EFFICIENT MULTI-OBJECTIVE PROMPT OPTIMIZATION VIA PURE-EXPLORATION BANDITS

Donghao Li* **Chengshuai Shi**† **Weijuan Ou**$^{\diamond}$ **Cong Shen*** **Jing Yang***
* University of Virginia, Charlottesville, VA 22904, USA
† Princeton University, Princeton, NJ 08544, USA
$^{\diamond}$ Southern University of Science and Technology, Shenzhen, Guangdong 518055, China

ABSTRACT

Prompt engineering has become central to eliciting the capabilities of large language models (LLMs). At its core lies *prompt selection* - efficiently identifying the most effective prompts. However, most prior investigations overlook a key challenge: the inherently multi-faceted nature of prompt performance, which cannot be captured by a single metric. To fill this gap, we study the multi-objective prompt selection problem under two practical settings: Pareto prompt set recovery and best feasible prompt identification. Casting the problem into the pure-exploration bandits framework, we adapt provably efficient algorithms from multi-objective bandits and further introduce a novel design for best feasible arm identification in structured bandits, with theoretical guarantees on the identification error in the linear case. Extensive experiments across multiple LLMs show that the bandit-based approaches yield significant improvements over baselines, establishing a principled and efficient framework for multi-objective prompt optimization.

1 INTRODUCTION

Prompt engineering has become a practical way to leverage large language models (LLMs) without expensive, time-consuming fine-tuning (Sahoo et al., 2024; Schulhoff et al., 2024). Early results show that zero-shot (Radford et al., 2019) and few-shot prompting (Brown et al., 2020) with a small number of examples can elicit strong performance from frozen models. More recently, chain-of-thought (CoT) prompting (Wei et al., 2022; Kojima et al., 2022; Zhang et al., 2023) has further unlocked step-by-step reasoning, matching or even surpassing task-specific fine-tuning on several complex benchmarks. Prompting has also become foundational across adjacent areas, including retrieval-augmented generation (RAG) (Kang et al., 2024), text-to-image generation (Brade et al., 2023; Mo et al., 2024), and jailbreak analysis and defenses (Yan et al., 2024; Mehrotra et al., 2024; Xu & Parhi, 2025).

Generally speaking, prompt engineering comprises *manual prompt design*, in which experts craft prompts through intuition and iteration, and *automatic prompt optimization*, in which algorithms search the prompt space systematically. Examples of the latter include evolutionary search for high-performing prompts (Guo et al.); gradient-based methods such as AutoPrompt and APO (Shin et al., 2020; Pryzant et al., 2023); reinforcement learning approaches that cast prompt construction as sequential decision making (Deng et al., 2022); and methods that use an LLM itself as the optimizer (Cheng et al., 2024; Tang et al., 2025). Despite their algorithmic differences, these methods share a core challenge: prompt selection, i.e., *given a finite candidate prompt set $\mathcal{X}$ and a limited evaluation budget B, how should one allocate queries to identify the prompts that maximize task performance?*

Prior work on prompt selection spans Bayesian optimization (Chen et al., 2024; Sabbatella et al., 2024), discrete search (Hu et al., 2024), and bandit formulations (Shi et al., 2024; Lin et al., 2024; Lin et al.; Kong et al., 2025). Despite these advances, existing methods predominantly optimize a *single* objective (e.g., task accuracy), leaving broader trade-offs among multiple objectives largely addressed. However, in many real-world applications, prompt performance is inherently multi-faceted, involving *multiple* objectives rather than a single metric. For example, in text summarization tasks, human and benchmark assessments consider coherence, faithfulness, fluency, and relevance, and no single metric captures all dimensions (Fabbri et al., 2021). In text style transfer tasks, outputs must satisfy both content preservation and style adherence (e.g., modern-to-Shakespearean translation (Caldas et al., 2018) and politeness transfer (Madaan et al., 2020)). In these multi-objective settings, a single prompt usually cannot be universally superior across all metrics. The trade-offs among these objectives require moving beyond scalarized prompt selection to procedures that preserve multiple criteria throughout the evaluation.

Meanwhile, research on multi-objective bandits offers principled tools for trade-off exploration under fixed evaluation budgets, including algorithms that learn with multiple criteria and handle explicit metric constraints with instance-dependent guarantees (Auer et al., 2016; Kone et al., 2024; 2025; Faizal & Nair, 2022). Yet this toolkit has not been systematically applied to multi-objective prompt selection.

In this work, we aim to bridge these areas by *formulating multi-objective prompt selection within a bandit framework and developing a principled, bandit-based approach for efficient prompt selection under stringent prompt evaluation budget constraint*. Our major contributions are three-fold:

- First, we bridge prompt selection under multiple evaluation criteria with the framework of multi-objective bandits. To the best of our knowledge, this is the first attempt to formalize multi-criteria prompt selection as a multi-objective bandit problem. This connection allows us to move beyond ad-hoc or single-metric prompt selection strategies and instead leverage the rich toolbox of multi-objective bandit algorithms. By doing so, we obtain a principled and efficient framework for balancing diverse evaluation criteria, leading to more robust and systematic prompt selection.
- Secondly, within this framework, we investigate two fundamental problems: best feasible prompt identification and Pareto prompt set identification. For the former, we introduce a general algorithm, GENSEC, and provide a theoretical characterization of its error rate under the linear reward setting. For the latter, we propose another general algorithm, GENPSI, which unifies and generalizes existing bandit algorithms for both the standard and linear settings. Importantly, both GENSEC and GENPSI are designed to accommodate general shared structures among prompts, thereby enhancing learning efficiency, particularly when the evaluation budget is limited.
- We evaluate the performance of GENSEC and GENPSI on two summarization benchmarks: XSum and CNN/DailyMail. For best feasible prompt identification, GENSEC based algorithms recover over 80% and 90% of the utility of the optimal constrained prompt on the two tasks, respectively, whereas the baseline methods achieve only 20–50%. For Pareto prompt set identification, GENPSI based algorithms recover more than 90% of the hypervolume of the ground-truth Pareto set, while the baselines remain in the low-to-mid 80% range.

2 RELATED WORKS

Single-objective prompt selection. InstructZero (Chen et al., 2024) applies Bayesian optimization in the continuous space of soft prompts, while Sabbatella et al. (2024) uses Bayesian optimization over hard prompts by modeling prompts as n-gram phrases. ZOPO (Hu et al., 2024) specializes in selection from a fixed candidate set and argues that a locally optimal prompt can outperform generation-based algorithms. Bandit-based methods improve sample efficiency via adaptive allocation and elimination (Shi et al., 2024; Lin et al., 2024; Lin et al.; Kong et al., 2025): TRIPLE (Shi et al., 2024) casts selection as fixed-budget best-arm identification; INSTINCT (Lin et al., 2024) leverages NeuralUCB with transformer features to optimize instructions; APOHF (Lin et al.) frames human-in-the-loop selection as a dueling bandit over pairwise preferences; and EXPO (Kong et al., 2025) addresses non-stationarity in agentic settings with adversarial bandits.

Multi-objective prompt engineering. Multi-criteria prompt optimization has been receiving increasing attention due to its practical impact across various domains. Evolutionary approaches, such as EMO-Prompts (Baumann & Kramer, 2024), InstOptima (Yang & Li, 2023), and MOPO (Resendiz & Klinger, 2025) use LLM-driven mutation and crossover to generate candidate prompts with high-quality trade-offs. Beyond evolutionary search, MORL-Prompt (Jafari et al., 2024) adapts multi-objective RL with Pareto-aware policy gradient, and GEPA (Agrawal et al., 2025) combines natural-language reflection with Pareto-guided evolution. *While those work aim to achieve trade-*

offs among different metrics in prompt optimization, they mostly focus on prompt generation while adopting uniform sampling as the selection method.

For constrained prompt optimization, CAPO (Zehle et al., 2025) adds cost awareness by combining evolutionary search with a length penalty, yielding prompts that balance task accuracy against token usage. Co-Prompt (Cho et al., 2023) focuses on token-wise prompt generation optimization, and uses another discriminator model to evaluate the generated token. *To the best of our knowledge, there are currently no prompt optimization studies that consider hard constraints, which is a key focus of our work.*

Multi-objective bandits. *Pareto set identification* in multi-armed bandits was first studied in the fixed-confidence setting (Auer et al., 2016). In the fixed-budget regime, Kone et al. (2024) introduced Empirical Gap Elimination (EGE), and showed that the misidentification probability decays exponentially with the budget, achieving the optimal rate up to constants. Kone et al. (2025) further investigated the linear bandits setting, while Zuluaga et al. (2013; 2016) investigate the problem from a Bayesian perspective. In the *constrained bandits* setting, regret minimization was investigated in Kagrecha et al. (2023); Pacchiano et al. (2021), while pure exploration strategies have been studied in Faizal & Nair (2022); Bian & Tan (2025) recently. *In summary, there has been active research on multi-objective bandits, providing a rich set of tools for principled prompt optimization. However, existing work primarily focuses on bandits with simple structures, which limits their applicability in practical scenarios. Our work builds on the bandit framework and extends it with application-inspired designs to address these gaps.*

3 Multi-objective Prompt Selection

We first introduce the basic notations used in the prompt-assisted interactions with LLMs (Zhou et al., 2022; Chen et al., 2024; Shi et al., 2024). Let x be a prompt, $\mathcal{D} = \{(q, a)\}$ a task dataset consisting of inputs q and reference answers a, and $\mathcal{M}$ a black-box LLM that maps the prompt x together with an input q to a distribution over the output language space $\mathcal{Y}$. Given (x, q), the LLM generates outputs according to $y \sim \mathcal{M}(q; x)$, where $y \in \mathcal{Y}$.

Due to the multi-criteria nature of prompt performance (Baumann & Kramer, 2024; Yang & Li, 2023; Agrawal et al., 2025), we consider m objectives represented by evaluation functions

$$f_j : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}, \quad j = 1, \ldots, m, \tag{1}$$

where each f_j assigns a numerical score to an LLM output $y \in \mathcal{Y}$ given a reference answer $a \in \mathcal{Y}$. For each prompt $x \in \mathcal{X}$, its expected performance vector is

$$\mu(x) = \mathbb{E}_{(q,a)\sim\mathcal{D}}\, \mathbb{E}_{y\sim\mathcal{M}(q;x)}\big(f_1(y, a), \ldots, f_m(y, a)\big) \in \mathbb{R}^m. \tag{2}$$

Here, the inner expectation averages over the stochasticity of the LLM given a fixed input, while the outer expectation averages over the dataset. Thus $\mu(x)$ summarizes how prompt x performs across all objectives on average. In addition, we assume that all metrics are defined such that larger values indicate better performance.

Motivating example. Figure 1 illustrates the evaluation of prompts on the XSum dataset using the Llama3 model, where each point corresponds to a prompt's ROUGE score (x-axis) and brevity score (y-axis). These two metrics exhibit inherent trade-offs, motivating joint consideration for prompt optimization.

In this work, we focus on the problem of multi-criteria prompt selection: given a set of candidate prompts $\mathcal{X}$ and evaluation metrics f_j, the objective is to identify a subset of prompts from $\mathcal{X}$ that achieves the desired trade-offs. We consider this problem under the following two settings: (i) best feasible prompt identification , and (ii) Pareto prompt set identification.

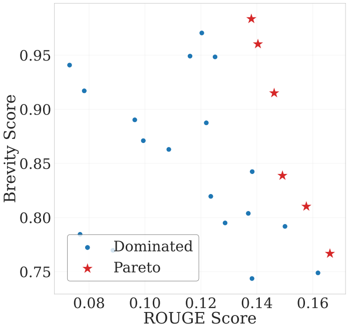


Figure 1: Trade-offs between ROUGE and Brevity.

Best feasible prompt identification. The first target arises when one objective is designated as the *primary* performance metric (e.g., task accuracy), while the remaining objectives act as constraints (e.g., safety above a threshold). Formally, let objective $j = 1$ be the primary objective to maximize, and denote the constraints on the other objectives as $\{\tau_j\}_{j \neq 1}$. A prompt x is *feasible* if

$$\mu_j(x) \geq \tau_j, \quad \forall j \neq 1. \tag{3}$$

The best feasible prompt is then defined as

$$x^\star = \arg\max_{x \in \mathcal{X}} \mu_1(x) \quad \text{s.t.} \quad \mu_j(x) \geq \tau_j, \forall j \neq 1. \tag{4}$$

This formulation is especially relevant for safe or regulated domains where utility must be achieved under explicit constraints.

Pareto prompt set identification. The second target is to recover the Pareto prompt set, which is the set of prompts that is not strictly dominated by any other prompt, as defined in the following.

Definition 1 (Pareto Optimality). *A prompt $x \in \mathcal{X}$ is* Pareto optimal *if there does not exist another prompt $x' \in \mathcal{X}$ such that $\mu_j(x') \geq \mu_j(x)$ for all $j \in \{1, \ldots, m\}$ and $\mu_j(x') > \mu_j(x)$ for at least one j.*

The set of Pareto-optimal prompts is denoted $\mathcal{X}^\star$, and $\{\mu(x) : x \in \mathcal{X}^\star\}$ is the *Pareto front*. In the absence of explicit constraints, Pareto set identification is the most general target for multi-objective optimization, which captures the best achievable trade-offs among objectives.

Fixed budget constraint. In practice, prompt evaluations are costly because each trial requires querying an LLM on multiple data examples and metrics. Therefore, in this work, we consider a fixed budget setting, where the total number of evaluations in the optimization procedure is upper bounded by B. Then, given a fixed budget constraint B, the learner's goal is to identify (i) the optimal feasible prompt x^*, or (ii) the Pareto set of prompts $\mathcal{X}^*$, as accurate as possible.

4 Prompt Optimization via Pure-exploration Bandits

Recent work (Shi et al., 2024) shows that prompt optimization can be formulated as a best arm identification (BAI) problem and solved efficiently by leveraging the rich toolbox from BAI in multi-armed bandits. While extensive experiments demonstrate the remarkable performance improvement of such an approach over baselines for a single performance metric, to the best of our knowledge, leveraging bandits algorithms to solve the more general multi-objective prompt optimization has not been studied previously. To fill this gap, in the following, we formulate the multi-objective prompt optimization problem under a fixed budget constraint as a pure-exploration multi-objective bandits.

Specifically, we model the prompt set $\mathcal{X}$ as the set of arms in a multi-armed bandits, and pulling an arm corresponds to evaluating a prompt $x \in \mathcal{X}$ on a randomly sampled input $(q, a) \in \mathcal{D}$, which produces an output $y \sim \mathcal{M}(q; x)$. The evaluation metrics $(f_1(y, a), f_2(y, a), \ldots, f_m(y, a))$ then serve as the stochastic *reward vector* observed for that pull. Thus, the expected reward of each arm is denoted as $\mu(x)$, which captures the mean performance of prompt x across all objectives.

Under the given budget constraint B, our objective is to leverage the algorithms from pure-exploration bandits to efficiently identify the optimal feasible prompt and the Pareto set of prompts (arms), respectively.

In practice, the candidate prompt set could be very large. Treating each prompt independently may not be very cost efficient, especially when the total budget is limited. On the hand other, the candidate prompts may inherently have certain correlations, and exploiting such correlation can potentially speed up the learning process and improve the learning accuracy. Motivated by those observations, we introduce a general bandits model to capture such inherent dependency among prompts.

Specifically, we let $\phi : \mathcal{X} \to \mathbb{R}^d$ be a known feature map that embeds a given prompt x to a d-dimensional feature vector $\phi(x)$. The expected reward of x, denoted as $\mu(x)$, is then equal to $g_\theta(\phi(x))$, where $g_\theta : \mathbb{R}^d \to \mathbb{R}^m$ is a function parameterized by an *unknown* parameter θ. Since g_θ is shared across all prompts, the observations obtained by evaluating any of the prompts will contribute to the estimation of θ, potentially speeding up the learning process.

For ease of exposition, in the following, we assume $m = 2$, i.e., there are two metrics for the prompt performance, denoted as μ_1 and μ_2, respectively. The algorithm design and analysis can be extended for a general m.

5 Best Feasible Prompt Identification

For best feasible prompt identification, the learner seeks a prompt that maximizes a primary objective while satisfying feasibility conditions on a secondary criteria, under a fixed budget constraint. Under the corresponding bandits formulation, bandits algorithm Constrained Successive Rejects (CSR) (Faizal & Nair, 2022) is shown to be able to obtain the optimal feasible arm with an exponentially decaying error probability for stochastic bandits.

5.1 General Framework

Under the general bandits formulation, we propose a round based arm elimination framework. The process consists of R rounds, each having n_r pulls. In total, it has $\sum_{r=1}^{R} n_r = B$. Each round begins with an active arm set A_{r-1}, and at the end of each round, only l_r arms will be kept. $(R, \{n_r\}_{r=1}^{R}, \{l_r\}_{r=1}^{R})$ are determined beforehand through a SCHEDULER, such as Successive Rejection (Audibert & Bubeck, 2010) or Sequential Halving (Karnin et al., 2013). We set $l_R = 1$, which corresponds to the estimated best feasible prompt. As the process proceeds, the major tasks in each round r are as follows.

(i) Budget allocation. each round r starts from the active set A_{r-1} and a planned budget n_r. The step determines which arms to be pulled for the given budget n_t. Denote $(x^{(1)}, \dots, x^{(n_r)})$ as the list of arms to pull. The budget allocation for the active arms can be flexible as long as a sufficient exploration on the active arms is performed (e.g., uniformly sampling each arm, or adopting G-optimal design to cover the feature space).

(ii) Arm pulling and evaluation. For each $t \in [n_r]$, sample $(q_t, a_t) \sim \mathcal{D}$, query the LLM to generate $y_t \sim \mathcal{M}(q_t; x^{(t)})$, evaluate y_t and obtain $f^{(t)} = (f_1(y_t, a_t), f_2(y_t, a_t))$. The new observations $(\phi(x^{(t)}, f^{(t)})$ will be included in the collected datasets X_r and Y_r.

(iii) Reward estimation. An estimator will then utilize (X_r, Y_r) to estimate the unknown parameter θ, based on which the estimated performance $\hat{\mu}(x)$ can be obtained for each $x \in A_{r-1}$. The estimation can also be performed in flexible ways. In the simplest approach, sample means can be used without accounting for prompt features. More efficient alternatives can be developed based on the considered parameterization function g_θ, e.g., (regularized) least squares.

(iv) Arm elimination. Due to the multi-objective setting, the arm elimination step should jointly consider the optimality and feasibility of the active arms. For that purpose, it first forms the empirical feasible arm set $\widehat{\mathcal{F}}_r$ by checking wether $\widehat{\mu}_{2,r}(x) > \tau$ and then selects an empirical best feasible arm $x_r^\star$ based on $\widehat{\mu}_{1,r}(x)$ from $\widehat{\mathcal{F}}_r$. A key component for this step is to estimate the constrained gap for all active arms as follows

$$\widehat{\Delta}_r(x) = \min\left\{\max\left\{\tau - \widehat{\mu}_{2,r}(x),\ \widehat{\mu}_{1,r}(x_r^\star) - \widehat{\mu}_{1,r}(x)\right\}, \widehat{\mu}_{2,r}(x_r^\star) - \tau\right\}, \tag{5}$$

where $x_r^\star$ is the empirical best feasible arm in round r. The gap design comes from Faizal & Nair (2022), which characterizes the empirical difficulty of classifying an arm as either infeasible or suboptimal.

Based on the estimated constrained gap, the learner then eliminates the arms that are determined to be infeasible or suboptimal, and keeps the l_r arms with the smallest $\widehat{\Delta}_r(i)$ to form A_r. It then proceeds to the next round. The algorithm is presented in Algorithm 1. We note that if all arms are treated independently, the round schedule follows Successive Rejects, and the evaluation budgets are allocated uniformly, then GENSEC recovers the Constrained Successive Rejection algorithm (CSR) (Faizal & Nair, 2022).

Algorithm 1 GENeralized Successive Elimination under Constraints (GENSEC)

```
Input: budget $B$, constraint threshold $\tau$, prompt set $\mathcal{X}$, feature map $\phi(\cdot)$, dataset $\mathcal{D}$
Initialization: $A_0 \leftarrow \mathcal{X}$; $(R, \{n_r\}_{r=1}^R, \{l_r\}_{r=1}^R) \leftarrow \textsc{Scheduler}(K, B)$; $X_0 \leftarrow \emptyset, Y_0 \leftarrow \emptyset$
for $r = 1 : R$ do
    $(x^{(1)}, \ldots, x^{(n_r)}) \leftarrow \textsc{Allocator}(n_r, A_{r-1})$
    At each step $t \in \{1, \ldots, n_r\}$, pull arm $x^{(t)}$ with feature $\phi(x^{(t)})$, collect evaluation $f^{(t)}$, and update observations as $X_r \leftarrow X_{r-1} \cup \{\phi(x^{(t)})\}, Y_r \leftarrow Y_{r-1} \cup \{f^{(t)}\}$
    Obtain estimator $\widehat{\mu}_r(x) \leftarrow \textsc{Estimator}(x; X_r, Y_r), \forall x \in A_{r-1}$
    Update empirical feasible arm set and empirical best feasible arm
        $\widehat{\mathcal{F}}_r \leftarrow \{x \in A_{r-1} : \widehat{\mu}_{2,r}(x) > \tau\}, \quad x_r^\star \leftarrow \arg\max_{x \in \widehat{\mathcal{F}}_r} \widehat{\mu}_{1,r}(x)$
    Estimate constrained gap
        $\widehat{\Delta}_r(x) \leftarrow \min\left\{\max\left\{\tau - \widehat{\mu}_{2,r}(x),\ \widehat{\mu}_{1,r}(x_r^\star) - \widehat{\mu}_{1,r}(x)\right\}, \widehat{\mu}_{2,r}(x_r^\star) - \tau\right\}, \quad \forall x \in A_{r-1}$
    Perform arm elimination: $A_r \leftarrow \textsc{Eliminator}\big(A_{r-1},\ l_r,\ \widehat{\Delta}_r(\cdot)\big)$
end for
Output: $A_R$
```

5.2 Theoretical Analysis with Linear Reward Functions

We instantiate the general framework to a special case where the reward function $g_\theta(\phi(x))$ is linear in both $\phi(x)$ and θ, i.e., $\mu(x) = \phi(x)^\top \theta$, where $\phi(x) \in \mathbb{R}^d$ and $\theta \in \mathbb{R}^{d \times m}$. This recovers the classical linear bandits setting. The following theoretical guarantee can be obtained.

Theorem 1 (Informal version of Theorem 2 **).** *Assume total budget $B \geq 45d \lceil \log_2 d \rceil$. With* SCHEDULER *being Sequential Halving,* ALLOCATOR *using the G-optimal design (Section A.3.1), and* ESTIMATOR *based on least squares, the probability that Algorithm 1 fails to return $x^\star$ satisfies $\Pr\left[x^\star \notin A_R\right] \leq 36 \lceil \log_2 K \rceil \cdot \exp\left\{-\frac{c_1 B}{H \lceil \log_2 K \rceil}\right\}$, where c_1 is a positive constant, $H = \max_{i \in [K]} \frac{1}{\Delta_i^2}$, and Δ_i is the constrained gap defined as $\Delta_i =$* $\min\left\{\max\left(\tau - \mu_2(i), \mu_1(x^*) - \mu_1(i)\right), \mu_2(x^*) - \tau\right\}$.

Compared to the existing upper bounds in the stochastic bandit problem (Faizal & Nair, 2022), our result makes a significant improvement in the dependency on K. Specifically, while the existing results exhibit a dependence of K^3 in the leading coefficient, our upper bound only depends on $\log_2 K$, greatly reducing the impact of K on the error probability. Furthermore, although we adopt a different definition for H, both results show similar influence of the budget B and the constrained gap on the error upper bound.

Proof sketch of Theorem 1. At a high level, the proof shows that the algorithm rarely eliminates the optimal feasible arm $x^\star$. It consists of the following major steps. **Step 1:** establish uniform concentration bounds for the empirical estimates, which is derived from a self-normalized concentration inequality for linear models as shown in Lemma 1. **Step 2:** use the uniform concentration bounds to show that any suboptimal or infeasible arm appears better than $x^\star$ only with exponentially small probability (see Lemma 2). During this process, **three types of arms** are carefully considered: feasible but suboptimal arms, deceiver arms (infeasible but better than the optimal feasible arm on the primary objective), and infeasible sub-optimal arms, which brings more complicated failure modes compared with previous single-objective analyses. **Step 3:** Lemma 3 argues that $x^\star$ can only be eliminated if many such arms are simultaneously misleading, which is very unlikely. Finally, **Step 4** combines the error probabilities across all rounds via a union bound. This yields an error bound that decays exponentially in the budget B, up to logarithmic factors.

6 Pareto Prompt Set Identification

In this section, we investigate the Pareto set identification problem and propose an algorithm named Generalized Pareto Set Identification (GenPSI), which can be found in Section B. Compared with Algorithm 1, GenPSI shares the same components such as *budget allocation*, *arm pulling and evaluation*, and *reward estimation*. The major difference lies in the last component, i.e., *arm elimination*. Due to the different objectives between best feasible arm identification and Pareto set identification, we use the empirical Pareto gap (Kone et al., 2024) as the metric for arm elimination.

We denote the empirical Pareto set as $\widehat{\mathcal{X}}^\star$, and employ the following notations

$$\widehat{m}(x,y) = \min_{i\in[m]} \widehat{\mu}_i(y) - \widehat{\mu}_i(x), \qquad \widehat{M}(x,y) = \max_{i\in[m]} \widehat{\mu}_i(x) - \widehat{\mu}_i(y),$$

$$\widehat{\delta}^+(x) = \min_{y\in\widehat{\mathcal{X}}^\star\setminus\{x\}} \min\left(M(x,y),\, M(y,x)\right),$$

$$\widehat{\delta}^-(x) = \min_{y\notin\widehat{\mathcal{X}}^\star} \left(\max(\widehat{M}(y,x),0) + \max_{y'\in\widehat{\mathcal{X}}^\star} \widehat{m}(y,y') \right).$$

Then, the empirical Pareto gap is defined as

$$\widehat{\Delta}(x) = \begin{cases} \max_{y\in\widehat{\mathcal{X}}^\star} \widehat{m}(x,y), & x \notin \widehat{\mathcal{X}}^\star, \\ \min\{\widehat{\delta}^+(x), \widehat{\delta}^-(x)\}, & x \in \widehat{\mathcal{X}}^\star. \end{cases}$$

The Pareto gap measures the difficulty of classifying an arm to be Pareto or sub-optimal.

We note that GenPSI recovers the Empirical Gap Elimination (EGE) (Kone et al., 2024) for stochastic bandits and G-optimal Empirical Gap Elimination (GEGE) (Kone et al., 2025) for linear bandits when the corresponding Allocator and Estimator are set in the same form.

7 Experiments

Datasets. We evaluate the prompt selection algorithms on two standard summarization benchmarks: **XSum** (Narayan et al., 2018) and **CNN/DailyMail** (Hermann et al., 2015). The XSum dataset contains approximately 227,000 documents paired with concise one-sentence summaries, whereas the CNN/DailyMail dataset comprises roughly 311,000 article-summary pairs with multi-sentence outputs.

Candidate prompt generation. We generate candidate prompts $\mathcal{X}$ using LLaMA-3. For each dataset, 200 prompts are generated based on randomly sampled examples from the dataset (Zhou et al., 2022). The generated prompts are then manually filtered to remove some completely irrelevant prompts and down-sampled to form the final prompt pool of size 100. Within the prompt pool, we further sample 50 and 30 prompts to form smaller candidate prompt sets.

Models. We evaluate the performance of candidate prompts on two LLMs: instruction-tuned LLaMA-3-8B-instruct and Gemma-7B-it. Given a prompt x and a query q, the LLM generates an output using greedy decoding with a maximum of 512 new tokens, which will then be evaluated according the metrics below.

Metrics. We evaluate LLM generated responses on two metrics: **ROUGE-L F1** (Lin, 2004), measuring the overlap with the reference summary, and **Brevity score**, measuring the token lengths. The details of the reward definition can be found in Section C.2.

Feature extraction. To facilitate the general bandits formulation, we use GPT-3.5-turbo to extract the embeddings of the prompts. Let $e(x) \in \mathbb{R}^p$ be the extracted embedding for prompt $x \in \mathcal{X}$. We then perform principle component analysis (PCA) to obtain a reduced feature representation. Let $U_d \in \mathbb{R}^{p\times d}$ denote the matrix of the top-d eigenvectors of the sample covariance of $\{e(x)\}_{x\in\mathcal{X}}$. The reduced feature of x is then $\phi(x) = U_d^\top \big(e(x) - \bar{e}\big) \in \mathbb{R}^d$, where $\bar{e} = \frac{1}{K}\sum_{x\in\mathcal{X}} e(x)$.

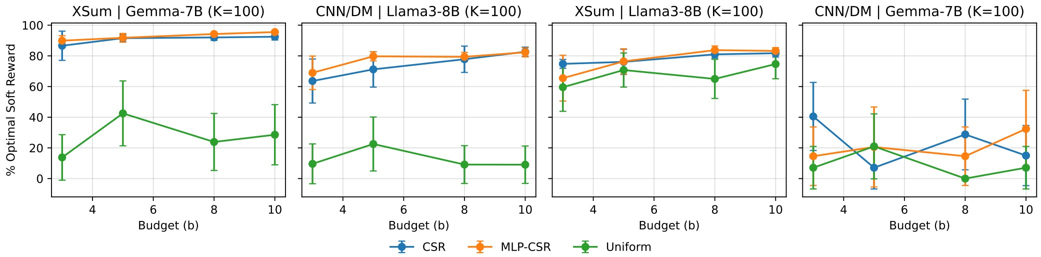


Figure 2: Average soft constrained reward vs. per-arm budget. Error bars denote 95% confidence intervals over 20 random seeds.

Table 1: Average soft constrained reward on XSum - Gemma.

K	Method	b = 3	b = 5	b = 8	b = 10
30	Uniform	0.015 ± 0.010	0.000 ± 0.000	0.030 ± 0.013	0.037 ± 0.014
	CSR	0.117 ± 0.013	0.137 ± 0.007	$\mathbf{0.144 \pm 0.000}$	$\mathbf{0.143 \pm 0.000}$
	MLP-CSR	$\mathbf{0.123 \pm 0.010}$	$\mathbf{0.139 \pm 0.002}$	0.140 ± 0.002	0.142 ± 0.001
50	Uniform	0.021 ± 0.011	0.021 ± 0.011	0.037 ± 0.014	0.036 ± 0.014
	CSR	0.122 ± 0.012	$\mathbf{0.143 \pm 0.001}$	0.141 ± 0.002	0.140 ± 0.002
	MLP-CSR	$\mathbf{0.143 \pm 0.002}$	0.142 ± 0.002	$\mathbf{0.147 \pm 0.001}$	$\mathbf{0.142 \pm 0.002}$
100	Uniform	0.021 ± 0.011	0.066 ± 0.016	0.037 ± 0.014	0.044 ± 0.015
	CSR	0.134 ± 0.007	0.141 ± 0.002	0.142 ± 0.001	0.143 ± 0.002
	MLP-CSR	$\mathbf{0.139 \pm 0.002}$	$\mathbf{0.142 \pm 0.002}$	$\mathbf{0.145 \pm 0.001}$	$\mathbf{0.147 \pm 0.001}$

7.1 Best Feasible Prompt Identification

In this subsection, we evaluate our constrained prompt selection algorithm, GENSEC, where the objective is to maximize task utility under a prescribed brevity constraint. We instantiate GENSEC into two variants, namely, CSR and MLP-CSR.

CSR. We first treat prompts independently, and adopt Successive Rejection, uniform pulling and sample averaging as the Scheduler, Allocator and Estimator, respectively, under which GENSEC reduces to CSR.

MLP-CSR. For the general bandits setting, we use an MLP consisting of a ReLU neural network with one hidden layer of 30 hidden states to model the reward function $g_\theta(\cdot)$. We set the Scheduler to be Sequential Halving to reduce the training rounds.

Denote the final output of the algorithms as $\hat{x}$. Then, we define *soft constrained reward* of the $\hat{x}$ as $\mu_1(\hat{x})$ if $\mu_2(\hat{x}) \geq 0.9\tau$; otherwise, it equals zero. We use this definition to tolerate slight violation of the constraint. In Figure 2, we report the average soft constrained reward as a function of the average budget per arm with $K = 100$ prompts. We normalize the value by $\mu_1(x^*)$, where x^* is the best feasible arm. We note that the proposed bandits based algorithms outperform the uniform baseline in almost all settings. The advantage is more significant when performing task CNN_Dailymail on Llama3 (denoted as 'CNN_Dailymail - Llama3') and performing task XSUM on Gemma (denoted as 'XSum - Gemma'). For both cases, when the average budget on each arm b is sufficiently large, the proposed bandits based algorithms recover more than 80% and 90% of the utility of x^* subject to the relaxed constraint, while the baseline only reaches 20% to 50% of that, respectively.

In Table 1, we present the soft constrained reward results by performing task XSum on Gemma. Across different prompt set sizes K, and per-arm budget b, both CSR and MLP-CSR consistently outperform the uniform pulling baseline. The uniform pulling baseline rarely finds a feasible near-optimal prompt, while our algorithms can reliably find a close-optimal prompt. Notably, MLP-CSR yields slightly higher average soft rewards than CSR, demonstrating the advantage of shared structure in reward functions.

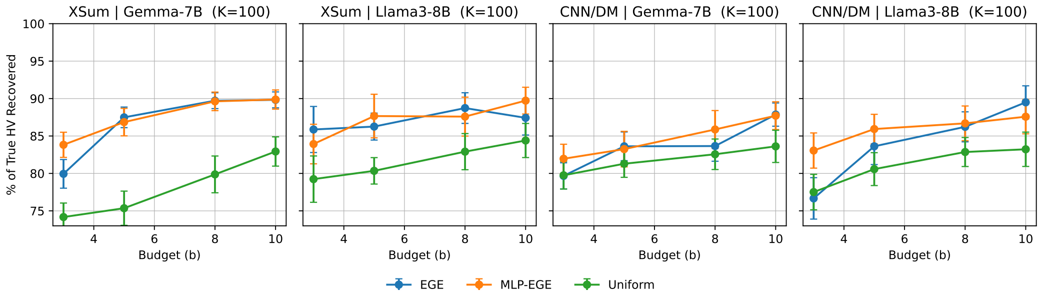


Figure 3: Hypervolume recovery vs. per-arm budget. Error bars denote 95% confidence intervals over 20 random seeds.

Table 2: Hypervolume (HV) on CNN/DailyMail - Llamma 3.

K	Method	b = 3	b = 5	b = 8	b = 10
30	Uniform	0.1577 ± 0.0039	0.1534 ± 0.0051	0.1685 ± 0.0028	0.1661 ± 0.0037
	EGE	0.1603 ± 0.0049	0.1680 ± 0.0028	0.1753 ± 0.0033	$\mathbf{0.1744 \pm 0.0043}$
	MLP-EGE	$\mathbf{0.1632 \pm 0.0034}$	$\mathbf{0.1688 \pm 0.0039}$	$\mathbf{0.1803 \pm 0.0022}$	0.1727 ± 0.0030
50	Uniform	0.1559 ± 0.0029	0.1543 ± 0.0031	0.1618 ± 0.0023	0.1646 ± 0.0031
	EGE	$\mathbf{0.1651 \pm 0.0023}$	0.1647 ± 0.0022	$\mathbf{0.1706 \pm 0.0032}$	$\mathbf{0.1720 \pm 0.0023}$
	MLP-EGE	0.1618 ± 0.0033	$\mathbf{0.1628 \pm 0.0027}$	0.1671 ± 0.0028	0.1673 ± 0.0026
100	Uniform	0.1519 ± 0.0024	0.1579 ± 0.0022	0.1624 ± 0.0019	0.1631 ± 0.0023
	EGE	0.1503 ± 0.0028	0.1639 ± 0.0024	0.1690 ± 0.0020	$\mathbf{0.1754 \pm 0.0022}$
	MLP-EGE	$\mathbf{0.1628 \pm 0.0024}$	$\mathbf{0.1684 \pm 0.0020}$	$\mathbf{0.1699 \pm 0.0023}$	0.1716 ± 0.0023

7.2 Pareto Prompt Set Identification

The proposed GenPSI framework for Pareto prompt set identification is also instantiated into two variants.

EGE. We first treat prompts independently, and adopt Successive Rejection, uniform pulling and sample averaging as the Scheduler, Allocator and Estimator, respectively, under which GenPSI reduces to EGE.

MLP-EGE. Following the same configuration of Scheduler and reward function as in MLP-CSR, we instantiate GenPSI to MLP-EGE.

To evaluate the performance of the algorithms on Pareto prompt set identification, we introduce a metric called hypervolume (HV) (Knowles et al., 2003). HV measures the Lebesgue volume of objective space dominated by the obtained Pareto set, which is a scaler indicator of the quality and diversity of trade-offs of the estimated Pareto set. HV is computed with respect to a reference point, which we set to the origin for both metrics. We also present the recovered HV as a percentage of the ground-truth Pareto-set HV, which is obtained by exhaustively evaluating all prompts in the candidate pool.

In Figure 3, we report the results when prompt set size $K = 100$. We note that both bandits based algorithms consistently outperform the baseline. When per-arm budget b is low, MLP-EGE achieves the highest performance on average, indicating the advantage of exploiting shared parameters under GenPSI. When per-arm budget $b = 8, 10$, both elimination approaches recover about 90% HV of the ground-truth Pareto set, whereas the baseline recovers around the low-to-mid 80% range.

Table 2 presents the average hypervolume on XSum dataset with Gemma for varying K and b. Bandits based algorithms consistently outperform the baseline, indicating robustness of the proposed GenPSI framework. More comprehensive evaluation results can be found in Section C.4.

8 Conclusion

In this work, we established a principled connection between multi-objective prompt selection and the framework of multi-objective pure-exploration bandits, representing (to the best of our knowledge) the first formalization of this problem in such a setting. Within this framework, we addressed two fundamental problems: best feasible prompt identification and Pareto prompt set identification. To this end, we proposed two general algorithms, GENSEC and GENPSI, which were theoretically grounded and designed to exploit shared structures among prompts to improve sample efficiency under limited evaluation budgets. Extensive experiments on XSum and CNN/DailyMail demonstrated the effectiveness of our approach. Our framework opened new opportunities for principled and scalable prompt optimization under complex evaluation criteria. Future directions include extending our methods to more diverse tasks and models, incorporating richer evaluation metrics such as fairness and efficiency, and exploring strategies for real-world, large-scale deployment.

References

Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, et al. Gepa: Reflective prompt evolution can outperform reinforcement learning. In *NeurIPS 2025 Workshop on Foundations of Reasoning in Language Models*, 2025.

Zeyuan Allen-Zhu, Yuanzhi Li, Aarti Singh, and Yining Wang. Near-optimal design of experiments via regret minimization. In *International Conference on Machine Learning*, pp. 126–135. PMLR, 2017.

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT-23th Conference on learning theory-2010*, pp. 13–p, 2010.

Peter Auer, Chao-Kai Chiang, Ronald Ortner, and Madalina Drugan. Pareto front identification from stochastic bandit feedback. In *Artificial intelligence and statistics*, pp. 939–947. PMLR, 2016.

Jill Baumann and Oliver Kramer. Evolutionary multi-objective optimization of large language model prompts for balancing sentiments. In *International conference on the applications of evolutionary computation (part of evoStar)*, pp. 212–224. Springer, 2024.

Jie Bian and Vincent YF Tan. Asymptotically optimal linear best feasible arm identification with fixed budget. In *Conference on Uncertainty in Artificial Intelligence*, pp. 296–331. PMLR, 2025.

Stephen Brade, Bryan Wang, Mauricio Sousa, Sageev Oore, and Tovi Grossman. Promptify: Text-to-image generation through interactive prompt exploration with large language models. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pp. 1–14, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Sebastian Caldas, Peter Wu, Tian Li, Jakub Konečný, H. Brendan McMahan, Virginia Smith, and Ameet Talwalkar. LEAF: A benchmark for federated settings. *CoRR*, abs/1812.01097, 2018. URL `http://arxiv.org/abs/1812.01097`.

Lichang Chen, Jiuhai Chen, Tom Goldstein, Heng Huang, and Tianyi Zhou. Instructzero: efficient instruction optimization for black-box large language models6518. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 6503–6518, 2024.

Jiale Cheng, Xiao Liu, Kehan Zheng, Pei Ke, Hongning Wang, Yuxiao Dong, Jie Tang, and Minlie Huang. Black-box prompt optimization: Aligning large language models without model training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3201–3219, 2024.

Sukmin Cho, Soyeong Jeong, Jeong yeon Seo, and Jong C Park. Discrete prompt optimization via constrained generation for zero-shot re-ranker. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 960–971, 2023.

Mingkai Deng, Jianyu Wang, Cheng-Ping Hsieh, Yihan Wang, Han Guo, Tianmin Shu, Meng Song, Eric Xing, and Zhiting Hu. Rlprompt: Optimizing discrete text prompts with reinforcement learning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 3369–3391, 2022.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409, 2021.

Fathima Zarin Faizal and Jayakrishnan Nair. Constrained pure exploration multi-armed bandits with a fixed budget. *arXiv preprint arXiv:2211.14768*, 2022.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. In *Neural Information Processing Systems*. Curran Associates, 2024.

Qingyan Guo, Rui Wang, Junliang Guo, Bei Li, Kaitao Song, Xu Tan, Guoqing Liu, Jiang Bian, and Yujiu Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. In *The Twelfth International Conference on Learning Representations*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiaoqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. Localized zeroth-order prompt optimization. *Advances in Neural Information Processing Systems*, 37:86309–86345, 2024.

Yasaman Jafari, Dheeraj Mekala, Rose Yu, and Taylor Berg-Kirkpatrick. Morl-prompt: An empirical analysis of multi-objective reinforcement learning for discrete prompt optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 9878–9889, 2024.

Anmol Kagrecha, Jayakrishnan Nair, and Krishna Jagannathan. Constrained regret minimization for multi-criterion multi-armed bandits. *Machine Learning*, 112(2):431–458, 2023.

Bongsu Kang, Jundong Kim, Tae-Rim Yun, and Chang-Eop Kim. Prompt-rag: Pioneering vector embedding-free retrieval-augmented generation in niche domains, exemplified by korean medicine. *arXiv preprint arXiv:2401.11246*, 2024.

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

Joshua D Knowles, David W Corne, and Mark Fleischer. Bounded archiving using the lebesgue measure. In *The 2003 Congress on Evolutionary Computation, 2003. CEC'03.*, volume 4, pp. 2490–2497. IEEE, 2003.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Cyrille Kone, Emilie Kaufmann, and Laura Richert. Bandit Pareto set identification: the fixed budget setting. In *International Conference on Artificial Intelligence and Statistics*, pp. 2548–2556. PMLR, 2024.

Cyrille Kone, Emilie Kaufmann, and Laura Richert. Bandit Pareto set identification in a multi-output linear model. In *AISTATS 2025-8th International Conference on Artificial Intelligence and Statistics*, 2025.

Mingze Kong, Zhiyong Wang, Yao Shu, and Zhongxiang Dai. Meta-prompt optimization for LLM-based sequential decision making. In *ICLR 2025 Workshop on Reasoning and Planning for Large Language Models*, 2025.

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Xiaoqiang Lin, Zhongxiang Dai, Arun Verma, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Prompt optimization with human feedback. In *ICML 2024 Workshop on Models of Human Feedback for AI Alignment*.

Xiaoqiang Lin, Zhaoxuan Wu, Zhongxiang Dai, Wenyang Hu, Yao Shu, See-Kiong Ng, Patrick Jaillet, and Bryan Kian Hsiang Low. Use your INSTINCT: Instruction optimization for LLMs using neural bandits coupled with transformers. In *International Conference on Machine Learning*, pp. 30317–30345. PMLR, 2024.

Aman Madaan, Amrith Setlur, Tanmay Parekh, Barnabas Poczos, Graham Neubig, Yiming Yang, Ruslan Salakhutdinov, Alan W Black, and Shrimai Prabhumoye. Politeness transfer: A tag and generate approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1869–1881, 2020.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *Advances in Neural Information Processing Systems*, 37:61065–61105, 2024.

Wenyi Mo, Tianyu Zhang, Yalong Bai, Bing Su, Ji-Rong Wen, and Qing Yang. Dynamic prompt optimizing for text-to-image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26627–26636, 2024.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *ArXiv*, abs/1808.08745, 2018.

Aldo Pacchiano, Mohammad Ghavamzadeh, Peter Bartlett, and Heinrich Jiang. Stochastic bandits with linear constraints. In *International conference on artificial intelligence and statistics*, pp. 2827–2835. PMLR, 2021.

Reid Pryzant, Dan Iter, Jerry Li, Yin Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 7957–7968, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Yarik Menchaca Resendiz and Roman Klinger. Mopo: Multi-objective prompt optimization for affective text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 5588–5606, 2025.

Antonio Sabbatella, Andrea Ponti, Ilaria Giordani, Antonio Candelieri, and Francesco Archetti. Prompt optimization in large language models. *Mathematics*, 12(6):929, 2024.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927*, 2024.

Sander Schulhoff, Michael Ilie, Nishant Balepur, Konstantine Kahadze, Amanda Liu, Chenglei Si, Yinheng Li, Aayush Gupta, HyoJung Han, Sevien Schulhoff, et al. The prompt report: a systematic survey of prompt engineering techniques. *arXiv preprint arXiv:2406.06608*, 2024.

Chengshuai Shi, Kun Yang, Zihan Chen, Jundong Li, Jing Yang, and Cong Shen. Efficient prompt optimization through the lens of best arm identification. *Advances in Neural Information Processing Systems*, 37:99646–99685, 2024.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pp. 4222–4235, 2020.

Xinyu Tang, Xiaolei Wang, Wayne Xin Zhao, Siyuan Lu, Yaliang Li, and Ji-Rong Wen. Unleashing the potential of large language models as prompt optimizers: Analogical analysis with gradient-based model optimizers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 25264–25272, 2025.

Chao Tao, Saúl Blanco, and Yuan Zhou. Best arm identification in linear bandits with linear dimension dependency. In *International Conference on Machine Learning*, pp. 4877–4886. PMLR, 2018.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Wenrui Xu and Keshab K Parhi. A survey of attacks on large language models. *arXiv preprint arXiv:2505.12567*, 2025.

Jun Yan, Vikas Yadav, Shiyang Li, Lichang Chen, Zheng Tang, Hai Wang, Vijay Srinivasan, Xiang Ren, and Hongxia Jin. Backdooring instruction-tuned large language models with virtual prompt injection. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 6065–6086, 2024.

Heng Yang and Ke Li. Instoptima: Evolutionary multi-objective instruction optimization via large language model-based instruction operators. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 13593–13602, 2023.

Tom Zehle, Moritz Schlager, Timo Heiß, and Matthias Feurer. Capo: Cost-aware prompt optimization. In *International Conference on Automated Machine Learning*, pp. 18–1. PMLR, 2025.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

Yongchao Zhou, Andrei Ioan Muresanu, Ziwen Han, Keiran Paster, Silviu Pitis, Harris Chan, and Jimmy Ba. Large language models are human-level prompt engineers. In *The eleventh international conference on learning representations*, 2022.

Marcela Zuluaga, Guillaume Sergent, Andreas Krause, and Markus Püschel. Active learning for multi-objective optimization. In *International conference on machine learning*, pp. 462–470. PMLR, 2013.

Marcela Zuluaga, Andreas Krause, and Markus Püschel. e-PAL: An active learning approach to the multi-objective optimization problem. *Journal of Machine Learning Research*, 17(104):1–32, 2016.

APPENDIX

A BEST FEASIBLE PROMPT IDENTIFICATION

A.1 SETTINGS AND NOTATIONS

In this section, we formally present the Linear Constrained Sequential Halving in Algorithm 2. We first recap the linear constrained bandit setting and notations. For the multi-objective bandit setting, we have the prompt or arm set $\mathcal{X}$ with $|\mathcal{X}| = K$, and the expected performance or reward vector $\mu(x), x \in \mathcal{X}$. For each arm, there is an associated feature $\phi(x) \in \mathbb{R}^d$. In the linear constrained bandit setting, we assume the number of reward dimensions is only two for simplicity and the expected rewards satisfy the linear structure,

$$\mu_1(x) = \phi(x)^\top \theta^{(1)}, \qquad \mu_2(x) = \phi(x)^\top \theta^{(2)}, \tag{6}$$

where $\theta^{(1)}, \theta^{(2)} \in \mathbb{R}^d$ are unknown. Moreover, for the theoretical results, we further assume that the one-time evaluation results satisfy,

$$f^{(t)} = (f_1^{(t)}, f_2^{(t)}) = (\mu_1(x^{(t)}), \mu_2(x^{(t)})) + (\eta_1^{(t)}, \eta_2^{(t)}), \tag{7}$$

where $f^{(t)}$ is the evaluation reward of one pull of the arm $x^{(t)}$ and $\eta_1^{(t)}, \eta_2^{(t)}$ are the independent σ-sub-Gaussian noise.

Definition 2 (σ-sub-Gaussian). *A random variable X is* 1-sub-Gaussian *if for all $\lambda \in \mathbb{R}$,*

$$\mathbb{E}\Big[e^{\lambda\,(X - \mathbb{E}[X])}\Big] \le \exp\!\left(\frac{\sigma^2\lambda^2}{2}\right).$$

A.2 CONSTRAINED GAP

Given a constrained threshold τ, the feasible set is denoted as $\mathcal{F} := \{\, x \in \mathcal{X} : \mu_2(x) \ge \tau\}$. The goal of the constrained bandit algorithm is to identify the optimal arm $x^\star$,

$$x^\star := \begin{cases} \arg\max_{x\in\mathcal{F}} \mu_1(x), & \mathcal{F} \ne \varnothing, \\ \arg\max_{x\in\mathcal{X}} \mu_2(x), & \mathcal{F} = \varnothing\ . \end{cases}$$

With the notations, we follow Faizal & Nair (2022) to define the violation gap and sub-optimal gap as follows

$$\mathrm{viol}(x) := \max\!\big(\tau - \mu_2(x),\, 0\big), \qquad \mathrm{subopt}(x) := \mu_1(x^\star) - \mu_1(x),$$

where the latter is used only when $\mathcal{F} \ne \varnothing$.

They characterize the difficulty to distinguish a prompt as infeasible or suboptimal. Then, the gap used to classify an arm as infeasible or suboptimal is denoted as

$$\delta(x) = \begin{cases} \max\!\big(\mathrm{viol}(x),\, \mathrm{subopt}(x)\big), & \mathcal{F} \ne \varnothing, \\ \mu_2(x^\star) - \mu_2(x), & \mathcal{F} = \varnothing\ . \end{cases}$$

Finally, note that the probability of failing to identify the optimal arm also includes the case where the optimal arm is estimated as infeasible. Therefore, we define the constrained gap as

$$\Delta(x) = \min\{\delta(x), \mu_2(1) - \tau\}. \tag{8}$$

We also assume that the arms are indexed in ascending order of their constrained gaps, with the first arm corresponding to the optimal feasible arm, such that $\Delta(1) \le \Delta(2) \le \cdots \le \Delta(K)$.

A.3 ALGORITHM DESIGN

We present the Linear Constrained Sequential Halving Algorithm for the linear reward setting in Algorithm 2. The algorithm is instantiated from GENSEC Algorithm 1 by using the Sequential Halving (SH) scheduler, the G-optimal design allocator and the linear estimator.

Algorithm 2 LINEAR CONSTRAINED SEQUENTIAL HALVING

```
Input: budget $B$, threshold $\tau$, prompt set $\mathcal{X}$, feature map $\phi(\cdot)$, tolerance $\epsilon$, parameter $\kappa \in (0, 1/3]$
Initialization: $A_0 \leftarrow \mathcal{X}$; $X_0 \leftarrow \emptyset, Y_0 \leftarrow \emptyset$
$(R, \{n_r\}_{r=1}^R, \{l_r\}_{r=1}^R) \leftarrow$ SCHEDULER$(K, B)$:
        Sequential Halving : $R = \lceil \log_2 K \rceil, \quad l_r = \left\lceil \frac{K}{2^r} \right\rceil, \quad n_r = \left\lfloor \frac{B}{R} \right\rfloor$
for $r = 1$ to $R$ do
    Allocate $n_r$ pulls across active arm set $A_{r-1}$ based on G-optimal design (Algorithm 3):
        $(x^{(1)}, \dots, x^{(n_r)}) \leftarrow$ G-OPTIMAL DESIGN$(n_r, A_{r-1}, \phi, \epsilon, \kappa)$
    Pull the arms and collect the observation data:
        $X_r \leftarrow \{\phi(x^{(t)})\}_t, \quad Y_r \in \mathbb{R}^{n_r} \leftarrow \{f^{(t)}\}_t$
    $\widehat{\mu}_r(x) \leftarrow$ ESTIMATOR$(x; \phi(\cdot), X_r, Y_r), \forall x \in A_{r-1}$:
  $\Phi_r \leftarrow [\,\phi(x_{r,1}), \dots, \phi(x_{r,k_r})\,] \in \mathbb{R}^{d \times k_r}$, a maximal linearly independent subset of $\{\phi(x)\}_{x \in A_r}$.
  $V_r := X_r^\top X_r, \quad V_r^\dagger := \Phi||r(\Phi_r^\top V_r \Phi_r)^{-1} \Phi_r^\top$
  $\widehat{\theta}_r = V_r^\dagger X_r^\top Y_r, \quad \widehat{\mu}_r(x) \;:=\; \big(\widehat{\mu}_{1,r}(x), \, \widehat{\mu}_{2,r}(x)\big) \;=\; \phi(x)^\top \widehat{\theta}_r$
    Construct empirical feasible set and identify empirical optimal arm:
        $\widehat{\mathcal{F}}_r \leftarrow \{x \in A_{r-1} : \widehat{\mu}_{2,r}(x) > \tau\}, \quad i_r^\star \in \arg\max_{x \in \widehat{\mathcal{F}}_r} \widehat{\mu}_{1,r}(x)$
    Obtain empirical constrained gap:
        $\hat{\delta}_r(i) \leftarrow \max\Big\{ \tau - \widehat{\mu}_{2,r}(i), \; \widehat{\mu}_{1,r}(i_r^\star) - \widehat{\mu}_{1,r}(i) \Big\},$
        $\widehat{\Delta}_r(i) = \min\big\{\hat{\delta}_r(i), \; \hat{\mu}_{2,r}(i_r^\star) - \tau\big\}, \qquad \forall i \in A_{r-1}$
    Rank arms in $A_{r-1}$ in the increasing order of $\widehat{\Delta}_r(\cdot)$ and keep the first $l_r$ arms to form $A_r$
end for
Output: $A_R$
```

A.3.1 G-OPTIMAL DESIGN

The G-optimal design follows Kone et al. (2025), achieving the same estimation error bound as in Lemma 1 for multi-objective bandits. It employs entropic mirror descent to solve the continuous relaxation of the G-optimal design problem, followed by an efficient rounding step to produce an approximate integer solution. These procedures are adapted from Tao et al. (2018) and Allen-Zhu et al. (2017), and are denoted by MD and ROUND, respectively.

The output N_i of the rounding procedure indicates the number of pulls allocated for arm i. The corresponding budget allocation $\{x^{(t)}\}_{t=1}^{n_r}$ can then be constructed accordingly. We present the subroutine below.

Algorithm 3 G-OPTIMAL DESIGN (KONE ET AL., 2025)

```
Input: budget $N$, active set $A$, feature map $\phi(\cdot)$, tolerance $\epsilon$, parameter $\kappa \in (0, 1/3]$
$\{w_i\}_{i \in A} \leftarrow$ MD$(A, \phi, \epsilon)$
$\{N_i\}_{i \in A} \leftarrow$ ROUND$(N, \{w_i\}_{i \in A}, \kappa, A, \phi)$
for $t = 1$ to $N$ do
    $x^{(t)} \leftarrow i$, if $t \in \left(\sum_{j=1}^{i-1} N_j, \sum_{j=1}^{i} N_j\right]$
end for
Output: $\{x^{(t)}\}_t$
```

A.3.2 Linear Estimator

For the linear case, the estimator only relies on the data collected in the current round to estimate θ. In round r, given the active arm set A_{r-1}, the arm pulled at step t satisfies $x^{(t)} \in A_{r-1}$ and yields the observed outcome $f^{(t)}$. We define X_r as the design matrix obtained by stacking the feature vectors $\phi(x^{(t)})^\top$ from round r, and Y_r as the vector of the corresponding outcomes $f^{(t)}$.

$$X_r := \begin{bmatrix} \phi(x^{(1)})^\top \\ \phi(x^{(2)})^\top \\ \vdots \\ \phi(x^{(n_r)})^\top \end{bmatrix} \in \mathbb{R}^{n_r \times d}, \qquad Y_r := \begin{bmatrix} f^{(1)} \\ f^{(2)} \\ \vdots \\ f^{(n_r)} \end{bmatrix} \in \mathbb{R}^{n_r}.$$

In each round r, define the Gram matrix

$$V_r := X_r^\top X_r \in \mathbb{R}^{d \times d}.$$

Denoting $\Phi_r = [\,\phi(x_{r,1}), \dots, \phi(x_{r,k_r})\,] \in \mathbb{R}^{d \times k_r}$ as a maximal linearly independent subset selected from $\{\phi(x)\}_{x \in A_r}$, we can define the pesudo inverse

$$V_r^\dagger \;:=\; \Phi_r \big(\Phi_r^\top V_r \Phi_r\big)^{-1} \Phi_r^\top \quad \in \mathbb{R}^{d \times d}. \tag{9}$$

Then, the estimation becomes

$$\widehat{\theta}_r \;=\; V_r^\dagger X_r^\top Y_r, \qquad \widehat{\mu}_r(x) \;=\; \phi(x)^\top \widehat{\theta}_r. \tag{10}$$

A.4 Proof of Theorem 2

Before we proceed to prove Theorem 2, we first introduce several key lemmas.

We first bound the estimation error in each round by following Lemma 2 of Kone et al. (2025).

Lemma 1 (Adapted from Lemma 2 in Kone et al. (2025))**.** *Let $\kappa \in (0, 1/3]$ and $N \geq 5d_{r-1}/\kappa^2$, where d_r = dim(span$\{\phi(x) : x \in A_r\}$), $\theta \in \mathbb{R}^{d \times m}$. Then, under Algorithm 2, for all $\epsilon > 0$ and $x \in A_{r-1}$,*

$$\mathbb{P}\left(\left\|(\theta - \hat{\theta})^\top \phi(x_i)\right\|_\infty \geq \epsilon\right) \leq 4 \exp\left(-\frac{a n_r \epsilon^2}{d_r}\right),$$

where $a = \frac{1}{2(1+6\kappa)\sigma^2}$.

Lemma 1 indicates that, by applying the G-optimal allocator and the linear estimator, the estimated rewards concentrate around the true performance mean.

Next, we leverage Lemma 1 to show Lemma 2, which bounds the probability that a sub-optimal prompt is empirically better than the optimal arm.

Lemma 2. *Assume that the best feasible arm $x^\star$ is not eliminated prior to round r. Let n_r be the total allocated budget in round r. Then, for any arm $i \in A_r$, define $G_r(i)$ as the event that arm i is empirically better than the best feasible arm (i.e., arm 1) at the end of round r, i.e.,*

$$G_r(i) = \{\widehat{\Delta}_r(i) < \widehat{\Delta}_r(1) \quad or \quad \hat{\mu}_2(1) < \tau\}.$$

Then,

$$\mathbb{P}(G_r(i)) \leq 12 \exp\left\{-\frac{a n_r}{4 d_r} \Delta^2(i)\right\}.$$

Proof. Let $F_r(1)$ denote the event that arm 1 is empirically feasible at the end of round r. Thus, we have

$$\begin{aligned}
\mathbb{P}(G_r(i)) &= \mathbb{P}(G_r(i) \cap F_r(1)) + \mathbb{P}(G_r(i) \cap F_r^c(1)) \\
&\leq \mathbb{P}(G_r(i) \cap F_r(1)) + \mathbb{P}(F_r^c(1)) \\
&\leq \mathbb{P}(G_r(i) \cap F_r(1)) + \mathbb{P}(\widehat{\mu}_{2,r}(1) < \tau) \\
&\overset{(i)}{\leq} \mathbb{P}(G_r(i) \cap F_r(1)) + 4 \exp\left\{-a n_k (\tau - \widehat{\mu}_{2,r}(1))^2 / d_r\right\}
\end{aligned}$$

$$
\begin{aligned}
&\overset{(ii)}{\leq} \mathbb{P}(G_r(i) \cap F_r(1)) + 4\exp\left\{-an_k\Delta^2(i)/d_r\right\} \\
&\leq \mathbb{P}(G_r(i) \cap F_r(1)) + 4\exp\left\{-\frac{an_r}{4d_r}\Delta^2(i)\right\}, \qquad (11)
\end{aligned}
$$

where inequality (i) uses Lemma 1 to bound $\mathbb{P}(\hat{\mu}_{2,r}(1) < \tau)$, and inequality (ii) is based on the definition of $\Delta(i)$.

The event $G_r(i) \cap F_r(1)$ corresponds to the case that arm 1 is empirically feasible at the end of round r but worse than arm i. This implies that arm i is empirically feasible and outperforms arm 1 on the primary objective, i.e.,

$$
\mathbb{P}(G_r(i) \cap F_r(1)) \leq \mathbb{P}\left(\hat{\mu}_{2,r}(i) \geq \tau, \hat{\mu}_{1,r}(1) < \hat{\mu}_{1,r}(i)\right).
$$

To bound $\mathbb{P}(G_r(i) \cap F_r(1))$, we further consider three possible cases.

Case 1: Arm i is a feasible suboptimal arm.

In this case, we have

$$
\begin{aligned}
&\mathbb{P}(G_r(i) \cap F_r(1)) \\
&\quad \leq \mathbb{P}\left(\hat{\mu}_{1,r}(i) \geq \hat{\mu}_{1,r}(1)\right) \\
&\quad = \mathbb{P}\left\{(\hat{\mu}_{1,r}(i) - \mu_1(i)) - (\hat{\mu}_{1,r}(1) - \mu_1(1)) \geq (\mu_1(1) - \mu_1(i))\right\} \\
&\quad \leq \mathbb{P}\left\{\hat{\mu}_{1,r}(i) - \mu_1(i) \geq \frac{\mu_1(1) - \mu_1(i)}{2}\right\} + \mathbb{P}\left\{\mu_1(1) - \hat{\mu}_{1,r}(1) \geq \frac{\mu_1(1) - \mu_1(i)}{2}\right\},
\end{aligned}
$$

where the last step follows from the fact that both $\hat{\mu}_{1,r}(i) - \mu_1(i)$ and $\mu_1(1) - \hat{\mu}_{1,r}(1)$ cannot be greater than $\frac{\mu_1(1)-\mu_1(i)}{2}$ and a subsequent union bounding argument.

Using Lemma 1, we have

$$
\begin{aligned}
&\mathbb{P}\left\{\hat{\mu}_{1,r}(i) - \mu_1(i) \geq \frac{\mu_1(1) - \mu_1(i)}{2}\right\} \\
&\qquad \leq 4 \cdot \exp\left(-an_r \cdot \frac{(\mu_1(1) - \mu_1(i))^2}{4d_r}\right) \\
&\qquad \leq 4 \cdot \exp\left(-\frac{an_r}{4d_r}\Delta^2(i)\right)
\end{aligned}
$$

and

$$
\begin{aligned}
&\mathbb{P}\left\{\mu_1(1) - \hat{\mu}_{1,r}(1) \geq \frac{\mu_1(1) - \mu_1(i)}{2}\right\} \\
&\qquad \leq 4 \cdot \exp\left(-an_r \cdot \frac{(\mu_1(1) - \mu_1(i))^2}{4d_r}\right) \\
&\qquad \leq 4 \cdot \exp\left(-\frac{an_r}{4d_r}\Delta^2(i)\right).
\end{aligned}
$$

Thus, for this case,

$$
\mathbb{P}(G_r(i) \cap F_r(1)) \leq 8 \cdot \exp\left(-\frac{an_r}{4d_r}\Delta^2(i)\right).
$$

Case 2: Arm i is a deceiver arm.

This case corresponds to the situation where an infeasible arm i appears to be empirically feasible. In this case, we have

$$\begin{aligned}
&\mathbb{P}(F \cap B_k) \\
&\quad \le \mathbb{P}\left(\hat{\mu}_{2,r}(i) \ge \tau\right) \\
&\quad = \mathbb{P}\left(\hat{\mu}_{2,r}(i) - \mu_2(i) \ge \tau - \mu_2(i)\right) \\
&\quad \overset{(i)}{\le} 4 \exp\left(-a n_r \cdot (\tau - \mu_2(i))^2 / d_r\right) \qquad (12) \\
&\quad \overset{(ii)}{\le} 4 \exp\left(-a n_r \cdot \Delta^2(i) / d_r\right) \\
&\quad \le 4 \cdot \exp\left(-\frac{a n_r}{4 d_r} \cdot \Delta^2(i)\right),
\end{aligned}$$

where inequality (i) follows from Lemma 1 and inequality (ii) follows from the definition of $\Delta(i)$.

Case 3: Arm i is an infeasible suboptimal arm.

Since $\Delta(i) \le \max\{\tau - \mu_2(i),\ \mu_1(1) - \mu_1(i)\}$, then

if $\Delta(i) \le \mu_1(1) - \mu_1(i)$, then this case follows from Case 1;

if $\Delta(i) \le \tau - \mu_2(i)$, then this case follows from Case 2.

Put pieces together: Since for all three case, we have $\mathbb{P}(G_r(i) \cap F_r(1)) \le 8 \cdot \exp\left(-a n_r \cdot \frac{\Delta^2(i)}{4 d_r}\right)$. Plugging the result into Equation (11), we have

$$\begin{aligned}
\mathbb{P}(G_r(i) \cap F_r(1)) &\le \mathbb{P}(G_r(i) \cap F_r(1)) + 4 \exp\left\{-\frac{a n_r}{4 d_r} \Delta^2(i)\right\} \\
&\le 12 \exp\left\{-\frac{a n_r}{4 d_r} \Delta^2(i)\right\}.
\end{aligned}$$

□

With Lemma 2, we can further bound the probability of eliminating the best feasible arm in a round.

Lemma 3. *The probability that the best feasible arm is eliminated on round r satisfies*

$$\mathbb{P}(x^\star \notin A_r | x^\star \in A_{r-1}) \le 36 \cdot \exp\left\{-\frac{aB \cdot \Delta^2(i_r)}{4 \lceil \log_2 K \rceil}\right\},$$

where $i_r = \arg\min_{i \in A_r} \Delta(i)$.

Proof. In each round, we eliminate arms in A_{r-1} to obtain A_r, where $|A_r| = \frac{1}{2}|A_{r-1}| = \lceil \frac{K}{2^r} \rceil$. For the proof, we further consider A'_r as the $\frac{3}{4}$ worst arms in A_{r-1} and use N_r to denote the number of arms in A'_r whose empirical gap is smaller than $\widehat{\Delta}_r(1)$.

By Lemma 2, we have

$$\begin{aligned}
\mathbb{E}[N_r] &= \sum_{i \in A'_r} \mathbb{P}(G_r(i)) \\
&\le \sum_{i \in A'_r} 12 \cdot \exp\left(-a n_r \cdot \frac{\Delta^2(i)}{4 d_r}\right) \\
&\le \sum_{i \in A'_r} 12 \cdot \exp\left(-\frac{a}{4} \cdot \Delta^2(i) \cdot \frac{B}{\lceil \log_2 K \rceil}\right) \\
&\le |A'_r| \cdot \max_{i \in A_r} 12 \cdot \exp\left(-\frac{a}{4} \cdot \Delta^2(i) \cdot \frac{B}{\lceil \log_2 K \rceil}\right) \\
&= |A'_r| \cdot 12 \cdot \exp\left(-\frac{aB \min_{i \in A_r} \Delta^2(i)}{4 \lceil \log_2 K \rceil}\right)
\end{aligned}$$

Furthermore, applying Markov's inequality, we get

$$\mathbb{P}\left(N_r > \frac{1}{3}|A'_r|\right) \leq \frac{3 \cdot \mathbb{E}[N_r]}{|A'_r|} \leq 36 \cdot \exp\left(-\frac{aB \cdot \min_{i \in A'_r} \Delta^2(i)}{4\left\lceil \log_2 K \right\rceil}\right).$$

If $x^\star$ is rejected at round r, there have to be at most $\frac{1}{2}|A_{r-1}|$ arms in A'_r is empirically worse than the optimal arm. Thus, we have

$$\begin{aligned}
&\mathbb{P}(x^\star \notin A_r | x^\star \in A_{r-1}) \\
&\quad \leq \mathbb{P}\left(|A'_r| - N_r < \frac{2}{3}|A'_r|\right) \\
&\quad \leq 18d \cdot \exp\left(-\frac{aB \cdot \min_{i \in A'_r} \Delta^2(i)}{4\left\lceil \log_2 K \right\rceil}\right),
\end{aligned}$$

which completes the proof. □

With Lemmas 1 to 3, we are now ready to prove Theorem 1. We first state a formal version of Theorem 1 as follows.

Theorem 2. *Given a fixed set of K arms $\mathcal{X}$ with the linear reward structure specified in Equations* (6) *and* (7)*, under total budget $B \geq 45d\left\lceil \log_2 d \right\rceil$, the probability that Algorithm 2 fails to output the best feasible arm $x^\star$ is upper bounded by,*

$$\mathbb{P}(x^\star \notin A_R) \leq 36 \cdot \left\lceil \log_2 K \right\rceil \cdot \exp\left\{-\frac{aB}{4\left\lceil \log_2 K \right\rceil H}\right\},$$

where

$$a = \frac{1}{2(1+6\kappa)\sigma^2}, \quad H = \max_{i \in [K]} \frac{1}{\Delta^2(i)},$$

and $\Delta(i)$ is defined in Equation (8)*.*

Proof. By algorithm design, A_R contains only one arm. Thus, the error rate is equal to the probability that the best arm survives the whole algorithm execution. By Lemma 3 and a union bound, the probability that the best arm is eliminated in one of the $\left\lceil \log_2 K \right\rceil$ rounds of the algorithm can be upper bounded as below.

$$\begin{aligned}
&\mathbb{P}(x \neq x^\star, \forall x \in A_R) \\
&\quad \leq \sum_{r=1}^{\left\lceil \log_2 K \right\rceil} 36 \cdot \exp\left\{-\frac{aB \cdot \min_{i \in A_r} \Delta^2(i)}{4\left\lceil \log_2 K \right\rceil}\right\} \\
&\quad \leq 36 \cdot \left\lceil \log_2 K \right\rceil \cdot \max_{i \in [K]} \exp\left\{-\frac{aB \cdot \Delta^2(i)}{4\left\lceil \log_2 K \right\rceil}\right\} \\
&\quad = 36 \cdot \left\lceil \log_2 K \right\rceil \cdot \exp\left\{-\frac{aB}{4\left\lceil \log_2 K \right\rceil} \cdot \frac{1}{\max_{i \in [K]} \frac{1}{\Delta^2(i)}}\right\} && (13) \\
&\quad = 36 \cdot \left\lceil \log_2 K \right\rceil \cdot \exp\left\{-\frac{aB}{4\left\lceil \log_2 K \right\rceil H}\right\} && (14)
\end{aligned}$$

□

B Pareto set identification

B.1 Restate the GenPSI

For the GenPSI framework, the main difference from GenSEC is the design of gaps. In GenPSI, we use the Pareto gap, following Auer et al. (2016) and Kone et al. (2024). To define the gap, we first introduce the following defintions.

Algorithm 4 GENeralized Pareto Set Identification (GENPSI)

```
Input: budget $B$, prompt set $\mathcal{X}$, feature map $\phi(\cdot)$, dataset $\mathcal{D}$
Initialization: $A_0 \leftarrow \mathcal{X}$; $(R, \{n_r\}_{r=1}^R, \{l_r\}_{r=1}^R) \leftarrow$ SCHEDULER$(K, B)$; $X_0 \leftarrow \emptyset, Y_0 \leftarrow \emptyset$.
for $r = 1 : R$ do
    $(x^{(1)}, \dots, x^{(n_r)}) \leftarrow$ ALLOCATOR$(n_r, A_{r-1})$.
    At each step $t \in \{1, ..., n_r\}$, pull arm $x^{(t)}$ with feature $\phi(x^{(t)}$, observe evaluation $f^{(t)}$, and update observations as $X_r \leftarrow X_{r-1} \cup \{\phi(x^{(t)})\}, Y_r \leftarrow Y_{r-1} \cup \{f^{(t)}\}$
    Obtain estimator $\widehat{\mu}_r(x) \leftarrow$ ESTIMATOR$(x; X_r, Y_r)$, $\forall x \in A_{r-1}$;
    Calculate the empirical dominating set
        $\hat{\mathcal{X}}_r^\star := \Big\{x \in \hat{\mathcal{X}} \Big| \nexists\, x' \in \mathcal{X}$ such that $\big[\forall i \in [m],\, \hat{\mu}_{i,r}(x') \geq \hat{\mu}_{i,r}(x)\big]$ and $\big[\exists i \in [m], \hat{\mu}_{i,r}(x') > \hat{\mu}_{i,r}(x)\big]\Big\}$
    Calculate the empirical Pareto gap
        $\widehat{\Delta}_r(x) = \begin{cases} \max_{y \in \widehat{\mathcal{X}}^\star} \widehat{m}(x, y), & x \notin \widehat{\mathcal{X}}_r^\star, \\ \min\big\{\widehat{\delta}^+(x), \widehat{\delta}^-(x)\big\}, & x \in \widehat{\mathcal{X}}_r^\star \end{cases}$
    where the $\widehat{m}, \widehat{\delta}^+, \widehat{\delta}^-$ are all the empirical version based on $\widehat{\mu}$
    Perform arm elimination: $A_r \leftarrow$ ELIMINATOR$\big(A_{r-1},\, l_r,\, \widehat{\Delta}_r(\cdot)\big)$
end for
Output: $A_R$
```

Definition 3 (Pareto Dominance). *We say that arm x Pareto-dominates arm y (denoted as $\mu(x) \succ \mu(y)$) if and only if*

$$\forall i,\ \mu_i(x) \geq \mu_i(y) \text{ and } \exists i,\ \mu_i(x) > \mu_i(y).$$

Definition 4 (Pareto set). *For a given set of arms $\mathcal{X}$, the Pareto set is defined as*

$$\mathcal{X}^\star \ := \ \Big\{ x \in \mathcal{X} \ \Big| \ \nexists\, x' \in \mathcal{X} \text{ such that } \mu(x') \succ \mu(x) \Big\},$$

i.e., the Pareto set consists of all arms that are not dominated by any other arm.

Then, we define $m(x, y)$ and $M(x, y)$ for any prompt pair $x, y \in \mathcal{X}$ as follows:

$$m(x, y) := \min_{i \in [m]} \big(\mu_i(y) - \mu_i(x)\big), \qquad M(x, y) := \max_{i \in [m]} \big(\mu_i(x) - \mu_i(y)\big).$$

These two terms quantify the level of y dominating x.

If $x \notin \mathcal{X}^\star$, the Pareto gap is directly

$$\Delta(x) = \max_{y \in \mathcal{X}^\star} m(x, y),$$

which quantifies the greatest dominance from a Pareto prompt.

If $x \notin \mathcal{X}^\star$, we further denote

$$\delta^+(x) \ := \ \min_{y \in \mathcal{X}^\star \setminus \{x\}} \ \min\!\big(M(x, y),\, M(y, x)\big), \qquad \delta^-(x) \ := \ \min_{y \notin \mathcal{X}^\star} \ \Big(\max\!\big(M(y, x), 0\big) + \Delta(y)\Big),$$

and define the Pareto gaps for $x \in \mathcal{X}^\star$

$$\Delta(x) = \min\big\{\delta^+(x),\, \delta^-(x)\big\}.$$

In Algorithm 4, we use the empirical estimates of the Pareto gaps for arm elimination.

B.2 EXTENSION TO GENERAL $m \geq 2$

Our framework extends naturally to settings with more than two objectives. The key quantities underlying our algorithms, such as the Pareto gaps and constrained gaps, can be generalized to m-dimensional objective spaces without changing the fundamental structure of the elimination or selection rules. The computational overhead associated with these extensions is modest: dominance checks and gap computations scale polynomially with m and remain efficient for the small number of objectives typically encountered in practice. Thus, the proposed algorithms retain both conceptual simplicity and computational tractability when applied to $m > 2$ multi-objective prompt selection problems.

C Experiments details and results

C.1 Models and template

We uses two target models Gemma-7b-it (Team et al., 2024) and Llama3-8b-instruct (Grattafiori et al., 2024). For the instruction models, we adopt the recommended system template as Figure 4.

System instruction template

```
<|begin_of_text|><|start_header_id|>system<|end_header_id|>

You are an intelligent assistant. Please finish the given task,
answer with the output only and reply nothing else.

<|eot_id|><|start_header_id|>user<|end_header_id|>

<prompt>

<|eot_id|><|start_header_id|>assistant<|end_header_id|>
```

Figure 4: The system prompt template is used for both Gemma and Llama3.

In addition, to generate the prompt sets, we use the generation template as Figure 5.

Prompt Generation Template

```
Input: [Input1]
Output: [Output1]


Input: [Input2]
Output: [Output2]


...

Please provide the instruction now.
```

Figure 5: Prompt generation with 3-5 examples.

C.2 Metrics used in Summarization Tasks

In our experiments, we mainly consider two metrics for the summarization task, the ROUGE and Brevity score.

ROUGE Score. The ROUGE (Lin, 2004) score is a widely used metric for the summarization task. It captures the token-wise similarity between two texts. We use its variant, ROUGE-Lsum, provided by the Hugging Face **evaluate** python package. In general, it computes the longest common subsequences between the generated text and the reference, and aggregates the results across sentences. It captures the similarity between the generated summary from the LLM and the gold reference from the dataset.

Brevity Score. The Brevity score function is an artificial score to map the token lengths into the range $[0, 1]$. It is a piecewise function

$$f_B(x) = \begin{cases} 1, & x \leq \tau_{\text{low}}, \\ \dfrac{\tau_{\text{high}} - x}{\tau_{\text{high}} - \tau_{\text{low}}}, & \tau_{\text{low}} < x < \tau_{\text{high}}, \\ 0, & x \geq \tau_{\text{high}} \ . \end{cases}$$

where x is the token length and $\tau_{\text{low}} < \tau_{\text{high}}$.

C.3 Implementation Details with General Reward Structure

We now specify the algorithm for general reward functions. We set the SCHEDULER to be Sequential Halving. A uniform budget ALLOCATOR is adopted in each round for ease of implementation.

For the ESTIMATOR, a neural network with parameter θ is used to approximate the reward function g_θ. In each round r, we use the observations collected from the pulled arms to optimize θ. Specifically, we let

$$\mathcal{L}_r(\theta; \lambda) = \tfrac{1}{n_r} \sum_{t=1}^{n_r} \|g_\theta(\phi^{(t)}) - f^{(t)}\|_2^2 + \lambda \|\theta\|_2^2. \tag{15}$$

where the ℓ_2 regularization is adopted to prevent model overfitting.

Then, the estimator obtains the estimates according to

$$\hat{\theta}_r = \arg\min_\theta \mathcal{L}_r(\theta; \lambda), \quad \widehat{\mu}_r(x) = g_{\hat{\theta}_r}(\phi(x)), \quad \forall x \in A_r.$$

C.4 Experimental Rsults

C.4.1 Best Feasible Prompt Identification

We list our main results of best feasible prompt identification in Table 3 and Table 4.

C.4.2 Pareto Prompt Set Identification

Here, we list our main results of Pareto prompt set identification in Table 5 and Table 6.

C.5 Ablation Studies

C.5.1 Distribution of prompt objectives

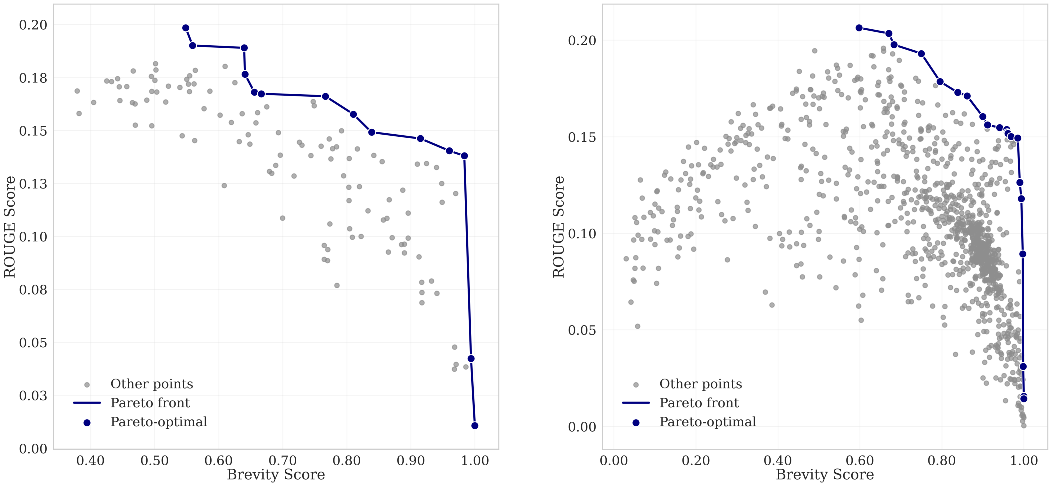
We visualize the objective distributions of the generated prompts to highlight the trade-off between the two metrics and to validate the reasonableness of our prompt-generation process and experimental design.

Figure 6a shows the objective distribution of the 100 generated prompts used in our main XSum experiments. The trade-off between the two metrics is clearly visible, demonstrating that identifying the Pareto-optimal prompts or the optimal feasible prompt is indeed a meaningful task in this environment.

We further extend the prompt set to 1000 candidates in Figure 6b. For this larger pool, no manual filtering is applied. Comparing the two figures, we observe that manual filtering primarily removes prompts that perform poorly on both Brevity and ROUGE, while preserving the overall structure of the distribution. Moreover, the Pareto-front shapes in both figures are similar, indicating that manual filtering does not materially alter the key characteristics of the candidate prompt set.

C.5.2 Varying Evaluation Budget

We next analyze how performance changes as the evaluation budget per prompt-selection run increases.



(a) 100 prompts

(b) 1000 prompts

Figure 6: Distributions of the objectives for generated prompts on Xsum.

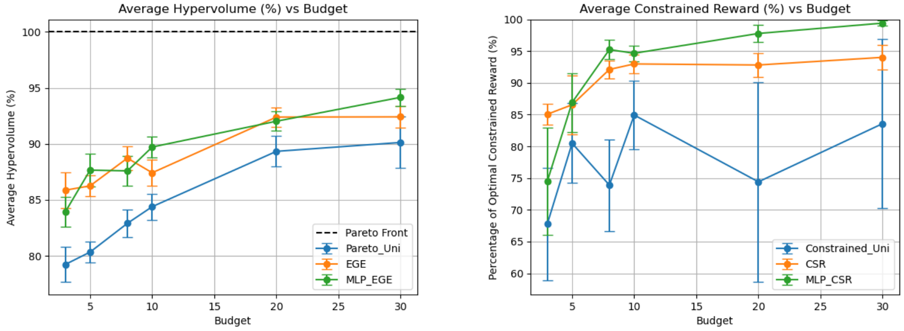


Figure 7: Average recovered hypervolume (%) and constrained reward on XSum using Llama-3.

Table 3: Average soft constrained reward on **XSum**.

K	Method	b = 3	b = 5	b = 8	b = 10
			Gemma-7B		
30	Uniform	0.015 ± 0.010	0.000 ± 0.000	0.030 ± 0.013	0.037 ± 0.014
	CSR	0.117 ± 0.013	0.137 ± 0.007	$\mathbf{0.144 \pm 0.000}$	$\mathbf{0.143 \pm 0.000}$
	MLP-CSR	$\mathbf{0.123 \pm 0.010}$	$\mathbf{0.139 \pm 0.002}$	0.140 ± 0.002	0.142 ± 0.001
50	Uniform	0.021 ± 0.011	0.021 ± 0.011	0.037 ± 0.014	0.036 ± 0.014
	CSR	0.122 ± 0.012	$\mathbf{0.143 \pm 0.001}$	0.141 ± 0.002	0.140 ± 0.002
	MLP-CSR	$\mathbf{0.143 \pm 0.002}$	0.142 ± 0.002	$\mathbf{0.147 \pm 0.001}$	$\mathbf{0.142 \pm 0.002}$
100	Uniform	0.021 ± 0.011	0.066 ± 0.016	0.037 ± 0.014	0.044 ± 0.015
	CSR	0.134 ± 0.007	0.141 ± 0.002	0.142 ± 0.001	0.143 ± 0.002
	MLP-CSR	$\mathbf{0.139 \pm 0.002}$	$\mathbf{0.142 \pm 0.002}$	$\mathbf{0.145 \pm 0.001}$	$\mathbf{0.147 \pm 0.001}$
			Llama3-8B		
30	Uniform	0.048 ± 0.016	0.069 ± 0.019	0.080 ± 0.020	0.089 ± 0.020
	CSR	$\mathbf{0.161 \pm 0.003}$	$\mathbf{0.154 \pm 0.004}$	$\mathbf{0.148 \pm 0.002}$	$\mathbf{0.149 \pm 0.003}$
	MLP-CSR	0.143 ± 0.003	0.126 ± 0.010	0.147 ± 0.004	$\mathbf{0.149 \pm 0.002}$
50	Uniform	0.063 ± 0.018	0.045 ± 0.017	0.051 ± 0.018	0.078 ± 0.019
	CSR	$\mathbf{0.144 \pm 0.011}$	0.134 ± 0.010	0.122 ± 0.014	$\mathbf{0.157 \pm 0.002}$
	MLP-CSR	0.141 ± 0.008	$\mathbf{0.151 \pm 0.003}$	$\mathbf{0.154 \pm 0.003}$	0.156 ± 0.002
100	Uniform	0.113 ± 0.015	0.134 ± 0.010	0.123 ± 0.012	0.141 ± 0.009
	CSR	$\mathbf{0.141 \pm 0.003}$	$\mathbf{0.144 \pm 0.008}$	0.153 ± 0.002	0.154 ± 0.002
	MLP-CSR	0.124 ± 0.001	$\mathbf{0.144 \pm 0.008}$	$\mathbf{0.158 \pm 0.003}$	$\mathbf{0.157 \pm 0.002}$

Figure 7 illustrates how the recovered hypervolume changes as we increase the evaluation budget for both our methods and the baselines. As expected, all algorithms benefit from larger budgets, but our methods consistently outperform the uniform-pulling baseline across all settings. This confirms that increasing the budget does not alter the overall conclusions. In the main experiments, we focus on smaller budgets in order to highlight that our algorithms remain effective even in the low-budget regime, which is the setting of primary interest.

C.5.3 Experiments with an Expanded Candidate Prompt Pool

We next examine how the algorithms behave when we increase the size of the candidate prompt pool.

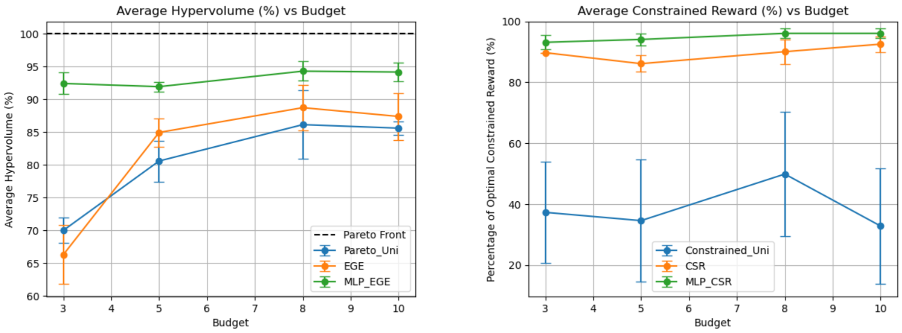


Figure 8: Average recovered hypervolume (%) and average constrained reward (%) with 1000 candidate prompts on XSum using Llama-3.

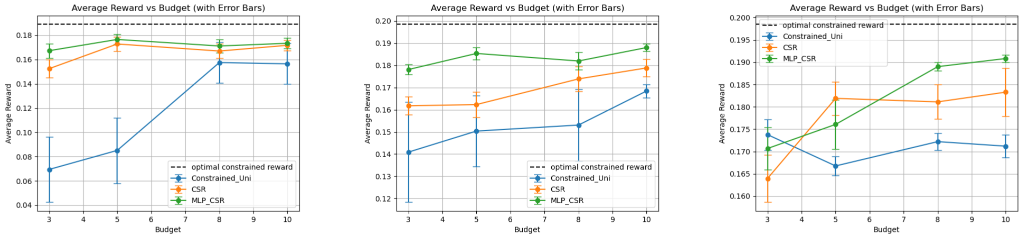
Table 4: Average soft constrained reward on **CNN/DailyMail**.

K	**Method**	**b = 3**	**b = 5**	**b = 8**	**b = 10**
Gemma-7B					
30	Uniform	0.021 ± 0.014	0.032 ± 0.017	0.021 ± 0.014	0.021 ± 0.014
	CSR	0.081 ± 0.018	0.134 ± 0.020	0.153 ± 0.012	$\mathbf{0.166 \pm 0.010}$
	MLP-CSR	$\mathbf{0.138 \pm 0.016}$	$\mathbf{0.164 \pm 0.011}$	$\mathbf{0.164 \pm 0.005}$	0.163 ± 0.005
50	Uniform	0.007 ± 0.007	0.020 ± 0.013	0.020 ± 0.013	0.000 ± 0.000
	CSR	0.091 ± 0.019	0.161 ± 0.013	0.161 ± 0.010	$\mathbf{0.167 \pm 0.011}$
	MLP-CSR	$\mathbf{0.146 \pm 0.015}$	$\mathbf{0.162 \pm 0.008}$	$\mathbf{0.172 \pm 0.007}$	0.153 ± 0.006
100	Uniform	0.015 ± 0.015	$\mathbf{0.045 \pm 0.022}$	0.000 ± 0.000	0.015 ± 0.015
	CSR	$\mathbf{0.088 \pm 0.023}$	0.015 ± 0.015	$\mathbf{0.062 \pm 0.024}$	0.032 ± 0.021
	MLP-CSR	0.032 ± 0.020	$\mathbf{0.045 \pm 0.027}$	0.032 ± 0.020	$\mathbf{0.072 \pm 0.026}$
Llama3-8B					
30	Uniform	0.000 ± 0.000	0.000 ± 0.000	0.008 ± 0.008	0.000 ± 0.000
	CSR	$\mathbf{0.157 \pm 0.002}$	$\mathbf{0.136 \pm 0.015}$	$\mathbf{0.148 \pm 0.011}$	$\mathbf{0.154 \pm 0.009}$
	MLP-CSR	0.142 ± 0.011	0.126 ± 0.015	0.142 ± 0.011	$\mathbf{0.154 \pm 0.004}$
50	Uniform	0.015 ± 0.010	0.016 ± 0.011	0.008 ± 0.008	0.010 ± 0.010
	CSR	$\mathbf{0.154 \pm 0.000}$	$\mathbf{0.157 \pm 0.009}$	0.155 ± 0.009	$\mathbf{0.165 \pm 0.003}$
	MLP-CSR	0.144 ± 0.005	0.155 ± 0.004	$\mathbf{0.160 \pm 0.003}$	0.160 ± 0.004
100	Uniform	0.019 ± 0.013	0.044 ± 0.017	0.018 ± 0.012	0.018 ± 0.012
	CSR	0.123 ± 0.014	0.138 ± 0.011	0.151 ± 0.008	$\mathbf{0.160 \pm 0.003}$
	MLP-CSR	$\mathbf{0.134 \pm 0.011}$	$\mathbf{0.154 \pm 0.003}$	$\mathbf{0.154 \pm 0.003}$	$\mathbf{0.160 \pm 0.003}$

Figure 8 reports the recovered hypervolume when scaling the candidate pool to 1000 prompts under different evaluation budgets. We observe that MLP-EGE performs best in this larger-pool setting, and EGE outperforms the uniform baseline when the budget is sufficiently large. These results indicate that our algorithms remain robust as the prompt pool grows substantially in size.

C.5.4 Varying constraint

Finally, we vary the constraint threshold and track the feasible average reward under different constraints in Figure 9. It indicates that our proposed algorithms consistently outperform the uniform baseline, and the choice of constraint does not compromise the effectiveness of our algorithms.



(a) Constraint = 0.6

(b) Constraint = 0.5

(c) Constraint = 0.4

Figure 9: Feasible average reward vs. per-armed budget on XSum using Llama3 with varying constraints.

C.6 Computing resources and costs

Our experiments are conducted on a server equipped with an AMD EPYC 9554 CPU, 755 GiB of system memory, and four NVIDIA H100 PCIe GPUs (80 GiB HBM each; driver 550.144.03)

Table 5: Hypervolume (HV) on **XSum**.

K	Method	b = 3	b = 5	b = 8	b = 10
			Gemma-7B		
30	Uniform	0.1105 ± 0.0019	0.1127 ± 0.0021	0.1165 ± 0.0013	0.1173 ± 0.0012
	EGE	0.1115 ± 0.0020	$\mathbf{0.1172 \pm 0.0014}$	$\mathbf{0.1188 \pm 0.0015}$	0.1193 ± 0.0013
	MLP-EGE	$\mathbf{0.1147 \pm 0.0021}$	0.1138 ± 0.0023	0.1172 ± 0.0018	$\mathbf{0.1195 \pm 0.0015}$
50	Uniform	0.1105 ± 0.0023	0.1169 ± 0.0019	0.1181 ± 0.0017	0.1192 ± 0.0014
	EGE	$\mathbf{0.1126 \pm 0.0028}$	$\mathbf{0.1180 \pm 0.0018}$	$\mathbf{0.1207 \pm 0.0019}$	$\mathbf{0.1212 \pm 0.0016}$
	MLP-EGE	0.1117 ± 0.0024	0.1152 ± 0.0025	0.1174 ± 0.0018	0.1187 ± 0.0012
100	Uniform	0.1007 ± 0.0013	0.1023 ± 0.0016	0.1084 ± 0.0017	0.1126 ± 0.0014
	EGE	0.1085 ± 0.0013	$\mathbf{0.1187 \pm 0.0010}$	$\mathbf{0.1218 \pm 0.0007}$	0.1219 ± 0.0007
	MLP-EGE	$\mathbf{0.1138 \pm 0.0012}$	0.1179 ± 0.0013	0.1216 ± 0.0009	$\mathbf{0.1220 \pm 0.0009}$
			Llama3-8B		
30	Uniform	0.1433 ± 0.0026	0.1481 ± 0.0023	0.1472 ± 0.0031	0.1493 ± 0.0031
	EGE	$\mathbf{0.1614 \pm 0.0008}$	$\mathbf{0.1496 \pm 0.0022}$	0.1535 ± 0.0031	$\mathbf{0.1560 \pm 0.0015}$
	MLP-EGE	0.1488 ± 0.0026	$\mathbf{0.1496 \pm 0.0031}$	$\mathbf{0.1577 \pm 0.0019}$	0.1548 ± 0.0017
50	Uniform	0.1500 ± 0.0022	0.1557 ± 0.0024	0.1586 ± 0.0020	0.1585 ± 0.0021
	EGE	$\mathbf{0.1626 \pm 0.0004}$	$\mathbf{0.1587 \pm 0.0023}$	$\mathbf{0.1596 \pm 0.0019}$	0.1604 ± 0.0020
	MLP-EGE	0.1520 ± 0.0020	0.1552 ± 0.0022	0.1595 ± 0.0013	$\mathbf{0.1616 \pm 0.0021}$
100	Uniform	0.1423 ± 0.0028	0.1443 ± 0.0016	0.1489 ± 0.0022	0.1516 ± 0.0021
	EGE	$\mathbf{0.1542 \pm 0.0028}$	0.1549 ± 0.0016	$\mathbf{0.1594 \pm 0.0019}$	0.1570 ± 0.0021
	MLP-EGE	0.1507 ± 0.0024	$\mathbf{0.1574 \pm 0.0027}$	0.1573 ± 0.0024	$\mathbf{0.1611 \pm 0.0016}$

running CUDA 12.4.131. We employ Gemma and Llama-3 in quantized, inference-only settings. Generally, the experiments are expected to be reproduced on a server with over 20G GPU memory.

D Language Model usage in paper writing disclosure

ChatGPT 5 is used during the writing of the paper, mainly for paraphrasing the sentences, improving the grammar, reorganizing the sentences in paragraphs, and generating a few paragraphs based on human-provided outlines. All the generated texts are reviewed and revised by humans. All technical claims, definitions, algorithms, theorems, and proofs are written by humans.

Table 6: Hypervolume (HV) on **CNN/DailyMail**.

K	**Method**	**b = 3**	**b = 5**	**b = 8**	**b = 10**
			Gemma-7B		
30	Uniform	0.1345 ± 0.0036	0.1393 ± 0.0029	0.1484 ± 0.0019	0.1459 ± 0.0021
	EGE	0.1327 ± 0.0038	0.1458 ± 0.0026	0.1472 ± 0.0023	0.1453 ± 0.0036
	MLP-EGE	**0.1441 ± 0.0029**	**0.1499 ± 0.0019**	**0.1505 ± 0.0016**	**0.1505 ± 0.0019**
50	Uniform	0.1310 ± 0.0038	0.1390 ± 0.0023	0.1414 ± 0.0021	0.1429 ± 0.0020
	EGE	0.1334 ± 0.0032	0.1413 ± 0.0020	**0.1455 ± 0.0024**	0.1445 ± 0.0019
	MLP-EGE	**0.1358 ± 0.0032**	**0.1430 ± 0.0017**	0.1434 ± 0.0020	**0.1463 ± 0.0016**
100	Uniform	0.1280 ± 0.0015	0.1305 ± 0.0015	0.1325 ± 0.0017	0.1342 ± 0.0018
	EGE	0.1279 ± 0.0014	0.1342 ± 0.0016	0.1343 ± 0.0016	**0.1411 ± 0.0013**
	MLP-EGE	**0.1316 ± 0.0016**	0.1336 ± 0.0019	**0.1379 ± 0.0021**	0.1408 ± 0.0015
			Llama3-8B		
30	Uniform	0.1577 ± 0.0039	0.1534 ± 0.0051	0.1685 ± 0.0028	0.1661 ± 0.0037
	EGE	0.1603 ± 0.0049	0.1680 ± 0.0028	0.1753 ± 0.0033	**0.1744 ± 0.0043**
	MLP-EGE	**0.1632 ± 0.0034**	**0.1688 ± 0.0039**	**0.1803 ± 0.0022**	0.1727 ± 0.0030
50	Uniform	0.1559 ± 0.0029	0.1543 ± 0.0031	0.1618 ± 0.0023	0.1646 ± 0.0031
	EGE	**0.1651 ± 0.0023**	0.1647 ± 0.0022	**0.1706 ± 0.0032**	**0.1720 ± 0.0023**
	MLP-EGE	0.1618 ± 0.0033	**0.1628 ± 0.0027**	0.1671 ± 0.0028	0.1673 ± 0.0026
100	Uniform	0.1519 ± 0.0024	0.1579 ± 0.0022	0.1624 ± 0.0019	0.1631 ± 0.0023
	EGE	0.1503 ± 0.0028	0.1639 ± 0.0024	0.1690 ± 0.0020	**0.1754 ± 0.0022**
	MLP-EGE	**0.1628 ± 0.0024**	**0.1684 ± 0.0020**	**0.1699 ± 0.0023**	0.1716 ± 0.0023