Isolating the Impact of Dimensionality on Bandit Regret via Fixed-Gap Gaussian Environments

Luhui Zheng

{ luhuizheng18@gmail.com }

School of Physics and Optoelectronics, South China University of Technology, Guangzhou, 510640, Guangdong, China

Abstract. This study tests how Explore-Then-Commit (ETC), Upper Confidence Bound (UCB), and Thompson Sampling (TS) behave in a static setting where the action space size, K, varies from 10 to 2,400. A Fixed-Gap Gaussian Environment based on MovieLens data is used to ensure the task difficulty stays the same. Experiments show that performance gets worse for every algorithm as K increases. ETC fails the most, with identification accuracy dropping below 20% because it wastes too much budget on exploration. UCB and TS do better but still show linear regret growth. This implies that probabilistic exploration is not enough to handle a massive number of sub-optimal arms. The findings suggest that standard strategies need structural changes, like subsampling, to work efficiently in large-scale environments.

Keywords: Multi-Armed Bandits, Action Space Scalability, Best Arm Identification, Subsampling.

1 Introduction

Multi-Armed Bandit (MAB) algorithms are now core to many systems, ranging from recommendation engines to high-frequency trading. A major practical issue appears when the action space—specifically, the number of arms, K—gets very large. In these high-dimensional settings, the trade-off between exploring and exploiting becomes hard to manage. If K is close to or larger than the time horizon T, standard methods often fail. The cost of checking every poor option simply outweighs the benefit of finding the best one. As a result, knowing how performance drops as K grows is key to building systems that work at scale.

Theoretical research has recently looked at these massive action spaces [1]. For instance, Gong and Sellke studied the limit of this problem, known as infinite-armed bandits [2]. They found that identifying a near-optimal arm (a ϵ-optimal arm) depends entirely on the underlying reservoir distribution. Bayati et al. pointed out a specific failure mode when $K \geq T$ [3]. They showed that the standard Upper Confidence Bound (UCB) algorithm explores too

ICIAAI 2026, May 22-24, Singapore, Singapore

DOI 10.4108/eai.22-5-2026.2365143

much in this case, causing high regret. Interestingly, simple greedy strategies can sometimes work better there. On the empirical side, Zhao compared algorithms using the MovieLens 1M dataset [4]. However, that work focused on changing rewards (dynamic environments) rather than isolating the effect of dimensionality in a stable setting.

This leaves a gap in understanding the intrinsic scalability of these algorithms when the distributional difficulty is fixed, but the dimensionality varies. This paper investigates the isolated impact of action space size K on the cumulative regret of ETC, UCB, and TS algorithms. By enforcing a fixed Gaussian distribution on arm means to eliminate environmental physical shifts, the research empirically quantifies the scalability limits and degradation patterns of these strategies as K expands from finite to many-armed regimes.

2 Methodology

2.1 Preliminaries

The study focuses on the standard stochastic MAB problem characterized by a tuple ($\mathcal{A}$,$\mathcal{D}$). Let $\mathcal{A} = \{1, \dots, K\}$ denote the set of available actions, where the cardinality K is the primary variable of interest in this study. At each discrete time step $t \in [T]$, where T represents the total time horizon, the agent selects an arm $a_t \in \mathcal{A}$ and observes a scalar reward r_t. This reward is drawn independently from an unknown distribution ν_{a_t} associated with the selected arm, having an expectation μ_{a_t}. The optimal mean reward is denoted as $\mu^* = max_{a \in \mathcal{A}} \mu_a$.

The objective of the learning agent is to maximize the expected cumulative reward over the horizon T, which is equivalent to minimizing the cumulative regret R_T. The regret is defined as the expected difference between the optimal strategy and the agent's realized choices:

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \left(\mu^* - \mu_{a_t}\right)\right] = \sum_{a \in \mathcal{A}} \mathbb{E}[N_T(a)]\, \Delta_a \quad (1)$$

Where $N_T(a)$ denotes the number of times arm a has been pulled up to time T, and $\Delta_a = \mu^* - \mu_a$ represents the sub-optimality gap of arm a.

2.2 Algorithms Under Study

Three representative strategies governing the exploration-exploitation trade-off are analyzed, with specific parameterizations adapted for large action spaces.

The Explore-Then-Commit strategy rigidly separates exploration and exploitation. To ensure fairness in the large K regime, the minimax optimal exploration schedule is adopted rather than a fixed linear proportion. Specifically, each arm is sampled $m = max(1, (T/K)^{2/3}(\ln K)^{1/3})$ times. The primary advantage of ETC lies in its implementation simplicity and the theoretical clarity of its regret bounds. By decoupling information acquisition from reward maximization, it avoids the computational complexity of continuously updating confidence indices, making it a robust baseline for measuring the cost of exploration.

The Upper Confidence Bound algorithm, specifically the UCB1 variant, is employed to follow the principle of optimism in the face of uncertainty. At each step t, the algorithm selects the

arm that maximizes the index $\hat{\mu}_a(t) + \sqrt{2 \ln t / N_t(a)}$, where $\hat{\mu}_a(t)$ is the empirical mean and $N_t(a)$ is the number of times arm a has been selected up to time t, and the second term represents the confidence interval width. The fundamental advantage of UCB is its strong theoretical guarantee; it provides a non-asymptotic logarithmic regret bound by automatically balancing the exploration of less-sampled arms with the exploitation of high-performing ones. This deterministic mechanism ensures that the algorithm never permanently ignores an arm unless its sub-optimality is statistically confirmed.

Thompson Sampling uses a Bayesian approach. It assigns a Gaussian posterior distribution $\mathcal{N}(\hat{\mu}_a, \hat{\sigma}_a^2)$ to the mean reward of each arm. At each time step, a value is sampled randomly from the posterior distribution of every arm. The algorithm then chooses the arm associated with the highest sampled value. This method allows the selection probability to match the probability that an arm is optimal.

2.3 Construction of The Fixed-Gap Gaussian Environment

To strictly measure the effect of K, a specific setup called the Fixed-Gap Gaussian Environment is used. This design makes sure the identification difficulty stays the same, even when more arms are added.

The process starts by generating K target means from a global Gaussian distribution. These are based on MovieLens statistics. A key adjustment is then made: the gap between the best mean μ^* and the second-best mean is fixed at $\delta = 0.5$. This prevents the task from getting harder just because the gap shrinks.

Finally, the reward distributions are matched to these means. The raw ratings from the MovieLens dataset are shifted mathematically. For any arm, the original rating distribution is moved so that its average equals the assigned target mean. This keeps the realistic variance of user data but enforces the fixed-gap structure needed for the test.

3 Experiment

3.1 Experimental Setup and Overview

All tests take place in the Fixed-Gap Gaussian Environment based on MovieLens 1M data. The time horizon T is set to 100,000 steps. This gives the algorithms enough time to settle. To check scalability, the action space size K changes from 10 to 2,405 arms. The results reported here are averages from 50 separate runs to ensure they are reliable.

Figs 1 and 2 below use a standard color code. Blue lines with circles represent the Explore-Then-Commit (ETC) strategy. Red lines with squares mark the Upper Confidence Bound (UCB) algorithm. Green lines with triangles show Thompson Sampling (TS).

3.1 Cumulative Regret Analysis

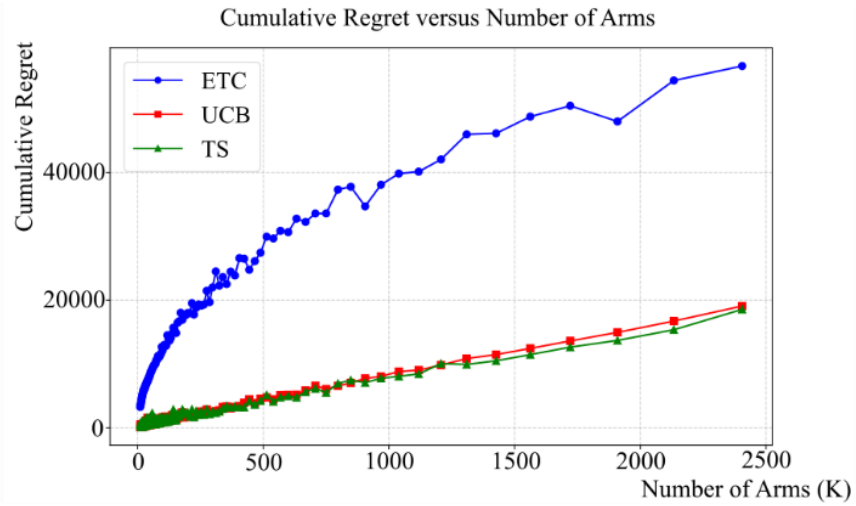

Fig. 1. Cumulative Regret Scaling with Action Space Size. Comparison of ETC, UCB, and TS algorithms as the number of arms K increases from 10 to 2,405 over a fixed horizon $T = 100{,}000$ (Photo/Picture credit: Original).

Fig. 1 plots the cumulative regret R_T. As the number of arms K grows, every algorithm loses more reward. ETC performs the worst. It ends up with the highest total loss. Its regret curve does look sub-linear relative to K, but the actual cost is much higher than the other methods. This happens because ETC spends a fixed chunk of time exploring, which wastes too many steps on bad arms.

UCB and TS, in contrast, show a linear growth pattern. Thompson Sampling does slightly better than UCB, with its curve sitting just below the red line. But the difference is small. This means that even though TS uses probabilistic exploration, it still finds it hard to handle the large number of sub-optimal arms. Both algorithms struggle equally to filter out these distractions.

3.2 Best Arm Identification Rate

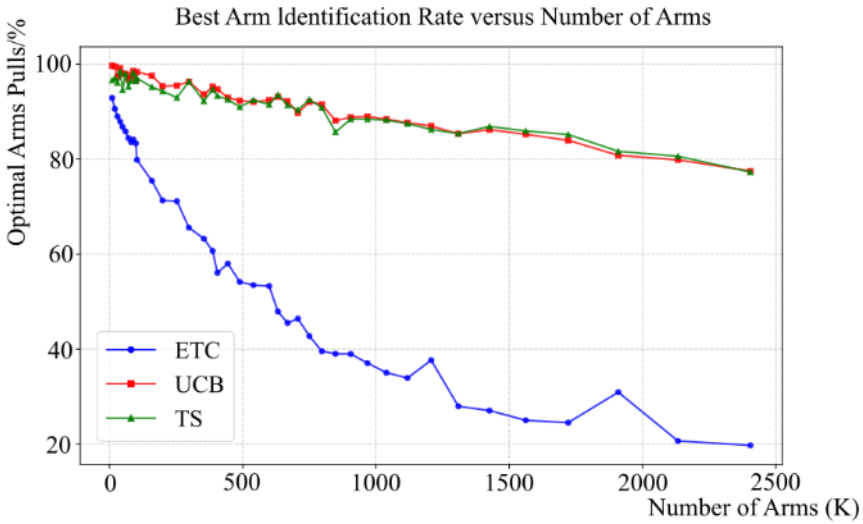

Fig. 2. Best Arm Identification Rate versus Number of Arms (K). The percentage of time steps where the optimal arm is selected for varying action space sizes (Photo/Picture credit: Original).

The results for the Best Arm Identification Rate appear in Fig. 2. There is a clear gap between static and adaptive methods. The ETC algorithm crashes. Its accuracy drops to about 20% once K hits 2,415. This low score shows that the algorithm rarely finds the best arm. Its budget is simply spread too thin across too many options.

UCB and TS show a steady, linear drop in accuracy. Their curves almost overlap. This suggests that the noise from the many sub-optimal arms hurts them both in the same way. As the action space expands, the chance that a bad arm looks like the best one goes up. This causes both strategies to make more mistakes.

4 Discussion

4.1 The Failure of Naive Exploration

At K=2,415, the ETC algorithm identifies the best arm only about 20% of the time. This sharp drop proves that naive exploration fails in high dimensions. The standard ETC strategy treats every arm as a separate box. It tries to test each one individually. This approach hits a wall when the number of arms is huge. The core problem is assuming independence.

Future improvements must stop exploring the entire space. Instead, finding connections between arms is key. Aouali proposes using linear diffusion models [5]. This method maps the large set of K arms to a smaller set of hidden features. Similarly, Zhou et al. show that grouping similar actions allows the agent to skip testing many poor arms [6]. These examples confirm that seeing the structure is better than blind testing.

4.2 The Necessity of Subsampling Strategies

UCB and TS face a different problem. Their performance drops linearly. This suggests that standard methods are inefficient when K is close to the horizon T. In the tests, both algorithms try to track data for every single arm. This creates a heavy startup cost because each option demands at least some attention. Randomness alone cannot filter out bad choices fast enough.

To stop this linear loss, algorithms need to get rid of bad arms earlier. Keeping a static list does not work. Baudry et al. argue that applying strict rules to remove arms is necessary [7]. Shao and Fang provide a working model for this [8]. They suggest using a fixed budget to eliminate sub-optimal arms dynamically. This ensures the agent focuses only on a few survivors, rather than wasting effort on a growing pile of weak options.

4.3 Reservoir Hardness and Distractor Density

These degradation patterns match recent theories about infinite-armed bandits. As Suk and Kim point out, the difficulty depends on the tail of the reward distribution [9]. In the Fixed-Gap Gaussian Environment, adding more arms adds more distractors. These are arms that look good but are not the best. Their number grows linearly with K. This noise confuses Thompson Sampling. The algorithm keeps picking these second-best options because their probability curves overlap with the winner.

Standard posterior sampling needs a tweak to handle this. Jeong et al. suggest adding Information Relaxation [10]. This technique penalizes arms that have high uncertainty but low value. It stops the agent from getting stuck in the long list of mediocre choices. Such corrections are vital when the environment is full of near-optimal noise.

5 Conclusion

This study tested how three classic bandit strategies—ETC, UCB, and Thompson Sampling—scale in high-dimensional settings. A Fixed-Gap Gaussian Environment was used. This isolated the effect of the action space size K from other environmental changes. Results show that performance drops for every algorithm as K grows from tens to thousands. ETC performed the worst. Its mandatory exploration phase took up too much of the time horizon. UCB and Thompson Sampling also struggled, showing a linear growth in regret. This means that finding the best arm gets harder just by adding more options, even if the reward gaps stay the same. Standard methods need help to cope.

The study has limits. It used an offline simulation with static data. This does not fully capture how real recommendation systems change over time. Also, assuming Gaussian distributions simplifies the heavy-tailed nature of real user ratings. Future work should focus on subsampling strategies, like limiting exploration to a random subset of arms. Extending this to Contextual Bandits would also be useful, as side information helps generalize across arms. Ultimately, the failure of these algorithms comes down to the high cost of exploration. Structural constraints are needed to make decisions efficiently in such large environments.

References

[1] A. Kazerouni and L. M. Wein, "Best arm identification in generalized linear bandits," Operations Research Letters, vol. 49, no. 3, pp. 365–371, 2021.

[2] E. X.-Y. Gong and M. Sellke, "Asymptotically optimal quantile pure exploration for infinite-armed bandits," in Advances in Neural Information Processing Systems (NeurIPS), vol. 36, 2023, pp. 52626–52654.

[3] M. Bayati, N. Hamidi, R. Johari, and K. Khosravi, "The unreasonable effectiveness of greedy algorithms in multi-armed bandit with many arms," in Advances in Neural Information Processing Systems (NeurIPS), vol. 33, 2020, pp. 17248–17257.

[4] B. Zhao, "Performance of multi-armed bandit algorithms in dynamic vs. static environments: A comparative analysis," ITM Web Conf., vol. 73, Art. no. 01016, 2025.

[5] I. Aouali, "Linear diffusion models meet contextual bandits with large action spaces," in NeurIPS 2023 Workshop on Foundation Models for Decision Making, 2023.

[6] Q. Zhou, M. Kozdoba, and S. Mannor, "Representative action selection for large action space meta-bandits," arXiv preprint arXiv:2505.18269, 2025.

[7] D. Baudry, Y. Russac, and O. Cappé, "On limited-memory subsampling strategies for bandits," in Proc. 38th Int. Conf. Mach. Learn. (ICML), vol. 139, 2021, pp. 752–762.

[8] Y. Shao and Z. Fang, "Linear streaming bandit: Regret minimization and fixed-budget epsilon-best arm identification," in Proc. 39th AAAI Conf. Artif. Intell. (AAAI), 2025.

[9] J. Suk and J. Kim, "Tracking most significant shifts in infinite-armed bandits," arXiv preprint arXiv:2502.00108, 2025.

[10] W. Jeong and S. Min, "Improving Thompson Sampling via Information Relaxation for Budgeted Multi-armed Bandits," in Proceedings of the Reinforcement Learning Conference (RLC), 2024, pp. 1–15.