Advanced Black-Box Tuning of Large Language Models with Limited API Calls

Zhikang Xie[1], **Weilin Wan**[1], **Peizhu Gong**[1*], **Weizhong Zhang**[2], **Cheng Jin**[1,3*]

[1] College of Computer Science and Artificial Intelligence, Fudan University
[2] School of Data Science, Fudan University
[3] Shanghai Key Laboratory of Intelligent Information Processing
{22307110187, wlwan23}@m.fudan.edu.cn, {pzgong, weizhongzhang, jc}@fudan.edu.cn

Abstract

Black-box tuning is an emerging paradigm for adapting large language models (LLMs) to better achieve desired behaviors, particularly when direct access to model parameters is unavailable. Current strategies, however, often present a dilemma of suboptimal extremes: either separately train a small proxy model and then use it to shift the predictions of the foundation model, offering notable efficiency but often yielding limited improvement; or making API calls in each tuning iteration to the foundation model, which entails prohibitive computational costs. In this paper, we argue that a more reasonable way for black-box tuning is to train the proxy model with limited API calls. The underlying intuition is based on two key observations: first, the training samples may exhibit correlations and redundancies, suggesting that the foundation model's predictions can be estimated from previous calls; second, foundation models frequently demonstrate low accuracy on downstream tasks. Therefore, we propose a novel advanced black-box tuning method for LLMs with limited API calls. Our core strategy involves training a Gaussian Process (GP) surrogate model with "LogitMap Pairs" derived from querying the foundation model on a minimal but highly informative training subset. This surrogate can approximate the outputs of the foundation model to guide the training of the proxy model, thereby effectively reducing the need for direct queries to the foundation model. Extensive experiments verify that our approach elevates pre-trained language model accuracy from **55.92%** to **86.85%**, reducing the frequency of API queries to merely **1.38%**. This significantly outperforms offline approaches that operate entirely without API access. Notably, our method also achieves comparable or superior accuracy to query-intensive approaches, while significantly reducing API costs. This offers a robust and high-efficiency paradigm for language model adaptation.

Code — https://github.com/kurumi8686/EfficientBBT

Introduction

Large Language Models (LLMs) have demonstrated remarkable capabilities in recent years. Adapting them to specific downstream tasks or aligning them with desired behaviors is essential for unlocking their full potential in real-world applications. Gradient-based methods, such as Adapter modules (Houlsby et al. 2019) and LoRA (Hu et al. 2022), are widely recognized as standard parameter-efficient fine-tuning techniques for LLMs. These methods adapt the pre-trained model to new tasks by tuning only a small subset of the model's parameters, and they have consistently achieved promising results in the literature. However, these methods require full access to model parameters, which is not feasible for many state-of-the-art LLMs, such as GPT-4 (Achiam et al. 2023) and Gemini (Team et al. 2023).

Black-box tuning is an emerging paradigm for adapting LLMs without direct parameter access. However, current strategies often involve a challenging trade-off. Specifically, **offline methods** (Liu et al. 2024) train a smaller proxy model independently and use it to adjust the black-box model's outputs during inference. While these methods are efficient, their performance is limited because the proxy model does not have direct access to the foundation model's internal knowledge during training. In contrast, **online methods**, such as Consistent Proxy Tuning (CPT) (He et al. 2024), integrate the black-box model into the proxy's training loop via iterative API calls. This approach improves alignment and performance but incurs significant computational and monetary costs. Consequently, practitioners face a dilemma: either sacrifice performance for efficiency or accept substantial costs for better adaptation.

In this paper, we argue that a more reasonable and resource-efficient way for black-box tuning is to train a proxy model with a strictly limited budget of these costly API calls. The underlying intuition for this approach is rooted in two key observations. Firstly, the training samples may exhibit inherent correlations and redundancies. This suggests that the foundation model's predictions for new, unseen inputs can often be effectively estimated or inferred from its responses to a smaller subset of previous calls. Secondly, even powerful foundation models do not always achieve perfect accuracy across all instances within the training data. Consequently, we posit that a well-informed approximation, rather than exhaustive querying of the foundation model, can still provide effective and comparable supervision for training a high-performing proxy model.

Therefore, we propose a novel advanced black-box tuning method specifically designed for LLMs operating under the limited API calls. Our core strategy involves leveraging a

*Corresponding Author.

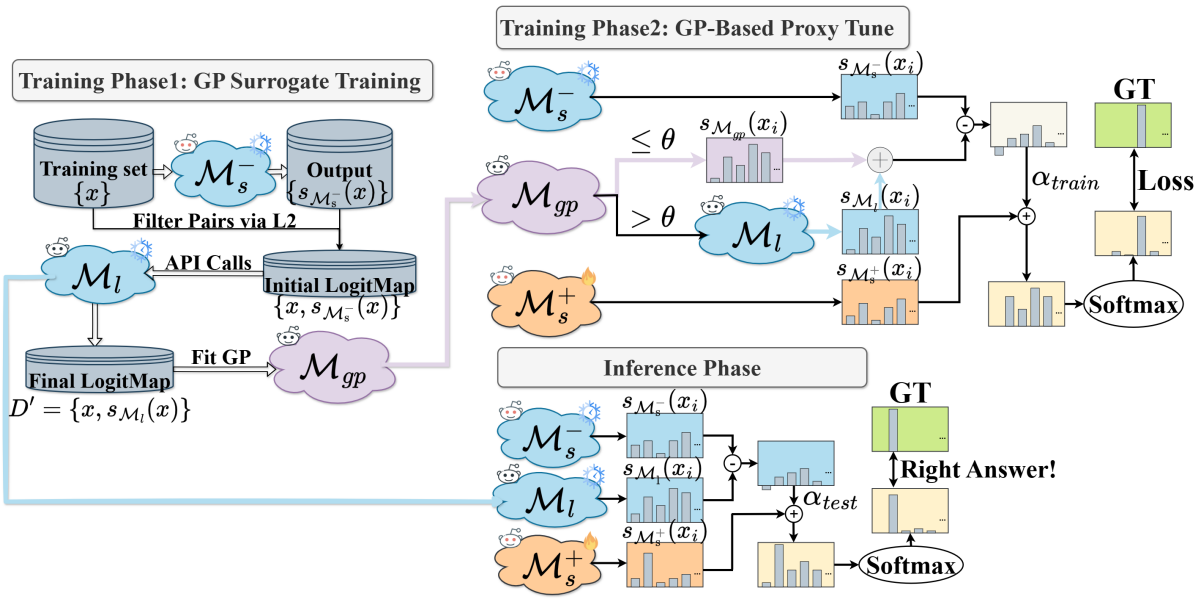

Figure 1: Overview of our proposed algorithmic framework. **Training Phase 1: GP Surrogate Model Training.** A Gaussian Process model $\mathcal{M}_{gp}$ is trained on a filtered subset of data to approximate the mapping between input embeddings and the output logits of the large black-box model $\mathcal{M}_l$. **Training Phase 2: GP-based Proxy Tuning.** The trained $\mathcal{M}_{gp}$ guides the fine-tuning of a small white-box proxy model $\mathcal{M}_s^+$, effectively incorporate knowledge from $\mathcal{M}_l$ into $\mathcal{M}_s^+$ via standard supervised training. **Inference Phase.** The final predictions are obtained by combining the outputs from the tuned proxy model $\mathcal{M}_s^+$ with an ensemble of the original proxy model $\mathcal{M}_s^-$ and the large black-box model $\mathcal{M}_l$.

Gaussian Process (GP) surrogate (Williams and Rasmussen 2006) to approximate the outputs of the foundation model to guide the training of the proxy model. As illustrated in Figure 1, the central idea is to train this GP surrogate on a small yet highly informative subset of the training data. We refer to these data points as "LogitMap Pairs", which consist of input embeddings and their corresponding output logits obtained from the foundation model. Once trained, the GP approximates the foundation's predictive behavior, thereby enabling logit-level supervision for the proxy model and highly reducing expensive black-box queries.

GPs, as non-parametric Bayesian models, are exceptionally well-suited for approximating complex functions and have been shown to emulate deep neural networks under certain conditions (Damianou and Lawrence 2013; Lee et al. 2017). A crucial advantage of GPs is their probabilistic nature, which allows for robust uncertainty estimation. This capability can help quantify the reliability of the surrogate's predictions and further inform strategic training decisions for the proxy model. We leverage this property to enhance the proxy model training process. Specifically, when the GP surrogate yields a prediction with high associated uncertainty (e.g., a large variance τ^2 exceeding a pre-defined threshold θ), we deem its prediction potentially unreliable. In such instances, our method falls back on invoking the black-box target model to obtain the true output. This adaptive mechanism ensures that primarily high-confidence surrogate predictions are utilized for training, thereby reducing the propagation of noise and improving the overall robustness and capabilities of the proxy model.

Extensive experiments across multiple NLP benchmarks demonstrate the effectiveness and scalability of our approach. It boosts the average accuracy of pre-trained language models from **55.92%** to **86.85%**, while reducing API query usage to just **1.38%** of that required by query-intensive methods such as CPT (He et al. 2024). Remarkably, despite this drastic reduction, our method achieves comparable or superior accuracy to these online methods, and significantly outperforms fully offline baselines. This highlights our approach as a robust and highly efficient paradigm for adapting LLMs in black-box settings.

Our contributions can be summarized as follows:

1. We propose a novel black-box tuning method that employs a GP surrogate to approximate foundation model outputs, enabling efficient proxy training with minimal API queries. As far as we have studied, our method improves from **55.92%** to **86.85%**, achieving **state-of-the-art** results and demonstrating its superior effectiveness.
2. We propose an effective data selection method for GP model training, requiring only **1.38%** training data.
3. Extensive experiments show that our method drastically reduces API usage compared to previous online approaches, while achieving better performance. This demonstrates a practical and cost-efficient paradigm for black-box tuning in real-world scenarios.

Related Works

Efficient Fine-tuning. The substantial cost of fully fine-tuning large models (Roziere et al. 2023; Groeneveld et al.

2024) has driven the development of parametric-efficient fine-tuning (PEFT) techniques (He et al. 2021; Lialin, Deshpande, and Rumshisky 2023). PEFT methods adapt models by modifying only a small parameter subset, aiming to preserve pre-trained knowledge while reducing resource demands. Common strategies involve inserting lightweight modules (e.g., Adapters (Houlsby et al. 2019), Compacter (Karimi Mahabadi, Henderson, and Ruder 2021)), optimizing continuous prompts or prefixes (e.g., Prompt Tuning (Lester, Al-Rfou, and Constant 2021), Prefix Tuning (Li and Liang 2021), P-Tuning v2 (Liu et al. 2021b)), or adjusting internal model parameters through approaches like low-rank updates (LoRA (Hu et al. 2022), QLoRA (Dettmers et al. 2023)), selective tuning (e.g., BitFit (Zaken, Ravfogel, and Goldberg 2021)), or learned activation scaling ($(\mathrm{IA})^3$ (Liu et al. 2022)). Despite their resource efficiency, these approaches typically require internal model access (weights and gradients), restricting their use in black-box scenarios.

Black-box Fine-tuning. Adapting LLMs without parameter access (i.e., in black-box settings) requires specialized fine-tuning techniques. Gradient-free optimization offers one approach, exemplified by Black-Box Tuning (BBT) (Sun et al. 2022), which optimizes input prompts by evaluating model outputs without gradient information. A dominant alternative involves employing smaller auxiliary models. Proxy-based methods, such as Proxy Tuning (PT) (Liu et al. 2024) and Consistent Proxy Tuning (CPT) (He et al. 2024), train an accessible white-box proxy and transfer task-specific knowledge by using differential signals from the proxy to guide the black-box model at inference. Other strategies train surrogate models to post-process or align the black-box outputs directly, using methods like sequence-to-sequence aligners (Ji et al. 2024) or adapting output probabilities (Ormazabal, Artetxe, and Agirre 2023; Lu et al. 2023). A key challenge across many methods relying on auxiliary models, especially CPT, is the potentially high cost associated with frequent API queries to the black-box LLM needed for training the auxiliary component.

Logit Arithmetic. Involving techniques that directly manipulate pre-softmax logits, often by aggregating signals from multiple sources or model states, is essentially an application of ensemble learning principles (Dong et al. 2020). For example, logit manipulation facilitates domain adaptation through ensembling logits from distinct models (Dou et al. 2019). In controllable generation, approaches include subtracting anti-expert logits (DExperts (Liu et al. 2021a)) and contrasting expert versus amateur model logits (CD (Li et al. 2022)). More recently, the principle has been extended to intra-model comparisons, where different layers are contrasted to enhance factuality (DoLa (Chuang et al. 2023)) or guide decoding via auto-contrastive objectives (Gera et al. 2023). The effectiveness and flexibility of logit ensembling motivate our exploration of logit-based adjustments as an efficient mechanism for black-box tuning.

Gaussian Process Models. Gaussian Processes (GPs) are non-parametric Bayesian methods well-suited for modeling complex functions and quantifying uncertainty (Williams and Rasmussen 2006). Key advances include sophisticated covariance functions, such as additive kernels for interpretable decomposition (Durrande, Ginsbourger, and Roustant 2011) and multiple kernel learning for integrating diverse properties (Gönen and Alpaydın 2011). GPs have also been integrated into more complex probabilistic frameworks to capture complex data structures. Notable examples include Warped GPs (Snelson, Ghahramani, and Rasmussen 2003), which transform the output space to model non-Gaussian likelihoods, and Gaussian Process Regression Networks (Wilson, Knowles, and Ghahramani 2011), which compose multiple GPs into deeper hierarchical models. The broad applicability of GPs across various natural language processing (NLP) tasks (Cohn, Preoţiuc-Pietro, and Lawrence 2014) further highlights their versatility. These advancements demonstrate the flexibility and modeling capabilities of Gaussian Processes, motivating our adoption of GP-based models as efficient surrogates.

Methodology

This section outlines our GP-based approach for efficient black-box tuning of large language models. Central to our method is a GP surrogate that approximates target models behavior, enabling high-quality adaptation with significantly fewer direct queries. We first provide a brief overview of existing proxy-based approaches, followed by a comprehensive and detailed description of our advanced framework, highlighting its key innovations and practical benefits.

Basics on Existing Proxy-based Methods

Proxy-Tuning Proxy-Tuning (PT) (Liu et al. 2024) adapts pre-trained LLMs at decoding time without access to their internal parameters, ideal for black-box or computationally constrained scenarios. It employs a small white-box proxy model, with a tuned version $\mathcal{M}_s^+$ and an untuned version $\mathcal{M}_s^-$. PT adjusts the logits of the large black-box model $\mathcal{M}_l$ by adding the logit difference from the small proxy models:

$$s_{pt}(x) = s_{\mathcal{M}_l}(x) + \left(s_{\mathcal{M}_s^+}(x) - s_{\mathcal{M}_s^-}(x)\right), \qquad (1)$$

where $s_{\mathcal{M}}(x)$ are logits from model $\mathcal{M}$. Originally, $\mathcal{M}_s^+$ (parameters θ_s^+) is trained independently on a task-specific dataset $D = \{(x, y)\}$ to minimize a loss L:

$$\theta_s^+ = \arg\min_{\theta_s^+} \mathbb{E}_{(x,y)\sim D}\big[L(\mathcal{M}_s^+(x;\theta_s^+), y)\big], \qquad (2)$$

this independent training of $\mathcal{M}_s^+$, however, overlooks its interaction with $\mathcal{M}_l$ and $\mathcal{M}_s^-$ during inference, potentially limiting performance.

Consistent Proxy Tuning Consistent Proxy Tuning (CPT) (He et al. 2024) refines PT by aligning the training objective of the small proxy model $\mathcal{M}_s^+$ with its actual usage during inference. This consistency is achieved by incorporating the influence of $\mathcal{M}_l$ and $\mathcal{M}_s^-$ into the training loss for $\mathcal{M}_s^+$. The training objective of CPT is:

$$\begin{aligned}\theta_s^+ = \arg\min_{\theta_s^+} \mathbb{E}_{(x,y)\sim D}\Big[L\Big(&\mathcal{M}_s^+(x;\theta_s^+) \\ &+ \alpha_{train}\big(\mathcal{M}_l(x;\theta_l) - \mathcal{M}_s^-(x;\theta_s^-)\big), y\Big)\Big].\end{aligned} \qquad (3)$$

During this training, parameters θ_l of $\mathcal{M}_l$ and θ_s^- of $\mathcal{M}_s^-$ are frozen, only θ_s^+ are fine-tuned. While CPT demonstrates improved performance due to this consistent objective, a significant practical drawback arises: optimizing $\mathcal{M}_s^+$ via Equation 3 necessitates frequent queries to the large black-box model $\mathcal{M}_l$, incurring substantial computational costs and API call limitations.

Proposed Method

To mitigate the high API call dependency of CPT, we introduce a GP model as a data-efficient surrogate for the large black-box model $\mathcal{M}_l$. The core idea is to pre-train a GP to approximate the logit outputs of $\mathcal{M}_l$ on the task-specific dataset D, and then use this GP surrogate during the CPT training of the small proxy model $\mathcal{M}_s^+$.

Gaussian Process Modeling of $\mathcal{M}_l$ Logits GPs are non-parametric Bayesian models adept at function approximation from limited data. Formally, a GP defines a prior distribution over $f(x) \sim \mathcal{GP}(m(x), k(x, x'))$, where $m(x)$ is the mean function and $k(x, x')$ is the kernel function. The kernel encodes prior beliefs about the function's properties, such as smoothness, by defining the covariance between function values at different input points x and x'.

Given a training dataset $D' = \{(x_j, \mathbf{s}_j)\}_{j=1}^M$, where $\mathbf{s}_j = s_{\mathcal{M}_l}(x_j) \in \mathbb{R}^V$ are the observed target black-box model output logits, the GP conditions on this data to form a posterior distribution. For a new input x_*, the predictive distribution for the logits $s_{\mathcal{M}_l}(x_*)$ is also Gaussian. A common and effective strategy for handling such multi-dimensional outputs, which we adopt in our implementation, is to model each of the V logit dimensions independently. This independent modeling assumption simplifies computation and often yields strong empirical results.

The predictive mean for the v-th logit dimension, $\hat{s}_{\mathcal{GP},v}(x_*)$, which serves as our approximation $s_{\mathcal{GP},v}(x_*)$, is given by formula 2.25 and 2.27 in page 17 of (Williams and Rasmussen 2006):

$$\hat{s}_{\mathcal{GP},v}(x_*) = \mathbf{k}(x_*, X_{D'})^T (K_{D'D'} + \sigma_{n,v}^2 I)^{-1} \mathbf{s}_{D',v}, \quad (4)$$

where the terms are defined as follows:

- $X_{D'} = \{x_j\}_{j=1}^M$ represents the set of M training data.
- $\mathbf{k}(x_*, X_{D'})$ is a vector in $\mathbb{R}^M$ denoting the covariances between the new input x_* and each training input $x_j \in X_{D'}$, with its j-th element being $k(x_*, x_j)$.
- $K_{D'D'}$ is the $M \times M$ covariance matrix computed from the training inputs, where each entry $(K_{D'D'})_{ij} = k(x_i, x_j)$ is the kernel evaluation between $x_i, x_j \in X_{D'}$.
- $\sigma_{n,v}^2$ is the noise variance hyperparameter for the v-th logit dimension, accounting for potential observation noise or model misspecification.
- I is the $M \times M$ identity matrix.
- $\mathbf{s}_{D',v}$ is a vector in $\mathbb{R}^M$ containing the observed values of the v-th logit dimension from the training set D'.

This framework, by applying independent GPs to each logit dimension, allows us to construct a composite multi-output GP model to predict the full logit vector $s_{\mathcal{M}_l}(x)$.

Algorithm 1: Algorithm for GP Training Set Construction

Require: Dataset D; proxy model $\mathcal{M}_s^-$; foundation model $\mathcal{M}_l$; thresholds $\tau_{\text{in}}, \tau_{\text{out}}$
Ensure: GP training set D'

Initialize $D_{\text{cand}} = \emptyset$, $D' = \emptyset$
For all $x \in D$, compute $\mathbf{v}_x = \text{embedding}(x)$ as input, compute $s_x = \mathcal{M}_s^-(x)$ as output
if $D \neq \emptyset$ **then**
    Seed D_{cand} with (x_1, s_{x_1})
end if
for each $x \in D \setminus \{x_1\}$ **do**
    diverse $\leftarrow$ true
    for each $(x_k, s_k) \in D_{\text{cand}}$ **do**
        if $\|\mathbf{v}_x - \mathbf{v}_{x_k}\| \leq \tau_{\text{in}} \vee \|s_x - s_k\| \leq \tau_{\text{out}}$ **then**
            diverse $\leftarrow$ false; **break**
        end if
    end for
    if diverse **then**
        Add (x, s_x) to D_{cand}
    end if
end for
for each $(x, s_x) \in D_{\text{cand}}$ **do**
    Query $\mathcal{M}_l$ for s'_x; add (x, s'_x) to D'
end for
return D'

Data Acquisition for GP via Selective Sampling A critical aspect is to train the GP effectively with minimal queries to $\mathcal{M}_l$. Instead of querying $\mathcal{M}_l$ for all $x \in D$, we construct a small but highly informative subset $D' = \{(x_j, s_{\mathcal{M}_l}(x_j))\}_{j=1}^M$, where $M \ll |D|$ (Typically, D' is approximately 1% the size of D). Then, D' is used by a filtering algorithm (detailed in Algorithm 1) that aims to maximize diversity and representativeness.

This filtering process primarily uses the input vector representations $\mathbf{v}_x$ and the output logits from the frozen small model $\mathcal{M}_s^-$. This ensures that the selection process itself is computationally inexpensive. Only once an input x is selected through this filtering, do we query the black-box large model $\mathcal{M}_l$ to obtain its true output logits $s_{\mathcal{M}_l}(x)$. These $(x, s_{\mathcal{M}_l}(x))$ then constitute the training set D' (i.e., LogitMap Pairs) for our GP model $\mathcal{M}_{gp}$.

As part of this filtering strategy, we experimented with several rule-based approaches to quantify the difference. Specifically, we evaluated Manhattan distance, Euclidean distance, and cosine similarity. Our results indicate that all three metrics can perform comparably, assuming appropriate input-output thresholds are set. Among them, Euclidean distance emerged as the most effective and computationally simple choice, and is therefore adopted in our final approach.

The Euclidean distances used for filtering are:

- Input distance: $d_{\text{input}}(x, x') = \|\mathbf{v}_x - \mathbf{v}_{x'}\|_2$.
- Output distance: $d_{\text{output}}(x, x') = \|s_{\mathcal{M}_s^-}(x) - s_{\mathcal{M}_s^-}(x')\|_2$.

If two data points have highly similar input representations and their outputs (as predicted by the inexpensive small proxy model) are also similar, they likely provide redundant

information. By filtering based on these metrics, we ensure D' is small yet rich in information, capturing diverse aspects of the input space and the proxy's initial assessment of output variations. This curated selection allows the GP to generalize effectively from fewer actual $\mathcal{M}_l$ queries.

GP-Enhanced Proxy Training With the trained GP model $\mathcal{M}_{gp}$ providing approximation for the outputs of the large foundation model $\mathcal{M}_l$, we introduce an uncertainty-aware training objective for the small proxy model $\mathcal{M}_s^+$. The objective is defined as:

$$\begin{aligned}\theta_s^+ = \arg\min_{\theta_s^+} \mathbb{E}_{(x,y)\sim D}\Big[L\Big(\mathcal{M}_s^+(x;\theta_s^+) \\ + \alpha_{train}\big(S_{gate}(x) - \mathcal{M}_s^-(x;\theta_s^-)\big),\ y\Big)\Big],\end{aligned} \tag{5}$$

where the gated supervision $S_{gate}(x)$ is determined by:

$$\begin{aligned}S_{gate}(x) = \mathcal{M}_{gp}(x;\theta_{gp}) \cdot \mathbf{1}_{\tau^2_{\mathcal{M}_{gp}}(x)\le\theta} \\ + \mathcal{M}_l(x;\theta_l) \cdot \mathbf{1}_{\tau^2_{\mathcal{M}_{gp}}(x)>\theta},\end{aligned} \tag{6}$$

where, $\tau^2_{\mathcal{M}_{gp}}(x)$ is the predictive variance of the GP model $\mathcal{M}_{gp}(x;\theta_{gp})$ for input x, and θ is the pre-defined variance threshold. The term $\mathbf{1}$ denotes the indicator function, where $\mathbf{1}_{\text{condition}}$ is 1 if the condition is true, and 0 otherwise.

During training, the parameters of the GP surrogate $\mathcal{M}_{gp}(\cdot;\theta_{gp})$ and the untuned proxy model $\mathcal{M}_s^-(\cdot;\theta_s^-)$ are kept frozen, while only the trainable proxy $\mathcal{M}_s^+(\cdot;\theta_s^+)$ is fine-tuned. To ensure robust supervision, we introduce a gating mechanism based on the GP's predictive uncertainty. For each input x, if the GP variance $\tau^2_{\mathcal{M}_{gp}}(x) \le \theta$, its prediction is used as the guidance signal $S_{\text{gate}}(x)$; otherwise, we query the target black-box model $\mathcal{M}_l(x)$ to obtain a reliable label.

Inference At inference time, our procedure closely follows that of PT / CPT. The adjusted logits for a new input x are computed as:

$$s_{\text{final}}(x) = s_{\mathcal{M}_s^+}(x) + \alpha_{test}\left(s_{\mathcal{M}_l}(x) - s_{\mathcal{M}_s^-}(x)\right), \tag{7}$$

where $s_{\mathcal{M}_l}(x)$ denotes the output of the target black-box model. Typically, we set $\alpha_{test} = \alpha_{train}$. The final prediction is then obtained by applying the softmax function to $s_{\text{final}}(x)$.

Experiments

Experimental Setup

Models Selection. To balance efficiency and reliability, we select models from **Llama2** (Touvron et al. 2023), **Mistral-7B** (Jiang et al. 2023), **Qwen3** (Yang et al. 2025) and **DeepSeek-R1-Distill** (DeepSeek-AI et al. 2025) families. For each series, we use a small model as the proxy and a larger one as the target. While the selected large models are technically white-box, we treat them as black-box during validation to simulate realistic constraints. We also directly fine-tune them to obtain an oracle upper bound, which serves as a reference for evaluating our method's effectiveness. For evaluations involving truly inaccessible black-box models, please refer to Appendix, where we conduct experiments under genuine black-box conditions.

Datasets Selection. We conducted experiments across diverse NLP datasets to showcase the versatility of our method. Our approach was evaluated on three major tasks: **(a).** Text Classification: We use AG-News (Zhang, Zhao, and LeCun 2015), CoLA (Warstadt, Singh, and Bowman 2019), SST-2 (Stanford Sentiment Treebank) (Socher et al. 2013), and QQP (Quora Question Pairs) (Shankar and Nikhil 2017). **(b).** Question Answering: We include ARC-C (AI2 Reasoning Challenge - Challenge Set) (Clark et al. 2018), CsQA (CommonsenseQA) (Talmor et al. 2018), and OB-QA (OpenBookQA) (Mihaylov et al. 2018). **(c).** Natural Language Inference: We consider MNLI (Multi-Genre Natural Language Inference) (Williams, Nangia, and Bowman 2017), QNLI (Question Natural Language Inference) (Rajpurkar, Jia, and Liang 2018), RTE (Recognizing Textual Entailment) (Dagan, Glickman, and Magnini 2005), and CoPA (Choice of Plausible Alternatives) (Roemmele, Bejan, and Gordon 2011). These datasets are widely used and well-established benchmarks, covering diverse linguistic phenomena and evaluation challenges.

Baselines. We compare our method against several representative baselines to demonstrate its effectiveness: **(a).** Zero-shot Inference: We evaluate the pretrained LLMs without any tuning, by directly applying them to the test sets. This provides a baseline for all tuning methods. **(b).** Direct Fine-tuning: We apply both LoRA (Hu et al. 2022) and full-precise fine-tuning on the LLMs. They provide upper-bound references to demonstrate the performance of our approach. **(c).** Proxy-Tuning and CPT: We compare our method with leading black-box tuning approaches, including Proxy-Tuning (Liu et al. 2024) and CPT (He et al. 2024), demonstrating that our approach achieves superior performance while requiring substantially fewer API calls.

Empirical Data Selection. As shown in Table 2, we empirically determined the data proportions for our two strategies. For the random strategy, we iteratively adjusted the sampling ratio until the resulting accuracy matched that of CPT. For the filter strategy, we tuned the input and output thresholds, and repeatedly applied Algorithm 1 to construct LogitMap Pairs, aiming to strike a balance between efficiency and performance. Based on extensive evaluation, we make the following recommendations:

- **Random-based:** For datasets fewer than 100K samples, 5% random sampling performs well. For larger datasets, sampling 5K examples is sufficient.
- **Filter-based:** For most datasets, selecting around 1% of the data via our filtering algorithm is sufficient. However, for extremely large datasets such as MNLI and QQP, which contain nearly 400K samples, fitting a GP on a proportional subset becomes computationally infeasible and may lead to numerical instability (e.g., NaN output). To address this, we recommend sampling approximately 2K examples, which balances efficiency and accuracy.

Compared to tuning the small proxy model, the phase of constructing LogitMap Pairs and Training Gaussian Process model is significantly more efficient in both time and memory. **Detailed results** are provided in Appendix.

Method	Accuracy (%) ↑												Avg.API ↓
	AG-News	CoLA	CoPA	SST-2	ARC-C	Cs-QA	OB-QA	MNLI	QNLI	RTE	QQP	Avg.	
Qwen3 Series													
Pretrain (8B)	85.86	83.22	93.60	92.09	86.62	78.13	83.20	84.20	85.54	85.56	83.38	85.58	-
Pretrain (14B)	83.78	84.85	96.40	88.42	88.96	80.51	85.60	81.03	82.67	83.39	77.86	84.86	-
LoRA-Tune (8B)	90.21	84.28	96.20	95.87	90.30	80.02	88.40	89.56	88.96	81.23	90.34	88.67	-
LoRA-Tune (14B)	88.67	81.50	94.00	95.41	91.30	81.57	90.20	90.51	92.88	89.17	90.97	89.65	-
Full Fine-tune (8B)	92.57	84.95	97.40	96.33	91.30	82.72	90.00	90.59	93.61	87.73	91.72	90.81	-
Full Fine-tune (14B)	93.46	87.25	98.40	97.59	93.31	87.22	90.20	91.20	94.62	91.70	93.21	92.56	-
Proxy Model Black-Box Tuning Methods													
Proxy-Tune	82.07	69.32	86.20	90.60	88.96	77.56	86.20	88.16	88.19	82.31	84.75	84.03	0%
CPT	93.54	86.29	98.20	95.87	**93.31**	85.01	90.40	90.06	**93.78**	90.61	**92.28**	91.76	100%
GP-random (ours)	92.18	85.71	98.60	95.18	91.97	85.01	89.20	91.01	92.99	90.97	91.25	91.28	6.94%
GP-filter (ours)	**93.22**	**86.77**	**98.80**	**96.10**	92.31	**86.49**	**90.80**	**91.05**	93.57	**92.06**	91.93	**92.10**	**1.58%**

Table 1: Experimental results comparing our GP tuning with other approaches, including white-box LoRA and black-box proxy tuning methods, across 11 datasets. Our techniques are denoted by GP-random and GP-filter. We use *Qwen3-8B* as a small white-box proxy model and *Qwen3-14B* as the black-box foundation model. "Pretrain" refers to zero-shot inference using official pretrained parameters, "LoRA-Tune" denotes fine-tuning via LoRA (Hu et al. 2022), and "Full Fine-tune" refers to directly fine-tuning all model parameters. The methods Proxy-Tune, CPT, GP-random, and GP-filter are grouped as *Proxy Model Black-Box Tuning Methods*. All datasets are evaluated by Accuracy (higher is better). Experimental results for other model families (Llama2, Mistral, and DeepSeek) are provided in the Appendix.

Main Results

The main experimental results are presented in Table 1. Our proposed GP-based tuning methods, particularly GP-filter, exhibit consistently strong performance across all evaluated models and datasets. Focusing first on the *Llama2* family, GP-filter improves the average accuracy from **55.92%** (pretrained) to **86.85%** (GP-filter tuned), even outperforming LoRA-Tune (85.79%) and approaching the performance of full fine-tuning (88.58%) on *Llama2-13B*. Compared to other proxy-based approaches, GP-filter achieves a higher average accuracy than CPT (86.41%) while using only **1.38%** of its API calls—to the best of our knowledge, this represents the state-of-the-art in both performance gain and API efficiency—demonstrating both effectiveness and remarkable cost-efficiency. It also outperforms offline Proxy-Tune by an average margin of 1.93 percentage points.

To validate generalizability, we conduct extensive experiments across other model families including *Mistral-7B*, *Qwen3*, and *DeepSeek-R1-Distill*. In all cases, GP-filter yields consistent improvements over the pretrained models, achieving average accuracy gains of **+9.94**, **+7.24**, and **+13.27** percentage points, respectively. Notably, in each setting, GP-filter uses few API calls (below 2%), highlighting its extreme cost-efficiency alongside strong performance.

We also apply GP-filter in a simulated **real-world black-box LLM** setting (*Qwen-Plus* model from Tongyi Qianwen) to further assess its practical applicability. As detailed in Appendix, the method remains highly effective even under multiple realistic constraints, confirming its robustness in truly black-box environments.

Ablation Study: Different Usage of API Calls

We trained GP on 6 datasets—CoPA, ARCC, CoLA, RTE, OBQA, and MRPC—using different API calls to generate variant GP surrogates, which were then used to guide the small proxy via the GP-filter method. Figure 2 compares the output distributions of GP and the target foundation model for CoLA. Results for others are provided in Appendix.

Under extreme data scarcity, the GP logits distribution becomes highly compressed, as observed in Figure 2 (the top left subfigure). With **only** 2 API calls (out of 8,551), the distribution nearly degenerates into a one-dimensional form, effectively reducing the GP model to a linear function. Surprisingly, the performance of the GP-filter method remains robust. In contrast, Figure ARCC (the bottom right subfigure, displayed in Appendix) shows the opposite extreme, where nearly all data (1,101 out of 1,119) is used, GP logits closely match those of the *Llama2-13B* model, and the GP-filter method achieves performance comparable to CPT.

These results highlight that GP-filter can perform well even with **extremely limited API calls**. Rather than replicating the large model's logits, the GP appears to approximate its underlying knowledge structure, with added noise that may serve as implicit regularization, enhancing the proxy's

Method	API Call Efficiency (%) ↓											
	AG-News	CoLA	CoPA	SST-2	ARC-C	Cs-QA	OB-QA	MNLI	QNLI	RTE	QQP	Avg.
CPT (He et al. 2024)	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
GP-random (ours)	3.33	11.69	10.00	5.94	9.83	10.27	8.07	1.02	4.77	10.04	1.37	6.94
GP-filter (ours)	1.21	0.74	1.20	1.38	2.95	2.77	2.08	0.83	1.79	1.08	1.37	**1.58**

Table 2: Large model API call efficiency. This table compares the percentage of large model API calls used by our methods versus CPT (He et al. 2024). For CPT, an API call is made for every training instance (i.e., 100% usage). Our GP-filter method requires only an average of **1.58%** for Qwen3 of these calls. Results for other model families are provided in the Appendix.

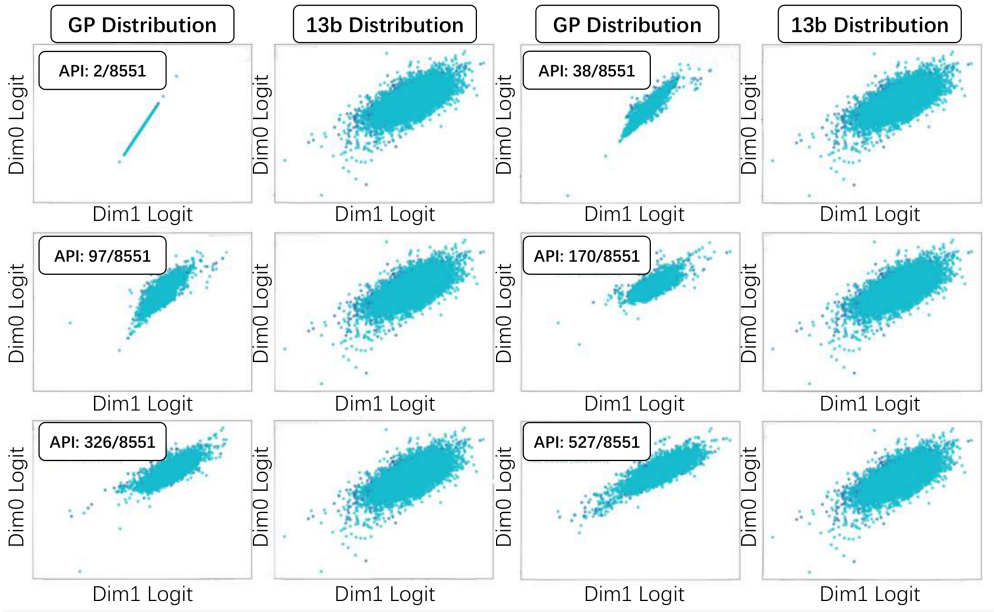

Figure 2: Logits distributions produced by the GP models (each left subfigures) and the target black-box model (each right subfigures) across two datasets. The figure consists of four columns, grouped as two pairs: the first and third columns show the output logits distributions of six GP models trained under different API budgets; the second and fourth columns display the output logits of the same *Llama2-13b* pretrained model under the same inputs, repeated for clearer comparison.

robustness. Even in extreme minimal-data regimes (e.g., 2 samples in CoLA), the GP model provides a useful corrective signal, capturing high-level structural patterns such as distributional tendencies and relative logit relationships.

Additional ablation studies provide further insights into the strengths of our approach. As shown in Appendix, under extreme data scarcity, directly fine-tuning the *Llama2-7B* model yields low effectiveness. In contrast, our GP-filter method maintains high performance, achieving 6.31 percentage points higher accuracy. Furthermore, Appendix shows that on more challenging datasets, our method significantly outperforms offline approaches such as Proxy-Tune. This improvement stems from the fact that Proxy-Tune only leverages the foundation model at the reference stage; however, in difficult tasks, both the fine-tuned proxy model and the pretrained large model may perform suboptimally. In contrast, our approach incorporates the foundation model's knowledge throughout the training process, leading to more effective proxy fine-tuning and consistently better results.

Conclusion

In this paper, we introduce a Gaussian Process (GP) based black-box proxy tuning approach that directly addresses the central challenge: aligning models effectively while minimizing expensive API calls. Our approach trains a GP surrogate on a small, curated dataset to approximate the target model's behavior and uses its predictive uncertainty to guide selective querying during proxy fine-tuning. Experiments demonstrate that our method matches or outperforms competitive online black-box tuning techniques while using substantially fewer APIs, and consistently surpasses existing offline strategies. These results not only highlight the practicality and cost-efficiency of proxy tuning, but also confirm its effectiveness in realistic black-box adaptation scenarios.

Acknowledgments

This work was supported by the National Natural Science Foundation of China Youth Student Basic Research Program (Grant No. 625B1002). It was also supported by High-Quality Development Project of Shanghai Municipal Commission of Economy and Informatization (Grant No. 2024-GZL-RGZN-02010) and AI for Science Foundation of Fudan University (FudanX24AI028).

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chuang, Y.-S.; Xie, Y.; Luo, H.; Kim, Y.; Glass, J.; and He, P. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Cohn, T.; Preoţiuc-Pietro, D.; and Lawrence, N. 2014. Gaussian processes for natural language processing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: Tutorials*, 1–3.

Dagan, I.; Glickman, O.; and Magnini, B. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, 177–190. Springer.

Damianou, A.; and Lawrence, N. D. 2013. Deep gaussian processes. In *Artificial intelligence and statistics*, 207–215. PMLR.

DeepSeek-AI; Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; Zhang, X.; Yu, X.; Wu, Y.; Wu, Z. F.; Gou, Z.; Shao, Z.; Li, Z.; Gao, Z.; Liu, A.; Xue, B.; Wang, B.; Wu, B.; Feng, B.; Lu, C.; et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in neural information processing systems*, 36: 10088–10115.

Dong, X.; Yu, Z.; Cao, W.; Shi, Y.; and Ma, Q. 2020. A survey on ensemble learning. *Frontiers of Computer Science*, 14: 241–258.

Dou, Z.-Y.; Wang, X.; Hu, J.; and Neubig, G. 2019. Domain differential adaptation for neural machine translation. *arXiv preprint arXiv:1910.02555*.

Durrande, N.; Ginsbourger, D.; and Roustant, O. 2011. Additive kernels for Gaussian process modeling. *arXiv preprint arXiv:1103.4023*.

Gera, A.; Friedman, R.; Arviv, O.; Gunasekara, C.; Sznajder, B.; Slonim, N.; and Shnarch, E. 2023. The benefits of bad advice: Autocontrastive decoding across model layers. *arXiv preprint arXiv:2305.01628*.

Gönen, M.; and Alpaydın, E. 2011. Multiple kernel learning algorithms. *The Journal of Machine Learning Research*, 12: 2211–2268.

Groeneveld, D.; Beltagy, I.; Walsh, P.; Bhagia, A.; Kinney, R.; Tafjord, O.; Jha, A. H.; Ivison, H.; Magnusson, I.; Wang, Y.; et al. 2024. Olmo: Accelerating the science of language models. *arXiv preprint arXiv:2402.00838*.

He, J.; Zhou, C.; Ma, X.; Berg-Kirkpatrick, T.; and Neubig, G. 2021. Towards a unified view of parameter-efficient transfer learning. *arXiv preprint arXiv:2110.04366*.

He, Y.; Huang, Z.; Xu, X.; Goh, R. S. M.; Khan, S.; Zuo, W.; Liu, Y.; and Feng, C.-M. 2024. CPT: Consistent Proxy Tuning for Black-box Optimization. *arXiv preprint arXiv:2407.01155*.

Houlsby, N.; Giurgiu, A.; Jastrzebski, S.; Morrone, B.; De Laroussilhe, Q.; Gesmundo, A.; Attariyan, M.; and Gelly, S. 2019. Parameter-efficient transfer learning for NLP. In *International conference on machine learning*, 2790–2799. PMLR.

Hu, E. J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W.; et al. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2): 3.

Ji, J.; Chen, B.; Lou, H.; Hong, D.; Zhang, B.; Pan, X.; Qiu, T. A.; Dai, J.; and Yang, Y. 2024. Aligner: Efficient alignment by learning to correct. *Advances in Neural Information Processing Systems*, 37: 90853–90890.

Jiang, A. Q.; Sablayrolles, A.; Mensch, A.; Bamford, C.; Chaplot, D. S.; de las Casas, D.; Bressand, F.; Lengyel, G.; Lample, G.; Saulnier, L.; Lavaud, L. R.; Lachaux, M.-A.; Stock, P.; Scao, T. L.; Lavril, T.; Wang, T.; Lacroix, T.; and Sayed, W. E. 2023. Mistral 7B. arXiv:2310.06825.

Karimi Mahabadi, R.; Henderson, J.; and Ruder, S. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. *Advances in Neural Information Processing Systems*, 34: 1022–1035.

Lee, J.; Bahri, Y.; Novak, R.; Schoenholz, S. S.; Pennington, J.; and Sohl-Dickstein, J. 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*.

Lester, B.; Al-Rfou, R.; and Constant, N. 2021. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*.

Li, X. L.; Holtzman, A.; Fried, D.; Liang, P.; Eisner, J.; Hashimoto, T.; Zettlemoyer, L.; and Lewis, M. 2022. Contrastive decoding: Open-ended text generation as optimization. *arXiv preprint arXiv:2210.15097*.

Li, X. L.; and Liang, P. 2021. Prefix-tuning: Optimizing continuous prompts for generation. *arXiv preprint arXiv:2101.00190*.

Lialin, V.; Deshpande, V.; and Rumshisky, A. 2023. Scaling down to scale up: A guide to parameter-efficient fine-tuning. *arXiv preprint arXiv:2303.15647*.

Liu, A.; Han, X.; Wang, Y.; Tsvetkov, Y.; Choi, Y.; and Smith, N. A. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Liu, A.; Sap, M.; Lu, X.; Swayamdipta, S.; Bhagavatula, C.; Smith, N. A.; and Choi, Y. 2021a. DExperts: Decoding-time controlled text generation with experts and anti-experts. *arXiv preprint arXiv:2105.03023*.

Liu, H.; Tam, D.; Muqeeth, M.; Mohta, J.; Huang, T.; Bansal, M.; and Raffel, C. A. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *Advances in Neural Information Processing Systems*, 35: 1950–1965.

Liu, X.; Ji, K.; Fu, Y.; Tam, W. L.; Du, Z.; Yang, Z.; and Tang, J. 2021b. P-tuning v2: Prompt tuning can be comparable to fine-tuning universally across scales and tasks. *arXiv preprint arXiv:2110.07602*.

Lu, X.; Brahman, F.; West, P.; Jang, J.; Chandu, K.; Ravichander, A.; Qin, L.; Ammanabrolu, P.; Jiang, L.; Ramnath, S.; et al. 2023. Inference-time policy adapters (ipa): Tailoring extreme-scale lms without fine-tuning. *arXiv preprint arXiv:2305.15065*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Ormazabal, A.; Artetxe, M.; and Agirre, E. 2023. Comblm: Adapting black-box language models through small fine-tuned models. *arXiv preprint arXiv:2305.16876*.

Rajpurkar, P.; Jia, R.; and Liang, P. 2018. Know what you don't know: Unanswerable questions for SQuAD. *arXiv preprint arXiv:1806.03822*.

Roemmele, M.; Bejan, C. A.; and Gordon, A. S. 2011. Choice of Plausible Alternatives: An Evaluation of Commonsense Causal Reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, 90–95.

Roziere, B.; Gehring, J.; Gloeckle, F.; Sootla, S.; Gat, I.; Tan, X. E.; Adi, Y.; Liu, J.; Sauvestre, R.; Remez, T.; et al. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Shankar, I.; and Nikhil, C., Dandekarand Kornel. 2017. First Quora Dataset Release: Question Pairs. https://quoradata.quora.com/First-Quora-Dataset-Release-Question-Pairs.

Snelson, E.; Ghahramani, Z.; and Rasmussen, C. 2003. Warped gaussian processes. *Advances in neural information processing systems*, 16.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Sun, T.; Shao, Y.; Qian, H.; Huang, X.; and Qiu, X. 2022. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, 20841–20855. PMLR.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. *arXiv preprint arXiv:1811.00937*.

Team, G.; Anil, R.; Borgeaud, S.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; Millican, K.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Warstadt, A.; Singh, A.; and Bowman, S. R. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7: 625–641.

Williams, A.; Nangia, N.; and Bowman, S. R. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Williams, C. K.; and Rasmussen, C. E. 2006. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA.

Wilson, A. G.; Knowles, D. A.; and Ghahramani, Z. 2011. Gaussian process regression networks. *arXiv preprint arXiv:1110.4411*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; et al. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Zaken, E. B.; Ravfogel, S.; and Goldberg, Y. 2021. Bitfit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. *arXiv preprint arXiv:2106.10199*.

Zhang, X.; Zhao, J.; and LeCun, Y. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.