Multi-Agent Contextual Bandits for Multi-Objective Recommendation Systems

Ziwen Zhu

First College of Liberal Arts and Sciences, University of Florida, Gainesville, FL, 32612, USA

Abstract. In this work, the use of multi-agent contextual bandits to optimize multi-objective recommendation systems in resource-constrained settings is examined. A baseline single-agent model with mean-aggregated rewards, a homogeneous multi-agent system with identical Bayesian Linear Thompson Sampling agents, and a heterogeneous multi-agent framework with a unique contextual bandit algorithm assigned to each objective were the three experimental configurations that were built. To resolve conflicts between conflicting aims, diversity limitations, and criticality-based adaptive weighting were implemented. In comparison to homogeneous and single-agent baselines, the results show that heterogeneous agent configurations produce faster convergence, reach steady state faster within the first 20000 steps, larger suggestion diversity, and more effective specialization across reward targets, with a Generalized Gini Index (GGI) score of around 11 indicating diverse weight across reward features. They do, however, also make agent coordination more difficult. Results indicate that, especially in settings with extremely varied reward distributions, heterogeneous multi-agent systems offer a useful compromise between efficiency and adaptability.

1 Introduction

While Large Language Models (LLMs) have recently shown impressive capabilities in multi-objective optimization for recommender systems, the performance of these models often comes at the cost of substantial computational and data demands. These models require extensive pretraining, large-scale datasets, and powerful hardware to operate effectively in online environments. These requirements limit their practicality for systems with constrained resources. In contrast, multi-armed bandit (MAB) algorithms offer a lightweight, online learning alternative that is especially suitable due to their adaptation capabilities and compatibility with limited computing budgets.

In addition to classic MABs, contextual bandits—an extension of the MAB framework—incorporate side information, such as user behavior, session metadata, or temporal cues, to inform more dynamic and personalized decisions. Algorithms like Linear Upper Confidence Bound (LinUCB) have historically shown strong results in leveraging context for real-time recommendation. Recent work has pushed this further by introducing mechanisms that

Ziwen.zhu@ufl.edu

"avoid pessimistic bias in unseen actions while retaining uncertainty calibration", enable "uncertainty-aware reasoning even in high-dimensional offline environments", and "combine overparameterized neural models with bootstrapped or conformal prediction intervals" to retain statistical reliability under distributional shift [1-3]. In this work, the problem of multi-objective recommendation is the problem where the system must balance competing goals, such as maximizing user clicks, cart additions, and final purchases. So, a triple-agent contextual bandit setting is introduced, in which each agent selfishly optimizes one behavioral signal: clicks (frequent), carts (intermittent), and orders (sparse). By assigning each objective to a specialized agent—Neural Upper Confidence Bound (NeuroUCB) for high-frequency feedback, contextual Thompson Sampling for mid-frequency actions, and Bootstrapped Thompson Sampling for rare, high-value outcomes—offering a modular, interpretable, and resource-efficient framework for multi-objective optimization that scales better in constrained environments than centralized, monolithic models like LLMs.

2 Related Work

Recent advances in contextual multi-armed bandit (CMAB) algorithms have laid a solid groundwork for increasingly adaptive and scalable recommender systems. Among these, a significant area of development involves multi-agent architectures for handling multi-objective optimization, where different behavioral goals—such as maximizing user clicks, cart additions, or purchase completions—must be balanced within a shared learning environment. These works extend the classic bandit paradigm by exploring how decentralized agents, each tuned to a distinct objective, can cooperatively optimize overall recommendation quality. Zafar et al. present an effective adaptive multi-agent system for electric vehicle charging that leverages combinatorial MABs, enabling localized decision-making with partial observability and criticality-based interaction rules [4]. Although focused on grid control, their framework highlights the utility of lightweight, model-free MAB agents in high-scale coordination scenarios—a highly applicable concept to modular recommender systems, where each agent targets a distinct engagement goal. Chen and Hou's Contextual Restless Bandits framework offers a powerful generalization for scenarios where arm rewards evolve over time [5]. By jointly modeling context and temporal dynamics using a Markovian structure, they introduce an index-based strategy with online learning extensions, demonstrating robustness in nonstationary environments—a crucial property for modern recommendation tasks where user preferences change rapidly.

In the domain of mental health, the CAREForMe system exemplifies how CMABs can be deployed for personalized interventions using streaming sensor data [6]. Their architecture decouples user clustering, state tracking, and contextual learning to handle feedback sparsity, while UCB-based bandits drive recommendation updates. This modularity resonates with specialized agents for differing reward densities, such as click-heavy versus sparse conversion events. Wakayama and Ahmed propose observation-augmented CMABs using Bayesian inference and probabilistic semantic data association [7]. Their approach allows noisy, semantic human inputs to be incorporated into active learning loops for robotic exploration. Their mixture-based prior modeling inspires the use of session-level features like hour-of-day and interaction type, while informing reward variance assumptions for agent tuning. Finally, NeuroSep-CP-LCB introduces a hybrid neural-CMAB architecture that combines conformal prediction with deep networks for calibrated decision-making in healthcare [8]. Its ability to make uncertainty-aware decisions under offline data constraints supports the choice to use NeuroUCB for high-frequency click-based objectives, where nonlinearity and adaptability are essential. Together, these contributions illustrate a broadening of the CMAB toolkit—from traditional UCB and Thompson Sampling baselines to hybrid systems that incorporate variance awareness, temporal modeling, semantic

uncertainty, and deep learning. This paper builds on these strategies by embedding them within a multi-agent contextual framework, where each agent specializes in a distinct objective and cooperates via shared contextual signals. This architecture aims to maintain interpretability, reduce computational overhead, and enable objective-specific adaptability in complex recommendation environments.

3 Methodology

This section includes a detailed work-through of the methodologies and techniques used in the multi-agent contextual bandit system. This section will focus on the feature processing, rationale behind the selection of contextual bandit algorithms for individual objectives, together with techniques that ease reward conflicts, and the evaluation metrics used on the system.

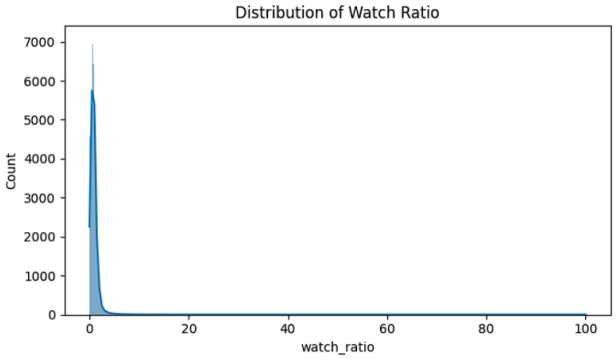
3.1 Data Preprocessing

The dataset of "KuaiRec" contains 465065 records, each described by 54 features that fall into two groups after processing: three reward targets (watch_ratio, completion_rate, and exposure_rate), and 21 other contextual attributes (with compressed one-hot tag columns) [9]. Major feature engineering on the cleaned and organized dataset includes percentile-based clipping and normalization to reward features to enhance numerical stability. The normalization process for each reward metric is defined as:

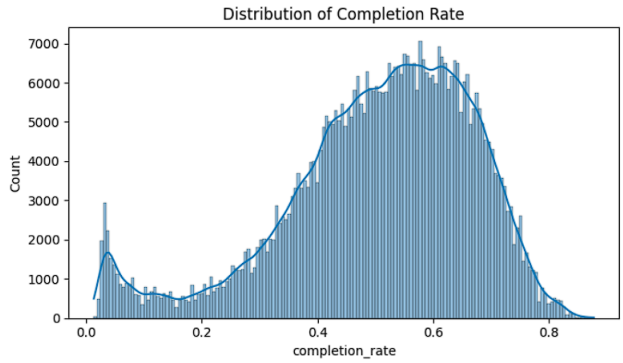
$$r_{normalized} = \frac{clip(r, r_5, r_{95}) - r_5}{r_{95} - r_5 + \epsilon} \quad (1)$$

where r is the original reward metric, r_5 and r_{95} denote the 5th and 95th percentiles of the reward distribution, respectively, and ϵ represents a small constant to prevent the division by zero problem. Temporal features, like hour or weekday, and interaction-based features, like daily_play_mean, daily_play_std, friend_count, and categorical tags tag_0 through tag_30, were systematically selected using causal scores to enrich the context representation and effectively capture more underlying user engagement behaviors.

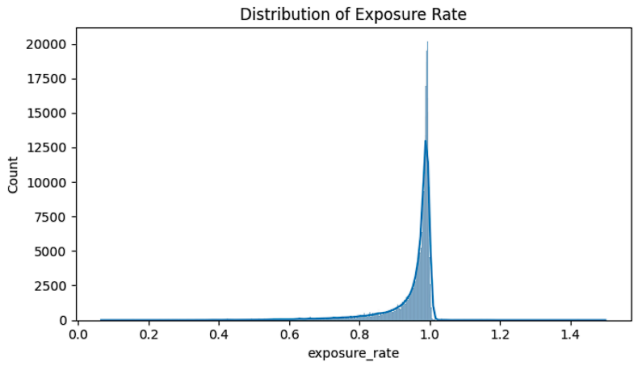
3.2 Model Selection



(a)



(b)



(c)

Fig. 1 (a) Distribution of reward objectives: watch_ratio (Picture credit: Original), (b) Distribution of reward objectives: completion_rate (Picture credit: Original), (c) Distribution of reward objectives: exposure_rate (Picture credit: Original)

To adapt to the multi-objective optimization workflow, a one-agent-for-one-goal structure is used in this problem, where each agent independently optimizes their own reward feature using distinct algorithms according to the feature characteristics, as shown in Fig. 1.

Bayesian Linear Thompson Sampling (TS): The 'watch_ratio' distribution is highly skewed, with most values clustered near zero and a long tail toward larger values. Bayesian Linear TS was selected to handle this sparsity by effectively modeling uncertainty and promoting cautious exploration. It updates its parameters by maximizing the Evidence Lower Bound (ELBO), which is formally expressed as:

$$ELBO(\theta) = \sum_{i=1}^{N} log\, p(y_i \mid x_i, \theta) - KL\,(q(\theta) \parallel p(\theta)) \tag{2}$$

$p(y_i \mid x_i, \theta)$ is the likelihood of observing label y_i given the input x_i and the parameter θ . $q(\theta)$ is the variational distribution approximating the true posterior of parameters. $p(\theta)$ is the prior distribution over model parameters. $KL\,(q(\theta) \parallel p(\theta))$ is the Kullback-Leibler divergence that penalizes deviation of the approximate posterior from the prior.

Contextual TS: The 'completion_rate' distribution is approximately symmetric and smooth, resembling a bell curve. Contextual TS was thus chosen due to its efficiency and robustness when rewards are moderately distributed and context sensitive. The parameter update is described by the Gaussian posterior distribution shown as:

$$\theta \sim N(\hat{\theta}, v^2 B^{-1}) \tag{3}$$

where B represents the covariance matrix of the observed contexts, and $\hat{\theta}$ is estimated based on historical context-reward interactions.

LinUCB: The 'exposure_rate' distribution is sharply peaked near 1.0, indicating low variance across samples. For this reason, LinUCB was employed to quickly exploit the small range of high-reward options through deterministic confidence-based action selection. The action selection criterion of this algorithm is defined as follows:

$$p(x) = x^T \hat{\theta} + \alpha\sqrt{x^T A^{-1} x} \tag{4}$$

where x denotes the context vector, α is the exploration parameter, $\hat{\theta}$ represents the parameter estimates from prior observations, and A^{-1} is the inverse covariance matrix of the context features.

3.3 Adaptive Balancing Mechanisms

With multiple reward targets, there is great potential for conflicts among them, which makes balancing between rewards necessary for such a system.

Criticality Score: This score is used to ensure balanced representation across objectives by dynamically computing for each agent and later comparing together. Criticality scores quantify each agent's urgency to optimize its objective based on recent performance. The score and normalization process are computed by:

$$C_{agent} = \frac{exp(S_{agent})}{\sum_i exp(s_i)} \tag{5}$$

$$s_{agent} = 1 - \bar{r}_\tau \tag{6}$$

C_{agent} is the normalized criticality score assigned to a specific agent. It reflects the relative urgency of this agent to improve its performance, compared to all other agents. $\exp(S_{agent})$ is the exponentiated raw criticality score of the current agent, which amplifies differences among agents. $\sum_i \exp(s_i)$ is the sum of exponentiated scores for all agents in the system, serving as a normalization factor to ensure that all C_{agent} values sum to 1. s_{agent} is the unnormalized urgency score for the agent, representing how far the agent is from optimal performance based on recent rewards. $\bar{r}_\tau$ is the average reward ($\bar{r}$) received by the agent over a recent fixed-size time window τ. A lower mean reward leads to a higher s_{agent}, signaling the need for more attention in subsequent recommendation rounds. This softmax-based criticality scoring ensures that agents' final recommendation lists are balanced, favoring agents experiencing suboptimal recent performance to foster equitable optimization across all agents.

Diversity Constraints: Another action to reduce reward conflicts and encourage equal optimization among all three agents is the soft conflict penalty, which directly influences the

selection of each agent's recommendation candidates. The constraint score is defined as follows:

$$score_a = score_o - \lambda \sum_{other\ agents} \max(0, \gamma - r_o) \quad (7)$$

$score_a$ is the final penalized score used to rank a candidate item for an agent, after accounting for potential conflicts with other agents. $score_o$ is the original output score computed by the agent's own bandit algorithm based on contextual features and learned parameters.λ is a tunable penalty weight that determines how strongly an agent should be discouraged from selecting items that perform poorly for other agents. In the experiment, this is fixed at $\lambda = 0.2$, chosen heuristically to balance diversity without excessively harming primary agent performance. γ is a reward threshold that defines the minimum acceptable reward from other agents. Rewards lower than γ trigger a penalty. In the implementation, this is set to 0.5, reflecting a neutral benchmark to distinguish acceptable versus poor performance across agents. r_o is the expected reward of the current candidate item as estimated by another agent (not the one making the current decision).

3.4 Evaluation Metrics

Two metrics were chosen to evaluate the effectiveness of the system's performance:

Cumulative Regret: Cumulative regret is commonly used in MAB experiment evaluation to see how the algorithm learnt from the environment. It measures the accumulated differences between the optimal available rewards and the actual obtained rewards at each time step. Lower cumulative regret indicates higher recommendation efficiency. Regret is defined as:

$$R(T) = \sum_{t=1}^{T} (r_t^* - r_t) \quad (8)$$

where the r_t^* denotes the optimal reward at time step t, and r_t is the reward received by the selection made by the algorithm.

GGI: The Generalized Gini Index quantifies the capability of the model to balance between multiple reward objectives. The balance of distributional fairness of rewards achieved by the system is computed as:

$$GGI = \frac{1}{T} \sum_{t=1}^{T} \sum_{i=1}^{n} (2i - n - 1) r_{(i)}^t \quad (9)$$

T is the total number of timesteps in the simulation. n is the Number of reward objectives. $r_{(i)}^t$ is the i-th smallest reward at timestep t, i.e., the sorted reward vector for that timestep. GGI applies a linear weighting to the sorted reward vector, assigning positive weight to higher-ranked rewards and negative weight to lower-ranked ones, thus emphasizing disparity. A lower value of this metric occurs when the reward values are numerically closer to each other, showing a more equally balanced system where all objectives are treated similarly. However, a lower GGI does not mean the system is optimal in all cases. It only indicates that the model is not prioritizing any reward feature. In contrast, a higher GGI reflects that some

objectives are receiving more favorable outcomes, which may either indicate an intentional prioritization or a learned imbalance. In terms of this experiment, the GGI metric is used comparatively across different algorithmic configurations. For instance, the single-agent Bayesian Linear TS model shows a significantly lower GGI due to its use of an averaged reward target, which inherently aligns all reward dimensions. Meanwhile, multi-agent settings tend to have higher GGI values because their agents optimize independently with possible trade-offs, thus exhibiting more imbalance but also allowing adaptive prioritization. This makes GGI not just a fairness diagnostic but also a proxy to understand how the system dynamically shifts its focus during training and when such prioritization stabilizes over time [10].

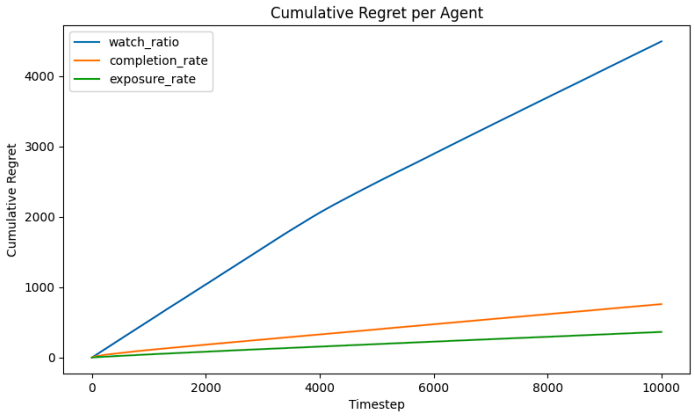
3.5 Model Structure

This part provides an illustration of the experiment's procedure, covering the baseline model and several multi-agent setup variations. A multi-agent heterogeneous setting (the intended structure for the experiment), a multi-agent setting with uniform agents (comparison group), and a single agent setting that employs a mean aggregation on various reward objectives (baseline) comprise the three main configurations of the experiment. Each reward objective in the heterogeneous environment is given a distinct algorithm: LinUCB for exposure_rate, Contextual TS for completion_rate, and Bayesian Linear TS for watch_ratio. Each agent is designed to improve on their own reward objectives, with some techniques like the criticality score that dynamically reweight agent influence according to recent performance, and the soft conflict penalty (the diversity constraint) helping them coordinate with each other and ease reward conflicts. Then, the final recommendation list is generated by aggregating candidates across agents, weighted by both criticality and selection rank. In addition to the heterogeneous environment, a similar workflow approach is used to build the multi-agent scenario with the same Bayesian Linear TS agent. Finally, a single Bayesian Linear TS agent configuration has been created, which is similar to the strategy of combining objectives first and then optimizing for that one feature.

4 Results and Analysis

This section summarizes the experimental findings under three distinct conditions: (1) a baseline single-agent model with aggregated rewards; (2) multi-agent homogeneous algorithms (Bayesian Linear TS for all objectives); and (3) multi-agent heterogeneous algorithms. The GGI evolution, average reward trends, suggestion variety, cumulative regret, and instantaneous regret are used to evaluate performance.

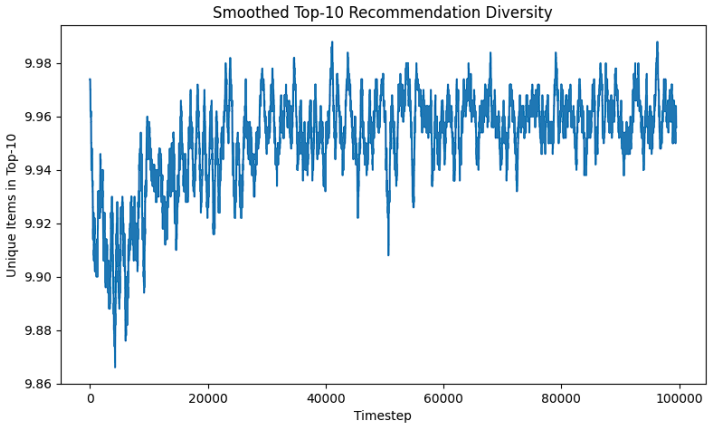
4.1 Heterogeneous Multi-Agent Setting



(a)

(b)

(c)



(d)

Fig. 2 (a) Heterogeneous setting cumulative regret (Picture credit: Original), (b) Heterogeneous setting instantaneous regret (Picture credit: Original), (c) Heterogeneous setting reward curve (Picture credit: Original), (d) Heterogeneous setting top-10 recommendation list diversity (Picture credit: Original)

As can be seen from the instantaneous regret curves and the cumulative regret graph in Fig. 2, the cumulative regret for all agents gradually decreases over time in this scenario where each agent is allocated a distinct method. Due to the inherent disparity in the difficulty of optimizing each reward feature, the reward trajectories show observable differences in mean reward levels across agents: peaked distributions result in higher average rewards, while features with broader optimal choices tend to yield lower average rewards.

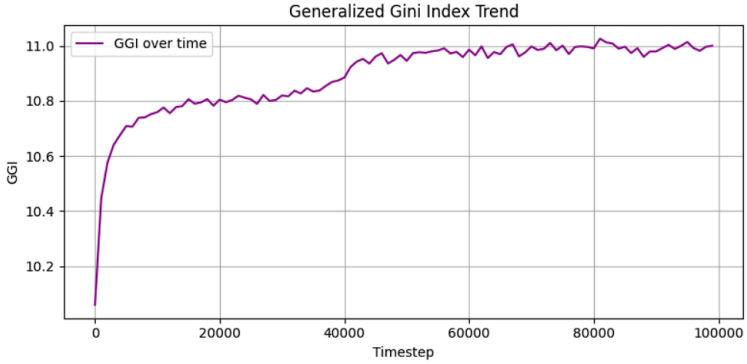
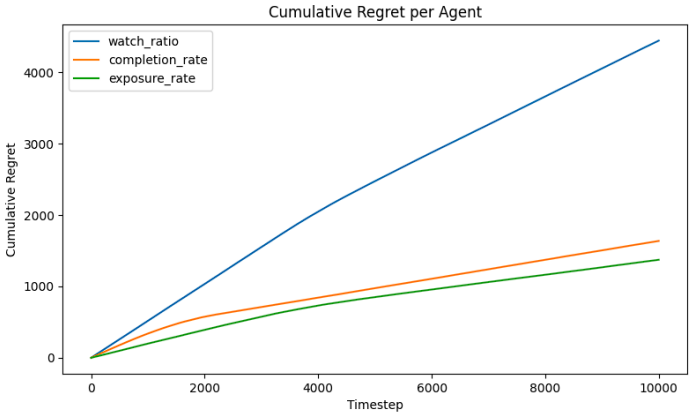


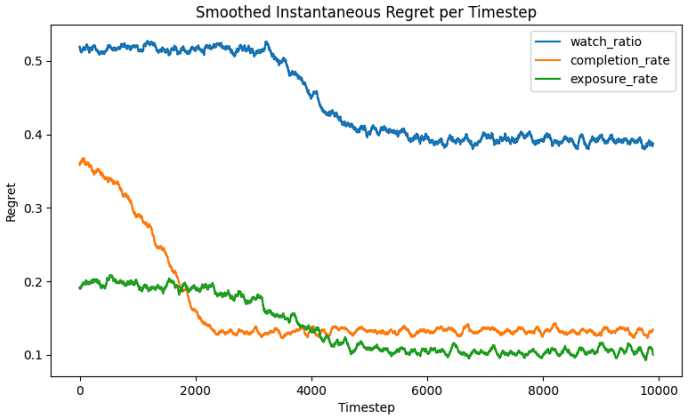
Fig. 3 Heterogeneous setting GGI trend (Picture credit: Original)

Throughout the simulation, the top-10 list for this configuration maintains a significant degree of variation in recommendations. This could be because agents favor different things due to algorithmic heterogeneity and distinct contextual feature spaces. The system swiftly approached a stable state in about 40,000 steps, according to the GGI curve in Fig. 3. The observed GGI variations and the instantaneous regret of the Bayesian Linear TS agent both show slight fluctuations after that, especially around step 40,000, which are probably due to the agent's ongoing improvement.

4.2 Homogeneous Bayesian Linear TS Multi-Agent Setting

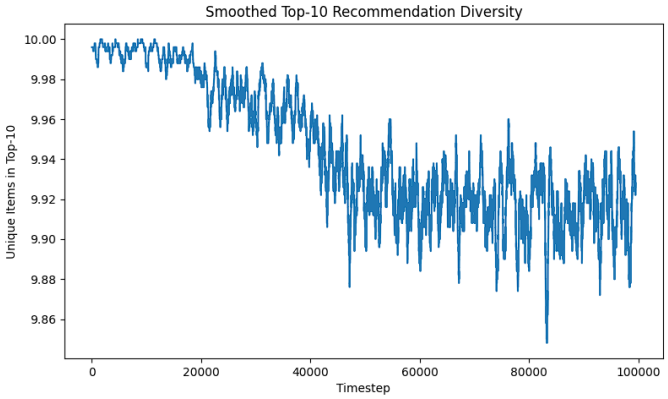


(a)



(b)

(c)



(d)

Fig. 4 (a) Homogeneous setting cumulative regret (Picture credit: Original), (b) Homogeneous setting instantaneous regret (Picture credit: Original), (c) Homogeneous setting reward curve (Picture credit: Original), (d) Homogeneous setting top-10 recommendation list diversity (Picture credit: Original)

Every agent in this scenario optimizes using the Bayesian Linear Thompson Sampling approaches. Fig. 4 illustrates how the reward per agent and cumulative regret graphs resemble the version in a heterogeneous context. In contrast to the heterogeneous setting, the instantaneous regret graphs for "exposure_rate" and "completion_rate" of the homogeneous setting show slower convergence, indicating that Bayesian Linear TS would need extra steps to properly adjust to various reward landscapes.

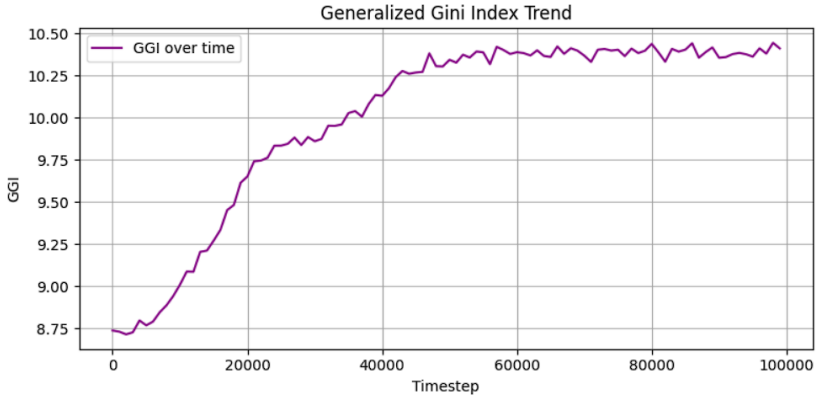
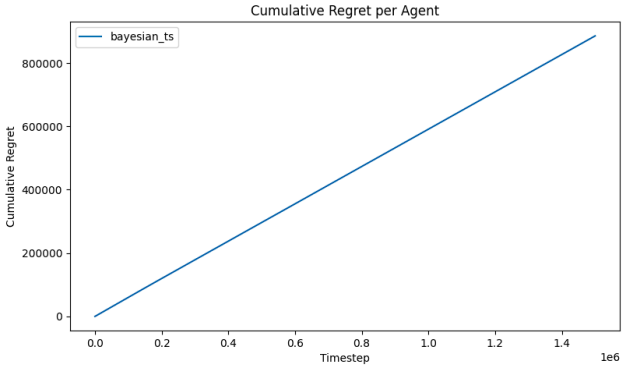


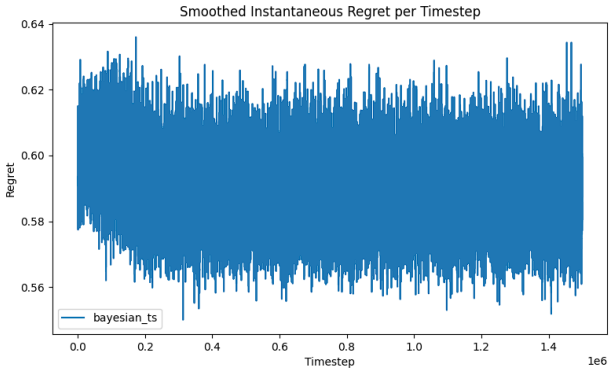
Fig. 5 Homogeneous setting GGI trend (Picture credit: Original)

When comparing homogeneous and heterogeneous settings for the recommendation list diversity graph, the former exhibits less diversity for recommendations. Because every agent uses the same learning algorithm, they all have a tendency to choose identical content, which lessens the variety of suggestions. Finally, as shown in Fig. 5, the GGI of this setting exhibits a continuous trend of increasing in the first 40000 steps and reaches a slightly lower value compared to the heterogeneous setting because of the similarities in learning mechanisms across homogeneous agents. This is because Bayesian Linear TS has a slower convergence speed than algorithms like LinUCB and contextual TS.

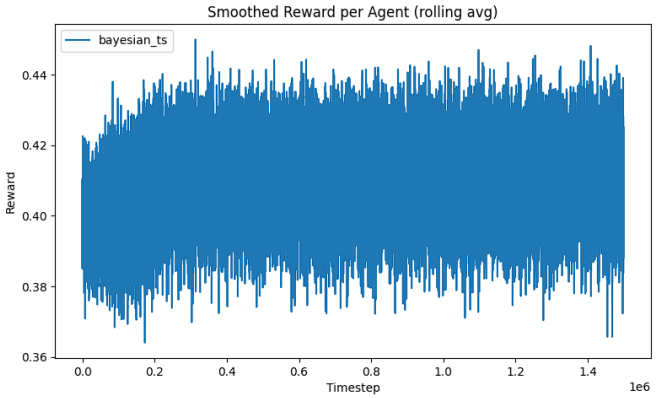
4.3 Single Bayesian Linear TS Agent Baseline (Mean-Aggregated Reward)



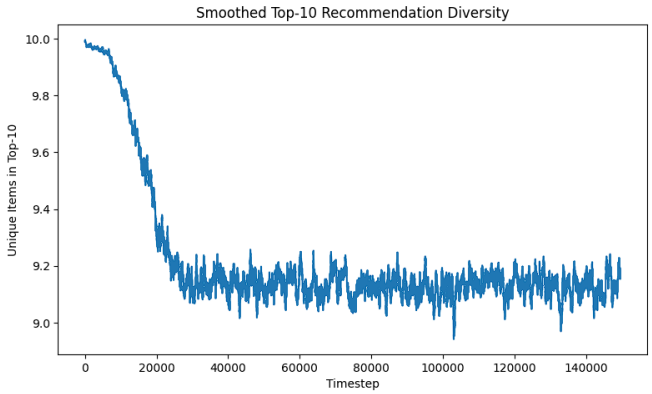
(a)



(b)



(c)



(d)

Fig. 6 (a) Single Bayesian Linear TS Agent cumulative regret (Picture credit: Original), (b) Single Bayesian Linear TS Agent instantaneous regret (Picture credit: Original), (c) Single Bayesian Linear TS Agent reward curve (Picture credit: Original), (d) Single Bayesian Linear TS Agent top-10 recommendation list diversity (Picture credit: Original)

This configuration serves as a reference point for comparing multi-agent contextual MAB configurations. The single agent finds it difficult to learn from such a complicated dataset, where there may be several optimal choices, if it merely averages the three reward features together. Fig. 6 illustrates how the cumulative regret increases linearly, with immediate regret being very noisy. While there is a small improvement in the beginning, the regret soon reaches a plateau and thereafter simply fluctuates randomly.

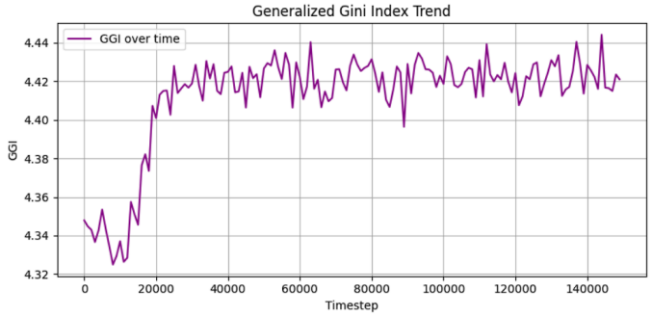


Fig. 7 Single Bayesian Linear TS Agent GGI trend (Picture credit: Original)

The significant decline in diversity indicates that the top-10 suggestion list quickly converges to a smaller number of things. Because the awards were purposefully averaged, the GGI value for this configuration is noticeably low, as shown in Fig. 7, which is in line with the equity-focused character of the GGI. The low GGI score, however, highlights a significant drawback of this configuration: the single-agent system's incapacity to dynamically prioritize more important reward targets when called for. Even though the GGI score rises, the recommendation system's functionality and adaptability are compromised.

5 Conclusion

With an emphasis on balancing varied reward optimization goals, this work examined the use of multi-agent contextual bandits for multi-objective recommendation systems. A

baseline single-agent model maximizing a mean-aggregated reward, a homogeneous multi-agent system utilizing Bayesian Linear Thompson Sampling across all agents, and a heterogeneous multi-agent system using different algorithms for various purposes were the three experimental configurations that were evaluated. Faster convergence, efficient specialization across objectives, and greater suggestion diversity were all displayed in the heterogeneous multi-agent setting. This configuration allowed agents to more accurately adjust to their individual optimization tasks by customizing the algorithm selection to the statistical characteristics of each reward attribute. However, managing inter-agent disputes and coordination made it more difficult to dynamically balance several goals.

Although it was easier to deploy, the homogeneous multi-agent setup showed less variation in recommendations and slower convergence in some objectives. The trade-off between model simplicity and system-wide diversity and specialization was highlighted by the overlapping selections that resulted from using the same procedure across agents. Because it directly optimized an averaged reward signal, the single-agent baseline had the lowest GGI score. However, as seen by near-linear cumulative regret accumulation and early stall in instantaneous regret reduction, this setting did not adaptively prioritize among objectives and displayed poor overall learning behavior. Overall, no one setting was found to be consistently ideal. The heterogeneous multi-agent system is especially well-suited for settings where reward distributions differ greatly between objectives because it provides greater flexibility and quicker adaptability. More simplicity is possible in homogeneous multi-agent setups, but redundancy may be an issue. Although computationally efficient, single-agent systems that optimize mean rewards forego adaptability and complex multi-objective trade-offs. Therefore, application goals, resource limitations, and the type of reward distributions encountered should all be taken into consideration when choosing one of these solutions.

References

1. Wang, L., Zhang, J.: 'Adaptive Multi-Armed Bandit Learning for Task Offloading in Edge Computing', arXiv preprint arXiv:2306.05856, 2023
2. Chen, X., Hou, I.H.: 'Contextual restless multi-armed bandits with application to demand response decision-making'. Proc. IEEE 63rd Conf. Decision and Control (CDC), December 2024, pp. 2652-2657
3. Wakayama, S., Ahmed, N.: 'Observation-Augmented Contextual Multi-Armed Bandits for Robotic Exploration with Uncertain Semantic Data', arXiv preprint arXiv:2312.12583, 2023
4. Zafar, S., Feraud, R., Blavette, A., et al.: 'Decentralized smart charging of large-scale evs using adaptive multi-agent multi-armed bandits'. Proc. IET Conf., June 2023, pp. 1130-1134
5. Ma, Y., Jiang, F., Zhao, Z., et al.: 'Locally Private Nonparametric Contextual Multi-armed Bandits', arXiv preprint arXiv:2503.08098, 2025
6. Qin, H., Jun, K.S., Zhang, C.: 'Achieving adaptivity and optimality for multi-armed bandits using Exponential-Kullback Leiblier Maillard Sampling', arXiv preprint arXiv:2502.14379, 2025
7. Wakayama, S., Candela, A., Hayne, P., et al.: 'Active Inference in Contextual Multi-Armed Bandits for Autonomous Robotic Exploration', arXiv preprint arXiv:2408.04119, 2024

8. Zhou, A., Beyah, R., Kamaleswaran, R.: 'NeuroSep-CP-LCB: A Deep Learning-based Contextual Multi-armed Bandit Algorithm with Uncertainty Quantification for Early Sepsis Prediction', arXiv preprint arXiv:2503.16708, 2025
9. Gao, C., Li, S., Lei, W., et al.: 'KuaiRec: A fully-observed dataset and insights for evaluating recommender systems'. Proc. 31st ACM Int. Conf. Information & Knowledge Management, October 2022, pp. 540-550
10. Busa-Fekete, R., Szörényi, B., Weng, P., et al.: 'Multi-objective bandits: Optimizing the generalized gini index'. Proc. Int. Conf. Machine Learning, PMLR, July 2017, pp. 625-634