}} - C/B^\gamma$, where f_{opt} is the asymptotic optimum under unlimited budget, C is a task-dependent constant, and $\gamma \in (0, 1)$ characterizes the convergence rate. A larger γ implies faster convergence with respect to the budget, whereas smaller values correspond to slower improvement even as additional tokens are consumed.

3.7 Algorithm of EcoTune

Algorithm 1 outlines EcoTune. At each iteration t, a configuration θ_t is proposed by maximizing the token-aware acquisition function (Eq. 3), which balances expected improvement against token cost. Its fidelity r_t is then scheduled using Eq. 4, either promoting candidates directly to $r_{\max}$ or assigning the most cost-efficient fidelity. The configuration is evaluated, and the observed performance and token cost update the surrogate model (Eq. 7) via marginal likelihood optimization. Budget statistics are tracked, and planning parameters $\hat{T}$ and η are adapted (Section 3.5). The procedure terminates when the token budget is exhausted, convergence is detected, or the maximum number of trials is reached, and returns the best configuration θ^*.

Algorithm 1: EcoTune

Input: Search space Θ, token budget B, promotion threshold τ, initial reduction factor η_0, dataset $\mathcal{D}$
Output: Best configuration θ^*
Initialize GP with empty data $\mathcal{D}_0 \leftarrow \emptyset$;
$B_{\text{used}} \leftarrow 0$, $\eta \leftarrow \eta_0$, $t \leftarrow 0$, $f^* \leftarrow -\infty$;
while $B_{used} < B$ ***and*** *not converged* **do**
  $\theta_t \leftarrow \arg\max_{\theta \in \Theta} \alpha_{\text{EIpT}}(\theta, r)$;
  `// token-aware EI (Eq. 3)`
  if $\alpha_{EIpT}(\theta_t, r_{\max}) > \tau$ **then**
    $r_t \leftarrow r_{\max}$ `// early promotion`
  else
    $r_t \leftarrow \arg\max_{r \in [r_{\min}, r_{\max}]} \alpha_{\text{EIpT}}(\theta_t, r)$
  $s \leftarrow f(\theta_t, r_t)$;
  $\mathcal{T}_t \leftarrow \mathcal{T}(\theta_t, r_t)$;
  $B_{\text{used}} \leftarrow B_{\text{used}} + \mathcal{T}_t$;
  $\mathcal{D}_{t+1} \leftarrow \mathcal{D}_t \cup \{(\theta_t, r_t, s, \mathcal{T}_t)\}$;
  Update GP surrogate;
  $B_{\text{remain}} \leftarrow B - B_{\text{used}}$;
  $\hat{T} \leftarrow \lfloor B_{\text{remain}} / \mathbb{E}[\mathcal{T}] \rfloor$;
  $\eta \leftarrow \eta \cdot \exp(-\lambda \cdot B_{\text{remain}}/B)$;
  if $s > f^*$ **then**
    $f^* \leftarrow s$;
  $t \leftarrow t + 1$;
return $\theta^* = \arg\max_{(\theta_i, r_i) \in \mathcal{D}} f(\theta_i, r_i)$

4 Experiment

In this section, we elaborate on our experimental studies designed to validate the effectiveness and token efficiency of EcoTune. We benchmark our method across diverse tasks and datasets, comparing against existing hyperparameter optimization baselines under controlled token budgets.

4.1 Experimental Setup

Datasets and Metrics. We evaluate our method on four diverse benchmarks to assess different capabilities of language models. *MMLU* (Hendrycks et al., 2020) (Apache License 2.0) covers 57 academic subjects with multiple-choice questions, using Exact Match (EM) for evaluation. *Humaneval* (Chen et al., 2021) (MIT License) includes 164 code generation problems, assessed by Pass@1. *MedQA* (Jin et al., 2021) (Available for research use only) is a medical QA dataset from professional exams, evaluated by EM. *OpenBookQA* (Mihaylov et al., 2018) (CC BY-SA 4.0) tests multi-hop reasoning with 5,957 science questions, also using EM. These tasks span across general knowledge, code generation, domain-specific QA, and reasoning, enabling the comprehensive evaluation.

Baselines and HPO Settings. We compare EcoTune against a broad suite of hyperparameter optimization baselines spanning different algorithmic families:(i) *single-fidelity methods* include Random Search (RS), Bayesian Optimization (BoTorch) (Balandat et al., 2020), quasi-Monte Carlo sampling (QMC) (Caflisch, 1998; Akiba et al., 2019), and CMA-ES (Nomura and Shibata, 2024);(ii) *multi-fidelity* methods are represented by BOHB (Falkner et al., 2018), which reduces evaluation cost through early-stopping. In addition, we report results from the **HELM leaderboard** (Liang et al., 2023)[1], which provides vendor-recommended configurations on continuously updated benchmarks. All methods are evaluated under a strict 50K-token budget per dataset, accounting for both input and output tokens. The hyperparameter search space is aligned with practical decoding ranges: temperature is varied within [0.0, 1.0], max_tokens from 1 (for multiple-choice tasks such as *MMLU*) up to 600 (for code generation tasks like *Humaneval*), repetition penalty in [1.0, 1.5], and length penalty in [0.8, 1.2]. Unless otherwise noted, all hyperparameters are uniformly sampled within these ranges.

Implementation Details. For evaluation, we adopt prompting strategies consistent with HELM. On *MMLU*, *MedQA*, and *OpenBookQA*, we use

[1] https://crfm.stanford.edu/helm/

	MMLU		Humaneval		MedQA		OpenBookQA	
	LLaMA-2-7B	LLaMA-3-8B	LLaMA-2-7B	LLaMA-3-8B	LLaMA-2-7B	LLaMA-3-8B	LLaMA-2-7B	LLaMA-3-8B
HELM (Liang et al., 2023)	42.6	60.2	13.4	32.9	39.2	58.1	50.1	76.6
Random	42.4	60.1	8.5	25.6	41.0	67.1	52.7	75.4
BoTorch	43.8	61.5	12.2	28.3	44.0	67.1	46.0	76.7
BOHB	45.8	60.8	12.6	32.9	35.0	68.3	54.0	76.8
QMC	44.4	61.3	16.2	34.8	43.1	65.8	53.3	77.3
CMA-ES	44.8	59.9	13.4	36.6	43.0	64.6	55.3	78.7
EcoTune (Ours)	**48.1**	**64.5**	**16.5**	**40.9**	**44.2**	**70.8**	**56.2**	**80.1**

Table 1: Comparison of performance across various HPO methods on LLaMA-2-7B and LLaMA-3-8B models across *MMLU*, *Humaneval*, *MedQA*, and *OpenBookQA*.

5-shot prompting with top-k sampling (k=5) in a Question/Answer format (Liang et al., 2023). On *Humaneval*, we use zero-shot prompting with a 600-token output limit, stopping generation when encountering reserved tokens such as `class`, `def`, `if`, or `print`. All evaluations run through local APIs with exact token accounting, and optimization continues until the token budget is fully consumed. BOHB uses `min_resource` = 0.01 and `reduction_factor` = 4 for aggressive early stopping. CMA-ES sets its hyperparameters by default.

4.2 Main Performance Comparison

Table 1 compares EcoTune with a diverse set of hyperparameter optimization baselines across four representative tasks and two backbone models, LLaMA-2-7B and LLaMA-3-8B. All the methods operate under a strict token budget per dataset, accounting for both input and output token usage.

Across all the tasks and model scales, EcoTune consistently achieves the best overall performance, demonstrating strong effectiveness under token constraints. On the knowledge-intensive task *MMLU*, EcoTune attains 48.1 and 64.5 accuracy on LLaMA-2 and LLaMA-3, respectively, surpassing the next-best baselines (BOHB and QMC) by 3.2 points. In the medical QA task *MedQA*, it reaches 70.8 EM on LLaMA-3, exhibiting superior stability. BoTorch performs reasonably under LLaMA-2 but degrades under LLaMA-3 due to poor handling of the fidelity-cost tradeoff. In contrast, EcoTune leverages adaptive fidelity and token estimation to conduct more informative evaluations.

On the reasoning benchmark *OpenBookQA*, EcoTune again delivers top results, achieving 56.2 and 80.1 EM on LLaMA-2 and LLaMA-3. Competing methods either plateau early (e.g., BOHB, QMC) or overcommit to expensive configurations, reducing the number of completed trials. On the code generation task *Humaneval*, where evaluation is particularly costly, EcoTune's adaptive fidelity mechanism and token-aware scoring yield 40.9 Pass@1 on LLaMA-3, surpassing CMA-ES by 4.3 points and the HELM default by 8.0 points.

While most baselines could theoretically converge given unlimited tokens, such evaluations are generally infeasible in real practice. Generation-heavy tasks are extremely expensive both in cost and runtime resources. For instance, exhaustively evaluating 2,000 configurations on *Humaneval* would easily require hundreds of dollars and several days. In contrast, EcoTune quickly identifies promising configurations early, ensuring robust and efficient optimization under realistic budgets.

In summary, EcoTune not only achieves higher performance but also exhibits strong robustness across models and tasks. Its dynamic resource allocation and token-aware design together make it particularly well-suited for budget-constrained inference-time optimization scenarios in practice.

4.3 Efficiency Analysis

Token Consumption Comparison. We compare the token efficiency of our proposed method against the existing multi-fidelity method, BOHB (Falkner et al., 2018), which integrates Bayesian optimization with the Hyperband technique (Li et al., 2018), one of the most widely used multi-fidelity methods. To quantify the efficiency gains, we measure the total token consumption required to reach convergence while achieving comparable performance on MMLU tasks. Since different tasks vary significantly in context length, we group them into three high-level domains: *natural sciences* including Algebra, Medicine, Physics, and Mathematics, *social sciences* including Economics, Global Facts, Sociology, and Miscellaneous, and *humanities* including Philosophy, Moral Disputes, World History,

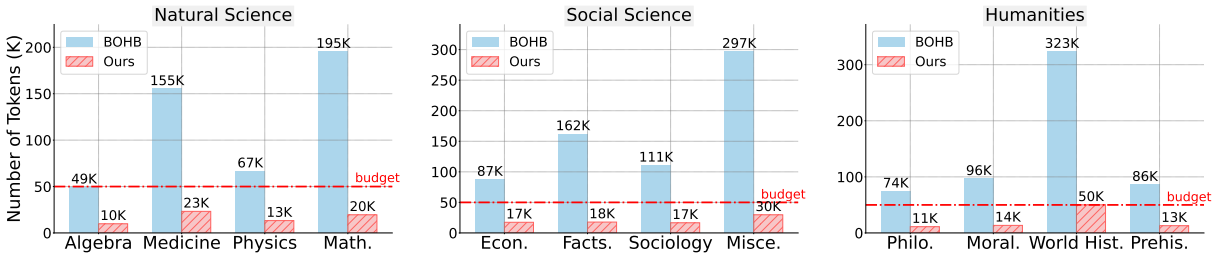


Figure 2: Token consumption comparison to achieve the same performance across different domains in MMLU.

Task	Inst.	Token (In/Out)	Thru. (In/Out)	Time/Inst. (s)	Speedup (×)
Abstract Algebra	116	370 / 1	1.6k / 4	0.55	5.0
Economics	131	620 / 1	3.0k / 4	0.48	5.9
World History	268	1.4k / 1	3.3k / 2	0.98	4.5
OpenBookQA	5.9k	42 / 4	3.8k / 13	0.36	4.3
MedQA	5k	10k / 1	15.3k / 13	0.20	3.8
HumanEval	164	170 / 74	540 / 327	0.59	4.8
Average	–	602 / 14	–	0.53	4.7

Table 2: Runtime statistics across tasks, including dataset scale, token lengths, throughput, and Speedup.

and Prehistory. Notably, tasks in the humanities tend to require longer contexts. The comparative results are presented in Figure 2.

Several key observations emerge from the results. First, our method consistently identifies effective configurations under strict token budgets and achieves substantial reductions in token usage, exceeding 80% savings compared to the baseline, highlighting its superior efficiency. Second, while BOHB exhibits near-linear growth in token consumption as the number of instances increases, our method maintains consistently low token usage. This stability indicates that our approach effectively selects and leverages representative instances, leading to robust efficiency regardless of dataset scale. Notably, the World History task incurs relatively higher token consumption due to the inherently long contextual inputs typical of historical content.

Wall-Clock Runtime Statistics. Beyond token-level analysis, Table 2 reports task-specific runtime characteristics, including dataset scale, token lengths, and throughput. These results demonstrate that EcoTune's token savings consistently translate into practical wall-clock improvements, achieving an average $4.7\times$ speedup over BOHB across diverse tasks. Notably, tasks with longer prompts (e.g., *MedQA*, *World History*) benefit more from input-side efficiency, while generation-heavy tasks such as *Humaneval* gain from output-side optimization. This confirms that EcoTune's design yields consistent efficiency advantages in both compute- and memory-bound inference regimes.

Convergence Behavior under Token Constraints. We evaluate convergence on the STEM subset of MMLU using the Empirical Cumulative Distribution Function (ECDF) (Van der Vaart, 2000), which captures detailed validation error dynamics under equal token budgets. As shown in Figure 3, EcoTune consistently dominates the top-left region, achieving lower errors with fewer evaluations. The denser ECDF curve further indicates more effective and systematic exploration, confirming both faster convergence and higher token efficiency. Compared to BOHB, EcoTune reaches competitive error levels with fewer trials, highlighting its practical advantage under strict resource constraints.

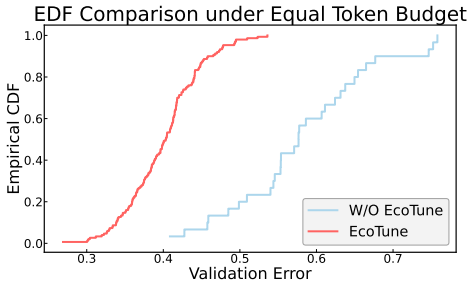


Figure 3: ECDF on Validation Error.

4.4 Combination of Decoding Strategies

Beyond fixed decoding settings such as temperature plus top-k, different stochastic sampling strategies can independently influence the diversity and quality of model outputs. To systematically assess robustness, we evaluate EcoTune on subsets of the MMLU dataset under four widely used decoding strategies: (i) pure temperature sampling ("Temp."), (ii) temperature combined with top-k sampling (k=5) (Fan et al., 2018), (iii) top-p sampling with p=0.95 (Holtzman et al., 2019), and (iv) min-p sampling with p=0.2 (Nguyen et al., 2024). These settings cover a representative spectrum from simple temperature control to nucleus- and probability-based sampling, enabling a comprehensive evaluation of decoding robustness.

	LLaMA-2-13B									LLaMA-2-70B								
	ANAT.	MATH.	PHY.	ECON.	LOGIC.	CHEM.	HIST.	JURIS.	**Avg.**	ANAT.	MATH.	PHY.	ECON.	LOGIC.	CHEM.	HIST.	JURIS.	**Avg.**
Temp. (HELM)	49.63	30.00	23.53	31.00	39.68	48.28	63.63	71.90	44.71	55.59	37.00	36.27	34.00	42.86	49.75	67.27	74.10	49.60
+ top-k	49.63	30.00	23.53	33.33	39.68	48.28	63.63	71.29	44.92	56.30	38.00	35.29	35.09	43.65	51.23	66.67	74.07	50.04
+ top-p	49.63	30.00	23.53	33.33	39.68	48.28	63.64	71.29	44.92	53.33	39.00	35.29	33.33	42.06	50.25	66.67	73.15	49.14
+ min-p	48.89	30.00	23.53	33.33	39.68	48.28	63.64	71.29	44.83	53.33	36.00	37.25	35.96	42.86	50.74	66.06	75.00	49.65
Temp. (Ours)	55.59	37.00	36.27	34.00	42.86	49.75	67.27	74.10	49.60	60.74	35.00	35.29	43.83	46.03	49.75	82.42	82.41	54.4
+ top-k	56.30	38.00	35.29	35.09	43.65	51.23	66.67	74.07	**50.04**	60.74	35.00	35.29	43.86	46.03	49.75	81.81	82.41	54.36
+ top-p	53.33	39.00	35.29	33.33	42.06	50.25	66.67	73.15	49.14	60.70	35.00	35.29	43.86	46.03	49.75	82.42	82.41	54.36
+ min-p	53.33	36.00	37.25	35.96	42.86	50.74	66.06	75.00	49.65	60.74	35.00	35.29	43.86	46.03	49.75	82.42	82.41	**54.43**

(a) LLaMA-2 series (13B and 70B).

	LLaMA-3-8B									LLaMA-3-70B								
	ANAT.	MATH.	PHY.	ECON.	LOGIC.	CHEM.	HIST.	JURIS.	**Avg.**	ANAT.	MATH.	PHY.	ECON.	LOGIC.	CHEM.	HIST.	JURIS.	**Avg.**
Temp. (HELM)	69.63	35.00	45.10	51.75	45.23	55.17	75.15	74.00	56.88	78.52	56.00	52.94	70.18	65.08	74.00	84.85	86.11	70.46
+ top-k	69.63	35.00	45.10	51.75	45.23	55.17	75.15	74.07	56.89	77.78	56.00	52.94	68.42	65.08	73.39	84.85	86.11	69.83
+ top-p	68.89	35.00	45.10	51.75	45.24	55.17	75.15	74.07	56.70	77.78	56.00	52.94	69.29	65.08	73.39	84.85	86.11	70.18
+ min-p	69.63	35.00	45.10	51.75	45.24	55.17	75.15	74.07	56.89	77.78	56.00	52.94	68.42	56.08	73.39	84.85	86.11	68.70
Temp. (Ours)	72.59	40.00	50.98	54.39	52.38	57.14	77.58	79.63	**60.59**	80.00	57.00	57.84	73.68	68.25	75.37	87.88	87.96	73.50
+ top-k	71.11	45.00	52.94	53.51	51.59	57.14	76.97	81.48	60.47	80.00	58.00	55.88	74.56	69.05	75.37	87.88	87.96	**74.09**
+ top-p	72.59	39.00	53.92	53.51	50.00	57.14	76.97	77.78	60.11	80.00	59.00	59.80	73.68	69.05	74.38	87.27	87.04	74.03
+ min-p	71.85	45.00	51.96	54.39	50.00	58.13	77.58	80.56	59.93	80.00	60.00	56.86	72.81	69.04	75.86	86.67	87.96	73.90

(b) LLaMA-3 series (8B and 70B).

Table 3: Ablation of mixed decoding strategy tuning across LLaMA series models.

As reported in Table 3, EcoTune consistently surpasses the HELM baseline across all stochastic decoding strategies, subject domains, and backbone models. On average, it improves overall accuracy by about 5.0 and 4.8 points on LLaMA-2-13B and 70B, and by 3.7 and 3.5 points on LLaMA-3-8B and 70B, respectively. Three observations stand out: (i) EcoTune often achieves the highest scores under top-k, top-p, or min-p, demonstrating robustness across diverse decoding strategies; (ii) performance gains are particularly pronounced in humanities and social sciences, where longer contexts and nuanced reasoning make decoding more challenging; and (iii) although improvements on STEM-oriented tasks are more modest, EcoTune still consistently outperforms the baseline, confirming its ability to generalize across domains.

5 Conclusion

In this paper, we propose EcoTune, a token-constrained multi-fidelity hyperparameter optimization method tailored for LLM inference. By leveraging the token-aware acquisition function and the dynamic fidelity adjustment mechanism, our approach effectively balances the performance and evaluation cost under strict token budgets. Extensive experiments on *MMLU*, *Humaneval*, *MedQA*, and *OpenBookQA* demonstrate that EcoTune achieves up to 24% higher accuracy while substantially reducing token consumption. It consistently delivers stable and reliable performance across different model sizes and task complexities. Comprehensive ablation and comparative studies further confirm the robustness and generality of our method across diverse benchmarks. Future work will explore incorporating a broader set of decoding hyperparameters and extending the method to account for latency and monetary cost constraints.

Limitation

EcoTune advances token-efficient hyperparameter optimization for LLM inference, yet its scope remains primarily at the algorithmic level. Broader system-level factors, such as including hardware constraints, are not fully considered, though integrating such dimensions could yield a more comprehensive and deployable optimization framework. Moreover, our evaluation is limited to a selected set of benchmarks, and further studies on more diverse tasks are needed to assess broader applicability.

Acknowledgement

This work is supported by National Key R&D Program of China under Grant No. 2024YFA1012700, and by the Guangzhou Industrial Information and Intelligent Key Laboratory Project (No. 2024A03J0628). It is also funded by the NSFC Project (No. 62306256) and the Natural Science Foundation of Guangdong Province (No. 2025A1515010261).

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

David H Ackley, Geoffrey E Hinton, and Terrence J Sejnowski. 1985. A learning algorithm for boltzmann machines. *Cognitive science*, 9(1):147–169.

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 2623–2631.

Chetan Arora, Ahnaf Ibn Sayeed, Sherlock Licorish, Fanyu Wang, and Christoph Treude. 2024. Optimizing large language model hyperparameters for code generation. *arXiv preprint arXiv:2408.10577*.

Maximilian Balandat, Brian Karrer, Daniel Jiang, Samuel Daulton, Ben Letham, Andrew G Wilson, and Eytan Bakshy. 2020. Botorch: A framework for efficient monte-carlo bayesian optimization. *Advances in neural information processing systems*, 33:21524–21538.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, and 1 others. 2021. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Russel E Caflisch. 1998. Monte carlo and quasi-monte carlo methods. *Acta numerica*, 7:1–49.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and 1 others. 2022. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*.

Weihua Du, Yiming Yang, and Sean Welleck. 2025. Optimizing temperature for language models with multi-sample inference. *arXiv preprint arXiv:2502.05234*.

Stefan Falkner, Aaron Klein, and Frank Hutter. 2018. Bohb: Robust and efficient hyperparameter optimization at scale. In *International conference on machine learning*, pages 1437–1446. PMLR.

Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833*.

Luciano Floridi and Massimo Chiriatti. 2020. Gpt-3: Its nature, scope, limits, and consequences. *Minds and Machines*, 30:681–694.

Alexander IJ Forrester, András Sóbester, and Andy J Keane. 2007. Multi-fidelity optimization via surrogate modelling. *Proceedings of the royal society a: mathematical, physical and engineering sciences*, 463(2088):3251–3269.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Dan Hendrycks and 1 others. 2020. Measuring massive multitask language understanding. *arXiv:2009.03300*.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Zohar Karnin, Tomer Koren, and Oren Somekh. 2013. Almost optimal exploration in multi-armed bandits. In *International conference on machine learning*, pages 1238–1246. PMLR.

Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. 2018. Hyperband: A novel bandit-based approach to hyperparameter optimization. *Journal of Machine Learning Research*, 18(185):1–52.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian

Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, and 1 others. 2023. Holistic evaluation of language models. *Transactions on Machine Learning Research*.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Ilya Loshchilov and Frank Hutter. 2016. Cma-es for hyperparameter optimization of deep neural networks. *arXiv preprint arXiv:1604.07269*.

Todor Mihaylov and 1 others. 2018. Openbookqa: Factual knowledge assessment in question answering. *EMNLP*.

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

Jonas Močkus. 1975. On bayesian methods for seeking the extremum. In *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974 6*, pages 400–404. Springer.

Minh Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2024. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *arXiv preprint arXiv:2407.01082*.

Masahiro Nomura and Masashi Shibata. 2024. cmaes: A simple yet practical python library for cma-es. *arXiv preprint arXiv:2402.01373*.

Benjamin Peherstorfer, Karen Willcox, and Max Gunzburger. 2018. Survey of multifidelity methods in uncertainty propagation, inference, and optimization. *Siam Review*, 60(3):550–591.

Matthew Renze. 2024. The effect of sampling temperature on problem solving in large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7346–7356.

Jon Saad-Falcon, Adrian Gamarra Lafuente, Shlok Natarajan, Nahum Maru, Hristo Todorov, Etash Kumar Guha, E Kelly Buchanan, Mayee F Chen, Neel Guha, Christopher Re, and 1 others. 2025. An architecture search framework for inference-time techniques. In *Forty-second International Conference on Machine Learning*.

Chufan Shi, Haoran Yang, Deng Cai, Zhisong Zhang, Yifan Wang, Yujiu Yang, and Wai Lam. 2024. A thorough examination of decoding methods in the era of llms. *arXiv preprint arXiv:2402.06925*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Aad W Van der Vaart. 2000. *Asymptotic statistics*, volume 3. Cambridge university press.

Chi Wang, Xueqing Liu, and Ahmed Hassan Awadallah. 2023a. Cost-effective hyperparameter optimization for large language model generation inference. In *International Conference on Automated Machine Learning*, pages 21–1. PMLR.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jules White, Quchen Fu, Sam Hays, Michael Sandborn, Carlos Olea, Henry Gilbert, Ashraf Elnashar, Jesse Spencer-Smith, and Douglas C Schmidt. 2023. A prompt pattern catalog to enhance prompt engineering with chatgpt. *arXiv preprint arXiv:2302.11382*.

Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, and 1 others. 2022. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *International conference on machine learning*, pages 23965–23998. PMLR.

Jian Wu, Saul Toscano-Palmerin, Peter I Frazier, and Andrew Gordon Wilson. 2020. Practical multi-fidelity bayesian optimization for hyperparameter tuning. In *Uncertainty in Artificial Intelligence*, pages 788–798. PMLR.

Shimao Zhang, Yu Bao, and Shujian Huang. 2024. Edt: Improving large language models' generation by entropy-based dynamic temperature sampling. *arXiv preprint arXiv:2403.14541*.

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Wayne Xin Zhao, Kun Zhang, Jing Liu, Junjie Wang, Yiqun Hou, Rui Wang, and Ji-Rong Wen. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

A Appendix

A.1 Main Experiment

Full experimental results across 57 sub-tasks of MMLU for LLaMA-2 (7B, 13B, 70B) and LLaMA-3 (8B, 70B) are presented in Table 4. For each model, we report the initial accuracy (Init) based on HELM-style evaluation, the accuracy after applying our tuning method, and the relative improvement ($\Delta\%$). Improvements exceeding 5% are highlighted in bold.

Subject	LLaMA-2-7B			LLaMA-2-13B			LLaMA-2-70B			LLaMA-3-8B			LLaMA-3-70B		
	Init	Imp	Δ%	Init	Imp	Δ%	Init	Imp	Δ%	Init	Imp	Δ%	Init	Imp	Δ%
Algebra	29.0	36.0	**+24.1**	27.0	43.0	**+59.3**	31.0	35.0	**+12.9**	33.0	41.0	**+24.2**	41.0	49.0	**+19.5**
Anatomy	45.0	49.0	**+8.2**	49.6	55.6	**+12.1**	60.7	67.4	**+21.5**	69.6	72.6	**+9.0**	78.5	80.0	+3.6
Astronomy	41.4	44.1	+4.5	56.6	58.6	+3.5	82.2	84.9	**+8.5**	71.1	72.4	+4.0	92.1	93.4	+3.2
BusEth	48.0	52.0	**+6.8**	53.0	60.0	**+13.2**	72.0	76.0	**+12.9**	66.0	69.0	**+9.1**	83.0	86.0	**+7.3**
ClinKnow	44.5	47.5	**+5.1**	60.0	61.1	+1.8	71.3	75.9	**+14.6**	75.1	76.6	+4.6	84.5	85.3	+1.8
Col Biology	42.4	46.5	**+7.0**	59.7	61.1	+2.3	84.7	85.4	+2.3	76.4	79.2	**+8.5**	94.4	95.1	+1.7
ColChem	27.0	39.0	**+44.4**	41.0	49.0	**+19.5**	50.0	55.0	**+16.1**	48.0	55.0	**+21.2**	57.0	62.0	**+12.2**
Col CS	39.0	43.0	**+6.8**	42.0	43.0	+2.4	59.0	61.0	**+6.5**	58.0	59.0	+3.0	70.0	74.0	**+9.8**
Col Math	29.0	40.0	**+37.9**	30.0	37.0	**+23.3**	35.0	43.0	**+25.8**	35.0	40.0	**+15.2**	56.0	57.0	+2.4
ColMed	46.0	48.6	**+5.5**	53.8	54.9	+2.2	65.3	68.8	**+11.0**	63.6	65.9	**+7.0**	79.8	80.9	+2.8
ColPhy	21.6	29.4	**+36.4**	23.5	36.3	**+54.5**	35.3	44.1	**+28.5**	45.1	51.0	**+17.8**	52.9	57.8	**+12.0**
CompSec	59.0	64.0	**+8.5**	68.0	71.0	**+4.4**	76.0	79.0	**+9.7**	80.0	81.0	+3.0	85.0	87.0	+4.9
ConPhys	42.8	45.1	**+5.4**	41.7	45.5	**+9.2**	66.4	68.5	**+6.9**	56.2	60.0	**+11.6**	83.8	84.3	+1.1
Econ	31.6	38.6	**+22.2**	31.0	34.0	**+11.4**	43.9	49.1	**+17.0**	51.8	54.4	**+8.0**	70.2	73.7	**+8.5**
EE	43.4	44.8	+3.2	49.0	51.7	**+5.6**	63.5	64.8	+4.5	66.9	70.3	**+10.4**	76.6	77.9	+3.4
ElemMath	25.9	29.1	**+12.3**	32.0	33.6	**+5.0**	41.8	46.0	**+13.7**	42.9	45.2	**+7.2**	63.0	65.6	**+6.5**
FormLog	27.8	33.3	**+19.8**	39.7	42.9	**+8.0**	46.0	52.4	**+20.5**	45.2	52.4	**+21.7**	65.1	68.3	**+7.7**
GlobalFacts	29.0	32.0	**+10.3**	38.0	42.0	**+10.5**	48.0	54.0	**+19.4**	33.0	40.0	**+21.2**	49.0	54.0	**+12.2**
HS Bio	51.3	53.6	+4.4	66.5	68.1	+2.4	81.6	83.9	**+7.3**	78.1	79.7	+4.9	90.0	91.6	+3.9
HS Chem	36.5	40.4	**+10.8**	48.3	49.8	+3.0	49.8	56.7	**+22.3**	55.2	57.1	**+6.0**	74.0	75.4	+3.3
HS CS	38.0	46.0	**+21.1**	59.0	62.0	**+5.1**	75.0	78.0	**+9.7**	67.0	70.0	**+9.1**	87.0	90.0	**+7.3**
HS EuroHist	62.4	64.2	+3.0	63.6	67.3	**+5.7**	82.4	84.2	**+5.9**	75.2	77.6	**+7.4**	84.9	87.9	**+7.4**
HS Geo	51.5	55.6	**+8.0**	71.7	72.7	+1.4	87.9	88.4	+1.7	82.8	84.9	**+6.1**	93.9	95.0	+2.5
HS Politics	67.9	70.0	+3.1	80.8	81.9	+1.3	92.8	94.3	**+5.0**	88.1	89.6	**+4.7**	97.9	98.5	+1.3
HS Macro	43.9	45.9	+4.7	51.0	52.6	+3.0	73.1	74.4	+4.1	63.6	64.9	+3.9	83.3	83.9	+1.3
HS Math	31.1	31.1	0.0	27.8	30.0	**+8.0**	34.8	40.0	**+16.7**	41.9	43.0	+3.4	48.9	51.1	**+5.4**
HS Micro	42.4	44.1	+4.0	59.2	60.5	+2.1	76.9	77.7	+2.7	72.7	76.1	**+10.2**	88.7	90.3	+4.1
HS Physics	29.8	31.1	+4.4	31.8	37.1	**+16.7**	45.7	49.0	**+10.7**	42.4	48.3	**+18.1**	57.6	60.3	**+6.4**
HS Psycho	63.7	65.0	+2.0	75.6	76.1	+0.7	88.6	89.7	**+3.6**	85.0	85.7	+2.2	93.9	93.9	0.0
HS Stat	27.8	30.6	**+10.0**	42.1	50.5	**+19.8**	62.0	65.7	**+12.0**	56.0	58.3	**+7.0**	73.6	75.0	+3.4
HS US Hist	53.4	56.9	**+6.4**	76.5	77.0	+0.6	89.7	92.7	**+9.5**	84.3	84.8	+1.5	95.1	95.6	+1.2
HS WorldHist	65.4	66.7	+1.9	70.9	73.4	+3.6	88.2	89.5	**+4.1**	82.3	83.5	+3.8	94.1	95.4	+3.1
Aging	55.2	56.5	+2.5	62.3	64.1	+2.9	79.8	80.7	+2.9	71.3	72.7	+4.1	82.5	83.4	+2.2
Sexuality	56.5	61.1	**+8.1**	61.1	64.1	**+5.0**	83.2	86.3	**+9.8**	74.8	78.6	**+11.6**	87.8	88.6	+1.9
Inter Law	62.8	64.5	+2.6	74.4	75.2	+1.1	86.8	89.3	**+8.0**	84.3	86.8	**+7.5**	90.1	91.7	+4.1
Juris	51.9	57.4	**+10.6**	71.9	74.1	**+3.6**	82.4	84.3	**+6.0**	74.0	79.6	**+17.1**	86.1	88.0	+4.5
LogFal	48.2	50.9	**+7.8**	68.7	70.6	+2.7	78.5	81.0	**+7.9**	74.9	77.9	**+9.3**	87.1	87.7	+1.5
ML	41.1	42.9	+4.4	28.0	40.0	**+42.9**	50.0	56.3	**+20.2**	54.5	58.0	**+10.8**	72.3	73.2	+2.2
Mgmt	56.3	60.2	**+6.9**	74.8	77.7	+3.9	83.5	86.4	**+9.4**	87.4	88.4	+2.9	91.3	92.2	+2.4
Marketing	70.0	70.5	+0.7	77.8	78.6	+1.1	89.3	89.7	+1.4	88.5	90.6	**+6.5**	94.0	94.4	+1.0
MedGene	53.0	57.0	**+7.6**	56.0	63.0	**+12.5**	71.0	75.0	**+12.9**	83.0	85.0	**+6.1**	89.0	91.0	+4.9
Miscell	62.8	64.8	+3.0	74.7	75.9	+1.5	85.7	86.2	+1.7	83.1	83.9	+2.3	91.7	92.2	+1.2
MoralDisp	48.3	50.9	**+5.4**	63.0	64.2	+1.8	76.9	78.9	**+6.5**	71.7	73.1	+4.4	85.0	85.8	+2.1
MoralScen	23.8	26.9	**+13.2**	37.4	37.8	+0.9	46.3	46.9	+2.2	41.3	42.2	+2.7	59.9	59.9	0.0
Nutrition	50.0	51.0	+2.0	62.4	63.4	+1.6	75.5	77.8	**+7.4**	76.1	77.8	**+5.0**	87.6	88.9	+3.2
Philosophy	59.0	61.0	+2.7	67.0	68.0	+0.5	78.8	80.7	**+6.2**	74.3	75.2	+2.9	86.5	88.4	+4.7
Prehistory	50.0	52.8	**+5.6**	62.8	66.4	**+5.6**	84.0	85.5	**+5.0**	73.5	75.3	**+5.6**	91.1	92.3	+3.0
ProAccount	35.8	39.0	**+8.9**	42.2	45.4	**+7.6**	57.4	58.2	+2.3	48.6	50.7	**+6.5**	64.5	67.4	**+6.9**
Pro Law	35.9	37.1	+3.3	42.1	42.6	+1.2	54.0	54.5	+1.5	46.7	47.1	+1.2	0.0	0.0	0.0
Pro Med	52.2	54.4	+3.7	56.1	56.1	0.0	73.9	75.4	**+4.7**	72.8	73.5	+2.2	87.9	88.6	+1.8
Pro Psy	44.8	45.6	+1.8	56.4	58.3	+3.5	75.5	76.1	+2.1	71.1	72.1	+3.0	87.4	87.6	+0.4
PR	51.8	52.7	+1.8	60.0	65.5	**+9.1**	74.6	77.3	**+8.8**	73.6	76.4	**+8.3**	72.7	76.4	**+8.9**
SecStud	42.9	46.5	**+7.1**	62.9	64.1	+1.9	78.8	80.8	**+6.6**	77.1	78.4	+3.7	83.3	86.1	**+7.0**
Sociology	62.2	66.7	**+8.7**	75.6	77.1	+2.0	89.6	92.0	**+8.0**	86.6	88.6	**+6.0**	93.0	94.0	+2.4
USFP	64.0	69.0	**+8.4**	84.0	85.0	+1.2	92.0	94.0	**+6.5**	88.0	89.0	+3.0	94.0	95.0	+2.4
Virology	40.4	44.6	**+8.2**	46.4	48.2	+3.9	53.6	56.0	**+7.8**	56.6	58.4	+5.5	59.0	59.6	+1.5
WorldRel	70.8	72.5	+3.4	77.2	79.0	+2.3	85.4	87.7	**+7.6**	81.9	82.5	+1.8	90.9	90.9	0.0

Table 4: Accuracy Improvements on MMLU Across LLaMA-2 and LLaMA-3 Models (Major Gains Highlighted).