Inference-Aware Prompt Optimization for Aligning Black-Box Large Language Models

Saaduddin Mahmud, Mason Nakamura, Kyle Hollins Wray, Shlomo Zilberstein

Manning College of Information and Computer Sciences
University of Massachusetts Amherst
{smahmud, mnakamura, kwray, shlomo}@umass.edu

Abstract

Prompt optimization methods have demonstrated significant effectiveness in aligning black-box large language models (LLMs). In parallel, inference scaling strategies such as BEST-OF-N Sampling and MAJORITY VOTING have likewise been shown to improve alignment and performance by trading additional computation for better output. However, existing prompt optimization approaches are inference strategy agnostic; that is, they optimize prompts without accounting for the inference strategy. This constitutes a significant methodological gap, as our empirical and theoretical analysis reveals a strong interdependence between these two paradigms. Moreover, we find that user preferences regarding trade-offs among multiple objectives and inference budgets substantially influence the choice of prompt and inference configuration. To address this gap, we introduce a novel unified framework named IAPO (Inference-Aware Prompt Optimization) that jointly optimizes the prompt and inference scale, while being aware of the inference budget and different task objectives. We then develop a fixed-budget training algorithm for IAPO, called PSST (Prompt Scaling via Sequential Trimming), and establish finite-budget guarantees on the error probability. Finally, we evaluate the effectiveness of PSST on six tasks, including multi-objective text generation and reasoning, and demonstrate the critical role of incorporating inference-awareness in aligning black-box LLMs using prompt optimization.

Introduction

Most state-of-the-art large language models (LLMs) are currently accessible exclusively through black-box APIs. Traditional alignment methods that require access to model weights or logits are therefore infeasible. To address this challenge, prompt-based alignment methods have gained substantial attention in recent work (Chang et al. 2024). These methods typically enhance input prompts by rewording them or appending additional instructions to better align the models' outputs with a task's objectives. Another broadly applicable alignment method for black-box models is scaling inference computations using strategies such as BEST-OF-N sampling or MAJORITY VOTING. These inference scaling methods generate multiple candidate responses for the same query and select the final response via ranking or voting mechanisms (Krishna et al. 2022; Wang et al. 2023; Gui, Gârbacea, and Veitch 2024; Yue et al. 2025).

Although existing prompt optimization techniques have achieved substantial success, they are typically agnostic to how model outputs are aggregated or sampled, overlooking the impact of such inference methods. Our initial empirical investigation reveals that the performance of optimized prompts is highly sensitive to the choice of inference scaling approach. Furthermore, our theoretical analysis reveals that decoupling prompt optimization from inference can lead to misalignment. Finally, we observe that optimal alignment requires careful consideration of user-specific preferences regarding the trade-offs among multiple objectives, as well as the computational resources users are willing to expend. These findings expose a critical gap in current methods: the absence of a unified framework that simultaneously accounts for prompt optimization, inference scaling strategies, user preferences, and computational resource constraints.

To bridge this gap, we introduce IAPO (Inference-Aware Prompt Optimization), a novel prompt optimization framework designed explicitly to produce aligned responses from inference-scaled black-box LLMs. IAPO simultaneously optimizes prompt design and inference scaling strategies while considering different task objectives and computational budgets (Figure 1). We formulate the task of identifying an optimal policy for the IAPO framework as a contextual best-arm identification problem. To efficiently solve this, we propose a fixed-budget training algorithm named PSST (Prompt Scaling via Sequential Trimming). Additionally, we introduce a warm-up heuristic that further improves performance within the training budget.

We begin our analysis by deriving theoretical finite-budget guarantees on the error probability of PSST. Next, we empirically demonstrate the effectiveness of PSST for learning IAPO policies across six diverse tasks, including multi-objective text generation, mathematical reasoning, and commonsense reasoning benchmarks. Additionally, our analysis shows that ignoring inference scaling during prompt optimization can lead to substantial misalignment, highlighting the critical role of inference-awareness in aligning black-box LLMs. The results establish that prompt quality cannot be decoupled from the inference strategies. By formalizing this interaction and introducing a practical algorithm that ex-

ploits it, our work offers a principled path toward more reliable and cost-effective alignment of black-box LLMs.

Related Work

In recent years, substantial efforts have been directed towards aligning large language models (LLMs) with human expectations in downstream tasks (Mahmud, Saisubramanian, and Zilberstein 2023; Minaee et al. 2024). Many widely adopted alignment approaches—such as Supervised Fine-Tuning (SFT), Reinforcement Learning from Human Feedback (RLHF), and Reinforcement Learning with Verifiable Rewards (RLVR) (Lambert 2025)—require access to model weights. This limitation has motivated a surge of interest in *black-box* alignment methods such as *prompt optimization*, which can align black-box models only through input manipulation (Ouyang et al. 2022; Zhou et al. 2023; Chang et al. 2024). Prompt optimization has demonstrated strong performance in both single-objective (Cheng et al. 2024; Trivedi et al. 2025) and multi-objective (Jafari et al. 2024; Zhao et al. 2025) settings. However, these methods remain agnostic to the inference strategy during deployment, potentially leading to suboptimal performance. In contrast, our approach explicitly captures the interdependence between inference-time strategies and prompt optimization.

Recently, Shi et al. framed prompt optimization as a fixed-budget best-arm identification (BAI) problem. While effective under limited evaluation budgets, the method remains inference agnostic and was only explored in single-objective settings. Our work builds on this foundation in two key ways: (1) we introduce a contextual formulation that models user preferences over multiple objectives and associated computational costs; and (2) we incorporate inference-awareness to ensure alignment with the deployed inference strategy. To learn an optimal policy, we introduce a fixed-budget contextual BAI algorithm, PSST, inspired by Sequential Halving (SH) (Karnin, Koren, and Somekh 2013). While SH was originally developed for the pure bandit setting, the IAPO framework features both inter-context full-information feedback and intra-context semi-bandit feedback. PSST leverages these structural properties to achieve more efficient optimization, extending beyond what standard SH can accommodate.

Another relevant line of work focuses on *inference-time alignment*, where model outputs are improved during inference without modifying model parameters. Some of these methods, such as GenARM and DEAL (Xu et al. 2025; Huang et al. 2025), require access to model logits, limiting their applicability in black-box settings. In contrast, BEST-OF-N sampling (BON) and MAJORITY VOTING (MV) methods operate purely on model outputs and have shown strong empirical gains by generating multiple candidates and selecting the best one (Krishna et al. 2022; OpenAI 2024; Yue et al. 2025). However, these approaches introduce a non-trivial computational cost, and to our knowledge, none of them explicitly optimize the trade-off between computational budget and output quality. Our initial experiments also indicate that inference scaling strategies have complex interactions with prompt design. Prompts that are optimized for single-shot decoding might not perform well with BON or MV, and the reverse is also true. Therefore, an inference-aware prompt optimization framework is required.

Finally, some white-box methods have recently integrated inference-awareness into the training process. Chow et al. (2025) proposed an inference-aware fine-tuning procedure that explicitly optimizes for exploration–exploitation trade-offs under BON. Similarly, BOND (Sessa et al. 2025) and BonBon (Gui, Gârbacea, and Veitch 2024) aim to distill BON policies into a single-pass decoding policy. While these approaches avoid the cost of sampling at inference time, they require full access to model parameters and do not generalize beyond BON-style strategies. In contrast, our method complements inference-aware fine-tuning and is designed to operate in fully black-box settings.

Inference-Aware Prompt Optimization

In this section, we first formalize the problem setup and introduce the IAPO framework. Next, we present an empirical example that highlights the need for inference-aware optimization. Building on these observations, we then establish theoretical conditions under which IAPO is necessary compared to disjoint optimization.

Problem Formulation

Let $\mathcal{X}$ be the set of user queries and $\mathcal{P}$ a finite prompt set. A pair $(x \in \mathcal{X}, p \in \mathcal{P})$ is submitted to a frozen black-box LLM, which, under fixed decoding hyperparameters, generates $N \in [N_{\max}]$ (i.e., $\{1, \dots, N_{\max}\}$) i.i.d. completions $\mathbf{y}_{1:N} = (y_1, \dots, y_N)$. K bounded objectives (e.g. *helpfulness*, *harmlessness*, *exact-match*) score each completion $O_k : \mathcal{X} \times \mathcal{P} \times \mathcal{Y} \to [o_k^{\min}, o_k^{\max}]$ where $\mathcal{Y}$ denotes the space of model completions. We also define the cost of producing a response as $\mathrm{Cost}(x, p, y_i)$, a composite function that takes into account various computational factors such as token count, time, and energy. We add it as a $(K{+}1)$-st objective $O_{K+1} = -\mathrm{Cost}(x, p, y_i)$. An external entity supplies a *context* $c = (w_1, \dots, w_{K+1}) \in \mathcal{C}$, where every w_k is chosen from a *finite* discrete domain. Given the above setup, we now formalize the inference strategies.

BEST-OF-N (BON). BON returns the largest weighted utility:

$$R_x^{\mathrm{BON}}(c, p, N) = \max_{i \le N} \underbrace{\sum_{k=1}^{K} w_k\, O_k(x, p, y_i)}_{\text{task reward}} + \underbrace{w_{K+1} \sum_{i=1}^{N} O_{K+1}(x, p, y_i)}_{\text{inference cost}}. \quad (1)$$

MAJORITY VOTING (MV). For query x, the pair (p, N) yields i.i.d. completions $\mathbf{y}_{1:N}$ and extracted answers $\ell_{1:N}$. For each distinct answer s, define the vote count $n_s = \sum_{i=1}^{N} \mathbb{1}[\ell_i = s]$, the maximum $n^\star = \max_s n_s$, and the tie multiplicity $t = \sum_s \mathbb{1}[n_s = n^\star]$. MV predicts uniformly at random among the t maximizers. With gold answer γ and the success credit defined as $O_1(x, p, \mathbf{y}_{1:N}) = \frac{\mathbb{1}[n_\gamma = n^\star]}{t}$,

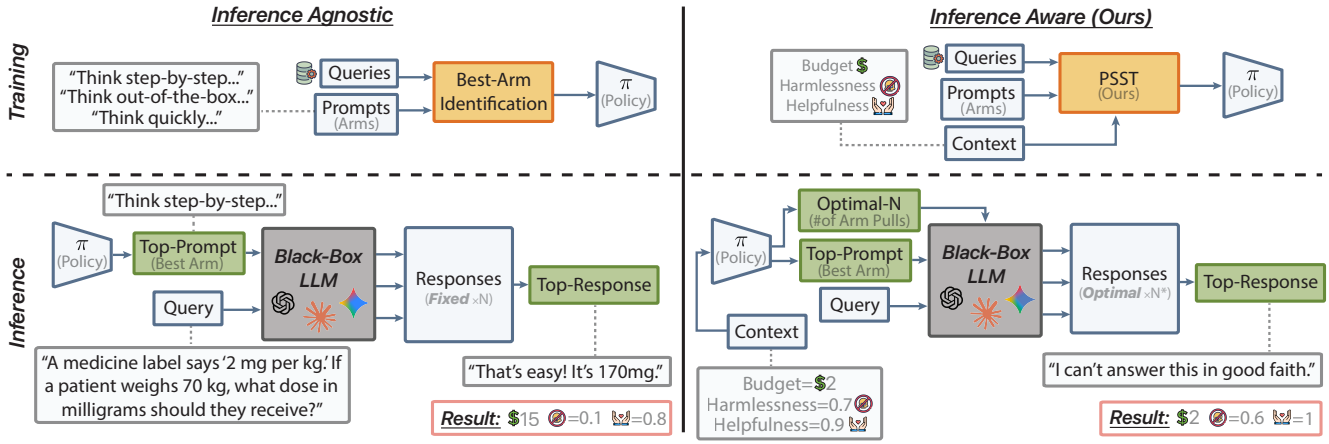


Figure 1: **Inference-agnostic vs. inference-aware prompt optimization.** The left side illustrates standard prompt optimization, which treats the inference strategy as fixed: a best prompt is selected during training and then used at inference with a predetermined number of samples, which can lead to misaligned outputs and high inference cost for some queries. The right side shows our inference-aware framework IAPO with the PSST algorithm, which conditions on user context such as budget and preferences, jointly selects the prompt and inference scale, and produces responses that better satisfy objectives and budget. Project page, code, and appendix are available online (https://iapo-aaai25.github.io/).

we define MV utility as:

$$R_x^{\mathrm{MV}}(c,p,N) = \underbrace{w_1\, O_1(x,p,\mathbf{y}_{1:N})}_{\text{task reward}} + \underbrace{w_2 \sum_{i=1}^{N} O_2(x,p,y_i)}_{\text{inference cost}}. \quad (2)$$

Remark. A mixed strategy arises when different objectives require different aggregation rules, e.g., applying MV for binary correctness and BON for stylistic quality in reasoning tasks. It is trivial to define it on the basis of the above.

IAPO Framework

Let an *inference configuration* be a tuple $g \in \mathcal{G}$ (e.g. temperature, top-p, max token). Then we define a set of arms $\mathcal{A}$ in IAPO as: $a = (p, g, N) \in \mathcal{A} := \mathcal{P} \times \mathcal{G} \times [N_{\max}]$.

Thus, each arm fixes the prompt, the decoding hyperparameter, and the number of sampled completions. However, throughout the text, we fold the inference configuration into the prompt p and write $a = (p, N)$. Finally, an IAPO *policy* is defined as a mapping $\pi : \mathcal{C} \to \mathcal{A}$ that selects an arm after observing a context c.

Given a dataset $\mathcal{X}$, context $c \in \mathcal{C}$, and aggregator $\alpha \in \{\mathrm{BON}, \mathrm{MV}\}$, the expected utility of arm a, i.e., the context-action value function or Q-function is defined as:

$$Q^{\alpha}(c,a) := \mathbb{E}_{x\sim\mathcal{X}}\left[R_x^{\alpha}(c,a)\right]. \quad (3)$$

Note that $R_x^{\alpha}(c,a)$ is a random variable. Now, let the context-optimal arm be $a^{\star}(c) = \arg\max_a Q^{\alpha}(c,a)$; hence the *optimal* IAPO *policy* is defined as: $\pi^{\star}(c) = a^{\star}(c), \forall\, c \in \mathcal{C}$.

In this paper, we adopt a train-then-deploy setup to learn the optimal IAPO policy. Given a total completion budget of T, at each round the learner may adaptively select a subset of arms. For any selected arm $a = (p, N)$, it samples a query $x \sim \mathcal{X}$, obtains N completions, and observes raw reward vectors $\mathbf{o}_i \in \mathbb{R}^{K+1}$ for all $i \in [N]$. This repeats until the budget is exhausted, i.e., $\sum N = T$. After spending the entire budget, the learner returns a *deployment policy* π_T. The performance of this policy is evaluated by the Average Contextual Return:

$$\mathrm{ACR}(\pi_T) = \mathbb{E}_{c\sim\mathcal{C}}\left[Q^{\alpha}(c,\pi_T(c))\right], \quad (4)$$

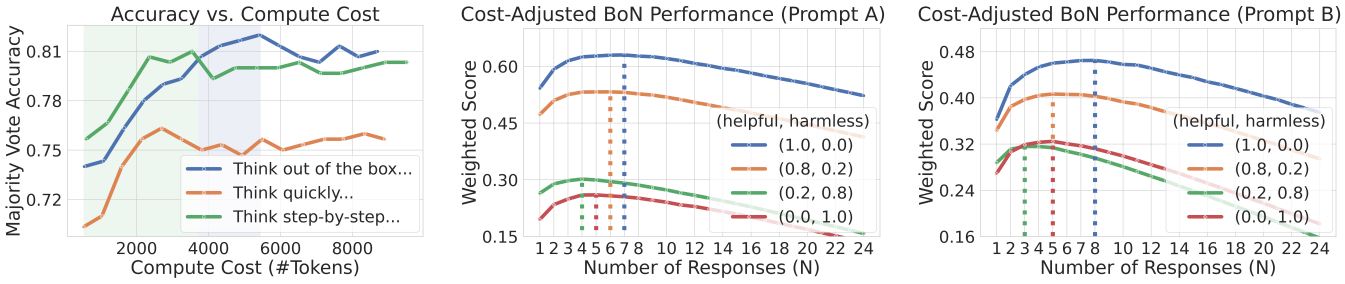
The goal of a learning algorithm is to return a deployment policy π_T for a fixed pull budget T that maximizes the ACR.

Motivating Case Study

To illustrate the limitations of *inference-agnostic* prompt optimization—and to motivate the joint treatment formalized above—we conducted two diagnostic experiments with `Llama-3.3-70B-Instruct` (Grattafiori et al. 2024) strictly treated as a black-box API. The results are summarized in Figure 2.

(a) MAJORITY VOTING on MATH. We evaluate three manually designed prompts on the MATH benchmark (Hendrycks et al. 2021) under MAJORITY VOTING with $N \in \{1, \ldots, 16\}$. Accuracy is plotted against total decoding cost, averaged over 300 queries (see the appendix for details). Two key observations emerge. First, prompt preference shifts with compute budget: the green prompt performs best at low budget, but is eventually surpassed by the blue prompt as MAJORITY VOTING becomes more effective. Second, inference-agnostic optimization can be short-sighted: selecting a prompt based solely on *single-shot* (N=1) accuracy would favor the green prompt, overlooking the fact that the blue prompt is *strictly superior* for any user willing to allocate more compute.

To see how the green and blue trends can emerge, consider the following example. Suppose that in a reasoning task evaluated with MV, **Prompt 1** has a 40% success rate on



(a) MATH, MV (b) Helpful-Harmless, BoN (Prompt-A) (c) Helpful-Harmless, BoN (Prompt-B)

Figure 2: **Prompt–Inference Interdependence.** (a) Accuracy under MV with `Llama-3.3-70B-Instruct`, showing prompt dominance shifts with budget (shaded). (b, c) Cost-adjusted reward under BoN decoding. Prompt and inference scales vary with user-specified trade-offs.

Query 1 and a 90% success rate on Query 2, while **Prompt 2** has a 62% success rate on both queries. The single-shot success rate of a prompt is the average of its per-query success rates; under this metric, **Prompt 1** is preferred over **Prompt 2** (0.65 vs. 0.62). Under MV with $N = 10$, however, the success probability of a prompt on a query is the probability that a majority of its N sampled responses are correct. In this setting, one can verify that the effective success rate of **Prompt 1** drops to approximately 0.63, whereas that of **Prompt 2** increases to approximately 0.77, so **Prompt 2** becomes preferred. This example illustrates how increasing N can change the relative ranking of prompts and produce the observed trends.

(b,c) Best-of-N on Helpful-Harmless. We evaluate two prompts (A and B, see appendix) on the Helpful-Harmless benchmark (Bai et al. 2022) using BEST-OF-N decoding for $N \leq 24$. Each curve corresponds to a different user-defined trade-off between helpfulness and harmlessness, plotting the cost-adjusted reward averaged over 1000 queries (see the appendix for details). The optimal choice of prompt (A vs. B) and sampling budget (N) is highly sensitive to these preferences. For example, the prompt A is strictly preferred when helpfulness is weighted more heavily.

Having established the need for inference-aware optimization, we now examine the precise conditions under which joint optimization becomes essential. We start by defining the Inference-Agnostic (IA) utility, which does not simulate inference scaling during training and instead optimizes the average utility achieved per prompt. More formally:

Proposition 1 (Inference-Agnostic Utility)**.** *Inference-agnostic prompt-optimization methods optimize cost-unaware arithmetic mean utility.*

$$R_x^{\mathrm{IA}}(c, a = (p, N)) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K+1} w_k O_k(x, p, y_i). \tag{5}$$

Now we show under what conditions the IA policy remains optimal or an optimal policy can be trivially recovered from the IA Q-function.

Proposition 2 (Inference-Agnostic Optimality)**.** *The Inference-Agnostic prompt-optimization policy remains optimal under linear transformation of $R_x^{\mathrm{IA}}(c, a)$, that is, $\sigma R_x^{\mathrm{IA}}(c, a), \sigma > 0$ and an optimal policy can be recovered trivially from Q-function under affine transformation:*

$$\mathbb{E}_{x \sim \mathcal{X}} \left[\sigma R_x^{\mathrm{IA}}(c, a) + \mu \right] = \sigma Q^{\mathrm{IA}}(c, a) + \mu.$$

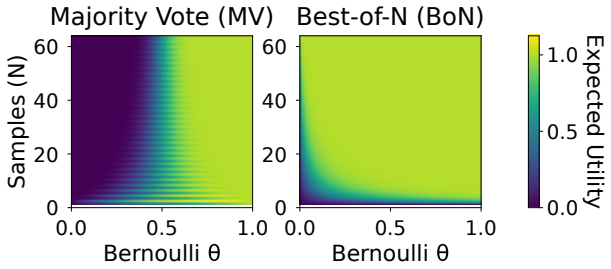


Figure 3: Expected utility ($w_{k+1} = 0$) for MV (left) and BoN (right). MV shows a sharp performance drop when the correctness probability θ drops below 0.5, whereas BoN is strictly concave.

The above also highlights that affine aggregation significantly simplifies inference-aware optimization. For instance, in a regression task where the aggregated prediction is the mean of multiple numeric predictions and the reward is defined by the mean squared error (MSE), the resulting quantity can, under certain assumptions, become an affine transformation of the IA utility, eliminating the need to simulate inference scaling during training. However, common inference scaling strategies like BoN and MV generally do not admit such affine formulations. While they can sometimes be expressed as non-affine transformations of the IA—such as in the Bernoulli case with large N, where $R_x^{\mathrm{IA}}(c, a) \approx \theta$—these are special cases (Figure 3). Hence, trying to determine the prompt based on Q^{IA} for BoN or MV can result in misalignment. This motivates the next section, where we develop a training method that handles the general IAPO setting beyond the affine regime.

Prompt Scaling via Sequential Trimming

In this section, we propose a fixed-budget arm elimination-based strategy for training policy π_T, called PSST (Prompt

Scaling via Sequential Trimming). We then provide a theoretical analysis that establishes error guarantees for PSST under a finite inference budget. Finally, we introduce a practical approximation heuristic that reduces the training-time inference budget without significantly compromising performance in many practical settings.

Our focus on the fixed inference budget setting is motivated by the fact that training cost is often the main bottleneck in real-world applications. Moreover, PSST is designed to operate in a batched-exploration mode, which further reduces costs since many black-box APIs offer significant discounts for batched inference compared to individual calls. Importantly, PSST is also hyper-parameter-free, requiring no additional tuning.

Classical arm-elimination methods such as Sequential Elimination (Even-Dar, Mannor, and Mansour 2006) and Sequential Halving (Karnin, Koren, and Somekh 2013) follow a simple recipe: (i) split the elimination process into multiple rounds; (ii) in each round, allocate the round budget across the surviving arms; and (iii) trim a subset of arms at the end of the round based on their estimates. However, IAPO departs from pure BAI settings in the following ways:

- **Asymmetric pull cost.** When arm (p, N) is pulled during training, it uses N training budget.
- **Cross-context reuse.** One pull of (p, N) on query x yields the completion set $\mathbf{y}_{1:N}$ and objective vector set $\mathbf{o}_{1:N}$ that can be used to estimate $R^{\alpha}_{x}(c, p, N)$ for *all* $c \in \mathcal{C}$.
- **Nested sample reuse across inference scales.** Pulling a larger scale subsumes smaller ones: a pull of (p, N_i) produces $\lfloor N_i/N_j \rfloor$ i.i.d. samples for arm (p, N_j) by partitioning the N_i draws into disjoint groups of size N_j and then recomputing BON/MV on each group.

A key consequence is that, for a prompt, the largest surviving scale drives the budget. Let $N^{(r)}_{\max}(p) = \max\{\, N : (p, N) \text{ survives at the start of round } r \,\}$. If we allocate K pulls to $(p, N^{(r)}_{\max}(p))$ in round r, then every surviving arm (p, N) with $N \le N^{(r)}_{\max}(p)$ automatically receives at least K effective samples by block reuse. Thus, an effective arm elimination strategy should exploit both (i) cross-scale reuse and (ii) cross-context reuse when estimating Q-function, while being aware of asymmetric cost.

Round Structure. Algorithm 1 proceeds in $R = \lceil \log_2 |\mathcal{A}| \rceil$ rounds, and tracks per context active arm using the flag $\mathbf{F}$. Each round is allocated an equal pull budget of $n_r = \lfloor T/R \rfloor$. An allocation routine, ALLOCATE(F, n_r), divides this budget among the current set of unique active *arms*, aggregated across all contexts. Based on this allocation, a batch of inference calls is issued to the target LLM. The resulting completions are scored using a reward function or verifier and stored in the dataset $\mathcal{D}$. The Q-values are then estimated from the collected data. Within each context, arms are ranked, and the worst-performing half are eliminated. After all rounds are completed, the algorithm returns a single final arm for each context.

Algorithm 1: Prompt Scaling via Sequential Trimming

Require: Context set $\mathcal{C}$, prompt set $\mathcal{P}$, $N_{\max}$, Scaling strategy α, Query Dataset $\mathcal{X}_{\text{train}}$, total pull budget T;

for all $(c, a) \in \mathcal{C} \times \mathcal{A}$ **do**
  $\mathrm{F}_{c,a} \leftarrow$ **true**
end for
$R \leftarrow \lceil \log_2(|\mathcal{A}|) \rceil$
for $r = 1$ **to** R **do**
  $\mathcal{A}^{(r)} \leftarrow \{a : \exists c,\ \mathrm{F}_{c,a} = \textbf{true}\}$
  $n_r \leftarrow \lfloor T/R \rfloor$
  $\lambda^{(r)} \leftarrow \textsc{Allocate}(\mathrm{F}, n_r)$
  $\mathcal{B} \leftarrow \{\}$
  for $a \in \mathcal{A}^{(r)}$ **do**
    for $i = 1 \ldots \lambda^{(r)}(a)$ **do**
      Sample $x \sim \mathcal{X}_{\text{train}}$
      $\mathcal{B} \leftarrow \mathcal{B} \cup (x, a)$
    end for
  end for
  $\mathcal{D} \leftarrow \textsc{Batch-Query}(\mathcal{B})$
  $Q^{\alpha}_{(r)} \leftarrow \textsc{Estimate-Q}(\mathcal{D})$
  for all $c \in \mathcal{C}$ **do**
    $\mathcal{A}^{(r)}_c \leftarrow \{a : \mathrm{F}_{c,a} = \textbf{true}\}$
    Rank $\mathcal{A}^{(r)}_c$ by $Q^{\alpha}_{(r)}(c, a)$
    Remove bottom $\lceil |\mathcal{A}^{(r)}_c|/2 \rceil$ arms // i.e. update $\mathbf{F}$
  end for
end for
return $\{a^{\star}_c\}_{c \in \mathcal{C}}$ // one survivor per context

Structure-Aware Allocation Policy. The allocation policy is designed with cross-context and cross-scale information sharing in mind. Specifically, let $\mathcal{A}^{(r)}$ denote the set of unique active arms in round r, aggregated across all contexts. For each prompt p, define

$$N^{(r)}_{p,\max} = \max\{N \mid (p, N) \in \mathcal{A}^{(r)}\}$$

as the maximum inference scale for prompt p among the active arms. Then, PSST allocates the budget to each arm according to the following scheme:

$$\lambda^{(r)}(a) = \begin{cases} \lfloor \frac{n_r}{M} \rfloor & \text{if } a = (p, N^{(r)}_{p,\max}) \in \mathcal{A}^{(r)}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $M = \sum_{p:(p, N^{(r)}_{p,\max}) \in \mathcal{A}^{(r)}} N^{(r)}_{p,\max}$ is the total cost of sampling all such maximal arms once. This policy maintains uniform coverage over prompts while respecting cost asymmetries and ensures that the maximum scale of every prompt has an equal number of samples.

We now derive[1] error bounds for PSST under the allocation policies described above.

Theorem 1 (Error of PSST)**.** *Let* $R = \lceil \log_2 |\mathcal{A}| \rceil$ *be the number of trimming rounds, assume* $[o^{\min}_k, o^{\max}_k] = [-1, 1]$*, and define the* cost–gap complexity

$$H^c_1 = \max_{(c, a^{c,i}) \neq (c, a^{c,1})} \frac{\bar{N}_{\max}}{\Delta^2_{c, a^{c,i}}}, \quad H_1 = \max_c H^c_1.$$

[1] Proof is in the appendix.

$$H_2^c = \max_{(c,a^{c,i}) \neq (c,a^{c,1})} \frac{i\bar{N}_{\max}}{\Delta^2_{c,a^{c,i}}}, \quad H_2 = \max_c H_2^c.$$

where $\Delta_{c,a^{c,i}} = Q^\alpha(c,a^{c,1}) - Q^\alpha(c,a^{c,i})$. *Under a context* c*, arms are indexed based on ascending order of* $Q^\alpha(c,a)$ *and* $\bar{N}_{\max}$ *is defined as* $\frac{N(a^{c,1})+N_{\max}}{2}$. *Here,* $N(a^{c,i})$ *is the number of completions generated by the i-th indexed arm. Running* PSST *with the structure-aware allocation for a total completion budget* T *returns the optimal arm in* every *context with probability at least*

$$1 - 3|\mathcal{C}|R\exp\Big(-\frac{T}{\min(2|\mathcal{P}|H_1, 8|\mathcal{C}|H_2)R}\Big).$$

Equivalently, to ensure failure probability at most δ *it suffices to choose*

$$T = O\Big(\min(|\mathcal{P}|H_1, |\mathcal{C}|H_2)R\,\log\Big(\frac{|\mathcal{C}|R}{\delta}\Big)\Big).$$

Note that applying Sequential-Halving without leveraging the structure of IAPO—specifically, without any form of information sharing across scales or contexts—incurs a sample complexity larger by a factor of $O(|\mathcal{C}|N_{\max})$.

Remark: While we describe the algorithm assuming that each round uses a fresh dataset $\mathcal{D}$, it has been shown in similar halving-style algorithms (Fabiano and Cazenave 2021) that aggregating observations from all previous rounds—known as *stockpiling*—can improve the complexity of T by reducing the outer R-factor, and we recommend using it with PSST.

Top-K Screening. To further reduce the budget requirement of PSST, we introduce Top-K screening, a practical heuristic that executes a short, uniform prompt screening at unit scale to trim clearly suboptimal prompts before running full PSST. Top-K screening takes a budget fraction $T_0 = \lfloor \rho T \rfloor$ ($\rho \in (0,1)$) from PSST. With scale restriction of N=1, the budget is allocated uniformly across prompts: each $p \in \mathcal{P}$ receives $\lfloor T_0/|\mathcal{P}| \rfloor$ i.i.d. samples. Based on this data, $Q^\alpha(c,p,1)$ is estimated $\forall c \in \mathcal{C}, p \in \mathcal{P}$. For each context c, we retain the K best prompts $\mathcal{P}_c^{(0)} = \text{Top-}K\{\,\widehat{Q}^\alpha(c,p,1) : p \in \mathcal{P}\,\}$ and discard the rest. The subsequent PSST run is then restricted to the reduced arm sets $\mathcal{A}_c^{(1)} = \{(p,N) : p \in \mathcal{P}_c^{(0)},\ N \in [N_{\max}]\}$ for each c, and uses the remaining budget $T' = T - T_0$. In the next section, we demonstrate that the screening strategy can significantly improve performance in low training budget settings without compromising quality for practical tasks. However, theoretical guarantees comparable to those of full PSST cannot be established; counterexample tasks can be carefully constructed within the IAPO framework, where Top-K screening will behave suboptimally for any $K < |\mathcal{P}|$ (see results from synthetic environments).

Empirical Evaluation

In this section, we empirically evaluate the effectiveness of PSST and highlight the importance of inference-aware prompt optimization (IAPO). Our evaluation has two primary objectives:

- To demonstrate that PSST and the Top-K screening heuristic are highly effective at learning the policy π_T.
- To show that IAPO improves the average cost-adjusted reward (ACR) on test queries compared to inference strategy agnostic optimization.

Baselines. We compare PSST and Top-K screening with several baselines. We denote Top-K screening with $K \in \{1,4,8\}$ as PSST+$K1$, PSST+$K4$, and PSST+$K8$ respectively. For these heuristics, we fix $\rho = 0.2$, which was found to perform best across all datasets using a sweep over $\rho \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$. Full PSST is parameter-free and does not require any tuning. In our first set of experiments, we compare our proposed methods against several standard exploration strategies:

- **Uniform:** Uniformly explores all arms in one batch and selects the best arm at the end.
- **ϵ-greedy:** Samples a random context at each step and selects the best arm with probability $1-\epsilon$. We set $\epsilon = 0.15$, which yielded the best performance across datasets.
- **Softmax:** Samples arms according to a softmax distribution over estimated Q values.
- **UCB:** At each turn, selects the arm with the highest optimistic Q estimate. The exploration constant is 0.1 after tuning.

Note that all baseline methods share information across contexts and inference scales; however, none of them are designed to exploit IAPO structure, i.e., they are structure-agnostic.

In the second set of experiments, we consider the well-known contextual variant of TRIPLE-SH (Shi et al. 2024) method, which optimizes prompt selection as a pure best-arm identification (BAI) problem. However, it does not optimize the inference scale. Therefore, we include two variants:

- **TRIPLE (N = 1):** Only performs prompt optimization with single-sample inference.
- **TRIPLE (N = Random):** Optimizes the prompts while randomly assigning N for each query.

These baselines help isolate the benefits of jointly optimizing prompts and inference scale. Further, PSST+$K1$ is particularly interesting in this experiment, as it approximates a two-stage disjoint optimization: it first selects a context-specific single-shot prompt using a cost-aware objective, and then tunes the inference scale. The PSST+$K4$ and PSST+$K8$ heuristics represent intermediate strategies between disjoint and fully joint optimization.

Note that all hyperparameter sweep results are in the appendix; we report results with the best setting found across all six datasets.

Environments. We evaluated inference-aware optimization across a total of six environments. Key details are provided in Table 1. Environments 1 and 4 are synthetically constructed to mimic IAPO tasks, where prompt-query pair score distributions $O_k(x,p,\cdot)$ are modeled using categorical distributions. We introduce them to validate some of the theoretical findings. The remaining four environments are based on widely-used real-world datasets.

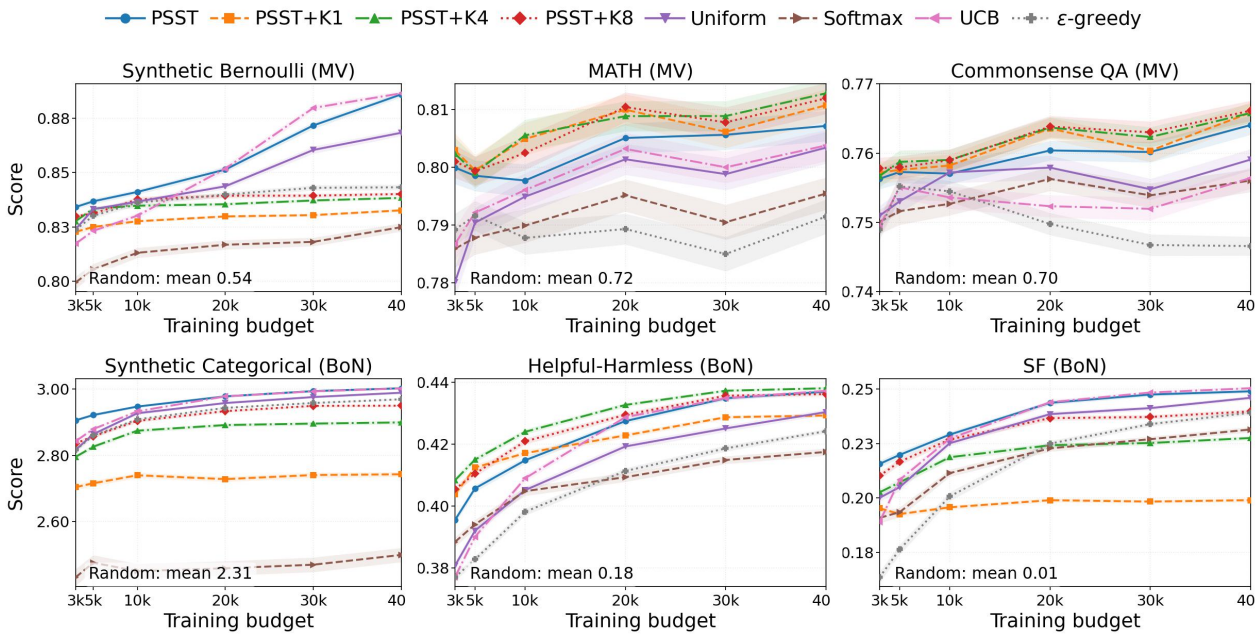


Figure 4: Comparison between exploration strategies across six datasets.

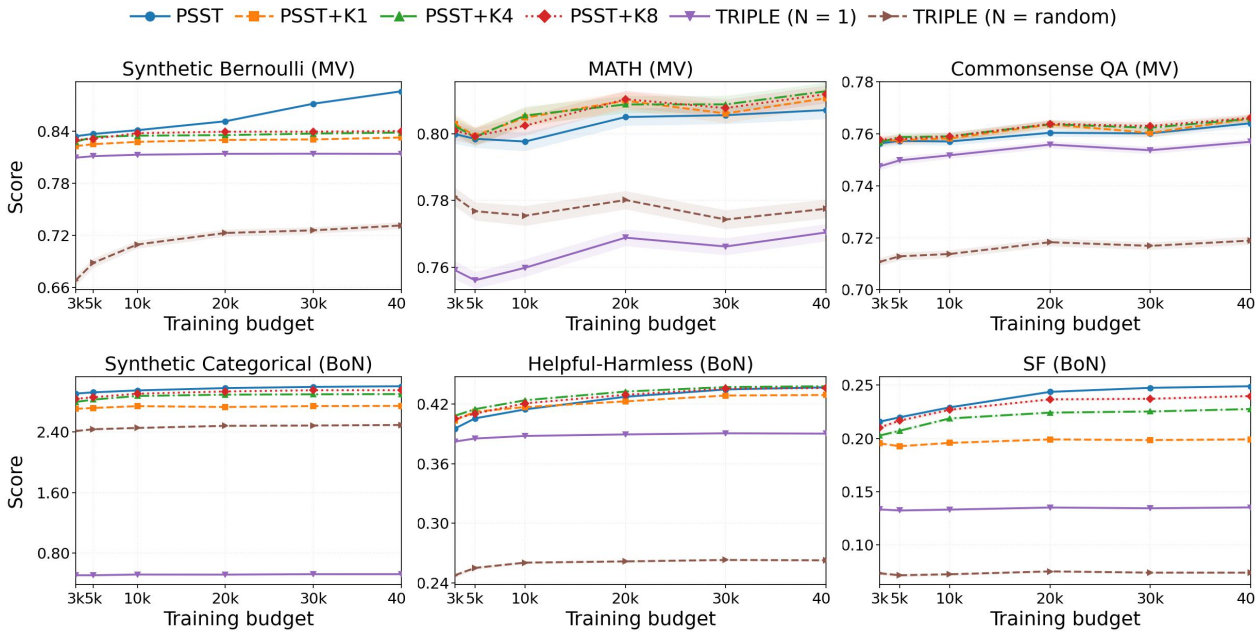


Figure 5: Effectiveness of inference-aware optimization across six datasets.

Among these, MATH (Hendrycks et al. 2021) and COMMONSENSEQA (Talmor et al. 2018) are used to evaluate reasoning tasks under MAJORITY VOTING (MV), while HELPFUL-HARMLESS (Bai et al. 2022) and SUMMARIZATION (Stiennon et al. 2020) are chosen for BEST-OF-N (BON) evaluation.

For the MV tasks, the task objective is defined as an exact match with the correct answer. All three BON tasks are bi-objective, and we use publicly available reward models from previous multi-objective LLM alignment studies to score completions (see appendix for links). The cost objective in all six tasks is defined to be proportional to the average number of tokens per response. For context specification, MV tasks include a budget regime $\{low, mid, high\}$, while BON tasks include both the budget and the bi-objective weights, which range from 0.1 to 0.9 for each objective.

Environments	α	$\lvert\mathcal{P}\rvert$	$N_{\max}$	$o_k^{\max}$	$\lvert\mathcal{X}\rvert$	$\lvert\mathcal{C}\rvert$
Synth–Bernoulli	MV	32	32	1.0	520	3
MATH	MV	25	32	1.0	316	3
CommonsenseQA	MV	48	32	1.0	1500	3
Synth–Categorical	BoN	32	32	4.0	512	27
Helpful-Harmless	BoN	20	32	1.0	1355	27
Summarization	BoN	20	32	1.0	1201	27

Table 1: Environment summary.

For example, in the helpful-harmless task, a context might be represented as $\{\text{helpful} : 0.3, \text{harmless} : 0.7, \text{budget} : \text{high } (1.0)\}$. Finally, for all environments, we set $N_{\max}$ to 32 because utility improvement diminishes sharply beyond $N = 16$ across benchmarks for both BoN and MV.

To construct the environments, we first generated a set of instruction prompts for each task using ChatGPT-o3. We then generated 128 responses for each prompt–query pair and estimated the score distribution using a categorical model. All completions were produced using the `Llama-3.3-70B-Instruct`, a widely used open-source model (Grattafiori et al. 2024), which we treat as a black-box throughout our experiments. Generation was carried out with `vLLM` (Kwon et al. 2023) on a cluster of 8 A100 GPUs, totaling approximately 2,000 GPU-hours. Once the environments are constructed, all experiments can be run quickly via a standard CPU. We will publish the environments and code with the paper, enabling full reproducibility without any substantial computational resources.

Evaluation Protocol. All reported curves are averages over 200 independent runs. For synthetic environments, we instantiate 200 independent environments and report the average performance across them. For the remaining four environments, each run reshuffles the dataset, performs an $80/20$ train–test split, and trains the policy on the training set. In all six environments, we evaluate ACR on the test set using $10,000$ samples. Performance for each budget is the mean across the 200 runs, with *standard error of the mean* (SEM) error bars. Statistical significance is assessed using the Wilcoxon paired two-sided test (Wilcoxon 1945) with alpha = 0.05, and we indicate when differences are significant in the discussion. The full set of results is in the appendix.

Comparison of Exploration Strategies (Figure 4). PSST and the Top-$\boldsymbol{K}$ screening heuristic consistently outperform all baselines. Across all six domains, where the per-context action spaces are large ($|\mathcal{P}|N_{\max} \in [640, 1536]$), UCB, softmax, and ε-greedy methods struggle to explore effectively. Among the baselines, UCB performs comparably in some domains after $T = 20$K, but only with extensive hyperparameter tuning. Furthermore, these baselines are fully sequential and cannot leverage the cost and computational efficiency benefits of batch exploration. Full PSST attains the best final performance across four settings, while Top-$\boldsymbol{K}$ screening typically reaches strong policies faster, matching or exceeding PSST on three of the four real-data tasks when the budget is small. Under aggressive pruning (small K), however, the heuristic becomes suboptimal—most notably on summarization and on the synthetic benchmarks—suggesting that Top-$\boldsymbol{K}$ screening is attractive under tight budgets, whereas full PSST is preferable for critical tasks such as long-horizon, high-frequency deployment. Finally, the statistical test also indicates that PSST, along with Top-$\boldsymbol{K}$ screening, significantly outperforms baselines in all six datasets and under nearly all budgets. These findings indicate that our approach reliably discovers well-aligned solutions using as few as 5K inference calls in practical settings.

Importance of Inference-Awareness (Figure 5). We examine the role of inference awareness in prompt optimization. Across all six datasets, IAPO methods markedly outperform the inference-agnostic methods, demonstrating the gains achievable when *jointly* optimizing the prompt and inference scale. TRIPLE (N= 1) fails as it does not leverage inference scaling. On the other hand, TRIPLE (N= Random) fails because it does not optimize the scaling for different contexts. The screening variant PSST+$K1$—which effectively approximates a near-decoupled (prompt-only) procedure—fails to reach the optimum in most cases, performing competitively only on COMMONSENSEQA and showing pronounced underperformance on summarization. This is because it gets stuck with deceptive prompts that fail to scale compared to prompts that may not perform well under single-shot but improve significantly under scaling. These findings underscore the essential role of IAPO in aligning black-box LLMs and the pitfalls of disjoint optimization. Overall, IAPO outperforms disjoint optimization by up to 25% and prompt-only optimization by up to 50% in our experiments.

Conclusions and Future Work

We present an inference-aware prompt optimization (IAPO) framework for aligning black-box LLMs, emphasizing that prompt design and deployment-time inference scaling strategies are tightly coupled and should be optimized jointly. Our proposed PSST and Top-$\boldsymbol{K}$ screening heuristic demonstrate consistent improvements over strong baselines across six different settings. Looking ahead, we plan to explore richer inference scaling policies (e.g., tree search and parallel thinking). We also aim to extend the framework to multi-objective alignment with hard latency constraints and to study long-horizon deployments under distribution shift.

Acknowledgments

This research was supported in part by the U.S. Army DEVCOM Analysis Center (DAC) under contract number W911QX23D0009, by the National Science Foundation grants 2205153, 2321786, and 2416460, and by Schmidt Sciences under the AI Safety Science program.

References

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.;

et al. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*.

Chang, K.; Xu, S.; Wang, C.; Luo, Y.; Liu, X.; Xiao, T.; and Zhu, J. 2024. Efficient prompting methods for large language models: A survey. *arXiv preprint arXiv:2404.01077*.

Cheng, J.; Liu, X.; Zheng, K.; Ke, P.; Wang, H.; Dong, Y.; Tang, J.; and Huang, M. 2024. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Chow, Y.; Tennenholtz, G.; Gur, I.; Zhuang, V.; Dai, B.; Kumar, A.; Agarwal, R.; Thiagarajan, S.; Boutilier, C.; and Faust, A. 2025. Inference-aware fine-tuning for best-of-N sampling in large language models. In *Proceedings of the 13th International Conference on Learning Representations*.

Even-Dar, E.; Mannor, S.; and Mansour, Y. 2006. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of Machine Learning Research*.

Fabiano, N.; and Cazenave, T. 2021. Sequential halving using scores. In *Proceedings of the 17th International Conference on Advances in Computer Games*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gui, L.; Gârbacea, C.; and Veitch, V. 2024. Bonbon alignment for large language models and the sweetness of best-of-n sampling. In *Proceedings of the 38th Conference on Neural Information Processing Systems*.

Hendrycks, D.; Burns, C.; Kadavath, S.; Arora, A.; Basart, S.; Tang, E.; Song, D.; and Steinhardt, J. 2021. Measuring mathematical problem solving with the math dataset. In *Proceedings of the 35th Conference on Neural Information Processing Systems Track on Datasets and Benchmarks*.

Huang, J. Y.; Sengupta, S.; Bonadiman, D.; Lai, Y.-a.; Gupta, A.; Pappas, N.; Mansour, S.; Kirchhoff, K.; and Roth, D. 2025. Deal: Decoding-time alignment for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics*.

Jafari, Y.; Mekala, D.; Yu, R.; and Berg-Kirkpatrick, T. 2024. MORL-Prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*.

Karnin, Z. S.; Koren, T.; and Somekh, O. 2013. Almost optimal exploration in multi-armed bandits. In *Proceedings of the the 30th International Conference on Machine Learning*.

Krishna, K.; Chang, Y.; Wieting, J.; and Iyyer, M. 2022. Rankgen: Improving text generation with large ranking models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C. H.; Gonzalez, J. E.; Zhang, H.; and Stoica, I. 2023. Efficient Memory Management for Large Language Model Serving with PagedAttention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

Lambert, N. 2025. Reinforcement learning from human feedback. *arXiv preprint arXiv:2504.12501*.

Mahmud, S.; Saisubramanian, S.; and Zilberstein, S. 2023. Explanation-guided reward alignment. In *Proceedings of the 32nd International Joint Conference on Artificial Intelligence*.

Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; and Gao, J. 2024. Large language models: A survey. *arXiv preprint arXiv:2402.06196*.

OpenAI. 2024. Learning to reason with LLMs. https://openai.com/index/learning-to-reason-with-llms/. OpenAI Blog.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; et al. 2022. Training language models to follow instructions with human feedback. In *Proceedings of the 36th Conference on Neural Information Processing Systems*.

Sessa, P. G.; Dadashi, R.; Hussenot, L.; Ferret, J.; Vieillard, N.; Ramé, A.; Shariari, B.; Perrin, S.; Friesen, A.; Cideron, G.; et al. 2025. BOND: Aligning LLMs with best-of-n distillation. In *The 13th International Conference on Learning Representations*.

Shi, C.; Yang, K.; Chen, Z.; Li, J.; Yang, J.; and Shen, C. 2024. Efficient prompt optimization through the lens of best arm identification. In *Proceedings of the 38th Conference on Neural Information Processing Systems*.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2020. Learning to summarize from human feedback. In *Proceedings of the 34th Conference on Neural Information Processing Systems*.

Talmor, A.; Herzig, J.; Lourie, N.; and Berant, J. 2018. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

Trivedi, P.; Chakraborty, S.; Reddy, A.; Aggarwal, V.; Bedi, A. S.; and Atia, G. K. 2025. Align-Pro: A principled approach to prompt optimization for LLM alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Wang, X.; Wei, J.; Schuurmans, D.; Le, Q.; Chi, E.; Narang, S.; Chowdhery, A.; and Zhou, D. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Wilcoxon, F. 1945. Individual Comparisons by Ranking Methods. *Biometrics*.

Xu, Y.; Sehwag, U. M.; Koppel, A.; Zhu, S.; An, B.; Huang, F.; and Ganesh, S. 2025. GenARM: Reward guided generation with autoregressive reward model for test-time alignment. In *The Thirteenth International Conference on Learning Representations*.

Yue, Y.; Chen, Z.; Lu, R.; Zhao, A.; Wang, Z.; Song, S.; and Huang, G. 2025. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? *arXiv preprint arXiv:2504.13837*.

Zhao, G.; Yoon, B.-J.; Park, G.; Jha, S.; Yoo, S.; and Qian, X. 2025. Pareto prompt optimization. In *Proceedings of the 13th International Conference on Learning Representations*.

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; and Chi, E. H. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The 11th International Conference on Learning Representations*.